HEWLETT·PACKARD JOURNAL

SEPTEMBER 1984

© Copr. 1949-1998 Hewlett-Packard Co.

HEWLETT·PACKARD
JOURNAL

September 1984 Volume 35 • Number 9

Contents

4 **Transmission Impairment Measuring Set Simplifies Testing of Complex Voice and Data Circuits,** *by David R. Novotny, Jeffrey Tomberlin, Charles P. Hill, James P. Quan, Gordon A. Jensen, and Jerry D. Morris* *Data communications users and carriers can both benefit from this instrument's state-of-the-art Bell-standard measurements.*

10 **TIMS Mechanical Design**

12 **Weight, Size, and Noise Impact Power Supply and Display Design**

13 **Master/Slave TIMS Operation Improves Productivity,** *by Teresa L. Reh* *One skilled craftsperson and two TIMS can do the job.*

14 **How Master/Slave Mode Works**

15 **Testing the TIMS,** *by Allan W. Dodge, Scott S. Neal, and Kurt R. Goldsmith* *Innovative approaches ensure correct performance and reliability of hardware and software.*

19 **Semiconductor Research Corporation: A Perspective on Cooperative Research,** *by Richard A. Lucic* *More than 30 U.S. companies are channeling semiconductor research funds through the SRC to U.S. universities.*

26 **A Hyphenation Algorithm for HPWord,** *by Paul R. Smit* *It's highly accurate and language-independent.*

31 **Designing Software for the International Market,** *by Heather Wilson and Michael J. Shaw* *Here are some of the basic requirements.*

35 **Authors**

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

© Hewlett-Packard Company 1984 Printed in U.S.A.

© Copr. 1949-1998 Hewlett-Packard Co.

In this Issue

Software localization is the process of translating software products into local languages. Within Hewlett-Packard, the responsibility for software localization rests partly with the HP Divisions that produce the software products, and partly with HP Application Centers in various countries. The Divisions have to design the products so that they can be localized easily, and the Application Centers do the actual localization. On page 26, Paul Smit of HP's Dutch Application Center describes a hyphenation algorithm he developed for the Dutch version of HPWord, a word processing package for the HP 3000 Computer. The algorithm has the property of being language-independent, and it will also be used for the Danish version of HPWord. On page 31, two software designers from HP's Office Productivity Division give us the Division's view of the localization problem. Their tutorial article deals primarily with office software products.

Because software localization seems to us to be important, timely, and of worldwide interest, we made it this month's cover subject. To illustrate it, we settled on the obvious approach of photographing the flags of many nations. Coincidentally, some of the soccer matches of the 1984 Summer Olympic Games were being played at nearby Stanford University, and art director Arvid Danielson asked for and received permission to photograph the flags flying there. The choice of flags wasn't ours, of course, and doesn't imply that HP software is available in all of the languages of the nations represented on the cover.

Shared research, or more properly, shared support of research, is getting a lot of attention in the U.S.A. these days, both from industry and from the news media. The idea that U.S. companies can join forces to support research at university laboratories *with* government encouragement and *without* incurring the wrath of government antitrust investigators is relatively new in this country. Last November, we published our first article on this subject. It described how several companies, including HP, are providing the funds to establish the Center for Integrated Systems at Stanford University. The article on page 19 of this issue describes another approach to cooperative research. More than thirty companies, HP among them, are channeling research funds to a number of universities through the Semiconductor Research Corporation of North Carolina. The SRC concept is already being emulated by other industries.

The exploding need for data communications has made it difficult for providers of data networks to keep up. As a consequence, we now find data flowing over every kind of circuit, from those originally designed to carry voice communications to specialized data circuits operating at many thousands of bits per second. Testing and troubleshooting this variety of circuits is the function of a new HP transmission impairment measuring set, the HP 4945A, which is described by its designers on pages 4 to 18 of this issue. Besides having the bandwidth needed to test high-speed data circuits, this new TIMS has many advanced features and offers a comprehensive set of measurements meeting the Bell standards used mainly in the U.S.A. and Canada.

-R. P. Dolan

What's Ahead

Next month's issue will cover the design and applications of a new family of printed circuit board test systems, the HP 3065. These systems provide high-yield testing of large, complex, digital, hybrid, and VLSI boards, with computer-controlled safeguards to prevent device damage (a problem with some testers).

© Copr. 1949-1998 Hewlett-Packard Co.

Transmission Impairment Measuring Set Simplifies Testing of Complex Voice and Data Circuits

This new TIMS' comprehensive measurement capabilities and powerful master/slave mode offer Bell-standard telephone companies and data communications users faster, more reliable testing and troubleshooting.

by David R. Novotny, Jeffrey Tomberlin, Charles P. Hill, James P. Quan, Gordon A. Jensen, and Jerry D. Morris

IN THE PAST TEN YEARS, data communications has moved from an activity used only by major corporations to a pervasive element of our way of life, affecting such ordinary things as bank deposits, airline reservations, and checking out a library book. In this same time period, the state of the art in data networks has advanced from dedicated circuits operating at 9600 bits per second to networks operating at 16,000 bits per second and even 56,000 bps. The demand for data services has far outstripped the capacity to build specialized data networks, so existing circuits, that is, telephone circuits originally installed for voice communications, are being used with only minor modifications.

In 1974, Hewlett-Packard introduced the HP 4940A Transmission Impairment Measuring Set to test voiceband circuits for use in high-speed data transmission.[1] This instrument was successful in analyzing voice-grade circuits for impairments, but now more is being demanded from the circuits and the people responsible for testing them.

Fig. 1 shows a typical data network. Before the AT&T divestiture, a data communications user in the U.S.A. could lease a complete communications system, including the modems, from the telephone company. Now, a user must go to separate vendors for modems, the local loop, and the long-haul facilities. This has created many interface points where testing must be performed and has increased the potential for one vendor to blame another when a problem occurs. The local loop, the wires running from the customer's location to the local telephone company office, is still the responsibility of the Regional Bell Operating Company (RBOC).

With divestiture, the long-haul segment may belong to one of a number of carriers such as AT&T Communications, GTE Sprint, MCI, and the like. These carriers must be able to certify to the customer that their segments meet the stated requirements. Data communications users also need equipment to resolve multivendor finger-pointing episodes. With this variety of measurement environments and user expertise, the test equipment must be easy to use, provide the measurements and bandwidth required for testing, be portable for field use, and be controllable for use in automatic test systems.

A New TIMS

The HP 4945A Transmission Impairment Measuring Set (TIMS), Fig. 2, is designed to meet the changing needs of the data communications test industry. The key contributions made by the HP 4945A are a comprehensive measurement set, the bandwidth required to test high-speed circuits, an improved master/slave capability for efficient use of skilled technical personnel, various interfaces for automatic control, and a CRT display for enhanced user friendliness.

The measurements and techniques required for testing

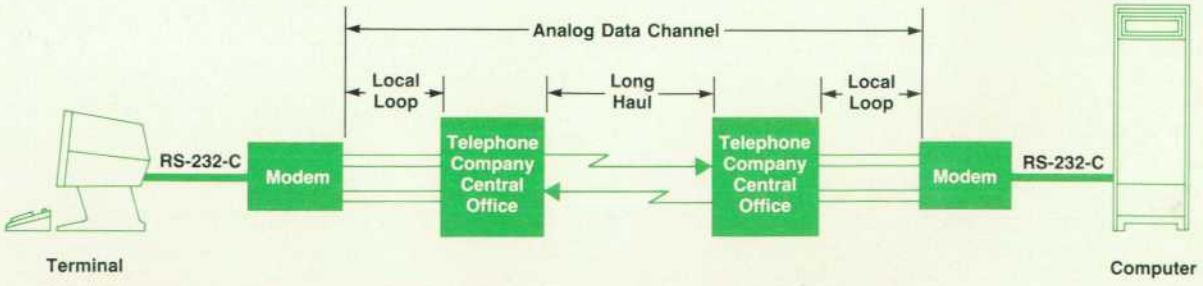


Fig. 1. *In a typical data network, different vendors are responsible for different segments, and there are many interface points where testing must be performed.*

© Copr. 1949-1998 Hewlett-Packard Co.

Fig. 2. *The HP 4945A Transmission Impairment Measuring Set tests voice and data telephone circuits according to the Bell standards used in the U.S.A. and Canada. A choice of three control interfaces is offered for integrating the instrument into an automatic system. One of these, HP-IL, makes the new TIMS compatible with battery-powered controllers and peripherals, such as the printer shown here.*

data circuits are specified by either the Bell System for the U.S.A. and Canada or by CCITT for the rest of the world. The HP 4945A adheres to Bell standards for measurement techniques. Parameters measured by the HP 4945A are:

- Loss
- Amplitude distortion (relative or absolute as a function of frequency)
- Gain slope (loss at 404 Hz and 2804 Hz relative to loss at 1004 Hz)
- Message circuit noise
- Noise with tone
- Signal-to-noise ratio
- Noise to ground
- Impulse noise exceeding three preselected levels
- Gain hits
- Phase hits
- Dropouts
- Envelope delay distortion (change of phase versus change of frequency)
- Intermodulation distortion (nonlinear distortion)
- Amplitude jitter in three bandwidths
- Phase jitter in three bandwidths
- Peak-to-average ratio (P/AR)
- Return loss (echo, singing low, singing high, and sine wave).

This measurement set provides the user with all the analog measurements needed to install, maintain, or troubleshoot any voice, voice-grade data, program (used by the broadcast industry), or wideband data circuit.

The frequency range of the transmitter and receiver of the HP 4945A is 20 Hz to 110 kHz. This allows testing of both voice and high-speed data circuits up to 72,000 bits per second. (Test standards say that loss measurements must be made at 1.5 times the bit rate being used).

In addition to the bandwidth, the HP 4945A provides noise-weighting filters required for making measurements over this frequency range, including C-message noise, 3-kHz flat, program, 15-kHz flat, and 50-kbit. The 50-kbit filter is used for making noise measurements on high-speed Digital Dataphone Service (DDS) circuits.

Master/Slave Mode

Data circuits must be measured on an end-to-end basis. Normally, this requires test equipment and a skilled technician at each end of the circuit. The technicians coordinate the testing activities on another telephone circuit. One person sends the appropriate test signal and the other receives the signal and records the result as measured by the test equipment. If the circuit being tested is local, dispatch and time coordination problems are usually small, but if a transcontinental circuit is being tested, time zone differences, work hours, and dispatch problems can cut the usable testing time down to just a few hours a day.

The master/slave technique used in the HP 4945A allows the unit at one end of the circuit to control and retrieve results from the unit at the other end of the circuit using only the lines under test for communication. This technique greatly increases the technician's productivity and reduces the chance for measurement error since all readings are made at one site. Master/slave mode is discussed in detail in the article on page 13.

Further increases in productivity can be obtained by using a calculator or computer to control an automatic test system. With the HP 4945A serving many customers with different needs, a variety of control interfaces are required. To meet these needs, the HP 4945A is designed to accept

© Copr. 1949-1998 Hewlett-Packard Co.

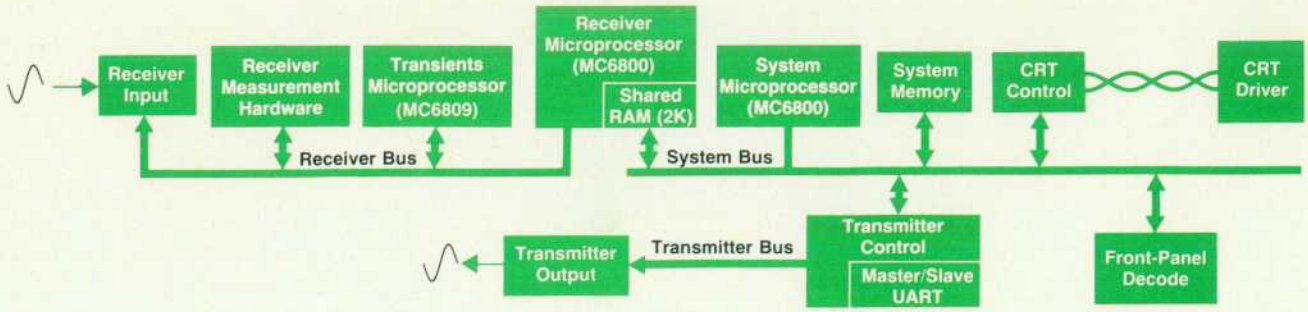


Fig. 3. *The HP 4945A TIMS consists of a transmitter, a receiver, and a system processor.*

three different control interfaces: HP-IB, HP-IL, and RS-232-C.

HP-IB (IEEE 488) is the choice for large test systems in which the controller is generally located in the same facility. RS-232-C, the interface of the communications industry, is used when the HP 4945A is not located near the controller and communication is via modems. This interface is also used when the controller does not have HP-IB capabilities. HP-IL (Hewlett-Packard Interface Loop) is provided for the field technician who needs to make automated measurements, but cannot afford the expense or bulk of an instrument controller using one of the other interfaces.

The HP 4945A uses a CRT display to enhance user friendliness. Previous HP instruments used segmented displays to present the measurement results to the user. With segmented displays, only a limited number of results can be displayed at any one time, forcing the user to select between various choices. Now, all the information pertaining to a particular measurement can be displayed simultaneously, thus eliminating user frustration and increasing productivity.

The display also enables the use of CRT-labeled softkeys. With softkeys, only choices appropriate to the measurement being made are presented. This greatly reduces the time needed for a person to learn the instrument and makes correct operation easy, even for the casual user.

Structure

Basically, the HP 4945A TIMS consists of a transmitter, a receiver, and a system processor (see block diagram, Fig. 3.) The transmitter generates the appropriate test signal for the selected measurement. The receiver measures the level and frequency of the test signal according to the selected test. The system processor keeps track of the measurement processes and handles the CRT and keyboard human interfaces. Each of these elements is discussed in detail in the following sections.

Transmitter

Like earlier HP TIMS designs, the HP 4945A transmitter uses a digital synthesizer to generate analog waveforms. These waveforms range from a simple sine wave to sums of sine waves or band-limited noise signals. Balancing, impedance matching, and floating of the signal lines is accomplished either by transformers or by an active output stage, depending upon the application.

The digital portion of the transmitter is a variation on the fractional-N synthesizer (Fig. 4). A binary number called the index, proportional to the output frequency of the transmitter, is latched and fed into a full adder. There it is added to an address generated by the previous cycle of the synthesizer. The sum is a new address, which is stored in an address accumulator. This accumulator feeds back into the adder for summing with the index in the next cycle.

Outputs from the address accumulator also drive the address lines of the waveform ROMs. These ROMs contain a scaled and shifted numerical representation of one period of a waveform. When the ROM outputs are applied to a digital-to-analog converter (DAC), the analog waveform is reproduced. Since the index controls the size of steps between points on the waveform, the transmitted frequency can be controlled by changing the index.

The majority of HP 4945A applications use the transmitter to produce a sine wave at a frequency between 20 Hz and 110 kHz. Stored in one of the waveform ROMs is a sine wave quantized to 1024 samples represented by 10-bit words. Clocking the fractional-N synthesizer at 524,288 Hz causes the system to cycle through the ROM addresses at a rate of $524288I/(1024 \times 2^W)$ Hz, where I is the index and W is the number of bits truncated when addressing the ROM. For example, if the adder output is a 15-bit number, then $W = 5$, since only the ten most-significant bits are used to address the ROM. In the HP 4945A, the adder output is a 19-bit number and $W = 9$, so the output frequency becomes simply I. Thus we guarantee frequency programmability with 1-Hz resolution. Furthermore, the sampling frequency of 524.288 kHz is sufficiently high to permit easy removal of quantizing noise by low-pass filtering.

Envelope delay distortion (EDD) is a measurement made

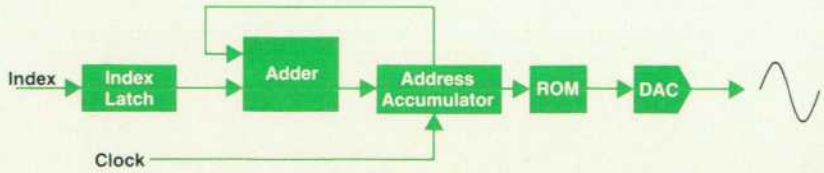


Fig. 4. *This basic fractional-N synthesizer is modified to generate waveforms transmitted by the HP 4945A.*

© Copr. 1949-1998 Hewlett-Packard Co.

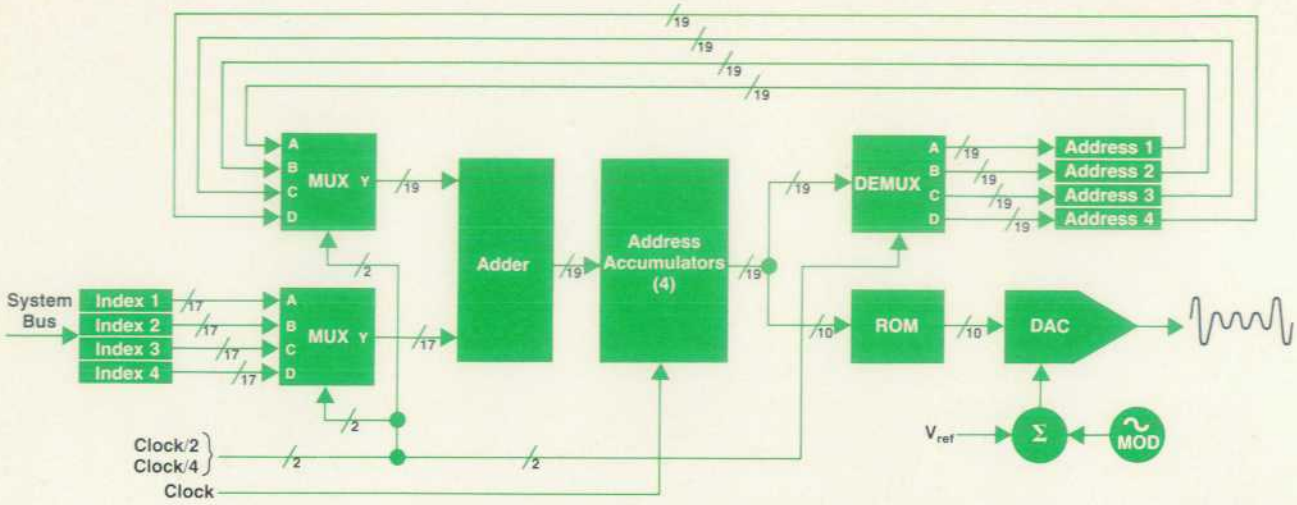


Fig. 5. *Modified fractional-N synthesizer used in the HP 4945A TIMS.*

with an amplitude modulated sinewave. For EDD measurements, the transmitter simply generates the modulation with a special oscillator and adds it to the reference input of the DAC.

For other tests, such as peak-to-average ratio (P/AR) and return loss, complex pseudonoise signals are needed. These were implemented by putting the frequency-domain specifications of the waveforms through an inverse Fourier transform algorithm and storing the time-domain equivalents in ROM. Lowering the sampling frequency and adjusting the index input to the synthesizer makes it possible to reproduce these waveforms just like the sine wave.

A third variety of applications requires the generation of signals that are the sum of two or four sine waves. These include four-tone intermodulation distortion (IMD) measurements, two-tone IMD noise correction, two-tone phase and amplitude jitter calibration, and frequency shift keying (FSK) for master/slave operation. Rather than take up ROM space with special waveforms for these cases, the fractional-N system was modified. Instead of a single index latch and a single address accumulator, there are four (see Fig. 5). On each cycle of the synthesizer, each of four indexes is added to its corresponding address accumulator in one of four sequential subcycles. When the outputs of all of the accumulators are multiplexed into the single sine ROM, a composite signal of up to four simultaneous sine waves can be obtained.

Analog Output Section

Once the signal has been produced by the DAC, it goes through an antialiasing filter to remove quantizing noise (see Fig. 6). For single sine waves, a third-order 175-kHz low-pass filter is adequate. Two-tone signals and the P/AR waveform tend to be more distorted because of their lower effective sampling rate, so for these, a third-order 5-kHz low-pass filter is used. When return loss noise signals are selected, an additional 2.5-kHz second-order section is cascaded with the 5-kHz filter to meet the tight specifications imposed by these measurements. Four-tone IMD, having the least noise tolerance of all, has its own sixth-order bandpass filter.

Output level control is accomplished by means of a multiplying DAC and a switchable coarse attenuator (Fig. 6). The filtered output signal connects to the reference input of a 10-bit multiplying DAC whose digital inputs are driven by the system processor. The output of the DAC is the

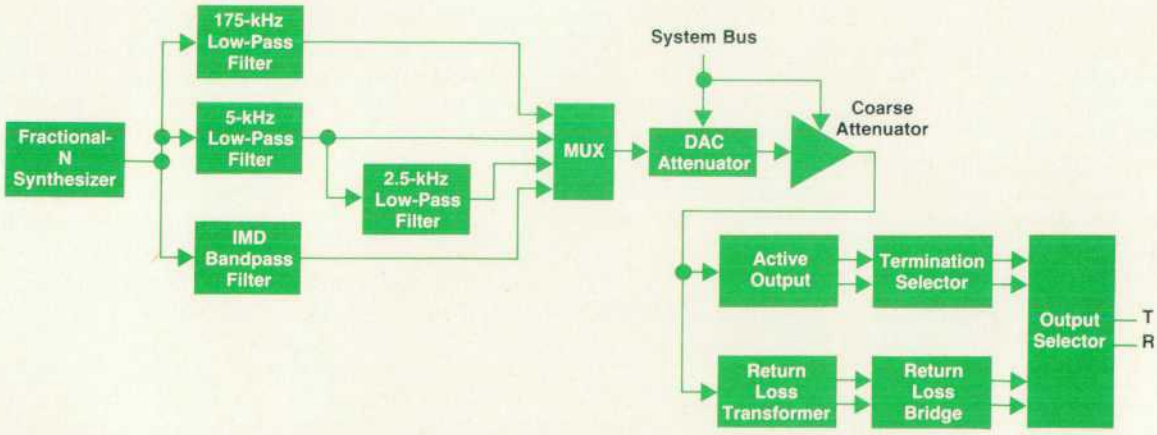


Fig. 6. *Analog output section of the HP 4945A transmitter.*

© Copr. 1949-1998 Hewlett-Packard Co.

product of the reference and the digital inputs, so the processor can multiply the signal level by M/1024, where M is an integer. This gives about 0.1-dB resolution up to about 10 dB of attenuation.

Since the output level adjustment range of the HP 4945A is over 70 dB, a coarse attenuator is used in conjunction with the DAC attenuator. Analog multiplexers select one of eight gain paths differing by 10 dB. Programming the DAC for 0 to 9.9 dB of attenuation and the coarse attenuator for 0 to 70 dB gives output control with 0.1-dB resolution over the entire range of the instrument.

Bell specifications require that test instrument transmitters be floating and balanced sources with close impedance matching to the line under test over the full frequency range. In earlier TIMS instruments this is accomplished by putting a single-ended signal through a specially designed transformer whose secondary matches the line impedance. Unfortunately, the HP 4945A's increased bandwidth (20 Hz to 110 kHz as opposed to 200 Hz to 4000 Hz in older TIMS) makes the design of such a transformer impractical. Our solution is an active output stage (Fig. 6).

The single-ended output of the coarse attenuator goes through a differential amplifier to produce an isolated and balanced signal. This signal drives a power output stage, which runs on specially derived floating power supplies. The result is a low-impedance, floating, balanced current source. Relays channel this source through resistor terminations to match the output impedance to the line.

The one exception to this is the two-wire return loss measurement. For this measurement, a bridge circuit measures power reflected back on the transmission line, and a transformer is substituted for the active output stage. It was found that excessive errors occurred because the bridge unbalanced the active stage. However, in this case, the bandwidth required is much less than in other measurements, so a transformer can be used, eliminating the error problem.

Receiver

The receiver section of the HP 4945A Transmission Impairment Measuring Set (Fig. 7) encompasses the major portion of the analog hardware of the instrument and occupies all or part of 10 of the 17 plug-in printed circuit boards. The measurement repertoire, which includes level, frequency, noise, signal-to-noise ratio, noise to ground, transients, intermodulation distortion, phase and amplitude jitter, peak-to-average ratio, envelope delay, and return loss, is one of the most comprehensive available today in one instrument. Despite this array of measurement hardware, this instrument almost halves the number of adjustments required in older TIMS receivers. Other features, such as an expanded frequency range of 20 Hz to 110 kHz and a

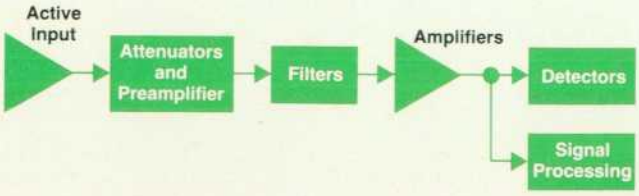


Fig. 7. *HP 4945A receiver block diagram.*

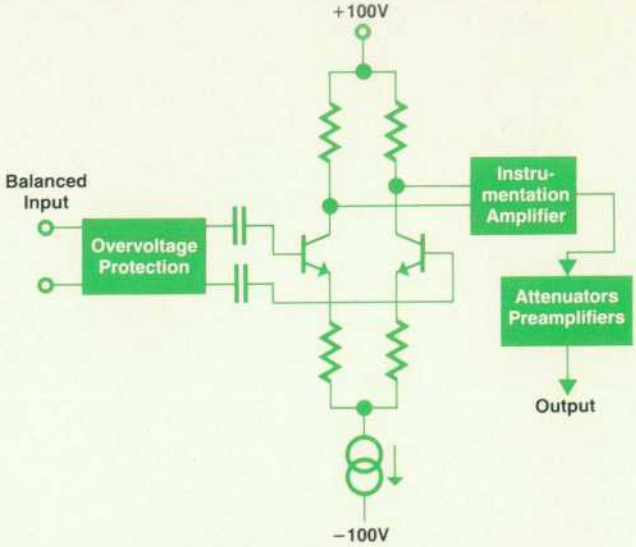


Fig. 8. *Instead of a transformer, the HP 4945A TIMS has an active input circuit that provides a wider bandwidth, a wider range of input impedances, smaller size, and lighter weight.*

dedicated microprocessor for flexible use of the hardware, make the HP 4945A a strong contribution to its field.

The design challenge in the HP 4945A receiver was to meet all of the specifications for all of the measurements in a single instrument. The combination of wide bandwidth, high dynamic range, low noise floor, low distortion, high common-mode voltage range, and high longitudinal balance* made the receiver design an exercise in creative compromise. The approach that was taken was to make the receiver a highly programmable device. This approach allows the receiver architecture to be configured to optimize performance for a given measurement. Take, for example, the tradeoff between noise floor and dynamic range. To reject interfering out-of-band noise, a selective filter needs to be placed near the front end of the receiver, but to get the lowest noise floor, gain needs to be placed near the front end. This classic receiver design tradeoff was dealt with in the HP 4945A by having a microprocessor control the gain and filters, switching them in or out depending on the measurement conditions.

Active Input

The required balanced input to a TIMS has classically been implemented with a transformer, which works well for the low bandwidths of older instruments (200 Hz to 4 kHz). In the HP 4945A, an active input circuit (Fig. 8) makes possible a wider range of input impedances, increased bandwidth, smaller size, and lighter weight. With the input exposed to high common-mode voltages and lightning that can be present on many telephone lines, the durability of the active devices in the input circuit was a major design challenge. The final design incorporates plasma discharge devices for lightning protection and a high-voltage differential input circuit of discrete devices that uses ±100V power supply rails. Noise, longitudinal balance, and other key input features do not suffer because

*A TIMS has high longitudinal balance at an input or output terminal pair if it has a high common-mode rejection ratio and equal impedances from the two terminals to ground.

© Copr. 1949-1998 Hewlett-Packard Co.

of the change from a passive to an active input.

Basic Level Measuring Function

The HP 4945A receiver needs to perform three kinds of ac measurements: average, rms, and peak. Fig. 9 shows the level-measuring circuit. The digital-to-analog converter (DAC), comparator, control logic, and up/down counter form a tracking analog-to-digital converter (ADC). For rms or average measurements, the input signal is detected by a square-law detector (rms detector in Fig. 9) or by a linear detector (absolute value circuit in Fig. 9). Many readings of the ADC output (counter contents) are taken by the microprocessor and averaged. For peak measurements, the absolute value circuit is used and the counter is allowed to count only up. Again, the microprocessor averages many measurements. The output of the analog-to-digital conversion process has some dither and the quantization error is averaged out.

A design goal was to minimize the number of manual adjustments. Accordingly, many of the receiver adjustments are done with software, that is, a known signal is applied, the result is measured, and the effect is quantified and stored in nonvolatile memory to be used later to correct measured values.

The level-measuring circuit is calibrated in this way. A known signal of great precision is applied to the average detector and the response is measured. In this case, the precision calibration signal is a triangle wave derived from the up/down counter, the digital-to-analog converter, and a precision voltage reference. The A-to-D measurement accuracy is determined by the accuracy of the voltage reference and the nonlinearity error of the DAC.

Intermodulation Distortion (IMD)

Another key to reducing the number of manual adjustments in the HP 4945A receiver is the implementation of the IMD measurement hardware. This circuitry measures the power levels of distortion signals in three bands centered at 520 Hz, 1900 Hz, and 2240 Hz, with bandwidths on the order of 90 Hz. Over each passband, the frequency response is very flat, yet large attenuation of out-of-band signals is required. An earlier approach to this problem uses three sixth-order bandpass filters of very high Q. This

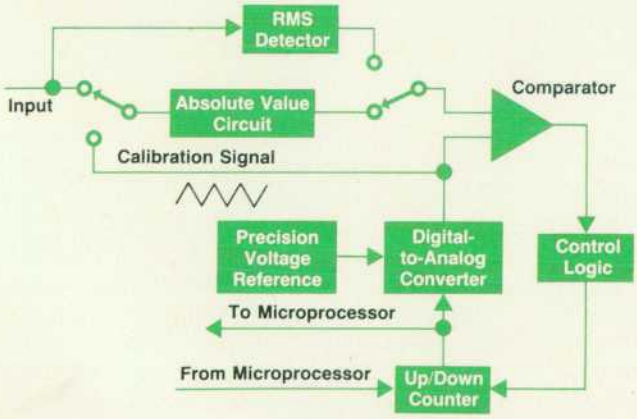


Fig. 9. *The level measuring circuit is self-calibrating. It is used for rms, average, and peak measurements.*

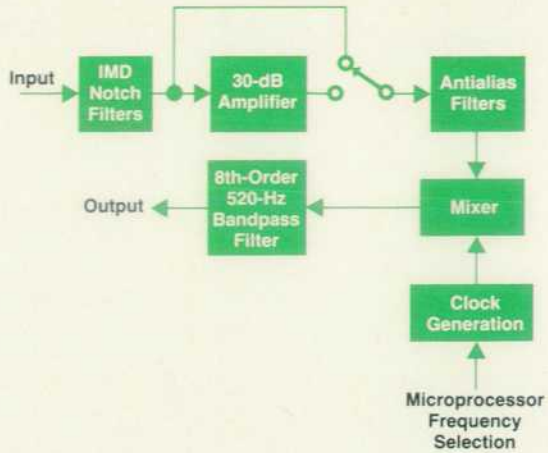


Fig. 10. *Manual adjustments of the intermodulation distortion hardware were reduced from 19 to four by this filter-mixer implementation.*

involves 19 adjustments in production for the alignment of the various filters.

To reduce the amount of hardware and thus the number of adjustments, the amount of printed circuit board space, and the material cost, a new scheme was devised (Fig. 10). One eighth-order bandpass filter centered at 520 Hz maximizes stability and minimizes the nonideal characteristics of the higher-frequency filters, and is preceded by a mixer, so the two higher-frequency distortion bands can use the same filter. Even with the addition of antialiasing filters, the number of components is reduced. The real benefit, however, is the elimination of 15 of the original 19 adjustments.

P/AR Measurement

Previous instruments that make peak-to-average ratio (P/AR) measurements use an analog circuit for measuring the peak value of the P/AR signal. However, this signal by its nature has a very high peak-to-average ratio, about 6.5:1, and the measurement is very sensitive to the accuracy of both the peak value and the average value measurements. The HP 4945A has a digital peak detector, which consists of a DAC, an up counter, and a comparator. The DAC in the HP 4945A circuit corresponds to the hold capacitor in conventional analog peak detectors. Peak detection is completely programmable in the HP 4945A, and its accuracy is limited mainly by the DAC, which is considerably more accurate than a hold capacitor implementation.

I/O Interfaces

The HP 4945A is designed to accommodate three kinds of input/output interfaces: the HP 18162A HP-IB Interface, the HP 18163A RS-232-C Interface, and the HP 18165A HP-IL Interface. The HP 4945A has two I/O ports in its rear panel, either or both of which may house an interface module. In a typical application, the interface module would be in port 1, since port 2 may also be used for the demodulated jitter output board.

The HP-IB interface is designed for the lab environment or a permanent rack-mounted application, in which the speed and flexibility of the HP-IB outweigh distance and cost considerations. The higher processing power of a

© Copr. 1949-1998 Hewlett-Packard Co.

TIMS Mechanical Design

The mechanical design of the HP 4945A involved several technologies new to Hewlett-Packard. These technologies make it possible to meet the weight requirements of the Bell specification for portable instruments with rack-mount capabilities.

Weight reduction is partly achieved through the use of a welded wire frame consisting of one-eighth-inch-diameter steel rod and 0.040-inch-thick C1010 sheet metal. All members are then strategically oriented so that few support brackets are required, thus keeping the weight down. The mounting of the printed circuit boards also contributes to the structural integrity, and the two 50-watt power supplies are mounted on 0.050-inch-thick sheet metal and formed in such a way that they provide additional support for the frame.

The front and rear frames are designed with a special plastic process, co-injection molding. Co-injection involves the injection of two different but compatible materials into the same mold. A solid material is injected first, covering the mold surface with a smooth solid skin. This is followed immediately by a second material containing a blowing agent, which forms a foamed interior in the frame. This process not only reduces weight, but also cuts costs by eliminating secondary painting operations.

The front-panel controls are operated through a membrane panel overlay. The overlay is attached with a peel-back adhesive, thus eliminating the need for any mechanical fasteners or switches.

Using these technologies, we were able to build a package that includes 23 printed circuit boards and a seven-inch CRT, and still maintain a instrument weight of thirty-three pounds.

Ernie Hastings
Product Designer
Colorado Telecommunications Division

mainframe or desktop HP-IB controller can be used to control a group of instruments and run a variety of tests automatically, storing the results on a disc and flagging out-of-specification measurements.

The RS-232-C module is designed for use at a remote site, with access provided through a set of modems. The RS-232-C module supports baud rates from 50 to 9600, a variety of word lengths and stop-bit configurations, both full and half duplex operation with and without modem handshaking, and a choice of Enq/Ack, DC1/DC3, or no software handshake. Echo can be programmed on and off by commands over the interface. The remote instrument can be dialed up by a computer for automated testing, or called by a craftsperson for interactive manual testing from a terminal.

The HP-IL module is intended for highly portable applications in which a handheld computer like the HP-41C or the HP-71B automatically runs a set of tests, prints the results, and stores them, or simply provides a go/no-go result.

A feature provided by all of the I/O modules is output mode. When a module is configured as an output device, a copy of the screen is output to a printer whenever the **OUTPUT** button on the front panel is pressed. This is accomplished in the HP-IB and HP-IL modules by entering talk-only mode, while the RS-232-C module just dumps the data, provided that the modem handshake and software handshakes are satisfied. An HP-IB or HP-IL printer must be in listen-always mode for the output to be accepted.

Interface Programming

The operating parameters of the interfaces are programmed from the HP 4945A front panel via the I/O setup menu. The various choices are displayed in the softkey label fields and are scrolled through allowed selections. The exception is the HP-IB address, which is entered directly from the keyboard. An I/O module cannot access its own setup menu, but it can access the setup menu of the other I/O port. The two ports may both be in the same mode, or one may be in normal mode and the other may be in talk-only mode. Two controllers may take turns controlling the instrument through the different ports, but it is recommended that they not attempt to execute commands at the same time, since this may cause one or both controllers to hang up.

Programming of the instrument is accomplished via mnemonic commands, all of which are three characters long and may or may not contain data. Rather than mimic the front panel in operation, each mnemonic programs the HP 4945A to a specific state. For example, VL3 programs the speaker to volume level 3, and FR01234 programs the transmitter to 1234 Hz. Either a semicolon or a line-feed character is accepted as a valid delimiter, and the HP-IL module accepts a colon delimiter to accommodate the HP-41C Handheld Computer. All of the modules accept lowercase commands.

I/O Operation

A set of data is requested from the HP 4945A with the command EXC. When the instrument receives the execute command from the module, it causes the receiver to make a measurement and return the data to the I/O module. The module appends a five-character header to each piece of data and sends it out to the controller. For example, receive frequency data has the header FRQCY. The last piece of data returned is a dummy, ENDST 0, which indicates that no more strings are coming. This allows the input routine to exit gracefully if the end-of-set data string is received before the desired string, for whatever reason. Should the controller attempt to enter data after receiving this string without sending the execute command, the bus will hang up, since no data is sent. When the HP 4945A is being used at a remote site with a terminal as the controlling device, the header string provides a quick identification of just what result each number represents, since most measurements result in more than one piece of data.

The heart of each of the I/O modules is a ROMless Z8 microprocessor and a standard system interface. The modules differ only in interface hardware and firmware. The modules decode the mnemonics, presenting the HP 4945A with a series of interrupts and keycodes, which the system handles in the same way as input from the front panel. To keep front-panel inputs from corrupting measurements, all interfaces incorporate optional lockout of the front panel. The modules also format the output strings, adding the header. All of the interface-dependent handshaking is

© Copr. 1949-1998 Hewlett-Packard Co.

taken care of by the I/O module processor, relieving the system processor of these tasks.

System Firmware

The HP 4945A system hardware is controlled by a 6800 microprocessor. The system firmware uses approximately 128K bytes of address space. To access this 128K-byte address space with the 6800, which can directly address only 64K bytes, a bank switching scheme is used. A 16K-byte address space is used for the banks. Bank 0 contains RAM for the CRT display. Banks 1 through 7 are set up for segments of ROM code. The other 48K bytes of the 6800's address space contains 36K bytes of ROM, 1K bytes of battery-backed RAM, 10K bytes of RAM, and 1K bytes of address decoding for hardware control.

The bank switching is made transparent to most of the firmware through the use of a segment management table and a segment entry point table for each bank. A special program was written for the firmware development system to aid in the generation and maintenance of these tables.

Each segment management table contains dummy entry points to the procedures in its code segment. When one of these dummy entry points is called, it calls the segment manager, passing the bank number and index into the segment entry point table for the called procedure. The segment management procedure then saves the current bank number and the return address of the calling procedure on a segment management stack. The return address of the calling procedure is then removed from the system stack, and the real entry point is called using the entry point table index passed by the dummy entry point.

On completion of the procedure called through the segment manager, control returns to the segment manager. The segment manager then switches to the bank of the calling procedure. Control is then given to the return address.

Receiver Interface

The HP 4945A receiver processor is controlled by the system processor through a section of shared RAM. The system and receiver processors also share a common clock, which allows them to address the shared memory on opposite phases of the clock. This makes accesses to the shared RAM possible without any arbitration hardware.

The shared RAM has two logical areas. One area contains a table of receiver control information and an access control variable. When the system needs to change the receiver function, the access control variable is set to deny receiver access. Then the receiver is interrupted to ensure that it will wait for permission to reenter the critical area.

The second logical area of the shared RAM is used to pass data from the receiver to the system. This is done with a queue data structure. The receiver processor puts data into the queue and updates the tail pointer. The system takes data out of the queue and updates the head pointer. Each pointer is one byte long and is updated only after an entire data element is added or used. This allows the two processors to process the data totally asynchronously.

Interrupt Service

All setup and control functions of the system occur as a result of interrupts. Both the maskable and the nonmaskable interrupt lines of the 6800 are used.

The nonmaskable interrupt (NMI) is used exclusively by a UART (universal asynchronous receiver/transmitter) which receives master/slave data. This data must be read as it is received to prevent data overrun errors. The data is put into a queue and a process to dequeue and process the data is scheduled for execution as soon as the system has finished executing all higher-priority processes.

The maskable interrupt can be generated from one of six sources: the front panel, a timer for scheduling processes, two I/O ports, the receiver, and the real-time clock. When an interrupt occurs, the hardware is polled to determine the source of the interrupt and the appropriate process is then executed to service the interrupt.

A background idling process looks for data in the receiver queue. When data is found it can be displayed in one or more places on the CRT and sent to one or both I/O ports. The destination for the data is determined by a data routing table that has been set up by an interrupt service procedure. After the data has been processed, it is removed from the receiver queue.

Error Handling Capabilities

Extensive error checking at both a user level and at the system level is incorporated throughout the system software. User errors are reported to the user with warning beeps and messages on the CRT. An attempt is made to make system-level error recovery transparent.

Having this extensive error checking in the code also proved very useful during the firmware development. The error handler was set up to display diagnostic information on the CRT when a system-level error occurred. This information included the type of error, the program address being executed, and a portion of the stack contents. After this, the error handler would wait for a front-panel key to be pressed to execute the recovery procedure. The diagnostic error messages and associated pauses are, of course, disabled in the production version of the firmware.

Self-Test and Calibration

The system software provides a number of service aids to verify that the system is functioning properly and to isolate hardware failures.

On power-up, a set of service switches on the processor board is read. If they are in one of the DSA (digital signature analysis) positions, a DSA routine is executed. A technician can then take signatures to isolate a digital hardware problem in the system.

If the service switches are in their normal positions, the processor performs a self-test on its hardware. This is followed by tests on the rest of the instrument including the receiver, the transmitter, and any I/O module connected to the instrument. If any malfunction is detected, an error code is displayed on the CRT and the power-on sequence is paused until the operator presses a front-panel button to continue.

After the instrument has gone through its power-up sequence, an extensive diagnostic self-test can be run to

© Copr. 1949-1998 Hewlett-Packard Co.

Weight, Size, and Noise Impact Power Supply and Display Design

Restrictions on weight and size and noise coupling to other circuits had considerable impact on many design decisions for the display circuit and power supply of the HP 4945A TIMS.

A CRT display with conventional magnetic deflection was chosen, and a design used in other HP products was adopted. An important criterion was that the display circuit have electronic positioning that is adjustable from the back side of the printed circuit board, the only side accessible in the HP 4945A's unusual mechanical package (see page 10). A certain amount of redesign was necessary to eliminate problems caused by the display's horizontal and vertical sweep frequencies of 20 kHz and 66 Hz coupling into the HP 4945A's transmitter and receiver. By using a six-layer board with the ground plane being the outermost layer, redesigning inductors with RM-8 ferrite cores, designing a new flyback transformer, and adding shields to the flyback transformer and deflection yoke, the coupling into other circuits was reduced to acceptable levels.

Two Power Supplies Improve Noise Immunity

The weight restriction had great influence on the power supply design. Power estimates were updated often throughout the design process so that the power supply would not be any larger (heavier) than necessary. Wherever possible, lower-powered components were designed in. The use of latching relays alone saved over five watts of dc power. Our initial estimates indicated that we needed about 70 watts of dc power.

Because of weight and size limitations, a switching power supply was the only type considered, and because of possible noise coupling problems, it was felt that the power supply switching frequency had to be outside the 110-kHz bandwidth of the instrument. After some investigation, we decided to use two 50-watt, HP 65000A Series Power Supplies.[1] In this design, the switching frequency is variable from 150 to 210 kHz. Thanks to the miniature magnetic components used at these high switching frequencies, there was room in our package for two supplies.

Some customization was done on the supplies to provide additional output voltages and integrate the supply modules into the HP 4945A package. The present maximum load on the supplies with two I/O modules installed is 75 watts. The total weight of the assemblies is just over 4.5 lb (2 kg). A conventional line-frequency, 75-watt output transformer alone would weigh that much.

The two-supply system gave us another advantage: noise immunity. Since the CRT display circuit pulls current at the sweep frequencies, we could have had 66-Hz and 20-kHz frequency components on all outputs generated from one supply. Instead, we dedicated one supply to provide power to the display circuit and five digital-only boards that are immune to the small sweep frequency components on the supply's outputs. These boards are the receiver processor, system processor, system memory, video generator, and keyboard. Any I/O modules and the dc fan, which draws current at approximately 120 Hz, also get their power from this supply.

The other supply provides the power to all the other assemblies, some of which are very sensitive to the display sweep frequencies. Considerable design still had to be done on some of the daughterboards to reduce power supply noise to acceptable levels.

Acknowledgments

Special thanks to all the people at HP's New Jersey Division for customizing their power supplies to fit our needs.

Reference

1. R. Myers and R. D. Peck, "200-kHz Power FET Technology in New Modular Power Supplies," *Hewlett-Packard Journal*, Vol. 32, no. 8, August 1981.

Kurt R. Goldsmith
ROLM Corporation
(formerly a development engineer at
HP's Colorado Telecommunications Division)

further verify the proper operation of the receiver or to isolate a hardware problem. A self-calibration procedure may also be run to ensure maximum measurement accuracy.

Acknowledgments

Many people played an important part in the success of the HP 4945A. The initial design efforts were led by Erhard Ketelsen and Don Dresch. Later, Dan Ackerhielm had project responsibility. Key design contributions were made by Larry Foreman (receiver), Kent Leung (receiver processor), Ron Morita (I/O), Tiki Perry (return loss software), Dave Kam (return loss hardware), Kurt Goldsmith (power supply and CRT drive), Mike Pozzi (jitter measurements), Chuck DeSostoa (IMD measurement and basic receiver measurements), Johann Heinzl (autorange hardware and filter design), and Dick Lee (transmitter). Special thanks to the product designers, who made the HP 4945A light, rugged, and producible: Peter Guckenheimer, Mark Gibble, Karrie Finkel, Ric Bechter, Fred DeVilliers, and Ernie Hastings.

Continued testing to electrical and environmental specifications was headed by Noel Damon, Al Dodge developed the automated test system used for all performance tests, Scott Neal developed the automated software QA programs, Scott Taylor headed the production engineering effort, and Don Marvel headed the service engineering task. Harold Wilkinson, Dan Ackerhielm, and Ken Hargrove helped hone the final product definition. Finally, special thanks to Bob Allen, whose constant inputs helped keep this project on track.

Reference

1. N.E. Damon, "Measuring Analog Parameters of Voiceband Data Channels," *Hewlett-Packard Journal*, Vol. 25, no. 12, August 1974.

© Copr. 1949-1998 Hewlett-Packard Co.

Master/Slave TIMS Operation Increases Productivity

by Teresa L. Reh

A TRANSMISSION IMPAIRMENT MEASURING SET (TIMS) measures most analog parameters of a channel in an end-to-end fashion. This means that at each end of the channel to be tested it is necessary to have a TIMS operated by a highly skilled craftsperson. Since these measurements need to be coordinated between the transmitter at one end and the receiver at the other end, the two craftspersons need to be in contact. This usually means that they need a voice channel for communication, in addition to the channel being tested.

The new HP 4945A TIMS has a master/slave mode that allows a single skilled operator to perform all necessary measurements from a central location. Two TIMS instruments are still necessary, but the one at the remote site only needs to be turned on and connected to the channel to be tested. An autocapture feature allows the master to capture the HP 4945A on the other end of the line without even putting the remote instrument into slave mode. This means that the person setting up the remote TIMS doesn't need to know how to operate the instrument at all. Nor does that person have to remain at the remote site. The additional channel previously needed for communication is no longer necessary.

Master/slave operation is made possible by a combination of hardware and firmware in the HP 4945A. The master and slave instruments communicate over the four-wire channel under test using internal modems. The master/slave firmware controls the level 2 (link layer) protocol and handles the data passed. It also passes control to other firmware modules. This firmware module is one of several modules that make up the interrupt service routines of the HP 4945A system firmware.

Master/slave operation is not a new concept. It is patented by HP and is available in earlier TIMS models, the HP 4943A and HP 4944A. The master/slave mode in the HP 4945A is an upgrade of the earlier version and is backward compatible with it.

The key philosophy behind the new master/slave mode is transparency. This means that from the user's point of view, performing a measurement in master/slave mode is essentially the same as performing the same measurement in normal (manual) mode.

The master can program all setup and measurement parameters in the slave that are necessary to perform meaningful measurements. These include the slave's transmit level and frequency settings. This programming is done by the same key sequences that one would follow in manual operation, with one exception. At some time in the sequence, the master/slave key must be activated and the direction of test chosen.

The implementation of master/slave mode in the HP 4945A provides major advantages over the previous version. With the HP 4945A, all necessary measurements and setups of the slave can be controlled through the master unit. Previously, the slave's transmit level could not be controlled, nor could the low impulse threshold setting for the impulse noise measurement or the slave's impedance and receiver bridge/terminate settings.

The major contribution of master/slave mode in the HP HP 4945A is flexibility. This flexibility is built into the communication protocol. When an HP 4945A master talks to a slave, it is given capability information about the slave, so that it knows what subset of its own capabilities the slave has. This enables the master to report error conditions and warn the operator of improper configurations or of the slave's lack of a given capability.

Master/Slave Communication

The communication protocol used in the HP 4945A for master/slave mode is specialized. It is a character-oriented protocol which in a way resembles IBM's bisynchronous protocol. Communication consists of message blocks exchanged between master and slave in a four-wire half-duplex fashion. There is also a secondary mode of communication that is compatible with the older master/slave version. In this mode, single bytes are exchanged between master and slave, again in half-duplex fashion.

The message blocks consist of four parts. The first character is a header byte that identifies the type of block (i.e., which one in a sequence). The second byte is called the byte count. It is equal to the number of bytes in the block following the byte count. The third part of the block is the data. As implied by the presence of the byte count, the data length (number of bytes) is variable. The block ends with a 16-bit (2-byte) block check character (BCC) used for error checking. The BCC is a cyclic redundancy code (CRC-16) which is computed for the entire block, including the header. This type of BCC provides detection of error bursts up to 16 bits in length and detects more than 99% of error bursts greater than 16 bits in length.

The secondary method of communication involves exchanging single bytes. Each byte contains its own byte-level error checking. The two least-significant bits are used for this purpose. A given byte in a communication sequence can be either a command or data. If it is a command, the two LSBs are 10. If it is data, they are 01. Hence, any byte containing a combination of LSBs other than the two stated will be regarded as erroneous and ignored. Likewise, if a byte with data format is received when a command is expected, or vice versa, it will be ignored.

Not only is the HP 4945A's level 2 communication protocol specialized, but so is its level 1 protocol. The instrument is equipped with an internal modem, which transmits and receives FSK signals. The FSK frequencies and the

© Copr. 1949-1998 Hewlett-Packard Co.

How Master/Slave Mode Works

Operation of an HP 4945A TIMS in master/slave mode consists of two phases—the linkup phase (Fig. 1) and the measurement phase (Fig. 2). Following proper setup of the two TIMS, the linkup begins when the master/slave softkey is activated. When the link between the master and the slave has been properly established, the measurement phase begins. It continues until a new measurement is begun or the link is terminated.

When two HP 4945As link up in master/slave mode they send message blocks alternately. The master begins by transmitting an inquiry/hello message, which tells the slave that it wants to begin a linkup process. This block also tells the slave that the master is an HP 4945A. The slave should then respond by sending to the master a block that tells what its capabilities and current setups are. The master will then know, for example, what noise filters the slave has, what its transmit level range is, etc. The current setup information tells the master what the slave's transmit level is and what its impedances are. The master responds by sending the slave a block that tells the slave what measurement to do and what its setups should be. For example, the setup information for a transients measurement includes threshold settings for impulse noise low, gain hits and phase hits thresholds, step size, and count rate. The last message block in the sequence is sent from slave to master. Here the slave tells the master whether or not it can perform the requested measurement; if it cannot, it tells whether or not it will loopback. (Loopback is a mode that a slave HP 4945A will go into if it cannot perform a requested measurement. The slave's transmitter repeats its incoming signal at the level at which its transmitter is set.) In this final block, the slave also includes ranges for its transmit level and frequency.

During linkup there is a mechanism for requesting retransmission of a message block. When the master sends the HELLO block (i.e., the first block), the slave will not respond until it has received a good HELLO block. After a given time, if the master has not heard from the slave, it will repeat the HELLO block. For the other blocks, any time an erroneous block is received, the instrument (master or slave) sends back a negative acknowledgment (NAK) byte, which tells the sender to retransmit the last block. If six NAKs occur in succession, the master will restart the entire linking process from the beginning.

The measurement phase doesn't involve handshaking like the linkup phase does. Once linked, the measurement begins. There are two different measurement directions, master-to-slave and slave-to-master. Measuring master-to-slave means that the two-wire circuit connected between the master's transmitter and the slave's receiver is the one being tested. Hence, the master transmits the proper test signal and the slave performs the measurement. The other channel is used to send the data to the master for display. The data is sent in message blocks like those used for the linking process messages. Measuring slave-to-master means that the circuit between the slave's transmitter and the master's receiver is the one being tested. The slave transmits the test signal while the master performs the measurement and displays the results. In this case, the other channel is used by the master to send programming information to the slave. This information tells the slave what level and frequency it should transmit. These message blocks are periodically repeated by the master since there cannot be a handshaking mechanism in the measurement phase to ensure that the message was properly received. Of course, the slave does not respond to the data if an error is detected in the block.

Now suppose a new measurement needs to be performed. To inform the slave of this event, a relink is done. A relink procedure is a subset of a full linkup. The master begins by sending the MEASUREMENT block (the third block of the previous linkup sequence). The slave replies with the RESPONSE block. In this case, it is not necessary to send the first and second blocks because the link has not been broken. The slave knows who its master is and the master knows what the slave's capabilities are. This abbreviated protocol speeds up the relink process.

Master/slave mode is terminated by unlinking. This is accomplished when the master sends five consecutive ABORT commands to the slave. Upon receiving two of these in succession, the slave stops what it was doing and returns to normal manual mode again.

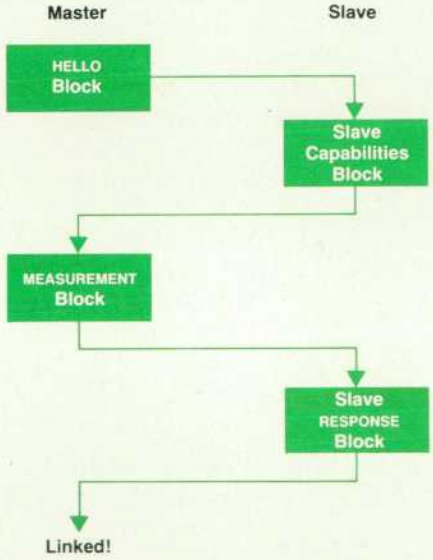


Fig. 1. *Master/slave exchanges in the linkup phase.*

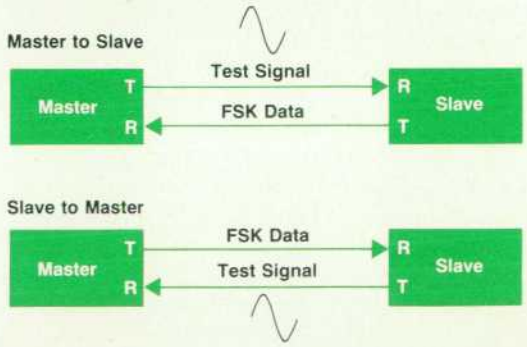


Fig. 2. *Master/slave communication in the measurement phase.*

© Copr. 1949-1998 Hewlett-Packard Co.

data rate are what makes it different. The mark (1200 Hz) and space (800 Hz) tones are modulated onto a fixed data carrier, 1990 Hz. The data rate was chosen to be 361 bits/second. These frequency choices were made to ensure compatibility with the previous master/slave modems in the HP 4943A and HP 4944A TIMS. The seemingly odd frequencies avoid harmonic interference problems in the envelope delay distortion measurement, which in the earlier TIMS is performed in the presence of FSK communication on the line under test.

Acknowledgments

Many people were involved in the development of master/slave mode in the HP 4945A. Jerry Morris was the system architect who wrote the majority of the system code—about 80K bytes—including some of the master/slave code. He was very helpful and contributed many ideas used in implementing master/slave mode. Jeff Tomberlin was the "jack-of-all-trades" on the project team. He worked on the transmitter, receiver, and system. He wrote all of the master/slave library and helped track down numerous sneaky bugs. Many thanks go to the other team members who worked on master/slave code at one time or another: Jim Quan, Gordon Jensen, Chuck Hill, Scott Neal, and Mike Pozzi. I would also like to acknowledge Chuck DeSostoa, who originally designed the modem hardware, and Jim Quan, who later assumed responsibility for it, adding enhancements to it.

Reference

1. J.E. McNamara, *Technical Aspects of Data Communication*, Second Edition, Digital Press, Bedford, Massachusetts, 1982.

Testing the TIMS

by Allan W. Dodge, Scott S. Neal, and Kurt R. Goldsmith

THE KEY OBJECTIVES in testing the HP 4945A TIMS were to measure its benchtop performance, to test its software, and to ensure high reliability to our customers. To measure the performance of the HP 4945A requires a very thorough HP-IB (IEEE 488) test system. This system completely tests the hardware and much of the software of the HP 4945A. Other stand-alone programs were written to test the software fully. In parallel with this effort was the development of a comprehensive environmental and reliability test plan for the HP 4945A.

Measuring Benchtop Performance

Before we explain how the performance is measured, it is necessary to discuss the sources of the specifications for the HP 4945A, because some are specific to this type of product.

The first source is the Bell System's Technical Reference PUB 41009, *Transmission Parameters Affecting Voiceband Data Transmission—Measuring Techniques*, May 1975. This document will be superseded by IEEE Standard P743, *Methods and Equipment for Measuring the Transmission Characteristics of Analog Voice Frequency Circuits*. These documents specify the desired performance for all the measurements made by the HP 4945A. The accuracy, range, response time, modulation, and other parameters are specified for each measurement. Other sections cover general types of performance, such as longitudinal balance and return loss of the transmitter and receiver.

The other major sources of the HP 4945A specifications are customers' inputs and the HP class B environmental and safety tests.

Test System

For single-frequency, sine wave transmitting and receiving, the frequency range of the HP 4945A is 20 Hz to 110 kHz at power levels of +13 to −60 dBm and impedances of 135, 600, 900, or 1200 ohms. The test system needs to be accurate to at least 0.1 dB over this range, but to 0.01 dB at 1 kHz and 0 to −20 dBm. To test the transmitter, the HP 3455A Voltmeter is used for the larger signal levels. For the smaller signals a special amplifier was designed to boost all the signals to greater than 100 millivolts.

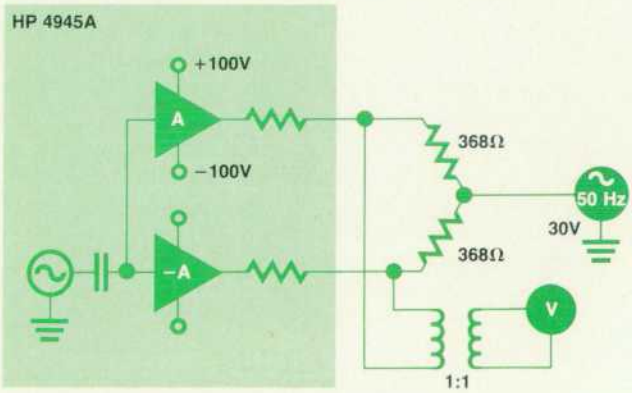


Fig. 1. *Longitudinal balance measurement with the transmitter off. Any difference in common-mode rejection or impedance to ground of the two sides will show up as a voltage at the transformer.*

© Copr. 1949-1998 Hewlett-Packard Co.

To test the receiver, two frequency synthesizers are required. The HP 3325A generates signals down to 20 Hz, while the HP 3336B, with its precision attenuator, generates signals as low as −60 dBm. These sources are also switched by a VHF switch to the HP 4945A under test and to the HP 3455A Voltmeter. The synthesizers are switched to the HP 3455A to calibrate them and to correct for any wiring losses, especially at 110 kHz.

The longitudinal balance of the transmitter and receiver needs to be greater than 90 dB at 50 Hz. This is not difficult to meet if transformers are used for signal coupling, but the HP 4945A is designed with active inputs and active outputs connected to a fixed ±100V supply. Thus the balance measurement is extremely important, since many components can cause the balance to become impaired. Fig. 1 shows the setup for measuring the balance of the transmitter when its output is off. To measure the balance, the source is set to 30 Vac at 50 Hz and is divided by the resistor network. Any difference in the impedance to ground or in the common-mode rejection of the two sides will show up as a voltage at the transformer. The resistor network is accurate to 0.001% and the 1:1 transformer has a balance greater than 120 dB. This setup works very well and is used to test each production unit.

Many other measurements in the receiver are tested by connecting the transmitter directly to the receiver and adding another tone that represents the noise or distortion for the measurement. Noise with tone, signal-to-noise ratio, and intermodulation distortion are measured in this manner with the HP 3336B Synthesizer supplying the noise tone. Amplitude jitter, phase jitter, and the transients measurements, however, require precise levels of amplitude or phase modulation. We are able to do this by using the HP 8116A Pulse Generator to modulate the HP 3325A Signal Generator. Some of the modulation frequencies are as low as 8 Hz while other modulation signals are 4-ms pulses. Setting of precise modulation levels is done by measuring the modulation sidebands with the HP 3585A Spectrum Analyzer.

Two Systems

There are actually two separate test systems, one to test the transmitter and to run all the general tests, and another system to run the receiver tests (see photo, Fig. 2). This doubles the final test throughput. Each set of tests has about 500 data points and takes about two hours to run. An HP 9000 Model 226 Desktop Computer is used to run the tests. An internal HP software package called Standardized Overhead Test Software manages the test programs. This package handles all the selecting of data points, printing of results, and storing of the data on flexible discs. Over 1800 data points are programmed into both systems, with 1000 of these included in the final tests.

Software Testing

To ensure quality in the HP 4945A software before it was committed to firmware, many separate efforts were initiated. An automated program was written to do an initial check after each version of the software was released from the lab. This testing was designed to catch major problems such as a missing feature or test function. The automated measurement test setup designed to test the hardware was also successful in finding the subtle software bugs. Many hours of actual use showed that changes in the human interface had to be made.

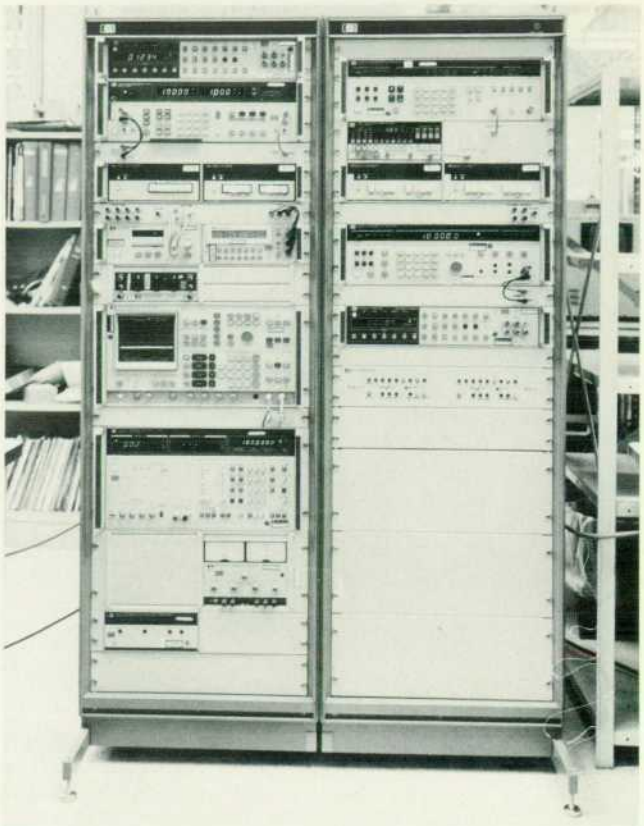


Fig. 2. *Systems for transmitter and general tests (left) and for receiver tests (right).*

Automated Program

Because of the many functions and features included in the HP 4945A, an automated software verification program was needed to reduce the time to verify any software changes. The HP 4945A has over 200 different functions. These functions have five different modes including master/slave and compatibility with the earlier HP 4943A and HP 4944A TIMS. To be able to test all the different functions in all the modes, the verification program was designed to use arrays containing proper responses for each of the different modes of operation. When a mode was selected, the program would fetch from disc memory the array containing the proper responses for all 200 functions in that mode. This method decreased memory requirements and the structured format was easy to modify and maintain.

The 200 functions are divided into two types. The proper operation of the first type can be verified by their I/O responses. The level frequency function is of this type and responds with a set of measurement values over the I/O channel. The second type of function cannot be verified over the I/O channel because a response is not part of the function. The action of selecting a filter for the noise measurement can only be verified by watching the selection process on the screen. The first type is easily verified by

© Copr. 1949-1998 Hewlett-Packard Co.

checking the value and syntax of each response. The array structure was used for this. The second type was verified with a program that would prompt the test engineer with the function that should have been executed and it was left to the engineer to verify that the proper function was carried out.

A special subprogram tested to see if random keypressing would be able to direct the processor to nonexecutable code (send the system "into the weeds"). This "monkey at the keyboard" program would send random keystrokes and check to see if the HP 4945A was still responding. The last 1000 keystrokes were always saved so that the offending sequence could be analyzed.

The special printer output feature was tested by programming the HP-IB interface of the computer running the tests (an HP 9000 Model 216) to be a listen-only device. The computer, with the help of a custom software driver, then looked like a printer to the HP 4945A.

The automated test program saved many hours of verification time by testing all of the HP 4945A functions and features in under four hours of test time. As software enhancements are released, this same test program will continue to save time.

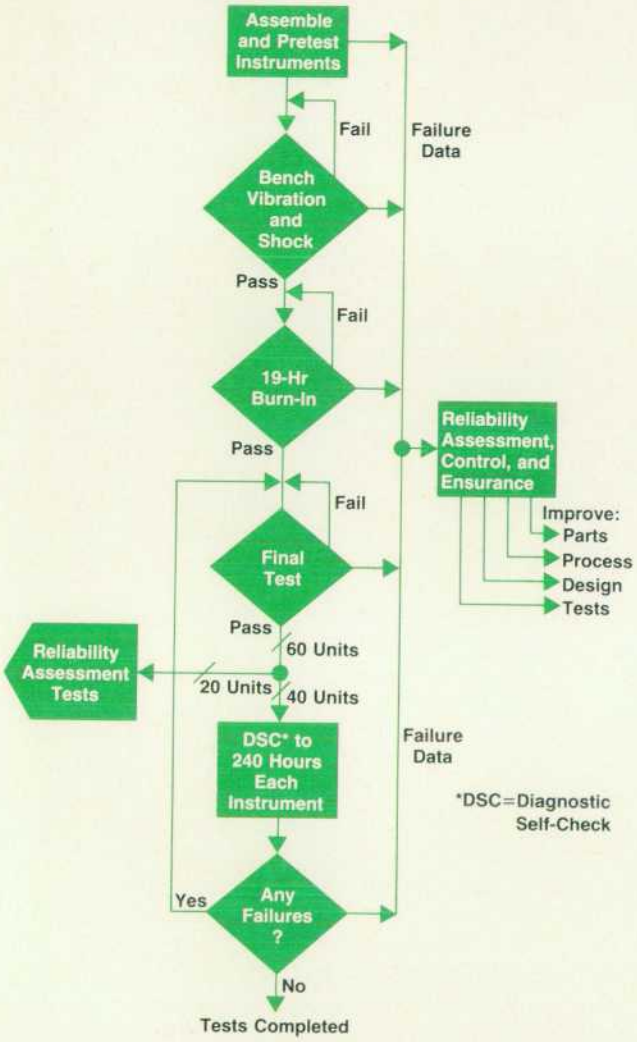


Fig. 3. *Production tests for 1983 shipments of the HP 4945A TIMS were designed to ensure reliability. Twenty of the first production instruments were put through additional reliability assessment tests (see Fig. 4). Failure data was analyzed to improve the design, the parts, the production process, and the production tests. Failure analysis continues for current production instruments, using warranty repair and field service data as well as production tests.*

Environmental and Reliability Test Program

In addition to the normal HP class B environmental testing, which is in no way trivial, early production HP 4945As went through an extensive reliability test program, acronymed RACE (Reliability Assessment, Control, and Ensurance). This RACE strategy had four key elements. First, every instrument produced went through a series of tests, accumulating at least 240 hours of test time before being shipped. Second, the first production run of twenty instruments was held from finished goods inventory and placed in a reliability assessment test program until 5000 instrument hours had been logged. Third, every failure observed, not only in the tests but throughout the production process, was recorded and examined for significance. Finally, the design and production teams responded to the results of the tests and corrected any weaknesses before any instruments were shipped to customers. Except for the reliability assessment tests, this program has been continued for current production instruments, using warranty repair and field service data as well as production test data.

Production Tests

Fig. 3 is a flow chart of the production tests. The instrument assembly and pretest block includes not only board-level tests, but also a burn-in of the power supplies and display assemblies. These are run at at 55°C for 48 hours to pack the CRT phosphor and catch any early failures. All of the daughterboards are then loaded into the cabinet assembly and calibrated, and then the instrument must pass its diagnostic self-check (DSC) on the technician's bench.

A benchtop vibration and shock test is then performed to catch any loose hardware or components. The 19-hour burn-in consists of ten cycles from 0 to 65°C with power being cycled as well. Research indicated that rapid temperature excursions of ten cycles or more would cause the majority of infant mortality failures to occur in electronic assemblies.[1]

After burn-in, the automated final test (performance verification) is run. The first production run of twenty units was then slated for further testing, while all other instruments run DSC until a total of 240 hours are accumulated on each instrument. There are two reasons for this. First, we feel that 240 hours is a minimum amount of on-time to provide a high degree of confidence in the reliability of an instrument of this complexity. Second, early in the HP 4945A's life, we wanted to make sure that the DSC had no hardware or software bugs in it that would result in additional "failures" and warranty costs. It was necessary to have as large a sample as possible to catch any possible bugs within the self-check.

Reliability Assessment Tests

Fig. 4 is a flow chart of the reliability assessment tests

© Copr. 1949-1998 Hewlett-Packard Co.

to which the first twenty instruments were subjected. In addition to the 240-hour minimum on each instrument, this program was required to accrue at least 5000 instrument-hours of test time. These tests were not designed to predict or measure the failure rate, but rather to stress the instrument to find weaknesses in our designs, components, and processes before any instruments were shipped to customers.

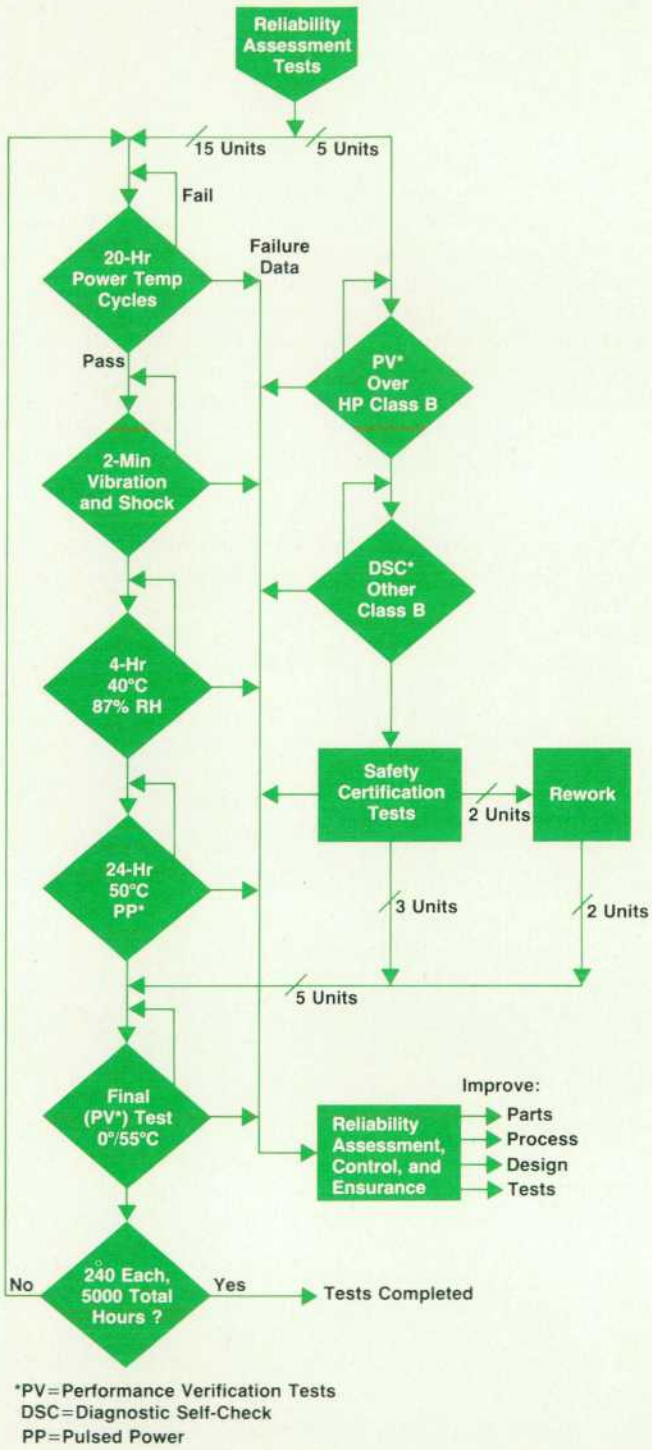


Fig. 4. *Reliability assessment tests were intended to stress the instruments to find weaknesses in the design, components, and processes.*

This program had two types of tests. Five of the first twenty units went through a traditional HP class B environmental test. However, the full final performance test was run at the temperature extremes of 0 and 55°C, as well as at 40° and 90% relative humidity. The diagnostic self-check was used as the performance criterion for all of the other class B testing. Final safety certification was also performed on a few of these production units.

The remaining 15 units took a different path in the test program. These units went through a series of temperature and power cycles just as in the production burn-in test, followed by the benchtop vibration and shock test to confirm mechanical integrity. After that, four hours of 40°C at 87% relative humidity was called for. The relative humidity of 87% was selected because it provides enough humidity to test the moisture resistance of the components and high-impedance nodes, but not so much moisture that condensation forms on the instruments during the test.

This high-humidity test was followed by a drying-out period of 24 hours at 50°C, with power occasionally being pulsed on and off to stress the semiconductor components.[2]

The full performance verification tests at the environmental extremes were so useful in finding weaknesses that all twenty units went through that series of tests, many units more than once. As a result, we logged nearly 13,000 hours of instrument test time.

The key to this reliability assessment program was not in performing the tests, but in doing something with the data taken during the tests, and reacting to the results of that data. All of the relevant data—which measurement failed, which component caused the failure, and the exact failure mechanism of the component—was logged and cross-referenced for tracking and correlation analysis.

All discrepancies were summarized and examined for correlation with measurement, environmental conditions, particular assembly, and component part number. This was done by engineers who were intimately familiar with the design, manufacture, and testing of the instrument. It was suprising how easily nonrandom failures were identified.

Acknowledgments

Special thanks to Paul Davis and Clark Nicholson of HP's Lake Stevens Instrument Division, who wrote the Standardized Overhead Test Software. Daryl Lockman and Alan Hansen did an outstanding job designing and configuring the final test stations for use by the test operators in production.

The entire design team was involved with the analysis of the failure data. Noel Damon and Bill Mayfield were instrumental in defining the reliability assessment tests, and Scott Taylor helped in the production test procedure. Ray Berry's expertise in performing the HP class B tests and in keeping the environmental chambers running at all times was invaluable. Greg Cummings and Dana Leavenworth were tenacious in logging all the instrument test hours.

References

1. W.J. Willoughby, Jr., *Navy Manufacturing Screening Program*, U.S.A. Dept. of the Navy, Naval Material Command, NAVMAT P-9492.
2. U.S.A. Dept. of Defense MIL-STD-781B, 15 November 1967.

© Copr. 1949-1998 Hewlett-Packard Co.

Semiconductor Research Corporation: A Perspective on Cooperative Research

Hewlett-Packard and other U.S. makers and users of semiconductor devices join forces to support universities in an innovative microelectronics research program.

by Richard A. Lucic

MUCH HAS BEEN PUBLISHED recently about industrial cooperative research programs. Hewlett-Packard is currently participating in several of these cooperative arrangements, including the Semiconductor Research Corporation.

The SRC's objective is to plan and implement an innovative university research program designed to help retain the vitality of U.S. semiconductor producers. This SRC cooperative research concept is already being emulated by other industries that recognize the need for cooperative research.

Why has there been a recent increase in the popularity of cooperative research? What is the impetus for these arrangements? The answer is threefold. First, there is the leverage obtained by combining scarce research funds to accomplish more in concert than any one company could achieve on its own. This is particularly true in the capital-intensive integrated circuit environment. A second factor is the elimination of unproductive duplication of effort. There exists a tremendous potential for increased efficiency if generic technology development can be focused in a few locations and the results shared with all participants. Finally, the Tax Powers Act of 1981 made it financially attractive for U.S. industry to participate in cooperative R&D efforts. In other countries, the integrated circuit industry has experienced a favorable cooperative climate for many years, but this favorable climate has only recently been developed in the United States. This accounts in part for the disadvantage perceived in this fiercely competitive market by U.S. semiconductor makers.

- Wide variety of manufacturing processes
- Need for microscopic control of cleanness as well as physical dimensions
- Computer-based control and information systems
- Sophisticated control and testing techniques to ensure quality materials for the manufacture of quality products
- Rapid advances in product design and manufacturing processes
- Trends toward automation in manufacturing
- Large capital investments
- Requirements for diverse materials including many that are hazardous
- Increasing public concern about environmental health and safety
- Vigorous international competition

Fig. 1. *VLSI manufacturing challenges facing the U.S. semiconductor industry today.*

The SRC was conceived in this environment of concern for the long-term viability of the U.S. semiconductor industry. It was recognized by some key industry leaders that the same old way of doing business was a prescription for failure. Their concept was that many of the current semiconductor research needs were of a generic nature and could be dealt with in a cooperative fashion, without compromising the proprietary contributions needed to keep the industry competitive. This concept evolved into the Semiconductor Research Corporation.

Funding Generic Research

There are two approaches to cooperative funding of the generic research needs of the industry. The first was articulated by Robert Noyce, Vice Chairman of Intel Corporation, in an address to the Industrial Research Institute in October 1981. He considered the basic scientific knowledge and

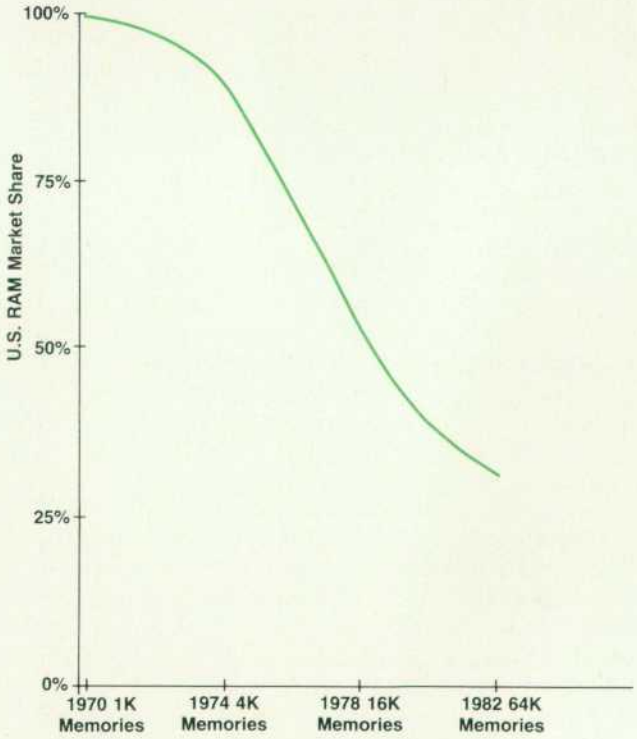


Fig. 2. *U.S. semiconductor manufacturers' market share has been slipping recently.*

© Copr. 1949-1998 Hewlett-Packard Co.

trained manpower of the semiconductor industry as common resources that are being depleted in the United States through exploitation by many users operating according to individual self-interest. The replenishment of these common resources was seen by Dr. Noyce as a collective responsibility of the industry, and although there is no way to assure that nonpayers would be nonbeneficiaries, the overall result is a positive-sum game, that is, the differential advantage to participants will be positive.

The second approach is voiced in a description of the SRC by Erich Bloch, the Chairman of the SRC's Board of Directors. He notes that U.S. universities have dual responsibilities for sustaining or increasing the technological momentum of industry: education and research. As the semiconductor industry and closely related industries such as computers and telecommunications have experienced a dramatic surge of technological progress, costs have increased rapidly for the universities that are educating future scientists and technologists and performing the relevant research on which the intermediate and long-term future depends. Many universities have found it difficult to keep pace with costs even with increased support by the government. The SRC is an initiative on the part of industry to increase support of the universities in their efforts to respond to these crucial industry needs. The principal theme of this initiative is cooperation by industry in the definition, guidance, and support of university-conducted relevant generic research. Benefits of the cooperative approach beyond those of individual company-university relationships include the following:

- Cooperation of semiconductor producers, users, and suppliers will provide a clearer pattern of technology direction and opportunities and will increase both the relevance and the exploitable yield of university efforts.
- Combined resources will enable the establishment of significant programs that are beyond the means of individual companies.
- The cooperative effort will spotlight the technology area and attract additional talent to the effort.
- Cooperation will promote efficiency by decreasing accidental redundancy in research efforts.

Both the industry and the universities are sure to benefit from the expanded scale of interactions and of research activities.

History and Status of the U.S. Semiconductor Industry

The SRC is a natural derivative of the development of semiconductor technology in the United States. The generic technology on which this industry is based was spawned by Bell Telephone Laboratories in the 15 years following the invention of the transistor, and was stimulated by a major infusion of funding from the U.S. defense establishment for research in both industrial and university laboratories. The growth of the semiconductor device industry was so rapid that by the mid-1960s, developments within the industry were outpacing the ancillary research community. Both Bell Laboratories and the U.S. Department of Defense refocused their efforts on the application of the fruits of their earlier research rather than the continued advancement of the generic technology. Support of academic research shifted in emphasis to longer-range, more speculative areas such as compound semiconductors.* Research directed to both the near-term and long-range needs of the mainstream semiconductor industry was largely confined to the laboratories of the companies producing the devices. Both merchant producers and the growing number of captive semiconductor producers were involved. The circumstances had negative impacts. First, the competitive companies were reluctant to share research results that might provide a competitive advantage. This slowed but did not stop the diffusion of new results. Second, the research efforts became highly redundant. Each company found it necessary to apply its efforts to very similar topics to avoid being left behind in a critical, rapidly developing technology. Gradually, research became focused on the solution of near-term problems, and the generic technology for the longer-range future was neglected. Third and most important, the structure of the industry and the economy resulted in an erosion of the resources available for research, and it was necessary for each company to apply these resources to current product development to remain competitive.

In recent years, the U.S. semiconductor industry has been faced with numerous challenges. Rapid technological advances have necessitated rapid turnover of capital investments whose replacement costs are escalating dramatically. Manufacturing processes are becoming more complex as well as more numerous. As the physical dimensions of semiconductor devices have shrunk, the need for microscopic control of cleanness as well as physical dimensions has become acute. Vigorous foreign competition based primarily on product quality has continued to erode market share. Also, at the same time that requirements for diverse materials, including many that are hazardous, have been on the rise in semiconductor processing, there has been increasing public concern about environmental health and

*Research in compound semiconductors is an important element of the generic technology for future semiconductor products. However, it represents just one of the areas that need to be addressed. It is just as possible that new generations of capabilities may be based on the evolution of silicon technology into radically new device structures as it is that new but less developed material systems, such as the III-V semiconductors, will provide these capabilities. Each candidate technology must be addressed in a comprehensive generic technology at a level commensurate with its perceived role.

Fig. 3. *Offices of the SRC in Research Triangle Park, North Carolina.*

© Copr. 1949-1998 Hewlett-Packard Co.

safety. Fig. 1 is a more complete list of these VLSI (very large-scale integration) manufacturing challenges now facing the semiconductor industry.

Many of these factors have acted in concert on the U.S. semiconductor industry at the same time that semiconductor users' expectations are increasing. It should be obvious that if customers' expectations are increasing and the competition can meet these expectations, anyone who does not respond to the challenges does not have a bright future. Indeed, the market share enjoyed by U.S. semiconductor manufacturers has been slipping in recent months (Fig. 2). The changes that are developing in the microelectronics industry and its technology base will have far-reaching effects on the participating professions and institutions.

Plight of Engineering Education

Engineering education has become a major concern for a country increasingly dependent on technological advancement. Colleges and universities are almost universally experiencing accelerating enrollment in engineering fields. This renewed interest in engineering careers is placing a tremendous burden on faculty and facilities. Many university laboratories contain hopelessly outdated equipment while escalating costs for new equipment keep modernization always out of reach. The industrial demand for qualified engineers is so great that salaries have been driven far above what most universities can afford. Some observers have concluded that this has caused a net loss of 400 university professors a year. In the field of semiconductor electronics, the problem has been intensified by dramatic reductions in government funding of silicon-related research. The result has been an increasingly limited supply of competent graduates prepared to work and contribute to the U.S. semiconductor industry. Both Japan and Russia are currently graduating many times more engineers per capita than the United States. It is obvious to many industry executives that this situation must be dramatically improved without delay since the IC technology turnover rate is now faster than the rate at which we can train new microelectronics engineers. If the education system could be corrected immediately, it would be approximately three years before industry would even begin to experience the results.

Microstructure Sciences

Materials, Phenomena, and Device Physics
Microsciences
Device Fabrication Technologies

Design Sciences

Design Automation
Chip Architectures

Manufacturing Sciences

Manufacturing Systems
Automation, Robotics
Process Modeling and Simulation
Packaging
Reliability and Durability

Fig. 4. *Major areas of interest of the SRC research program.*

SRC History and Organization

Given the current status of the U.S. semiconductor industry, there is a major role for the type of institution with which HP is now associated, the Semiconductor Research Cooperation. This role is centered on the evolution of a generic technology base that serves as a common foundation on which to build the competitive efforts of the future. This generic technology includes all of the material, device, design, and system skills required to build future complex systems on silicon.

The SRC was conceived in late 1981, when executives of major U.S. companies that produce and/or use semiconductor products recognized that the erosion of the generic technology base coupled with government-financed efforts in other countries constituted an important competitive threat to their industry. Recognizing that a government-based response to this threat was unlikely, they decided to undertake a cooperative industry-initiated response. After considerable discussion, a concept emerged that has become the SRC. According to this concept, the SRC will:

- Be sponsored by member companies through fees based on semiconductor sales or use, or semiconductor-related sales
- Provide knowledgeable and professional planning of an industrywide research program
- Contract for research to be coordinated by an industry/SRC technical advisory board
- Play an active role in monitoring research progress and transferring results back to industry
- Establish both:
  - University centers of excellence with long-term, major thrust areas
  - Additional research activities based on short-term, more precisely defined objectives
- Result in:
  - Establishment of a major, focused, generic research program supportive of the needs of the semiconductor industry
  - More relevant graduate education and a larger supply of graduate students
  - Improved communication within the U.S. semiconductor community.

Subsidiary research objectives that are consistent with this concept include the following:

- Provide a clearer view of directions, limits, opportunities, and problems in generic technologies
- Decrease fragmentation and redundancy in U.S. semiconductor research
- Establish above-threshold research efforts for critical areas that are beyond the resources of many cooperating companies
- Enhance the image of the industry to attract needed talent
- Increase industry-university ties
- Provide a scientific and technical information base for industry development efforts
- Attract highly qualified students to graduate programs in semiconductor research
- Help universities attract and retain competent faculty in semiconductor-related areas.

As part of this set of objectives, it was agreed that, to broaden the institutional base, the SRC program should

© Copr. 1949-1998 Hewlett-Packard Co.

encompass universities that are not presently leaders in semiconductor research, while increasing the level of effort in universities already active in this technology.

By early 1982, the SRC concept was sufficiently advanced so that, under the aegis of the Semiconductor Industries Association, the SRC was incorporated and bylaws were written. In May 1982, Larry W. Sumney, former director of the Department of Defense very high-speed integrated circuit program (VHSIC), was hired as Executive Director of the SRC. In September 1982, offices of the SRC were established in the Research Triangle Park of North Carolina (Fig. 3), initial staff was hired, and efforts to establish the research program were initiated. In November, eight universities were selected for award of the first research contracts and 166 proposals were received from 63 universities. By the end of 1983, over thirty-eight U.S. universities were participating in the SRC research program.

The SRC exists as a cooperative effort of companies that produce or use semiconductor devices or are major suppliers to the semiconductor industry. Present membership includes:

Advanced Micro Devices, Incorporated
American Telephone and Telegraph Company
Burroughs Corporation
Control Data Corporation
Digital Equipment Corporation
E.I. du Pont de Nemours & Company
Eaton Corporation
E-Systems, Incorporated
GCA Corporation
General Electric Company
General Instrument Corporation
Goodyear Aerospace Corporation
Harris Corporation
Hewlett-Packard Company
Honeywell, Incorporated
IBM Corporation
Intel Corporation
LSI Logic Corporation
Monolithic Memories, Incorporated
Monsanto Company
Motorola, Incorporated
National Semiconductor Corporation
Perkin-Elmer Corporation
RCA Corporation
Rockwell International
SEMI (Chapter)*
Silicon Systems, Incorporated
Sperry Corporation
Texas Instruments, Incorporated
Union Carbide Corporation
Varian Associates, Incorporated
Westinghouse Electric Corporation
Xerox Corporation
Zilog Corporation

The SRC is operated by a Board of Directors, which establishes major policy for the SRC, approves major contract awards, approves the appointment of Technical Advisory Board members and key members of the professional staff, and oversees the development and operations of the organization. Members of the Board of Directors are selected by the member companies, usually from their own upper levels of management.

The Technical Advisory Board has an active role in the evolution of the research strategy and the evaluation of proposals. It provides an important conduit for the flow of information from industry to SRC research contracts, and for the flow of research results from the SRC to industry. The Technical Advisory Board advises the SRC on what research should be funded. Technical Advisory Board members are appointed by the member companies. They are well qualified to give technical guidance and to represent their companies' positions.

The full-time SRC staff consists of both regular, permanent employees and industry personnel assigned to work with the SRC for defined tours of one year or more. This staff, under the direction of the Board of Directors and with the advice of the Technical Advisory Board, formulates and implements the research program of the SRC. It is intended that the permanent staff remain small so that the active participation of the Board of Directors, Technical Advisory Board, and industry assignees will continue to be necessary to carry out, manage, and direct the SRC program. In addition, it is planned that each researcher will have an individual identified as an industry resource con-

*Several small manufacturers who could not separately afford the minimum SRC entry fee have established a chapter under the aegis of the Semiconductor Equipment and Materials Institute. In this way, several small companies can participate in the SRC for one entry fee.

Microstructure Sciences

- Integratable high-speed logic elements with state discrimination capability in the 5-to-10-femtojoule range
- Compatible interconnection technology, including mixed technologies, low-impedance conductors, low-impedance contacts, wafer-scale integration
- High-density dynamic RAMs
- Logic chips with greater than 1-million-gate equivalence
- Accurate 16-bit analog-to-digital and digital-to-analog conversion
- Field-reconfigurable chip technology

Design Sciences

- Chip functional designs with 10× performance advantages over the existing state of the art
- Chip functional designs with reduced interconnection requirements
- System synthesis capabilities at the 10^8-logic-element, 10^{11}-bit-memory level
- Affordable generic testability methods with >95% fault coverage
- Reconfigurable and/or fault tolerant system design methodologies
- Hierarchical design systems that require <6 engineer months between system specification and error-free layouts

Manufacturing Sciences

- Defect control permitting production of chips with defect densities below $0.25/cm^2$
- Process automation permitting wide product mixes from the same fabrication line and a 5× improvement in productivity
- Reduction in fabrication line capital costs for a given production level
- Real-time correlation of process, device, and circuit models in the production environment
- Cost-effective package technologies that extend to 100W dissipation, high-speed interfaces, optical input/output, system-level packaging, 400 ports
- Product quality assurance at $1:10^6$ reject level and improvement in chip reliability of 2× without burn-in
- Materials and controls that eliminate yield degradation caused by material variables
- Metrology techniques and accuracies that support other manufacturing sciences goals.

Fig. 5. *Research goals in the major areas of interest.*

© Copr. 1949-1998 Hewlett-Packard Co.

tact (industrial mentor) to provide assistance and information when required.

SRC Research Program

Before the SRC research program was initiated, a strategy for the program was adopted that identifies the major areas of interest. The major areas are microstructure, design, and manufacturing sciences (see Fig. 4).

A three-tier layer of research activities is envisioned consisting of individual project awards, centers of excellence, and lead centers. The first level is intended to include those research activities centered around one or two faculty members and their students. Contracts will be awarded in a competitive process based on solicitation of proposals for innovative research. At the second level, centers of excellence are being established through negotiation and competition. These centers will consist of team efforts with themes that can correspond to major areas in the strategy or other mutually acceptable topics that are deemed to be of considerable importance. An effort is being made to supplement center themes with research vehicles that provide additional focus to the efforts. The third tier of planned research activity, the lead center, provides a framework for amalgamating related efforts and facilities of multiple institutions into a structure of coordinated research. The implementation of lead centers will evolve from the present project-center activities at some time in the future.

In addition to the three-tier structure of research de-

Microstructure Sciences

Centers and Programs	
Cornell	Microscience and Technology, Dr. Jeffrey Frey
RPI	Beam Technology, Dr. Andrew J. Steckl
MIT	3D Circuits and Systems Technology, Dr. Paul Penfield, Jr.
UC Santa Barbara, Stanford	III-V Memory, Dr. Jim Merz
Materials and Phenomena	
Illinois	Interactions During Vapor Phase Growth, Dr. J.E. Greene
Yale	Process-Induced Radiation Effects, Dr. Tso-ping Ma
Stanford	Origin of Interface States, Dr. Clayton W. Bates, Jr.
Yale	Thin Insulators, Dr. Richard D. Barker
Device Structures and Behavior	
Purdue	Heterostructure Devices, Dr. M.S. Lundstrom
Stanford	Complementary MESFETs, Dr. James D. Plummer
Carnegie-Mellon	Polysilicon in IC Processes, Dr. D.W. Greve
Arizona	Bipolar Transistors, Dr. B.M. Wilamowski
Vermont	Low-Temperature VLSI, Dr. Richard L. Anderson
Illinois	Reliability Physics, Dr. C.T. Sah
Florida	Polysilicon Emitters, Dr. Dorothea E. Burk
Interconnections and Contacts	
Minnesota	Low-Resistance Ohmic Contacts, Dr. Gary Y. Robinson
Mississippi State	Multilevel Interconnectors, Dr. Thomas E. Wade
Stanford	Multilevel Interconnections and Contacts, Dr. James D. Meindl
Arizona	Silicide CVD, Dr. James N. Fordemwalt
UCLA	MBE Silicides, Dr. K.L. Wang
Wisconsin	Silicide Metallizations, Dr. Max G. Lagally
Processes	
Notre Dame	Annealing in Silicon, Dr. Richard Kwor
Pennsylvania State	Thermal Nitridation, Dr. Richard E. Tressler and Dr. Joseph R. Monkowski
Minnesota	Low-Temperature Epitaxy, Dr. Ray M. Warner, Jr.
Pensylvania State	Plasma and Reactive Ion Etching, Dr. Joseph Stach
USC	Laser Mask Repair, Dr. Susan D. Allen
Johns Hopkins	Cluster Ion Beams, Dr. Kit H. Bower, Jr.

Design Sciences

Centers and Programs	
Carnegie-Mellon	Design Automation/CAD Center, Dr. S.W. Director
UC Berkeley	Design Center in CAD/ICs, Dr. Donald O. Pederson
Chip and Circuit Design	
North Carolina	Software Methodology, Dr. Frederick P. Brooks, Jr.
Iowa	Speed-Independent VLSI, Dr. S.M. Reddy
Texas A&M	Analog CAD, Dr. Phillip E. Allen
Brown	Silicon Compilation, Dr. John E. Savage
Arizona	MOS Simulations for CAD, Dr. Olgierd A. Palusinski
Testability, Verification, and Simulation	
Illinois	Design Verification and Testing, Dr. Timothy N. Trick
Illinois	Test Design, Dr. Jacob Abraham
Arizona State	Three-Dimensional Simulator, Dr. Lex A. Akers
Carnegie-Mellon	Testable VLSI, Dr. John Paul Shen
Chip Layout	
Columbia	VLSI Circuit Layout, Dr. Omar Wing
MCNC	Symbolic Layout, Dr. Jonathan B. Rosenberg
Rochester	CAD for Layout, Dr. Edwin Kinnen
Chip Architecture	
South Carolina	Signal Processors, Dr. Matthew Yuschik
Illinois	VLSI Arrays, Dr. F.P. Preparata, Dr. D. J. Brown

Manufacturing Sciences

Centers and Programs	
Stanford	Manufacturing Science and Technology, Dr. James D. Meindl
MCNC	IC Manufacturing Technology, Dr. Carlton Osburn
Michigan	Automation in Semiconductor Manufacture, Dr. Ken Wise
Clemson	VLSI Reliability Research, Dr. Jay W. Lathrop
Analytical Techniques	
Minnesota	Acoustical Microscopy, Dr. Rolf K. Mueller
North Carolina	Digital SEM, Dr. Roy H. Propst
Packaging	
Arizona	VLSI Packaging and Interconnection Research, Dr. John Prince
Stanford	Cooling Techniques, Dr. R. Fabian Pease
Georgia Tech	VLSI Bond Interfaces, Dr. B.R. Livesay
Cornell	Interface Defects in Ceramic Substrates, Dr. Rishi Raj
Yield Enhancement	
MIT	Si Defects and Internal Gettering, Dr. Harry C. Gatos

Fig. 6. *Current SRC-supported centers, programs, and projects.*

© Copr. 1949-1998 Hewlett-Packard Co.

scribed above, another program component referred to as a "new thrust" is provided. This is a mechanism for implementation of focused research efforts that respond to needs, opportunities, or technological threats beyond the outlined research strategy. Such efforts will be initiated either through competitive solicitations or from unsolicited proposals. Examples might include areas such as compound semiconductors, high-power ICs, novel architectures, electrooptical ICs, or precision processing.

The initial implementation of the SRC research program is based on a broad solicitation for research proposals that was distributed in the fall of 1982. As noted, the competitive solicitation resulted in 166 proposals that were evaluated and ranked in the three major categories of the research strategy. This process resulted in a decision to make awards of approximately 40 contracts to 26 universities. In addition, 37 proposals from two universities, Rensselaer Polytechnic Institute and Massachusetts Institute of Technology, and one university group, the Microelectronics Center of North Carolina, were removed from the competitive evaluation and instead are serving as a basis for the establishment of three well-focused research programs. In parallel with this broad solicitation, centers of excellence have been established at Cornell University in the area of microscience and technology and at the University of California at Berkeley and Carnegie-Mellon University in computer-aided design.

A set of goals was established to direct and focus the SRC research program (see Fig. 5). These goals enumerate capabilities desired by SRC sponsoring companies in the 1990 time period. The SRC recognizes that the goals appear to emphasize an orderly evolution of its present technology; however, creative and original alternative approaches to these goals will also be supported.

The individual goals are stated in ways that provide specific benchmarks. For a coherent body of technical capability, each goal must be viewed in the context of the others. The capabilities addressed by these goals extend to demonstrable producibility. However, commercial practice is expected to occur in industries that support the SRC, not at the universities. The task of the universities is to provide the research necessary to support the goals.

A current list of supported research includes the centers, programs, and projects shown in Fig. 6. One of these is highlighted in Fig. 7. Fig. 8 shows the current and planned size of the SRC program.

University-Industry Linkages

One promise of the SRC program is for improved contact and communication between industry and the academic research community. The SRC is not only aware of the potential for this enhanced communication, but also is considering mechanisms for transferring research results from universities to member companies. Programs have been implemented to accomplish these objectives:

- SRC Monitoring/Management. An SRC technical project manager is assigned to each of the research contracts. It is this manager's responsibility to provide the prime interface to the contractor for the technical direction of the research and to formulate the nature and level of continued SRC support. To carry out these duties, the technical project manager uses knowledge of the research and its progress, as well as inputs and advice from the Technical Advisory Board, the industrial mentor (see below), SRC members, and whatever other sources may be available, and must use the project evaluation strategy now being developed by the SRC.
- Industry Assignees. Two modes of participation for industry assignees in SRC activities are available. In one, the assignee joins the SRC staff for a period of at least one year and participates in the implementation and management of the SRC research program. This provides opportunities to obtain experience in program management, broad knowledge of semiconductor research activities, and a working introduction to the academic research community. It also constitutes a major contribution to the development of the SRC. The second industrial assignee mode, identified as the Researcher in Residence, is more focused technically. The assignee spends a defined period of time working in the university laboratory with one or more academic researchers. The research is strengthened by the assignee's experience and capabilities while the assignee gains intimate knowledge of the university research. This is probably the most effective method of technology transfer. The potential for industrial assignee participation exists with all SRC contractors, but in some of the more popular technology areas it may be necessary to rotate opportunities among SRC members. Both junior and senior technical staff may be considered for industrial assignee opportunities in SRC-supported university laboratories, and educational objectives or participation in teaching are desirable additions to the technology transfer function. Minimum stays of three months and normal tours of one year are suggested. The assignee has the responsibility for defining with the appropriate university person the specific mode of research participation and other activities.
- SRC Information Central. The dissemination of technical

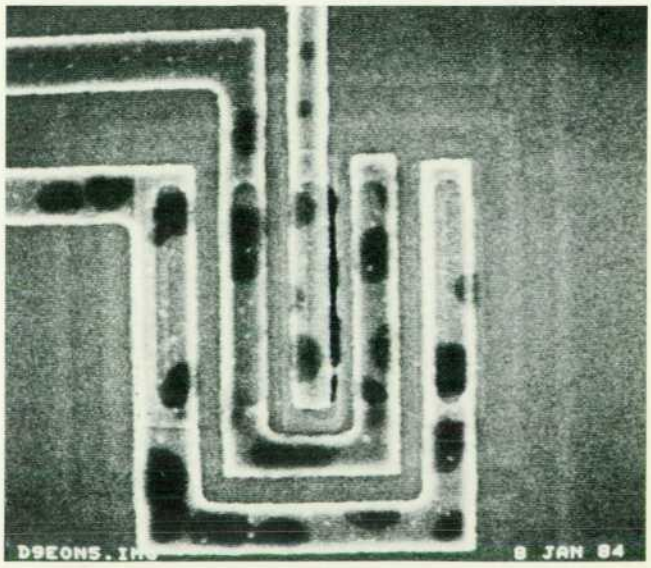


Fig. 7. *New techniques employed at the University of North Carolina for digitally enhancing scanning electron microscope images provide quantitative data from SEMs.*

© Copr. 1949-1998 Hewlett-Packard Co.

	Number of Universities	Annual Support Level	Faculty	Graduate Students
Beginning 1983	32	$ 8,800,000	125	150
End 1983	36	11,500,000	150	190
Planned (1984)	44	15,000,000	170	210
Potential	60	32,000,000	400	800

Fig. 8. *Growth of the SRC research program.*

information on SRC research projects will initially take place by distribution of report abstracts and of research reports. This distribution will be integrated with existing information systems of member companies where appropriate. It is planned that through CSNET (Computer Science Network), an electronic information distribution system, research program outputs in various forms will be rapidly distributed to designated individuals in the member companies by the SRC (Fig. 9).

- Topical SRC Research Meetings. The concept for these meetings is that specialists in a fairly narrow technical subject area, CAD layout tools or silicides for example, will meet to discuss research activities in that area. Industry attendees and SRC researchers will be invited to discuss their work and plans. It is expected that as many as twelve topical conferences per year will be convened.
- Industrial Mentors. Communication linkages between the universities and industry must be two-way. Industry technologists can help the university community identify important problem areas, may from time to time be able to provide direct technical assistance, and might steer this research away from nonproblem areas. For this supportive function, on the advice of the Technical Advisory Board, an industry engineer or scientist in an SRC member company has been identified as a direct contact point for each of the 53 contracts on our list. This individual is called an industrial mentor for the contract. Each Technical Advisory Board member provides nominations for this role. This linkage has received enthusiastic endorsement of the university researchers with whom it has been discussed and has the potential for materially increasing the effectiveness of the SRC research program.
- Program Reviews. The comprehensive activities of the SRC centers of excellence will be reviewed annually. Since these reviews will include broad areas of technical interest, they will be designed to attract industry representatives and Technical Advisory Board members with a broad range of interests and responsibilities and will be a key element in the evaluation of center activities.

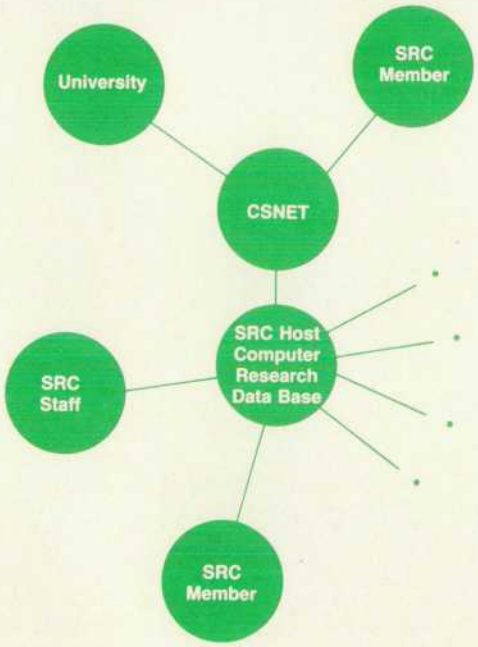


Fig. 9. *CSNET is an electronic information distribution system that will be used for rapid distribution of SRC research results to designated individuals in the member companies.*

Hewlett-Packard and the SRC

HP's involvement with the SRC goes back to the initial conceptual discussions. George Bodway, Manager of HP's Computer Integrated Circuits Division and a member of the Semiconductor Industries Association Board of Directors, was an early supporter of the SRC concept and was instrumental in HP's joining the SRC. HP Components Group Manager John Blokker was HP's representative on the SRC's Board of Directors until recently, when Corvallis Components Operation Manager Fred Schwettmann was appointed to that position. HP Laboratories' High-Speed Devices Director Dragan Ilic is a member of the Technical Advisory Board. The author is on temporary assignment from HP to the SRC, serving as Manager of Manufacturing Sciences Research Programs.

Three HP employees are currently participating in the industrial mentor program. Jerry Gladstone of the HP Santa Rosa Technology Center is mentoring the Illinois VPE/MBE project, Steve Shevenock of the HP Corvallis Components Operation is the mentor of the thermal nitridation program at Pennsylvania State, and Hugh Grinolds of HP Laboratories serves as the industrial mentor for the University of Minnesota acoustical microscopy research project.

Summary

The SRC with a 1984 budget of approximately $13 million is rapidly establishing a structured research program that responds to the generic technology needs of the semiconductor industry. Its initial research efforts have focused on addressing nearer-term problems. However, to achieve balance and maximize the long-term return on investment to its members, the SRC will address all areas of science and technology that will significantly contribute to the advancement of semiconductor technology both in this decade and on to the turn of the century. With the support of a growing number of industrial participants, this research will become a major factor in establishing the technology for the information age and will help assure the U.S. industry a position of continued leadership in the semiconductor field as well as in the myriad application areas that depend on advanced semiconductor devices.

Acknowledgments

I would like to acknowledge the contributions of Larry Sumney, the Executive Director of the SRC, and Dr. Robert Burger, SRC Assistant Director for Research, to this paper.

© Copr. 1949-1998 Hewlett-Packard Co.

A Hyphenation Algorithm for HPWord

Originally developed for the Dutch version of HPWord, this pattern recognition algorithm can be adapted to hyphenate words in many different languages.

by Paul R. Smit

THE HYPHENATION PROGRAM described here has been developed as a subprogram of the HPWord word processing software for HP 3000 Computer Systems. HPWord has multilingual capabilities, that is, documents can be produced in different languages. This affects some layout features as well as the hyphenation of words, which is done automatically.

In developing the hyphenation algorithm there were some demands the final program had to meet, not only as regards quality (an error percentage of 1% seemed reasonable to aim for), but also with respect to the efficiency and flexibility of the program. To achieve this, a somewhat different method is used compared to that of most existing hyphenation programs. Although roughly the same linguistic rules are used, the way to store and approach these rules is different.

The main advantage of the algorithm described here is its flexibility; the linguistic rules can be changed without changing the program. This makes it possible to use the program for other languages as well. The same program is being implemented in the Danish version of HPWord. It can also be implemented easily in other text processing programs. It is, for instance, easy to convert the algorithm to other programming languages such as Pascal.

Moreover, the problem of efficiency and of hierarchy in the linguistic rules has been completely solved by using a tree structure and an extra data segment to store all linguistic rules. This way searching takes place in main memory instead of on disc.

Straightforward Methods

In discussing hyphenation programs, we consider at first stand-alone programs which isolate *all* syllables in a word. As we will see later, this is not really necessary in a text processing program like HPWord. If a line is filled with the first syllable(s) of a word, only *one* breakpoint needs to be found.

In general, hyphenation algorithms work in the following way.

1. The first step is to search for the word in an exception dictionary. If the word is found, the algorithm stops.
2. If not, the various prefixes and suffixes are removed.
3. Next, some rules are applied to the rest of the word. Let C be any (combined) consonant and V any (combined) vowel. A combined letter is a combination of letters that is considered as one letter (e.g., qu or sh in the English language). In most languages there are some combinations of vowels that can be considered as one letter (e.g., aa, oei in the Dutch language). For the Dutch language, the following rules are often used.
   - The sequence VV is hyphenated as V-V.
   - The sequence VCV gives V-CV.
   - VC...CV is handled differently. As many consonants as possible will be put into the last syllable. This depends on whether the last two or three consonants are pronounceable. In the Dutch language, for instance, the letter combinations rz and tt are not pronounceable within one syllable, as in the word voor-zitter. The combination gr, however, can be pronounced, as in the word in-te-gra-tie.

A frequently used algorithm for the Dutch language that works along these lines is described by Brandt Corstius.[1]

Although the quality obtained by these straightforward programs can be very high, sometimes thanks to a large exception dictionary, as is normally the case in hyphenation algorithms for the English language, there are some drawbacks. Because the handling of different kinds of linguistic rules is hard-coded in the program, it is difficult to make changes, and in general, these programs are not very efficient because of the linear search mechanisms used.

Method Used in HPWord

Another way to approach the problem is to keep the algorithm and the linguistic (hyphenation) rules strictly separated from each other. The function of the algorithm is twofold. On the one hand, it has to build an environment in which the development of a hyphenation algorithm for any language is reduced to the formulation of linguistic rules in a suitable notation (as is also mentioned by Raiha,[2] and by Boot and Koppelaar[3]). On the other hand, the algorithm has to provide an efficient search mechanism to determine which rules must be used to hyphenate a given word.

The separation between the algorithm and the linguistic rules, together with the building of an environment that

Symbols Used in this Article

$s \in S$	s is an element of the set S
$\forall$	Read "for all"
$S \subset T$	Set S is included in the set T
$S \cup T$	Union of sets S and T
$\bigcup_{i=1}^{m} S_i$	Union of the sets S_i
$S \cap T$	Intersection of sets S and T
$\varnothing$	Null set, empty set

© Copr. 1949-1998 Hewlett-Packard Co.

we can use to edit the rules (the first job of the algorithm), makes the program flexible. Rules can be easily added or changed, and it is even possible to make a hyphenation program for another language by replacing the rules.

Structure of the Linguistic Rules

Before discussing the problems involved in making the algorithm efficient, we first give an explanation of the structure of the linguistic rules used by the program.

Consider a word of n letters:

$$l_1, l_2, \ldots, l_n$$

Between every two letters and at the start and the end of the word a sign s_i is placed:

$$s_0, l_1, s_1, \ldots, l_n, s_n$$

The signs s_i can have the following values:

0	Not decided yet.
1	Breakpoint in this position; actually indicates the beginning or the end of a syllable or word.
2	No breakpoint in this position.

Initially, the signs s_i have the following values:

$$s_0 = s_n = 1$$

$$s_i = 0 \ (0 < i < n).$$

The program analyzes every subset of this word:

$$s_{i-1}, l_i, s_i, \ldots, l_{i+k}, s_{i+k} \ (i > 0, i+k \leq n) \qquad (1)$$

Subsets of this form are called *patterns*. In analyzing patterns like this, the program tries to change the values of the s_i. It uses linguistic rules to do this. The format of such a linguistic rule is as follows:

$$s_0, l_1, s_1, \ldots, l_j, s_j \rightarrow s'_0, s'_1, \ldots, s'_j \qquad (2)$$

If the left side of the linguistic rule matches a pattern in the word, the values of the s_i are changed according to the values s'_i on the right side of the arrow. Matching occurs when the values of the s_i and l_i in (1) are equal to the corresponding values of the s_i and l_i in (2).

Generalized Rules

If we use the linguistic rules as they are described above, we have the following problem. We can regard the linguistic rules as a function:

$$F_k: \mathbf{L}^{k+1} \times \mathbf{S}^{k+2} \rightarrow \mathbf{S}^{k+2} \qquad (k=0,1,2, \ldots)$$

There is a function for each fixed pattern length. **L** is the set of all letters, $k+1$ is the pattern length (the number of letters in each rule), and **S** is the set of the signs. The number of elements in **S** is $|\mathbf{S}| = 3$. If we add 14 combined letters to the 26 letters in the normal alphabet, this makes $|\mathbf{L}| = 40$.

The total number of possible arguments in this function for fixed k equals $40^{k+1} \times 3^{k+2}$. This means, for instance, that we would have to store 5×10^6 rules if we considered only rules for patterns with three letters! To store the function (or rules) explicitly is clearly not the right method. It is not necessary either, because the arguments of the function can be generalized.

Consider the linguistic rule that there will always be a word break between (combined) vowels. To solve this problem we would like to have one rule for all patterns of the form:

$$s_{i-1}, l_i, s_i, l_{i+1}, s_{i+1} \ (l_i, l_{i+1} \in \mathbf{V}; s_{i-1}, s_i, s_{i+1} \in \mathbf{S})$$

where **V** is the set of all vowels and **S** is the set of all values of the signs. Such a generalized rule could have the format:

$$\{(s_0, l_1, s_1, l_2, s_2) \rightarrow s_0, 1, s_2 \mid s_0, s_1, s_2 \in \mathbf{S}; l_1, l_2 \in \mathbf{V}\} \ (3)$$

We need yet another notation to handle such generalized patterns and rules. Generalized signs and letters will be represented as Greek letters. Now the format of a generalized linguistic rule is as follows:

$$\sigma_0, \lambda_1, \sigma_1, \ldots, \lambda_k, \sigma_k \rightarrow \sigma'_0, \sigma'_1, \ldots, \sigma'_k$$

λ_i can represent a specific letter (written as a lowercase character), but can also have the value C (which means any consonant), the value V (any vowel), or the value * (any letter). σ_i can also have the value * besides the aforementioned values 0,1,2. On the left side of the arrow, the * means "any value"; on the right side it means "no change".

Now we can write (3) as follows:

$$* \ V \ * \ V \ * \rightarrow * \ 1 \ *$$

For example, one of the hyphenation rules in the Dutch language is the following: If there are two consonants between vowels, hyphenation will take place between the two consonants. (This is a very general rule with many exceptions, to be discussed later.) The rule is written as follows:

$$* \ V \ 0 \ C \ 0 \ C \ 0 \ V \ * \rightarrow * \ 2 \ 1 \ 2 \ *$$

On the left side of the arrow, the sequence V C C V indicates the two consonants between the vowels. The values * on each side of the pattern indicate that the actual values of the signs in those positions of the word are not important. The values 0 mean that no decisions for these positions have been made yet.

On the right side of the arrow the changes in the signs are given. The 1 in the third position forces a breakpoint between the consonants. Because there are no syllables with only consonants, the other values 0 will be changed into 2 (breakpoint inhibited). The values * mean that there will be no change in these positions.

Making the Algorithm Efficient

To make the search through the linguistic rules efficient, we would like to order these rules in some way. If we managed to do this we could use a binary search method instead of simply going through a list. One of the things we also have to keep in mind is how to incorporate a certain hierarchy in the linguistic rules into the data structure we are going to use, that is, which rule will take precedence in the case of conflicting rules. (We must be able to handle exceptions to exceptions, etc.)

The way we stated the generalized linguistic rule in (3) suggests a solution to both of these problems. We can treat the generalized rules as sets. The elements of these sets are the original (not generalized) rules defined in (2). As will be shown in the examples that follow, we can identify the hierarchy mentioned above with the inclusion relation of

© Copr. 1949-1998 Hewlett-Packard Co.

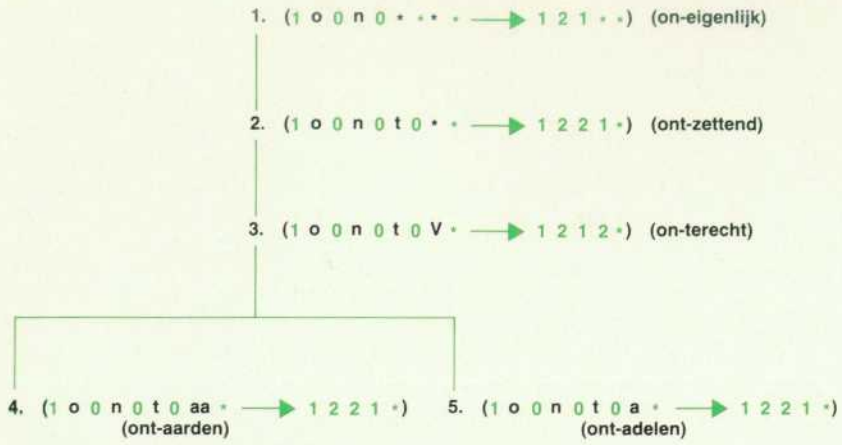


Fig. 1. *This part of the tree of linguistic rules handles the prefix* on(t). *For example, in analyzing the pattern* ontz, *the hyphenation algorithm will search the tree until it finds rule no. 3, which does not apply. Rule no. 2 will then be used.*

set theory. The inclusion relation of sets can be used to introduce a partial ordering of these sets (or linguistic rules).

Now that we have an ordering of the rules, they can be organized into a kind of tree structure in which the most general rules are placed near the root and the more explicit rules are placed near the leaves. The idea is to follow a path in the tree until a vertex is reached with the most explicit rule that still matches the given letter combination (pattern).

Later in this article we will give a detailed description of the way the tree is organized and how the searching algorithm works.

In the two examples that follow, the first example explains the notion of hierarchy, and the second example shows the use and structure of the tree.

For the first example, consider again the rule

(a) * V 0 C 0 C 0 V * → * 2 1 2 *

This rule has exceptions (in the Dutch language). With certain pairs of consonants, hyphenation will take place in front of the two consonants. One of these combinations is gr, as we saw before in the word in-te-gra-tie (note that the combination nt behaves normally). The rule for gr is as follows:

(b) * V 0 g 0 r 0 V * → * 1 2 2 *

Rule (b) is more specific than (a), that is, all patterns that match rule (b) also match rule (a). The reverse is not true. If we consider the rules as sets we can say that (b) is included in (a). The search mechanism always takes the most specific rule, so it will decide to use rule (b) instead of (a) if the combination of consonants happens to be gr.

For the second example, Fig. 1 shows the part of the tree that handles the prefix on(t). In words, these rules are: on at the start of a syllable is considered a prefix, unless the third letter is a t, in which case ont is the prefix, unless ont is followed by a vowel, in which case the prefix is again on; however, if that vowel equals a or aa, the prefix is ont.

The algorithm searches the tree until a rule is found that cannot be applied. The last rule found that can be applied is used. In analyzing the pattern ontz, for example, the tree will be searched until rule no. 3 in Fig. 1 is found, and then rule no. 2 will be used.

Organization of the Tree

In the last section we saw that we could use the inclusion relation of set theory to (partially) order the linguistic rules. In the case of a total ordering, a binary search tree would have sufficed. However, with a partial ordering we have to act differently.

Suppose we have n linguistic rules:

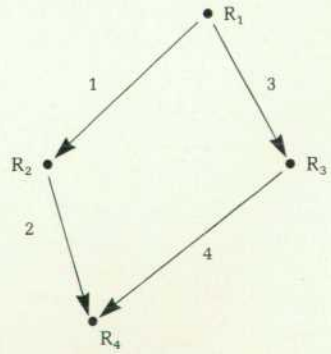
$$R_1, R_2, \ldots, R_n$$

We are going to identify these rules with the vertices of a network as follows. The successors of a rule R_a are *all* rules R_i, R_j with the properties:

1. $R_i \subset R_a$ $\quad (R_i \neq R_a)$
2. $R_i \not\subset R_j$ $\quad (\forall i, j \text{ with } i \neq j)$

To find all rules that we can apply to a specific pattern we do the following. Start at the source of the network (the most general rule). Find all successors that apply and follow all these paths until a vertex is found with the property that none of its successors contains rules that apply. This vertex contains one of the rules that we can apply to our specific pattern. This is the same idea that we have seen before in the example of Fig. 1, only here we have a network instead of a tree.

For any vertex we visit, there may be more than one successor that applies. All these paths must be followed, so it is possible to visit a vertex that has been visited before. To prevent redundancy, the searching mechanism can put a flag on each vertex that has been visited. Another way to prevent this redundancy is to adapt the structure of the network. Suppose we have the following situation:



© Copr. 1949-1998 Hewlett-Packard Co.

If we leave branch 4 out, we get a normal tree structure and do not need flags anymore. However, now there is a chance that we will apply R_3 *and* R_4, although R_4 is more specific. In practice, this has turned out not to be a serious problem, so we have decided to use this method.

We can refine the structure of this tree even more. This is especially necessary when there are many successors for each vertex, because in that case it would not be very efficient to visit all these vertices. To solve this inefficiency we use yet another property of sets: disjunction.

Let R be a vertex in the tree and

$$\Delta^R = \{R_1, \ldots, R_n\}$$

the set of all successors of R. Now we construct subsets of Δ^R

$$\Delta^1, \Delta^2, \ldots, \Delta^m$$

in such a way that:

1. $\bigcup_{i=1}^{m} \Delta^i = \Delta^R$
2. $\Delta^i \cap \Delta^j = \varnothing \quad \forall\, i, j;\ i \neq j$
3. $R_k \cap R_l = \varnothing \quad$ with $R_k, R_l \in \Delta^i; i = 1, \ldots, m$

Now we have a partition of Δ^R with the property that the elements of a subset in the partition are mutually disjunct. Note that the elements of these subsets are generalized linguistic rules, that is, they are sets themselves.

It turns out to be possible to use a simple alphabetical ordering when dealing with disjunct rules, so that in each subset of the partition we can use a binary search mechanism. This makes the search considerably more efficient. An outline of the recursive procedure vertex, which fills a stack with all vertices of a (sub)tree which contain the most specific rules that apply to a given pattern, is as follows:

```
procedure vertex (pattern, root'pointer, stack);
member := true;
call fillsets (sets, numsets);
i := 1;
while i ≤ numsets do
    call binsearch (sets(i), pattern, found, pointer);
    if found then
        member := false;
        call vertex (pattern, pointer, stack)
    endif;
    i := i + 1
endwhile;
if member then
    call push (stack, root'pointer)
endif;
return;
```

The procedures and variables used in this procedure are:

pattern	The pattern being investigated.
root'pointer	Points to the root of the (sub)tree where the searching takes place.
stack	The stack to be filled with the rules that apply to pattern.
member	This logical variable decides if the rule at the root of the (sub)tree will be put into the stack.
fillsets	This procedure fills the array sets with the subsets of the partition. numsets is the number of subsets.
binsearch	Performs a binary search in a subset. If a rule is found that applies to the pattern, found will be set true and pointer will point to this rule.
push	Puts the rule referred to by root'pointer into the stack.

Description of the Algorithm

Input for the algorithm is the word to be hyphenated together with left and right limits that define a "red zone." A word break has to be found within the red zone. The word to be hyphenated is assumed to contain only uppercase alphabetic characters—no numerics or specials.

The procedures used in the algorithm are:

make-stack (pattern)

This procedure fills the stack with all generalized rules applicable to this pattern. To do this, the procedure vertex is called.

analyse-break (pattern, ready)

This procedure analyzes breakpoint positions in the neighborhood of breakpoints just found. The logical parameter ready is set true if there is a word break within the red zone.

The algorithm consists of the following steps:

1. Search the exception dictionary. If the word is found the algorithm stops.
2. If the procedure is called for the first time in an HPWord session, an initialization will take place. See the next section on implementation.
3. The next step of the program consists of condensing the word. Here a mapping takes place from all combinations of two or three letters that can always be considered one letter (such as QU) into an arbitrary set of lowercase characters.
4. Next, the program performs a first analysis of the word. The program searches the word for combinations of two letters that cannot be pronounced within one syllable. If such a combination is found, a breakpoint will be forced between those two letters.
5. In the last step a stack is built containing all rules applicable to a specific pattern. The pattern is changed according to these rules. This is done for all possible patterns in the word, with a maximum of seven letters in one pattern. The procedure is as follows:

```
n := min (7, wordlength);
while n > 0 do                    The tree consisting of rules for pat-
    j := 1;                       terns with n letters is searched.
    while j ≤ (wordlength − n) + 1 do
      call make-stack (pattern);
      while "stack not empty" do
          "Use top of the stack to change (s_{j−1}, ..., s_{j+n−1})";
          call analyse-break (pattern, ready);
          if ready
          then "return word breaks";
               "stop the algorithm"
          endif;
          call pop (stack)
```


© Copr. 1949-1998 Hewlett-Packard Co.

```
        endwhile;
        j := j + 1
    endwhile;
    n := n − 1
endwhile.
```

Implementation on the HP 3000

The program is written in SPL/II, the HP 3000 system programming language. It uses a file DUHYPRUL and an extra data segment. The file DUHYPRUL contains the hyphenation rules for the Dutch language and some additional information (parameters, etc.).

The first time the algorithm is called when working with HPWord the contents of the file DUHYPRUL are put into some owned variables* and into an extra data segment created at this moment (maximum size 25K words). The extra data segment now contains all the hyphenation rules and the pointers for the tree structure. There is room for approximately 1000 rules.

The extra data segment is organized as shown in Fig. 2. Each of the first 1024 words contains a pointer to the rule with the same number. Thus word no. 40 points to rule no. 40.

Fig. 3 shows the organization of a single rule. This is a part of the stack and contains the code of the rule:

$$\sigma_0, \lambda_1, \sigma_1, \ldots, \lambda_k, \sigma_k \rightarrow \sigma'_0, \sigma'_1, \ldots, \sigma'_k$$

The 10-bit variables p^j_σ and p^j_λ are the pointers to other rules and refer to the tree structure. They are indirect pointers, that is, they each contain the address of the extra data segment in which the address of the actual rule can be found (see Fig. 2).

Performance

Various test files were used to test the algorithm. These files consisted of newspaper articles, part of a book about word processing, and an internal HP report. The total number of words in the test files was 6906, and the total number of breakpoints found was 7096. For regular Dutch words, the error percentage was 0.63%. The actual error percentage was 0.96%, including borrowed words and foreign words.

The majority of the errors occurred in compounds, especially those linked by an s (bedrijf-sadministratie instead of bedrijfs-administratie). The testing was carried out by a stand-alone test program outside HPWord. This test program tried to find *all* breakpoints, while HPWord only needs to find *one* breakpoint. Because in general the safest breakpoints are found first (in step 4 of the algorithm), the error percentage of the hyphenation in HPWord will probably be lower than the 0.96% found.

*Owned variables are local variables in a procedure that keep their values when returning to the main program.

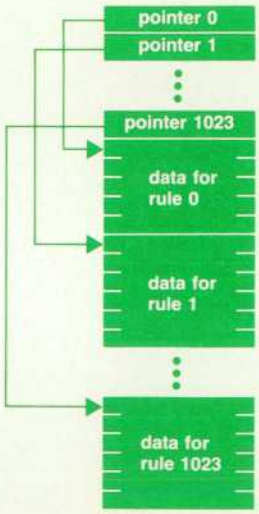


Fig. 2. *Hyphenation rules and pointers for the tree structure are stored in an extra data segment organized as shown here. There is room for 1024 rules.*

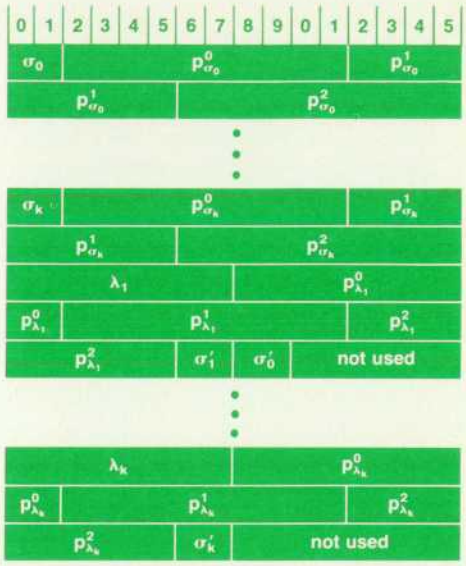


Fig. 3. *Organization of a single rule in the extra data segment. The σ and λ variables are part of the rule and the p variables are pointers to other rules in the tree structure.*

Acknowledgments

The research done in developing this algorithm formed the final stage of the author's studies in computer science. Supervising the project were Professor Dr. A. Ollongren of the University of Leiden and Tan Eng Hoo of HP Netherlands.

References

1. H. Brandt Corstius, "Exercises in Computational Linguistics," *Mathematical Centre Tracts*, no. 30, Amsterdam, 1970.
2. Kari-Jouko Raiha, "Regular Expressions: A Tool for Specifying Hyphenation Rules in Text Manipulation Systems," *Proceedings of the International Computing Symposium 1983 on Application Systems Development*, German Chapter of the ACM, Stuttgart, 1983.
3. M. Boot and H. Koppelaar, *De Tekstmachine*, Academic Service, The Netherlands, 1982.

© Copr. 1949-1998 Hewlett-Packard Co.

Designing Software for the International Market

A designer has to allow for differences in spelling, syntax, character sets, times, date formats, terminal capabilities, and many other factors.

by Heather Wilson and Michael J. Shaw

USERS OF OFFICE SOFTWARE products are generally not technically trained. They want a user interface that is as easy as possible to understand. On the most fundamental level, an office product such as a word processor must interact with the user in the user's native language. To make this possible, office products must be designed so that they can be localized. Localization is the adaption of a software application for use in different countries or environments.

HP software products are designed with localization in mind. HP Application Centers within each country have the responsibility for adapting each product to the local environment. It is the responsibility of the designers of the product to make the Application Centers' job as easy as possible.

Many non-English speaking countries do not have a wide range of office products available in their native languages. By producing localizable products and then localizing those products, Hewlett-Packard offers international companies the opportunity to buy one office automation solution and use it within different countries.

General Design Principles

What problems do we face when trying to design localizable products? It is important that the user of the localized product does not suffer reduced functionality. Nor should any restrictions be imposed as a result of localization of the product.

The person who localizes the product should not have to touch the source code. Localization should be possible without taking the chance of introducing new bugs. This means that a localizable application will not have any functions within the source code that are dependent on the local language or the customer.

To give an example, imagine that a product has the string Please type your name as a prompt. To localize the product into French, this prompt would have to be translated. If this string were embedded in the code, we could tell the French localizer where it could be found. However, there could be several instances of this string within the source code, and the French localizer may fail to translate all of them. Furthermore, tampering with the source code could result in changing unrelated parts of the code by mistake. If the string appeared in a menu, localization could result in an unaligned and unattractive menu. Obviously some method must be devised to allow localization of the menus and messages without touching the code or affecting the screen layout.

A localizable application will not make any assumptions about the meaning of particular data characters. It will be possible to increase the length of messages and prompts to allow languages with long words to provide a true translation of the user interface without having to resort to abbreviations.

Although many of HP's products are menu-driven, there are some that are command-driven. Let us consider how

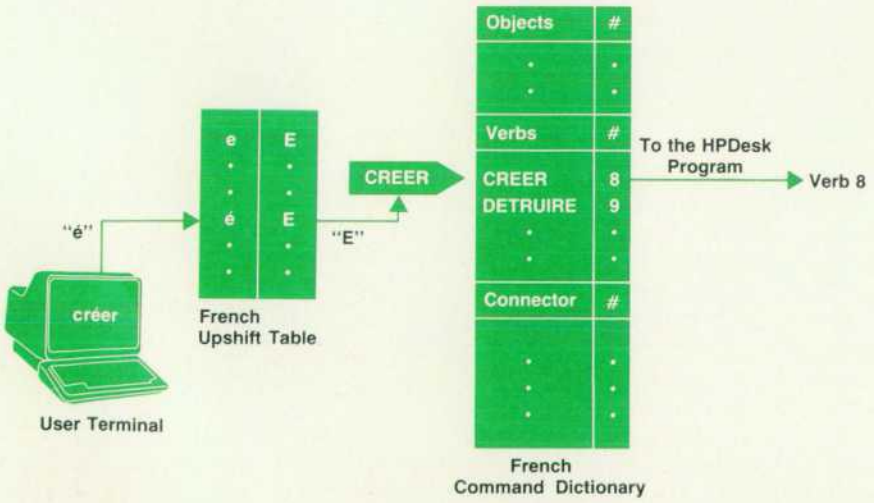


Fig. 1. *How HPDesk recognizes créer in French.*

© Copr. 1949-1998 Hewlett-Packard Co.

we might detect when the user wants to delete something if we have a command-based interface. In our program we could have a statement something like this:

```
if userInput = 'DELETE' then ...
```

Then, when the French localize the product they could change that line of code to read:

```
if userInput = 'DETRUIRE' then ...
```

If we use this method, we're again faced with having to change the program code.

Localization of command recognition presents us with other problems. HPDesk is an electronic mail system that runs on the HP 3000 Computer.* It allows users to create, send, file, and work on electronic messages. Within HPDesk there exists the concept of the current message. If I'm sent a message, I might read it, decide it is not worth keeping, and so decide to delete it. I could do this by typing delete it. Were I French, I would like to say le détruire because détruire le does not make sense. It is like asking me to type it delete in English! I therefore want to be able to tell my command recognizer which order to expect the verb (delete) and the object (it) in. And I do not want to have to change the program when the French want the order to be object, then verb.

Character Set Differences

If a product is going to be used in an English speaking country, then only ASCII characters need be displayed. However, problems arise if that product is then localized into French. If the ASCII character set is used, there is no way that you can display or store é. This is a serious problem, since é is the most common character in the French language.

Thus, a localizable program must be able to recognize and process all local character sets. It must display, take as input, and print each local character set. When localizing an electronic mail system, for example, the system should be able to send a message in one local character set and still be able to display that message at a destination where another local character set is being used.

Localization would be a lot simpler if all terminals were able to display and transmit HP extended ASCII characters. Then, we would not need to know which terminal we were talking to. Unfortunately, because our current ideas about localization did not arrive overnight, all terminals are not the same. Historically, this is what happened.

HP's earliest terminals only knew about ASCII characters. The eighth bit in each byte was used for parity checking. The HP 2642 is one such terminal.

It was then realized that some of the less frequently used ASCII characters could be used to represent characters special to the country in which the terminal was being used. For example, in Sweden the HP 2645 has its] key replaced by an Å. Whenever this key is pressed an Å is displayed, but the ASCII representation of] (decimal 93) is transmitted.

The next set of terminals sent not one but two characters down the line for many of the non-ASCII characters. For example, on an HP 2626W Terminal, the user would get an é to be displayed by first hitting the key with an acute on it and then the e key. After the first key was hit the cursor would stay where it was so that when the e was hit, an é would be displayed. Hitting these two keys, however, would cause two characters to be transmitted: an acute (decimal 168) and then an e (decimal 101).

HP then introduced a standard for extended ASCII which it called the Roman8. HP terminals have more than one character set. There is the base set containing the normal ASCII characters, the math set with integral signs, etc., and the line drawing set. HP then brought out a new set called the Roman extension set. By selecting this set as the alternate character set of your terminal, you can access HP's Roman8 characters. The HP 2623A is a terminal that behaves in this way.

The latest generation of terminals, including the HP 2628, the HP 2627, and the HP 150 Personal Computer, still have the Roman8 characters available to them. Now, however, the Roman8 characters are in the base character set, so you do not have to select an alternate character set to get at them.

Having so many terminals presents many problems for a product that has to try to store and display all European characters as best it can given the capabilities of the terminal.

Other Differences

Related to the display of text, but not actually terminal dependent, is the concept of hyphenation used in secretarial word processors. One such word processor, HPWord, which runs on the HP 3000, will word wrap text as it is input. If a word is long, it will try to hyphenate it rather than put it onto the next line. In English, the hyphenation algorithm will not hyphenate words that are five letters long and have an s at the end. In Italian, such a rule would produce strange results because Italian plurals are not made by adding an s to the word. The English algorithm also has a rule that it will not split two-consonant groups like ph, th, and ch. In other languages, however, it might be quite acceptable to split such groups.

Office products face many localization problems when information they produce crosses either time or language boundaries. HPTelex is an HP 3000 office product that allows the user to create the text of a Telex message using a word processor, and then to ask the computer to send

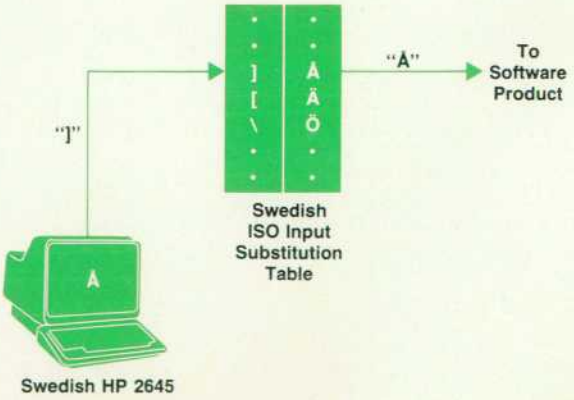


Fig. 2. *ISO input substitution.*

*The full name of HPDesk is HPDeskManager. It was originally called HPMail.[1]

© Copr. 1949-1998 Hewlett-Packard Co.

this message whenever it thinks best. The Telex network transmits only a very limited character set—capitalized ASCII characters and a few other characters. What these "few other characters" are varies from country to country. For example, @ is valid in most but not all countries. Because Telex messages often travel across country boundaries, all characters that are not standard across the world must be filtered out by the HPTelex software.

An HPDesk user must specify to whom electronic mail is to be sent. Suppose an HPDesk user in the U.S.A. wants to send mail to a Mr. Förster in Germany. We really cannot expect the American sender to know that the o has an umlaut over it. However, were we to just upshift all names as they were input this would lead to problems. Å is a totally different letter from plain A in Finnish, for example. So people with an Å in their names would have their names changed when upshifted.

As one might imagine, a diary product hits problems at time boundaries. Within HPDesk there is a time management facility which, among other things, allows users to send messages requesting appointments. To record the appointment so that HPDesk will produce a reminder of it later, the recipient simply types file it in diary. If I, in Britain, request a conference with some people in California for 5:00 p.m. on 13.12.84 British time, those people in California must see an appointment for 9:00 a.m. on 12/13/84. Making this appointment highlights two problems. The first problem is that the date format in Europe is day/month/year but it is month/day/year in the U.S.A. The second problem is that there are time differences between countries that we must compensate for.

Design Solutions

Let us see how we can overcome the problems that we have outlined. The solutions we shall offer are those we have used for office products. They may, however, be readily applied to other fields.

As we mentioned before, ASCII character representation uses only seven bits if no parity checking is required. The Roman8 European character representation standard developed by HP uses all eight bits of a byte. All ASCII characters remain as they were, except that the eighth bit is always off. When the eighth bit is on we are able to represent another 128 characters. So the ç character is the ASCII 5, but with the eighth bit set. Documents and messages should, wherever possible, be stored in Roman8 form.

Although we discussed menu and message localization problems in the same breath, we shall consider their solutions separately here. We shall look at menus first. So that we do not have to change the program file when we localize a menu-based product, we put our menu definitions in an auxiliary file. Our program knows about this file and accesses it at run time. When we localize we only change the contents of this file. On an HP 3000 Computer the menu definitions that go into this file are built using VPlus/3000. VPlus/3000 allows us to define not only menu layouts and wording, but also where on the menu the user may enter information. When an Application Center translates our menu set they only change the menu wording, not the layout or areas where the user may enter information. VPlus/3000 also provides us with a set of procedures that a menu-based program may use. They allow the caller to draw menus and to find out what information the user typed into the menu. Note that this method satisfies our requirements that neither the program nor the menu layouts be affected by localization.

The message localization problem is overcome in a similar way to menu localization. Again we use an auxiliary file, but this time it contains messages rather than menus. Instead of our program asking the user's name like this:

```
writeln ('What is your name ?')
```

the program calls a procedure that gets a message from our message file. This procedure then outputs the message to the terminal, just like the writeln would do. Each message in the message file has a unique number. We might have made our question message number 6 in the message file, for example. The message file is prepared using a simple editor by the people writing the program. The message output procedure also allows us to substitute run-time values into our messages. We might, for example, want to tell our users how many messages they had received. We would do this by putting a message

```
'You have ! new messages'
```

into the message file. When this message is output we would tell the output procedure that we want the ! to be replaced by the number of new messages.

The Application Centers translate the message file into their languages. All characters in the message file are stored in Roman8 form. Some terminals, such as the HP 2645, can only display ASCII characters. So, if one of our messages contains an é (decimal 197) we must translate the character to an e (decimal 101) before we output it to the terminal.

If we do not have a menu-based user interface, we must be able to recognize commands. To allow localization of commands, HPDesk uses yet another file, which it calls a dictionary. This file contains a list of verbs (like create), objects (like text), and connectors (like of) that HPDesk will accept in the local language. Let us consider what happens if an English-language HPDesk user wants to create some message text. To do this the user would type create text. This line will first be upshifted to capitalized ASCII characters so that the user need not worry about the case of the commands. This means that not only will e be translated

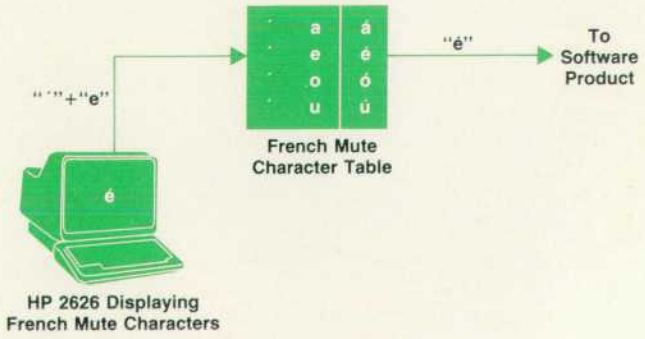


Fig. 3. *Mute character input.*

© Copr. 1949-1998 Hewlett-Packard Co.

to E, but é will be made into an E also. Both the CREATE and the TEXT will be compared with the list of words in the dictionary file. Each comparison tells two things: whether the word is a verb, object, or connector, and a number indicating which verb, object, or connector it is. CREATE might be found to be the eighth verb and TEXT the second object. What corresponds to verb 8 or object 2 is determined by the Application Centers. In France, for example, they might decide that the English command CREATE should be known as CREER.

To upshift e to E or é to E, we use a table called the upshift table, which is defined by the Application Centers. Unfortunately, upshifting is not a one-to-one mapping. The German character ß, for example, should upshift to the characters SS. Fig. 1 illustrates how HPDesk would recognize the French command créer.

When outlining the problems of localization we mentioned that delete it translates to le détruire in French. To accommodate this, the command recognition method needs to allow the Application Centers to specify the expected order of the user's input. In English delete it is verb, then object, but in French we would expect object, then verb.

Tables for Character Selection

Coping with HP's range of terminals and their different approaches to non-ASCII characters presents us with many problems. This can be overcome by asking the Application Centers to specify to the software product the language that is being used, and by designing the software product to interrogate the terminal to find out what sort of terminal it is. Then, whenever any text is input from the terminal, or any messages or text are output to the terminal, one of four tables can be used to filter the characters. These tables are the degradation table, the ISO substitution input table, the ISO substitution output table, and the mute character table.

The degradation table is used for those terminals that can only display ASCII characters. If I am in France working on an HP 2645 Terminal and I want to read a piece of text that contains the ç character, each ç must be translated to c before it appears on my terminal. If the Roman8 representation of ç were sent to my HP 2645 undegraded, then my terminal would ignore the eighth bit of the character and a 5 would appear. As with the upshifting used for command matching, the degradation should account for not only one-to-one degradations, but also one-to-two degradations like ß to ss.

The ISO input substitution table is used to handle input from those terminals that redefine little-used ASCII characters to represent their local characters. The Swedish HP 2645 is one such terminal. If I were to hit the Å key on this terminal, a] would be transmitted because the software product would know that it was talking to an HP 2645 and that the language in use was Swedish, and would use the ISO input substitution table set up by the Swedish Application Center to store this] (decimal 93) as an Å (decimal 208) as the user intended. Fig. 2 illustrates this.

The ISO output substitution table is like the ISO input substitution table but in reverse. That is, if a piece of text or a message contains a Roman8 Å and it is to be output to an HP 2645 in Sweden, we must change the transmitted character to an ASCII] so that it will appear as an Å on the Swedish screen.

The mute table is used when accepting input from those terminals that transmit local characters as two characters. We saw earlier that an é on an HP 2626W can be displayed by hitting the acute key and then the e key. When the acute character is transmitted, the software realizes that this character can go with an a, e, o, or u to form an á, é, ó, or ú. It knows this because the mute table set up by the Application Center tells it so. When the e is transmitted, the software looks in the mute table to find out what Roman8 character should replace an acute and an e. In this case, the mute table says it should be an é. Fig. 3 illustrates this.

Solutions for Names

Remember the American trying to send to Herr Förster? Earlier in the discussion we showed that upshifting names was not good enough because this causes them to be transformed in some countries. The solution is to store and display names in Roman8 form, but upshift them only for comparison. So when the message for Mr. Forster (the American who sent the message did not put an umlaut on the o) arrives at Herr Förster's computer, the upshifted version of Forster will match with the upshifted version of Förster). However, whenever Herr Förster types his name, he will see it as Förster and not FORSTER.

While on the subject of names let us discuss another problem that English speaking people rarely encounter, but is common in Europe. How do we deal with a surname containing more than one word? Suppose a user gives his

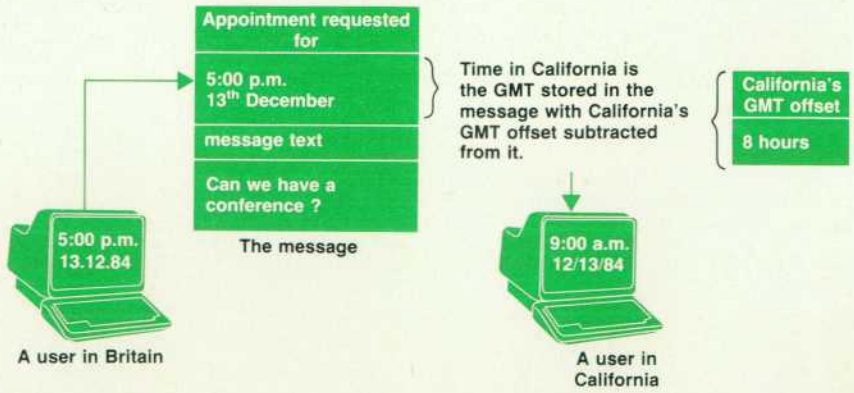


Fig. 4. *Sending an HPDesk message requesting an appointment.*

© Copr. 1949-1998 Hewlett-Packard Co.

name to the software product as Pierre de Sainte Maresville. What should be taken as the surname? Is it "Maresville," "de Sainte Maresville," or "Sainte Maresville?" This problem can be overcome to a certain extent by allowing hyphens to appear in names and by insisting that the surname be all one word. So Pierre de Sainte Maresville must specify his name as Pierre de-Sainte-Maresville.

One last thing about names. European names are often a lot longer than English names. So if name fields are used, they must be able to accommodate long names.

Dates and Times

Let us now try to solve the date and time problems that diary products face. The date format problem can be easily overcome by allowing the Application Centers to specify the format. A stickier problem is that of time differences. HPDesk solves this by making each computer keep a record of its offset from Greenwich Mean Time (GMT). The date and time on a message requesting an appointment are stored in normalized GMT form. When a date or time is displayed on a specific computer it is corrected by deducting that computer's GMT offset from the normalized GMT date and time. So, when an appointment request message for 5:00 p.m. on 13th December is sent from Britain to California, that date and time is stored in the message because Britain is on GMT in December. However, when the message is displayed in California, eight hours are subtracted because California is eight hours behind GMT. Fig. 4 illustrates this.

Conclusion

It is obvious that it is not easy to make software applications easy to localize. Message lengths, character sets, time and data formats, and the syntax of commands must all be taken into account. All of these and more must be easy to change without having to touch the source code of the product. Once you change the source code, the result is a new product.

Reference

1. I.J. Fuller, "Electronic Mail for the Interactive Office," *Hewlett-Packard Journal*, Vol. 34, no. 2, February 1983.

Authors

September 1984

4 Transmission Impairment Measuring Set

David R. Novotny

Born and raised in La Mesa, California, David Novotny graduated from California State Polytechnic University at Pomona with a BSEE degree in 1973. In 1977 he received his MSEE degree from Stanford University. He joined HP in 1973, helped design the 83001A Portable Microwave Repeater, then joined the Colorado Telecommunications (then Delcon) Division in 1975 to work on the receiver and HP-IB control of the 4942A, 4943A, and 4944A TIMS. Loaned for six months to the Colorado Springs Division, he designed the HP-IB system for the 1640A Serial Data Analyzer. He led the early design efforts on the 4935A TIMS and in 1980 moved into marketing. He's now product marketing engineer for the 4945A TIMS. David is married, has two children, and lives in Colorado Springs. His interests include flying, amateur radio, gardening, and family activities.

James P. Quan

After graduating from California Polytechnic State University in San Luis Obispo in 1980 with a BSEE degree, Jim Quan joined HP's Delcon Division (now Colorado Telecom) to work on the 4945A. He contributed to the later R&D stages on much of the receiver hardware and followed the instrument through the first few months of production. Jim spends his free time participating in intramural sports and other outdoor activities.

Jerry D. Morris

Jerry Morris majored in communication electronics in high school in his native Des Moines, Iowa. His college education was interrupted by a three-year tour of duty in the U.S. Army, where among other things he served as an instructor in computer repair. He received his BSEE degree from Iowa State University in 1975 and then joined HP. He worked as a production engineer for nine months before starting his career as a development engineer in analog and digital hardware design. He received his MSCS degree from the University of Santa Clara in 1983. His contributions to the 4945A TIMS included the initial design of the video generation electronics and a major portion of the system processor firmware design. His outside interests include photography, music, and home remodeling. He's married and lives in Colorado Springs, Colorado.

Jeffrey Tomberlin

Jeff Tomberlin joined HP's Colorado Telecommunications Division in 1980 as a development engineer after graduating with a BSEE degree from Georgia Institute of Technology. He took over responsibility for Dick Lee's designs in the 4945A, contributing to the transients measurement hardware and software. His interests include tennis, golf, swimming, and (recently) skiing.

Gordon A. Jensen

Gordon Jensen designed the hardware and software for the three I/O modules of the 4945A TIMS. A 1981 BSEE graduate of the University of California at Davis, he joined HP's Colorado Telecommunications Division in 1981. He lives in Colorado Springs and enjoys skiing, climbing, and reading "trash" fiction.

Charles P. Hill

Chuck Hill joined HP's Data Terminals Division in 1978 to work on graphics terminals. He then moved to the Colorado Telecommunications Division to work on the 4945A TIMS receiver. He is now designing protocol analyzers. Chuck was awarded BSEE and MSEE

© Copr. 1949-1998 Hewlett-Packard Co.

degrees from Stanford University in 1978. He lives in Black Forest, Colorado and enjoys sports cars, amateur radio, and remodeling his home.

13 Master-Slave TIMS Operation

Teresa L. Reh

Terri Reh joined HP in 1979 as a development engineer after receiving her BSEE degree from the University of California at Santa Barbara. She designed the system digital hardware and the master/slave firmware for the 4945A TIMS, and is now writing applications software for protocol analyzers. She's also working on her MSEE degree at Colorado State University. Married to another HP R&D engineer, Terri lives in Colorado Springs and enjoys sewing, cooking, camping, and backpacking.

15 Testing the TIMS

Scott S. Neal

Scott Neal joined HP in 1978 with four years' experience in IC and memory design. As a member of HP's technical staff, he's done IC, digital, and software design and developed the automated software QA programs for the 4945A TIMS. He received a BSEE degree and a BA degree in economics from Rutgers University in 1974. Scott comes from Columbus, Ohio. His interests include physics research, skiing, and hiking. He's a resident of Woodland Park, Colorado.

Allan W. Dodge

Al Dodge received his BSEE degree from the University of California at Los Angeles in 1968 and his MSEE from Stanford University in 1976. He joined HP in 1974 after serving in the U.S. Air Force as an electrical engineer. He designed many of the cable maintenance and construction products of the former Delcon (now Colorado Telecom) Division, and developed the automated 4945A performance test system. A native of Alhambra, California, Al is married, has two children, lives in Colorado Springs, and enjoys skiing and cycling.

Kurt R. Goldsmith

A native of Salem, Oregon, Kurt Goldsmith graduated with his BSEE degree from Oregon State University in Corvallis in 1978. He joined HP's Corvallis Division in 1978, and worked that summer and fall on the HP-85 Computer. Kurt and his wife moved to California in 1979 when he took a position with HP's Delcon Division, and later moved with the division to Colorado Springs in 1981. Kurt contributed to the power supply, CRT drive, and other circuits of the 4945A TIMS and was responsible for the reliability and environmental test program. He received his MSEE degree from Stanford University in 1983. He enjoys the snow skiing in Colorado and watching his 13-month-old son grow up.

19 Semiconductor Research Corporation

Richard A. Lucic

Formerly manufacturing manager of HP's Loveland Technology Center, Rich Lucic is now serving under HP's loaned executive program as the Semiconductor Research Corporation's manager of manufacturing sciences research programs. A native of Sharon, Pennsylvania, he received his Engineer of Metallurgy degree in 1966 from the Colorado School of Mines, and his MS degree in 1971 from Stanford University. An HP employee since 1966, he has authored several articles on IC manufacturing technology and is named as an inventor on a patent on cooling techniques for sputter deposition. Rich is married, has two children, and now lives in Raleigh, North Carolina. He serves on a pair of local school committees, has an interest in personal computers, and relaxes by jogging and fishing.

26 Hyphenation Algorithm

Paul R. Smit

Paul Smit is a consultant at HP's Dutch Application Center at Amstelveen, The Netherlands. He joined HP in 1983, developed the hyphenation algorithm described in this issue for the Dutch version of HPWord, and is now helping to implement the algorithm for the Danish version of HPWord. Born in Sassenheim, The Netherlands, he studied computer science at the University of Leiden, receiving his Doctoraal diploma (equivalent to a Master of Science degree) in 1984. He is married, lives in Leiden, plays bridge and flute, and enjoys science fiction and bicycle touring.

31 Software for the International Market

Heather Wilson

Heather Wilson is localization coordinator for office products at HP's Office Productivity Division. A native of Perth, Scotland, she received her BSc degree in computer science from the University of Strathclyde in 1983 and joined HP the same year. She's a member of the British Computer Society. Now a resident of Reading, Berkshire, she enjoys ballroom dancing and theater.

Michael J. Shaw

Mike Shaw holds a BSc degree in physics from Manchester University and an M.Phil. in operational research and control engineering from Cambridge University. He joined HP's Office Productivity Division at Pinewood, near London, in 1982 and has contributed to the development and localization of HPDesk. Mike was born in St. Albans, Hertfordshire and now lives in Bracknell, Berkshire. His interests include Latin American dancing and graphic art.

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

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

HEWLETT-PACKARD JOURNAL

SEPTEMBER 1984 Volume 35 • Number 9

Technical Information from the Laboratories of Hewlett-Packard Company

Hewlett-Packard Company, 3000 Hanover Street
Palo Alto, California 94304 U.S.A.
Hewlett-Packard Central Mailing Department
Van Heuven Goedhartlaan 121
1181 KK Amstelveen, The Netherlands
Yokogawa-Hewlett-Packard Ltd., Suginami-Ku Tokyo 168 Japan
Hewlett-Packard (Canada) Ltd.
6877 Goreway Drive, Mississauga, Ontario L4V 1M8 Canada

0200020707&&&BLAC&CA00
MR C A BLACKBURN
JOHN HOPKINS UNIV
APPLIED PHYSICS LAB
JOHNS HOPKINS RD
LAUREL MD 20707

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

5953-8527

© Copr. 1949-1998 Hewlett-Packard Co.