

HEWLETT PACKARD  
**JOURNAL**

July 1994



HP.com



## table of contents

August 1994,  
Volume 45, Issue 4

### Articles

- 1 **An Advanced Scientific Graphing Calculator,**  
*by Diana K. Byrne, Charles M. Patton, David Arnett,  
Ted W. Beers, and Paul J. McClellan*

---

- 2 **HP-PAC: A New Chassis and Housing Concept  
for Electronic Equipment,**  
*by Johannes Mahn, Jurgen Haberle, Siegfried Kopp,  
and Tim Schwegler*

---

- 3 **High-Speed Digital Transmitter  
Characterization Using Eye Diagram Analysis,**  
*by Christopher M. Miller*

---

- 4 **Thermal Management in Supercritical Fluid  
Chromatography,**  
*by Connie Nathan and Barbara A. Hackbarth*

---

- 5 **Linear Array Transducers with Improved  
Image Quality for Vascular Ultrasonic Imaging,**  
*by Matthew G. Mooney and Martha Grewe Wilson*

---

- 6 **Structured Analysis and Design in the Redesign  
of a Terminal and Serial Printer Driver,**  
*by Catherine L. Kilcrease*

---

- 7 **Data-Driven Test Systems,**  
*by Adele S. Landis*

---

# An Advanced Scientific Graphing Calculator

The HP 48G/GX combines an easy-to-learn graphical user interface with advanced mathematics and engineering functionality, expanded memory capability, and seven new plot types.

by Diana K. Byrne, Charles M. Patton, David Arnett, Ted W. Beers, and Paul J. McClellan

The HP 48G/GX, Fig. 1, is a state-of-the-art graphing calculator that combines an easy-to-learn graphical user interface with advanced mathematics and engineering functionality. It is a continuation of the HP 28S<sup>1</sup> and HP 48S/SX<sup>2</sup> series of calculators, which are designed for high power, extendability, and customizability.

The HP 48G/GX includes improvements to address the needs of both novice and advanced users of scientific and graphing calculators. For the new user or the user who does not use certain functionality very often, the calculator has a dialog-box-style, fill-in-the-blanks user interface.



**Fig. 1.** HP 48GX scientific graphing calculator.

For the user who needs to do advanced problem solving, the calculator offers the following features:

- Differential equation solvers
- Polynomial root finder
- Financial problem solver
- Library of engineering equations and constants
- Fourier transforms
- Matrix manipulations
- Linear algebra operations.

For the user who needs more memory and extendability, the GX version has 128K bytes of built-in RAM, compared to 32K bytes in the S and G versions. The only other difference between the G and the GX is the two memory card slots in the HP 48GX. The second memory card slot accepts up to 4M bytes of RAM or ROM.

The graphing capability has been expanded with the addition of seven new plot types, for a total of fifteen. The HP 48S/SX has function, polar, parametric, conic, truth, histogram, bar, and scatter plots and the HP 48G/GX has these plus differential equation, slope field, wireframe, parametric surface, grid map, pseudocontour, and y-slice cross-section plots.

The HP 48S/SX functionality has been retained in the HP 48G/GX. The new calculator has twice the ROM and retains much of the original code. The HP 48S/SX and HP 48G/GX both have the following features:<sup>2</sup>

- Unit management
- MatrixWriter
- EquationWriter
- HP Solve numeric solver
- RPN-style stack calculation
- Symbolic mathematics
- Time and alarms
- Statistics operations
- Variables and directories for data storage
- User-definable keyboard and custom menus
- RPL programming language
- Two-way infrared communications link
- RS-232 serial cable connector.

## Education Trends

The creation of the HP 48G/GX can be traced to the American Mathematical Society (AMS) meeting in January 1992. We had been closely following the trend of using technology in the mathematics classroom because our software team

includes former mathematics educators and a calculus textbook author, and because we visit mathematics, engineering, and education conferences and talk to educators both to promote existing products and to find out what teachers and students would like to see in future products.

We had watched the interest in graphing calculators grow steadily each year, and had been involved in workshops for teachers using technology in the classroom through an HP grants program. Although the HP 48S/SX was becoming a standard for engineering students and professional engineers, it was just the first step in meeting the needs of the education community, and we continued to hear that it was too difficult to use and too expensive for classroom use. By January 1992, when we talked to educators attending the AMS meeting about their needs and about the calculators that they were considering for use in the classroom, it became obvious that we needed to have a new product for the education market no later than the 1993 back-to-school period. This resulted in the formation of an education advisory committee, a group of six mathematics professors who would help us design the calculator to fit the needs of the education community, and then give us feedback on our implementation.

### **Design Objectives**

Our number one objective for the new calculator was ease of learning. The users of the HP 48S/SX told us that they appreciated its power, but its complexity made it difficult for both novice users and experienced users. The novices tended to be intimidated by the extensive owner's manual and the difficulty of mastering so many new operations, while the experienced users had a hard time remembering how to use some operations that they did not use frequently.

Creating a state-of-the-art graphing calculator was our second objective. We needed to add some graphing capabilities, such as tracing along a graph and shading between graphs, just to maintain parity with other graphing calculators, but we also wanted to go far beyond the competition with features such as 3D graphing and animation.

Our third objective was to enhance the high-end mathematics capability of the HP 48S/SX with the addition of features such as differential equation solving, polynomial root finding, more matrix operations, and Fourier transforms, thereby strengthening our position as the most powerful technical calculator in the world.

By offering two models, the HP 48G and the HP 48GX, we intended to please customers at both ends of the education spectrum. The HP 48G has a list price that represents a substantial decrease from the list price of the HP 48SX or HP 48S. This makes the HP 48G competitive with other graphing calculators and makes it appealing even at the high school level. The HP 48GX has more appeal for college students, with four times as much built-in memory and two plug-in card ports that are expandable to 4M bytes of memory.

### **Operating System**

A calculator or computer operating system is primarily a set of conventions for memory organization, data structures, and resource allocation together with a set of software tools to aid in performing operations in accordance with those conventions. In contrast, an application is software built using

the resources and conventions of the operating system. As new hardware resources become necessary and available, the operating system must grow to manage those resources effectively and as transparently as possible to the applications built on the system.

The operating system (and system programming language) in the HP 48G/GX is the RPL operating system, first used in the HP 18C and HP 28C and subsequently in a number of other machines including the HP 28S, HP 48S/SX, and now, with extensions, in the HP 48G/GX.

### **HP 48G/GX Fundamentals**

The key concept underlying the operation of the calculator is the idea of objects on the stack. A stack is a data structure that is similar to a stack of cafeteria trays. The clean trays are added to the top of the stack, and as trays are needed, they are removed from the top of the stack. This type of last in, first out ordering characterizes the HP 48G/GX stack. All operations take their arguments (if any) from the stack and return their results (if any) to the stack.

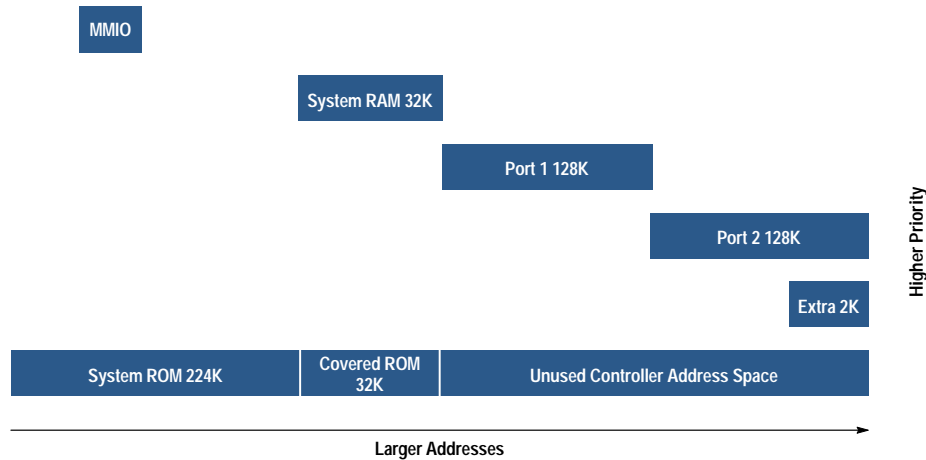
There is only one data stack in the HP 48G/GX. This resource is shared by the user and the system RPL programmer, who must take great care to make sure that any objects that belong to the user are preserved through the operation of system RPL programs. For example, the user may have a few numbers sitting on the stack, then decide to plot the graph of a function. The system RPL program that runs when the DRAW key is pressed does many operations that require the use of the stack, such as recalling the plotting parameters, checking that they are valid, calculating the range over which to plot, evaluating the user's function, and converting the function values to pixel coordinates. After the graph is complete (or if the drawing of the graph is interrupted by the user), when the user sees the stack again, the same numbers that were there to begin with should not have been disturbed.

Instead of trays, users may collect various types of numeric, symbolic, and graphic objects on the HP 48G/GX stack. The types of objects available in the HP 48G/GX include real and complex numbers, real and complex arrays, binary integers, names, characters, strings, tagged objects, algebraic objects, unit objects, and graphic objects. There are also backup objects, library objects, directories, programs, and lists. (HP 48 object types are discussed in more detail in reference 2.)

In a key-per-function calculator, there is a single key that the user needs to press to get the machine to perform any operation, such as cosine. The HP 48G/GX has many more operations than the 49 keys on the keyboard, so there needs to be a way to access all the functionality without assigning one operation to each key on the keyboard. This is accomplished through the use of menus and softkeys. The top row of keys on the keyboard do not have anything printed on them because they correspond to menu labels that appear along the bottom of the screen. These keys are called softkeys, and their meaning changes whenever the corresponding labels on the screen are changed by the software.

### **HP 48S/SX Memory Controller Configurations**

We will now discuss the memory controller configurations used in the HP 48S/SX and how these are used in implementing the various types of expanded address modes developed for these products. The next section outlines the



**Fig. 2.** Standard memory controller configuration for the HP 48S/SX calculator. Memory sizes are in bytes.

differences in configuration between the HP 48S/SX and the HP 48G/GX and discusses how these differences are used to extend and refine the expanded address technology to provide access to a total of 4.75M bytes of code and data as transparently as possible.

The CPU bus architecture first developed for the HP 71 and used in all HP calculators since that time has several useful features. One of the nicest is its address configuration capabilities. All chips attached to the bus are required to be able to change, on command of the bus, the range of addresses that evoke a response from the chips. Such a system eliminates, once and for all, the inconvenience and headache of configuring jumper switches on cards designed to plug into the machine. For a consumer product like a calculator this is not only a nicety, it is a necessity.

In the early days of the architecture (HP 71 to HP 28C), the CPU bus lines were actually routed around the circuit board and any RAM, ROM, or memory mapped I/O that was attached to the bus had to be custom-made with the bus interface attached. This had the advantage of allowing an arbitrary number of parts to be added to the system with assurance that the system would be capable of handling all of them in one way or another. It had the grave disadvantage of putting a price premium on such essential items as ROM and RAM.

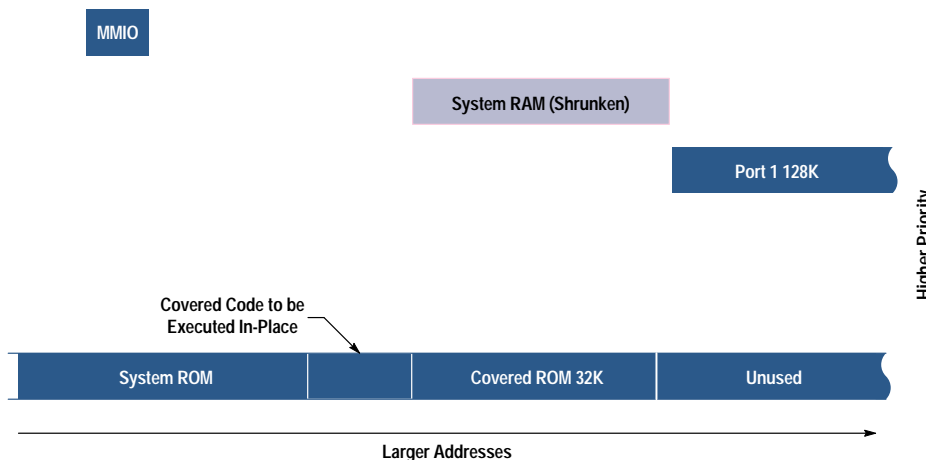
In the second-generation CPU chip, a fixed number of memory controllers were included onboard the CPU. The CPU bus was then, for all practical purposes, completely hidden within the CPU itself. The combination of external standard

RAM or ROM together with one of the internal memory controllers was then equivalent (so far as the CPU bus is concerned) to a standard bus device.

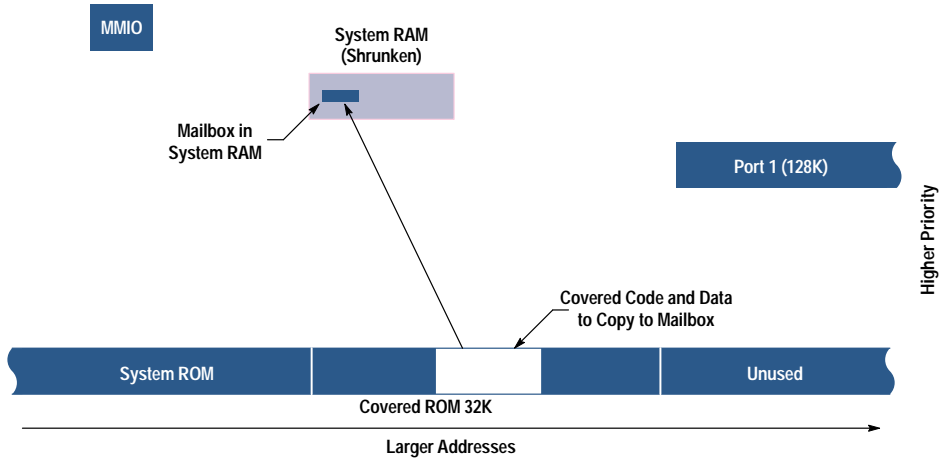
In the standard device implementations, the size of the device (that is, the address space occupied by the device) is designed into the device. In the second-generation chip, the size of the controllers was mask programmed at the time of manufacture since we knew exactly what size each controlled device would be.

With the advent of plug-ins for the HP 48S/SX, the configuration capabilities of the memory controllers had to be expanded to include varying the apparent size of the memory controller to conform with the device being plugged in. This is one of the many advanced features in the third-generation, HP 48S/SX implementation of the architecture. This resizing feature, in addition to allowing plug-ins of various sizes, also presented the opportunity to explore expanded address modes, which we have come to call the "covered" technology, for reasons that will be apparent shortly.

The third-generation CPU chip has six memory controllers. In the HP 48SX, these are allocated to memory mapped I/O, system RAM, port 1, port 2, and system ROM, and there is one extra controller. Their configuration in the usual state is shown in Fig. 2. The memory controllers are shown with their sizes and locations in the address space (00000h to FFFFFh). They are also pictured as having a vertical location in "priority space." In the CPU bus definition the devices are chained, with the result that devices closest to the CPU on



**Fig. 3.** Execute-in-place configuration for HP 48S/SX covered code.



**Fig. 4.** Copy-to-mailbox configuration for HP 48S/SX covered data.

the chain have the first opportunity to respond to bus requests. In consequence, if two devices are configured with overlapping address ranges, the one closer to the CPU on the chain effectively hides the more distant one. In Figs. 2 to 12, higher priority can be interpreted as “closer to the CPU” or “hides those below.”

As shown in Fig. 2, the memory controller for system RAM hides the section of ROM shown as covered. This is the reason for the name “covered” technology.

Fig. 3 shows more detail of the covered ROM and the first way in which it is used. In one section of the covered ROM there is assembly language code (mostly math routines) that requires no RAM resources outside the CPU for execution. This code is executed in-place in the covered ROM by shrinking and/or moving the memory controller for system RAM so that the relevant section of code is temporarily uncovered. When the routine finishes execution, system RAM is returned to its normal configuration.

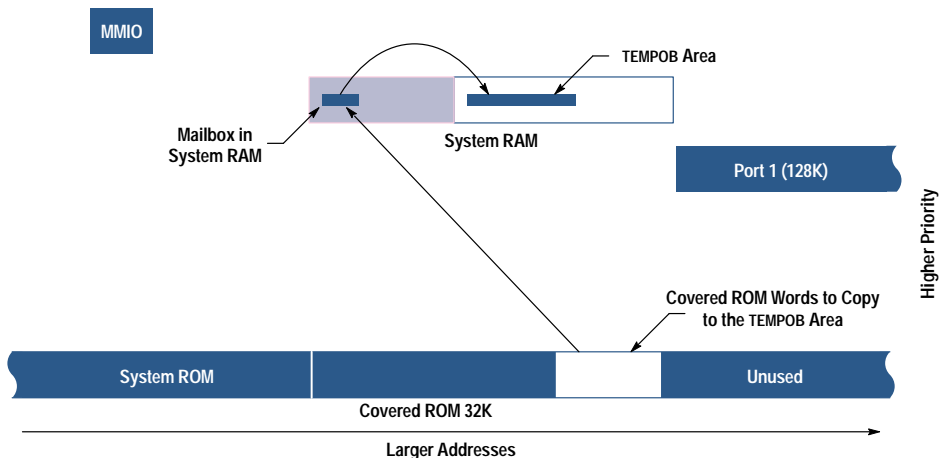
A second set of routines, all of which only need access to a fixed set of locations within system RAM, can execute with system RAM in any one of 16 locations, as long as they themselves are not currently covered by system RAM.

Fig. 4 shows a second way in which the covered ROM is used. In this case, code and data (mostly data) are copied from covered ROM to a mailbox at a fixed location in system RAM. After the copy is completed, system RAM is returned

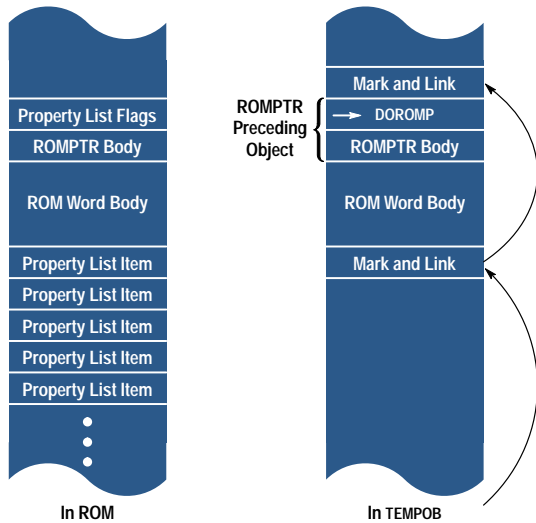
to its normal configuration and the code and data are available to the rest of the system. Coders using this data must remain aware that it is volatile and can be destroyed by another fetch of data from covered ROM. In this sense, this method is not transparent.

Another way in which covered ROM is used is shown in Fig. 5. It is as transparent as the execute-in-place method but entails fewer restrictions on the code and data that can be included. In the HP 48SX code, this system is usually tied to the execution of ROMPTRs. Recall that ROMPTRs are RPL objects that substitute for hard addresses of objects whose precise location is not known in advance (and in fact might not even be present.) They are midway between hard addresses that only change at compile/link time and identifiers whose corresponding objects may move between subsequent calls at run time.

If, during the conversion of a ROMPTR to an address, it is determined that the corresponding object lives in covered ROM, the object is copied from covered ROM, through the mailbox, to the TEMPOB (temporary object) area. The address of its new location in the TEMPOB area is then returned. Fig. 6 shows a comparison of a named ROM word (keyword or command) as it would exist in covered ROM and as copied to the TEMPOB area. Although we’ll refer back to Fig. 6 later, for now notice that in addition to the object itself, an additional piece is added to the image in the TEMPOB area. This piece is a ROMPTR preceding the object itself. This allows



**Fig. 5.** Copy-to-TEMPOB configuration for HP 48S/SX covered ROM words.



**Fig. 6.** Comparing the structure of a ROM word as resident in ROM and when copied to TEMPOB using covered technology.

the routine converting the ROMPTR to an address to check whether the object in question is a copy of one residing elsewhere. This method of covered ROM access, which we call “covered ROM word access,” will be especially relevant to our discussion of the HP 48G/GX.

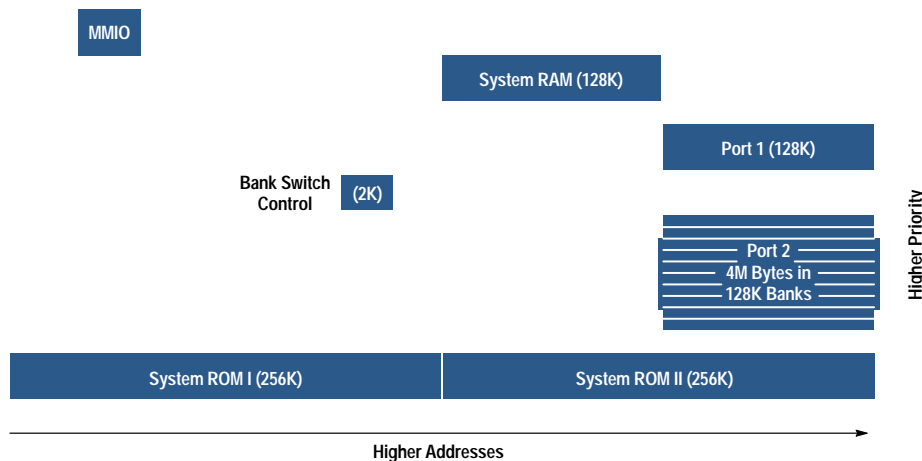
Preexisting design elements of the RPL system contributed greatly to the practicality and transparency of covered ROM word access, including:

- Encapsulation of code and data into RPL objects that are of determinable size
- Indifference of RPL object execution to object location in RAM or ROM
- Equivalence of direct and indirect execution of RPL objects, which allows (noncircular) structures to be stored and used in the same format.

### HP 48G/GX Memory Controller Configurations

The HP 48GX has a number of important features including:

- Up to 128K bytes of built-in system RAM
- One plug-in port electrically equivalent to the HP 48SX ports
- Access to 512K bytes of system ROM
- Access to 4M bytes of RAM or ROM at a second port using industry-standard parts.



**Fig. 7.** HP 48GX standard memory controller configuration.

These features required increasing the usable address space from 0.5M bytes to 4.75M bytes, an 850% increase over previous machines.

While the HP 48G/GX has CPU functionally equivalent to the third-generation CPU discussed above and thus has six memory controllers, these controllers are configured and used differently. Fig. 7 shows the standard HP 48GX configuration. The controller previously allocated to port 2 is now used as a bank switch control, and the extra controller is now allocated to port 2. Furthermore, there are now as many as 34 layers over the last 128K bytes of address space.

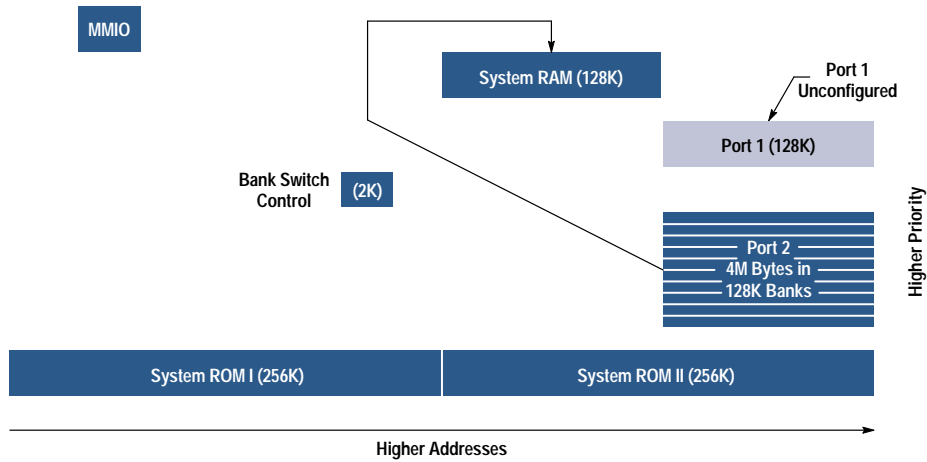
Eliminated in this configuration is the HP 48S/SX covered ROM. This means that all of the functionality included in the HP 48S/SX can be accessed more quickly. Two things that are visibly enhanced are plotting (since the math routines are not covered) and screen update (since the font bitmaps are not covered.) Since there are a great many more covered places to access, however, there are many more “temporary” configurations to keep track of while working with the covered data.

To simplify the system, we use only a single covered technique, namely, covered ROM word access, with appropriate modifications. Without this simplification, the number of access method and configuration combinations would be unmanageable. Moreover, this is the only feasible method of covered access to code written for the HP 48S/SX or not expressly written for the the new configuration.

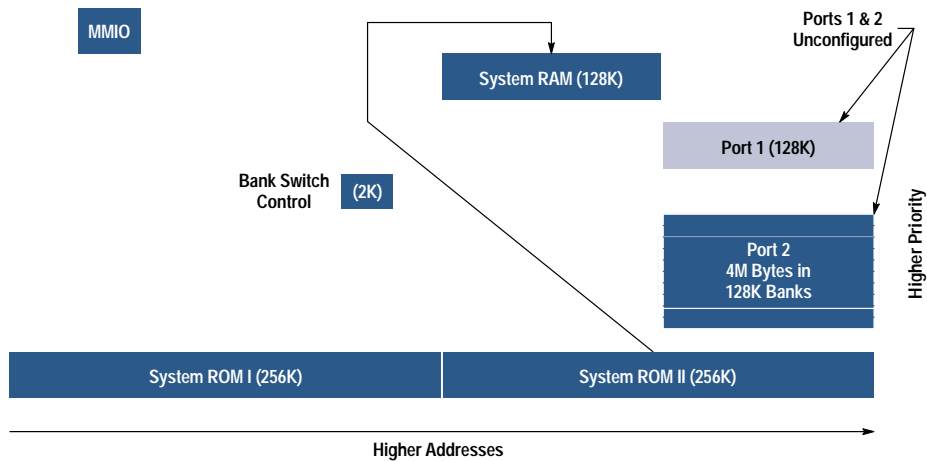
Fig. 8 shows the configuration while copying an object from a bank of port 2 to the TEMPOB area. Port 1 is unconfigured. In the unconfigured state, the controller responds to only a handful of bus commands and acts as if it weren't there for data access.

Fig. 9 shows the configuration while copying an object from the second half of the upper system ROM. In this case, both ports are unconfigured.

Fig. 10 shows the configuration while copying an object from the first half of the upper system ROM. Since a controller move or resize operation takes many more CPU resources than configure or unconfigure, it is often necessary to copy objects from this section, through a mailbox, and then into the TEMPOB area.



**Fig. 8.** HP 48GX configuration for copying an object from a bank of port 2 to TEMPOB.



**Fig. 9.** HP 48GX configuration for copying an object to TEMPOB from the upper half of system ROM.

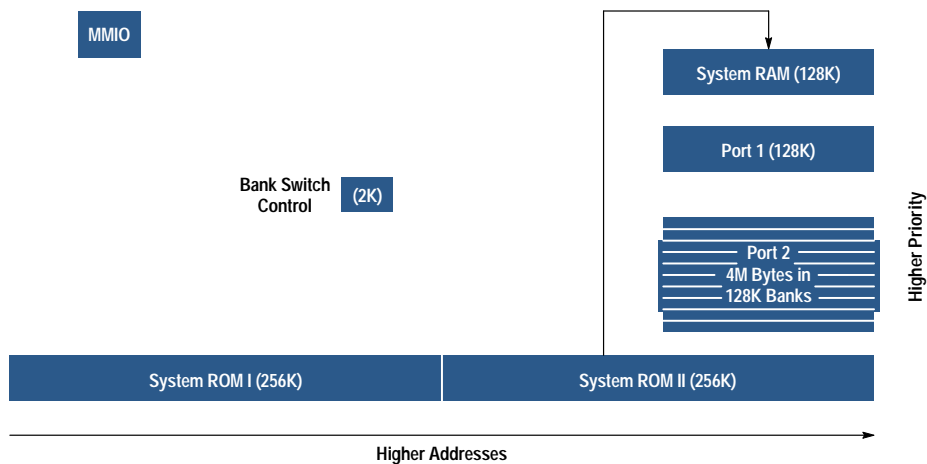
Fig. 11 shows the configuration when it is determined that nothing is plugged in at all. In this case, the only covered access is to the first half of the upper system ROM. Again, it is likely to be necessary to copy this material through a mailbox. Otherwise, all the ROM words can be executed in-place.

Fig. 12 shows the standard HP 48G configuration, which is identical to Fig. 11 except for the smaller size of system RAM. While it is not strictly necessary to use this configuration, which matches one of the HP 48GX configurations, there are advantages. First, it allows maximal code sharing

between the two machines. In fact, the code can be identical in this case. Second, it gains the advantage of faster access to the base functionality, providing a more responsive implementation.

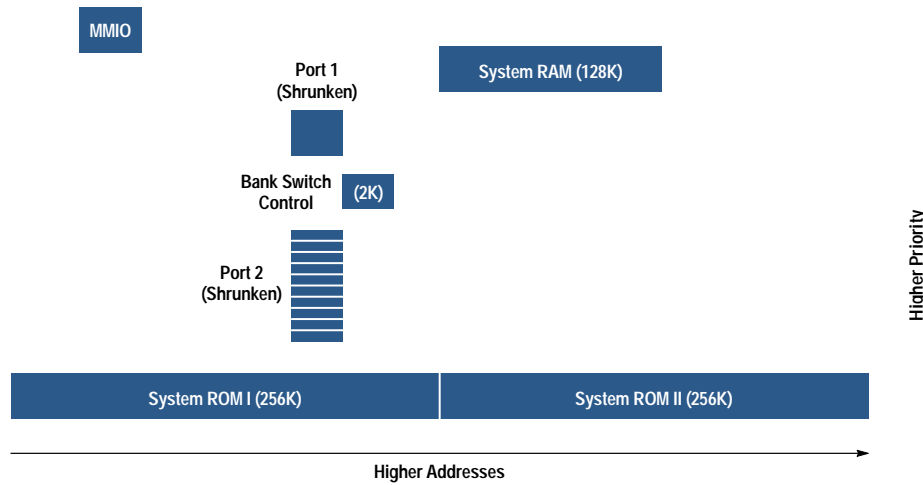
### Hardware Design

The heart of the HP 48G/GX is a fourth-generation CPU chip. This custom ASIC is built around the original HP 71 processor, and its development was key to the creation of



**Fig. 10.** HP 48GX configuration for copying an object to TEMPOB through a mailbox.





**Fig. 11.** HP 48GX all-ports-empty configuration.

the HP 48G/GX. This chip has four advantages over the third-generation chip used in the HP 48S/SX. First, it is produced using a different CMOS process, allowing better stability with onboard voltage regulation circuitry. Second, these improved voltage characteristics and several low-level optimizations allow the new CPU to operate at twice the speed of its predecessor. This speed increase gives it a 4-MHz bus rate. Third, the new CPU is packaged in a 160-pin quad flat-pack, improving the manufacturability of the HP 48G/GX. Fourth, with all these improvements, the final cost is lower, increasing the budget for other hardware improvements to the calculator.

The faster processing speed of the HP 48G/GX CPU gave the software team incentive to improve the user interface, implementing graphical routines that would not have been acceptable at the slower processing rate. This added functionality required an increase in data storage space, so we boosted the size of ROM and RAM. We also decided to add the facilities to bank-switch a data card plugged into card port two.

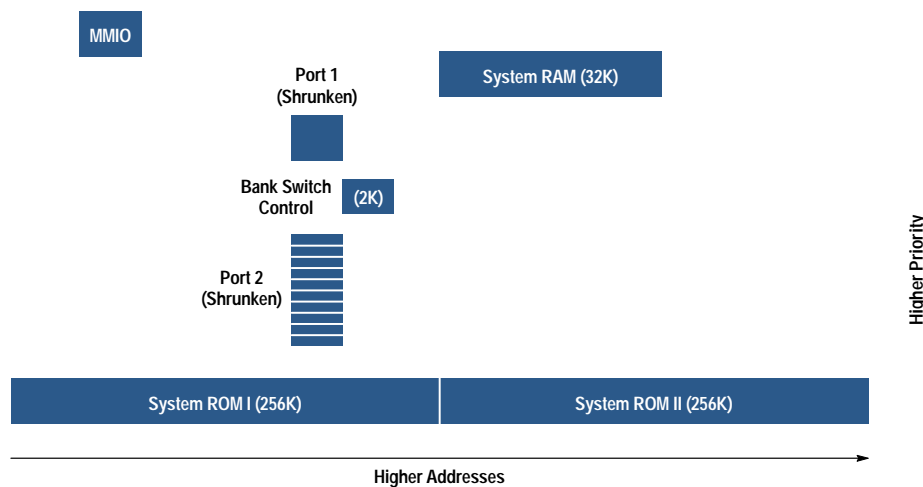
The HP 48G/GX circuitry, with its additional components, had to fit in the same physical space as in the HP 48SX. The product plan and schedule did not allow changes to production tooling or plastic parts except for those that were absolutely necessary. At times we felt like poets trying to write crossword puzzles. The HP 48SX circuit board design was optimized such that it did not leave us much free space.

These space constraints affected many of the HP 48G/GX hardware design choices.

The RAM increased from 32K bytes in the HP 48SX to 128K bytes in the HP 48GX, while the HP 48G retained the original 32K-byte chip. This difference between the G and the GX offers two advantages. First, it provides more differentiation between the functions and cost of the G and the GX, increasing the product family's market appeal. Second, the difference in RAM size provides a way for the calculator to know whether it is a G or a GX. If the calculator scans the RAM and finds only 32K bytes, then there will never be a plug-in data card installed. With this information the covered memory options become much simpler. The RAM memory size becomes an internal product type identifier, and several software routines are optimized for faster performance on the HP 48G.

### ROM Changes

The HP 48G and GX share a common ROM code set. They also share a common circuit board. While this simplifies documentation, manufacturing, and stock control, it also complicates some areas. The HP 48GX RAM chip is wider and longer than the chip used in the HP 48G: the 32K RAM is in a 28-pin small-outline package (SOP), and the 128K device is a 32-pin SOP. Both conform to the JEDEC pinout standard. A 128K device was chosen that has an extra chip select line at pin 30. This chip select is tied high, allowing pins 1 through



**Fig. 12.** HP 48G standard configuration.

28 of the smaller RAM to overlay pins 3 through 30 of the larger device. The extra chip select of the HP 48GX RAM matches the  $V_{dd}$  line of the HP 48G RAM chip, and all of the other lines are pinout-compatible.

The difference in physical package width also posed a problem. The foil patterns on the circuit board had to be modified to accept RAM chips with different lead spacings across the package. The immediate response was simply to stretch the oval-shaped patterns. However, this resulted in the foil extending well under the body of the 128K chip, a situation that could have led to solder bridging where the solder paste contacted the part body. This is avoided by using two different solder stencils on the manufacturing line. A paste of solder is laid on the blank circuit board before parts are loaded onto the board. A metal stencil defines the pattern of the solder paste, just as a silk screen controls the pattern of ink on a shirt. By using a different stencil pattern for the G and GX circuit boards, we control the original location of the solder paste and keep it out from under the 128K-byte RAM body. Once all the components are loaded onto the circuit board and the solder is heated to a molten state, a danger might again exist for the solder to flow under the part body. Fortunately, the nature of the mechanical contact between the RAM lead and the circuit board foil tends to cause the solder to pool or wick to the lead rather than spreading across the elongated foil pad.

The packaging of the ROM chip was also changed between the HP 48SX and the HP 48GX. The SX used a square 52-pin quad flatpack for its 256K bytes of program data. The code size of the HP 48GX is doubled to 512K bytes. Its package is a 32-pin SOP like the 128K-byte RAM chip. Their common package configuration allowed us to conserve space in the placement of the two chips and in the routing of signal wires between them.

Use of a standard SOP ROM chip also allowed us to use one-time programmable (OTP) ROMs in prototype calculators. An OTP uses the same semiconductor core as a UV-erasable EPROM. To get the semiconductor chip into an SOP, however, the manufacturer omits the familiar glass window in the chip, covering the device in opaque plastic. The resulting ROM is no longer erasable.

Typically, a product schedule requires months between code release and the start of production so that ROMs can be built with the software code built-in. The use of OTPs on this project cut the required time from months to days. For one prototype run, the time between code availability and product build was only a few hours.

The HP 48G/GX CPU multiplexes the highest address bit, A18, with an additional chip enable line, CE3. The original idea was to allow future expansion of the HP 48 family, either to use a larger ROM chip or to include an additional memory mapped device. By the time the HP 48G/GX design was complete, we had decided to do both. We doubled ROM to  $2^{19}$  bytes, and we added bank switching to card port 2. Two small HCMOS chips were added to the board to demultiplex these signals. The two chips are a quad NAND chip and a hex D flip-flop, similar to the standard TTL devices. The multiplexing is accomplished by simply toggling a control bit inside the HP 48G/GX CPU. To demultiplex the A18 and CE3 signals, we developed a protocol for mirroring the state of the internal bit to one of the external D flip-flops. The NAND

gates handle signal demultiplexing, and the remaining five flip-flops form a register for the card port 2 bank address.

### Other Hardware Changes

The card ports of the HP 48SX were designed for Epson memory cards. Several unused lines were adapted to provide external video signals to drive an enlarged display for classroom use. On the HP 48GX, the video lines are retained only on card port 1. On card port 2, the video lines are replaced by five additional address lines. The system software allows the card in port 2 to be subdivided into 128K-byte sections, with each section treated as a virtual plug-in card. Five bank select address lines permit up to 32 virtual plug-ins in card port 2, yielding a maximum card size of 4M bytes in the plug-in port. With the ROM, RAM, and plug-in options, an HP 48GX can access 4,980,736 bytes of onboard data.

Since the inception of the HP 48 family of calculators, liquid crystal display technology has progressed significantly. The display in the HP 48G/GX provides improved visibility by improvements in pixel contrast. The display is thinner than before. This change in glass thickness reduces the parallax between the pixel within the display and its shadow on the rear face of the display. In the HP 48SX, the pixel contrast was lower and the shadow was not dark enough to cause problems, but in testing the new HP 48G/GX display under various light conditions, we found that shadow effects made the display hard to read. With the thinner glass now used, the pixel and its shadow appear as one image, and the shadow now enhances the appearance of the pixel.

Changes to plastic parts were not permitted, except where necessary. The back case of the HP 48G/GX required changes. The changes were all accomplished by making mold inserts. Where text on the mold needed changes or additions, the affected area of the mold was milled away. A piece of steel was placed into the hole to make a perfect fit, and the face of this new piece was etched or inscribed with the new textures and features. The new back case helps identify the differences between card ports 1 and 2, updates the copyright information, adds a mark indicating that the HP 48G/GX complies with Mexico's importation laws, and adds an area for a customized nameplate. The customized nameplate is a piece of metal with adhesive on one side. The customer's name can be engraved on the plate and attached to an inset area of the back case. This is the same nameplate used on HP's palmtop computer family.

The result of these changes is a computer platform that is more powerful than its predecessor, is well-suited to the enhanced user interface developed by the software team, is more versatile for both the user and the design engineer, and is less expensive to produce. It started as a processor upgrade and became a major product improvement.

### User Interface

With ease of learning and ease of use the primary goals for the new HP 48G/GX calculator, the user interface and many built-in applications have been largely redesigned.

*Input forms* provide the common starting point for the new and rewritten applications in the HP 48G/GX. Looking much like dialog boxes in an Apple Macintosh or Microsoft®

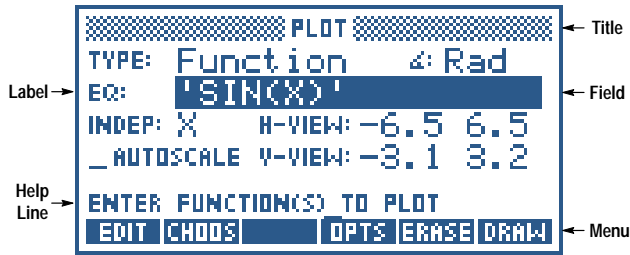


Fig. 13. A typical input form.

Windows PC, input forms provide a fill-in-the-blanks guide to the input needed for a task, plus application-specific menu keys for acting on that input.

For selecting an application in a particular topic and for picking an input from among several choices we developed *choose boxes*, a type of pop-up menu that suggests alternatives and narrows the input focus.

We designed *message boxes* to make feedback to the user more manageable within our increasingly crowded display space. Message boxes appear on top of whatever the user is working on and provide more flexibility for formatted messages and icons than the two-line, fixed-location error messages they replace. They also preserve the context that can otherwise be lost when something surprising happens within an application.

### Input Forms

An input form provides both a means to enter data pertinent to an application and operations that permit the user to direct actions.

Visually, an input form consists of (see Fig. 13):

- A title suggesting the form's purpose
- One or more fields, typically with explanatory labels, which are used to gather and display user input
- A help line that details the input expected in the selected field
- Menu keys that provide more options for working within or exiting the input form.

Each input form field can be one of four types. Most input forms, such as the Set Alarm and I/O Transfer input forms, contain several or all types of fields. *Text fields* are used to enter arbitrary HP 48G/GX objects like real numbers and matrices; the object types allowed are specific to each text field. In Fig. 14, a text field is used to enter an alarm message in the Set Alarm input form.

When a single choice among several is required, *list fields* are used to eliminate invalid input and to help focus user actions. To select an entry in a list field, a choose box is displayed. In Fig. 15, a list field is used to specify the transfer format in the I/O Transfer input form.

Sometimes only a simple yes-no, do-or-don't type of choice is needed. For this we use *check fields*. Fig. 16 shows how the overwrite (OVR) field is used to specify whether or not an existing variable should be overwritten.

Finally, when arbitrary input is possible but logical choices are also available, *combined text/list fields* are employed. In

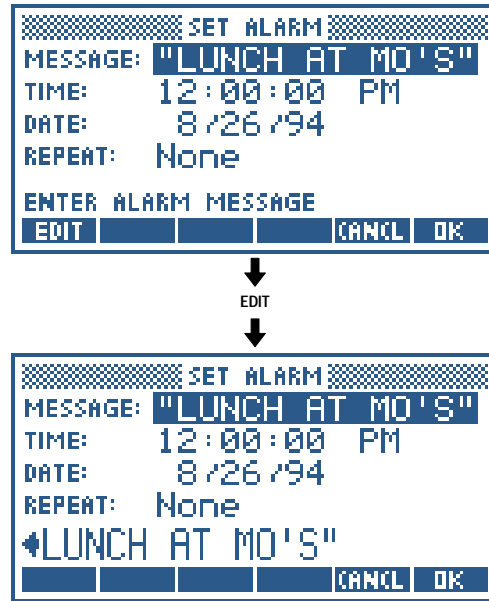


Fig. 14. Using a text field in an input form.

the Transfer input form, the Name field is a combined field that permits new names to be entered or the names of existing HP 48G/GX variables or PC files to be selected (see Fig. 17).

As the figures illustrate, each of the three base field types has associated with it a dedicated menu key that triggers the unique feature of that field type. This feature is an important part of how we maintained a calculator key-per-function-style interface within the constraints of a small display and with no pointing device. In other graphical user interfaces, visual elements such as list arrows are activated by mouse clicks to elicit different behaviors from fields. In the HP

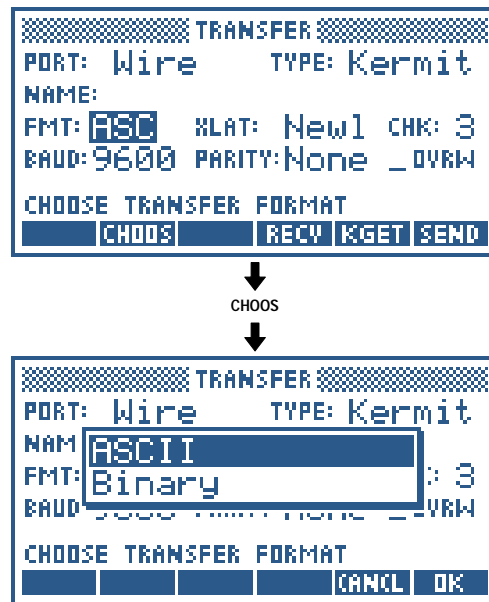


Fig. 15. Using a list field in an input form.



**Fig. 16.** Using a check field in an input form.

48G/GX, the user's finger acts as the pointing device, triggering the desired behavior by pressing the appropriate action button for each field. Consistent location of the three types of action buttons helps the user navigate an input form confidently.

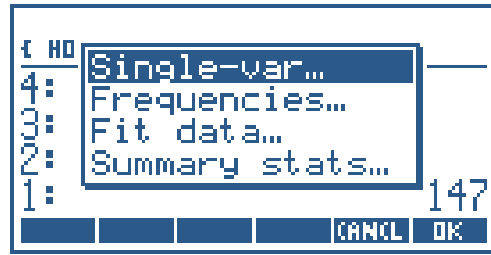
Some input form menu keys perform application-specific operations—for example, DRAW in Plotting. In the second row of the input form menu are more advanced input form operations for resetting a field or the entire form, displaying the object types allowed in a field, and temporarily accessing the user stack to calculate or modify a field value.

**Choose Boxes**

Choose boxes are used to make a choice in an input form list field. They are also used in most subject areas to choose



**Fig. 17.** Using a combined text/list field in an input form.



**Fig. 18.** A typical choose box.

a specific application from among several. Fig. 18 shows the choose box that is displayed when the STAT key is pressed to perform statistical calculations.

When circumstances require, choose boxes can include any or all of several advanced features. The Memory Browser application, for example, is actually a maximum-size choose box embellished with a title, multichoice capability, and a custom menu (see Fig. 19).

**Message Boxes**

Message boxes are used primarily for reporting errors that require attention before proceeding. For example, if the user attempts to enter a vector in the EXPR field of the Integrate input form, a message box appears to inform the user of the problem (see Fig. 20).

Some applications also use message boxes to give additional information. For example, in the Solve Equation input form, the user can press INFO any time after a solution has been found to review the solution and determine how it was calculated (see Fig. 21).

**Input Form Implementation**

For the HP 48S/SX, we developed an RPL tool called the parameterized outer loop<sup>3</sup> to speed development of new interfaces such as the MatrixWriter by automating routine key and error handling and display management. The input forms in the new HP 48G/GX embrace this concept—in fact, the input forms engine is a parameterized outer loop application—and take it one step farther to automate routine matters of application input entry and selection of options. The input forms engine brings a uniform interface to all new HP 48G/GX applications.

While narrowly focusing the task of application development by managing command input tasks, the input forms engine also leaves much room for the customization that helps optimize the HP 48G/GX for ease of use. Since an important measure of our progress towards our goals for the calculator



**Fig. 19.** The Memory Browser: a complex choose box.

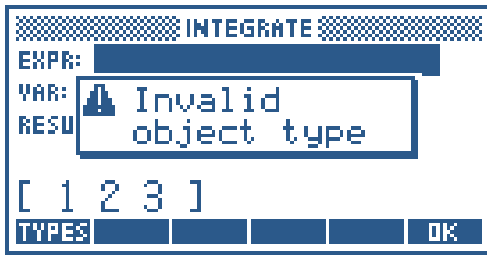


Fig. 20. A typical message box.

was to be feedback from typical users throughout the development cycle, we designed the input forms engine from the ground up to be highly customizable. This was accomplished in a programmer-friendly manner by including over fifty hooks into the input forms engine's responses to external and internal *events*. *External events* are triggered by users and include low-level events such as key presses and high-level events such as completion of a field entry. *Internal events* are usually activated by external events, such as formatting a completed field entry for proper display. A single external event can trigger a half dozen or more internal events, all of which are customizable.

Input form applications can customize any or all form-level events such as title display or field events such as displaying a help line. Each field has a *field procedure* associated with it, and the entire form has a *form procedure* associated with it. Whenever an event occurs, the appropriate field or form procedure is called with an identifying event number and perhaps additional information. If the procedure does not customize the event, it returns FALSE to the input forms engine. If it does customize the event, the procedure performs the custom behavior and returns TRUE. In this manner, every event first queries the proper form or field procedure to determine if custom behavior is needed, then handles the event normally only if it isn't customized. If a form or field has no custom behavior, it specifies a default procedure that quickly responds FALSE to all event queries.

The reason for a form procedure and multiple field procedures is to spread the burden of customization throughout the form. Since each field procedure only checks for the events that pertain to it, and since the form procedure only checks for form-level events, no single event processing is slowed by a highly customized form that would otherwise have to compare the event number against a lengthy list of event and field combinations.

For the HP 48G/GX project we needed another layer of regularity not enforced by the input forms engine. Because we sought and reacted to usability feedback almost until the

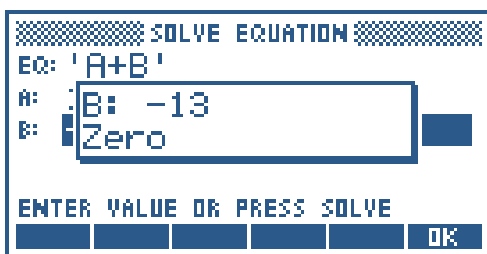


Fig. 21. The Solve Equation INFO message box.

code was released to production, the user interface details for each subject area were subject to constant change. It was imperative, therefore, that we maintain a strict and formal division between unchanging and well-understood tasks—such as getting and saving problem domain information and calculating results—and the user interface details that were changing regularly. We developed a set of conventions that were embodied in what we called *translation files*. We used naming rules and constrained responsibilities to greatly mitigate the effects of user interface changes on the underlying problem-solving functionality. For example, one RPL word in the plotting translation file has the simple task of reading the current horizontal plot range from calculator memory. Since the word has no presumptions about how and when it will be called, references to it could be (and often were) changed around as the fields populating the plotting input form were worked out.

### Choose Box Implementation

The choose box engine is very much like the input forms engine. For customization, the programmer can supply a *choose procedure* that responds to 26 messages.

A feature of choose boxes that simplifies their use is the option—heavily used by the built-in applications—of items that encapsulate both display and evaluation data. For example, when an angle measure—degrees, radians, or grads—is to be chosen in certain input forms, the choose box engine displays plain descriptions but returns an RPL program that sets the selected angle measure. This circumvents the need for branching according to the returned object and simplifies the extension of choices.

### Results: Benefits and Costs

Initial feedback from the educational advisory committee and user reviews suggests that the use of input forms and other graphical user interface elements has greatly improved the ease of use of the HP 48G/GX over the HP 48S/SX. However, the path we took to this accomplishment was more challenging than we planned.

Event customization, originally conceived as a means to extend the functionality of input forms in unforeseen ways, turned out to be a key component of our ability to prototype new user interface ideas rapidly. As their name may imply, the original intent of input forms was very modest compared to the role they now play. We designed input forms to be the standard means by which applications gather data for a task. One or more input forms would be displayed as necessary within the context of another, undefined, application context. This original concept is applied successfully throughout the calculator. For example, in the Memory Browser, when NEW is pressed to create a new variable, an input form is used to get the information required (see Fig. 22). In this context, the user can do only three things in the input form: enter data, cancel the form, or accept (OK) the form. This simple but effective behavior was the model used for the original input form design.

As the project developed, however, it became apparent that an input form could serve not only as a information gatherer but also as an action director. Input forms thus graduated from simple dialog boxes to full-fledged application environments.



Fig. 22. Memory Browser NEW input form.

Interestingly, no major changes to the input forms engine were necessary or even desirable to support their new role. Instead, the essence of input form functionality remained always data management, and the events customization was applied selectively where needed to enhance application forms.

In a similar manner, the event-driven choose box engine was eventually pressed into service as a powerful base for list-style applications like the Memory Browser.

The combination of lean, focused, standard feature sets for input forms and choose boxes and high customizability proved invaluable during the calculator design refinement. Throughout the middle portion of the project, when the basics had been settled but many user interface details were still unclear, we were able to prototype new ideas quickly and realistically by customizing event responses.

Translation files were another development effort that helped us keep the design and implementation moving forward. However, we learned over time that their overhead caused some duplication of code and inefficiency to creep into the interface between the input forms and the calculator main-frame. We addressed this issue where possible by making simple and safe code substitutions while leaving the interface concepts intact to enable high-confidence code defect fixes late in the project. In effect, we made a choice between maintainability and high performance that still remains a controversial topic among the HP 48G/GX developers.

### 3D Plotting

The functionality described in this section is a suite of 3D graphing and viewing utilities for the HP 48G/GX. We had several requirements to consider in creating these routines. Our aims were that they be psychologically effective and require only a small amount of code.

In exploring visualization techniques on a variety of machines we found that increasing “realism” (ray-traced, Phong-shaded, hidden-line, etc.) in the graphical presentation of functions of two variables did not necessarily correlate with increasing ease of comprehension. The HP 48G/GX routines represent the results of some of these experiments (including time-to-completion as an important factor).

All of the 3D plotting routines are intended as seamless extensions of the other built-in plotting utilities. In particular, they share the same standard user interface and are selected as alternative plot types. The 3D plotting routines are SLOPEFIELD, WIREFRAME, YSLICE, PCONTOUR, GRIDMAP, and PARSURFACE.

Like the other plotting routines, all the 3D plotting routines assume that the function of interest is stored in EQ.<sup>3</sup> Further, they assume, by default, that the function is represented as an expression in the variables X and Y—for example,  $u, v \rightarrow \sin(u+v)$  is represented as  $\text{SIN}(X+Y)$  in EQ. The use of other variable names is provided for by input form options or by the INDEP and DEPEND keywords.

While this section is titled “3D Plotting,” a better name would be “visualization techniques for functions of two variables.” This would cover the perspective view of the graph of a scalar function of two variables (WIREFRAME), the slicing view of a scalar function of two variables (YSLICE), the contour-map view of a scalar function of two variables (PCONTOUR), the slope interpretation of a scalar function of two variables (SLOPEFIELD), the mapping grid visualization of a two-vector-valued function of two variables (GRIDMAP), and the image graph of a three-vector-valued function of two variables (PARSURFACE).

Given this unity of purpose, there is considerable overlap in the global parameters (options) used in these routines. These plotting parameters are stored in the variable VPAR, analogous to PPAR.<sup>3</sup> The main data structure stored in VPAR describes the *view volume*, a region in abstract three-dimensional space in which most of the visualizations occur (see Fig. 23).

VPAR quantities controlling the view volume are:

- $X_{\text{left}}$  and  $X_{\text{right}}$ , controlling the width of the view volume
- $Y_{\text{far}}$  and  $Y_{\text{near}}$ , controlling the depth of the view volume
- $Z_{\text{low}}$  and  $Z_{\text{high}}$ , controlling the height of the view volume
- $X_e, Y_e,$  and  $Z_e$ , the coordinates of the *eye point*.

In addition to these, VPAR contains other quantities used by some of the routines. These are:

- $XX_{\text{left}}$  and  $XX_{\text{right}}$ , an alternative X input range, used for GRIDMAP and PARSURFACE
- $YY_{\text{far}}$  and  $YY_{\text{near}}$ , an alternative Y input range, used for GRIDMAP and PARSURFACE (note that this differs from the current Suite3D interpretation)
- $N_x$  and  $N_y$ , the number of X and Y increments desired, used in all of the routines instead of or in combination with RES.

#### SLOPEFIELD

The SLOPEFIELD plot type draws a lattice of line segments whose slopes represent the function value at their center point. Using SLOPEFIELD to plot  $f(x,y)$  allows your eye to pick out integral curves of the differential equation  $dy/dx = f(x,y)$ . It is quite useful in understanding the arbitrary constant in antiderivatives.

The number of lattice points per row is determined by  $N_x$  and the number of lattice points per column is determined

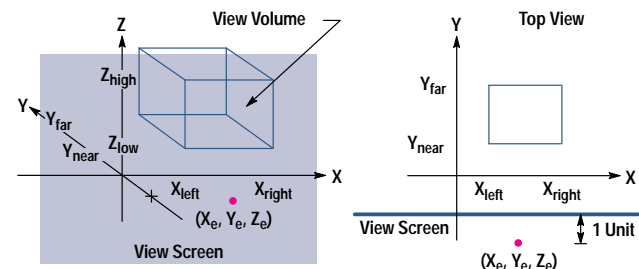
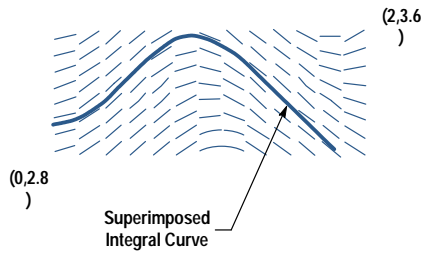


Fig. 23. VPAR parameters in relation to the view volume.



**Fig. 24.** SLOPEFIELD plot of  $dx/dt = \sin(xt)$ .

by  $N_y$ . The input region sampled is given by  $X_{left} < X < X_{right}$  and  $Y_{near} < Y < Y_{far}$ .

The input form in this case allows the user to:

- Choose or enter the defining expression for the function to be plotted
- Choose the names of the two variables (identical to INDEP and DEPEND)
- Choose  $X_{left}$  and  $X_{right}$  (default to their current value, or XRNG if no current value)
- Choose  $Y_{near}$  and  $Y_{far}$  (default to their current value, or YRNG if no current value)
- Choose  $N_x$  and  $N_y$  (default to their current value or 13 and 8 if no current value)
- Verify and/or choose RADIANS, DEGREES, or GRADS mode.

In trace mode for SLOPEFIELD, the arrow keys jump the cursor from sample point to sample point indicating both the coordinates of the sample point and the value of the slope at that point.

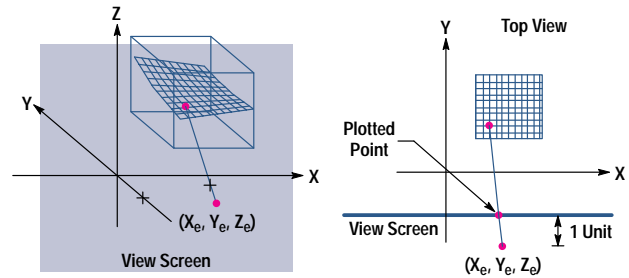
**Example Problem:** Determine graphically whether all solutions of the differential equation  $dx/dt = \sin(xt)$  with initial conditions  $3.0 < x(0) < 3.1$  satisfy  $2.8 < x(t) < 3.6$  for all  $t$  in  $[0, 2]$ .

**Solution:** Choose SLOPEFIELD plot type and enter  $SIN(X \cdot T)$  as the current equation. Choose  $T$  as the independent variable and  $X$  as the dependent variable. Choose 0 as  $X_{left}$ , 2 as  $X_{right}$ , 2.8 as  $Y_{near}$ , and 3.6 as  $Y_{far}$ . Verify RADIANS mode, and draw the result. As seen in Fig. 24, almost all of the integral curves in this region leave the window either through the top or the bottom. Therefore, not all the integral curves satisfy  $2.8 < x(t) < 3.6$  for  $t$  in  $[0, 2]$ .

### WIREFRAME

The WIREFRAME plot type draws an oblique-view, perspective, 3D plot of a wireframe model of the surface determined by  $z = f(x, y)$ . The function determined by the current equation is sampled in a grid with  $N_x$  samples in each row and  $N_y$  samples in each column. Each sample is perspective-projected onto the view screen along the line connecting the sample and the eye point (see Fig. 25).

Neighboring samples are connected by straight lines. The sampled region is determined by the base of the view volume ( $X_{left}$ ,  $X_{right}$ ,  $Y_{near}$ ,  $Y_{far}$ ). The region of the view screen represented in the PICT GROB (graphics object<sup>3</sup>) and hence on the display is determined by the projection of the view volume on the view screen (see Fig. 26).



**Fig. 25.** Perspective projection of a point in the view volume onto the view screen.

The input form in this case allows the user to:

- Choose or enter the defining expression for the function to be plotted
- Choose the names of the two variables (identical to INDEP and DEPEND)
- Choose  $X_{left}$  and  $X_{right}$  (default to their current value, or XRNG if no current value)
- Choose  $Y_{near}$  and  $Y_{far}$  (default to their current value, or YRNG if no current value)
- Choose  $Z_{low}$  and  $Z_{high}$  (default to their current value, or default YRNG if no current value)
- Choose  $X_e$ ,  $Y_e$ , and  $Z_e$  (default to their current value, or 0, -1, 0 if no current value)
- Choose  $N_x$  and  $N_y$  (default to their current value or 13 and 8 if no current value)
- Verify and/or choose RADIANS, DEGREES, or GRADS mode.

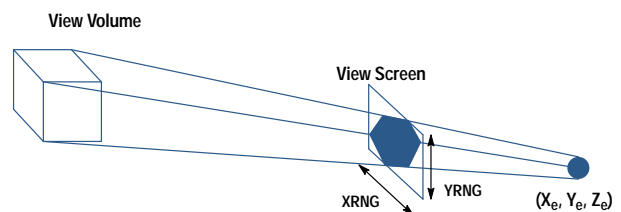
In trace mode for WIREFRAME, the arrow keys jump the cursor from sample point to sample point and the display indicates all three coordinates of the sample point.

**Example Problem:** Determine graphically whether the surface defined by  $z = x^4 - 4x^2y^2 + y^4$  is, at the origin, concave up, concave down, or neither.

**Solution:** Choose WIREFRAME plot type and enter  $X^4 - 4 \cdot X^2 \cdot Y^2 + Y^4$  as the current equation. Choose  $X$  and  $Y$  as the independent and dependent variables. Choose -1 for  $X_{left}$ , 1 for  $X_{right}$ , -1 for  $Y_{near}$ , 1 for  $Y_{far}$ , -1 for  $Z_{low}$ , and 1 for  $Z_{high}$  so that the view volume surrounds the origin. Choose 4 for  $X_e$ , -10 for  $Y_e$ , and 3 for  $Z_e$  to give a distant, oblique view of the graph. As seen in Fig. 27, the graph displays a “monkey saddle” which is neither convex nor concave at the origin.

### New Interactive Features

The picture environment, which is invoked automatically when graphs are drawn or by pressing the PICTURE key, allows the user to interact with a graph. The user can move



**Fig. 26.** Relationship of view volume and eye point to XRNG and



**Fig. 27.** WIREFRAME plot of the surface determined by  $z = x^4 - 4x^2y^2 + y^4$  with view volume  $[-1,1] \times [-1,1] \times [-1,1]$  and eye point  $(4,-10,3)$ .

the cross hairs around using the arrow keys, trace along the graph, add picture elements such as dots, lines, and circles, or do interactive calculus operations such as finding the derivative at the cross hairs location.

### Trace, Faster Cross Hairs

The HP 48S/SX cross-hair-moving code was rewritten for the HP 48G/GX. The cross hairs needed to be faster, to fade less as they moved, and to accommodate added functionality such as tracing and shading. The cross hair code originally came from the HP 28 and has been maintained and modified over the years. In the HP 28, most of the cross hair code was written in high-speed assembly language and actually contained a routine called SLOW which was buried deep within RPL subroutine calls. SLOW was needed to slow the cross hairs down to an acceptable speed. The use of this word was discovered during the process of porting the code from the HP 28 to run on the bigger HP 48S/SX display. There were many occasions during the HP 48G/GX project when we were trying to speed up various operations and wished we could just find the word SLOW and take it out!

The fading of the cross hairs as they moved was improved by changing the code so that the time between turning the cross hairs off at one pixel and turning them on at the next pixel is minimized. To do this, all the calculations required for moving are now done before turning the cross hairs off. Unfortunately, the new display on the HP 48G/GX trades off response time for contrast, so although it is brighter and has higher contrast than the HP 48S/SX display, it takes longer for pixels to turn dark. Thus, much of the work to reduce fade in moving cross hairs was canceled out by the new screen characteristics. The user can darken the display by holding down the **ON** key and pressing the **+** key a few times, and this will make the moving cross hairs easier to see.

Tracing along a graph with the cross hairs presents a challenge because the user's function must be evaluated at every point, so in effect the system RPL programmer must turn control over to the user at each point of the graph. This required careful attention to error handling and to managing the data stack, which is a shared resource. The procedures for tracing vary with the different plot types. The procedures are kept in the property list associated with each plot type, and then the appropriate procedure is passed in and evaluated when trace mode is turned on. It required quite a bit of rewriting to implement this object-oriented, extensible approach because much of the existing cross hair code had previously undergone rewriting and optimizing for speed and code size.

### Animation

The ANIMATE command is a program that was easy to write and that the user could have written in user-RPL programming language, but we added it for the sake of convenience. Also, it is used as part of Y-slice 3D plotting. It sets up a loop that repeatedly puts graphics objects into the PICT display area.

A quick way to get started with animation is to press PICTURE to go to the interactive graphics environment, where you will be able to create some pictures to animate. Press EDIT, then DOT+ to turn on the etch-a-sketch-style drawing mode in which pixels are turned on wherever you move the cross hairs. Using the up, down, left, and right arrow keys, sketch something, then press STO to send a copy of your picture to the stack. Continue sketching, press STO again, and repeat this procedure, continuing to add to your sketch until you have a handful of pictures on the stack, say six of them. Press CANCEL to leave the PICTURE environment, and you will see the the picture objects sitting on the stack. They are called GROBs, which is short for graphics objects.<sup>3</sup> To use the ANIMATE command, all you have to do is enter the number of GROBs (for example, press **6** then **ENTER** if you created six pictures), then press the ANIMATE key, which you will find in the GROB submenu of the PRG menu. Your series of sketches will come to life as the ANIMATE command flips through them.

### Mathematics

Several new mathematical features were added to meet the needs of the educational market and to match or exceed corresponding features recently introduced by our competition. Design trade-offs made for and inherited from earlier, less capable platforms were reconsidered, and relevant software developed for earlier machines but not used in the HP 48S/SX was used wherever appropriate.

### Design and Implementation Issues

The HP 48G/GX is targeted at the college-level mathematics, science, and engineering educational market. We hoped, also, to achieve more success in high school advanced placement courses. In these environments calculators are used as pedagogical tools, illustrating mathematical and modeling concepts introduced in the courses.

As a pedagogical tool, the calculator's accuracy and reliability are paramount design goals. Speed of execution is important but secondary to the validity of the computed results. To achieve high accuracy and reliability the computational methods needed to be more numerically sophisticated than typical textbook methods. This greater complexity is hidden from the casual user wherever possible, but made available to the sophisticated user so the methods can be tuned to their needs.

One means of achieving maximum accuracy and reliability is to read the current literature and consult with expert specialists to obtain the best methods, then implement those methods from scratch. We have employed this approach in the



## User Versions of Interface Tools

Although the primary focus of the new user interface for the HP 48G/GX was to enhance our built-in applications, it became apparent as the project progressed that calculator owners who program would want access to the same capabilities to enhance their efforts. For the choose box, message box, and especially the input form tools, the biggest challenge involved scaling back the numerous features to produce simple user commands that still offer customization potential.

The message box command, MSGBOX, was designed to display pop-up messages with a minimum of fuss. Thus, it takes just one argument—the message string—and produces a word-wrapped normal-sized message box.

The choose box command, CHOOSE, is slightly more complicated. To enable but not require the same object-oriented use of choose boxes as the built-in applications, the CHOOSE command accepts a list of items in two formats. In the simplest format, an item is specified by a single object, which is displayed and returned if chosen. In the alternate format, an item is specified by a two-element list object. The first element is displayed in the choose box, and the second element is returned if the item is chosen.

For simplicity of the user interface, CHOOSE displays a normal-sized choose box without the multiple-choice capability used by some built-in applications.

The MSGBOX and CHOOSE commands largely follow the same interface specification methods as their system-level counterparts. This differs markedly from the input form user command, INFORM. To maintain complete flexibility over all elements of form layout and behavior, the input forms engine takes three arguments for each label and thirteen arguments for each field, specifying such details as exact location and size, display format, and so on. Added to that are global arguments for the form procedure and form title and some other details. All together, an input form with four labeled fields requires 68 arguments. While this amount of information is justified for the varied needs of built-in applications, it is an unnecessary burden for programmers just wanting to get some simple input from the user.

For the INFORM command, therefore, we developed an automatic form layout scheme that serves most needs, with options for further detailing. Basically, the INFORM input form is viewed as a grid that is filled with fields starting in the upper-left corner and proceeding from left to right and top to bottom. The number of

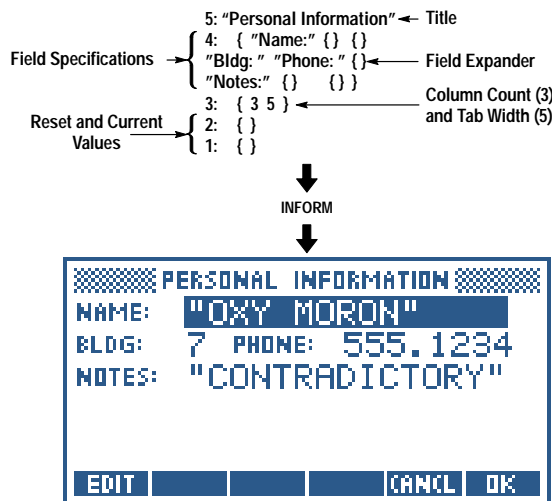


Fig. 1. A custom input form created by INFORM.

columns in the grid is specified as one of INFORM's arguments, and each field's width is determined by the width of its label and by the user-supplied tab width, which places invisible tab stops within each column to help align fields vertically. A field can span multiple columns with a special field-expander specification. Help text and object type restrictions can be included for any field, but aren't required.

Fig. 1 shows an example of a custom input form created by INFORM. Notice that, despite the relative simplicity of the input arguments, an input form with aligned fields of varying widths is presented. This technique for building input forms proved so valuable that it was used to create the Solve Equation input form, which changes according to the number and names of variables in the equation to be solved.

past with success, but it can be time-consuming, expensive, and risky.

Another approach sometimes available is to consult standard computational libraries used by the professional scientific community. Several such public-domain libraries are available that represent the current state of the art. In some development environments these libraries can be used directly. In others, they can at least provide high-quality methods and implementations that when judiciously used facilitate meeting tight development schedules at low cost. We found the LAPACK library<sup>4</sup> of FORTRAN 77 numerical linear algebra subroutines particularly helpful in this regard.

As usual, code was reused whenever possible to achieve timely and reliable implementations. In addition to the source code for the HP 48S/SX and its Equation Library card, we had implementations dating from the HP 71 Math Pac that were revised for the HP 48S/SX but didn't find ROM space in that product.

While reusing code, we took advantage of the HP 48G/GX CPU clock speedup and larger RAM environment over the HP 48S/SX to reconsider some of our previous implementation trade-offs in an effort to achieve greater accuracy. In

some cases we decided to employ more computational effort and to store intermediate values in higher precision to achieve more accurate results.

### New Mathematical Features

The HP 48G/GX includes many new mathematical features over those provided by the HP 48S/SX. These are array manipulations, additional linear algebra operations, a polynomial root finder and related operations, two differential equations solvers and associated solution plotters, discrete Fourier transforms, and financial loan computations.

The array manipulation commands are primarily pedagogical tools. These include a random array generator and commands to add or delete rows or columns of matrices or elements of vectors, decompose matrices into or create matrices from row or column vectors, extract diagonal elements from a matrix or create a matrix from its diagonal elements, perform elementary row and column operations, and compute the row-reduced echelon form of a matrix.

We significantly improved and expanded the linear algebra functionality of the HP 48G/GX over the HP 48S/SX. The determinant, linear system solver, and matrix inverter were

revised to be more accurate through additional computation and by storing all intermediate values in extended precision. We added a command to compute a condition number of a square matrix, which can be used to measure the sensitivity of numerical linear algebra computations to rounding errors, a command to compute a solution to an underdetermined or overdetermined linear system by the method of least squares, commands to compute eigenvalues and eigenvectors of a square matrix, commands to compute the singular value decomposition of a general matrix, and commands to compute related matrix factorizations and functions. These linear algebra commands accept both real and complex arguments and perform all intermediate computation and storage in extended precision.

The HP 48G/GX has commands to compute all roots of a real or complex polynomial, to construct a monic polynomial from its roots, and to evaluate a polynomial at a point. The polynomial root finder is a modification of the HP 71 Math Pac's PROOT command, extended to handle complex coefficients. It uses the Laguerre method with deflation for fast convergence and constrained step size and an alternate initial search strategy for reliability.

The HP 48G/GX has commands to compute the discrete Fourier transform or the inverse discrete Fourier transform of real or complex data. These commands were leveraged from the HP 71 Math Pac's FFT and IFFT commands, requiring the data lengths to be a nonzero power of 2, and were modified slightly to match the customary definitions of these transformations.

Finally, we included time-value-of-money commands. These commands have appeared in our financial calculators and were available on the HP 48SX Equation Library card. Since engineering feasibility studies must include at least rudimentary time-value-of-money computations it seemed useful to include these commands in the HP 48G/GX.

### Differential Equation Plotting

The HP 48G/GX contains two differential equation solvers and solution plotters. These solvers and solution plotters can be accessed via their input forms or invoked programmatically via commands. We provide a programmatic interface to the differential equation solvers and their subtasks so the user can use them with the calculator's general solver feature to determine when a computed differential equation solution satisfies some condition, or to implement custom differential equation solvers from their subtasks.

In implementing the differential equation solution plots, one challenge was to identify and implement good solution methods. Another challenge was to merge this new plot type with the new 3D plot types described earlier and with the existing HP 48SX plot environment in a backward-compatible manner.

The HP 48G/GX specifically solves the initial value problem, consisting of finding the solution  $y(t)$  to the first-order equation  $y'(t) = f(t,y)$  with the initial condition  $y(t_0) = y_0$ . Here  $y'(t)$  denotes the first derivative of a scalar-valued or vector-valued solution  $y$  with respect to a scalar-valued parameter  $t$ . Higher-order differential equations can be expressed as a

first-order system, so this problem is more general than it might at first appear.

Many solution methods have been developed over the years to solve the initial value problem. We decided to implement two methods, a Runge-Kutta-Fehlberg method for simplicity and speed of execution and a Rosenbrock method for reliability. The first method is easier to use, requiring less information from the user, but can fail on stiff problems.\* The Rosenbrock method requires more information from the user, but can solve a wider selection of initial value problems. Both initial value problem solution methods require the user to provide the function  $f(t,y)$ , the initial conditions, the final value of  $t$ , and an absolute error tolerance. The Rosenbrock method also requires the derivative of  $f(t,y)$  with respect to  $y$  (FYY) and the derivative of  $f(t,y)$  with respect to  $t$  (FYT).

All plot types use the contents of the variable EQ, typically to specify the function to be plotted. If the user selects the stiff (Rosenbrock) method the extra functions are passed to the solver by binding EQ to a list of functions  $f(t,y)$ , FYY, and FYT. Otherwise, EQ is bound to the function  $f(t,y)$  needed by the Runge-Kutta-Fehlberg method.

Both methods solve the initial value problem by computing a series of solution steps from the initial conditions towards the final value, by default taking steps as large as possible subject to maintaining the specified error tolerance. The solution plotter plots the computed values and by default draws straight lines between the plotted points. However, although the computed steps may be accurate, the line segments drawn between the step endpoints may poorly represent the solution between those points. The plot parameter RES is used by many plot types to control the plot resolution. If RES is zero the initial value problem solution plotter imposes no additional limits on the step sizes. If RES is nonzero the plotter limits each step to have maximum size RES.

For the scalar-valued initial value problem it is typical to plot the computed solution  $y(t)$  on the vertical axis and the parameter  $t$  on the horizontal axis. However, in the vector-valued case the choice of what is to be plotted is not as clear. The user may wish a particular component of the computed solution plotted versus  $t$  or may wish two components plotted versus each other. The HP 48G/GX allows the user to specify the computed scalar solution, any component of the computed vector solution, or the parameter  $t$  to be plotted on either axis. This flexibility was introduced into the plot environment by expanding the AXES plot parameter. Previously, this parameter specified the coordinates of the axes origin. This parameter was expanded so that an optional form is a list specifying the origin and the horizontal and vertical plot components.

By judiciously expanding the meaning of the various plot parameters we were able to accommodate the differential equation solution plot type while maintaining backward compatibility with previous plot types.

\* Stiff problems typically have solution components with large differences in time scale. More information is needed by a solver to compute a solution efficiently.

## Acknowledgments

We would like to acknowledge the rest of the software development team: Jim Donnelly, Gabe Eisenstein, Max Jones, and Bob Worsley. Bill Wickes also contributed software.

Clain Anderson and Ron Brooks from the marketing department were involved in the day-to-day design process and kept us informed about user needs. Dennis York oversaw both the R&D and marketing aspects of the project, which helped generate synergy between R&D and marketing. We would like to thank Dan Coffin, our manual writer, and John Loux from the technical support group for going out of their way to participate in the design work and for providing many valuable ideas.

## References

1. W.C. Wickes, "An Evolutionary RPN Calculator for Technical Professionals," *Hewlett-Packard Journal*, Vol. 38, no. 8, August 1987, pp. 11-17.
2. W.C. Wickes and C.M. Patton, "The HP 48SX Scientific Expandable Calculator: Innovation and Evolution," *Hewlett-Packard Journal*, Vol. 42, no. 3, June 1991 pp. 6-12.
3. T.W. Beers, et al, "HP 48SX Interfaces and Applications," *Hewlett-Packard Journal*, Vol. 42, no. 3, June 1991 pp. 13-21.
4. E. Anderson, et al, *LAPACK Users' Guide*, SIAM, Philadelphia, 1992.

Microsoft is a U.S. registered trademark of Microsoft Corporation.

# HP-PAC: A New Chassis and Housing Concept for Electronic Equipment

HP-PAC replaces the familiar metal chassis structure with expanded polypropylene (EPP) foam. Large reductions are realized in mechanical parts, screw joints, assembly time, disassembly time, transport packaging, and housing development costs.

by Johannes Mahn, Jürgen Häberle, Siegfried Kopp, and Tim Schwegler

Business competition between PC and workstation manufacturers has resulted in shortened life cycles for computer products, faster development and production times, and steadily decreasing market prices. The Hewlett-Packard Böblingen Manufacturing Operation and its Mechanical Technology Center are faced with this trend, along with others, such as tightened environmental protection guidelines and take-back regulations.

These trends call for new concepts—environmentally friendly materials and matching manufacturing methods. Assembly and disassembly times for computer products have to be as short as possible. Assembly analysis of some HP products clearly showed the necessity to reduce parts as well as to improve manufacturing and joining techniques.

At the Mechanical Technology Center, these observations provided the motivation to look for a new packaging and assembly concept for computer products, one that would leverage existing techniques and incorporate new technological ideas.

Our objectives were to reduce the number of components and the number of different part numbers, to achieve considerable savings in the area of logistics and administration, to save time in building a chassis, to automate the mounting of parts on the chassis, and to reduce overall chassis costs.

## Genesis of an Idea

After we had critically weighed all of the technologies known to us—namely, producing enclosures and chassis of sheet metal or plastics—the only reduction potential seemed to lie in reducing the number of parts and using snap fits to save on fasteners and assembly times. However, in contrast to our expectations, we could not do this to the extent we had in mind.

Using snap fits is a disadvantage, since disassembly is time-consuming and can lead to destruction of the components or the enclosure. In the future, enclosures not only need to be assembled quickly but also need to be disassembled within the same amount of time to make recycling easier and cheaper.

We could not get out of our minds the idea of fixing parts in such a way that they are enclosed and held by their own geometrical forms. The idea is similar to children's toys that

require them to put blocks, sticks, cards, or pebbles into matching hollows and at the same time keep track of positions and maintain a certain order at any time during the game. We applied this idea to our problem and thought about how our game collection would have to look in terms of composition and performance for us to be able to package and insert components for a workstation. It seemed most feasible to apply this idea at the assembly level, that is, to use the new method to fix conventional assemblies such as the disk, speaker, power supply, CPU board, and fan.

The only problem was what kind of material could we use to realize this idea. How could we achieve a form fit and not compromise on tolerances, feasibility, and price? Not to condemn the idea almost seemed impossible. It became obvious that we could no longer use conventional methods and standards to find the ideal material. We were forced to deal with a completely different field. The solution seemed to be to jettison everything we had learned before and direct our orientation towards something totally new.

The material we were looking for had to be pliable and bouncing—like foam, for example. Could foam be used for a form fit?

## Raw Material Selection

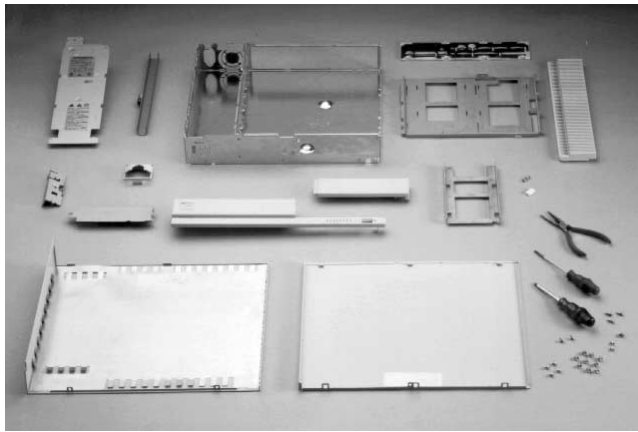
After the idea had been born to use foam, we started our search for a suitable material. The goal was to embed all components necessary for an electronic device in one chassis made of foam synthetic material. We had plenty of material to choose from, including polyurethane, polystyrene, and polypropylene. The material had to be:

- Nonconductive to hold and protect electronic components
- Able to hold tolerances in accordance with HP standards
- Able to fix components without fasteners.

With the help of our internal packaging engineers and an external supplier we soon found a suitable material: expanded polypropylene (EPP) with a density of 60 g/l.

In comparison to other foam synthetic material, EPP has the following advantages:

- Excellent mechanical long-term behavior
- Moisture resistance
- Resistance to chemicals
- Heat resistance



**Fig. 1.** Parts required for a workstation using the existing packaging concept.

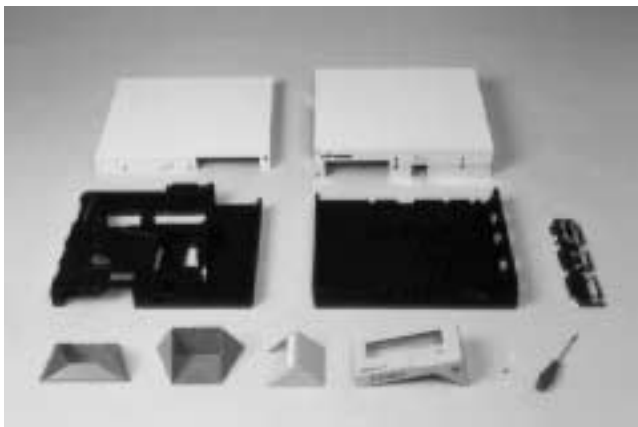
- 100% recyclability. Granules produced from recycled EPP can be used for manufacturing other parts such as packaging material and shock absorbers.

EPP foam parts can be produced in densities of 20 to 100 g/l. Lower density provides excellent shock absorption, while higher density offers tighter manufacturing tolerances. Thus design trade-offs are possible.

#### From the Idea to a Workstation

The next step was to apply this new concept to an already existing workstation. One workstation seemed suitable for the conversion. The existing concept (see Fig. 1), consisting of sheet-metal chassis (top and bottom), electrical components, sheet-metal enclosure, EMI liner, and plastic parts, was transformed into a foam chassis, electrical components, sheet-metal sleeves, integrated EMI liner, and modified plastic parts (see Fig. 2). In the new technology, all of the components are held by their own geometry in form-fitting spaces in the foam chassis. The connections between them are achieved through cabling held in foam channels (Fig. 3).

Time was saved by processing the foam chassis, the sheet metal, and the plastic parts in parallel. To obtain the foam parts quickly, we rejected the ordinary way of creating drawings with the help of a CAD system and instead created a 2D



**Fig. 2.** Parts required for the workstation of Fig. 1 using the HP-PAC concept.



**Fig. 3.** Channels in the foam carry cooling air (shown) and cabling.

cardboard layout showing the placement of the components. A packaging company placed their sample tooling shop at our disposal for a few days. The first prototype was built step by step. We milled, cut, and glued, applying a lot of imagination.

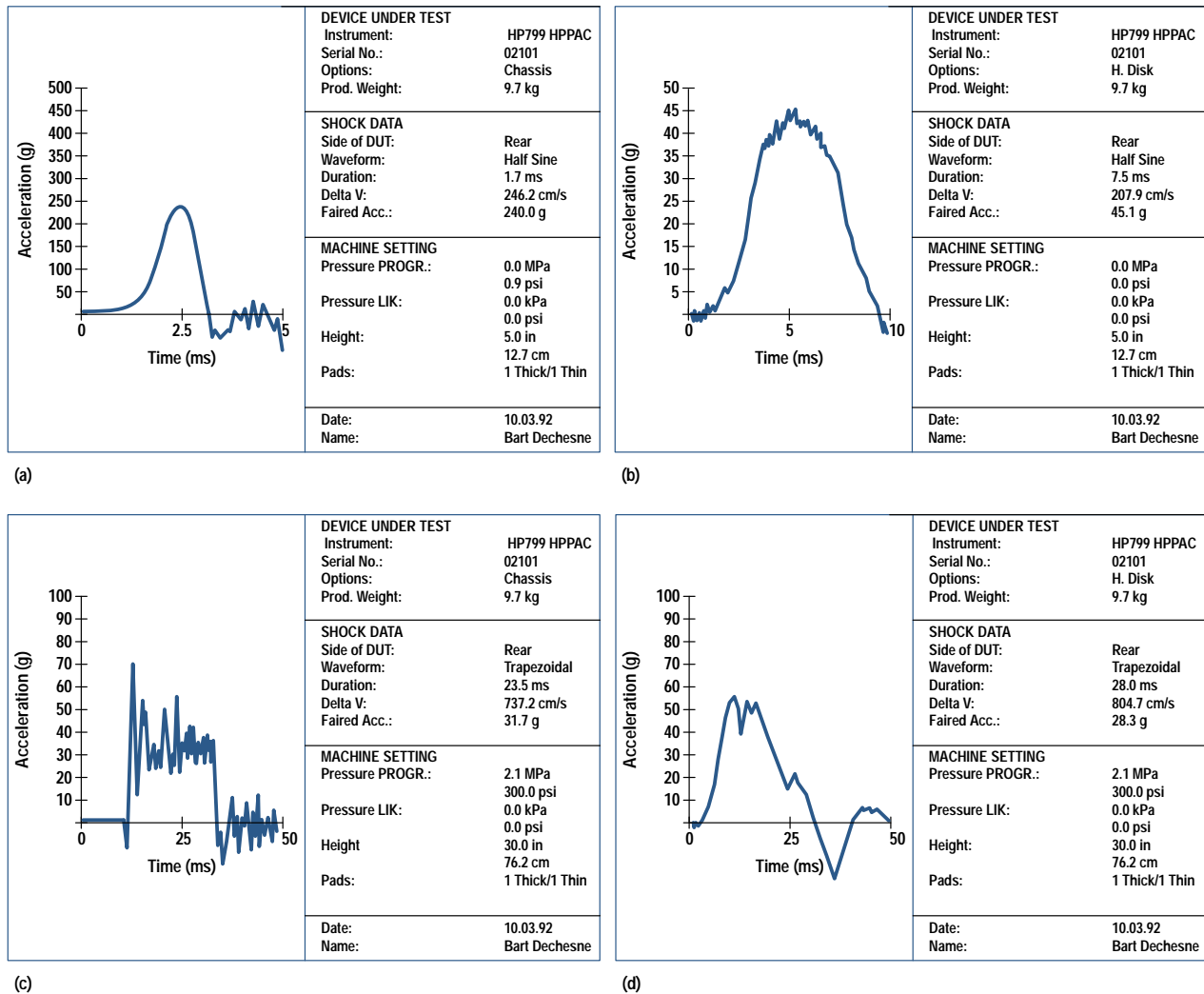
After two days the first prototype was nearly finished. Components were fixed in the necessary form fit and we were all aware that we had taken a step in the right direction. Back at HP we made a few minor changes to the EPP chassis and the remaining enclosure with the help of a knife and finished the prototype.

The major question now was, "Will it run?" We ran software on the workstation and started testing, surrounded by our production staff. The programs worked!

Next, temperature, humidity, and environmental tests were performed. Temperature problems were corrected by altering the air channels through cutting and gluing. HP class B2 environmental tests were passed (see Table I and Fig. 4).

**Table I**  
Thermal Test Results  
CPU Temperatures (°C)

Test Point	HP-PAC	Original
UB7	29.1	32.1
UD15	33.0	37.5
UB20	36.6	44.8
UH25	35.7	46.7
TO-220	49.3	59.8
UM10	39.9	44.5
UM25	58.7	67.5
UR30	56.7	73.7
MUSTANG	70.0	82.9
CPU Average	45.44	54.39



**Fig. 4.** Impacts transmitted to a hard disk drive by HP-PAC foam. In (a) and (c) the sensor is on the workstation. In (b) and (d) the sensor is on the hard disk drive held by HP-PAC.

Fig. 4 shows the impacts transmitted by HP-PAC to a hard disk for two different types of shocks (half sine, trapezoid). These impacts could be minimized by optimizing the design of the supports and form fits for the devices.

### Savings and Advantages

Comparisons were drawn between a traditional HP workstation and an HP workstation in which the system components such as the CPU board, disk drive, and flexible disk were mechanically integrated using the HP-PAC concept. The HP-PAC workstation showed:

- A 70% reduction in housing mechanical parts
- A 95% reduction in screw joints
- A 50% reduction in assembly time
- A 90% reduction in disassembly time
- A 30% reduction in transport packaging
- A 50% reduction in time and expenditure for the mechanical development of the housing.

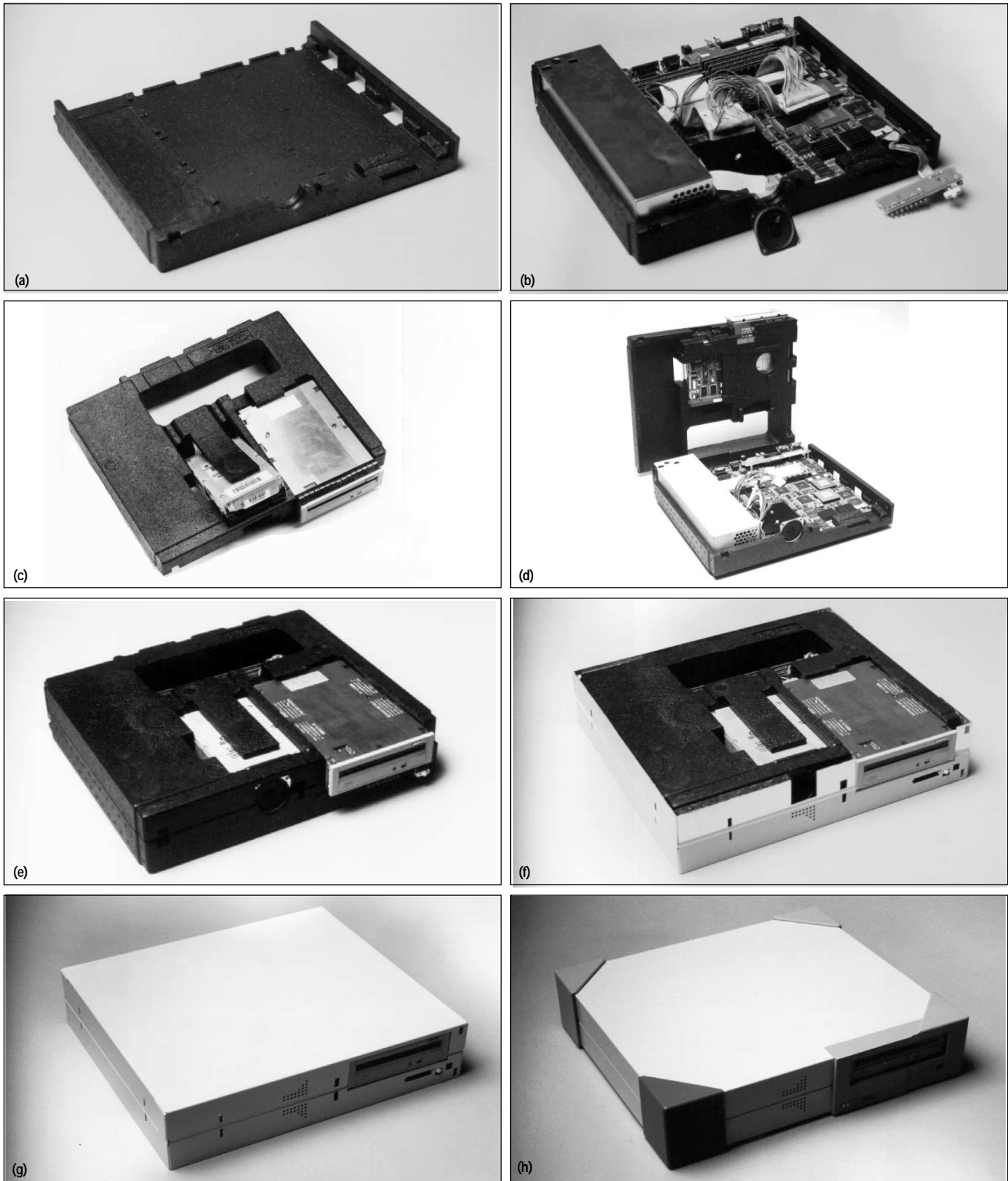
Compared to conventional chassis concepts, HP-PAC's advantages include:

- A reduction in the number of chassis parts.

- Separation between functionality and industrial design. The external enclosure is designed after definition of the mechanical interfaces between the enclosure and the chassis and between the enclosure and the components.
- One production step to produce molded parts.
- Simple, fast, and cost-effective assembly of the components (see Fig. 5). The assembly process is almost self-explanatory as a result of the indentations in the molded parts, and no additional joining elements and assembly tools are necessary. Assembly at the dealer's site is feasible.
- Reduced product mass because of the lighter chassis.
- Good protection against mechanical shock and vibration.
- On-the-spot cooling of components as a result of air channels in the foam.
- Cost savings during almost all working processes.
- Reduced transport packaging as a result of the good absorption of the chassis material. Also, less transport volume.

### Impact on the Development Process

With HP-PAC, a 100% recyclable and environmentally friendly material is used for the construction of the chassis.



**Fig. 5.** HP-PAC workstation assembly sequence. (a) Foam bottom chassis. (b) Loaded bottom chassis. (c) Partially loaded foam top chassis. (d) Open loaded chassis. (e) Assembled loaded chassis. (f) Lower enclosure added. (g) Upper enclosure added. (h) Enclosure completed.

Development of a chassis only means spatially arranging components within a molded part and adhering to certain construction guidelines (function- and production-specific). The external enclosure is developed separately after definition of the interfaces. There are hardly any tolerance problems as a result of material flexibility. Changes are simple to

perform by cutting, gluing, and additional grinding so an optimal solution can be reached quickly.

It is possible to perform all relevant environmental tests on the first prototype. Prototypes can also be used for the first

functional tests. It is relatively simple to make changes during the tests, since industrial design and functionality are clearly separated and changes on the molded part can be performed in the lab. Once the design is complete, manufacturing of the molding tool is fast and cost-effective, and tool changes are not required.

The result is a short, cost-effective chassis development phase.

### Material Development

HP-PAC places high demands, some of which are new, on the material used. Expanded polypropylene meets these demands in almost all ways.

So far EPP has been mainly used in reusable packaging and to an increasing extent in the automobile industry, where bumper inlays and side impact cushions for car doors are typical applications. Traditionally the automobile business has placed high demands on the quality of components. In terms of precision and long-term behavior these demands are identical to those of HP-PAC.

However, the situation is somewhat different for two new HP-PAC-specific material requirements: ESD (electrostatic discharge) suitability and flame retardant properties. We are working with raw material manufacturers in the U.S.A., Japan, and Germany to develop optimum raw material for HP-PAC in the medium and long term. For the short term, procedures had to be found and checked to meet these demands. In terms of ESD suitability this meant spraying or dipping parts in an antistatic solution. A suitable flame retardant was developed and patented together with a company specializing in flame retardants. Like plastic molding, treatment with flame retardant places an additional burden on the environment and impairs recyclability. However, a good product design renders flame retardants unnecessary. Three prototype HP-PAC products without flame retardant already have UL/CSA and TÜV approval.

We expect that the incentive for suppliers to develop suitable raw materials will steadily increase as the number of products using HP-PAC grows. Thus, in the future we hope to have custom-made materials available that will allow further improvements in quality at reduced cost.

### EPP and Its Properties

EPP raw material is available on a worldwide basis. Some EPP manufacturers and their trade names for EPP are JSP ARPRO, BASF NEOPOLEN P, and Kaneka EPERAN PP. EPP comes in the form of foam polypropylene beads. Its chemical classification is an organic polymer, one of the class of ethylene polypropylene copolymers.

EPP contains no softeners and is free of CFCs. The product does not emit any pollution. Compressed air, steam, and water are used during the molding process. According to one EPP manufacturer, no chemical reactions take place during this process.

The specifications quoted here are for the JSP raw material we used. EPP from other manufacturers should vary only a little or not at all from these specifications.

**Mechanical Properties.** The following list shows the relevant mechanical properties of parts made of expanded polypropylene foam with a density of 60 g/l.

Density: 60 g/l

Tensile strength: 785 kPa

Compressive strength at 25% deformation: 350 kPa

Residual deformation after 24 hours at 25% deformation: 9%

Deformation under static pressure load (20 kPa): 1.2%

After 2 days: 1.3%

After 14 days: 1.4%

**Thermal Properties.** After the molding process, the parts are tempered so that the dimensions become consistent. Any further temperature influences will not result in significant contraction, expansion, or changes of mechanical properties between  $-40^{\circ}\text{C}$  and  $110^{\circ}\text{C}$ . The coefficient of thermal expansion is  $4.2 \times 10^{-5}/^{\circ}\text{C}$  from  $-40^{\circ}\text{C}$  to  $20^{\circ}\text{C}$  and  $7.5 \times 10^{-5}/^{\circ}\text{C}$  from  $20^{\circ}\text{C}$  to  $80^{\circ}\text{C}$ . Thus, a 100-mm length of foam at  $20^{\circ}\text{C}$  will be 100.375 mm long at  $70^{\circ}\text{C}$ .

The material changes state above  $140^{\circ}\text{C}$ . Thermal dissolution occurs at  $200^{\circ}\text{C}$  and the ignition point is  $315^{\circ}\text{C}$ .

There was no permanent deformation in a temperature loop test consisting of:

4 hours at  $90^{\circ}\text{C}$

0.5 hour at  $23^{\circ}\text{C}$

1.5 hours at  $-40^{\circ}\text{C}$

0.5 hour at  $23^{\circ}\text{C}$

3 hours at  $70^{\circ}\text{C}$  and 95% humidity

0.5 hour at  $23^{\circ}\text{C}$

1.5 hours at  $-40^{\circ}\text{C}$

0.5 hour at  $23^{\circ}\text{C}$

**Electrical Properties.** EPP has good electrical insulation properties. This means that the foam parts can easily acquire an electrical charge. Consequently, methods are being developed to produce antistatic EPP. There is no noticeable interference between EPP material and high-frequency circuits with square-wave signals up to 100 MHz. Tests at very high frequencies ( $> 100$  MHz) have not yet been conducted.

We can infer from solid polypropylene some of the electrical properties of expanded polypropylene:

- Dissipation factor of injection-molded EPP at 1 MHz:  $\tan < 5 \times 10^{-4}$
- Breakdown voltage of injection molded EPP: 500 kV/cm
- Surface resistance at  $23^{\circ}\text{C}$  and 49% relative humidity, untreated:  $10^{11}$  to  $10^{12}$  ohms.

**Chemical Resistance.** EPP has good chemical resistance because of its nonpolar qualities. It is resistant to diluted salt, acid, and alkaline solutions. EPP can resist lye solutions, solvents at concentrations up to 60%, and alcohol. Aromatic and halogenated hydrocarbons found at high temperatures in grease, oil, and wax cause it to swell. When EPP is mixed with other substances, dangerous chemical reactions do not take place.

**Reaction to Light.** In general, EPP is sufficiently resistant to radiation at the wavelengths of visible light.



**Reaction to Humidity and Water.** Humidity has little or no effect on the mechanical properties of EPP. Water absorption is 0.1% to 0.3% by volume after one day and 0.6% after seven days. No changes are visible after 24 hours in water at 40°C.

### **Manufacturing Process**

The raw material beads are injected in a precombustion chamber at a pressure of approximately 5 bar which reduces the pellet volume. The beads are then injected into the mold at a pressure of approximately 4 bar until a particular filling ratio is reached. The pressure is then reduced to normal so that the beads can reexpand and fill the mold. Once the mold is filled, steam at 180°C is injected into the mold through nozzles, warming the surface of the beads and fusing them together. This defines the foam part, which is left to cool down and then removed from the mold. Subsequently, by means of specified temperature cycles, controlled maturing and dimensional changes are induced in the part, resulting in its final form. This form remains constant within a specified temperature range.

### **Recycling of EPP**

Polypropylene foam material can be recycled and used for manufacturing of other products. Manufacturers of polypropylene take back EPP waste free of charge.

EPP can be melted and fed back into source material polypropylene in thermoplastic form. Compression, melting, and granulation take place in gas extruders. The extruded recycled material can be used for polypropylene injection molded or extruded products. Recycling trials with a bumper system made out of short glass fiber (approximately 20% of

weight), EP rubber (approximately 20% of weight) and polypropylene produced a granule that can be used for complex injection molding.

### **Conclusions**

To protect the HP-PAC technology in an appropriate manner we have filed for a patent under European patent application number 0546211.

It goes without saying that we will continue to develop the technology further. Efforts in which we are currently engaging are material development, prototype manufacturing, quality assurance, and marketing of HP-PAC.

We have not yet set any specific limits on user distribution. Possible areas for user application range from the electronics and electromechanical industries to home electronic equipment and transportation. At the Mechanical Technology Center, we offer various services ranging from consulting to complete solutions, not only for HP-PAC but also for sheet-metal and plastic parts. We have experience in the computer, analytical, and instrument businesses and are in contact with others.

### **Acknowledgments**

We would like to thank the Novaplast Company and the following HP entities: Böblingen Computer Manufacturing Operation, Exeter Computer Manufacturing Operation, Böblingen Manufacturing Operation Mechanical Technology Center, Böblingen Manufacturing Operation environmental test laboratory, Workstation Systems Group quality department.

# High-Speed Digital Transmitter Characterization Using Eye Diagram Analysis

The eye diagram analyzer constructs both conventional eye diagrams and special eyeline diagrams to perform extinction ratio and mask tests on digital transmitters. It also makes a number of diagnostic measurements to determine if such factors as waveform distortion, intersymbol interference, or noise are limiting the bit error ratio of a transmission system.

by Christopher M. Miller

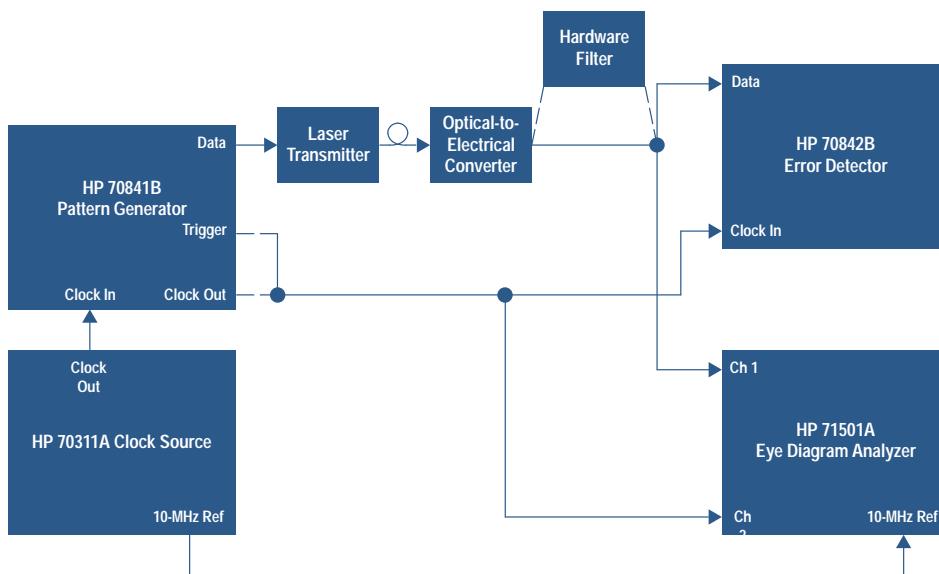
The goal of any transmission system is to deliver error-free information reliably and economically from one location to another. The probability that any bit in the data stream is received in error is measured by a bit error ratio (BER) test. This test is performed using an error performance analyzer, commonly referred to as a BER tester or BERT. Generally, a pseudorandom binary sequence (PRBS) from a pattern generator is used to modulate the transmission system's source, while an error detector compares the received signal with the original transmitted pattern. The BER is defined as the number of bits received in error divided by the number of bits transmitted, which equals the error count in a measurement period divided by the product of the bit rate and the measurement period.

In general, BERT measurements tend to be pass/fail in nature, and convey very little information about a failure. Moreover, some additional tests are usually required on components to ensure that they will meet the desired BER when they are installed into a system. For these reasons, it is desirable to perform a number of parametric measurements

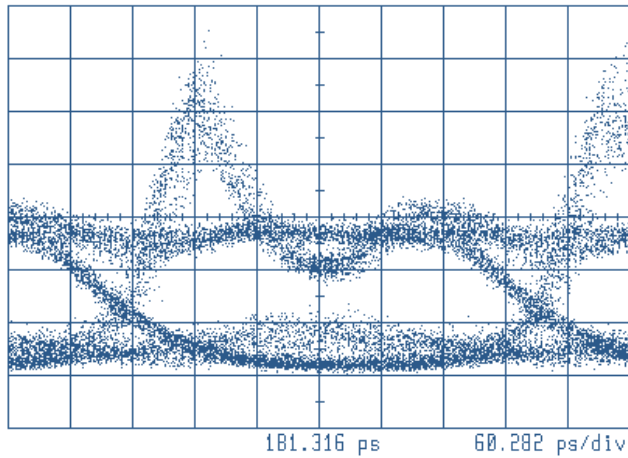
on the transmitted waveform in the time domain. Typically, an oscilloscope or an eye diagram analyzer is added to the BERT system as shown in the typical optical transmitter measurement setup in Fig. 1.

The pattern generator is still used to provide the stimulus. Different time-domain displays can be obtained depending on the choice of the trigger signal. When the pattern trigger or frame provides the trigger signal, a stable portion of the pattern appears on the display. When the clock frequency is used as a trigger signal, the data pattern waveform, superimposed on itself, produces a waveform display that is referred to as an eye diagram as shown in Fig. 2a. In general, the more open the eye is, the lower the likelihood that the receiver in a transmission system may mistake a logical 1 bit for a logical 0 bit or vice versa.

In an effort to standardize the high-speed telecommunication systems that are being developed and deployed, standards have been adopted for equipment manufacturers and service providers. Two such standards are the synchronous optical

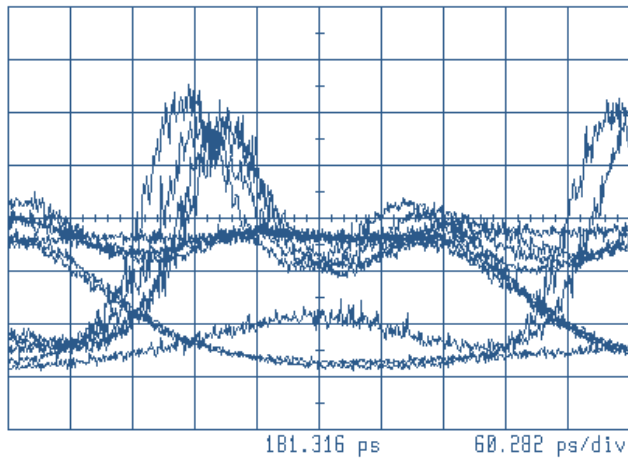


**Fig. 1.** Digital transmitter parametric test setup using an eye diagram analyzer.



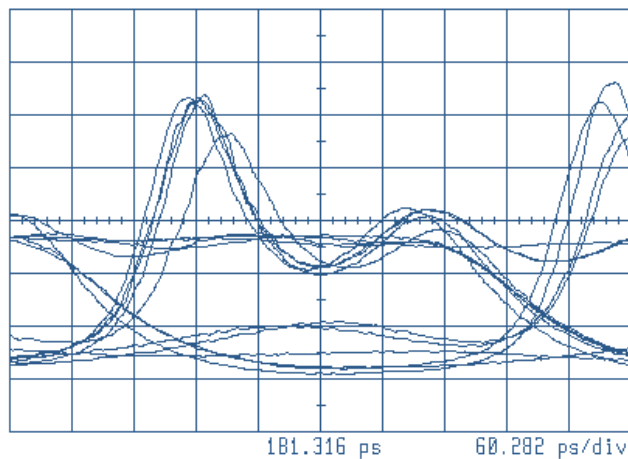
Tr1=Ch1  
80 mV/div  
240 mV ref

(a)



Tr1=Ch1  
80 mV/div  
240 mV ref

(b)



Tr1=Ch1  
80 mV/div  
240 mV ref

(c)

**Fig. 2.** (a) Conventional eye diagram. (b) Eyeline diagram. (c) Eyeline diagram with eye filtering.

network (SONET), a North American standard, and the synchronous digital hierarchy (SDH), an international standard. Both standards are for high-capacity fiber-optic transmission and have similar physical layer definitions. These standards define the features and functionality of a transport system based on principles of synchronous multiplexing. The more widely used transmission rates are 155.52 Mbits/s, 622.08 Mbits/s, and 2.48832 Gbits/s.

One of the goals of the standards is to provide “mid-span meet” so that equipment from multiple vendors can be used in the same telecommunications link. The standards specify extinction ratio and eye mask measurements on the transmitted eye diagram to help ensure that transmitters from various vendors are compatible.<sup>1,2</sup> The eye diagram shown in Fig. 2a is from a laser transmitter operating at 2.48832 Gbits/s. It shows the characteristic laser turn-on overshoot and ringing.

### Eye Diagram Characterization

An important specified test parameter for these transmission systems is the extinction ratio (ER) of the eye diagram. It is typically defined as:

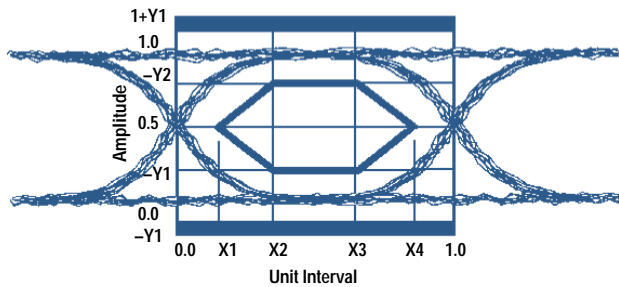
$$ER = 10 \log \frac{P_{avg(logic\ 1)}}{P_{avg(logic\ 0)}}$$

where  $P_{avg(logic\ 1)}$  is the mean or average optical power level of the logic 1 level and  $P_{avg(logic\ 0)}$  is the mean or average optical power level of the logic 0 level. For SONET/SDH transmission systems, the minimum specified extinction ratio is 10 dB. In some cases, the extinction ratio is expressed as the linear ratio of the two power levels. A good extinction ratio is desired in these systems to maintain an adequate received signal-to-noise ratio.

Although the definition of extinction ratio is relatively straightforward, the measurement methodology to determine the mean logic levels is not specified in the standards, such as SDH standard G.957. The histogram and statistical analysis capability of digitizing oscilloscopes can be used to determine the mean and standard deviation (sigma) of a waveform. However, there are no standard criteria for setting the windows and limits for the collection and evaluation of the data to determine the mean logic levels.

The Telecommunications Industry Association/Electronics Industry Association (TIA/EIA) has developed a recommended methodology for making eye diagram measurements called the Optical Fiber Standard Test Procedure #4 (OFSTP-4).<sup>3</sup> It recommends that voltage histograms be used to determine the most prevalent logic 1 and 0 levels of the eye pattern measured across an entire bit period. The OFSTP-4 also points out the importance of removing any residual dc offset from the extinction ratio measurement because this can dramatically affect the measurement accuracy.

Over the years, the designers of digital transmission systems have learned that the eye diagram should have a particular shape to achieve a good BER. Often these designers have constructed areas, or masks, inside and around the eye diagram. The eye diagram waveform should not enter into these masked areas. The polygons in the center of the eye diagram shown in Fig. 3 and the lines at the top and bottom correspond to the mask used to evaluate optical transmitters



	STM-1 (155.52 Mbits/s)	STM-4 (622.08 Mbits/s)
X1 / X4	0.15 / 0.85	0.25 / 0.75
X2 / X3	0.35 / 0.65	0.40 / 0.60
Y1 / Y2	0.20 / 0.80	0.20 / 0.80

	STM-16 (2.48832 Gbits/s)
X3 - X2	0.20
Y1 / Y2	0.25 / 0.75

**Fig. 3.** Laser transmitter eye diagram masks for SONET/SDH transmission systems.

intended for use in SONET/SDH systems. Depending on the transmission bit rate, the size and shape of the mask changes. The x and y coordinates are specified for each bit rate and their relative positions are based on the mean logical 1 level and the mean logical 0 level. The mask for the lower bit rates is a hexagon, whereas the mask for 2.48832-Gbit/s transmission is a rectangle. The receiver bandwidth for the measurement of the transmitted eye diagram is specified to be a fourth-order Bessel-Thomson response with a reference frequency at three fourths of the bit rate. This ensures a common reference bandwidth for transmitter evaluation. Hardware low-pass filters are commonly employed to achieve this response.

To date, existing instrumentation has been inadequate to design, build, and test optical transmitters sufficiently to meet the requirements of these transmission standards in certain key areas. Repeatable extinction ratio measurements are often difficult to obtain, particularly extinction ratio measurements of the low-power optical signals common in these systems. Easy-to-use mask compliance testing, with default standard masks that automatically scale to the data would be a convenience. But, most significantly, a tool to aid designers in diagnosing transmitted BER problems would be a major contribution.

### Eye Diagram Analyzer

The HP 71501A eye diagram analyzer combines the HP 70820A transition analyzer module,<sup>4</sup> the HP 70004A color display and mainframe, and the HP 70784A eye diagram personality. The personality is stored on a 128K-byte ROM card and can be downloaded into the instrument. The instrument can be used with a number of optical converters. When the eye diagram analyzer is combined with a pattern generator such as a member of the HP 71600 family of pattern generators, a number of formerly difficult transmission measurements can be made easily. This instrument configuration is shown in Fig. 4.



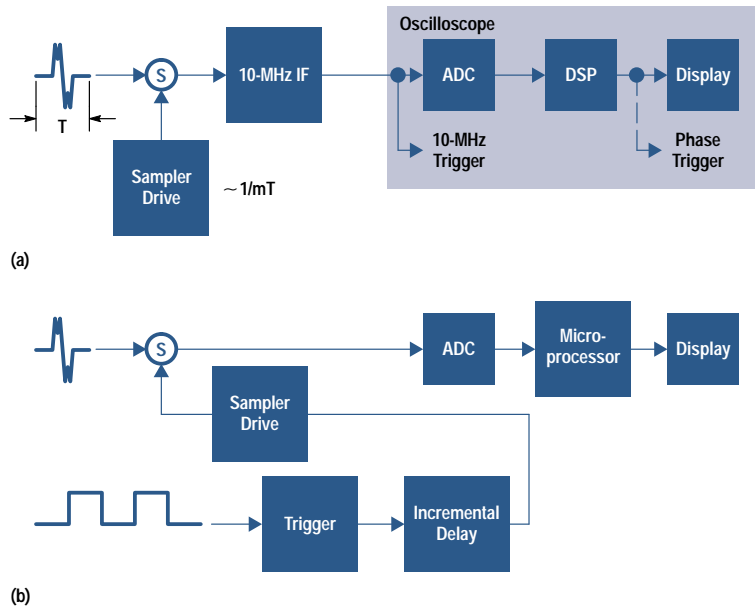
**Fig. 4.** Photograph of the HP 71501A eye diagram analyzer with an HP 71603B pattern generator system.

The operation of the HP 71501A differs from that of a conventional digital repetitive sampling oscilloscope. As shown in Fig. 5, both instruments have microwave samplers to sample the incoming waveform before it is digitized by an analog-to-digital converter (ADC). A digitizing oscilloscope has a trigger input that is used to start a sample. In addition, an incremental delay is added to the trigger signal so the samples step through the input waveform. After many cycles of the incoming signal, a complete trace of the input waveform is constructed.

The sample rate of the eye diagram analyzer is not determined by an external trigger, but it is set according to the frequency content of the incoming signal itself. As the incoming signal is digitized, it is analyzed to determine an appropriate sample frequency to down-convert it optimally into the intermediate frequency (IF) section.

As shown in the block diagram, Fig. 6, the HP 71501A has two identical signal processing channels which can sample and digitize signals from dc up to 40 GHz. Input signals to each channel are sampled by a microwave sampler at a rate ( $f_s$ ) between 10 MHz and 20 MHz. The sample rate is dependent upon the signal frequency and the type of measurement being made. The outputs of the samplers are fed into the dc-to-10-MHz IF sections. The IF sections contain switchable low-pass filters and step-gain amplifiers. The dc components of the measured signal are tapped off ahead of the microwave sampler and summed into the IF signal separately. The outputs of the IF sections are sampled at the same rate as the input signal and then converted to a digital signal by the ADCs.

Once the signals are digitized, they are fed into the buffer memories. These buffers hold the samples until the trigger point is determined. The buffer memories make it possible to view signals before the trigger event without using delay



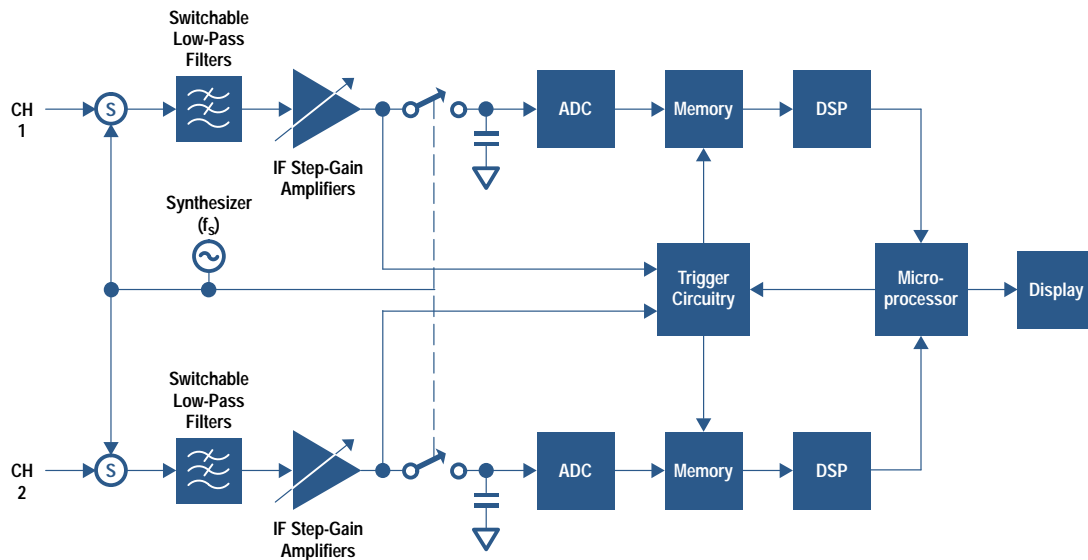
**Fig. 5.** Simplified architectural comparison of the eye diagram analyzer (a) and a conventional sampling oscilloscope (b).

lines. By triggering on the IF signal, the HP 71501A is able to trigger internally on signals with fundamental frequencies as high as 40 GHz. Once the trigger point has been determined and all the necessary data has been acquired, the appropriate data is sent to the digital signal processing (DSP) chips.

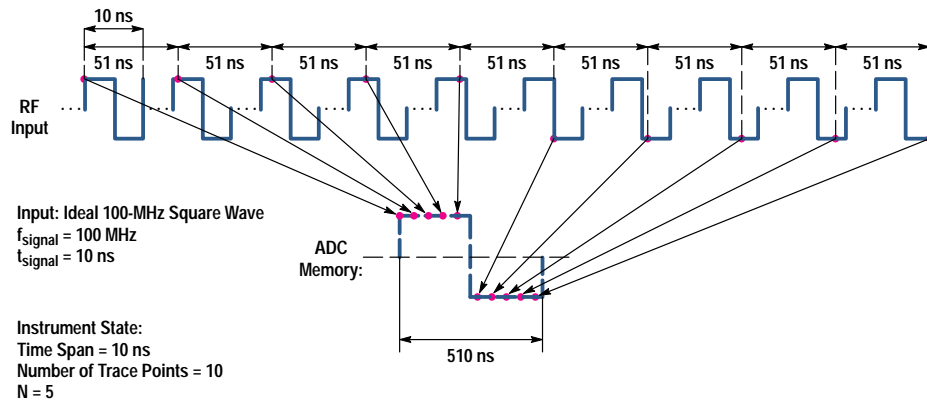
An FFT is performed on the time data that is sent to the DSP chips. With the time data now converted into the frequency domain, IF corrections are applied to the data. The IF corrections compensate for nonidealities in the analog signal processing path. In addition, in certain modes of operation, the nominal measurement 3-dB bandwidth of 22 GHz can be extended to 40 GHz by applying RF corrections. The RF corrections compensate for microwave sampler conversion efficiency roll-off versus frequency. Similarly, in these modes of operation, user-entered corrections or filtering can be applied at this point as frequency-domain multiplication. As we will see later, this is a very useful capability. Finally, the inverse FFT is performed.

### Generating Eye and Eyeline Diagrams

Fig. 7 shows how the HP 71501A acquires data in the time domain using a technique called harmonic repetitive sampling. The sample rate is set so that successive sample points step through the measured waveform with a specified time step. The sampling period,  $T_s$ , is computed using the fundamental signal period, the time span, and the number of trace points.  $T_s$  is set such that an integer number ( $N$ ) of signal periods plus a small time increment ( $\Delta T$ ) occur between successive sample points. For example, suppose that the input signal is a 100-MHz square wave as shown in Fig. 7. The minimum sampling period of the HP 71501A is 50 ns. Therefore, five cycles of the input waveform are skipped between samples. However, if the sample period were set to exactly 50 ns, sampling would occur at the same point on every fifth cycle, and no new information would be gained. For a time span of 10 ns and the number of trace points



**Fig. 6.** Simplified block diagram of the eye diagram analyzer.



$$T_s = \frac{1}{f_s} = (N \times T_{\text{signal}}) + \Delta T$$

$$\Delta T = \frac{\text{Time Span}}{\text{Number of Trace Points}} = \frac{10 \text{ ns}}{10} = 1 \text{ ns}$$

Therefore:  $T_s = 51 \text{ ns}$

**Fig. 7.** Harmonic repetitive sampling of a waveform.

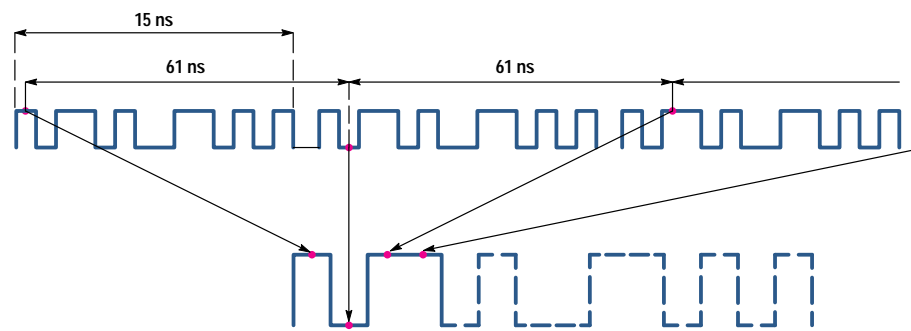
equal to 10 points, a time resolution or  $\Delta T$  of 1 ns is required. For this time resolution, the actual sample period is  $50 \text{ ns} + 1 \text{ ns} = 51 \text{ ns}$ . Thus, at every fifth cycle of the input waveform the sampling point moves forward 1 ns, and after fifty cycles of the input waveform, one complete trace of the input signal would be displayed.

With the signal frequency set equal to the clock frequency, the HP 71501A uses the process of harmonic repetitive sampling to generate eye diagrams similar in appearance to those of sampling oscilloscopes. The eye diagram of a pseudorandom bit sequence (PRBS) obtained in this manner is shown in Fig. 2a. Like the display of a conventional digital sampling oscilloscope, the eye diagram is constructed from samples of a number of bits in the sequence, which are displayed as a family of dots when persistence mode is activated. The only relationship between the samples or dots is their position relative to a trigger point, which is usually the rising or falling edge of a clock signal.

The HP 71501A uses a modified version of this same technique of repetitive sampling to construct the *eyeline diagrams* of PRBS signals or any bit sequence whose pattern repeats. Fig. 2b shows an eyeline diagram of the same laser

transmitter as Fig. 2a. In this mode, successive samples come from the same or adjacent bits in the pattern. The samples or dots in these eyeline diagrams can be connected, and a whole trace or portion of the bit sequence is displayed at one time. When the display is in persistence mode, a number of traces of different portions of the bit sequence are superimposed, forming the eyeline diagram. The continuous traces that make up the eyeline diagram convey much more information than the sampled smear of dots observed in the conventional eye diagram. The pattern dependent traces that make up the eyeline diagram, as well as the bit transitions, are readily seen. With the eyeline display it is possible to observe whether noise, intersymbol interference, or mismatch ripple is causing eye closure.

When the HP 71501A is placed in the eyeline mode,  $f_{\text{signal}}$  is set to the clock rate divided by the pattern length, and  $T_s$  is set such that successive sample points step through the waveform pattern.  $\Delta T$  is computed exactly as before. For the example shown in Fig. 8, the clock frequency is 1 GHz, the pattern length is 15 bits, and  $f_{\text{signal}}$  is 66.67 MHz. For a time span of 15 ns and the number of trace points equal to 15, a  $\Delta T$  of 1 ns is required.  $T_s$  can be computed to be 61 ns.



Input: 1-Gbit/s, 15-Bit Pattern  
 $f_{\text{clock}} = 1 \text{ GHz}$   
 Pattern Length = 15 bits  
 $f_{\text{signal}} = 66.67 \text{ MHz}$   
 $T_{\text{signal}} = 15 \text{ ns}$

Instrument State:  
 Time Span = 15 ns  
 Trace Length = 15  
 N = 4

$$T_s = \frac{1}{f_s} = (N \times T_{\text{signal}}) + \Delta T$$

$$\Delta T = \frac{\text{Time Span}}{\text{Trace Length}} = \frac{15 \text{ ns}}{15} = 1 \text{ ns}$$

Therefore:  $T_s = 61 \text{ ns}$

**Fig. 8.** Harmonic repetitive sampling of a PRBS waveform to generate an eyeline diagram.

Thus, every fourth cycle of the pattern, the sampling point moves forward 1 ns, and after sixty cycles of the pattern waveform, one complete trace of the pattern is displayed.

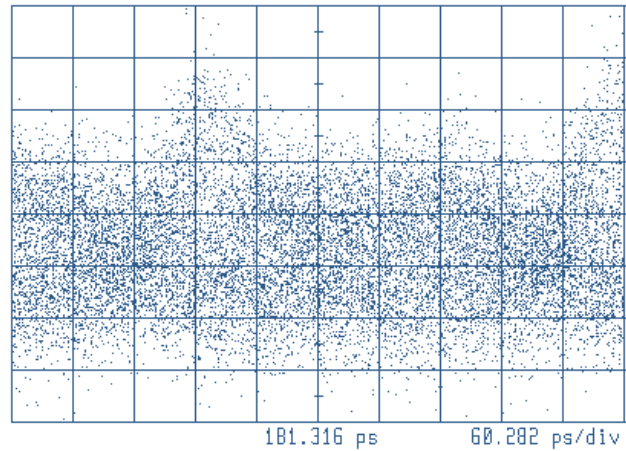
### Eye Filtering

The eyeline mode traces can be filtered to remove the noise and nonpattern dependent effects. As shown in Fig. 2c, this allows an enormous improvement in the ability to view the individual traces. The reduction in trace noise and the improvement in signal-to-noise ratio (SNR) available in eyeline mode are enabled by turning on the eye filter. Trace averaging, a common technique to improve the SNR of time-domain displays of a stable pattern, cannot be applied to conventional eye diagrams because the averaged waveform would converge to the dc or average level.

For signals with pattern repetition frequencies greater than 10 MHz, turning on the eye filter switches in a 100-kHz IF hardware filter. This effectively reduces the IF bandwidth from 10 MHz to 100 kHz, which provides a fixed 20-dB signal-to-noise improvement. However, for most PRBS signals, the pattern repetition frequency is much less than 10 MHz. In this case, additional digital signal processing is performed to improve the SNR. In essence, a number of samples are taken at each time record point in the measured waveform. This is accomplished by sampling at a rate equal to or harmonically related to the pattern repetition frequency. For each time point, the samples are passed through a narrow-band filter implemented with an FFT. To sample the next trace point, the internal synthesizer controlling the sampling rate is phase-shifted a precise amount. This process is repeated for each trace point. The actual amount of noise reduction or processing gain is a function of the number of samples taken at each trace point. From an operational standpoint, the amount of processing gain is represented by an equivalent noise-filter bandwidth. The bandwidth of the filter indicates the relative noise reduction normalized to the 10-MHz IF bandwidth. As the effective bandwidth is reduced, the number of samples increases, and the trace update rate gets slower. The following table shows the SNR ratio improvement for a given number of samples.

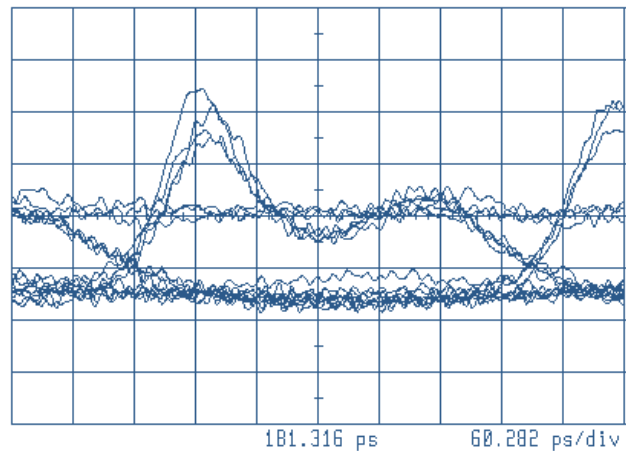
Equivalent Filter Bandwidth	SNR Improvement	Number of Samples
10 MHz	0 dB	1
2 MHz	7 dB	16
1 MHz	10 dB	32
500 kHz	13 dB	64
250 kHz	16 dB	128
125 kHz	19 dB	256
62.5 kHz	22 dB	512

The eyeline mode with eye filtering offers a significant improvement over conventional eye diagram measurements in the ability to observe low optical signal levels. Shown in Fig. 9a is the same laser transmitter whose output now has been optically attenuated. This conventional eye diagram display is of poor quality because of inherent sensitivity limitations and the inability to perform trace averaging on a PRBS waveform. Enabling the eyeline mode and activating the eye filter makes the eye diagram once again clearly visible as shown in Fig. 9b. This function makes it easy to observe signals that are only several millivolts in amplitude, which is impossible



Tr1=Ch1  
2 mV/div  
4 mV ref

(a)



Tr1=Ch1  
2 mV/div  
4 mV ref

(b)

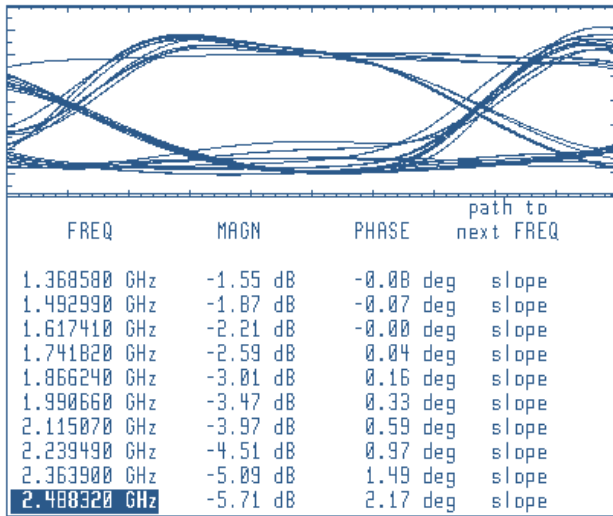
**Fig. 9.** (a) Conventional eye diagram of a low-level signal. (b) Eyeline diagram of a low-level signal with eye filtering applied.

to do with conventional high-speed digital sampling oscilloscopes.

An application of this measurement capability is the observation of the eye diagram of a transmitted optical signal after the signal has passed through a long length of optical fiber. Because of the attenuation of the fiber, eye diagrams are often difficult to observe in this manner. With the eyeline mode, it is possible to determine if the chirp of the laser transmitter, along with the dispersion in the fiber, is causing any waveform distortion.

### Software Corrections and Filtering

RF corrections can be applied to extend the measurement bandwidth of the HP 71501A to 40 GHz while the instrument is in the eyeline display mode, in which a unique mapping exists between the IF frequency and the input RF frequency. When the IF signal is digitized, an FFT is used to convert it to the frequency domain. The IF frequencies are then mapped into the corresponding RF frequencies, and the



**Fig. 10.** Eye diagram with software filtering applied and user correction table corresponding to a fourth-order Bessel-Thomson response.

appropriate correction is applied at each frequency. Finally, an inverse FFT is applied to the result to transform it back to the time domain. These same processing routines are available for user-defined corrections or filters. For example, this capability can be used to calibrate the eyeline lightwave measurements by correcting for any optical converter roll-off. In addition, it can be used to evaluate eyeline diagrams under different filter response conditions.

As previously mentioned, the transmission standards require that the eye mask measurements be made in a receiver bandwidth that corresponds to a fourth-order Bessel-Thomson response with a reference frequency at three fourths of the bit rate. Hardware filters or special lightwave converters are often employed to achieve this. By using the user corrections available in the HP 71501A, the ideal transfer function can be implemented with a software filter. The filtered display shown in Fig. 10 was made with a software filter applied that corresponded to the desired ideal Bessel-Thomson response.

Up to 128 magnitude and phase points can be loaded as user corrections. Shown in the lower half of Fig. 10 is a portion of the user correction table that corresponds to the fourth-order Bessel-Thomson response. This table of magnitude and phase corrections was generated directly from trace math functions that are available in the HP 71501A. The linear portion of the phase has been corrected to remove the delay. This allows the trace to remain in the same place when the corrections are applied. Software filters for the major SONET/SDH transmission rates are stored on the ROM memory card. This capability could also be used to develop an equalizer or tailor a specific response to improve the received eye diagram. Thus, the transmitted "equalized" eye diagram could be simulated and observed before the final hardware equalization filter is constructed.

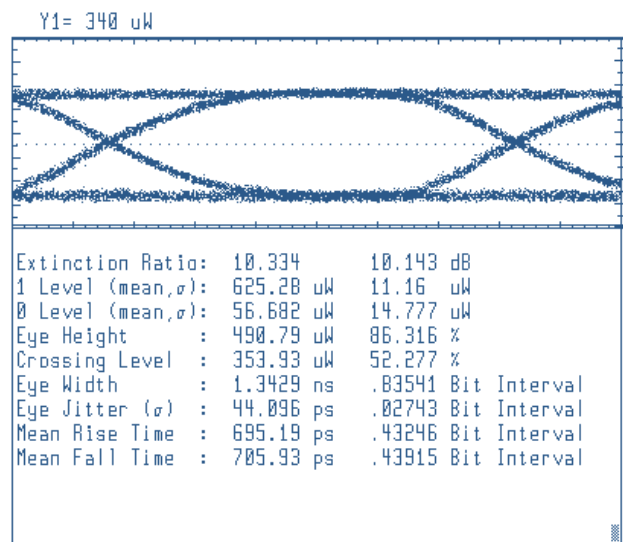
## User Interface

An application-specific user interface was designed for the eye diagram analyzer. The goals of the user interface were to offer turnkey measurements that meet the requirements of the standards, to be easy to use, and to make the interface to the pattern generator as transparent as possible. Many of the specific measurements will be described in the next sections. One ease-of-use feature is the ability to set the time base in bit unit intervals, instead of having to recall the clock period to display the eye diagram. Time delay can also be set in bit intervals, which makes it convenient to scroll through the bits that make up the pattern. A menu key, READ PAT GEN, is used to interrogate the pattern generator. It returns the clock frequency, the pattern length, and the data, clock, and trigger levels, which are used to set up the display.

The eye diagram application program is written in HP Instrument BASIC and is downloaded into the host instrument, the HP 70820A microwave transition analyzer module. The microwave transition analyzer is an extremely flexible, general-purpose instrument that offers both time-domain and frequency-domain signal analysis capability.<sup>4</sup> This versatile measurement capability is available simultaneously with the eye diagram measurement capability, and is simply accessed through its own menu.

## Eye Diagram Measurements

A number of parametric measurements are often performed on eye diagrams to determine their quality. Some of the more prevalent measurements include eye opening height, eye opening width, amplitude of the crossing level, jitter at the transition point, and the rise and fall times of the bit transitions. Fig. 11 shows a number of these measurements made with the HP 71501A eye diagram analyzer on a laser transmitter operating at 622.08 Mbits/s. The extinction ratio was determined by taking a vertical histogram of the eye diagram windowed over one bit interval. An algorithm that recursively adjusts the limits for subsequent evaluations of the histogram data converges on the most prevalent logical



**Fig. 11.** Table of eye diagram parametric measurements.



1 and 0 levels. The peaks of the histogram are used to set initial limits for the computation of the 1 and 0 levels. The initial mean and standard deviation (sigma) of the 1 level are based on histogram data above the relative 50% point of the peaks. The limits for the next evaluation of the histogram data are set to the initial mean level plus or minus one sigma. The new mean and sigma for the 1 level are then determined. This process iterates several times until the sigma becomes small and the mean converges on the most prevalent 1 level. The determination of the most prevalent 0 level is based on the same algorithm, except that the initial mean and sigma of the 0 level are based on histogram data below the relative 50% point of the peaks. This algorithm for determining extinction ratio has been demonstrated to be more repeatable than merely taking the peaks of the histogram distributions. The other eye parameter measurements are also based on histograms.

At times, it is convenient to display the trace in optical power units. This can be easily done with the HP 71501A. With the responsivity of the optical converter entered as in Fig. 11, we can easily see using the marker functions that the laser is putting out about 340  $\mu$ W of average optical power. With the responsivity entered, the eye measurements are displayed in the appropriate optical power units.

To get a good extinction ratio measurement result, it is especially important to get an accurate determination of the average power corresponding to the low logic level. It can be readily shown that a measurement error of a given magnitude has a much greater impact on the value of the low level than the high level. Because of sensitivity limitations, accurate measurement of the low level may be difficult. However, this is an area where the eye filtering capability of the HP 71501A can make a contribution. When making measurements on high-speed laser transmitters with unamplified photodiode converters, the resulting detected voltages are only millivolts in amplitude, making conventional extinction ratio measurements next to impossible. Yet, with the eye filter enabled on the HP 71501A, the extinction ratio can be readily determined.

### Mask Measurements

The HP 71501A has a general-purpose mask testing capability with internally stored default masks for the major SONET/SDH transmission rates to test compliance with these standards. These default masks can be autoscaled to the data according to the specifications in the SONET/SDH standard. Mask margin testing is also provided. Fig. 12 shows a SONET/SDH eye diagram mask test performed by the HP 71501A. Once again, the transmission rate was 622.08 Mb/s. The measurement bandwidth was fixed by a hardware filter with a fourth-order Bessel-Thomson response and a reference frequency of 466.56 MHz. For this transmission rate, the default mask consists of a hexagon, M1, in the center of the eye, and upper and lower limit lines, L2 and L3. The default mask has been autoscaled to the data. The error count for each of the mask regions is displayed on the lower portion of the screen, along with a number corresponding to the total traces evaluated. The transmitter measurement shown passes the mask test without any violations. In some instances, it is useful to determine by how much margin a transmitter passes the mask test. It is often desirable in production testing to

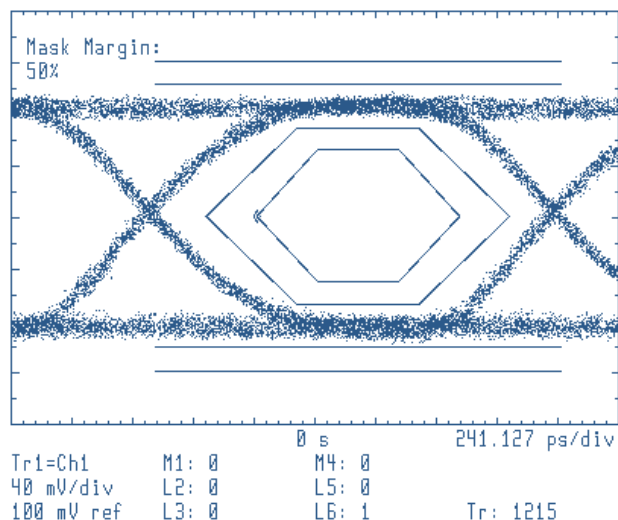


Fig. 12. Eye mask measurement.

allow some additional guardband to ensure that the transmitter can pass the standard mask. M4, L5, and L6 denote the margin mask regions, and one can test for errors simultaneously in those regions as well as the standard mask regions. As shown, there was a margin mask violation of the lower limit after 1215 traces had been evaluated. Custom masks can also be easily constructed for other transmission systems or to assist in the design and troubleshooting of laser transmitters.

### Error Trace Capture

The jitter in the data transitions is a very important transmission parameter that has a direct impact on the BER. In many systems only the random jitter is specified. But in optical systems, a deterministic component of jitter that is dependent on the optical pattern is often present. Using the eye-line mode with eye filtering one can observe the peak-to-peak variations in the crossing points as shown in Fig. 13.

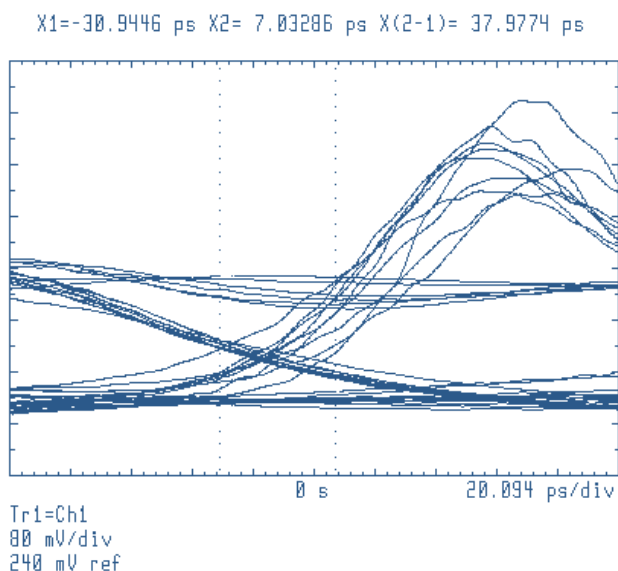
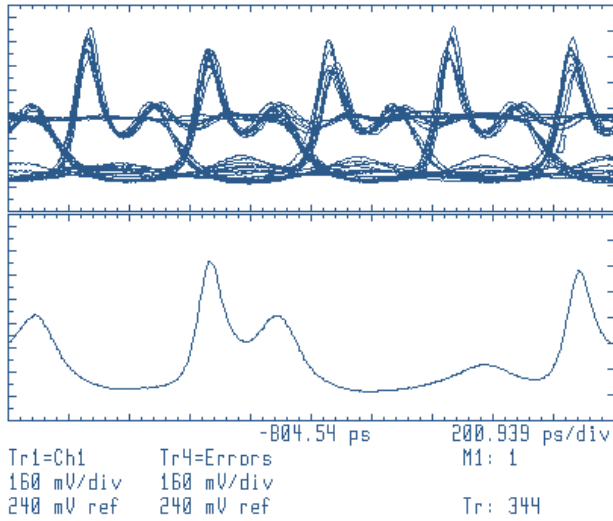


Fig. 13. Jitter in data transitions.



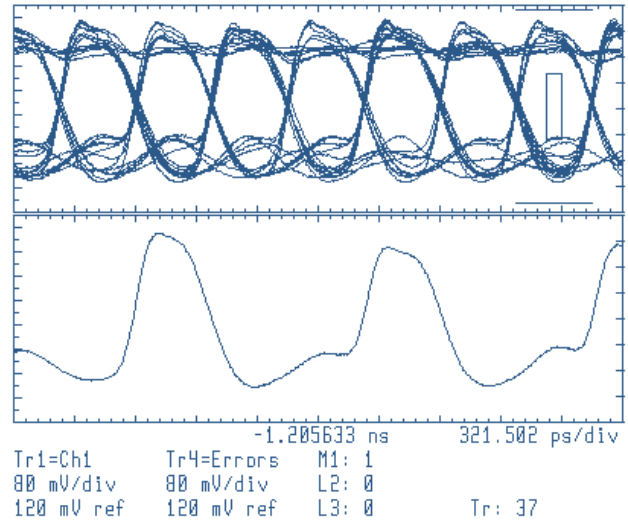
**Fig. 14.** Error capture of a jittered transition.

To determine if a particular pattern is responsible for the jittered transition, a custom mask can be employed. By displaying several preceding bit intervals and enabling the error trace capture capability of the HP 71501A, one can observe in the lower trace of Fig. 14 that the delayed transition seems to be caused by intersymbol interference from a preceding 100 pattern. The error trace capture is a unique feature of the eye diagram analyzer, made possible by the eye-line display mode and a triggering architecture that allows data before the trigger point to be viewed.

The error trace capture can be applied to a number of measurements. It can be used to observe the pattern leading up to a standard default mask violation. Displayed in Fig. 15 are a transmitter waveform and the associated mask for the 2.48832-Gbit/s transmission rate displayed in the appropriate bandwidth. The error trace here also seems to be a result of intersymbol interference.

### Summary

High-speed telecommunication standards require that eye diagram measurements be made on digital transmitters. The HP 71501A eye diagram analyzer is designed to meet these measurement needs by performing industry-standard mask and extinction ratio measurements. It can construct both conventional eye diagrams and unique eyeline diagrams, which can be used for bit error analysis.



**Fig. 15.** Error capture of a mask violation.

### Acknowledgments

The eye diagram analyzer involved the contributions of a number of people. Chris "CJ" Johansson wrote the IBASIC application program. Steve Peterson was primarily responsible for revisions and upgrades to the host instrument firmware. Steve, along with John Wendler and David Sharrit, provided technical input. John Wilson provided market research and many of the ideas that make up the operation of the application. Finally, Mike Dethlefsen and Mark Slovick modified the existing base instrument's alignment and calibration routines to achieve the eye diagram analyzer's current state-of-the-art time-domain measurement performance.

### References

1. *American National Standard for Telecommunication - Digital Hierarchy Optical Interface Specifications, Single Mode*, ANSI T1-106-1988.
2. *Optical Interfaces for Equipments and Systems Relating to the Synchronous Digital Hierarchy*, CCITT Recommendation G.957, 1990.
3. *OFSTP-4 Optical Eye Pattern Measurement Procedure*, TIA/EIA-526-4, 1993.
4. D.J. Ballo and J.A. Wendler, "The Microwave Transition Analyzer: A New Instrument Architecture for Component and Signal Analysis," *Hewlett-Packard Journal*, Vol. 43, no. 5, October 1992, pp. 48-62.

# Thermal Management in Supercritical Fluid Chromatography

In supercritical fluid chromatography, very high degrees of accuracy are required for temperature control. On the fluid supply end of the system, cooling is critical. On the separation end, heating is important. This paper discusses temperature control in the HP G1205A supercritical fluid chromatograph.

by **Connie Nathan and Barbara A. Hackbarth**

Supercritical fluid chromatography (SFC) is a technique that has gained acceptance in the analytical chemistry marketplace as a complement to gas chromatography (GC) and liquid chromatography (LC). In the development of the HP G1205A supercritical fluid chromatograph (Fig. 1), leveraging of major components from HP GC and LC products was a primary goal. As a result of this goal, thermal management in the system was a challenge because the components were not intended to operate in the temperature range required for SFC. For example, the LC pumping module was designed to pump fluids at room temperature, while for SFC fluids optimal delivery is at 5°C or lower.

Modifications were made to the components to integrate them into one product. Some components were optimized for SFC. This helped to improve the chromatographic technique by incorporating new ways to manage and control temperature. This paper examines the design modifications made to components to meet the thermal requirements for SFC while leveraging current HP analytical product components.

## SFC System

An SFC unit consists of four major systems: fluid delivery (pumps), separation, detection, and data collection. A brief description of chromatography accompanies this article to provide the reader with an overview of the technology (see page 39).

The HP G1205A SFC project goal of using, wherever possible, components from already proven HP GC and LC instrumentation resulted in reuse of the HP 5890 GC oven, the HP 1050 LC pumping module, a variety of both GC and LC detectors, and the HP ChemStation instrument control software. The major components of the HP G1205A SFC system will now be described in further detail.

**Pumping Module.** The HP G1205A SFC system is available as a single-pump system using CO<sub>2</sub> (or other fluids) for its supercritical mobile fluid or as a dual-pump system that allows modifiers to be added to the CO<sub>2</sub>. Both pumps consist of reciprocating dual pistons in series, allowing for continuous, reliable, and unattended pumping. This eliminates the inconvenience of refilling syringe pumps and allows for control of the flow and changing of the composition. The pumps have feedback control algorithms that dynamically compensate for



**Fig. 1.** HP G1205A supercritical fluid chromatography system.

optimum fluid compressibility and minimize the pressure ripple of the reciprocating pistons.

The HP G1205A modifier pump design parallels the HP 1050 LC pump closely since both are intended to pump incompressible liquids. This is not the case with the supercritical fluid pump, which required more extensive modifications. To pump compressible fluids like CO<sub>2</sub> efficiently, the

## What is SFC?

Chromatography is a process in which a chemical mixture, carried by a mobile phase, is separated into components as a result of differential distribution of the solutes as they flow over a stationary phase. The distribution is the result of differing physical and/or chemical interactions of the components with the stationary phase. On a very basic level, chromatography instrumentation consists of (1) a delivery system to transport the sample within a mobile phase, (2) a stationary phase (the column) where the separation process occurs, (3) a detection system that identifies or distinguishes between the eluted compounds, and (4) a data collection device to record the results (see Fig. 1).

The choice of which chromatographic method to use depends on the compounds being analyzed. In gas chromatography (GC), the mobile phase that carries the sample injected into the system is a gas. GC is generally a method for volatile and low molecular weight compounds. High-performance liquid chromatography (HPLC) is primarily used for analysis of nonvolatile and higher molecular weight compounds. A combination of desirable characteristics from both of these methods can be obtained by using a supercritical fluid as the mobile phase. A supercritical fluid is a substance above its critical point on the temperature/pressure phase diagram (see Fig. 2). Above the critical point, the fluid is neither a gas nor a liquid, but possesses properties of both.

The advantage of SFC is that the high density of a supercritical fluid gives it the solvent properties of a liquid, while it still exhibits the faster physical flow properties of a gas. In chromatographic terms, supercritical fluids allow the high efficiency and detection options associated with gas chromatography to be combined with the high selectivity and the wider sample polarity range of high-performance liquid chromatography. Applications that are unique to SFC include analysis of compounds that are either too polar, too high in molecular weight, or too thermally labile for GC methods and are undetectable with HPLC detectors. Another benefit of SFC over LC is the reduction of toxic solvent use and the expense associated with solvent disposal. This aspect has become increasingly important as environmental awareness becomes a larger issue.

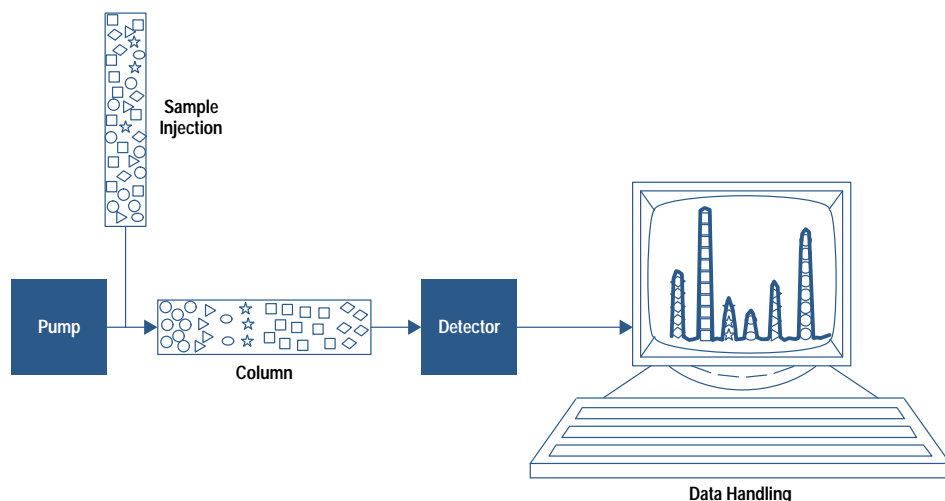


Fig. 1. Basic components of a chromatographic system.

supercritical fluid pump head must be cooled to lower temperatures. Peltier cooling was selected for its clean, self-contained, quiet, and reliable operation. Additionally, the Peltier elements provide precise temperature control, minimizing pump flow noise and allowing more accurate compressibility compensation.

**Separation Phase.** The column is where the actual separation of components occurs. The unique interactions of the compounds with a given stationary phase at certain conditions

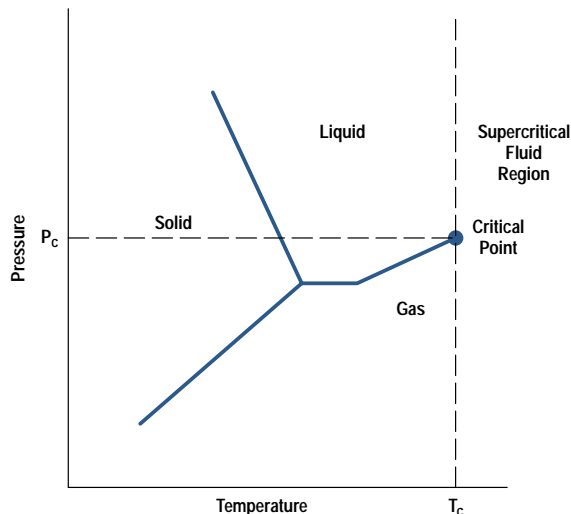


Fig. 2. Phase diagram with critical point and supercritical region.

Hewlett-Packard developed and manufactured its first SFC instrument in 1982. For the past decade, SFC has primarily been used in R&D laboratories. The market has now expanded to include routine analysis for process and quality control as SFC is continuing to gain acceptance as a complementary technique to GC and LC.

Connie Nathan  
Barbara Hackbarth  
Development Engineers  
Analytical Products Group

results in different elution times from the column. The HP G1205A system can be used with capillary, narrow bore, or packed columns. Viscosity of a supercritical fluid is at least one order of magnitude higher than the viscosity in the gaseous state, but is one to two orders of magnitude less than in the liquid state. This translates into a pressure drop across the column that is ten times greater with HPLC than it is with SFC. The lower viscosity of SFC allows longer columns, which yield better separation of closely related compounds.

Selectivity refers to the selective physicochemical interactions between the sample components and the column. In SFC, the selectivity of compounds is adjusted by changing temperature, mobile phase density, and/or composition of added modifiers. These control parameters are a combination of those available in either a pure GC or a pure LC application. The column is installed in a temperature-controlled environment, the GC oven.

**Detection.** SFC not only supports both capillary and packed columns but also a variety of GC and LC detector options. Detectors are devices that sense the presence of the different compounds eluting from the column and convert this information into an electrical signal. Selection of which detector to use depends on several factors including the type of information needed from the analysis, the sensing level required, and the complexity of the compounds.

GC detectors that are available in SFC are the flame ionization detector, the nitrogen phosphorus detector, and the electron capture detector. The electron capture detector is a halogen-sensitive detector primarily used in pesticide analysis. The flame ionization detector is a universal detector for a wide variety of organic compounds. The nitrogen phosphorus detector is a selective detector for compounds containing nitrogen and phosphorus, and is also used for pesticide and clinical applications. A new dimension to detection capability is made possible with the use of modifiers with the electron capture and nitrogen phosphorus detectors. The flame ionization detector and the nitrogen phosphorus detector required design changes for SFC use.

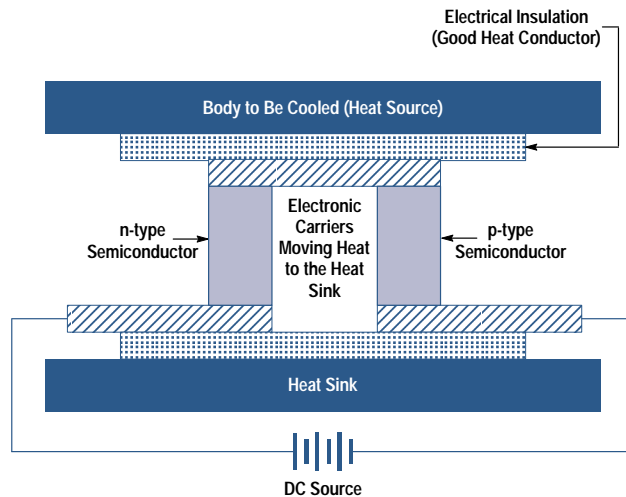
LC detectors available in SFC include the multiple wavelength detector, the variable wavelength detector, and the diode array detector. These cover application needs for high sensitivity and selectivity.

**Data Acquisition System.** The data acquisition system sorts and executes the many different signals, functions, and commands. For example, the first thing the instrument needs to know is how it is configured (one pump, two pumps, which detectors, injection method, etc.). Commands are given to operate the system at particular conditions (fluid flow rate, pressure, temperature, etc.). Electronic signals from the detectors are collected and converted to useful chromatographic information. Analytical results are stored and later presented in a meaningful report format determined by the user. The software platform for the HP G1205A SFC was leveraged from existing ChemStation platforms. Working through Microsoft® Windows as the operating control system, it can multitask with other applications and network with other data systems.

The remainder of this article focuses on the thermal design of the SFC pump and one of the detectors.

### SFC Pump

The pump module is an LC pump optimized to operate with supercritical fluids at low temperatures and high pressures. Supercritical fluids such as CO<sub>2</sub> are pumped through the system at rates as high as 5 milliliters per minute and pressures as high as 400 bars. The initial approach was to use cryogenic CO<sub>2</sub> to cool the pump surface. The proposed design was a channel of metal tubing with cryogenic CO<sub>2</sub> flowing through it. Using a network of channels, the cryogenic CO<sub>2</sub> is evenly



**Fig. 2.** Thermoelectric (Peltier) cooler.

distributed through the pump and provides an efficient method of cooling. The design could be further optimized by using materials with high thermal conductivity for the pump head to improve the efficiency and enhance the cooling.

The HP supercritical fluid extractor (SFE), introduced in 1990, used a similar approach. The SFE uses a supercritical fluid to extract the sample and prepare it for analysis. The SFE pump, like the SFC pump, must be cooled. Cryogenic CO<sub>2</sub> through a single stainless steel tube is used to cool the surface of the LC pump head. Although this design met the functional objective, it required an additional source of CO<sub>2</sub>. Supercritical fluid grade CO<sub>2</sub> is not used for cryogenic purposes because it is more expensive and a purer form of fluid.

To make the product more compact and self-contained, a thermoelectric (Peltier) device was investigated. Thermoelectric cooling provides one of the simplest means of refrigerating electronic equipment without using compressors or liquid refrigerants. A thermoelectric device is a heat pump that operates electronically. The device is made from two semiconductors, usually bismuth telluride, that either have an excess (n type) or deficiency (p type) of electrons. Current passes through the dissimilar conductors creating a temperature differential across the device. Heat energy is transferred from the cold side (body to be cooled) to the hot side (heat sink) and dissipated. This is known as the Peltier effect. Fig. 2 is a schematic diagram of a thermoelectric device. Table I provides a comparison of a thermoelectric heat pump with other types of refrigerant cooling systems.

In applying thermoelectric technology successfully, three design issues to be considered are:

- The operating environment
- How to dissipate the heat efficiently
- Estimated amount of heat the device needs to remove.

To maintain a 5°C operating temperature, the final design (see Fig. 3) includes several thermoelectric devices sandwiched between an aluminum heat sink and a copper cold plate.

Preliminary results showed that the thermoelectric device could maintain a temperature of 4°C at the pump head surface. If the heat sink reaches the temperature of the hot side

**Table I**  
**Refrigerant Cooling Comparison**

	Thermoelectric	Cryogenic Fluids
Source	No liquid coolant	Requires liquid coolant source
Temperature Operating Range	Can accommodate temperature extremes (-100°C to +100°C)	Provides cooling only
Operation	Electronically controlled	Moving components or valves to control fluid flow
Physical Attributes	Modular and compact	Bulky with extra hardware (valves, tanks, etc.)

of the Peltier device, then the heat sink alone cannot effectively dissipate the heat. A boxer fan at the rear of the pump cools the heat sink. The interfaces between the Peltier cooler, the heat sink, and the cold side of the pump head are secured with screws and epoxy to minimize thermal losses and provide a good thermal circuit.

Further efforts to optimize the design included the selection of materials with high thermal conductivity. A nickel heat exchanger with porous sintered nickel is incorporated in the design to cool the CO<sub>2</sub> conductively before it enters the system. The temperature of the CO<sub>2</sub> as it enters the pump is near room temperature. The heat exchanger is in thermal contact with the cold plate. The cold temperature is conducted into the heat exchanger to lower the fluid temperature to 5°C.

The thermoelectric device combined with conductive and convective cooling techniques allows the pump module to be self-contained and modular. This approach requires no external source for cryogenic cooling.

### Flame Ionization Detector

As described above, the separation and detection module includes a choice of columns and detectors. The flame ionization detector is thermally optimized for SFC. It is mounted to the top of the GC oven, which controls the temperature of the column that contains the sample. The flame ionization

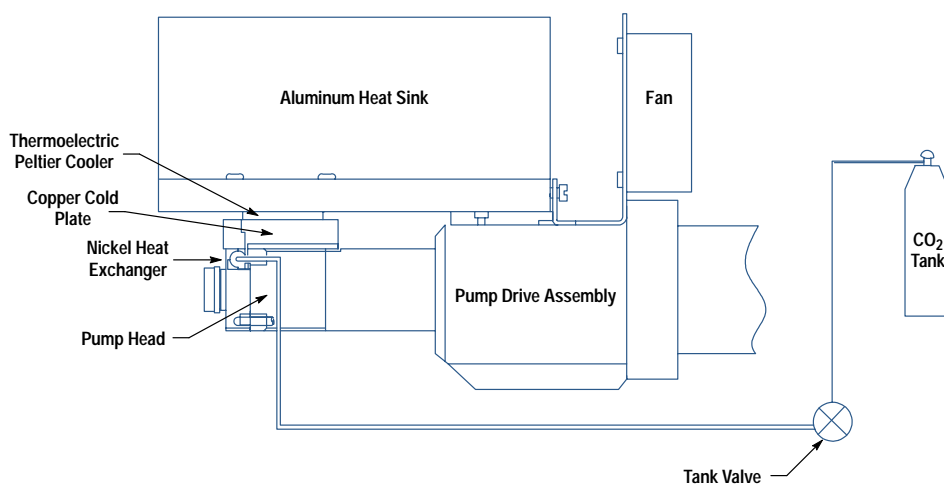
detector has two separate temperature-controlled heated zones. The base of the detector protrudes into the oven. The top of the detector, the collector, is a controlled zone exposed to room temperature. The oven and detector base both operate at temperatures up to 450°C with less than 0.5°C variations.

A flame ionization detector is a stainless steel block with a jet (a stainless tube) attached to it. Heat is applied by a rod heater. At the tip of the flame ionization detector jet, a flame is produced by the combustion of hydrogen and air. The sample elutes into the flame and ionizes. The detector generates a signal that represents the amount of carbon in the sample. Fig. 4 compares GC and SFC flame ionization detectors. The temperature of the zone just below the jet is critical. Slight changes in the temperature affect the reproducibility of the results.

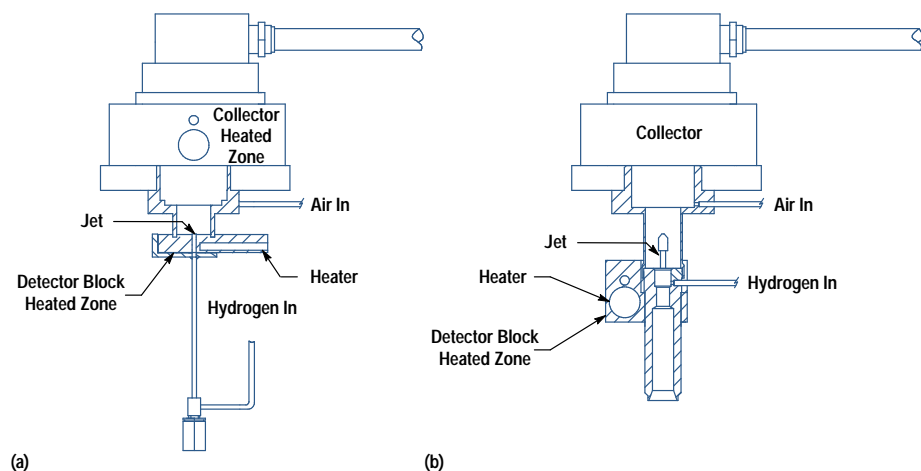
The GC flame ionization detector was evaluated for potential use with SFC. The GC detector has a long heated zone with a removable jet positioned above the zone as shown in Fig. 4. A temperature gradient exists between the heated block and the jet tip of the GC flame ionization detector because of the position of the heater and sensor and the location of the jet relative to the heater. When this GC design is used for SFC, spiking and flameout occur. Spiking is a fluctuation in the signal that shows up in a chromatogram. It is the result of the CO<sub>2</sub> icing. Flameout occurs if there is too much CO<sub>2</sub> flowing through the jet. A long heated zone also causes undesirable density drops in CO<sub>2</sub> that allow the sample to precipitate.

Customers familiar with flame ionization detectors expect the results to be clear and reproducible. The primary design consideration for optimization of the flame ionization detector for SFC was to solve the problems seen with the GC design, that is, to minimize density drops and eliminate spiking and flameout.

A shorter heated zone flame ionization detector was designed. The jet is an integral part of the heated zone (it is physically brazed to the zone) to ensure that there is no thermal break between the zone base and the jet. The heater and sensor are positioned close to the jet. The temperature gradient is minimized. The jet orifice size is optimized to prevent flameout at high CO<sub>2</sub> flow rates. A second heated zone is added to the SFC flame ionization detector to eliminate problems of spiking and condensation above the jet.



**Fig. 3.** Sketch of the SFC pump showing the major components and the thermoelectric (Peltier) device.



**Fig. 4.** Sketches of the SFC (a) and GC (b) flame ionization detectors. Note the difference in the heated block size and the location of the jet.

Condensation forms from the combustion of hydrogen and air when the detector block temperature is below 100°C. Table II is a comparison of the GC and SFC detectors.

**Table II**  
**Comparison of SFC and GC Flame Ionization Detectors**

	SFC Detector	GC Detector
Physical Size	6 mm thick, compact with jet permanently attached	25 mm thick with jet a separate part
Temperature Gradient	1°C from sensor to jet tip	17°C from sensor to jet tip
Problem Area	No spiking or flameout	Spiking and flame-out when used as an SFC detector
Collector	Heated collector zone	Collector not actively heated

The SFC flame ionization detector is insulated from the oven so that the heat from the oven has minimal effect on it. The SFC detector's minimum operating temperature is the oven temperature plus 10°C because of convection and conduction of heat from the oven.

To achieve temperature stability, the SFC flame ionization detector heated zone required changes in the temperature control parameters. The controller uses the Ziegler-Nichols algorithm<sup>1</sup> to obtain the proportional (P), integral (I), and derivative (D) terms in the control algorithm. This algorithm defines the PID constants such that the object reaches its desired steady-state temperature without large offsets or deviations. The PID controller is designed for larger masses such as the GC heated detector zones. Although it takes longer for larger masses to heat up, the oscillations are minimal and therefore it is easier to dampen the response and bring temperature under control. The SFC flame ionization detector is 1/4 the size of the GC design, so it heats up very quickly. Using the GC PID constants, large oscillations were seen because of poor temperature control. New initial values for the PID parameters for the SFC detector were derived using the Ziegler-Nichols tuning algorithm. A trial and error method was then applied to fine-tune the PID parameters to eliminate oscillations.

## Results

The major result was the introduction of an HP G1205A SFC system in May 1992 that meets its performance goals. It was a challenge to reuse existing components that were not designed with SFC in mind. In SFC, control of temperature in different regions of the system is critical to the success of the chemical analysis and the performance of the instrument. To make components perform under conditions for which they were not originally designed was a major success for the team. Another tangible result was the reduction of product cost by consolidation and reuse of components. The product is a compact modular design and maintains the integrity of the existing designs it leverages. The introduction of a thermoelectric Peltier-cooled pump head provides an advantage over designs that use a second source of CO<sub>2</sub> to cool the pump head. Other HP analytical products have incorporated the Peltier device. The SFC flame ionization detector, with its shortened zone, has improved the quality of the analytical results.

## Acknowledgments

The work described in this paper is the result of several years of development efforts by the R&D project team. Specifically, Hans Georg Haertl is responsible for successfully incorporating the Peltier device into LC pump design. Hans Georg was an engineer on loan to the site from the HP Waldbronn Division in Germany. The other members of the project team included Chris Toney (project manager), Elmer Axelson (software), Paul Dryden (firmware), Bill Wilson (chemist), Terry Berger (chemist), Mahmoud Abdel-Rahman (firmware and hardware), and Joe Wyan (hardware). Our success must also be shared by several other functional groups (manufacturing, marketing, purchasing, and administrative support) which were instrumental in the release of the product. Without the combined efforts of these individuals, the SFC would not have met its targeted release date.

## Reference

1. G.K. McMillan, *Tuning and Control Loop Performance*, Instrument Society of America, 1990, Chapter 1.

Microsoft is a U.S. registered trademark of Microsoft Corporation.

Windows is a U.S. trademark of Microsoft Corporation.

# Linear Array Transducers with Improved Image Quality for Vascular Ultrasonic Imaging

This project not only achieved its goal of improving the near-field image quality of an existing transducer design, but also added two-frequency operation.

by Matthew G. Mooney and Martha Grewe Wilson

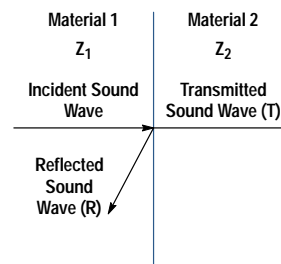
Medical ultrasonic imaging is a real-time technique that uses high-frequency sound waves to image many different parts of the body including the heart, vessels, liver, kidney, developing fetuses, and other soft tissue. The focus of this article is on noninvasive imaging of the blood vessels, which is more commonly referred to as vascular imaging. We begin with a general overview of ultrasonic imaging and then focus on the basic design aspects of a transducer used for imaging. Next, we examine the vascular market and the customer requirements, and then describe the design process used to develop two new vascular transducers. Finally, we present the clinical results.

The ultrasound waves used for imaging are generated by a transducer, which is held against the patient's body. The sound waves are produced by a piezoelectric material in the transducer. When a voltage is applied across the piezoelectric material, it deforms mechanically, creating vibrations in the material. These vibrations are acoustic (sound) waves and can have frequencies of 0.5 to 30 MHz. As the sound wave travels through the body and through various tissues, it bounces off the tissue interfaces, creating many reflections. The reflected sound waves are then detected by the transducer, providing information on the location of the tissues. A characteristic of the tissue called acoustic impedance determines the fraction of the energy that is transmitted from one tissue to another. The acoustic impedance,  $Z$ , is defined as:

$$Z = \rho c,$$

where  $\rho$  is the density of the material and  $c$  is the velocity of sound through the material.

Fig. 1 shows the sound wave at an interface and the transmission equation at that interface. In the body, the impedances are often very similar, so the system must be sensitive to these differences to distinguish blood from muscle, for example. As shown in Table I, the impedance of bone is much different, thus causing a great deal of reflection at that interface.<sup>1,2</sup> Since the sound waves cannot penetrate bone, natural "windows" in the body such as intercostal spaces between the ribs are often used. Attenuation, which is the result of absorption and scattering of energy within a material, creates another challenge in ultrasonic imaging.



$Z = \rho c$   
 where  $Z$  = Acoustic Impedance  
 $\rho$  = Density of the Material  
 $c$  = Velocity of Sound through the Material

Transmission Equation:

$$T = \frac{2Z_2}{Z_2 + Z_1}$$

Reflection Equation:

$$R = \frac{Z_2 - Z_1}{Z_2 + Z_1}$$

Fig. 1. Sound waves traveling between two materials.

Once the wave is reflected from the different interfaces, it travels back through the body and is sensed by the transducer piezoelectric material. The energy is then transformed into electrical signals which are propagated over a cable to the ultrasound imaging system, where all of the signal processing and image display occur.

Modern ultrasound systems are very complex and can consist of analog and digital portions. These systems process electrical signals in terms of amplitude and time, creating a real-time picture of the part of the body being scanned. Fig. 2 shows the HP Sonos 1000 cardiovascular ultrasound imaging system.

One type of ultrasound imaging system is the phased-array system. These systems have multiple channels for transmitting, receiving, and processing sound wave signals (64 and 128 channels are the most common). A typical transducer for these systems is divided into many individual transmitter/

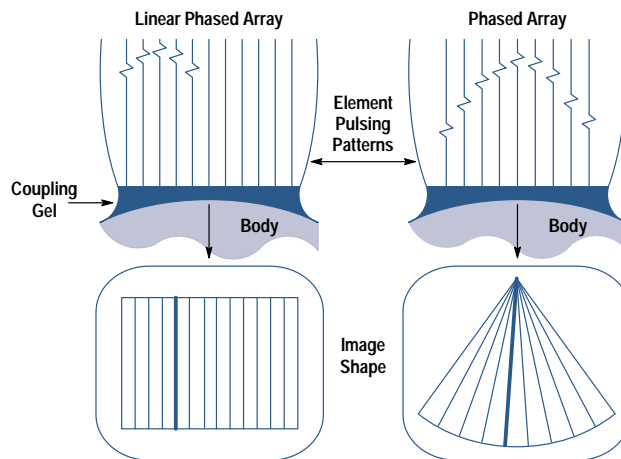


**Table I**  
**Characteristic Acoustic Impedances**  
**of Several Biological and Nonbiological Materials**

Material	Acoustic Impedance ( $10^6$ Rayls)
Air at S.T.P.	0.0004
Water at 20°C	1.48
Blood	1.67
Muscle	1.70
Fat	1.38
Soft Tissue	1.63
Kidney	1.62
Liver	1.64
Bone	3.8 to 7.4
Polyethylene, Low-Density	1.8
Vinyl (rigid)	3.0
Lucite	3.2
Valox, Black (glass-filled nylon)	3.8
Aluminum	17.0
Lead Zirconium Titanate	28 to 36
Stainless Steel	45.4



**Fig. 2.** HP Sonos 1000 ultrasound imaging system.



**Fig. 3.** Comparison of linear phased-array and phased-array ultrasound transducers.

receivers which are called elements. Typically, each transducer element is connected to one system channel. Two different types of transducers are commonly used: sector phased-array and linear phased-array transducers. The main difference is how these transducers are electrically excited. Each element in a sector phased-array transducer is activated at a slightly different time delay, which allows the sound wave to be shaped into a beam of sound and steered at different angles, producing a picture shaped like a pie slice. Linear phased arrays have the additional ability to activate groups of elements in a type of sequential scanning of the image, producing a rectangular-shaped picture. Fig. 3 shows a pictorial comparison of the element pulse patterns and the respective image shapes produced by the linear and phased-array transducers.

There are three main modalities in ultrasonic imaging: 2D, Doppler, and color flow. The 2D image is a real-time gray-scale image display. A typical 2D linear image of a carotid artery in the neck is shown in Fig. 4. Doppler is a way of measuring the flow velocity and movement within an image and is named for the principle it uses. The information is presented either with an audible tone or a visual plot. Color flow imaging detects the flow of blood and color-codes it depending on the direction and velocity of flow. An image showing color flow in the carotid artery is shown in Fig. 5.

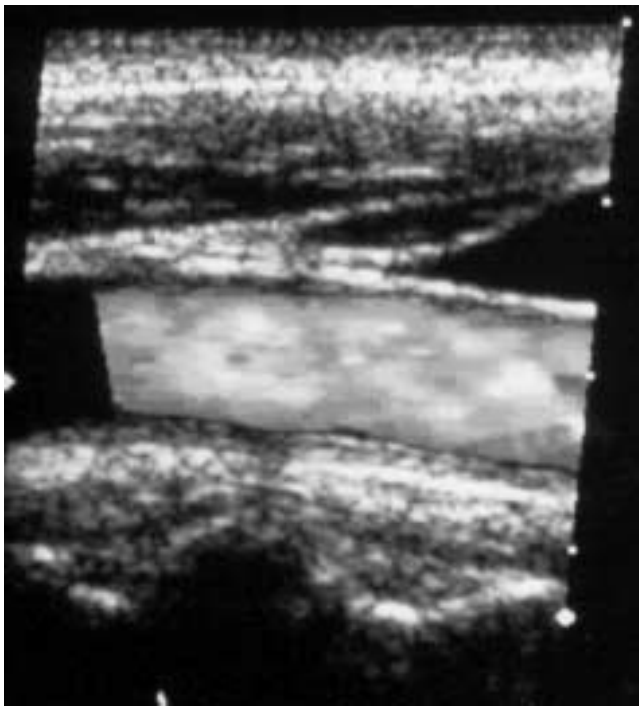
The references provide more detailed information on phased-array ultrasound imaging systems<sup>3</sup> and on color flow and Doppler processing.<sup>4</sup>

### Transducer Design

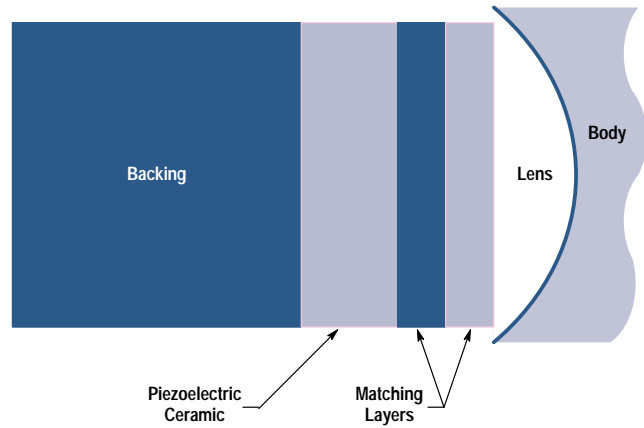
As mentioned above, a transducer consists of many small elements. An element is a multilayer sandwich of piezoelectric and other materials. The basic acoustic transducer element is shown in Fig. 6. Lead zirconium titanate, PZT, is a commonly used piezoelectric ceramic sensor material having an acoustic impedance between 28 and  $36 \times 10^6$  kg/m<sup>2</sup>s (Rayl). Recall that soft tissue has an impedance in the 1-to-2-MRayl region. Because of the impedance mismatch, there is a lot of reflected energy at the transducer/tissue interface, and such a transducer would not couple much energy to the human body. To improve coupling, a front matching layer—a coupling material with an intermediate acoustic impedance—is



**Fig. 4.** 2D linear phased-array image of a carotid artery.



**Fig. 5.** Color flow image of a carotid artery.

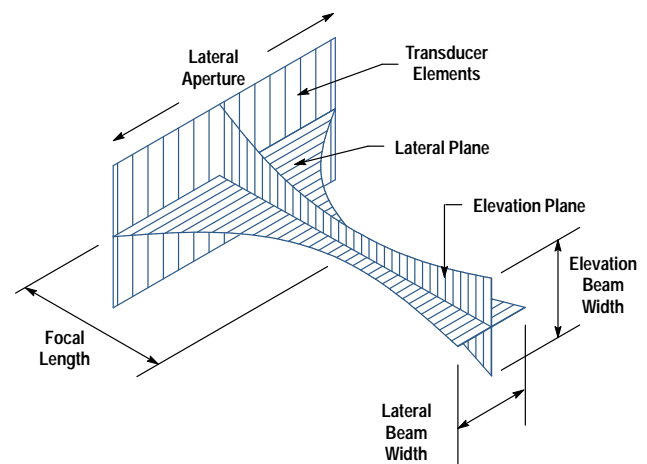


**Fig. 6.** Enlarged side view of one element of an ultrasound transducer.

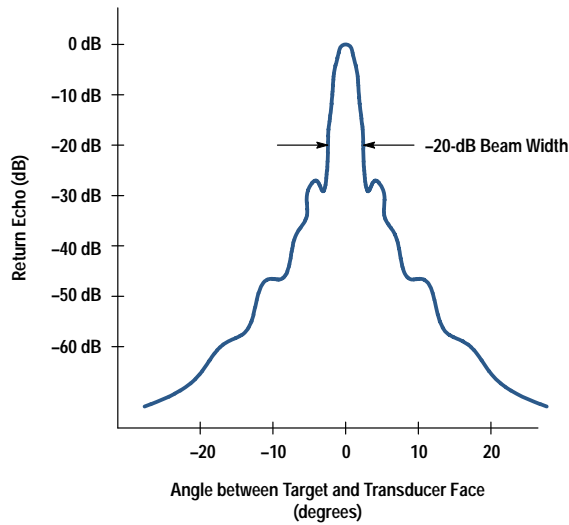
added to the front face of the piezoelectric to aid in transferring the sound wave more efficiently to and from the body. An acoustically absorbing material called a backing is added to the back of the sensor to dampen energy that might cause additional mechanical vibrations (greater pulse length).<sup>5,6</sup>

An important aspect in ultrasound imaging is the ability to detect and resolve small structures in the body. This is largely determined by how well the beam of ultrasound is focused. Beam focusing for linear and phased arrays is determined by two different measures: elevation beam width and lateral beam width. Fig. 7 shows a picture of a focused beam and the elevation and lateral planes.<sup>7</sup>

The lateral beam width is a measure of transmitted beam width in the lateral plane and it changes as a function of many parameters including distance from the transducer. It can be measured by keeping the transmitted beam fixed while moving a receiving hydrophone in an arc in the lateral plane. Lateral beam width can be changed by electronically switching elements on or off to change the aperture size. The electrical impulses delivered to the elements can be advanced or delayed in time to provide additional focusing in the lateral plane. Fig. 8 shows a typical lateral beam plot at some distance from a transducer. Beam width is extracted from the beam plot and is a measure of how wide the main



**Fig. 7.** Focused beam, showing elevation and lateral planes.

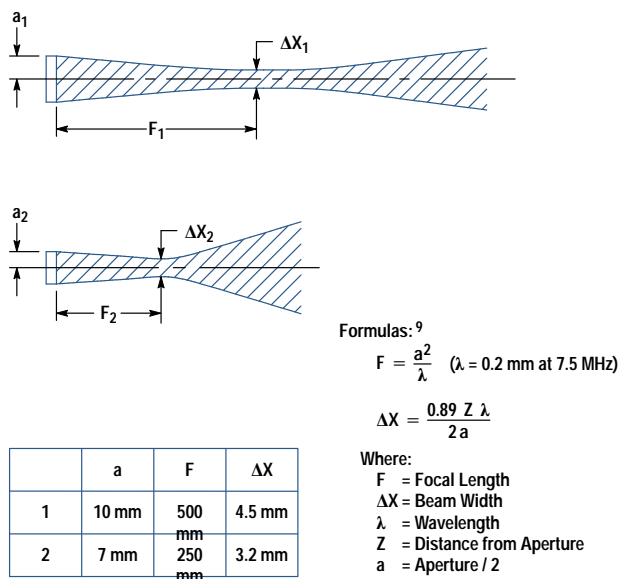


**Fig. 8.** Typical lateral beam plot at some distance from the transducer, showing beam width measurement.

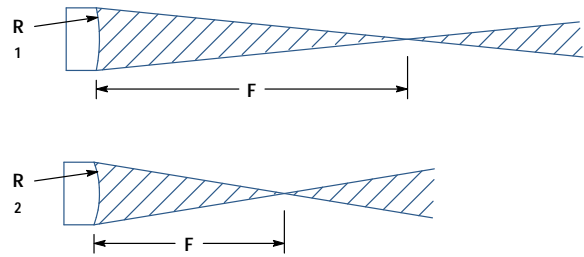
transmitted lobe is at 6, 10, 20, or even 30 dB down from the maximum.

The elevation beam width is a measure of transmitted beam width in the elevation plane. Like the lateral beam width, it varies with the distance from the transmitting element. It is different from the lateral beam width because the size of the transducer in the elevation direction is not the same as the size in the lateral direction. Also, since most transducers are not divided into multiple elements in the elevation direction, elements cannot be electronically switched to change the size of the aperture, or phased to provide additional focus in the elevation plane. As such, the choice of elevation aperture is critical to any transducer design. Fig. 9 shows the effect of aperture size on elevation beam width and focal point.

Instead of electronic elevation focusing, a lens is placed over the elements to provide some focusing in the elevation



**Fig. 9.** Effect of aperture size on elevation beam width and focal point.



	R	F
1	25 mm	50 mm
2	15 mm	30 mm

Formula:<sup>10</sup>

$$F = \frac{R}{\frac{V_B}{V_L} - 1}$$

Where:

- F = Focal Length
- R = Radius of Lens
- V<sub>B</sub> = Velocity of Sound in Body
- V<sub>L</sub> = Velocity of Sound in Lens

**Fig. 10.** Influence of lens radius on focal point and elevation beam width.

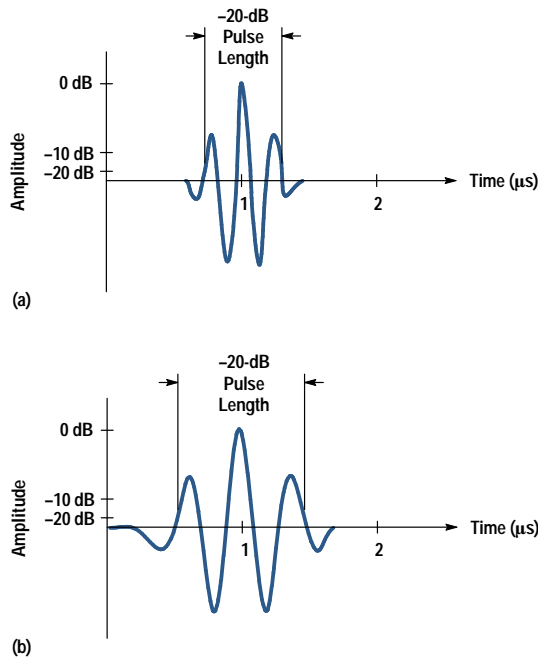
plane, but this focal point remains constant for a given choice of lens. The radius of curvature in conjunction with material properties of the lens determines where the narrowest point in the elevation beam will lie. This narrowest point is called the elevation focal point. The sketch in Fig. 10 illustrates how the lens radius influences both the focal point and the elevation beam width.

Another important factor in resolving small structures is time resolution. A well-focused beam allows small targets that are side by side to be resolved, while a short pulse length allows resolution of small targets that are separated by short distances into the body. Because the depth of an echo is determined by its time of arrival at the transducer, this attribute is called time resolution. Time resolution measures how well small structures can be resolved along the beam axis, so it is also called axial resolution. Axial resolution is measured in seconds of duration after excitation before the transmitted acoustic pulse fades to a certain level, usually 20 or 30 dB below maximum. The transducer can be designed to have a higher frequency, which increases axial resolution, but the body absorbs higher frequencies more, thereby reducing the depth into the body that can be imaged. Two pulses at two different frequencies are shown in Fig. 11. The higher-frequency transducer produces a shorter pulse. The choice of frequency requires a trade-off between axial resolution and depth of penetration.

### Vascular Linear-Array Transducers

Since each imaging application has very specific requirements, different systems and transducers are sold for different applications. The medical ultrasound market can be segmented into cardiology, vascular, radiology, and obstetrics. Hewlett-Packard's product line is focused on the cardiology and vascular markets. Our project developed two transducers for the vascular ultrasound market.

In June 1990, HP introduced the HP 21258A 7.5-MHz linear phased-array transducer. This transducer offered several new technologies, one of which is continuous steering of the image. Since color flow and Doppler imaging are based on the Doppler principle, they are most sensitive when the blood



**Fig. 11.** Comparison of ultrasound pulses from transducers operating at different frequencies. (a) 7.5-MHz pulse. (b) 4.5-MHz pulse.

vessels and the blood they carry point towards or away from the transducer. Unfortunately, the 2D image has its best sensitivity when the blood vessels run parallel to the transducer. Steering can be done on the linear arrays to account for the conflicting angle requirements. This continuous steering is made possible by combining 288 transducer elements into 128 signal paths, which match the 128 channels on the ultrasound system. The ability to steer the 2D image to get the best grayscale picture while independently steering the color flow image to get the best color filling allows the user to get the best of both modalities. 2D image steering is an important feature of HP linear array transducers.

The introduction of the HP 21258A was followed in April 1991 by the lower-frequency HP 21255A 4.5-MHz linear-array transducer. The HP 21255A employed the same technology as the HP 21258A, but the lower-frequency ultrasound energy of the HP 21255A could penetrate deeper into the body. As expected, the axial resolution was not as good as the HP 21258A, since the frequency was lower. Together, these two linear array transducers provided the vascular ultrasound user with alternatives to the penetration/resolution trade-off.

As with any new product, improvements were planned for the next version soon after the introduction of the version A transducers. The improvements were centered around customer feedback and were organized into two major categories: the large size of the transducers and the inability of the HP 21258A to resolve small structures close to the surface of the skin (near field).

A cross-functional project team was formed to define and develop new versions of the version A transducers. The version B vascular transducer team soon divided the work to be done into two categories to match customer needs: ergonomic improvements and near-field image quality improvements. The remainder of this article will discuss the process used to address the near-field improvements.

## Customer Feedback

The version B near-field image quality team set as an initial goal to be able to produce near-field images as good as the best competitor while maintaining our ability to produce good images in the far field.

Knowing that the near field needed improvement was a good start, but the project team required more detail. What did the customer define as near-field? Could this improvement be made at the expense of other attributes? How much improvement was needed? In addition to finding out exactly what the customer wanted, the project team also needed to figure out how to improve the near field. Should the frequency be changed? The aperture size? The focal point?

At this point in time, we needed a framework for organizing the information we had as well as some way to highlight areas that required more data. An organizational tool called quality function deployment (QFD) was identified as one way to help us translate what the customer wanted into a product. The QFD method is based on the construction of a "house of quality."<sup>8</sup>

The first step in building a house of quality is to tabulate customer wants and assign them weighing factors.<sup>8</sup> Given enough initial feedback, our team was able to develop and distribute a survey that listed many customer "wants" and asked for relative weights for each. The next step in building the house of quality is to tabulate those engineering characteristics that affect some or all of the customer wants. This list contains all the parts of the design that engineering could modify to give customers what they want.

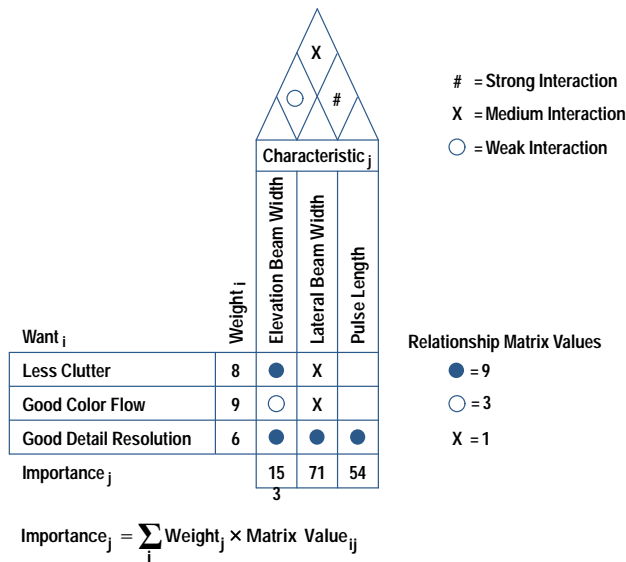
The final steps in the housebuilding process took the most time. The competition's products were benchmarked relative to HP's to see in what areas of customer wants we were winning or losing. The engineering characteristics were related to customer wants in matrix form so that we could see what characteristics affected which customer want and by how much. The final house of quality provided a graphical means of displaying all this data in a readable format. A small piece of our house is shown in Fig. 12. The actual house had 20 wants and 30 characteristics and several "rooms" like the example in Fig. 12.

## Engineering Design

The house of quality was an effective tool to show what we had to do to give customers what they wanted. Next the project team needed to change the design of the HP 21258A transducer to create the HP 21258B.

Using customer data, other rooms in the house of quality showed that the HP 21258A transducer was approximately equivalent to its competitors in the areas of color flow and Doppler performance. However, it was inferior to its competitors in near-field 2D image quality. The house of quality also showed that elevation beam width was the engineering characteristic most strongly related to 2D image quality. This became the first area of redesign on the HP 21258B.

The house of quality provided information that our customers considered the near field to be over a very specific depth range into the body. New HP 21258B designs were built and evaluated for elevation beam widths in this range. The HP 21258B designs included combinations of smaller elevation



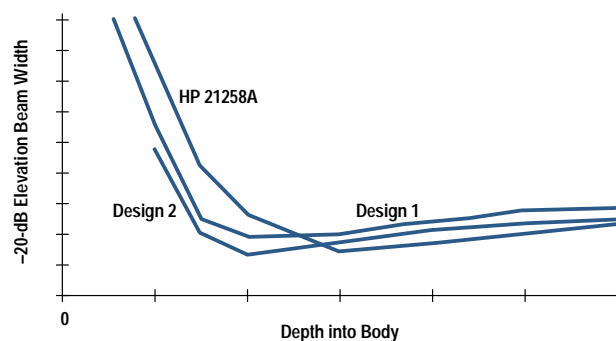
**Fig. 12.** A small portion of the QFD (quality function deployment) "house of quality" for the vascular ultrasound transducer.

apertures and tighter lens radii. The graph in Fig. 13 shows the elevation beam width of two such designs compared to the HP 21258A.

At least eight different designs for the 21258B were built and tested in terms of the critical few engineering characteristics shown to be important to customer satisfaction. A few of the most promising designs were taken to preference trials and tested against the competition.

### Initial Clinical Trials

The initial preference clinical trials were held in-house with internal experts and some invited outside vascular ultrasound users. Images were acquired using the various transducer designs and customers were asked to grade the images relative to each other. Some preference trials were conducted at customer sites. The initial comments on the new designs were very positive from users of the HP 21258A. According to these users, the initial HP 21258B was definitely an improvement over the HP 21258A, especially in the near field. Unfortunately, the HP 21258B designs did not fare well in the eyes of those users that had experience with our competition. One technologist summed up the best of our new designs as "better than the old one ... about 80% as good as my brand X."



**Fig. 13.** Elevation beam widths of the HP 21258A transducer and two improved designs.

The disappointing performance of the initial HP 21258B designs required reexamination of the engineering design to identify further opportunities for improvement. The next most important engineering characteristics as defined by the house of quality were not transducer design changes but ultrasound system changes involving lateral beam width. There were some ongoing investigations into system improvements, but with the preference trial results, these efforts received more attention.

In terms of transducer improvements, many of the secondary engineering characteristics such as pulse length were shown to be moderately important to customer wants. Not one but many characteristics would have to be changed to make large improvements in transducer performance. Since we had already been through preference trials, we were pretty far along the product development cycle and a redesign would have taken too much time. At this point, some technique to change the transducer design significantly in a short period of time was needed.

### Second Matching Layer

It was clear that an improvement in the axial resolution was required. A decrease in the pulse length would help. After investigating different ways to improve the pulse response of the transducer, one idea was to add a second matching layer to the sensor stack design.

Matching layers are very important to the transducer construction. The matching layer is attached to the front face of the sensor material and its main function is to help efficiently couple the energy to and from the body. A matching layer is one quarter of a wavelength thick to provide for constructive interference as the sound waves travel through it. By adding more than one matching layer, the energy is even more efficiently coupled, resulting in reduced pulse length. This would correspond to an improvement in the axial resolution.

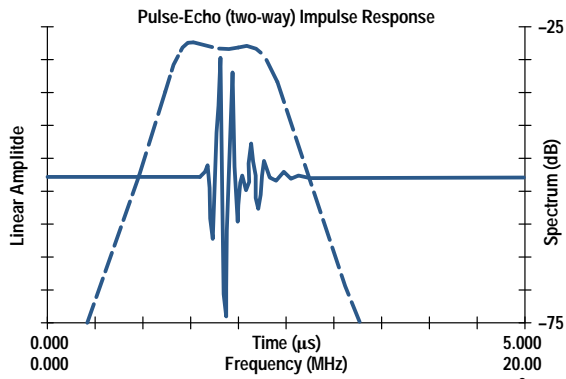
The investigation of adding a second matching layer began by using computer models that showed a reduction in the pulse length when a second matching layer was added to a sensor. A comparison of the two modeled 7.5-MHz designs is shown in Fig. 14. The next steps in the process were to define the desirable material properties for the second matching layer material and then select and test an appropriate material.

In terms of transducer design, there are two important material properties of a matching layer: the acoustic impedance and the attenuation. Typically, the preferred acoustic impedance of the matching layer is the geometric mean of the impedances of the two materials it is sandwiched between. For example, if the first matching layer has an impedance of 8 MRayls and the body is 1.5 MRayls, the desirable impedance of the second matching layer would be 3.5 MRayls:

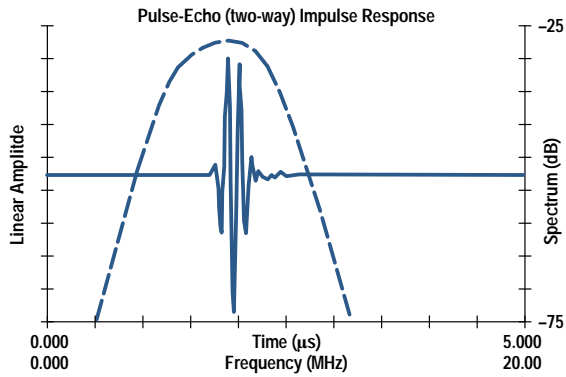
$$\begin{aligned}
 Z_{ML} &= (Z_1 Z_2)^{0.5} = (8 \text{ MRayls} \times 1.5 \text{ MRayls})^{0.5} \\
 &= 3.5 \text{ MRayls.}
 \end{aligned}$$

Ideally, the attenuation should be low to minimize the amount of energy lost when the wave travels through the matching layer.

In addition to the acoustic requirements, other material properties of the second matching layer were important. The material needed to be bondable since it was going to be



(a)



(b)

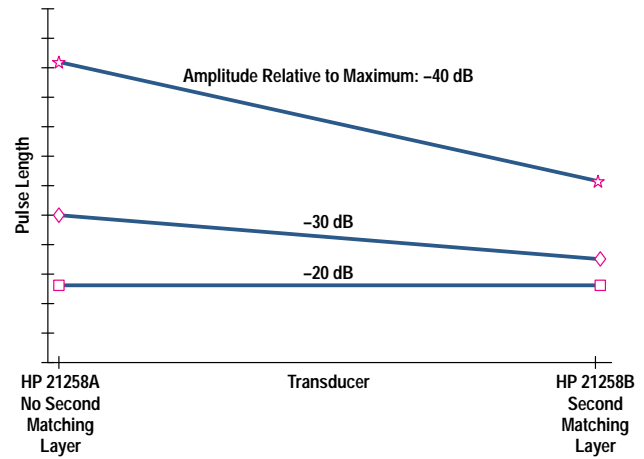
**Fig. 14.** (a) Pulse and spectrum of the modeled HP 21258A 7.5-MHz transducer. (b) Pulse and spectrum of the modeled HP 21258B 7.5-MHz transducer.

attached to the acoustic stack. It required good resistance to changes in temperature and humidity and resistance to chemicals to which the transducer would be exposed, such as acoustic coupling gel and disinfectants. The second matching layer also needed to be an electrical insulator for patient safety. To maintain consistent performance, each of the material properties needed to be homogeneous within the second matching layer.

Polymers generally have impedances between 1 and 4 MRayls as noted in Table I. To meet the acoustic requirements, many polymers were evaluated and the selection was narrowed based on the acoustic criteria. Next, more complete material property testing was done and a final polymer material was chosen.

To ensure that the second matching layer material selected was appropriate in terms of transducer reliability, extensive environmental testing was done. The testing included strife (stress to failure) testing in severe temperature and humidity environments. As a result of the strife testing, design changes were implemented to improve the robustness of the product.

A comparison of the acoustic response of transducers with and without second matching layers was done. The results showed that the pulse lengths are greatly reduced with second matching layers. Acoustic pulse and spectrum test data comparing 7.5-MHz transducers with and without second matching layers is shown in Fig. 15. The graph shows that there is a significant decrease in the  $-30$ -dB and  $-40$ -dB pulse lengths of the transducers with second matching layers.



**Fig. 15.** Comparison of pulse lengths at various transmitted beam amplitudes below the maximum for HP 7.5-MHz transducers with and without second matching layers.

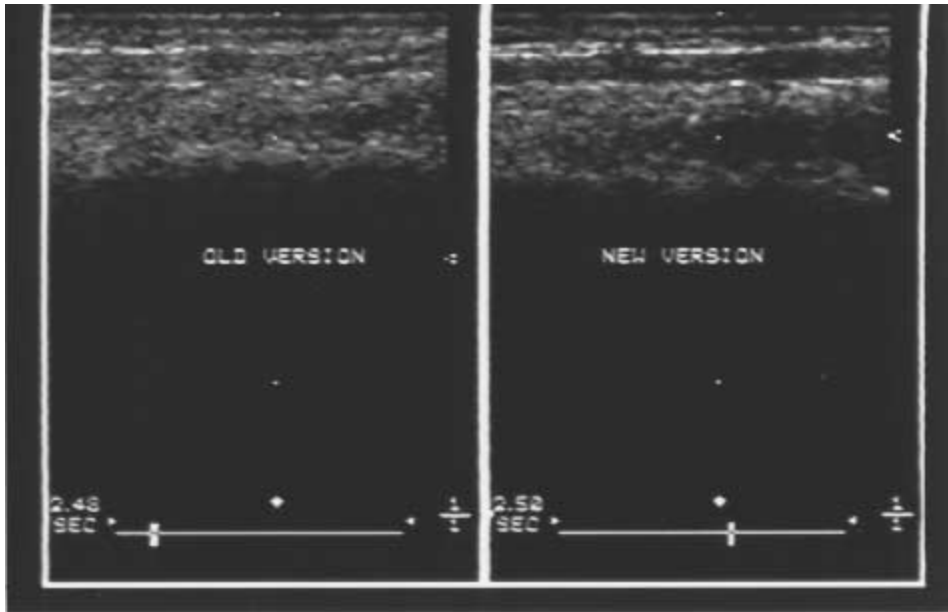
The main drawback to adding a second matching layer is acoustic cross-coupling between neighboring elements, which causes energy to be lost when the array is steered off-axis (roll-off). From the investigation, we found that the roll-off at  $45^\circ$  did increase slightly with the second matching layer. To gain a better understanding of how this would affect the 2D image quality and the color flow and Doppler performance of the transducer, clinical trials were done.

#### Further Clinical Trials

Several preference clinical trials were done in-house and at many different sites, both vascular and mixed cardiac/vascular. The main goal was to test the new version B linear arrays with added second matching layers against the version B linear arrays without second matching layers. In addition, clinical information was gathered on the performance of these new version B linear arrays versus the version A linear arrays and the competition's linear-array transducers. The two main areas of interest were the near-field resolution and the color flow performance.

The clinical trial results were very positive. A clear improvement in the 2D image quality was seen in all of the tests in which the version B transducer with the second matching layer was compared to the version B transducers without second matching layers. An example is shown in Fig. 16, which shows two 2D images of the radial artery in the arm (0.5 cm deep). The first image was taken with the version A 7.5-MHz linear array and the second was taken with the version B 7.5-MHz linear array. The near-field image quality is much improved with the version B transducer, as demonstrated by the clarity of the horizontal artery near the top of the righthand image in Fig. 16. The color flow and Doppler performances were determined to be about equivalent or slightly better. The version B transducer with the second matching layer also performed well against the competition's transducers.

Fig. 17 shows a bar graph comparing the clinical performance of the version A and B 7.5-MHz transducers with two competitors; the higher score indicates better performance. Overall, the near-field performance was improved, thus achieving the design goal.



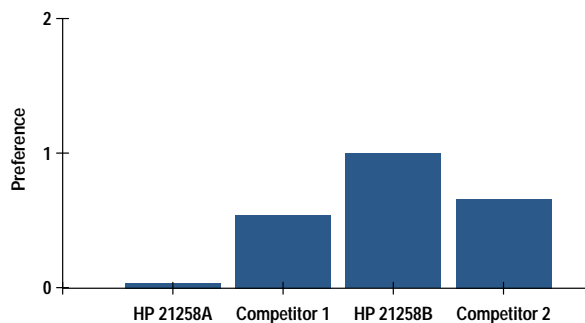
**Fig. 16.** 2D images of the radial artery in the arm. (left) HP 21258A transducer. (right) HP 21258B transducer.

### Additional Features

In addition to the shorter pulse lengths, the new linear arrays have the ability to operate at two frequencies as a result of the increase in their bandwidth. With some system changes, the HP 21258B became a 7.5/5.5-MHz transducer and the HP 21255B became a 4.5/3.7-MHz transducer. This dual-frequency feature enhanced the performance of these transducers in addition to making them unique at that time in the vascular imaging market.

Manufacturing methods were also improved for the version B linear phased arrays. Taking advantage of state-of-the-art assembly techniques in one manufacturing step resulted in a threefold decrease in cycle time while making the process easier for the operators.

Another feature that was added was the ability of the transducers to image an expanded view. This new imaging format is called trapezoidal imaging and shows a much larger area. Trapezoidal imaging was developed outside the transducer image quality improvement project and is not covered in detail in this article. A comparison of a typical linear array image and a trapezoidal image is shown in Fig. 18. The expanded view allows sonographers to see more of the larger structures such as the kidney all in one image. Feedback on this new feature has been extremely positive. Customers



**Fig. 17.** Clinical trial results showing measured user preference for image quality of vascular performance of approximately 7-MHz transducers. Preference is measured relative to the HP 21258A.

have reported better and faster diagnoses with trapezoidal imaging. The trapezoidal imaging format is another distinct feature of these new linear-array transducers.

The ergonomic portion of this project was also very successful. The size of the version B transducer was reduced by more than 25% compared to version A. New cable technology allowed smaller and lighter cables, resulting in a version B assembly that is two-thirds the weight of version A.

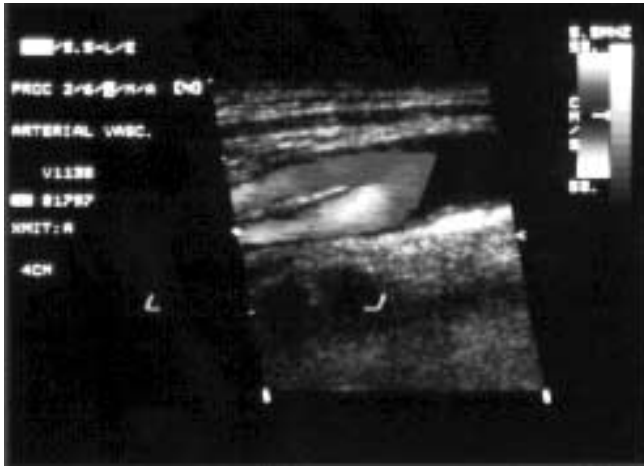
The version B vascular linear array transducers are shown in Fig. 19.

### Conclusion

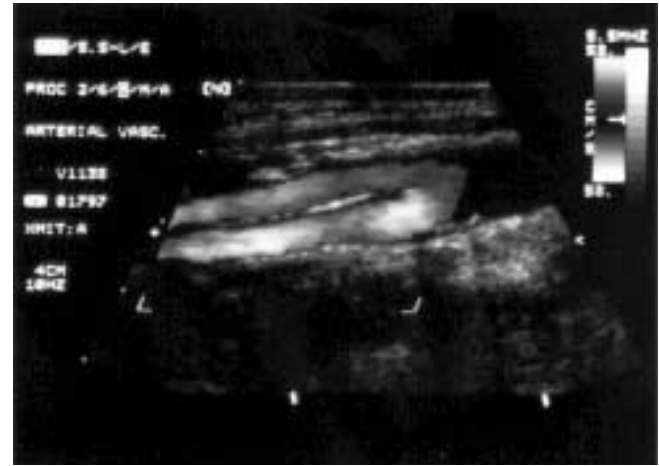
In June of 1993, the two version B linear phased-array transducers were introduced at the Society of Vascular Technology and the American Society of Echocardiology conferences. The transducers were well-accepted by physicians and sonographers, and the order rate for the version B linear phased-array transducers is several times that for the version A linear phased-array transducers. Today these new vascular transducers are helping clinicians provide more accurate diagnoses with improved image quality, improved ergonomics, and trapezoidal imaging format.

### Acknowledgements

Many people contributed to the success of the transducer portion of the version B linear array project. Rick Snyder championed steerable linear arrays at the HP Imaging Systems Division in the early stages. Ray O'Connell was the R&D manager for this project. Some of the original image quality team members were Martha Keller, Gary Seavey, and Ed Parnagian. The ergonomics team members included Ed Parnagian, Reggie Tucker, Martha Moriondo, and Greg Peatfield. The teams had great technician help from Linh Pham, Juliana Ciarla, Sandy Decker, Tony Cugno, and many other manufacturing operators. The engineering teams were later expanded to include Greg Vogel, Hashi Chakravarty, Troy Nielsen, Sean Cranston, Paul Cartier, Jon Rourke, and Dan Stempel. A lot of help came from marketing which was represented by Pat Venters and Stockton Miller-Jones. Dan



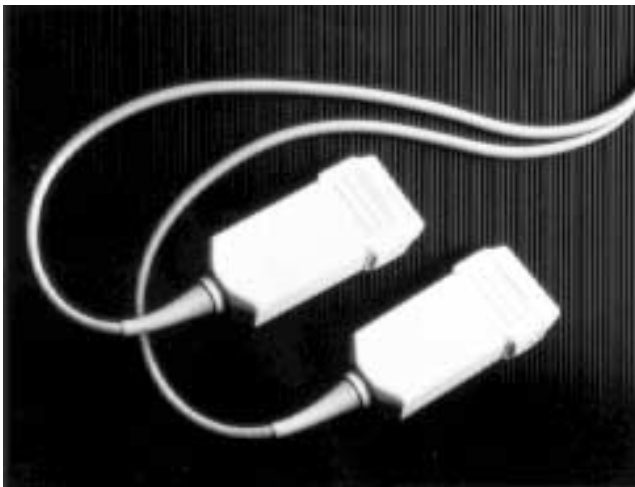
(a)



(b)

**Fig. 18.** 2D images showing the benefits of the trapezoidal format. (a) Linear-array format. (b) Trapezoidal format.

Cote was quite helpful with the QFD and customer surveys. Finally, Gail Zwerling, Paul Magnin, Ray O'Connell, Don Orofino, and Rick Snyder were all helpful in editing and providing advice in writing this article.



**Fig. 19.** HP 21255B and 21258B linear phased-array vascular ultrasound transducers.

### References

1. F. Duck, *Physical Properties of Tissue, A Comprehensive Reference Book*, Academic Press, 1990.
2. A. Selfridge, "Approximate Material Properties in Isotropic Materials," *IEEE Transactions on Sonics and Ultrasonics*, Vol. SU-32, May 1985, pp. 381-394.
3. *Hewlett-Packard Journal*, October 1983, entire issue.
4. *Hewlett-Packard Journal*, June 1986, pp. 20-48.
5. J. Larson, III, "An Acoustic Transducer Array for Medical Imaging—Part I," *Hewlett-Packard Journal*, Vol. 34, no. 10, October 1983, pp. 17-22.
6. D. Miller, "An Acoustic Transducer Array for Medical Imaging—Part II," *Hewlett-Packard Journal*, Vol. 34, no. 10, October 1983, pp. 22-26.
7. T. Szabo and G. Seavey, "Radiated Power Characteristics of Diagnostic Ultrasound Transducers," *Hewlett-Packard Journal*, Vol. 34, no. 10, October 1983, pp. 26-29.
8. J. Hauser and D. Clausing, "The House of Quality," *Harvard Business Review*, Vol. 66, no. 3, May-June 1988, pp. 63-73.
9. G. Kino, *Acoustic Waves*, Prentice-Hall, Inc., 1987.
10. D. Halliday and R. Resnick, *Fundamentals of Physics, Second Edition*, John Wiley & Sons, 1981.



# Structured Analysis and Design in the Redesign of a Terminal and Serial Printer Driver

The project team felt that the objectives could not be met with a traditional design approach. Structured analysis with real-time extensions and structured design provided an effective alternative.

by Catherine L. Kilcrease

This paper describes the use of structured analysis with real-time extensions and structured design in the redesign of the terminal and serial printer driver for the MPE/iX operating system on the HP 3000 computer system. The redesign project objectives were to:

- Maintain the current block mode performance (the main mode of data transfer for terminal I/O is to transfer characters in blocks of data)
- Improve HP 3000 transaction processing performance on industry-standard benchmarks by 5% to 10% through a 20% to 40% reduction in the terminal driver path lengths
- Maintain the current level of functionality
- Produce a high-quality, supportable, and maintainable product.

The project team felt we could not achieve these goals with the then-current development techniques. Object-oriented methods were ruled out because of the performance requirements. We elected to use structured analysis<sup>1</sup> with real-time extensions and structured design.<sup>2</sup>

## The Redesign Project

The original driver was based on the terminal driver of the HP 3000 MPE V operating system. During its design, specification of the terminal and printer subsystem was unclear and led to many problems. Since the original driver had added many features since its first release, it was important to have a complete specification of the subsystem to meet the project goals. Structured analysis provided this.

The original driver consists of seven modules that handle the I/O between the HP 3000 file system, the MPE/iX operating system, and the data communication and terminal controller (DTC) (Fig. 1). There are two storage managers. The terminal storage manager provides the interface between HP 3000 file system read and write intrinsic calls and the terminal logical device manager or fast write concat procedure. The serial printer storage manager provides the interface between HP 3000 file system write intrinsic calls and the serial printer logical device manager. High-level I/O is the old path between the file system and the logical device managers. It generally handles non-read/write I/O (controls, opens, closes). There are two logical device managers: one for terminals and one for serial printers. The logical device managers transfer data between the user stack and the data communication buffers for reads, writes, and controls. The fast

write concat procedure processes writes received from the terminal storage manager and sends them to the terminal and serial printer device manager (it provides a faster write path for terminals). The terminal and serial printer device manager communicates read, write, and control information to the data communication and terminal controller (DTC) through the Avesta Device Control Protocol. The lower interface of this protocol is the flow control manager. The flow control manager provides reliable transport between the HP 3000 and the DTC by implementing a transport protocol called the Avesta Flow Control Protocol. The storage managers and fast write concat procedure are invoked by procedure calls. The interface between the other modules in the driver and the operating system is message-based via MPE/iX ports.

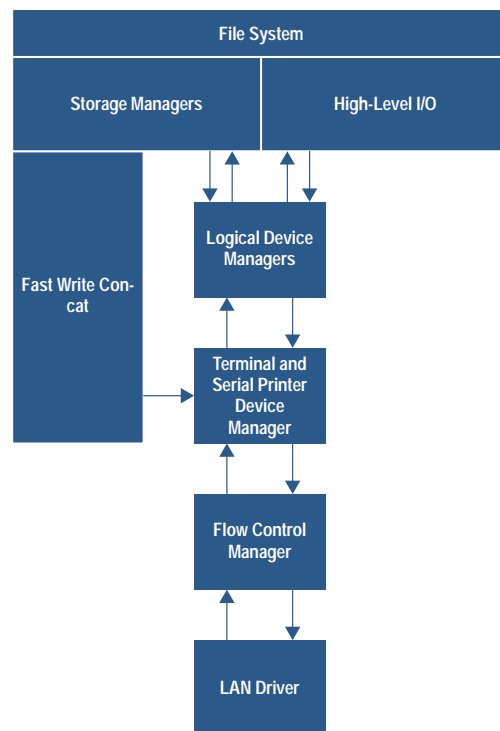
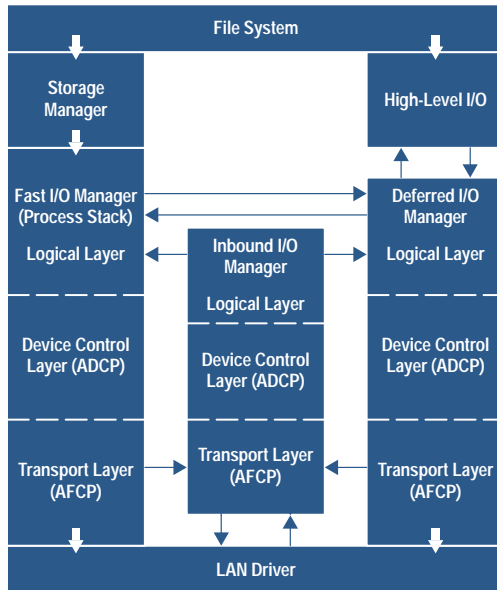


Fig. 1. Original driver architecture.



ADCP = Avesta Device Control Protocol  
 AFCP = Avesta Flow Control Protocol

Fig. 2. Redesigned driver architecture.

Path trace utility traces of the original driver were analyzed to determine good opportunities for path reduction. The redesign architecture then incorporated the best reduction ideas. The performance improvement is gained from streamlining the path for the most common I/O through the use of direct procedure calls, and from a design emphasizing efficient operation.

The new redesigned driver consists of three modules with three layers in each module (Fig. 2). The three layers are logical, device, and transport. The logical layer acquires the resources needed to complete the I/O request and transfers the data from (to) the user stack into (out of) a data communication buffer. The device layer handles the Avesta Device Control Protocol, which is the mechanism to communicate with the DTC. The transport layer implements the Avesta Flow Control Protocol (transport protocol) and interfaces with the LAN driver.

The three modules in the redesigned driver are the fast I/O manager, the deferred I/O manager, and the inbound I/O manager. The fast I/O manager is invoked by the file system with a procedure call. It handles the most commonly executed I/O: reads, writes, and terminal controls (e.g., change speed, parity, etc.). It attempts to process each request to completion. If it cannot complete the processing because of the lack of some resource, the fast I/O manager will block until the resource is available. If the I/O cannot be blocked (i.e., it is no-wait I/O), then the fast I/O manager sends the I/O to the deferred I/O manager. If the request cannot be completed because the "window" is closed at the transport level, the request is also sent to the deferred I/O manager. The deferred I/O manager has a message-based interface. It

handles deferred requests from the fast I/O manager and I/O requests that are made through the "old" high-level I/O path, such as open, close, and preemptive writes. The inbound I/O manager also has a message-based interface. It receives inbound packets from the LAN driver. If the packet is a reply to an I/O request, the inbound I/O manager sends a message to either the fast I/O manager or the deferred I/O manager, depending on which of the two initiated the request. It also handles asynchronous events.

### Software Life Cycle

There was some concern that structured analysis and structured design documents would not fit into documents produced by the product life cycle. With our recently revised life cycle<sup>3</sup> this turned out to not be an issue. Our software product life cycle contains the following phases and produces the following lab documents:

Phase	Method	Document
Proposal		
Investigation		Investigation Report
Development		
Specify	Structured Analysis	External Specification Internal Specification
Design	Structured Design	Internal Design
Integration/Test		Test Plan
Support		
Discontinuance		

The external specification document describes the environment in which the product operates, the functional capabilities of the product, and the details of the product's user interface. The internal specification describes the internal requirements of the system and the internal interfaces between the system components. The internal design contains the complete detailed description of the algorithms and data structures to be used in the implementation of the product. The test plan outlines the types of tests to be used to guarantee the quality of the finished product upon release from the lab.

### Training

There are four groups of people who need training in structured analysis: development engineers, inspectors, online and offline support engineers, and maintenance engineers. Training in structured design for the nondevelopment engineers is not necessary. The structured design document components are easy to comprehend. The project team took a class in structured analysis with real-time extensions and structured design at the start of the project during the investigation phase.<sup>4</sup> It would have been helpful to have had the training and some experience with the method before the start of the project. The structured analysis training for the nondevelopment engineers was developed by the project lead during the structured analysis phase before inspection of the internal specification. It

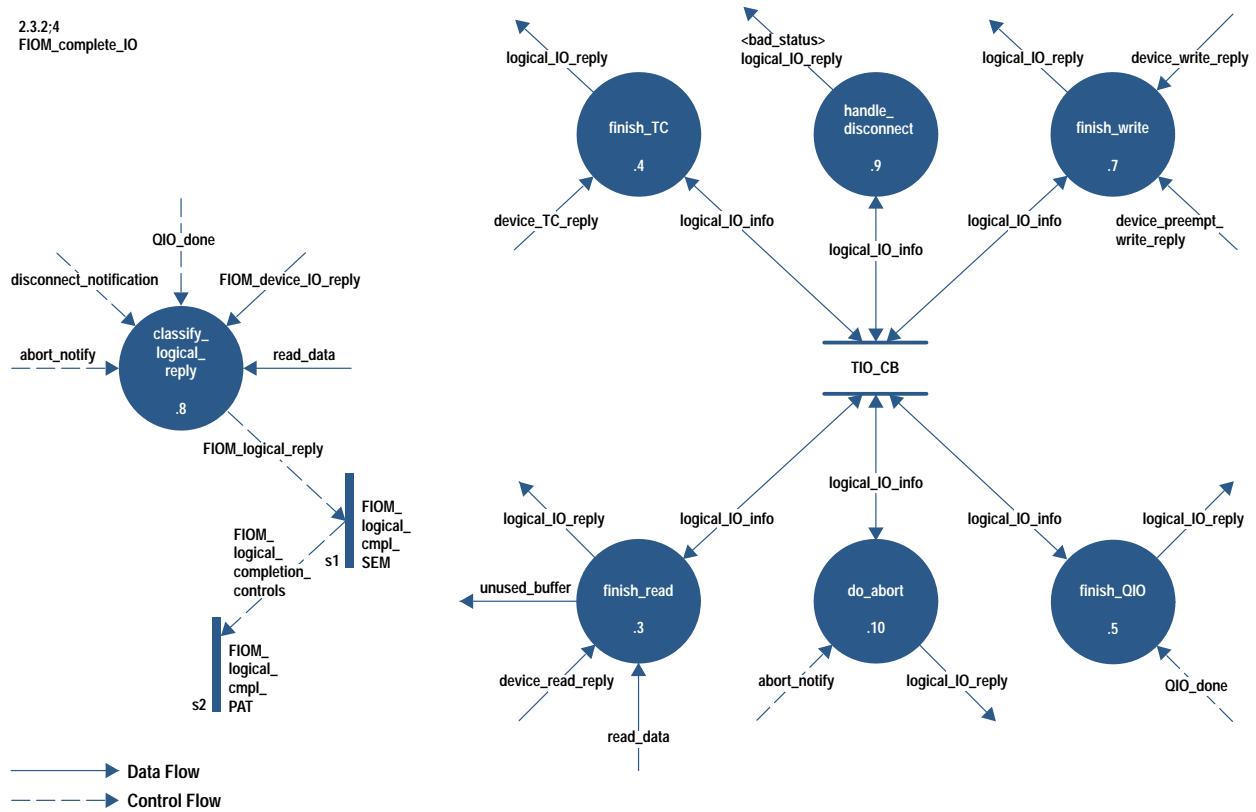


Fig. 3. Data flow diagram.

**Structured Analysis Overview**

Structured analysis is the use of tools to produce a structured system functional specification. A structured specification is easier to read and understand than the classical textual functional specification because it is graphical and contains many small specifications. The system is broken into small understandable pieces. The tools of structured analysis can be categorized into five functions. The redesign project used an extension of structured analysis for real-time systems. In structured analysis processes are independently data-triggered and infinitely fast (i.e., a process will transform the data when the data is present). Real-time extensions (i.e., the use of control information) allow the system to take other factors or conditions into consideration before enabling or disabling a process.

**Function**

- Partition the Requirements
- Describe Logic and Policy
- Show the Flow of Control
- Describe Control Processing
- Track and Evaluate Interfaces

**Tool**

- Data Flow Diagrams
- Process Specifications
- Control Flow Diagrams†
- Control Specifications†
- Data Dictionary

The data flow diagrams show the major decomposition of function and the interfaces among the pieces. They show the flow of data, not control. It is the system from the data point of view. Process specifications document the internals of the primitive data flow diagram processes in a rigorous way through the use of structured English, decision tables, or decision trees. They describe the rules of data transformation and the policy, not the implementation. Control

† Real-time extensions

flow diagrams share the same characteristics and relationships as data flow diagrams except that they deal with controlling the system. They show the flow of control in the system. A control specification converts input control signals into output control signals or into process controls. It has two roles: one to show how control is processed, and the other to show how processes are controlled (activated or deactivated). The data dictionary is an ordered list of data and control flow names and data and control store names and their definitions. Data flow diagrams and control flow diagrams can be combined together into one diagram.

Figs. 3, 4, and 5 illustrate the components of structured analysis with real-time extensions. Fig. 3 is a combination data flow diagram and control flow diagram. The solid arrows are data flows, the broken arrows are control flows, the solid vertical bars are state matrices, and the circles are processes. The finish\_read process transforms the device\_read\_reply (indicator that read data is ready) and the read\_data (buffer of data input by the user) data flows into the logical\_IO\_reply data flow using information from logical\_IO\_info. The freed buffer flows out (unused\_buffer data flow). The data dictionary entries for the data flows are:††

```
device_read_reply (data flow) = *read reply from device layer to      *
                               *logical layer. Contains read status,  *
                               *length, and data pointer.              *
```

```
status
+ length
+ data_pointer
```

†† Here the asterisks indicate comments, the square brackets indicate a choice of one of the enclosed items, the vertical bar means OR, and the plus sign means AND. TC means terminal control, QIO is quiesce I/O (flush outstanding input/output and wait for completion), RID means request identification number, and TIO is terminal I/O.

```

FIOM_device_IO_reply (data flow) = [ device_read_reply
| device_write_reply
| device_TC_reply
| device_preempt_write_reply
]

```

```

FIOM_logical_reply (control flow) = [ logical_read_reply
| logical_write_reply
| logical_TC_reply
| logical_QIO_reply
| logical_disconnect
| logical_abort
]

```

```

logical_IO_info (data flow) = FIOM_IO_pending
+ FIOM_IO_wait_port
+ DIOM_IO_pending
+ logical_RID_pending
+ logical_abort_RID_pending

```

```

logical_io_reply (data flow) = [ logical_read_reply
| logical_QIO_reply
| logical_TC_reply
]

```

```

read_data (data flow) = buffer_ID
+ read_status

```

The process specification for finish\_read (Fig. 4) describes how data is transformed. Control information is a little trickier to understand. For example, the FIOM\_device\_IO\_reply is transformed into a control flow, FIOM\_logical\_reply, by the classify\_logical\_reply process shown in Fig. 3. The control flow enters the state event matrix, FIOM\_logical\_cmpl\_SEM. The state event matrix has memory, that is, it remembers the state of the fast I/O manager. From the FIOM\_logical\_cmpl\_SEM event

```

NAME:
2.3.2.3:3

TITLE:
finish_read

INPUT/OUTPUT:
read_data : data_in
device_read_reply : data_in
unused_buffer : data_out
logical_IO_info : data_inout
logical_IO_reply : data_out

BODY:
transfer data (if any) to destination, doing backspace processing and freeing
unused buffers during the transfer;

send logical_IO_reply with status from device_read_reply or read_data msg;

```

**Fig. 4.** Process specification for process finish\_read.

matrix (Fig. 5), one can see that the finish\_read process is activated when the FIOM\_logical\_reply is a logical\_read\_reply and the state is read\_pending. Empty boxes indicate error conditions.

### The Project and Structured Analysis

During the investigation phase, the team considered five different architectures. The final architecture is a refined version of one of them. At the start of the development phase, we needed to start the specification of the system, define the architecture, determine the changes that were needed in the TIO support modules and operating system, and update the investigation report. Determining the changes we needed from the MPE/iX operating system lab and the project that handled the driver configuration modules would have made more sense in the design phase and not the specification phase, but it could not wait until then.

2.3.2-s1:4  
FIOM\_logical\_cmpl\_SEM

event state	FIOM_logical_reply = "logical_read_reply"	FIOM_logical_reply = "logical_write_reply"	FIOM_logical_reply = "logical_TC_reply"	FIOM_logical_reply = "logical_QIO_reply"	FIOM_logical_reply = "logical_disconnect"	FIOM_logical_reply = "logical_abort"
closed						
open_pending						
idle		finish_write/ idle			do_disconnect/ close_pending	do_abort/ idle
read_pending	finish_read/ idle	finish_write/ idle			do_disconnect/ close_pending	do_abort/ idle
QIO_pending				finish_QIO/ idle	do_disconnect/ close_pending	do_abort/ idle
TC_pending			finish_TC/ idle		do_disconnect/ close_pending	do_abort/ idle
close_pending						
close_timer_running						

Note: If an entry is blank, it is an "impossible" condition which should not be encountered due to subqueue restraints, etc. If the condition is hit, error code (not shown) will take appropriate action.

**Fig. 5.** State event matrix.

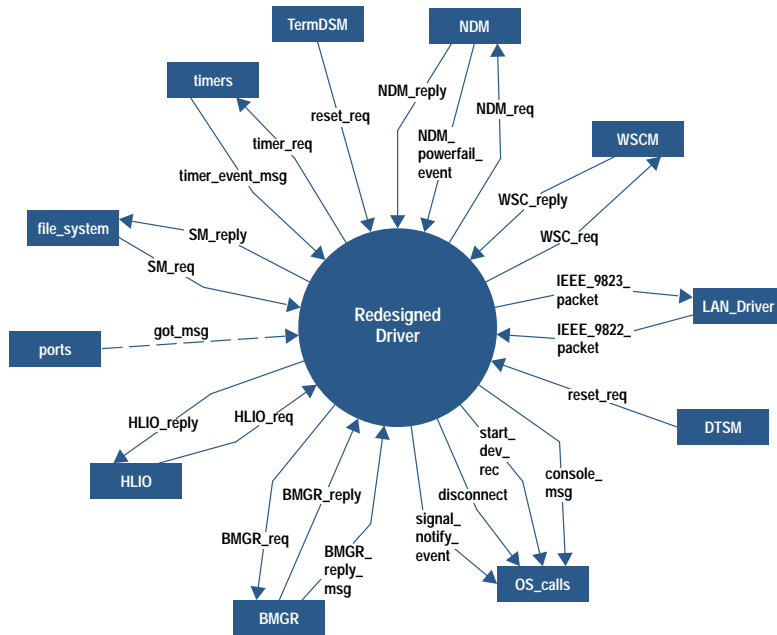


Fig. 6. Context diagram.

We started structured analysis using the fragmentation technique. This technique selects a set of inputs and outputs and creates a fragment model of processes that transform that set of data. The composite model is created by grouping the fragments. We tried to keep the system specification separate from the architecture. We broke the system into parts based on the type of I/O. For example, one fragment modeled the read path. It specified what happened with read requests that were processed completely without blocking for a resource, read requests that blocked for a resource (such as a datacomm buffer), and read requests that could not be processed because of a lack of a resource but could not block (no-wait read). The last type of read request could not be processed in a procedure call environment, but needed to be handled in a message-based environment (deferred I/O manager) so that the user process could continue running even though the read request had not been completely processed.

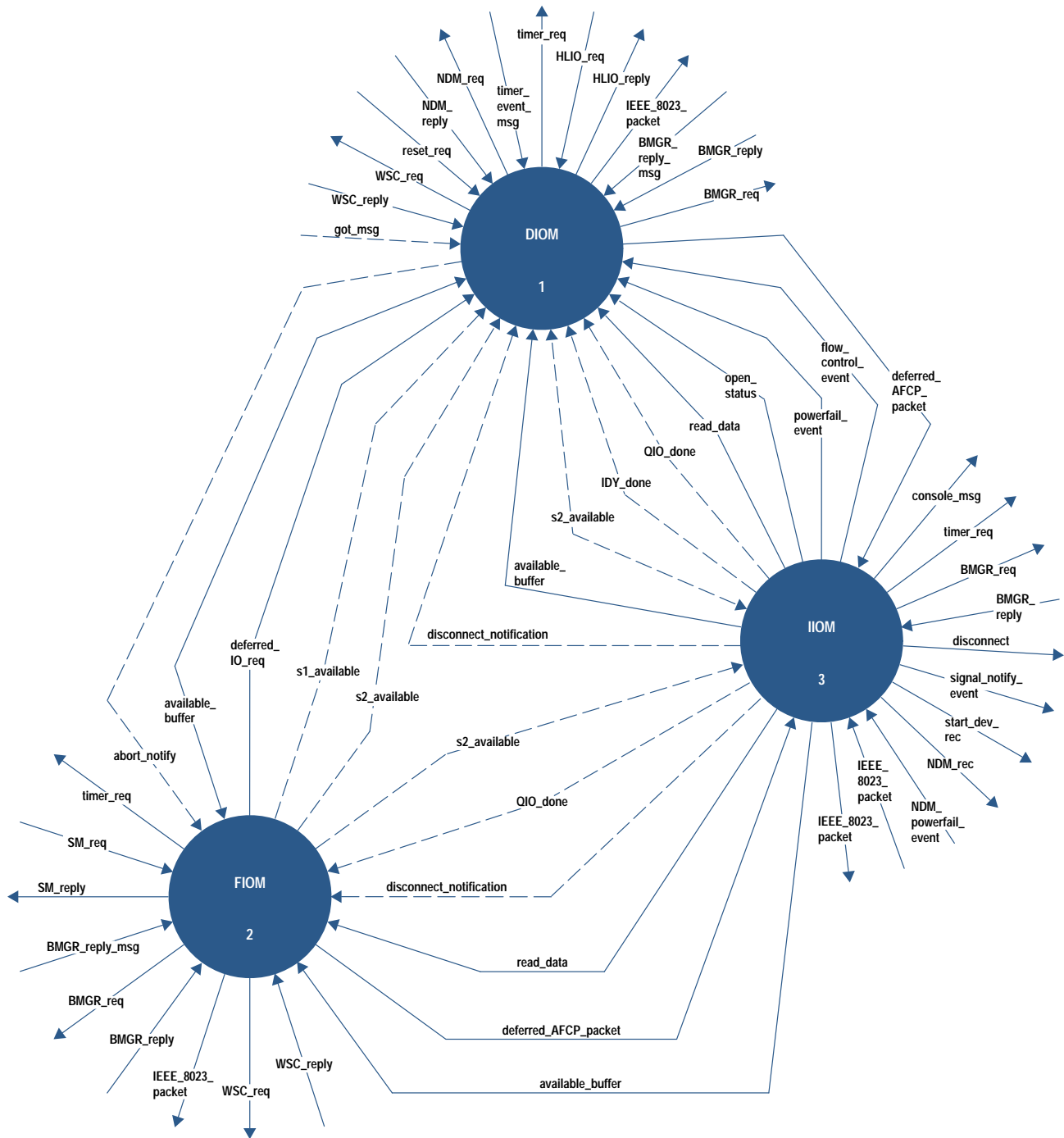
It became increasingly clear when we tried to tie the structured analysis fragments together that not taking the architecture into consideration was a problem. The goal of the redesign was to improve the performance while maintaining the same level of functionality. We had captured the functionality of the driver in our fragments based on type of I/O. However, each fragment contained fast paths (fast in terms of number of instructions—the fast path could block on a resource) and slower paths (required the message-based interface, which is much slower than a procedure call interface). It was difficult to figure out how to combine all the fast paths, which were spread out across many fragments. This is where one major difficulty with structured analysis arose—how to relate the functional specification to the architecture. The architecture had been selected as the best way to achieve the performance goals. We felt that to create the specification without consideration of the architecture would make the design phase more difficult. We stopped structured analysis work for awhile, and concentrated on completing the architecture. Hatley and Pirbhai<sup>5</sup> helped us resolve the architecture-versus-specification dilemma.

Viewing the system from the data point of view carried over into our parallel architecture discussions. At one point, we physically simulated data flowing through the driver using pens and erasers as data and people as modules. This helped us visualize the interface operation and problems that arise from a mixed procedural and message-based environment.

For complex areas, we used existing code wherever possible to derive decision trees and state transition diagrams. As we became more comfortable with structured analysis, we were able to assign work to each member. Material was reviewed and discussed at project meetings.

One aspect of structured analysis is the iterative nature of the method. One makes a first pass at the data flow diagram, and then discards or revises it until satisfied. Once we felt satisfied with our architecture, we set aside our old structured analysis work and started again. This time our approach was to use structured analysis to specify the system given the architecture instead of specifying the system independent of the architecture. We felt this was necessary to meet our performance goals and to help clarify the interfaces. Where before we based the specification on the type of I/O, this time we based the specification on fast (able to complete within the driver), deferred (needs operating system help to complete), or inbound paths. Using the data interviewing technique, we started with the context diagram (Fig. 6) and the level 1 and 2 diagrams (Figs 7 and 8). The level 1 diagram has three processes: DIOM, FIOM, and ILOM, which make up the redesigned driver. The level 2 diagrams (of which Fig. 8 is an example) have a process for each layer of the architecture and some general utility processes. After these diagrams were done, we broke the work up by process. We were able to use many of the diagrams from the earlier structured analysis work.

When reviewing data flow diagrams and process specifications, issues, questions, and problems were easy to detect. They tended to stand out on the diagrams. It was easy to see if data was missing or wrong or hadn't been initialized. For example, in the finish\_read process specification (Fig. 4),



**Fig. 7.** Driver level 1 diagram showing the three major modules: the fast I/O manager FIOM, the deferred I/O manager DIOM, and the in-bound I/O manager IIO.

unused\_buffer was a data flow out of the process but the buffer wasn't freed in the first draft of the specification. We kept a list of issues and questions and their resolutions during the structured analysis process.

A month before the external specification was due, we stopped structured analysis work and concentrated on writing the external specification document. Much of it was taken from existing documents since our upper and lower interfaces didn't change. The top-level structured analysis

diagrams (see Fig. 6) were used to determine interfaces and to help define the TIO support and operating system changes that the driver required.

Originally, we did not have an internal specification in our plans. However, the internal design was appropriate for the structured design but not the analysis, and we needed a document for the structured analysis work. Therefore, we split the design period into two periods and added an internal specification document. The internal specification contained

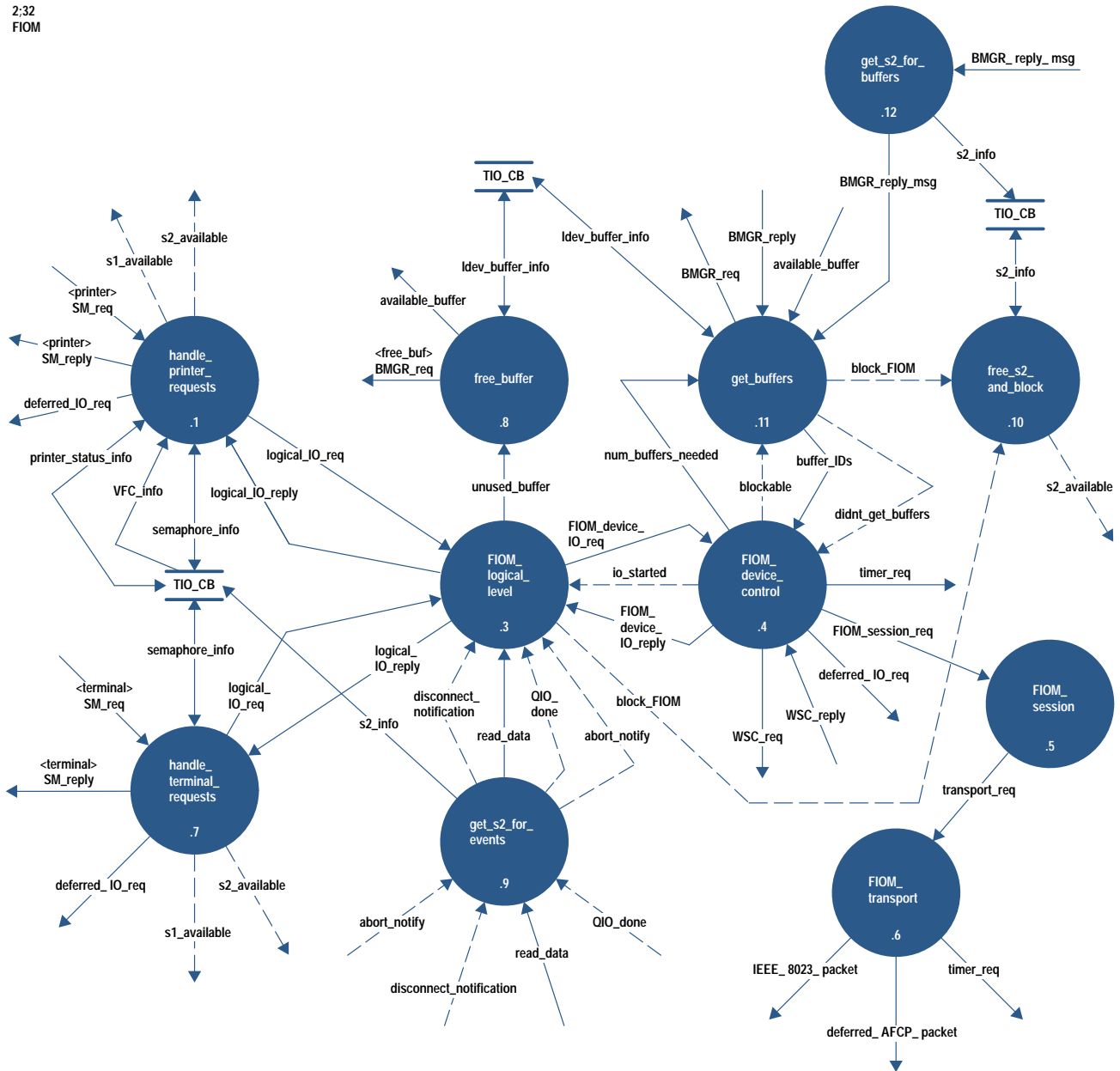


Fig. 8. FIOM level 2 diagram.

a brief introduction about the context diagram and descriptions of the interfaces. The rest of the document was generated using Teamwork, a software tool for structured analysis and structured design from Cadre Technologies, Inc.

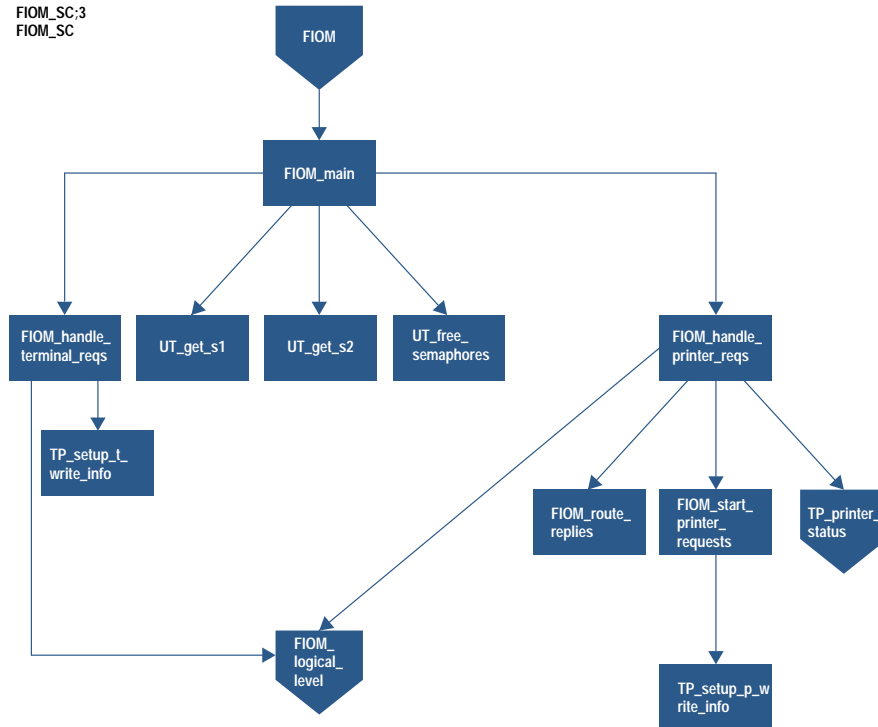
**Structured Analysis Recommendations**

Data flow diagrams generally “feel right or wrong.” Tom DeMarco<sup>1</sup> encourages engineers to throw away diagrams several times. The use of structured analysis to specify a system naturally raises the questions that need to be answered about the product.

If we had to do it all over again, we would have resolved the architecture-versus-structured-analysis problem earlier, and not tried to do structured analysis without considering the architecture of the system. We did not use the approach outlined by Hatley and Pirbhaj,<sup>5</sup> but we did use something related to it. There is a fine line between considering the

architecture and including design in the specification. Because this was a redesign and performance was important, we had already analyzed the original driver to find opportunities for shortening the path lengths. These opportunities needed to be incorporated into the design. We didn’t know how to do that at the design phase if they weren’t included in the specification as well, so we put the architecture in at the top levels of the specification (fast I/O manager, deferred I/O manager, inbound I/O manager).

We should have spent more time keeping the data dictionary entries up to date. All through the project, the lack of attention paid to data definitions was a major failing. The data needs to be defined as the diagrams and process specifications are created. Data dictionary entries that were related to data structures in the original driver were easy. However, we did not always type in the complete definitions. We did not document new data dictionary entries rigorously and this



**Fig. 9.** FIOM structure chart.

weakened the entire data dictionary. We would have also planned time for the internal specification, its inspections, and the rework in the schedule.

**Structured Design Overview**

Design is the process of transforming the specification of what must be done into the plan of how it will be done. Classical design produces a narrative document with some graphics. It starts with the procedural characteristics. Information is often repeated throughout the design document, and the document is generally not specific enough. This results in some design improvisation during implementation.

Structured design introduces structure and graphics into the design process to cope with the largeness of the system. The goal of structured design is a highly maintainable, comprehensible, and easily tested top-down design. The system is partitioned into components which interact to achieve the functionality of the system.

To create the structured design, the data flow diagram processes are grouped into processes that deal with inputs, deal with outputs, transform inputs into outputs, or handle transactions between inputs and outputs. Modules are then created from these groups. Evaluation and refinement techniques—called cohesion, coupling, and packaging—complete the building of the documents. The structure chart, design dictionary, and module specifications are the documents produced through this process. A structure chart shows the basic components (modules) and their interfaces in a top-down graphical manner. The design dictionary defines the interfaces. It uses similar language to the data dictionary of structured analysis. The module specifications define the procedural part of the design and the sequence of interactions. Each module specification describes what part of the specification is being satisfied, what the module needs to communicate, and how it performs the function. A module specification is written in structured English, as a decision tree or table, or as a state transition diagram.

Fig. 9 shows the first structure chart for the fast I/O manager. There are a main module, some utility modules (e.g., UT\_get\_s1), and device-specific modules (e.g., FIOM\_handle\_terminal\_reqs) which eventually led to the FIOM\_logical\_level module. Fig. 10 is the module specification for the fast I/O manager module, FIOM. It shows the sequence in which the other modules are called. Design dictionary entries look similar to data dictionary entries.

**The Project and Structured Design**

Once we had finished the internal specification, it was unclear how to derive the structured design from it. We decided on the following approach. Each structured analysis process was turned into a module. A hierarchical structure was developed for each level by “promoting a boss” or “hiring a boss” module when necessary. Comparing the fast I/O manager data flow diagram (Fig. 8) with the fast I/O manager structure chart (Fig. 9), one can see that the FIOM\_main module is “hired as a boss” (created) to handle semaphores and to call the FIOM\_handle\_terminal\_reqs and FIOM\_handle\_printer\_reqs modules.†

When the rough drafts of the structure charts were ready, they were entered into the Teamwork tool. The team then determined what each module does and knows. The next step was to walk through the design and be sure it works. This involved tracing each I/O through the design to be sure

† Note: Some of the process names are capitalized or spelled differently in Figs. 8, 9, 10, and 11 and in the text of this paper. Some names did change between the structured analysis document and the structured design document. In the module specification, Fig. 10, the engineer chose to capitalize the function codes, request types, and procedures called. In Fig. 11, the names are capitalized because the coding convention that was followed capitalized procedure calls. The MPE/iX operating system is case insensitive, so the differences are insignificant.



```

NAME:
FIOM_main:2

TITLE:
FIOM_main

PARAMETERS:
SM_CB
SM_arg_list
return_status

LOCALS:
request
status

GLOBALS:
BODY:
.....

Calling parms:

SM_CB    -- pointer to SM control block
SM_arg_list -- pointer to storage manager arguments
.....

TIO_CB := SM_CB^.fwrt_cb; * used to be fast write control block *

case SM_arg_list^.sm_generic_parms.func_code of

SM_READ_FN:
  request := READ_REQ;

SM_WRITE_FN:
  request := WRITE_REQ;

SM_CONTROL_FN:
  if (SM_arg_list^.sm_control_parms.t.* terminal control * <> TC_QUIESCE_IO)
    request := TC_REQ;
  else
    request := QIO_REQ;

SM_DEVCONTROL_FN:
  request := TC_REQ;

end;

* since this is the FIOM, and FIOM only handles waited I/O, always block *
* if necessary to get semaphores.

status := UT_GET_S1 (FIOM, TRUE);
if (status <> TRUE)
  * error, log and exit *;

status := UT_GET_S2 (FIOM, TRUE);
if (status <> TRUE)
  * error, log and exit *;

if (TIO_CB^.device_type = * terminal *)
  status := FIOM_HANDLE_TERMINAL_REQS (request, SM_arg_list);
else
  status := FIOM_HANDLE_PRINTER_REQS (request, SM_arg_list);

return_status := MAP_TO_OS_STATUS (status);

UT_FREE_SEMAPHORES;

```

**Fig. 10.** FIOM module specification.

that the right information was available to the modules, and that the modules were doing the right things. During the entire process, modules were collapsed when it was reasonable. We did not do much evaluation of the interfaces (data coupling or cohesion) because of a lack of time. The internal design document contained only Teamwork structure charts and module specifications.

After the internal design was inspected and the rework completed, we worked on defining the procedure declarations, data structures, software configuration (file structure), and coding standards. The module specifications were the basis for the Pascal procedures. Coding was mainly a matter of converting pseudocode to Pascal. Fig. 11 is the outer block of the fast I/O manager module. Comparing this to the fast I/O manager module specification, Fig. 10, one can see how closely related they are. The code adds more detail to the module specification framework. (Fig. 10 also illustrates a problem we had with module specifications: many of them

```

begin { tio_fiom }

try
begin { Try section }

tio_cb := tio_cb_ptr_type( fiom_cb_ptr );

l_index := 0;

{-----}
{ The FIOM only handles blocked I/O, always block }
{ if necessary to obtain semaphores }
{-----}
if ( CUT_GET_S1 (tio_cb, Fiom) ) then
if ( CUT_GET_S2 (tio_cb, Fiom) ) then
if ( not (tio_cb^.device_type.printer) ) then
  { terminal }
  status := FIOM_HANDLE_TERMINAL_REQUESTS
    ( sm_arg_list_ptr_t(arg_list) )
  else
  { printer }
  status := FIOM_HANDLE_PRINTER_REQUESTS
    ( psm_parm_ptr_t(arg_list) )
  else
begin
  { **** LOG S2 error **** }
  status := Bad_status;
  l_tio_status.int_status.status_code := Internal_err;
  l_tio_status.int_status.layer := Main_layer;
  l_tio_status.int_status.proc_number := Pn_tio_fiom;
  l_tio_status.int_status.location := 0;
  l_tio_status.int_status.llio_flag := False;
  l_tio_status.ext_status_hpe := status;

  CUT_LOGMSG ( tio_cb,
    l_tio_status,
    Main_layer,
    Fiom
  );

end
else
begin
  status := Bad_status;
  { **** LOG S1 error **** }
  l_tio_status.int_status.status_code := Internal_err;
  l_tio_status.int_status.layer := Main_layer;
  l_tio_status.int_status.proc_number := Pn_tio_fiom;
  l_tio_status.int_status.location := 1;
  l_tio_status.int_status.llio_flag := False;
  l_tio_status.ext_status_hpe := status;

  CUT_LOGMSG ( tio_cb,
    l_tio_status,
    Main_layer,
    Fiom
  );

end;

CUT_FREE_S2 (tio_cb);

CUT_FREE_S1 (tio_cb);

end; { Try section }

recover
begin { Recover section }

case ESCAPECODE of
0 ;;
otherwise
begin
  l_tio_status.int_status.status_code := External_err;
  l_tio_status.int_status.layer := Main_layer;
  l_tio_status.int_status.proc_number := Pn_tio_fiom;
  l_tio_status.int_status.location := 2;
  l_tio_status.int_status.llio_flag := False;
  l_tio_status.ext_status_hpe := hpe_status(ESCAPECODE);

  CUT_LOGMSG ( tio_cb,
    l_tio_status,
    Main_layer,
    Fiom
  );

end;
end;
end; { Recover section }
end; { tio_fiom }

```

**Fig. 11.** FIOM code.

were too code-like. The case statement is an unnecessary level of detail, and is actually not carried through into the code since the `arg_list` is passed to the `HANDLE` procedures). Analysis of the code shows a 33% reduction in code size compared to the original driver. The reduction comes from the reuse of procedures as a result of structured design, and from a structure that allows common routines to be shared between the DIOM, FIOM, and IIO modules instead of requiring one copy per module. In the original driver, there was a logical device manager for terminals and one for serial printers. In the redesigned driver, the fast I/O manager and deferred I/O manager are able to handle both terminals and serial printers.

### Structured Design Recommendations

In general, not enough attention was paid to defining design dictionary entries or data structures. Structured design added the next level of detail to the design from the specification. It was easy to develop the design from the specification once we had our approach. We occasionally created module specifications that are too much like code. Module specifications are not meant to handle the small details that are best suited to coding, but to define how the function is performed. If too much attention is paid to code-like details less is paid to how the specification is to be implemented.

If we had to do it all over again, we would have paid more attention to the interfaces and better defined the data passing between modules. Since we were using a lot of existing interfaces, we did not put the time into the data or design dictionaries. This meant that the data elements were poorly defined, since we tended to assume that everyone knew how the data was defined. We would also have developed a module specification standard to create consistency and avoid variations in the degree of code-like English in the internal design.

### Inspections

The internal specification required some basic structured analysis training for the inspections. This was because of the graphical nature of the diagrams and the decision structures. The internal design (structured design document) was easier to understand since it consisted of structure charts, pseudo-code, and data descriptions. Inspections were done by a depth walk through the processes and hence concentrated on architectural levels as opposed to interfaces between levels. The inspections did an excellent job of checking for functionality and design flaws, but were weak in the area of interface checking.

The original driver internal design documents were narrative with some state transition diagrams. They were much shorter than the internal specification and internal design created with structured methods. Past reviews of project documentation (external specification, internal specification, internal design) took place at one 2-to-4-hour meeting per document. Since the structured documents were larger and contained more detail, it took about 300 engineer-hours to inspect or review them. The participants felt that the reviews of the internal specification and internal design (structured

analysis and structured design documents) were very valuable. The material was much easier to inspect for completeness, correctness, and functionality than the narrative internal specification and internal design documents.

### Schedule

There is a concern that using structured analysis and structured design has a large impact on the schedule. The negative impact of structured analysis and structured design on our schedule consisted of training time (two weeks), time spent becoming familiar with structured analysis (one month of intermittent structured analysis), and longer inspections. A positive result of using structured analysis and structured design is that each step is a progression from the last. Each document is built on the previous document, unlike the classical design in which each document is not so closely tied to earlier ones. The effort required at each phase was less than the one before, and was mainly directed at adding more details. The positive impact of structured analysis and structured design on the schedule was shorter test cycles. Nonregression tests were passed ahead of schedule.

### Results

The redesign project improved block mode performance and achieved a greater than 30% reduction in the terminal driver path length. The code passed all functional tests including 24-hour and 72-hour nonregression tests. 75% of the testing errors were coding errors. The majority of the remaining defects were found in areas that were known to be weak from the design and inspection phases. The engineers valued learning structured methods and tools both for the quality and completeness of the design and the acquired skill set. The project team believes that the design of the new HP 3000 MPE/iX terminal and serial printer driver would not have been accomplished in the same amount of time with the same amount of thoroughness by the traditional design techniques.

### Acknowledgments

I would like to thank the project engineers—Jeff Bandle, Jon Buck, Larry White, and Erik Ness—for their hard work and team spirit, with special thanks to Jeff for his help with the figures for this paper. It was a pleasure to work with them as project lead engineer. Without them, this paper on our success with structured analysis and structured design would not have been possible. I would also like to thank the project manager, Celeste Ross, for her willingness to try something new and for her support, and the inspectors, reviewers, and moderators for their much appreciated time and effort.

### References

1. T. DeMarco, *Structured Analysis and System Specification*, Prentice-Hall, Inc., 1979.
2. M. Page-Jones, *The Practical Guide to Structured Systems Design*, Yourdon Inc., 1980.
3. *Grenoble Networks Division Product Life Cycle Guidelines, Edition 2*, Hewlett-Packard, April 1991.
4. R. Merckling, *Structured Analysis/Real-Time and Structured Design/Real-Time Class Notes*, 1991.
5. D. Hatley and I.A. Pirbhaj, *Strategies for Real-Time System Specification*, Dorset House, 1988.

# Data-Driven Test Systems

In a data-driven test system, all product-specific information is stored in files. Within a product classification, the test software contains no product-specific information and does not have to be changed to test a new product. This concept lowers new product introduction costs.

by Adele S. Landis

In today's competitive marketplace, new products must be designed, manufactured, and delivered to customers in an effective and timely manner. This is the only way to maintain or create a desirable market share. At the HP Santa Rosa Systems Division (SRSD), the introduction of a new product from the research and development laboratory into a production environment is called the new product introduction cycle. This introduction cycle can be lengthy and expensive. One of the most critical and time-consuming phases in the new product introduction cycle is the development of the electrical test process.

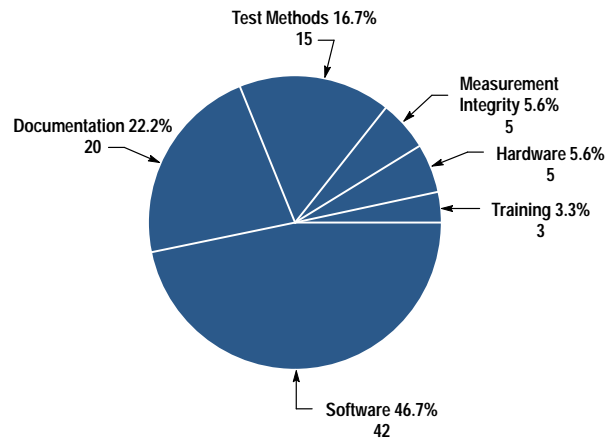
The electrical test process is that part of the manufacturing cycle that verifies that the product meets published customer specifications. The electrical test process is a combination of manufacturing resources, including:

- Test methods for doing each test that verifies a product's specifications
- Software that properly implements the tests
- The equipment necessary to perform the test methods
- Measurement integrity, which includes traceability, error analysis, and calibrations
- Documentation, including test procedures, test support documents, and the like
- Training of manufacturing personnel.

Fig. 1 shows an average distribution of the time spent on these resources for a group of electrical test processes recently developed at SRSD. The software portion of the process tends to be high because the division is moving towards completely automated testing.

Analysis has repeatedly shown that designing the test software by product classification rather than by individual products can result in large cost and time savings during the development process. A product classification can be defined as a group of products that have similar block diagrams and require the same type of testing. Some of the different product classifications that are manufactured at SRSD are network analyzers, spectrum analyzers, scalar analyzers, oscillators, amplifiers, and mixers.

Historically, production test software was written for and dictated by specific products or models. When test software was required for a new product or follow-on model, the test software was implemented or leveraged in a product-specific manner as shown in Fig. 2. This approach required continual introduction and/or reworking of the test software. The greater the number and variety of new products, the more time and software were needed. Also, this approach required

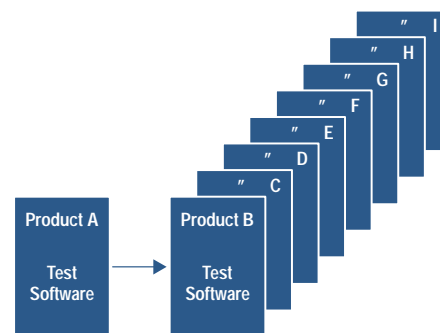


**Fig. 1.** Average proportions of time spent developing various manufacturing resources for a group of electrical test processes. Software dominates because the division is moving towards completely automated testing.

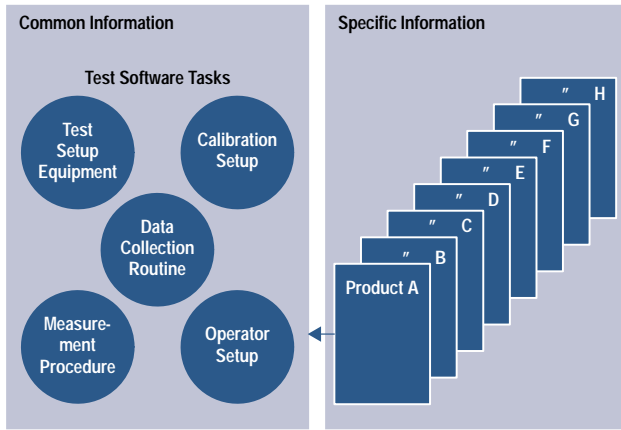
that an individual with software expertise be assigned for every cycle of new products that emerged from the lab.

## Data-Driven Software

Continued analysis has shown that by designing test software to handle large variations within a product classification, the overall development time of the electrical test process can be drastically reduced for all products within a classification. This greatly benefits both the manufacturer and the customer.



**Fig. 2.** Product-specific test software requires introduction or reworking of the software for each new product.



**Fig. 3.** In data-driven software, product-specific information is removed from the test software and stored in files. The software doesn't have to be changed to test a new product within the product classification.

A study of a large group of SRSD test programs across different product classifications was recently completed. The results indicate that all test software consists of certain types of tasks, including test equipment setup, calibration setup, operator setup, measurement procedures, and data collection.

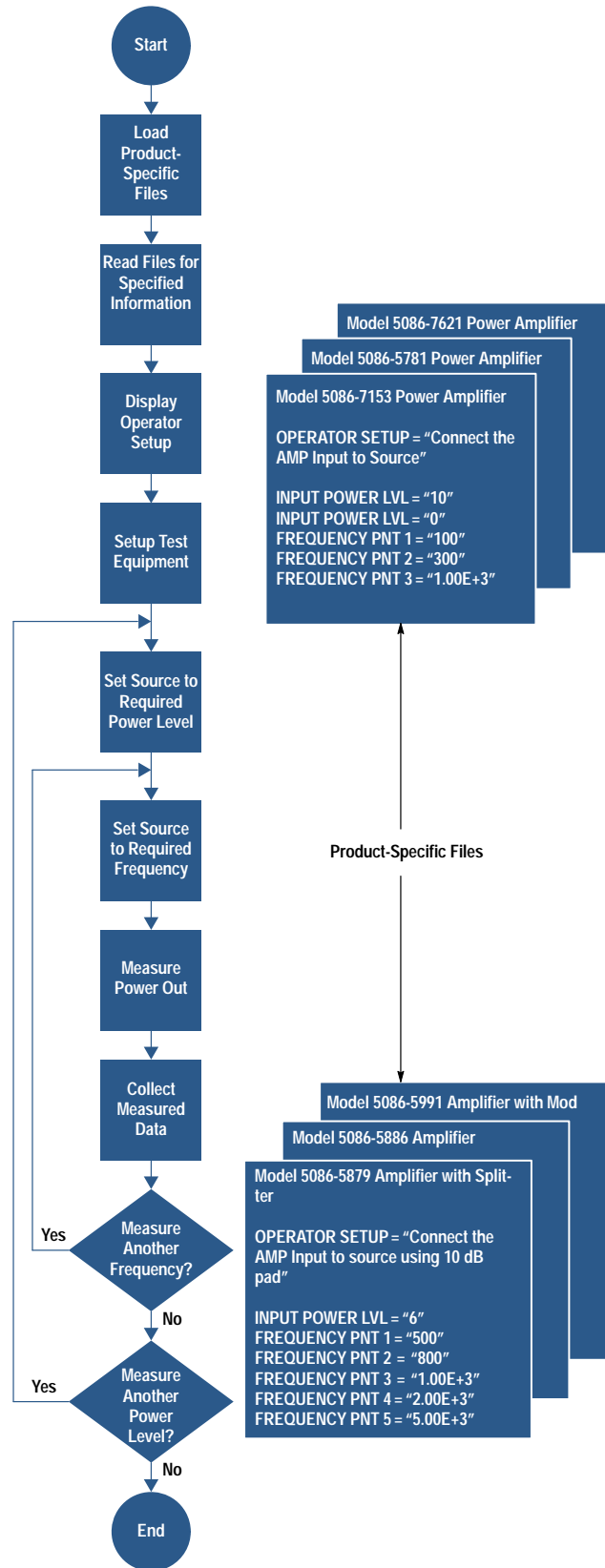
Regardless of which of these tasks are required for specification testing, all of these tasks can be broken down into two basic information types: common and specific. Common information is common to all products within the product classification. Specific information is specific to the individual products of that classification. Removing the specific information and placing it in external files leaves the common information to reside in the software (Fig. 3).

The separation of these information types results in static test software and dynamic product-specific files. The static test software contains the knowledge of the specific information required from the test's associated external file, the location of this file, how to load and read the file contents, proper measurement implementation, and the needed data collection routine. Since the test knows what information is required from its external specific file (but not necessarily the exact values), this file is considered defined.

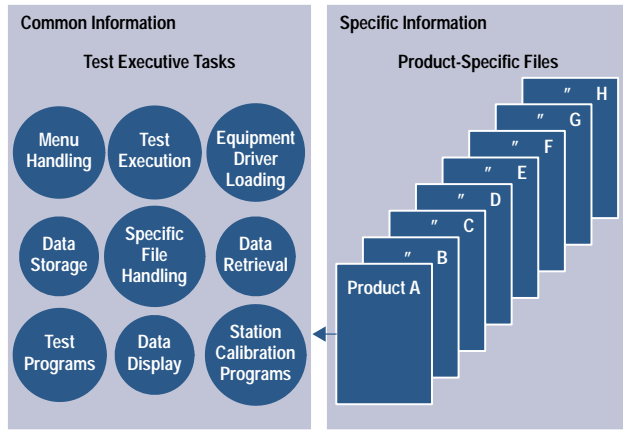
Test software that knows how to make the measurement, while the external files know where to make the measurement, can be called data-driven software.

Since data-driven test software is static, this method allows total test software reuse. Any product variation is handled in the external files. The building of the specific files can be automated easily since the type of information within these files has been previously defined by the test software. The specific file building is automated using a master form of the defined file and an editor. The editor only allows value entry. After the new values are entered into the master form, the product-specific file is stored. This approach, which does not require any software expertise, is extremely fast for new product setup.

A good example of data-driven software is an amplifier power test (see Fig. 4). The common information in this test, depicted by the software flow chart in Fig. 4, shows the tasks that are common to all amplifiers needing this type of test.



**Fig. 4.** Data-driven amplifier power test.



**Fig. 5.** Like data-driven test software, the data-driven test executive is made up of common and specific information.

The product-specific files contain the product measurement parameters and values required by the program to make the measurement. This same test software can be reused for any amplifier by building a product-specific file that contains the amplifier's measurement parameters.

### Test Executive

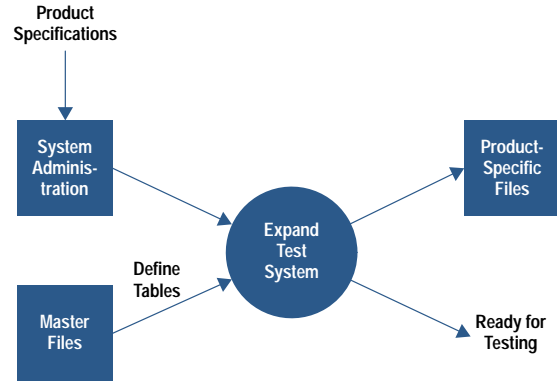
All manufacturers that are concerned with product test data need some form of a test executive. A test executive is a software package that handles the administrative and management responsibilities associated with test data. Even though the duties of the many test executives available vary depending upon user needs, all test executives have the same basic tasks as shown in Fig. 5.

Like the test software, test executive software is made up of common and specific information. By applying the data-driven theory to the test executive software, a data-driven test executive results. The test executive software then becomes static and any varying product information required by the test executive is available from external product-specific files. The building of these files can also be automated using master files and an editor.

The combination of data-driven test software and a data-driven test executive is a data-driven test system. The major benefit of a data-driven test system is that all the software within the test system is static. This benefit reduces the expensive and time-consuming cycle of product-specific software regeneration across a product classification. For test methods required by a new product that are not available in an existing data-driven system, the test method and associated test software must be developed. If the new tests are developed using the data-driven theory, then the tests will become part of the system's reusable tests and will be available for the next new product. Other key benefits are test system support and the automation of test system expansion.

### Expansion

A data-driven test system can be supported by a system administrator who resides on the production line where the test system is used. The system administrator can be a high-level technician who has some measurement and test equipment knowledge. Because the test system software is static, this



**Fig. 6.** Data-driven test system expansion.

individual does not need to have any software expertise. The system administrator's main duty is test system expansion.

Test system expansion is the process of adding a new product to a test system that already exists in the manufacturing environment. This process, whether for the traditional product-specific test system or a data-driven test system, can be broken down into two steps: adding the product into an existing test system and the initial product testing.

In the traditional product-specific test system, product addition takes approximately one day to eight weeks. This time frame is directly proportional to the amount of product-specific values that are imbedded in the test system software. Product addition in this type of test system must be performed by individuals with test software and test executive expertise. The time for product addition in a data-driven test system is approximately one hour and can be accomplished by individuals who have no test software or test executive knowledge.

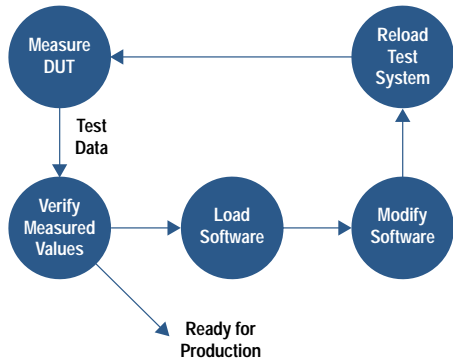
Expansion of a data-driven test system can be accomplished easily and rapidly with a programmatic procedure. The system expansion process is shown in Fig. 6.

The system administrator receives a new product and its specifications. Using a system expansion program, a set of master specific files, and an editor, the system administrator builds all the necessary external product-specific files required by the test executive and test programs.

### Initial Testing

After a product is added into a test system, the initial product testing must be done. This first testing is where the software used to measure the product specifications is verified to be working properly. In the product-specific test systems, this initial testing can be a very tedious process as shown in Fig. 7.

When the new product is measured for the first time and the measured values are not satisfactory or the test system stops because of a programmatic error, the software must be loaded into the computer, modified (debugged) and restored, and the test system reloaded. Then the product is measured again. Even though this process can be made faster using two computers, one for performing the actual measurement and the other for software debugging, initial product testing is



**Fig. 7.** Product-specific initial product testing.

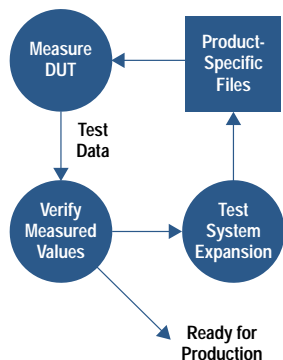
time-consuming and has to be performed by individuals with software expertise.

The initial product testing on a data-driven system is much easier, as shown in Fig. 8. After the product is added into a data-driven test system, the system administrator performs the initial product testing. If the measured values are not correct, any changes required are made to the external (product-specific) files through an editor immediately accessible from the keyboard. Once the system administrator is satisfied with the measurement, the product is measured a final time and the data is presented to the appropriate engineer for review.

This feature of a data-driven test system, which shifts the laborious work of measurement debugging from the test software to the product-specific files, is extremely beneficial. It not only allows nonengineers to perform the initial testing, but also, since proven test methods are being used, the overall initial testing process is very short.

### Adding Test Stations

All test systems whether old or new must be able to handle a variety of test stations. In the traditional product-specific test system, the system contains test-station-specific values imbedded in the software. With these station-specific values in the software, adding stations with extended capabilities or station duplication can be very time-consuming and a software expert is needed. This problem doesn't exist in a data-driven test system since the data-driven test system is station



**Fig. 8.** Data-driven initial product testing.

independent. These systems not only have the ability to handle numerous test stations with different measurement capabilities, but they also allow for test station duplication whenever production capacity is exceeded.

For extending a data-driven test system test station capability, the system administrator builds the new station hardware, assigns the station a new station number and/or name, and enters this information into the appropriate test system's station table.

Using this data-driven approach, SRSD has designed two complete test software systems as a pilot program. These data-driven test systems focus on measurement types within a product classification. These systems contain a set of well-designed tests that perform typical measurements required by a product classification.

### Mixed-Model Amplifier Test System

The mixed-model amplifier test system was the first system to validate the data-driven software theory. The system performs a set of basic tests that are required for complete amplifier characterization. The system has five different test stations: two network analyzer stations (40-GHz and 50-GHz frequency ranges) for complete s-parameter measurements and three scalar network analyzer stations (26.5-GHz, 50-GHz, and high-power 26.5-GHz frequency ranges).

This system began with two test stations with specific measurement capabilities. With the addition of more products, the need for stations with extended measurement capabilities arose. Since the data-driven system software is independent of test station equipment, we were able to add three additional stations with extended measurement capabilities without modifying any of the test system software.

This test system now provides complete characterization for over fifty separate products within the amplifier classification. Whenever necessary, the test system is expanded to incorporate new products by the system administrator. The total average product addition time is approximately three hours per product. The test process development time and expenses for these products have been greatly reduced.

For comparison, a product-dedicated test system used in microelectronic manufacturing was expanded for a new follow-on product. This system, which used some data-driven test software, still required one week of engineering time for complete system expansion because of the product-specific information imbedded throughout the test system software. Expanding this system for another product required three weeks of engineering/software time since the original tests could not be reused. This type of costly regeneration cycle will repeat itself again and again unless a data-driven test system replaces the current system.

The mixed-model amplifier test system design and set of tests required 487 days of software development time. Even though the initial test system design was very expensive, large savings occur each time a product is added to the test system. Test system expansion for a data-driven test system costs approximately 50 to 100 times less than the same expansion of a product-specific system.

The break-even point on the mixed-model amplifier test system was found to be four products, while the number of

products tested by the system is more than 50. The return on investment of the data-driven test system shows that this type of test system is necessary to stay competitive in today's marketplace.

### Vector Network Analyzer Test System

More recently, a data-driven test system was designed and implemented for a family of network analyzers. The complexity of instrument testing dictated that the data-driven test system be limited to a product family instead of a product classification. The system is currently testing five products. The new product introduction engineer estimated software generation time saved was approximately 500 engineering hours. More time would have been saved except that additional tests were required. These new tests were developed and added to the system in the data-driven format and are now part of the set of tests to be used by future family products.

In summary, by designing test software to handle large variations within a product classification, the overall manufacturing resources required for the electrical test development process are reduced over an entire product classification. This enormous savings occurs because the continual introduction or reworking of new product test software is eliminated.

### Acknowledgments

I would like to thank Claude Ashen for his support and guidance on structured analysis and system design, Steve Waite for his futuristic vision and at times blind faith that the test system would be completed in time to ship his products, and Wes Ponick for his enormous help in putting this project into words.

---

---

## Authors

August 1994

---

---

### 6 Scientific Graphing Calculator

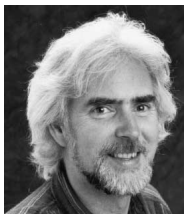
#### Diana K. Byrne



With HP since 1988, Diana Byrne is an R&D project manager for calculators at the Corvallis Division. She was born in Portland, Oregon and received a BS degree in mathematics from Portland State University in 1982, an MS degree in mathematics from the University of Oregon in 1987, and an MS degree in computer science from the same university in 1988. On her first HP project, she

developed software for plotting, graphics, and the EquationWriter for the HP 48SX calculator. She was R&D project manager for the HP 48G/GX calculator software and hardware. She is coauthor of a 1991 HP Journal article on HP 48SX interfaces and applications. Diana has two sons. Her favorite leisure activities are bicycling and reading.

#### Charles M. Patton



Software scientist Charlie Patton received a BA degree in mathematics and physics from Princeton University in 1972 and MA and PhD degrees in mathematics from the State University of New York in 1974 and 1977. He joined the HP Corvallis Division in 1982 as a software R&D engineer and has contributed to the development of the mathematics ROMs for the HP 75, HP 71B, HP 28C/S, and HP 48S/SX calculators. For the HP 48G/GX calculators, he worked on the RPL system and the 3D graphing routines, and was a general consultant and troubleshooter. He is currently working on several projects in the areas of software research, computer visualization, and symbolic and numeric techniques. He's also coauthoring a calculus textbook that incorporates technology as a teaching tool. He's named an inventor in three patents on operating systems and symbolic computation methods for handheld machines, and is the author or coauthor of technical papers on general relativity, representation theory, handheld computation, and calculus. He is a member of the American Mathematical Society, the American Association for the Advancement of Science, and the Internet Society. His outside interests include birdwatching, native plants, rafting and canoeing, fishing, gardening, Celtic harp, concertina, and calligraphy. A long-time HP telecommuter, he is also involved in remote sensing and internetworking.

#### David Arnett



David Arnett is a hardware design engineer at the Corvallis Division and has been with HP since 1991. He designed hardware and was the manufacturing liaison for the HP 48G/GX calculators. Currently, he designs hardware for the HP OmniBook product line. David was born in Cleveland, Ohio and attended Brigham Young University, from which he received a BSEE degree in 1989. He worked on avionics design at General Dynamics and on superconductivity research at Oregon State University before joining HP. He's a member of the IEEE. David is married, has two children, and enjoys music, both as a performer and as a conductor.

#### Ted W. Beers



Software R&D engineer Ted Beers came to HP's Corvallis Division in 1985 and has contributed to the development of the HP 48S and the HP 28C/S calculators. His work on the HP 48S includes the interactive stack, high-level display management, and user customization. He worked on user memory organization for the HP 28S and performed extensive testing on the HP 28C. For the HP 48G/GX calculators, he was responsible for the user interface elements. His work on a software technique for data and text entry has resulted in a patent, and he is the author or coauthor of three other technical articles, one written while he was in high school. Ted was born in West Lafayette, Indiana and received a BS degree in computer and electrical engineering from Purdue University in 1984. He's married and enjoys hiking with his wife and two beagles. He's also