HEWLETT-PACKARD
JOURNAL

SUM OF AUTOCORRELATIONS

HEWLETT

HEWLETT-PACKARD JOURNAL

December 1988 Volume 39 • Number 6

Articles

© Hewlett-Packard Company 1988 Printed in U.S.A.

Departments

In this Issue

If you read "What's Ahead" in the October issue, you probably expected to find articles on HP's NewWave environment for personal computers in this issue. Don't bother to look. They're not here. It seems that the second version of the NewWave environment—the version described in the articles—hasn't been released yet. We'll bring you the articles when the new version is released.

In place of the NewWave environment, we have five articles on the design of the HP 8145A Optical Time-Domain Reflectometer. This story is distinguished by one of those remarkable flashes of inspiration that instantly changes "No one knows how to do this" to "Of course! How simple!" An optical time-domain reflectometer, or OTDR, is an instrument for characterizing the long optical fibers used in telephone and data networks. Similar in principle to a radar system, an OTDR sends a pulse of invisible infrared light into one end of the fiber, measures the light that comes back as a result of scattering and reflection, and plots the returned optical power as a function of time, which in this case is interpreted as distance. On this plot, one can see how lossy the fiber is and locate irregularities such as splices, connectors, and breaks. Like many modern radar systems, but unlike earlier OTDRs, the HP 8145A uses a spread-spectrum technique. Instead of a single pulse of light, it transmits a coded sequence of pulses, and then correlates (a mathematical operation) the returned signal with the transmitted signal. The transmitted code is spread out in time, but its correlation function is a sharp, localized impulse that jumps out of the noise and is capable of resolving very closely spaced irregularities. OTDR designers have known for a long time that spread-spectrum techniques have theoretical advantages, but only if the correlation function of the code used has a value of zero everywhere except at the single point where the impulse occurs. Unfortunately, no known code has zero sidelobes. This is where matters stood until researchers at HP Laboratories looked at certain pairs of complementary codes and found that, although the correlation functions of both codes of a pair have sidelobes, the sidelobes cancel perfectly when the paired correlation functions are subtracted. And so the HP 8145A was conceived. Beginning on page 6, you'll find a discussion of OTDR principles, a description of the HP 8145A and its contributions, and some measurement results. For the theory of complementary correlation, see page 14. The transmitter, receiver, and optical system designs, which had to meet stringent restrictions on loss, noise, and nonlinearity, are described beginning on page 22. The data processing system and user interface designs are presented in the articles on pages 29 and 35.

One of the principal objectives for the HP DeskJet printer (October 1988 issue) was that it be able to use plain paper, "plain" in this case meaning commonly used office papers. This turned out to be a major challenge for the designers because of the variety of papers found in a typical office. Correspondence papers gave relatively uniform inkjet print quality, but copier papers, surprisingly, varied widely. The main factor in their print quality variability was found to be the amount of sizing or surface starch applied in manufacturing. On page 39, Steve Bares describes the plain paper studies done for the DeskJet project.

Many of the advanced capabilities of today's electronic instruments, VCRs, automobiles, and other products are provided by built-in microprocessors. In developing these systems, a designer needs to be able to check their operation in a variety of ways. A microprocessor development system, such as the HP 64000, provides subsystems called emulators, which plug into the microprocessor socket of a system under development and allow the designer to control and observe its operation under realistic conditions. With the new HP 64700 Series emulators, designers can use personal computers or engineering workstations for microprocessor system development. The hardware and software designs for these emulators are discussed in the articles on pages 45 and 52.

We do have one article in this issue that's related to the NewWave environment. It's on page 57, and it's about an expanded memory card for the HP Vectra ES Personal Computer. This high-performance memory system supports expanded and extended memory capabilities defined by the LIM 3.2 and 4.0 industry standards. These capabilities are needed by the NewWave environment and other applications. The challenge for the designers was to make a significant contribution within a strict definition of compatibility.

The paper on page 74 is a rare species for this publication—the authors aren't from Hewlett-Packard. Although we occasionally receive papers from outside sources, in most cases the subject matter isn't sufficiently HP-related for us to justify publishing them. In this paper, the authors present a scheme for adjusting quadrature phase data sampled by the HP 5180A Waveform Recorder so that the spectrum computed from the data corresponds to simultaneous sampling of the two input channels. The recorder actually samples one channel 100 nanoseconds later than the other one.

R.P. Dolan

Cover

The autocorrelation functions of two complementary Golay codes merge into their sidelobe-free sum in this representation of the signal processing technique implemented in the HP 8145A Optical Time-Domain Reflectometer.

What's Ahead

Two product designs will be featured in the February issue. The HP 5371A Frequency and Time Interval Analyzer samples the zero crossings of one or two input signals to extract frequency modulation and characterize frequency-agile sources. It implements continuous count technology, reading zero-dead-time counters on the fly to make blocks of measurements without stopping. The HP 8904A Multifunction Synthesizer creates complex signals from common sine, ramp, square, triangle, and noise waveforms. A custom VLSI integrated circuit calculates the signal parameters in real time.

A High-Speed Optical Time-Domain Reflectometer with Improved Dynamic Range

This article presents basic information on optical time-domain reflectometry and introduces the HP 8145A, which uses a data correlation technique to increase measurement speed and dynamic range.

by Michael Fleischer-Reumann and Franz Sischka

THE TIME SEEMS LONG PAST when one had to explain what fiber optic means, and that it is not only possible to transmit data and information through tiny glass fibers, but it can be done with higher bandwidth and over longer distances than with copper coaxial cables. In truth, it was only about ten years ago, in the early seventies, when fiber optic transmission began over multimode fibers at a wavelength of 850 nm and data rates of several megabits per second. Today's state-of-the-art long-haul fiber transmission systems operate at speeds up to 2.3 Gbits/s on 9-μm single-mode fibers with 1300-nm lasers spanning link lengths of 50 km. Even longer distances, about twice as long, can be achieved operating in the third transmission window, using 1540 nm as the carrier wavelength. This is especially important for submarine cables, a rapidly grow-ing application area.

Thanks to high-volume production, component prices are dropping, so that fiber as a transmission medium is also becoming attractive for shorter distances, such as sub-scriber loops, local area networks (LANs), and wide area networks (WANs).

Fiber Optic Measurements

Going along with installation and use of fiber optic trans-mission links is the need for reliable measurement equip-ment to characterize components and systems in R&D, pro-duction, and maintenance. Basic measurement instru-ments, including sources (light-emitting diode and laser diode), power meters, attenuators, and accessories, have been described in previous articles.[1] The table below lists

Fig. 1. *The HP 8145A Optical Time-Domain Reflectometer makes one-port impulse response mea-surements of optical fibers and cables. It operates at 1300 nm, 1540 nm, or both, and has 0.01-dB power resolution, 1-m display res-olution, and 200-km maximum range.*

the principal measurements and the types of companies that need to make them.

User	Measurement
Fiber Manufacturer	Diameter, roundness, centricity, dispersion, bandwidth, magnitude and uniformity of loss and attenuation, length
Cable Manufacturer	Dispersion, bandwidth, magnitude and uniformity of loss and attenuation, length
Installation Company, Telecom Carrier	Overall loss, attenuation of sections, losses and locations of splices and connectors, end of fiber, breaks.

In general, after the fiber is manufactured and/or has been put into a cable, there is no more need for mechanical or transmission-speed measurements. Focusing on measurements of the transmission medium itself, be it a bare fiber or a completely installed, on-duty, long-haul telecommunication link, all of the very different users need to make basically the same measurements. What is important is that these measurements can all be performed by a single stand-alone instrument, the optical time-domain reflectometer (OTDR).

Optical TDR Principles

Looking at the list of different results that various users want, one might think that an OTDR would have to operate in many different measurement modes. This is not true. An OTDR can only perform a single measurement.

An OTDR performs a one-port measurement of the impulse response of a fiber by stimulating the fiber under test with a pulse and displaying the amplitude of the reflected (Fresnel) and backscattered (Rayleigh) power[2,3] versus distance on a (half logarithmic) screen. Distance is measured as it is in a radar system, by measuring the time between transmission of the pulse and reception of the reflected and backscattered power. This measurement makes it possible to analyze:

- Overall link loss (dB) or attenuation of the fiber (dB/km)
- Locations and losses of inhomogenities like bends, splices, and connectors
- Location of the end of the fiber.

Since the OTDR provides all of this information by performing only a one-port impulse response measurement, the impulse response of the fiber must contain all the desired information. For someone who is accustomed to measuring with electrical TDRs, this might be surprising. The difference between electrical and optical time-domain reflectometry stems from the presence of the Rayleigh scattering, which is the main physical reason for the attenuation within a glass fiber.

What is Rayleigh scattering and why does it contain so much information? At every point of the fiber a small portion (depending on fiber type and wavelength) of the transmitted power is scattered in all directions. A small portion of this scattered power—about -48 dB at 1300 nm in a single-mode (SM) fiber for a 1-μs probe pulse width—is

LEFT:	0.628 km
RIGHT:	67.803 km
DIST:	67.175 km
LOSS:	18.378 dB

0.000 km   n: 1.4580   lc: 1.0000   72.000 km
0.000 km   1.000 us   1300 nm   72.000 km
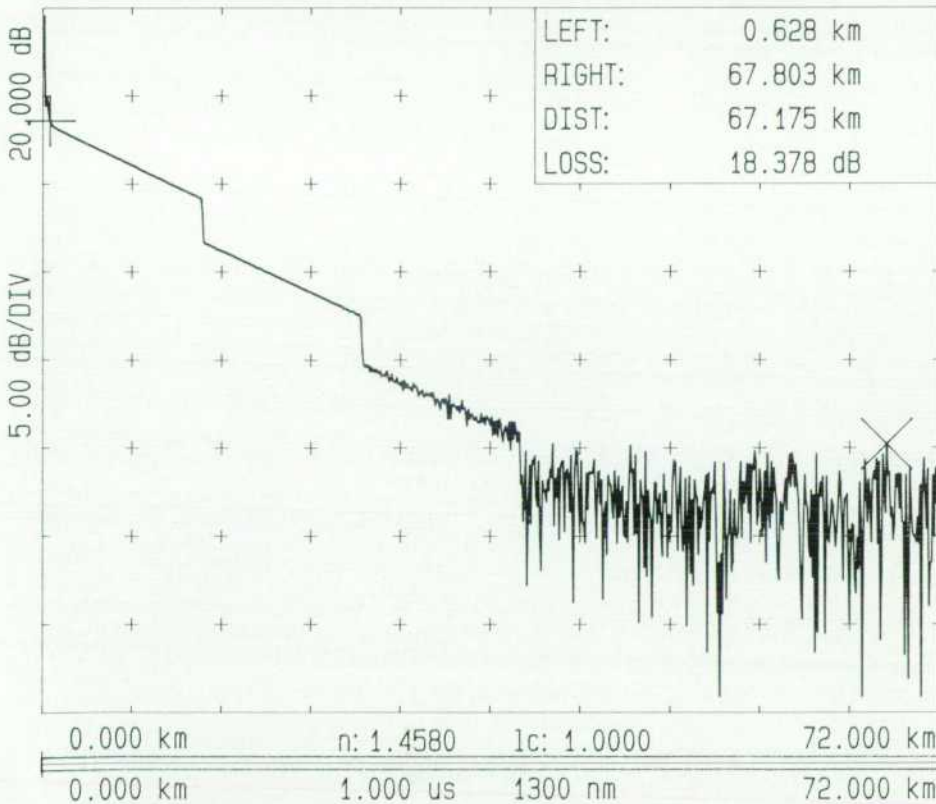
Fig. 2. Backscatter curve of a 50-km fiber consisting of four 12.5-km segments. Dynamic range is 18 dB peak for a ten-second measurement.

Technical Risk Reduced by Joint Development Effort

The HP 8145A Optical Time-Domain Reflectometer resulted from an unusual-for-HP joint effort between HP Laboratories (HPL) and the HP Böblingen Instrument Division (BID), made necessary by an unusual set of circumstances and market conditions.

By 1984, some of HP's major competitors had established strong positions in the fiber optics test and measurement field. Their product lines were anchored by a key piece of measurement gear, the optical time-domain reflectometer (OTDR). At that time, HP had no such product. In an effort to remedy this situation, two independent efforts got underway to address the problem, one in BID's R&D laboratory and the other in HPL's wave technology department.

Late in 1984, the HPL team proposed a new implementation of a complementary correlation measurement technique that potentially offered a quantum leap in the performance of the OTDR measurement. Within a few months, they successfully reduced the idea to practice in a laboratory prototype. However, it was clear that many technical and practical unknowns stood between this prototype and the realization of this new technique in a rugged field instrument.

Months of reworking and discussion followed, and in July of 1985, BID's management team made a bold decision. Reasoning that an entry into the market with a "me too" instrument would be worse than not entering at all, they decided to accept the technical risks required to provide the market with a major performance contribution. However, they did so on the condition that the HPL team that had championed the correlation idea would remain involved in the advanced development of the product—all the way through to its introduction, if necessary. This degree of involvement was highly unusual for HP Laboratories. However, HPL management recognized the importance of this project, and agreed to cross organizational boundaries to pull together a team of engineers to work with BID. Thus, the groundwork was laid for an unusual joint project.

All technology transfers between a research organization and a manufacturing division must overcome a series of difficult obstacles. Development of a feeling of shared ownership of a technology requires patience and a spirit of teamwork. Each team must learn how to deal with different working styles, different personalities, and different perspectives on how the project should proceed. In addition, the physical separation of two working groups can lead to communications problems that are difficult to overcome.

In the case of this joint OTDR project, these problems were magnified many times The teams at BID and HPL were both relatively young and inexperienced. Most of the scientists on the HPL team had never been involved with a product transfer, and initially had little idea of what was required to bring a product into production. At the same time, the team at BID was taking on the challenge of building a product much more complicated than anything that it had done before. For example, the digital signal processing requirements of the OTDR go far beyond those of any of the fiber optics instruments previously made at BID. Part of the risk (and challenge) of breaking new technical ground is that unforeseen problems inevitably arise. Many such problems did arise, and they presented difficult technical challenges for both groups.

These obstacles were compounded by the challenge of maintaining good communications and understanding. The groups at BID and HPL were separated by over 6000 miles and a nine-hour time difference. Since telephone communications were extremely inconvenient, there was a heavy reliance on electronic mail. Even under ideal circumstances, this can lead to delays and misunderstandings because of messages crossing in the mail and the inevitable "reading between the lines." In this case, language and cultural differences exacerbated the problems.

Despite all of these obstacles to progress and success, however, both teams worked hard, and in the end, achieved gratifying results. The product's performance meets and in some areas exceeds initial projections. By combining the efforts of the two teams, HP was able to introduce the OTDR in September 1987, only 26 months after the joint effort began.

Perhaps the most significant long-term result of this project has been the personal and professional growth of the members of both of the teams. The HPL team has come away with a much deeper understanding and appreciation of what it takes to turn a blue-sky idea into a product. The team at BID has acquired a broad set of technical skills, along with the enthusiasm and confidence to ably apply them to future projects. In short, this unusual-for-HP project serves as an example of how formidable technical and communications obstacles can be overcome by hard work, patience, perserverence, and dedication to a common goal.

Michael Fleischer-Reumann
Project Manager
Böblingen Instrument Division

Steve Newton
Project Manager
Hewlett-Packard Laboratories

guided by the fiber backwards to the origin.

Assuming that the fiber is homogeneous (the backscattering coefficient is constant versus distance), at every successive section of the fiber there is less power than at the sections closer to the launching port, just because of the Rayleigh scattering. Consequently, the absolute value of the backscattered power is also lower by the same factor. Because the backscattered power has to travel all the way back through the fiber to the launching port, it is attenuated again. When it reaches the launching port, it carries, in its amplitude, information about the fiber loss. Measuring the backscattered power as a function of time relative to the launching pulse gives a plot of loss versus distance, which is shown on the display.

The longer the duration of the probe pulse, the higher the amount of backscatter from a specific location. This can be easily understood from the following simplified model. Assuming that a pulse of duration T consists of a number n of short pulses of duration dt so that $T = ndt$. If every time a short pulse of duration dt propagates across a specific location z a certain amount power $P_{back}(z)$ is backscattered, then the amount of backscatter for the pulse with duration T is $nP_{back}(z)$. From this it can be seen that the Rayleigh scattering is proportional to the energy of the

probe pulse, not only to the peak power level of the pulse.

If we now assume that the fiber consists of several homogeneous parts that are spliced or connected, then at these locations there is a nondistributed or localized loss (fiber attenuation is distributed loss). The forward traveling power will be attenuated just by the amount of the localized splice or connector loss and consequently all backscatter that stems from locations beyond this point will be lower by the same amount. The backscattered light reaching the launching port will also carry this information, that is, the amount of the splice or connector loss and, of course, its location.

At a connector, if there happens to be no physical contact between the two connected fibers, there will be not only a localized loss as described above, but also a reflection caused by the change in the refractive index (an impedance mismatch in electrical terms). This holds true not only for connectors (or mechanically, elastomeric splices) but also for the glass/air transition at the end of the fiber link. At that location, the reflected power is only proportional to the power level (not the energy), and is a maximum of about 4% of the incident power (-14 dB). This is several orders of magnitude higher than the backscatter. These reflections are seen on the display as large spikes, and under certain conditions (no clipping in the receiver) the return losses at those locations can be measured.

If the fiber breaks by accident, then the end surface will in most cases not be connector-like (rectangular and well-polished), and the reflected power will be below 4%. Sometimes the fiber is loosely guided in a tube filled with oil,

which can act as an index matching fluid, so that nearly no end reflection occurs, but the backscatter drops to zero after the break.

To summarize, because of Rayleigh backscattering, the impulse response of the fiber contains information on the amount and location of losses. Distributed losses are fiber attenuation and nondistributed losses are splice, bend, or connector losses. Because of Fresnel reflections, or changes in the refractive index, the impulse response contains information on the location and return loss of nonphysical connections and the fiber end.

A more detailed mathematical description will be given in the following articles.

OTDR Design Objectives

The HP 8145A Optical Time-Domain Reflectometer (Fig. 1) is designed to make OTDR measurements with greater speed, convenience, and dynamic range than conventional OTDRs. Objectives for its design were based on analysis of the features an OTDR should have to make it a valuble and helpful tool.

In fiber manufacturing, it is mainly performance parameters that are are important. The instrument should be able to show the backscatter of the fiber under test as precisely as possible, operate under computer control, and allow users (in most cases highly knowledgeable engineers who understand the physics of fibers and OTDRs) to manipulate the measurement parameters to optimize the results.

On the other hand, when the OTDR is used in a field environment during acceptance testing or maintenance,
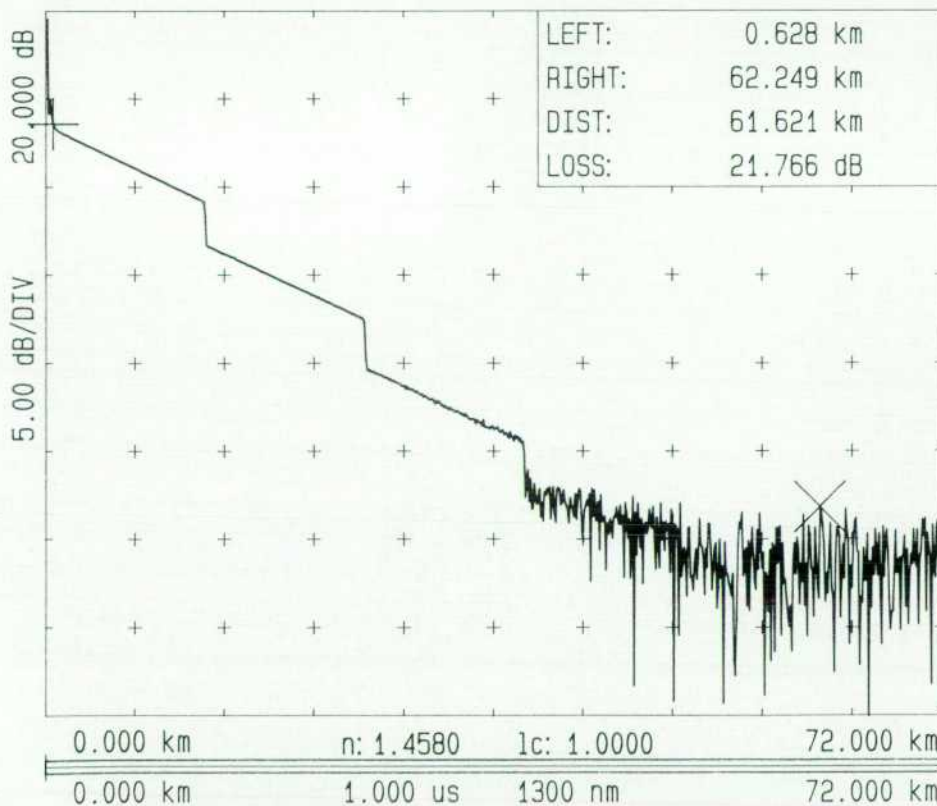
Fig. 3. A two-minute measurement on the fiber of Fig. 2 gives a dynamic range of 22 dB peak.

LEFT:	0.628 km
RIGHT:	62.249 km
DIST:	61.621 km
LOSS:	21.766 dB

20,000 dB

5.00 dB/DIV

| 0.000 km | n: 1.4580 | lc: 1.0000 | 72.000 km |
| 0.000 km | 1.000 us | 1300 nm | 72.000 km |

weight, power consumption, ease of use, ruggedness, wide temperature range, and short warm-up time, which are not very important in the factory or laboratory, become key issues. The instrument should be able to operate from batteries, and should contain as few as possible mechanical components, because these usually limit the ability to withstand harsh environments. Flexible disc drives as a storage medium for measured curves or thermal printers to generate hard copies are therefore not desirable.

HP 8145A Features

The HP 8145A is a high-performance, HP-IB programmable, optical time-domain reflectometer designed for both field maintenance and bench applications. It uses a data correlation technique (see article, page 14) that results in a dynamic range of 28 dB at 1300 nm, single-mode. This high dynamic range is measurable and displayable with only one measurement. Unlike conventional OTDRs, the HP 8145A does not require the user to patch partial measurement results together manually.

The correlation technique also drastically reduces the measurement time. The HP 8145A performs measurements much deeper in the fiber and up to 150 times faster than conventional OTDRs. It has a power resolution of 0.01 dB, a display resolution of 1 meter, and a maximum range of 200 km.

For field applications, the HP 8145A is lightweight and rugged. It can operate from batteries (12V to 30V) or ac power (90V to 260V). To eliminate the need for external data loggers or disc drives, a RAM memory (divided into an internal 12-curve memory and an external, plug-in, 114-curve module) serves as the storage medium for measurement information. When hard copies are needed, any measurement can be printed on an HP ThinkJet or QuietJet printer in about 30 seconds; no computer/controller is needed.

Two cursors are provided for distance and loss measurements. To use the internal 32-bit microprocessor to calculate splice loss automatically, up to five cursors can be activated. Zoom capabilities make it possible to expand sections of a long cable or fiber for closer examination. The cursor positions are not affected by zooming.

The HP 8145A provides probe pulse widths of 125 ns, 250 ns, 500 ns, 1 μs, 2 μs, 4 μs, and 8 μs. Wider pulses result in faster measurements, while narrower pulses give better resolution. With a 125-ns pulse and a 500-m display window, two fusion splices less than 40 m apart at a distance of 14 km can be resolved.

Any previously taken trace stored in RAM can be recalled and compared with the one being measured. This is useful for detecting inhomogeneities and attenuation changes like those caused by microbending during cabling. Traces at different wavelengths can also be compared. Two optional user-installable laser modules allow operation at 1300 nm or 1540 nm or both wavelengths. A low-loss wavelength division multiplexer ensures that the dynamic range is not affected by the number of installed laser modules.

Four exchangeable connector options and a variety of adapter cables are available.
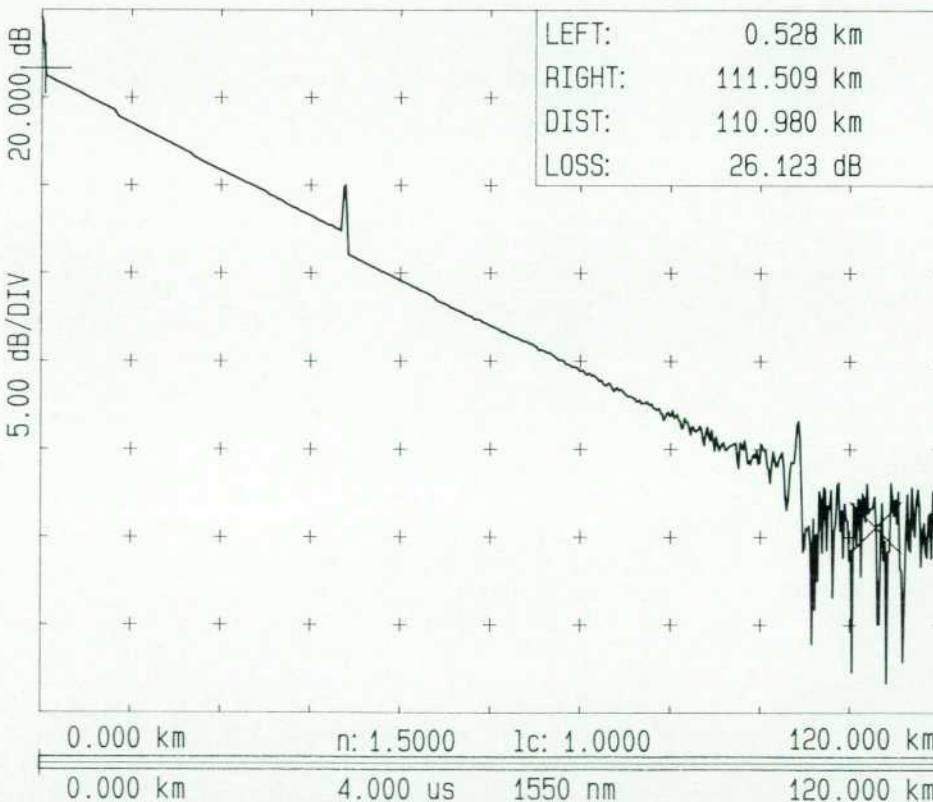

Fig. 4. *A fiber break at 102 km after a ten-second measurement.*

Measurement Performance

Figs. 2 and 3 show the backscatter curve of a 50-km-long fiber consisting of four 12.5-km segments. The splices between the segments have a loss of about 2.5 dB each. As demonstrated in Fig. 2, the HP 8145A's data correlation technique is able to give a dynamic range of 18 dB peak after only 10 seconds of measurement time. Fig. 3 shows that the dynamic range is extended to about 22 dB peak after another two minutes. These results are valid for a medium pulse length of 1 μs. Using a long pulse of 8-μs duration will lead to a dynamic range of more than 25 dB peak after only 10 seconds.

The effectiveness of the data correlation technique depends on various fiber characteristics such as Fresnel reflections and lossy splices. However, for fiber breaks, which typically have no Fresnel reflection, measurements can be done with the longest probe signals and measurement results can be obtained very rapidly. This is demonstrated by the measurement results given in Figs. 4 and 5. Fig. 4 shows a fiber break at a distance of 102 km after only 10 seconds of measurement time. Fig. 5 is for the same fiber, but with a measurement time of eight minutes.

Fig. 6 gives an estimate of the measurement speed required for long fibers used in high-speed transmission lines. This diagram shows the achievable dynamic range (signal to peak noise) as a function of the measurement time and the pulse width for several different long fibers at a wavelength of 1300 nm. For example, the HP 8145A achieves a dynamic range of 25 dB after only ten seconds of averaging, using an 8-μs probe pulse width. With con-
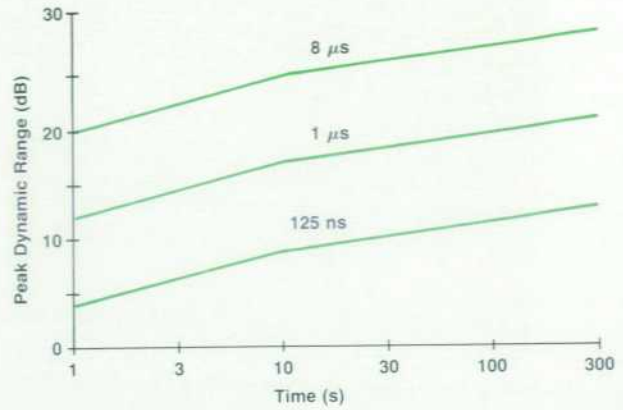
Fig. 6. *Dynamic range as a function of measurement speed for typical 1300-nm fibers used in high-speed transmission lines.*

ventional OTDRs, every doubling of the measurement time improves the signal-to-noise ratio by 0.75 dB. The slopes of the curves of Fig. 6 are steeper than this, showing the superiority of code correlation over single-pulse probing.

Two-point resolution is depicted in Figs. 7 and 8. Fig. 7 shows the measurement result after five minutes of averaging time. Two single mechanical splices 62 m apart at a distance of 37 km are clearly visible. Fig. 8 shows the result of a 30-second measurement of a series of fibers interconnected with fusion splices. The fiber lengths are 13.914 km, 74 m, 36 m, and 9 km. The short 36-meter-long piece
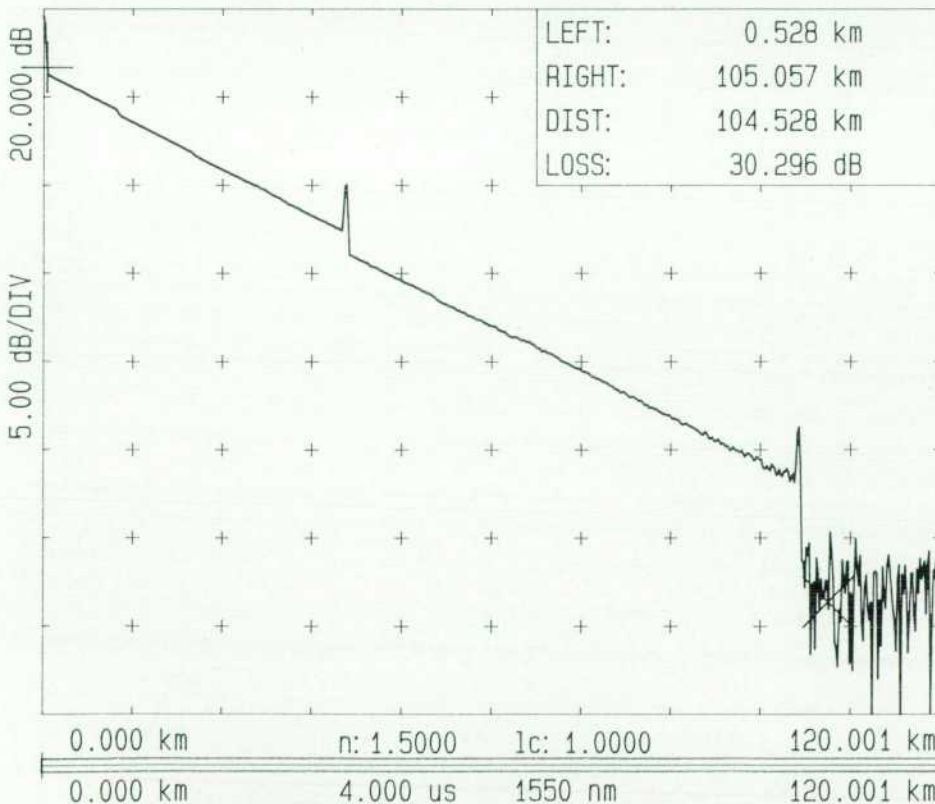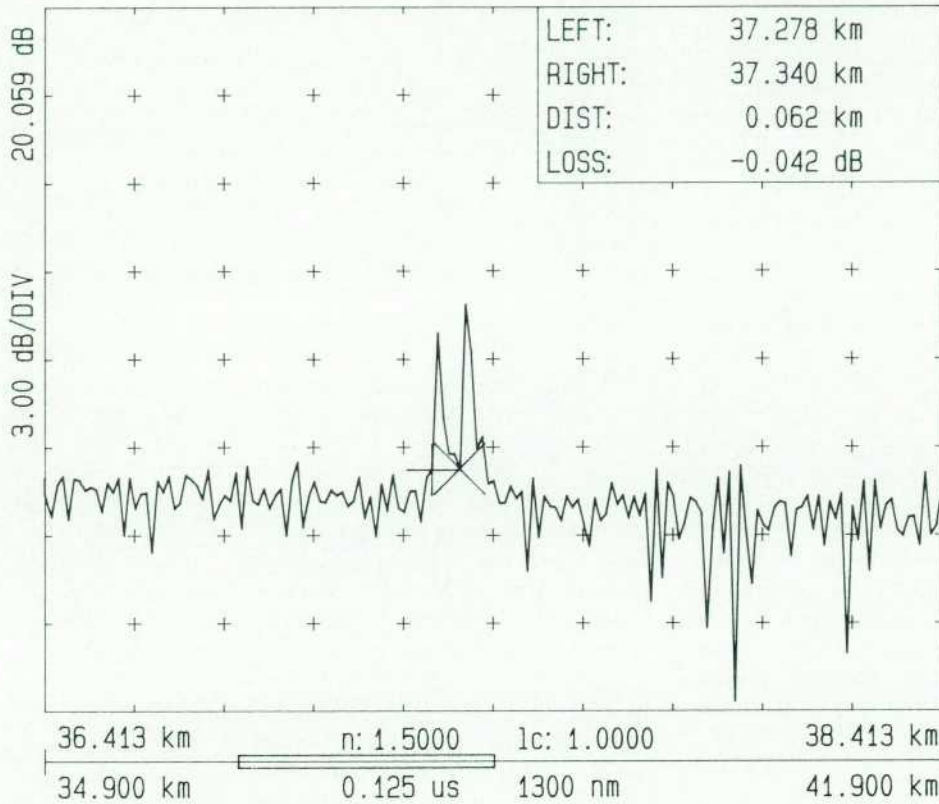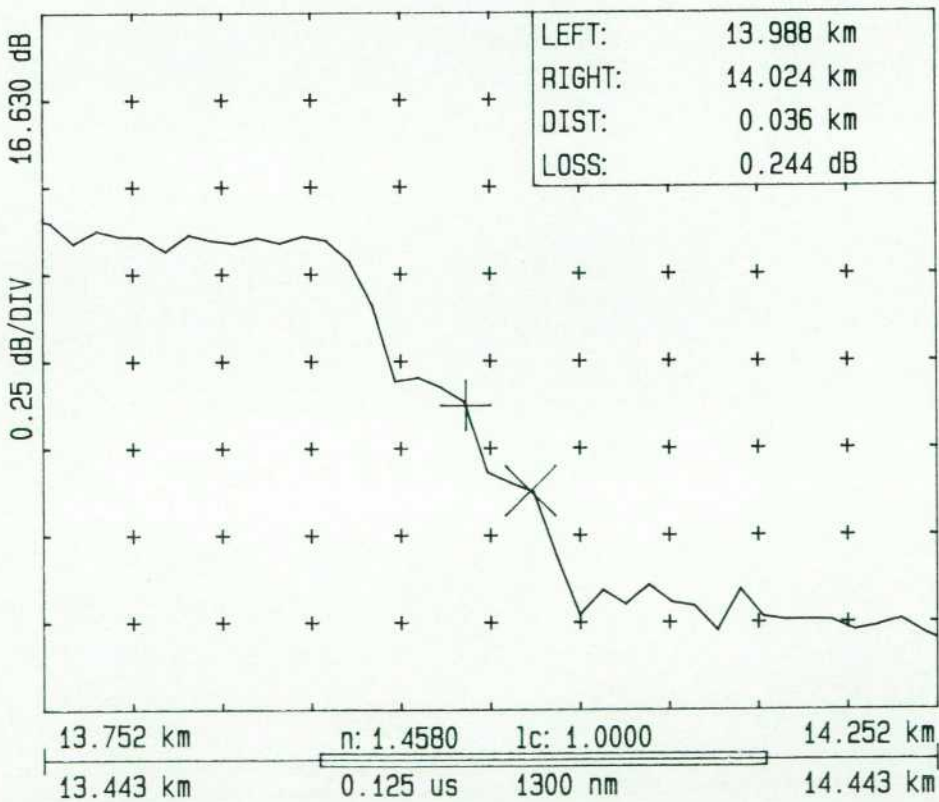
LEFT:	0.528 km
RIGHT:	105.057 km
DIST:	104.528 km
LOSS:	30.296 dB

0.000 km   n: 1.5000   lc: 1.0000   120.001 km

0.000 km   4.000 us   1550 nm   120.001 km

Fig. 5. *A fiber break at 102 km after an eight-minute measurement.*

Fig. 7. Two connectors 62 m apart at approximately 37 km. The measurement time was five minutes.

Fig. 8. A 30-second measurement of four fibers spliced together. The fiber lengths are 13.914 km, 74 m, 36 m, and 9 km. The short 36-m fiber at 14 km is easily seen.
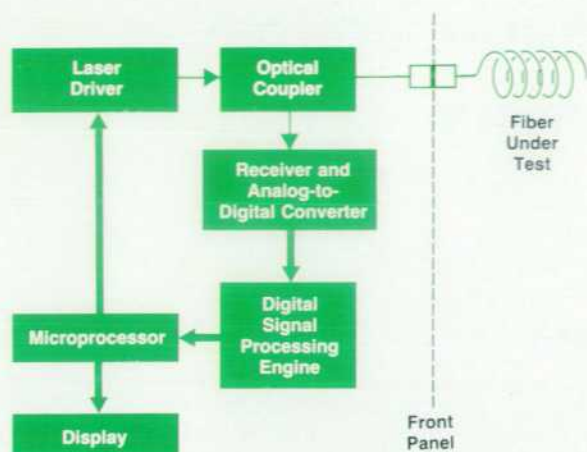
Fig. 9. *Block diagram of the HP 8145A Optical Time-Domain Reflectometer, which uses a correlation technique for improved dynamic range and measurement speed.*

of fiber at a distance of about 14 km is easily seen.

Because the backscattered signal in most fibers is far below the receiver noise, it is important to have data on the repeatability of fiber loss and splice loss measurements. This requires many measurements, because the effect of the noise on the measurement results is only statistically predictable. Our test data indicates that splice loss repeatability is better than 0.015 dB with a probability of 95%. Fiber loss readouts have an uncertainty of less than 0.015 dB/km with 95% probability. Both uncertainties are valid

for the HP 8145A's specified temperature range of −10°C to 55°C.

Architecture

The architecture of the HP 8145A OTDR is shown in the block diagram, Fig. 9. Details of the various blocks, the correlation technique, and the user interface and firmware are given in the other articles in this issue.

Acknowledgments

References

1. *Hewlett-Packard Journal*, Vol. 38, no. 2, February 1987, pp. 4-36.
2. C. Hentschel, *Fiber Optics Handbook*, Hewlett-Packard publication no. 5952-9654.
3. S.D. Personick, "Photon Probe: An Optical-Fiber Time-Domain Reflectometer," *Bell System Technical Journal*, Vol. 56, 1976, p. 355.

Complementary Correlation Optical Time-Domain Reflectometry

The autocorrelation function of a complementary Golay code pair has zero sidelobes, making these codes ideal for spread-spectrum optical time-domain reflectometry.

by Franz Sischka, Steven A. Newton, and Moshe Nazarathy

AN OPTICAL TIME-DOMAIN REFLECTOMETER (OTDR) is an instrument that characterizes optical fibers by launching a probe signal into a fiber under test and detecting, averaging, and displaying the return signal. The distance to a given feature is determined by measuring the time required for the signal to travel to the feature and back again to the measuring instrument.

A block diagram of a generic OTDR is shown in Fig. 1. An electronic probe signal generator is used to modulate the intensity of a laser, and the light output of the laser is launched into the fiber under test. In a conventional OTDR, the probe signal is a single square pulse. In field portable instruments, a semiconductor laser with an output power of several milliwatts is typically used as the source. The output of the laser is coupled into the fiber under test using a 3-dB fiber directional coupler. The unused output of this coupler is terminated by a method such as refractive index matching to prevent the probe signal from being reflected back into the receiver.

The return signal from the fiber under test is coupled to the receiver via the coupler. It is important that this beamsplitter be polarization independent, because polarization variations in the backscattered light would be indistinguishable from true changes in the overall power level of the return signal. After detection and amplification in the receiver, the return signal is processed, usually by averaging, before being transferred to the display.

A more detailed description of the optical components used in OTDRs and in other photonics applications is given in reference 1.

Display Calibration

The OTDR operates as a one-port instrument. The probe light travels out to a given point on the fiber and the return light travels back. Therefore, the return signal has been attenuated twice.

When observing the OTDR display, the user is only interested in the backscattering signature of the fiber under test and not in the details of the two-way propagation of the light that probes the fiber. For this reason, the horizontal (range) axis is calibrated to show the true one-way distance to a reflection or other feature despite the fact that the probe light has actually travelled twice this distance, that is, from the instrument to the feature and back again. For example, signals from two reflections that are 100 m apart arrive back at the OTDR separated in time by 1 μs. Simply converting this delay to distance using the speed of light

in fiber (c/n $\approx$ 200,000 km/s, where n is the refractive index of the fiber) would yield a separation of 200 m, which is the round-trip distance between the two reflections. The OTDR displays this interval as 100 m, which is the actual separation of the faults. In general, then, the distance Δz is displayed according to the formula:

$$\Delta z = (c/n)\Delta t/2.$$

Likewise, the vertical (power) axis of the display is calibrated to show the actual optical power levels divided by two (in dB), as if the probe light had propagated in only one direction.

Limitations of Conventional OTDRs

Several factors can limit the performance of conventional OTDRs. The most important of these is the fundamental trade-off between the signal-to-noise ratio (SNR) and the response resolution. Several key OTDR specifications, including the dynamic range and the measurement time, are really different manifestations of the SNR. The response resolution of a measurement is the minimum separation at which two different faults (reflections, splices, etc.) can be resolved.

The dynamic range of an OTDR refers generally to the ratio (in dB) of the largest signal level the instrument can handle to the smallest signal level it can detect. For the purposes of specification, dynamic range is usually defined as the ratio of the initial backscattering level (which itself depends on the probe signal energy and the type of fiber) to the rms noise level. Normally the one-way dynamic range is specified, that is, as if the probe light had only propagated in one direction. Before the introduction of the HP 8145A,
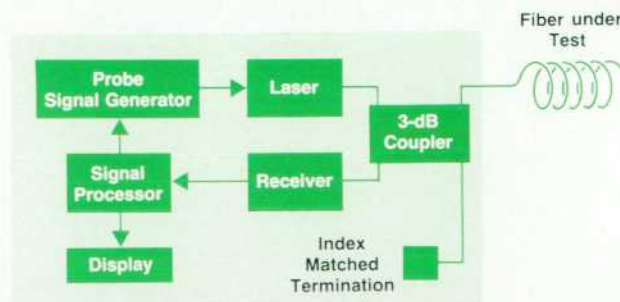

Fig. 1. *Block diagram of a conventional optical time-domain reflectometer (OTDR).*

the state-of-the-art one-way dynamic range of commercial OTDRs was 22 to 24 dB. The HP 8145A's one-way dynamic range is 30 dB.

These dynamic ranges are more impressive than they might initially appear because of the relationship between optical and electrical power. Since the detection process converts optical power to a proportional electrical photocurrent, a 3-dB change in optical power produces a 6-dB change in electrical power. Thus a 30-dB optical one-way dynamic range equals a 60-dB optical two-way dynamic range, which is equivalent to a 120-dB electrical dynamic range! The optical powers involved may range from 100 nanowatts (-40 dBm) down to 100 femtowatts (-100 dBm), with corresponding photocurrents of roughly 70 nanoamperes and 70 femtoamperes, respectively.

The Level Diagram

A helpful tool for visualizing the dynamic range of an OTDR measurement at a certain distance is the level diagram. It shows at a glance how the distance measurement capabilities depend on the pulse width , the signal-to-noise ratio, the influence of noise, and the magnitude of the Fresnel reflections relative to the Rayleigh backscatter level. An example is depicted in Fig. 2 using typical OTDR power levels.

Consider a laser that launches 8 mW into a single-mode fiber. This is equivalent to 9 dBm. Referring to the block diagram of Fig. 1, this level is then attenuated by 3 dB in the directional coupler. We therefore have a 6-dBm optical probe pulse in the fiber under test. Assuming a 0.35-dB/km fiber attenuation at a wavelength of 1300 nm, this probe pulse is attenuated to -36 dBm after 120 km.

However, the backscattered (Rayleigh) and reflected (Fresnel) signals that the OTDR must detect are orders of magnitude lower. For example, Fresnel reflections at glass-to-air interfaces are 4% of the incident power level, or 14 dB lower. Rayleigh scattering is 43 to 59 dB lower, depending on the duration of the probe pulse used. In addition, these signals suffer a round-trip propagation loss that corresponds to their round-trip propagation path from the OTDR. Therefore, in Fig. 2 they are plotted with a slope of 0.7 dB/km. Taking into account another 3-dB loss when the return light is split again in the coupler, we end up with a maximum power level at the receiver photodiode of -11 dBm for Fresnel reflections and -40 to -56 dBm for the maximum initial Rayleigh backscattered power using pulse widths of 4 μs and 100 ns, respectively.

Linearity and Noise

Even along segments of the backscattered signal where the signal-to-noise ratio is good, it is of the utmost importance that measurement linearity be maintained to ensure that the displayed data is valid. This means that the response of each of the component parts of the measurement—detector, amplifier, analog-to-digital converter (ADC), averager, and so on—must be linear over a wide dynamic range.

Assuming that all signals above the highest Rayleigh level are clipped (Fresnel reflections don't contain significant fiber loss information) and an 8-bit ADC is used, the instrument will have a 24-dB dynamic range between -40 dBm and -64 dBm.

Let's now consider how noise can effect this dynamic range. The two-way 24-dB dynamic range appears as only 12 dB on the OTDR's display. Even this would seem to assume that the receiver noise is less than or equal to the LSB (least-significant bit) of the ADC. This would mean having a receiver with a noise equivalent power (NEP) of -64 dBm.

The receiver noise level can be greatly reduced by averaging. This will extend the useful dynamic range provided that the ADC has a linear response, which it does not. However, there is a way to linearize the response of the ADC, as well as to extend this linear response well below the level of the LSB. The solution is to make use of the noise.
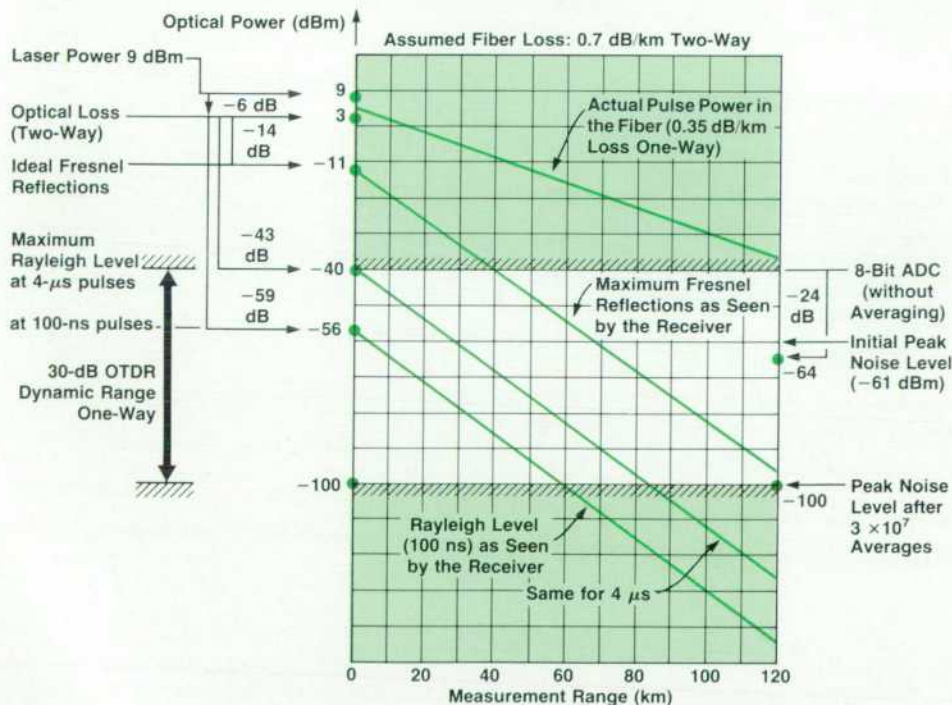

Fig. 2. An example of a level diagram for a typical OTDR.

This can be done by designing the receiver and ADC circuitry so the receiver noise extends over several ADC steps (each step equal to one LSB). After sufficient averaging is done, the well-known sawtooth error function of the ADC is smoothed to a sine function with drastically reduced amplitude. If the peak noise level is about twice the LSB, the remaining ADC linearity error is less than the LSB by more than 36 dB.

An example of this ADC error function reduction is given in Fig. 3. Here Gaussian noise is assumed, with a probability density function as depicted at the top. The level at which the ADC records the analog input is varied randomly from measurement to measurement by the receiver noise, which has a standard deviation denoted σ. The ADC error function is given in the large curve and shows the decreasing error as a function of the noise overlap (σ/LSB) over the steps of the ADC transfer function. Using these results and referring back to the level diagram of Fig. 2, we see that by choosing the LSB level to be half of the peak receiver noise, we are able to obtain a linear dynamic range of 60 dB two-way and 30 dB one-way, provided that enough measurements are averaged.

Measurement Time

Unfortunately, a conventional OTDR takes a long time to take enough averages to achieve this dynamic range. Even using a receiver that has an extremely low noise level, the majority of the dynamic range must be obtained by extensive averaging. For example, consider a conventional OTDR that transmits a single pulse every millisecond. If we begin with a receiver whose peak noise equivalent power is −61 dBm, the averaging of two separate measurements (requiring another 1 ms) will lower the noise by roughly 1.5 dB to −62.5 dBm. This is because the SNR is improved by the square root of the number of averages—in this case, by $\sqrt{2}$, or roughly 1.5 dB. Further reducing the effective noise from −62.5 dBm to −100 dBm, which is 37.5 dB or a factor of 5623, would require 5623 × 5623 × 1 ms = 31,618,129 ms of measurement time—about 9 hours!

In practice, the problem of noise is even more severe in that it is not sufficient simply to reduce the noise to the level of the signal of interest. The OTDR must detect changes in splice losses that are less then 0.1 dB, or variations of approximately 2% in the signal level. In fact, a signal-to-peak-noise ratio of 20 dB is required to ensure that the contribution of the noise will not perturb the backscattered signal by more than ±0.044 dB. In terms of the one-way display of the OTDR, this means that a ±0.022-dB variation of the signal will be caused by a peak noise level 10 dB below the signal at that point on the fiber. This limitation of amplitude accuracy by noise is shown in Fig. 4, which plots the noise-induced signal distortion (minimum splice loss detectability) versus the SNR (one-way) for a given point of interest on the curve.

Noise Improvement by Averaging

Clearly, the quality of an OTDR measurement is closely tied to the ability of the instrument to reduce the noise level by averaging. As mentioned previously, it can be shown that each time we double the number of averaged measurements, we improve the SNR by the square root of 2, or approximately 1.5 dB. The SNR (expressed in dB) of a measurement at a range z (km) can be simply expressed by the "OTDR maker's formula:"

$$\text{SNR(dB)} = P_{init} - 2\alpha z - \text{NEP} + 1.5 N_{oct},$$

where α is the fiber loss in dB/km, z is the range in km, NEP is the receiver noise equivalent power in dBm, N_{oct} is the number of probe shots (measurements) expressed in octaves (i.e., $N_{oct} = \log_2 N$, where N is the number of shots), and P_{init} is the initial value of the backscattered power in dBm. P_{init} can be approximately expressed by:

$$P_{init} \approx \text{constant} \times P_{in}T,$$

where P_{in} is the peak input power, T is the probe pulse width, and the constant factor is proportional to the fiber
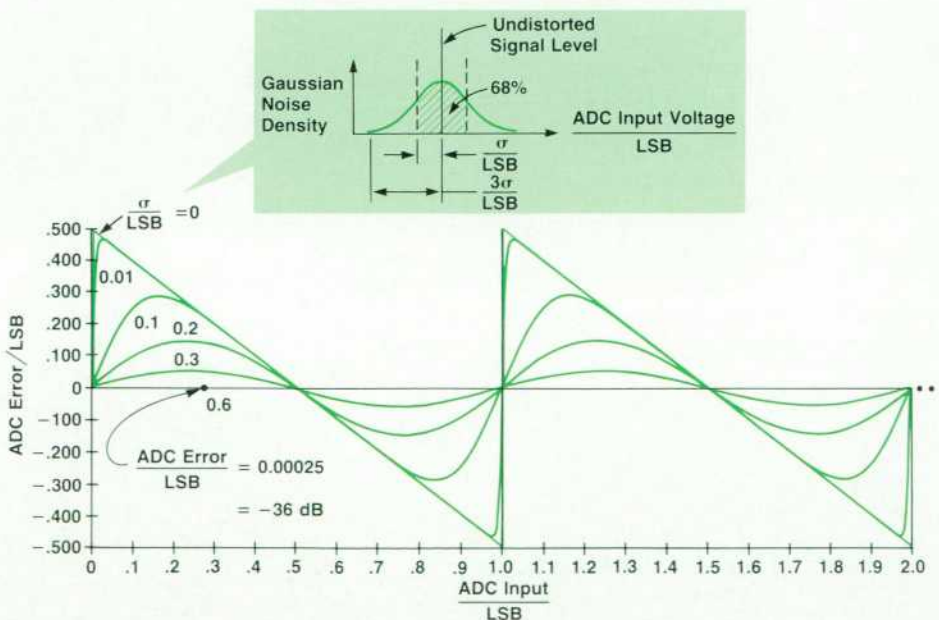

Fig. 3. Analog-to-digital converter (ADC) error as a function of noise on the analog signal. Gaussian noise (probability density at top) and an infinite number of averages are assumed. LSB is one ADC step or least-significant bit.

HP 8145A/2 Fig. 3

loss, the group velocity in the fiber, and the ability of the fiber to capture scattered light in the backward direction.

On the user display, this SNR (dB) is divided by 2, which leads to the relationship mentioned in the preceding article, that is, the displayed SNR improves by only 0.75 dB for each doubling of the measurement time. This is why it can seem that, after about 20 minutes of measurement time, very little additional SNR improvement is occurring.

It is easily seen in the above equations that increasing the number of averages directly contributes to an improved SNR, albeit at the expense of increasing the measurement time. We can also see that in a given measurement time, the only ways to improve the SNR are by increasing the laser power or by increasing the duration of the probe pulse.

Response Resolution

In any OTDR measurement, the received signal s(t) can be expressed as the convolution (*) of p(t), the signal that probes the fiber, f(t), the backscattering impulse response of the fiber, and r(t), the impulse response of the receiver.

$$s(t) = p(t) * f(t) * r(t)$$

The response resolution of the measurement is therefore degraded according to the durations of the receiver response and the probe signal (Fig. 5). In a conventional single-pulse OTDR, the probe signal is simply a square pulse having duration T. If $T = 1 \mu s$, for example, the two-point response resolution can be no better than approximately 100 meters, even with an ideal receiver. So strictly from the point of view of response resolution, it is desirable that the probe pulse width be as small as possible. On the other hand, the strength of the received signal is proportional to the energy of the probe signal, which is the product of the peak power and the probe pulse width. Once we reach the peak power limit of our laser source, our only means of increasing the probe energy (and thus the SNR) is to increase the duration of the probe pulse. But as we have just seen, this leads to degraded resolution. The trade-off between the SNR and the response resolution is therefore an important and fundamental limitation on the overall performance of a conventional single-pulse OTDR.
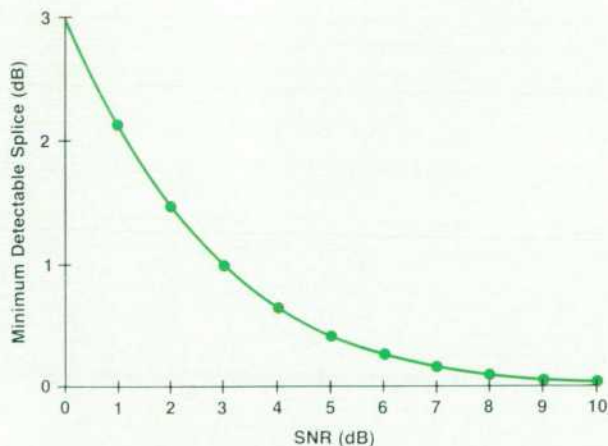
Design Alternatives

Faced with this performance trade-off, the HP OTDR design team was faced with several alternative means of building a high-performance long-haul OTDR. One alternative was to obtain special high-power laser sources. However, the price of such lasers was high and their availability questionable. Furthermore, these lasers require high drive currents which could lead to limited lifetimes and noise problems.

Another approach would be to use a high repetition rate to pack more measurements into a given time interval, increase the number of averages per second, and thereby reduce the noise more quickly. Unfortunately, the repetition period is limited by the round-trip time delay corresponding to the maximum range of the measurement. Decreasing the repetition period beyond this time can lead to ghost traces that are time-aliased from previous or subsequent probe shots. While at least one commercial instrument uses this approach anyway, we considered it unacceptable.

Most commercial OTDRs give in to the trade-off and use long pulses to maximize their dynamic ranges. In some cases, probe pulses 1 km in length are used. Even for a long-haul instrument, we felt that this was clearly not a good solution for the measurement problem we faced.

In the end, we decided to pursue a new and different approach that had never before been successfully implemented in a practical instrument. This required the implementation of several new techniques, including a spread-spectrum scheme that involves probing the fiber with coded sequences of pulses rather than a single pulse. The coded sequences have complementary autocorrelation properties. Subsequent processing of the return signals allows accurate recovery of the fiber response in a way that takes advantage of the extra energy injected into the fiber while maintaining the resolution that corresponds not to the width of the code sequence but to the width of only a single pulse—one bit—of the code sequence.

Correlation Technique
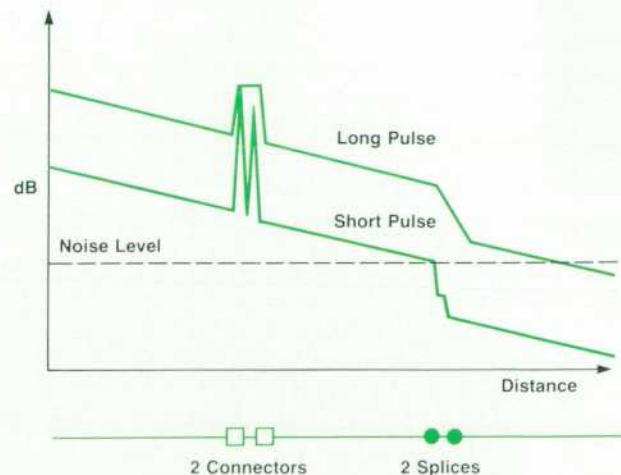
Spread-spectrum techniques such as correlation offer the

Fig. 4. *Minimum detectable splice versus the signal-to-peak-noise ratio (one-way).*

Fig. 5. *Differences in response resolution for different pulse widths.*

possibility of providing measurements with improved SNR without sacrificing response resolution. Such techniques are commonly used in radar and other peak-power-limited systems where increases in the transmitted energy would otherwise result in degraded resolution.

One way of applying correlation to OTDR measurements[2,3] is to correlate (⋆) the detected signal s(t) with the probe signal p(t):

$$s(t) \star p(t) = [p(t) * f(t) * r(t)] \star p(t) = [p(t) \star p(t)] * [f(t) * r(t)]$$

To the extent that the autocorrelation function of the probe signal [p(t)⋆p(t)] approximates a delta function, the fiber backscattering response f(t) can be accurately recovered, subject, as always, to the response of the receiver:

$$[p(t) \star p(t)] * [f(t) * r(t)] = \delta(t) * [f(t) * r(t)] = f(t) * r(t)$$

In this case, the duration of the autocorrelation function of the probe signal determines the response resolution (two-point resolution) and not the duration of the probe signal itself, which may be a long and energetic sequence.

An example of how correlation works is shown in Fig. 6. The autocorrelation function of code X is obtained by shifting an identical version of code X across itself by a certain number of bits, multiplying all overlapping time-slots and adding the partial results. This sum represents the value of the autocorrelation function for that particular delay. To obtain all of the other values of the autocorrelation function, code X is repeatedly shifted bit by bit and the procedure is repeated again and again until there is no further overlap of the fixed and shifted codes. In the example of Fig. 6, code X consists of the elements 1, −1, 1, 1 and the autocorrelation result is 1, 0, −1, 4, −1, 0, 1.

This autocorrelation function somewhat resembles a delta function. What is detrimental to the OTDR measurement are the nonzero values of the result for certain nonzero values of relative delay. If these values are too large relative to the central peak, these so-called autocorrelation sidelobes result in distorted OTDR measurements. All previous attempts to build practical OTDRs using correlation techniques failed because no finite coded probe sequence having sufficiently small autocorrelation sidelobes could be found.

Complementary Codes

The method described here realizes the full advantage of correlation by probing and correlating with pairs of probe signals that have complementary autocorrelation properties. These probe signals are the complementary codes first introduced by M.J.E. Golay in the late 1940s as a method of improving the performance of multislit spectrometers.[4] Golay codes have the property that if A and B are an L-bit complementary code pair, then
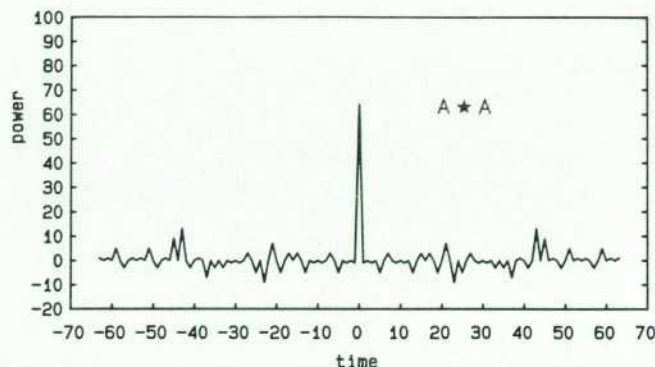
$$(A \star A) + (B \star B) = 2L\delta(t).$$
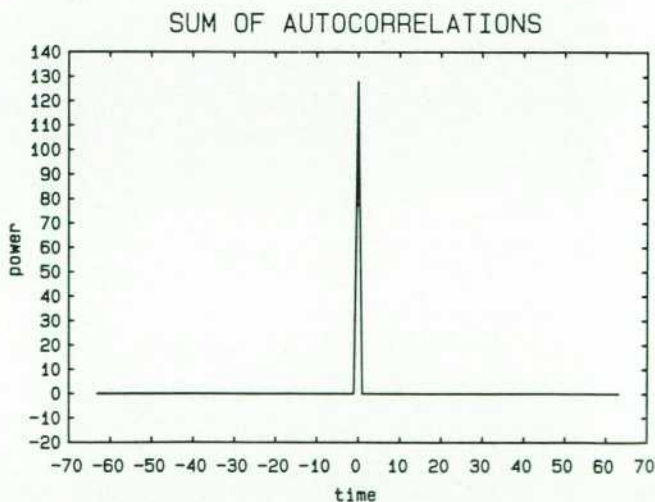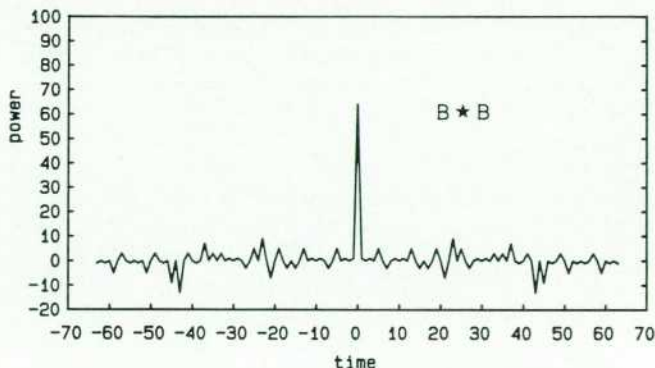
Fig. 7. The two upper plots show the autocorrelation functions of the two Golay codes of a complementary pair. The bottom plot shows the sum of the two upper plots.

Fig. 6. The correlation process.

The unique autocorrelation properties of the Golay codes are shown graphically in Fig. 7. The upper two plots show the individual autocorrelation functions of each member of a 64-bit complementary code pair. Each of the individual codes consists of a pattern of +1s and −1s. The value of each of the autocorrelation peaks is equal to the number of bits in the individual code. Each of the individual autocorrelation functions also exhibits sidelobes that are up to 10% of the peak height. However, when the autocorrelation functions are added, the peaks add to a value of 2L and the sidelobes cancel exactly.

It is this contribution of all of the bits to the autocorrelation peak along with the complete cancellation of the sidelobes that allows the correlation technique to work in practice. In designing a practical system of this kind, it is essential to work with an autocorrelation function that is perfect in principle, since finite sidelobes will always exist in a real, nonideal system. Using complementary codes, the sidelobes in a real system can be low enough so that the full advantage of correlation can be realized.

Probing the Fiber

Because the complementary Golay codes consist of the elements +1 and −1, while the quantity being modulated—optical power—is only positive (there are no negative photons), the bipolar codes are transmitted with a bias equal to half of the available peak power. The fiber is probed by successive groups of four probe shots, one shot for each member of an L-bit Golay code pair and its ones complement. Subtracting the measured backscatter signals in pairs causes the component of the signals generated by the bias to cancel, while the component generated by the codes is reinforced.

Signal Processing

The signal processing sequence is depicted in the block diagram of the HP 8145A OTDR in Fig. 8. Fig. 9 shows the partial results during signal processing as the final backscatter signal of the fiber is reconstructed.

Let's proceed through Fig. 9. A code generator establishes the complementary codes A and B, which are sent to the offset shifter and the correlator. The offset shifter drives the laser diode with 1s and 0s, thus generating the on-off modulation of the optical power. The modulated light proceeds through the coupler and into the fiber, generating a superposition of impulse responses because of the many pulses of the code. The reflected and backscattered light is amplified by the receiver and digitally converted.

For complementary Golay codes A and B, the probe light signals for the four shots are designated p_{A+}, p_{A-}, p_{B+}, and p_{B-}. The corresponding return signals are s_{A+}, s_{A-}, s_{B+}, and s_{B-}. The complementary nature of the fiber impulse responses is demonstrated in the two upper plots of Fig. 9, which show the responses s_{A+} and s_{A-} resulting from probe shots p_{A+} and p_{A-}.

These partial results are subtracted in the averager, yielding the signal $A_{received}$. In the same way the complementary responses are processed, yielding the signal $B_{received}$.

The next step is correlation. $A_{received}$ and $B_{received}$ are correlated with the codes A and B. The partial results after the correlation of both sequences are shown in the third row of Fig. 9. Notice that the two correlator output signals are complementary in terms of sidelobes and in the beginning of the backscatter curve. Finally, addition and logarithmic conversion lead to the OTDR display showing the fiber loss in dB plotted as a function of distance.

Correlation Gain

The effect of correlation on the performance of an OTDR is given by the "correlation OTDR maker's formula:"[2]

$$SNR = P_{init} - 2\alpha z - NEP + 1.5(N_{oct} + L_{oct})$$

where the parameters are the same as in the conventional OTDR maker's formula except for the addition of the term L_{oct}, which is the code length expressed in octaves (i.e., $L_{oct} = \log_2 L$). As with a conventional single-pulse OTDR, each time the number of averages is doubled the SNR is improved by 1.5 dB (0.75 dB on the display). In the case of the complementary correlation OTDR, however, the SNR
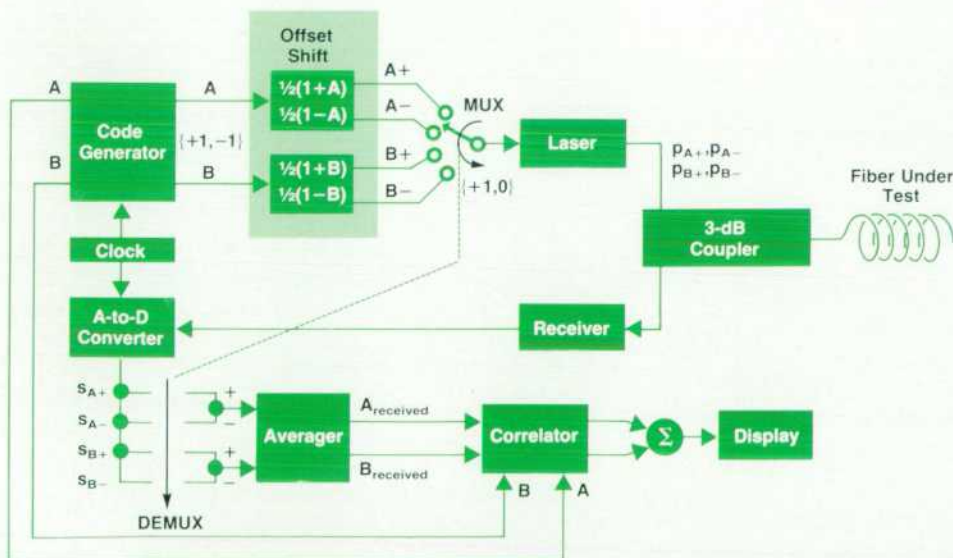

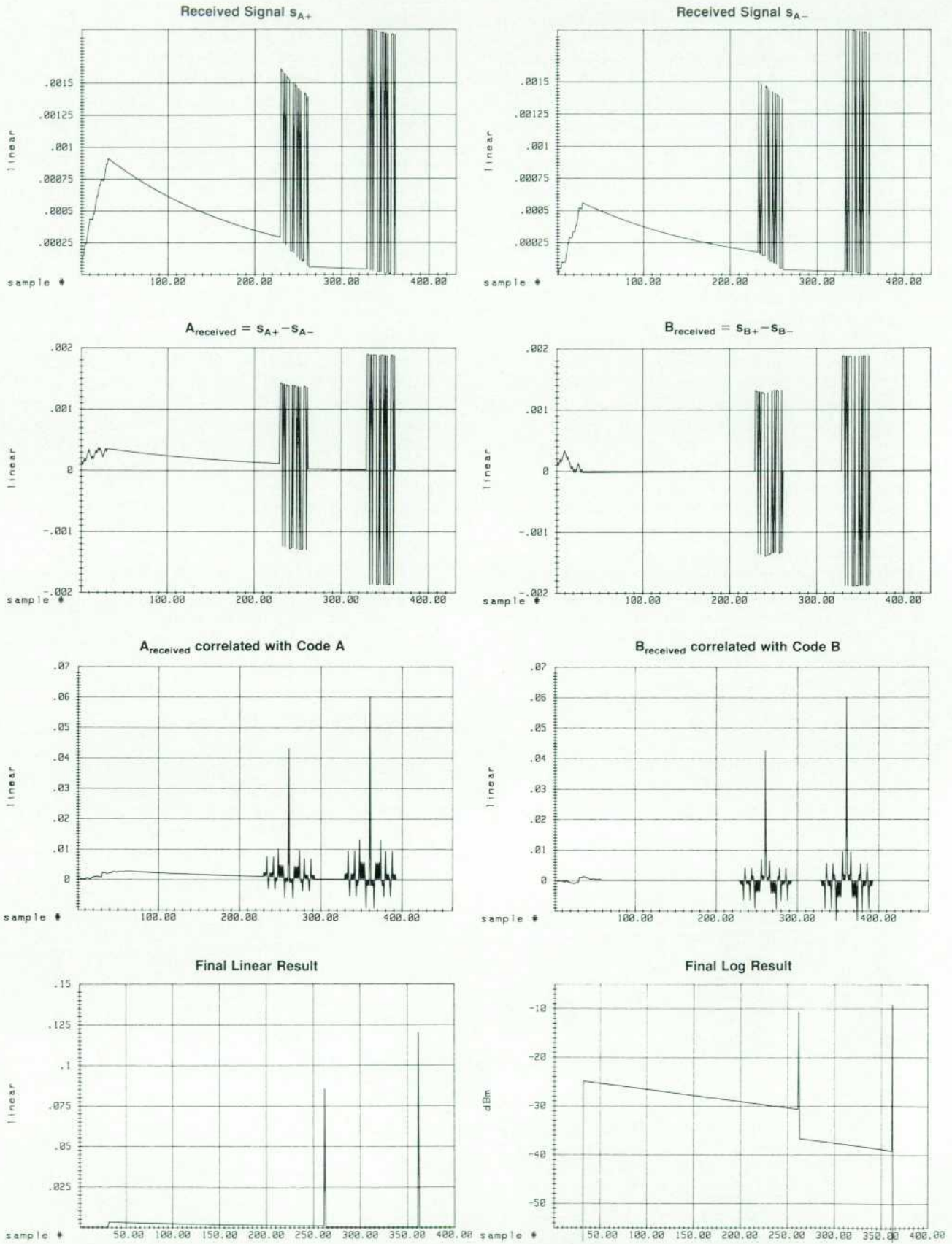**Fig. 8.** Block diagram of the HP 8145A Optical Time-Domain Reflectometer.

Fig. 9. *Received signals for a fiber under test, the A and B correlation functions, and the resulting measured impulse response of the fiber.*

improves by an extra factor of 1.5 dB for each octave of code length in the probe signal. This equation also explicitly shows that each bit of the probe signal effectively provides an extra average of the data, thereby reducing the measurement time.

Noise Considerations

It has been shown that, compared to a conventional single-pulse OTDR, the correlation technique effectively allows more averages. To make a fair comparison, however, one should take into account that the conventional probe signal can be sent out every measurement cycle, while the correlation OTDR needs four cycles to transmit all four probe shots into the fiber. Also, because the correlation OTDR transmits each pulse with a bias equal to half the available peak power, the peak pulse power is only half that of a conventional OTDR that has the same peak power capability. If the peak pulse power were the same, each correlation OTDR probe shot would provide a signal improvement equal to a factor of L, or 4L for four probe shots. Because the peak pulse power is halved, the improvement is 2L for four shots.

Like the conventional technique, the correlation technique is affected by receiver noise. For four shots the noise increases by a factor of $\sqrt{4L}$. Hence the correlation technique provides a signal improvement of 2L every four signal shots and a noise addition of $2\sqrt{L}$, for a net improvement of $\sqrt{L}$ (i.e., $1.5L_{oct}$ dB). The conventional technique provides a signal improvement of 4 for four shots and a noise addition of 2. Thus the SNR improvement of the correlation OTDR over a conventional one is:

$$\frac{\text{SNR(code of length L)}}{\text{SNR(4 single pulses)}} = \frac{\sqrt{L}}{2}$$

As a result, performance equivalent to that of a conventional OTDR is obtained in the correlation OTDR by using a code length of 4. Longer codes provide a performance advantage for correlation.

Performance Advantage

The performance advantage of the correlation OTDR can be seen in two ways. Assuming that the same laser and receiver are used, the correlation OTDR using codes 256 bits long achieves a 9-dB greater two-way dynamic range in a one-minute measurement of a 0.1-dB fault. This means the correlation OTDR can achieve the same sensitivity 9 dB farther into the fiber in a one-minute measurement. Depending on the fiber loss, this might correspond to a measurement range improvement of 10 to 15 km. Looking at the performance advantage in terms of measurement time, the same measurement range obtained with a conventional OTDR is obtained with the correlation OTDR 64 times faster. Thus a measurement requiring ten minutes of averaging using a conventional OTDR can be accomplished with a correlation OTDR in less than ten seconds.

The performance advantage of a correlation OTDR over a conventional single-pulse OTDR using the same laser, receiver, and other components can be summarized as follows. When coded probe signals consisting of L bits are used to probe a fiber, dynamic range is increased by a factor of $\sqrt{L/4}$ for a given measurement time, while maintaining the response resolution corresponding to the width of a single pulse. Alternatively, the result obtained with a single-pulse OTDR can be obtained a factor of L/4 faster using complementary codes. The resolution of a code correlation OTDR is exactly the same as that of the single-pulse OTDR, because the two-point resolution is only determined by the length of the code bits and not by the total code length.

A more detailed explanation of the performance advantage can be found in reference 5.

References

1. C. Hentschel, *Fiber Optics Handbook*, second edition, Hewlett-Packard Publication 5952-9654.
2. S. A. Newton, *A New Technique in Optical Time-Domain Reflectometry*, Hewlett-Packard Publication 5952-9641, 1987.
3. F. Sischka, "Introducing Spread-Spectrum Techniques to Optical Time-Domain Reflectometry," 1988, to be published.
4. M.J.E. Golay, "Complementary Series," *IRE Transactions on Information Theory*, Vol. IT-7, 1961, p. 82.
5. Moshe Nazarathy, et al, "Real-Time Long-Range Complementary Correlation Optical Time-Domain Reflectometer," to be published in *IEEE Journal of Lightwave Technology*, Vol. 7, no. 1, January 1989.

Optical Component Design for a Correlation-Based Optical Time-Domain Reflectometer

The major requirements for the laser driver, optical system, and receiver were single-mode, two-wavelength operation, high linearity, low noise, and low insertion loss.

by Jürgen Beck, Siegfried Gross, and Robin Giffard

THREE PRINCIPAL COMPONENTS of an optical time-domain reflectometer are the laser diode and its driver, the optical system, and the receiver, as shown in Fig. 1. The laser diode emits the probe light pulses. This light is guided and coupled into the fiber under test by the optical system. The reflected and backscattered light coming out of the fiber under test is split out of the forward ray path and guided to the receiver by the optical system. The output of the receiver, a sampled digital signal, is fed to the digital signal processor.

This article discusses the design of the laser driver, the optical system, and the receiver. The digital processor is discussed in the article on page 29.

Optical System

The key component of the optical system is the splitting device. The backscattered light can be split in various ways. Basically, three groups of splitting devices can be distinguished:

■ Bulk optical systems with passive components (e.g., splitting cubes with dielectric or metal coatings, or polarization prisms)
■ Bulk optical systems with active components (e.g., acousto-optic deflectors)
■ Fiber-guided systems (e.g., directional couplers).

The choice of a splitting technique depends on the requirements of the instrument. For the HP 8145A OTDR, two important requirements are single-mode operation and the two-wavelength capability (1300 or 1540 nm, upgradable to both). Additionally, the measurement process requires low insertion loss and very low polarization sensitivity. The cost and availability of the components and the complexity and difficulty of the assembly process are other important factors.

The optical system of the HP 8145A OTDR is based on a fiber-guided design. Fig. 2 shows a block diagram of the optical system. Fig.2a shows the basic version, operating at a wavelength of 1300 nm. The probe light emitted by the laser diode passes through a fiber connector to a 3-dB single-mode directional coupler. This coupler, built using the fused biconical taper technique, serves as a splitter for both wavelength bands, 1280 to 1320 and 1520 to 1560 nm. Half of the probe light is coupled to the fiber under test through the connector interface at the instrument front panel. The interface accepts adapters for the various connector types. The half of the probe light not coupled to the output of the wavelength independent coupler is lost in the second coupler output port, which is terminated to avoid backreflections into the system. The backscatter signal coming out of the fiber under test passes through the coupler in the backwards direction and is again split with a coupling ratio of 50% (3 dB). One half of this signal is conducted by a pigtail to a photodiode at the receiver input. The pigtail is connected to the wavelength independent coupler by a fusion splice.

The system is converted to dual-wavelength operation by inserting an upgrade subassembly between the 1300-nm laser diode and the wavelength independent coupler, as shown in Figs. 2b and 2c. In the two-wavelength version, the light waves of both laser diodes are combined into one fiber by means of a second directional coupler. In this case, the coupler has a wavelength dependent coupling ratio, which yields a high combining efficiency and low loss. This coupler is called the wavelength division multiplexer. The remaining ray path of the upgraded version is the same as in the basic version. Besides the enhanced optical system, the two-wavelength version requires an additional laser diode driver circuit for the 1540-nm laser diode. Fig. 3 shows the complete upgrade module including the optical system, the 1540-nm laser diode, and the driver circuitry.

Loss Mechanisms

The power efficiency of the optical system is characterized by its two-way insertion loss. The two-way insertion loss is the loss in the forward direction between the laser diode pigtail and the the fiber under test, plus the loss in
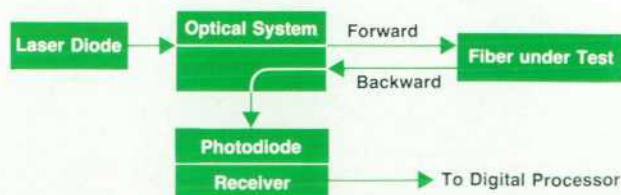
Fig. 1. *Principal components of an optical time-domain reflectometer.*

the backward direction between the fiber under test and the pigtail of the photodiode. Fig. 4 shows typical two-way insertion loss and its origins for the basic and two-wavelength versions of the optical system.

The main loss component is the coupling loss of the wavelength independent coupler, that is, the amount of light that is lost in the terminated end of the coupler. Since the coupler is used bidirectionally, a broad tolerance (±8%) for the coupling ratio can be specified. The difference in two-way coupling loss for a 50% coupler and a 42% coupler is only 0.1 dB, because the backward coupling ratio is simply 100% minus the forward coupling ratio.

Besides coupling loss, the wavelength independent coupler also exhibits excess loss, which is the light that is lost to the surroundings within the coupling region.

For the wavelength division multiplexer, an overall insertion loss is specified. It includes coupling loss, excess loss, and variations in coupling ratio caused by polarization effects. The wavelength division multiplexer has a high coupling ratio for 1300 nm and a low coupling ratio for 1540 nm. The specified insertion loss is 0.5 dB maximum for both nominal wavelengths.

Other losses in the optical system are caused by connections between two fibers. Fig. 4 shows typical values. The fusion splice at the photodiode has almost no loss, since this is a single-mode/multimode transition. The most critical connection is the connector interface at the instrument front panel, because it is accessible to the user and frequently opened. A bad connection here not only causes extra loss, but can also produce strong reflections, which reduce the performance of the instrument. The user should therefore be aware of the need to keep the optical interface of the OTDR in good conditon.

Precision Laser Driver Circuit

As explained in the article on page 14, the cancellation of the correlation sidelobes is only exact when the entire
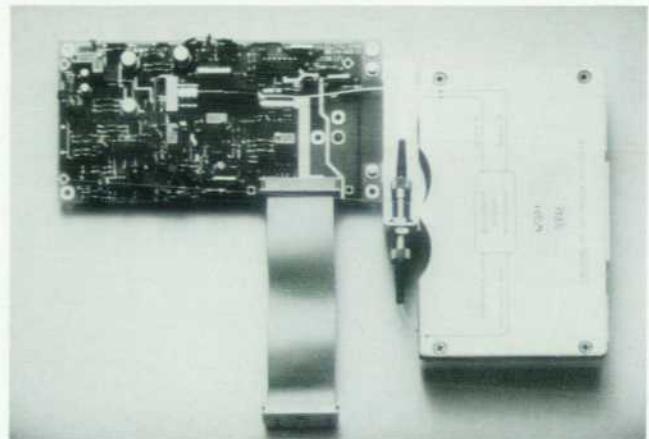
Fig. 3. User-installable laser module (1550 nm shown).

measurement system is perfectly linear and the transmitted pulses are of uniform amplitude. This is especially challenging for the design of the laser driver circuit because of the temperature-dependent nonlinear relation between optical power output and the driving current of the laser. The problem was to build an optical signal source with very high amplitude accuracy during the maximum-length code sequence. This means an error of less than 1000 ppm during any 128-μs interval.

The main contribution to sidelobes is the decay of the optical power during a code sequence. The reasons for this decay are the poor efficiency and the temperature sensitivity of semiconductor lasers. Only about eight percent of the electrical power driving the laser is transformed into optical power. The remaining 92 percent heats up the laser chip. Since the efficiency of semiconductor lasers has a negative temperature coefficient (less power at the same driving current for a higher temperature), the optical power decays as the laser is heated. Sidelobes generated by this
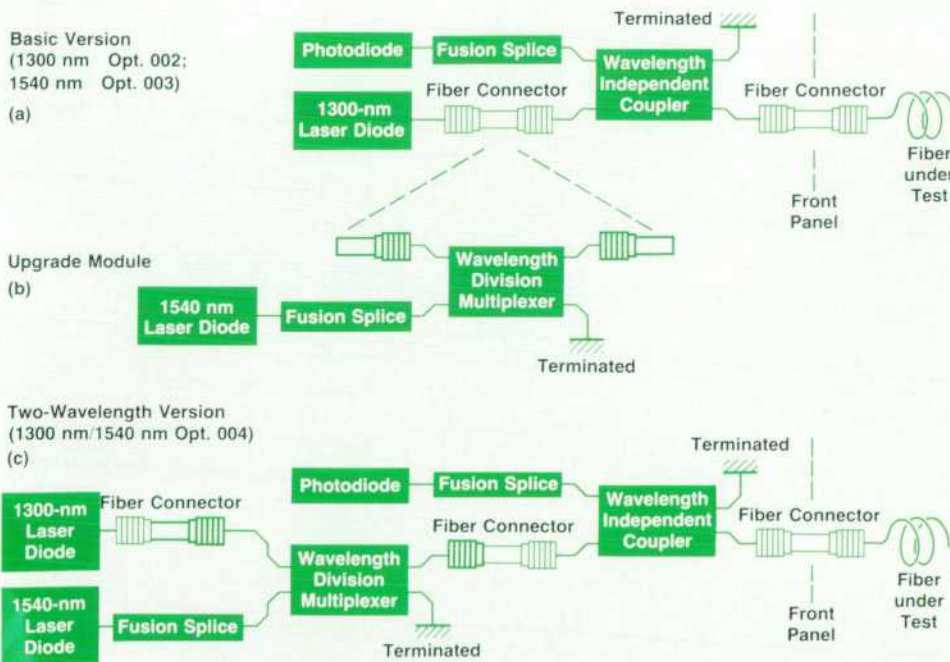
Fig. 2. Optical system of the HP 8145A Optical Time-Domain Reflectometer.
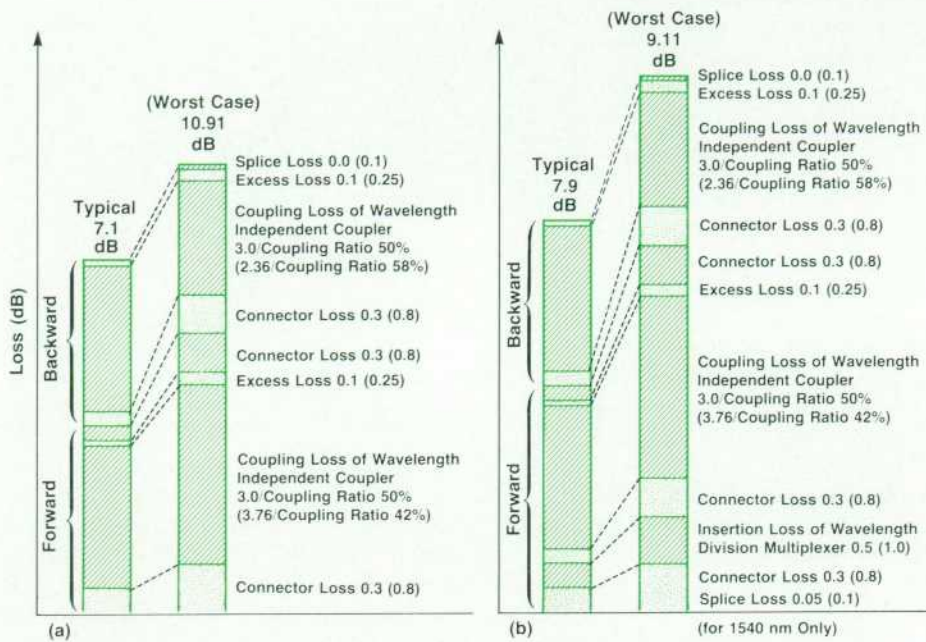
Fig. 4. *Two-way insertion loss of the HP 8145A optical system at 1300 nm. (a) Basic version. (b) Two-wavelength version.*

effect will disturb the backscatter signal in the neighborhood of a high Fresnel reflection. They extend for a distance of $\pm (L/2) \times t_{bit} \times c$ at both sides of a peak, where c is the velocity of light, L is the code length in bits, and t_{bit} is the pulse width of one bit of the code.

With an uncontrolled laser diode, a peak-to-sidelobe ratio of less than 25 dB was found to be typical, corresponding to a decay of two percent during a 128-μs code sequence. Computer simulations show that it is necessary to have a peak-to-sidelobe ratio greater than 35 dB for good system performance.

It is not possible to overcome this problem with a thermal control loop (e.g., with a Peltier cooler) because the thermal time constants are too large in this application. The only possibility is to provide a control loop with the built-in backfacet diode of the laser as the feedback source.

In a control loop a new problem arises. For OTDR applications, the on/off ratio of the laser power has to be better than 60 dB, so the laser cannot be biased at the threshold current. The time it takes to overcome the threshold current when switching from the off state to the on state appears as dead time in the control loop, and can easily cause instabilities.

It is important to avoid any overflow in the analog-to-digital converter (ADC), because clipping is a strong nonlinearity and generates very large sidelobes. In the HP 8145A, this requires that it be possible to attenuate the laser power by 7.5 dB. Thus a further difficulty is that the amplitude accuracy has to be the same for very different power levels.

Laser Diode Selection

To overcome these difficulties it is first of all necessary to find a laser diode that is appropriate to this special application. This means that the laser diode needs to have a built-in backfacet diode to provide feedback for controlling the optical output power. The laser diode should also be able to handle continuous-wave power because of the relatively high duty cycle of the code signal. The chip

should be mounted on a copper block with very good thermal contact to the environment to be sure that the uncompensated decay error is relatively small.

It is important not to use a laser diode mounted on a Peltier cooler, because this kind of thermal control system has a good thermal coupling ratio to ambient only when the temperature changes are slower than the cut-off frequency of the Peltier control system. Beyond that frequency, thermal contact becomes inadequate.

For the HP 8145A, a CW laser diode was selected for the
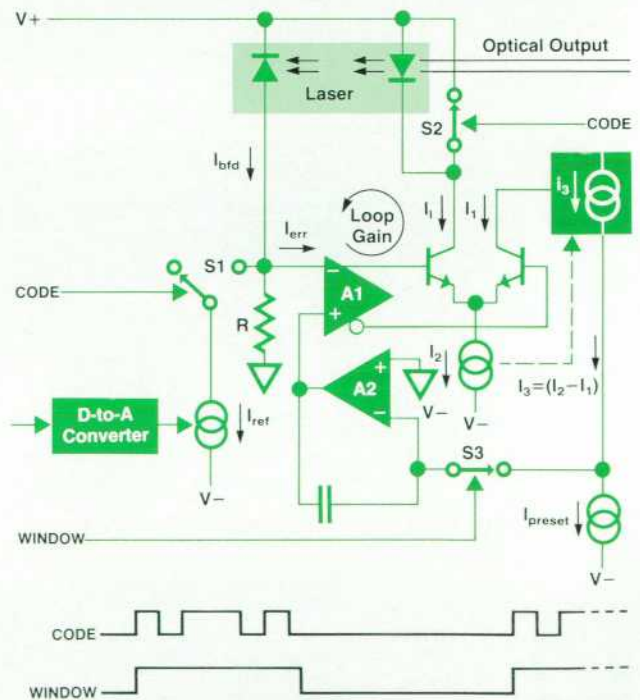
Fig. 5. *Simplified diagram of the laser driver circuit. Open-loop gain is 20.*

1300-nm source. This laser is specified for high-power pulse applications, and is also available at 1540 nm in the same housing and with very similar characteristics.

Control Loop

To reduce the decay of the laser output power, an electronic control loop is built around the laser using the backfacet diode as feedback source. Fig. 5 is a simplified diagram of the driver circuit. It is well-known from control theory that errors in a controlled system are inversely proportional to the open-loop gain. Therefore, to reduce the decay of 2% for a 128-μs code duration to an acceptable value of 0.1%, an open-loop gain of 20 is required.

The responsivity r_{bfd} of the backfacet diode and the efficiency g_{laser} of the laser diode have very small values (typically, $r_{bfd} \times g_{laser} = 0.004$), so it is necessary to have high-gain stages in the loop to achieve a loop gain of 20. In the circuit of Fig. 5, this gain is delivered by the voltage amplifier A1, the transconductance of the differential current source, and the input resistor R. The total open-loop gain is given by:

$$A_{loop} = A_1 \times \frac{S}{2} \times g_{laser} \times r_{bfd} \times R,$$

where A_1 is the gain of A1 and S is the transconductance of the differential current source. Thus the loop gain can be adjusted by choosing the value of R.

We also know from control theory that circuit stability decreases with increasing open-loop gain. This effect can be compensated by making the loop slower. For an optimum compromise between speed and stability, the cutoff frequency of the amplifier A1 is found to be 1 MHz.

The current source I_{ref} and the switch S1 make it possible to switch the laser from $P_{out} = 0$ mW to a power level defined by I_{ref}.

Assuming that the noninverting input of A1 is connected to ground and S1 is opened, the loop forces the laser to $P_{out} = 0$ mW because the inverting input of A1 has to have the same potential as the noninverting one, and this is

possible only if $I_{bfd} = 0$ mA.

If S1 is closed by the CODE signal, I_{ref} is fed to the input of A1. For the same reason as before, $I_{err} = I_{bfd} - I_{ref}$ has to be zero. This is only possible if I_{bfd} equals I_{ref}. Therefore, I_{ref} forces the laser to a certain power level P_0. To make the output power programmable, the reference current is controlled by a digital-to-analog converter.

Because I_{ref} is constant, I_{bfd} is also held constant by the loop, and the loop corrects the optical decay by increasing I_l slightly during a code sequence.

Presetting

Increasing the loop stability to an acceptable value by reducing the cutoff frequency of A1 to 1 MHz makes the pulse width of the optical signal strongly dependent on the desired power level because of the time the amplifier needs to increase I_l to a value above the threshold current. This effect adds more dead time to the control system, leading to poor stability.

Therefore, a presetting circuit consisting of A2 and the surrounding components in Fig. 5 was added to the laser driver. This circuit is a second control loop that has a very low cutoff frequency (0.1 Hz). During the off state of the laser driver (CODE = low) the presetting circuit controls the current I_l by comparing $I_3 = (I_2 - I_1) = I_l$ with a current source called I_{preset} so that A2 forces I_l to be equal to I_{preset}.

An electronic switch S2, in parallel with the laser, is closed when CODE is low so that the current I_l flows through S2.

Now if a code sequence is started, the WINDOW signal (which builds a frame over CODE) switches the integrator A2 into the hold mode. S2 is opened very quickly by the CODE signal so that the laser is almost immediately driven by $I_l = I_{preset}$. Since I_{preset} is larger than the threshold current, there is a very small rise time to some small power level. At this point the loop takes over and increases I_l until the desired optical power is reached.

The presetting circuit solves the problem of the amplitude dependency of the pulse width and reduces the dead

Ch. 1	=	100.0 mvolts/div	Offset	=	100.0 mvolts
Ch. 2	=	4.000 volts/div	Offset	=	4.120 volts
Timebase	=	5.00 usec/div	Delay	=	18.2000 usec
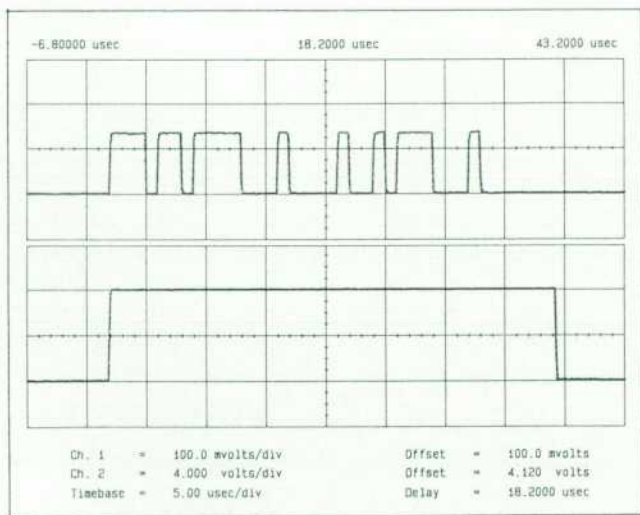
Fig. 6. Oscilloscope display of typical CODE (top) and WINDOW (bottom) signals.
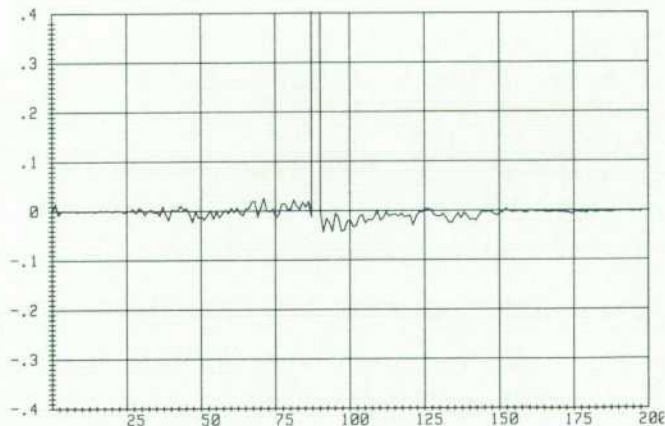
Fig. 7. Autocorrelation function of the HP 8145A OTDR for a 64-bit code. The vertical scale is in least-significant bits of the analog-to-digital converter. The peak is at 200 ADC LSB units. The horizontal scale is in sample points. The peak-to-sidelobe ratio is 37 dB.

time to a negligible value.

Laser Driver Performance

The performance of the laser driver circuit is depicted in Figs. 6 and 7. Fig. 6 shows a two-channel oscilloscope display of a code of length L = 32 and pulse width t_{bit} = 1 μs at channel 1 and the WINDOW signal at channel 2.

Fig. 7 shows the correlation output of the OTDR system for a code of length L = 64. This picture is vertically scaled in least-significant bits of the ADC. The value of the peak is at 200 ADC LSB units. The horizontal axis is scaled in sample points. It can be seen that a peak-to-sidelobe ratio of better than 37 dB is achieved. This is enough to ensure very good correlation results.

The smallest pulse width the laser driver is designed for is 125 ns and the longest code has 256 bits.

High-Performance Optical Receiver

The function of the optical receiver is to detect the backscatter signal and other reflections from the fiber. The output of the receiver is a sampled digital signal, which is fed to the digital signal processor.

The most critical specification imposed on the optical receiver is the noise level required to achieve the specified range using a given forward pulse energy and averaging time. To avoid degrading the range resolution, the bandwidth of the receiver must be adequate to reproduce the shortest transmitted pulse length. Because correlation is used to process the output of the receiver, overall linearity must be maintained over a wide dynamic range. Finally, the complete receiver chain must recover to the noise level as quickly as possible after overload.

The receiver operates in a noisy digital environment, so complete shielding is necessary. Careful attention must be given to the handling of power supplies and grounds, particularly at the analog-to-digital converter.

Input Stage Sensitivity and Noise

The choice of a pin diode optical sensor was made in view of the dual-wavelength requirement of 1300 and 1540 nm and the modest system bandwidth of 4.0 MHz. Since a pin diode has no electrical gain, the amplifier following it must have the lowest possible noise. A junction FET is used as the first stage. As shown in the box on page 27, the pin diode must have good quantum efficiency and low self-capacitance. To maintain the noise level over the working temperature range of the instrument, the dark current of the diode must be small. Fig. 8 shows the measured noise of the receiver in the form of a log-log plot of equivalent optical noise power per root-Hz as a function of frequency. The low-frequency spectral density is determined by the value of the feedback resistor in the transimpedance amplifier, and the increase at high frequencies is caused by the voltage noise of the FET.

Because of the increase in the input noise spectral density at high frequencies, the output of the receiver must pass through a suitable anti-aliasing filter before reaching the ADC. If this is not done, noise from outside the useful bandwidth will be added to the true noise in the digitization process. Since the receiver gain falls off quickly, a simple anti-aliasing filter is adequate. In choosing the filter characteristic, ringing must be avoided since the time-domain performance of the complete receiver is significant.

When the OTDR transmitted pulse is long, the effective bandwidth of the receiving chain can be reduced, thereby reducing the noise level. To avoid the use of switched filters, the bandwidth reduction is performed by a simple real-time digital filtering algorithm operating on the ADC output. The procedure consists of decimation in time by powers of 2, and the resulting signal-to-noise ratio is quite close to the ideal value that would be obtained with an optimal filter.

Transimpedance Amplifier

Fig. 9 shows a simplified diagram of a transimpedance amplifier. The pin diode output current is represented by the current generator I_d, and capacitor C represents the sum of the diode capacitance and the total input capacitance of the amplifier. The amplifier is assumed to have a high input impedance and gain −A, and R is the feedback resistor. Elementary circuit analysis shows that the output voltage V_2 is given by:

$$V_2 = -I_d R'/(1 + as), \qquad (1)$$

where s is complex frequency,

$$R' = AR/(1 + A), \text{ and } a = R'C/A. \qquad (2)$$

A characteristic of this configuration is that the maximum input current for linear operation is simply given by the maximum output voltage of the amplifier divided by the value of R'. The frequency response of the circuit
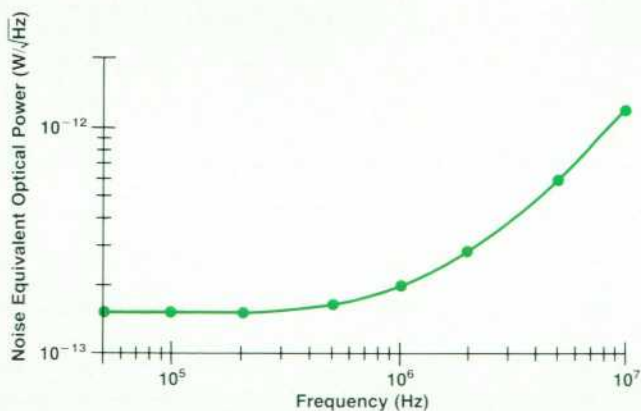
Fig. 8. *Measured noise of the HP 8145A OTDR optical receiver.*
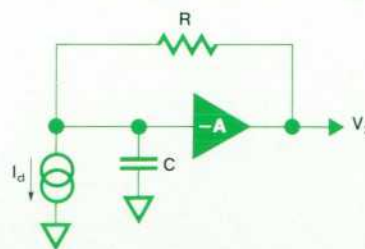
Fig. 9. *Simplified transimpedance amplifier.*

Signal-to-Noise Ratio for Detection Using a PIN Diode

Fig. 1 shows an equivalent circuit for calculating the signal-to-noise ratio resulting when a pin diode is coupled to a FET amplifier. The voltage V developed across the resistor R by the signal current I_s and the noise current I_n is detected by the amplifier which shows equivalent voltage noise E_n.

If power P_s at wavelength λ falls on the diode, the maximum possible output current is equal to $P_s(e\lambda/hc)$, where e is the electronic charge, h is Planck's constant, and c is the velocity of light. In practice the output current is reduced by a factor η, the quantum efficiency of the diode. Thus the signal current I_s is given by:

$$I_s = P_s(e\eta\lambda/hc). \tag{1}$$

The noise current I_n contains shot noise components resulting from the signal current I_s, the diode dark current I_d, and the amplifier input leakage current I_l. Thermal noise in the resistor R can be represented by a further noise current. The spectral density (single-sided) of the total noise current is given by:

$$S_i(f) = 2e(I_s + I_d + I_l) + 4k_BT/R, \tag{2}$$

where the last term is for the thermal noise. T is the temperature of R and k_B is Boltzmann's constant.

The voltage noise in the amplifier is represented by the noise generator E_n. In the case we are considering, the first stage of the amplifier is a FET and the spectral density $S_v(f)$ of E_n is given by:[1]

$$S_v(f) = 4k_BT(0.7/g_m), \tag{3}$$

where g_m is the transconductance of the transistor and T is the chip temperature. (In general, there is also an induced gate noise current, but this is insignificant in this circuit.) We have made the reasonable assumption that 1/f noise is not important in the frequency band of interest.

The voltage at the input terminals of the amplifier is given in the frequency domain by:

$$V = E_n + R(I_s + I_n)/(1 + j\omega RC) \tag{4}$$

To express the total noise in terms of optical power falling on the diode, we first find the value of I_e, the equivalent signal current
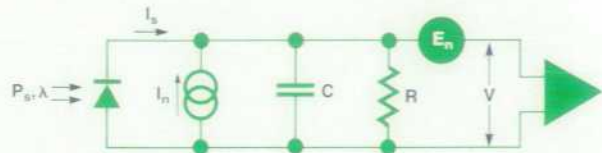
Fig. 1. *Equivalent circuit for calculating the signal-to-noise ratio for a pin diode coupled to a FET amplifier.*

amplitude that would reproduce the same value of V when substituted into equation 4 with E_n and I_n set to zero. This current is given by:

$$I_e = E_n(1 + j\omega RC)/R + I_n \tag{5}$$

Using equation 1 to express I_e in terms of an equivalent optical noise power P_e with spectral density $S_p(f)$, and substituting uncorrelated spectral densities into the Fourier transform equation (5), we obtain (when $g_mR >> 1$ and $RI_s << 50$ mV):

$$S_p(f) = (hc/e\eta\lambda)^2(4k_BT/R + 2e(I_d + I_l) + (\omega C)^2 (2.8k_BT/g_m)) \tag{6}$$

It is usually possible to choose components so that the shot noise term in equation 6 is negligible. The small-signal noise spectral density expressed as input optical power squared then contains just two components, a frequency independent component caused by the thermal noise in R, and a component proportional to frequency squared caused by voltage noise in the amplifier. The frequency f_t at which the spectral density begins to rise is given by:

$$f_t = (1/2\pi)(g_m/0.7RC)^2. \tag{7}$$

It is important to realize that the arguments given above are equally true when R is the feedback resistor in a transimpedance amplifier. The application of negative feedback does not change the noise referred to the input of the diode, as long as the overall gain reduction does not cause the noise of a later stage to become significant.

Reference
1. F.N.H. Robinson, *Noise and Fluctuations*, Clarendon Press, Oxford, 1974.

is characterized by a real pole at an angular frequency of a^{-1}. Equation 2 shows that the bandwidth of the circuit can be increased by increasing the amplifier gain. For example, suppose that a bandwidth of 5.0 MHz is required with R = 2 MΩ and C = 10 pF. Substitution shows that an amplifier gain of 630 is required. In practice the amplifier must not have a phase shift of more than about 30 degrees at the upper cutoff frequency, which at this gain represents something of a challenge.

In the final design of the receiver for the HP 8145A, the problem of implementing the amplifier was reduced by designing the first stage using active bootstrapping. This technique reduces the effective total capacitance at the input node by more than an order of magnitude, reducing the amplifier gain requirement by the same amount.

Another problem involved in realizing a bandwidth of 5.0 MHz is presented by the self-capacitance of the feedback resistor. A simple calculation shows that with the desired feedback resistor value of 2 MΩ in the configuration shown in Fig. 9, the maximum acceptable capacitance is 0.016 pF. This problem is addressed by introducing a lag network in the feedback path as shown in Fig. 10. If R_1 is much smaller than R, the effect of the stray capacitance C_s is exactly compensated when $C_1R_1 = C_sR$.

Another advantage of the transimpedance configuration is minimum sensitivity to voltage offsets in the first stage. However, since very stable baselines are required in this application, a programmable autozero circuit is incorporated using a track-and-hold circuit.

The low-frequency response of the complete receiver
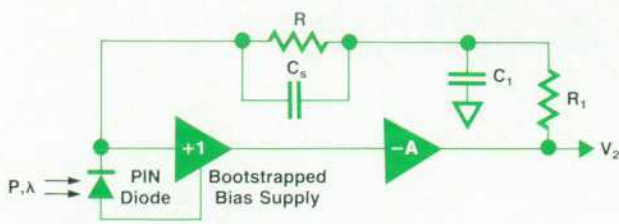
Fig. 10. *HP 8145A simplified transimpedance amplifier with bootstrapped input stage and lag network.*

chain is very important. Because the pin diode output signal is unipolar, it will excite any low-frequency pole-zero pairs in the response function, and the decaying responses of these may introduce baseline offsets. For example, assume that a bypassing network gives the response a pole-zero pair that results in a gain increase of 1% in the closed-loop response at 100 Hz. If the receiver is excited by a 1-μs optical pulse, it is easy to show that the output will contain a 1.6-ms tail with an amplitude equal to 6.3×10^{-6} times the amplitude of the original pulse. At the deepest level of averaging used, the noise floor might easily be as small as 10^{-6} of the pulse reflected from the front end of the fiber, and a serious baseline aberration will be observed. To minimize these effects, dc biasing and coupling are used wherever possible in the receiver, and bypass time constants are made either very long or very short.

In use in the field, it has been found that fiber front-end reflections can be very large, far exceeding the 3% to 4% expected from a perfect cleave. The result is that very high degrees of overload occur in the receiver chain. Various clipping and current steering techniques are used throughout the receiver to optimize the recovery from these events.

During the development of the receiver a number of components with "thermal tail" problems were found and rejected.

Analog-to-Digital Converter

The architecture of the HP 8145A was developed around a basic clock frequency of 8.0 MHz. At this digitizing rate it was only economical to use an 8-bit ADC, an important decision because it determined the linear dynamic range of the receiver chain, and, indirectly, the averaging time required to obtain a given range in the presence of significant discrete reflections. The converter adopted is a CMOS flash type, which does not require a sample-and-hold circuit.

The input noise level and the overall gain of the receiver determine the noise level at the converter input. To minimize digitization noise and assure linearity, the noise level can be made much larger than the step size of the converter. However, this reduces the dynamic range. After a lot of analysis and experimentation, a noise level of about 0.5 step rms was chosen, and a baseline dithering scheme, which considerably improves the overall linearity, was devised. With the deepest averaging and full digital filtering in use, the noise floor of the instrument is equivalent to about 5.0×10^{-4} of a converter step. The output of the converter is capable of reproducing this signal level a few tens of microseconds after a full scale input—a surprising freedom from settling problems.

Data Processing in the Correlating Optical Time-Domain Reflectometer

A powerful special-purpose digital signal processor, a general-purpose main processor, and pipelined measurement firmware work with the optical components to make measurements.

by Jochen Rivoir and Wilfried Pless

THE DIGITAL SIGNAL PROCESSING ENGINE is a powerful 32-bit digital signal processor with an architecture designed especially for the correlating HP 8145A Optical Time-Domain Reflectometer. With its 200 MSI ICs, the DSPE outperforms any available microprocessor or signal processor for its tasks.

After being started by the main microprocessor, the DSPE can work independently while the main processor performs other measurement tasks. The firmware for these tasks is designed for pipelined operation to reduce overhead and maximize throughput.

DSPE Tasks

As explained in the article on page 14, the DSPE sequentially probes the laser driver and thus the fiber with offset-shifted Golay codes A and B and their ones complements. The four probe shots in each sequence are designated p_{A+}, p_{A-}, p_{B+}, and p_{B-}.

Simultaneously the DSPE receives the fiber responses s_A and s_B from the ADC (analog-to-digital converter). To reduce noise, the responses are added into the integration memories (IMEMA and IMEMB, respectively) containing all previous responses.

Each of the two integration sums is then correlated with the appropriate Golay code, and the two correlation results are added to extract the desired impulse response of the fiber f(t). Mathematically the calculated term looks like:

$$\sum_i (s_{A+,i} - s_{A-,i}) \star A + \sum_i (s_{B+,i} - s_{B-,i}) \star B$$

where the $\star$ indicates correlation.

Besides these calculation-oriented tasks, the DSPE is responsible for several timing parameters, like the start delay (determines the start distance) and the repetition rate of subsequent probes into the fiber. The DSPE manages the following main tasks:

- Timing control
- Code generation
- Integration
- Correlation
- Adding results A and B.

DSPE Architecture

The architecture of the DSPE is shown in Fig. 1. The heart of the DSPE is a 32-bit ALU (arithmetic logic unit). To allow pipelining, both of the input ports and the output

port of the ALU are buffered by registers. This allows memory access and ALU calculations to be performed simultaneously. The integration memory is fast enough to put an operand in register REGA or REGB and store the accumulator (ACCU) contents in a single 125-ns machine cycle.

The address generator is a sophisticated set of counters and registers that handles the indexes during integration and correlation.

The code generator logically consists of two Golay code memories for probes p_A and p_B. The selected memory is read bitwise, synchronously with the sampling in the ADC. The duration of one code bit—the pulse width—can be adjusted to an integer multiple of 125 ns. This integer is called the transmit decimation.

The entire measurement, including code generation, timing control, integration, and correlation is controlled by a microcoded state machine.

Microcoded State Machine

Fig. 2 shows a simplified block diagram of the state machine, which runs at 8 MHz. The state machine provides the DSPE hardware with a total of 32 control signals, which
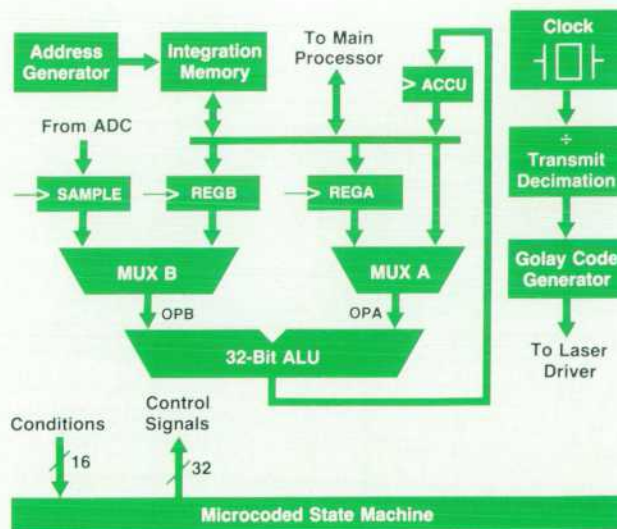
Fig. 1. *Architecture of the digital signal processing engine (DSPE) of the HP 8145A Optical Time-Domain Reflectometer.*

are associated with the current state. The current state is a function of the last state and one selected input. The microcode memory is no more than a large table containing a next state for each given state and input value, together with a set of control signals for the hardware.

The state machine is a very simple but powerful concept. The disadvantage of this horizontal design is the need for a large 2K × 48-bit microcode memory. Compared to a standard processor, the advantage is that control signals can change simultaneously with a jump command, a capability needed by the DSPE.

The state machine also allows the main processor to read from and write to the microcode memory, force a selected state, or read the actual state.

Microcode for the state machine is generated by a special Pascal program running on an HP 9000 Computer. This program makes use of the internal data structure of the Pascal compiler's constants. A single composite Pascal constant was defined whose bit representation corresponds exactly to the contents of the microcode memory. Since the Pascal compiler checks constant declarations during compile time, there was no need for writing an assembler. Only a macro expander had to be written.

This program transfers the microprogram into microcode memory via HP-IB commands, without burning any PROMs. It provides a set of about 100 powerful test programs that are used in production to test the DSPE in real time.

Integration

Fig. 3 shows the hardware involved in the integration process. As described above, integration is the successive component-by-component summation of arrays. The formula is:

```
FOR sample_counter := 0 TO nmb_samples DO
    IMEM(sample_counter) := IMEM(sample_counter) + sample
```

Fig. 4 illustrates the pipelined nature of the integration process. First, the sample counter SCTR is cleared (step 1). On the next machine cycle (step 2) this zero arrives in the READ_ADDR register and the integration memory puts the old integration sum IMEM(READ_ADDR) onto the IMEM_DATA bus. The ADC now provides the SAMPLE register with the first value. In step 3, the old sum and the first sample are clocked into REGA and SAMPLE, respectively, so that the ALU can calculate the new sum and apply it to the ACCU register, where it will be stored in step 4. Meanwhile, the
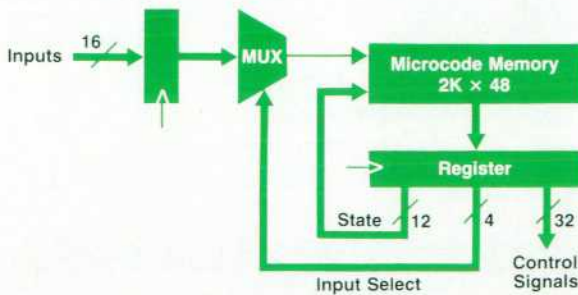
READ_ADDR value of step 2 has arrived (via the ALU_ADDR register) in the WRITE_ADDR register so that the new sum can be stored at the old address.

Since all hardware elements are used for only one machine cycle, all four steps can be pipelined. Attention has to be paid to the integration memory, because it is used in both a read and a write cycle. Fortunately, the write cycle fits in the first half of a machine cycle, and the read cycle fits in the second half.

Fig. 5 is a flow chart of the integration process. Each box in the flow chart represents one step, or one machine cycle. Special attention has to be paid when filling the pipeline that no random writing occurs. Clearing the pipeline is not critical, since writing is inhibited then. The microprogram takes into account the two-step delay from the input of the state machine to its control signals.

Decimation

When the fiber is probed with pulse widths longer than 125 ns, the bandwidth of the receiver must be reduced accordingly. The bandwidth is reduced digitally in the DSPE by adding several successive samples and treating the sum as one sample. The number of samples added is the same as the transmit decimation mentioned above, and the name "transmit decimation" is used for both the number of samples added and the process of adding them.

Transmit decimation is done in real time by keeping partial sums in the ACCU register, feeding the ACCU back to the ALU, and adding new samples to the ACCU. Finally, with the last decimated sample, the sum is stored in the integration memory.

Correlation

The next main task of the DSPE is to correlate the integration sum with the Golay code. The correlation of these sequences, IMEM(i) and GOLAY(g), is given by:

$$\text{Result}(k) = \sum_{g=0}^{L-1} \text{IMEM}(g+k) \times \text{GOLAY}(g).$$

where L is the Golay code length. The fact that Golay codes are binary codes, that is, they consist of ones and minus-ones, reduces hardware requirements significantly, because the multiplication by GOLAY(g) can be replaced by

Fig. 2. *Simplified microcoded state machine.*
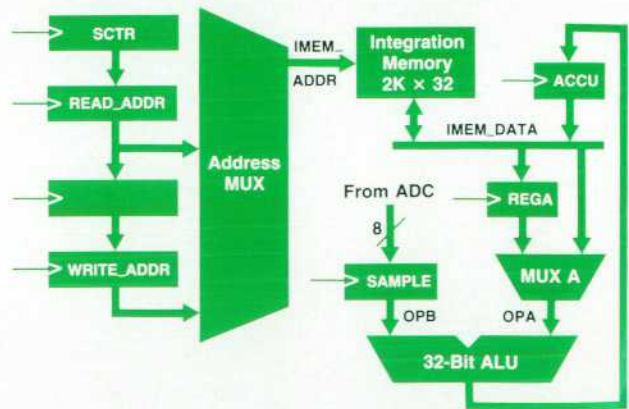
Fig. 3. *Integration hardware.*

Data Path

Step	Sample	Read IMEM	ALU	Write IMEM
1				
2	SAMPLE ← ADC	REGA ← IMEM		
3			ACCU ← REGA + SAMPLE	
4				IMEM ← ACCU
Loop	$i-1$ / i / $i+1$	$i-1$ / i / $i+1$	$i-1$ / i / $i+1$	$i-1$ / i / $i+1$

Address Path

Step	SCTR	READ_ADDR	ALU_ADDR	WRITE_ADDR
1	SCTR ← 0			
2		READ_ADDR ← SCTR		
3			ALU_ADDR ← READ_ADDR	
4				WRITE_ADDR ← ALU_ADDR
Loop	$i-1$ / i / $i+1$	$i-1$ / i / $i+1$	$i-1$ / i / $i+1$	$i-1$ / i / $i+1$
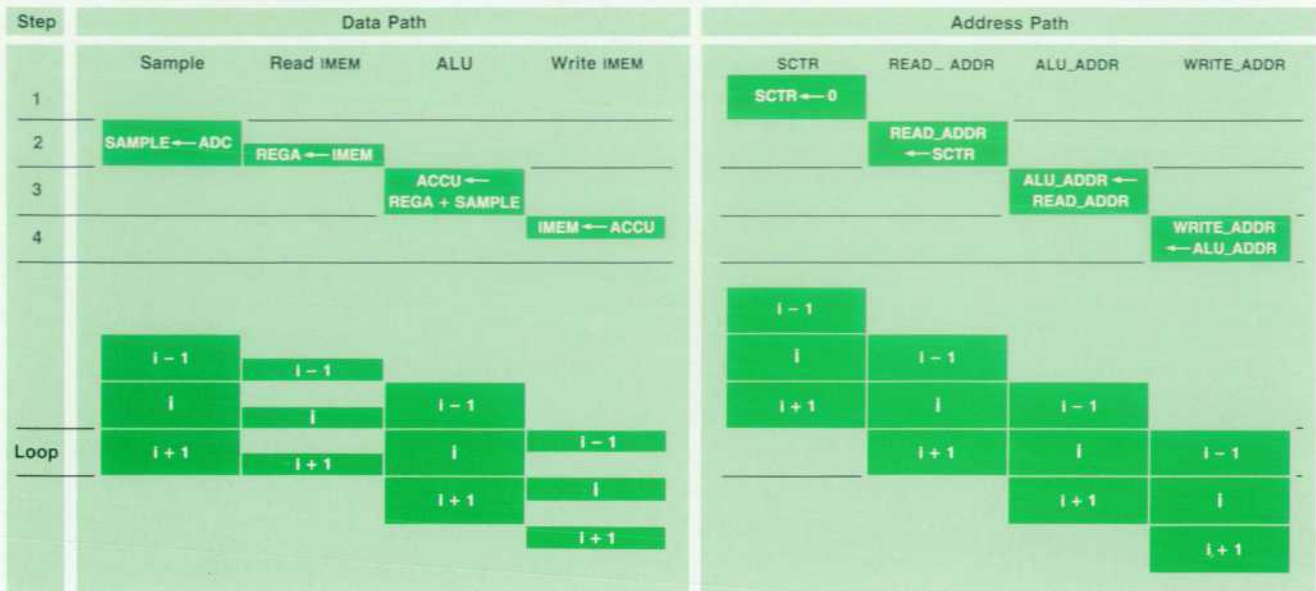
Fig. 4. *Pipelined integration process.*

the decision of whether IMEM$(g+k)$ will be added to or subtracted from the partial sum.

Fig. 6 shows the correlation hardware, which corresponds directly to the formula above. For clarity, only the most important blocks are shown. Every variable or partial result is mapped into a register, a counter, or a memory. This method yields the best speed performance, but needs a significant amount of hardware.

To avoid having to use a second ALU for the sum $g+k$, a special counter for $i = g+k$, the I counter, is used. This I counter is preloaded with k and then incremented simultaneously with the counter for g, the G counter. Thus the I counter has the value $g+k$ without the need of calculating a sum. The following register transfer program shows the correlation algorithm in terms of hardware use.

```
K_counter := 0 ;
REPEAT
    ACCU := 0 ;
    I_counter := K_counter ;
    G_counter := 0 ;
    REPEAT
        CASE GOLAY_MEMORY(G_counter) OF
            0 {−1} : ACCU := ACCU − IMEM(I_counter)
            1 {+1} : ACCU := ACCU + IMEM(I_counter)
        END_CASE ;
        G_counter := G_counter + 1 ;
        I_counter := I_counter + 1 ;
    UNTIL G_counter = L − 1 ;
    IMEM(K_counter) := ACCU ;
    K_counter := K_counter + 1 ;
UNTIL K_counter = number_of_data_ points − 1 ;
```

The specialized hardware architecture of the DSPE makes it possible to execute the whole inner loop within a single machine cycle of 125 ns. By contrast, a standard processor would have to do the following steps:

- Read a bit from the Golay memory
- Do a branch according to the contents
- Read 32 bits from the integration memory
- Do a 32-bit addition or subtraction into the accumulator
- Increment a register containing g
- Increment a register containing k
- Compare g with $L-1$ (the loop limit)
- Take the branch (in most cases).

The DSPE executes a correlation of 500 24-bit data points with a code sequence of length 256 in less than 20 mil-
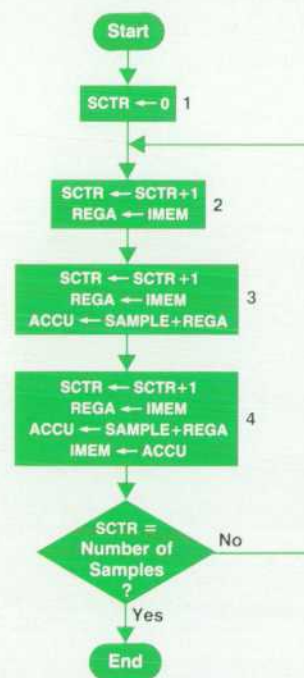
Fig. 5. *Flow chart of the integration process.*
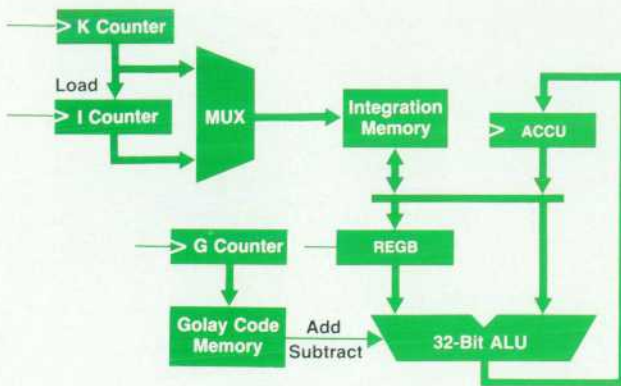
Fig. 6. *Correlation hardware.*

liseconds, much less than any other solution available.

Measurement Firmware Tasks

After being initiated, the digital signal processing engine (DSPE) is designed to work independently of microprocessor control for a period of time. The time is limited by the DSPE's memory depth (and by the patience of the user, who is waiting for a new display update).

While the DSPE is doing its decimating, integrating, and correlating tasks, the microprocessor postprocesses previously taken data under control of the measurement firmware. The microprocessor does lin/log conversion, normalization, stitching, and discontinuity detection. These terms, and especially the consequences of the latter two for parameter setting, will be explained in the following paragraphs.

Fig. 7 shows the tasks of the measurement firmware and its interdependencies with the DSPE. As shown in Fig. 7, a measurement is done in stages or frames. To ensure that frames 0 and 1 have valid data that is not corrupted by sidelobes these frames are measured with single pulses (L = 1).

Operation is pipelined. During the first part of each frame, the microprocessor transfers data from the DSPE and sets up the DSPE for the next frame. Within a short period of time, the microprocessor system transfers data from the DSPE to the microprocessor RAM, does a further integration of the linear data for additional noise reduction, and executes a firmware routine that finds out the best code length and gain combination to maximize the noise

reduction without incurring analog-digital converter (ADC) clipping for the next measurement frame.

During the second part of each frame, the microprocessor postprocesses the DSPE data while the DSPE probes the fiber.

Initialization

After starting the measurement, the microprocessor firmware transforms user-chosen parameters such as start distance, span, average mode, wavelength, and pulse width to register values that are understood by the hardware. Start distance and span are converted to timing parameters that determine values for various DSPE registers.

Data Transfer

In continuous-average mode, the DSPE data of a frame is continuously transferred from DSPE RAM to microprocessor RAM, where it is added to previously taken data. The word size of this linear result memory allows theoretical noise reductions of more than 40 dB, depending on the pulse width.

In refresh mode, the DSPE data is transferred to the microprocessor directly and is not added to previous results.

Lin/Log Conversion

As mentioned earlier, the display of an OTDR is a semilogarithmic plot of the fiber impulse response. Fig. 8 shows a flow chart of the lin/log conversion routine that converts the linear impulse response data to logarithmic units.

Stitching

Measurement accuracy requires that the measurement system consisting of laser driver, fiber, receiver, and ADC be absolutely linear. Careful design of the laser driver makes this component highly linear. Nonlinearities of the fiber are negligible at the HP 8145A's power level. However, strong reflections at the front end of the fiber cause saturation effects in the receiver and/or clipping of the ADC, both highly nonlinear effects.

If the ADC clips at least once in the area of interest, this information is latched by the hardware, and is read by the firmware. Since the ADC clips before the receiver saturates, detection of ADC clipping is sufficient to detect both ADC clipping and receiver saturation. If ADC clipping is de-
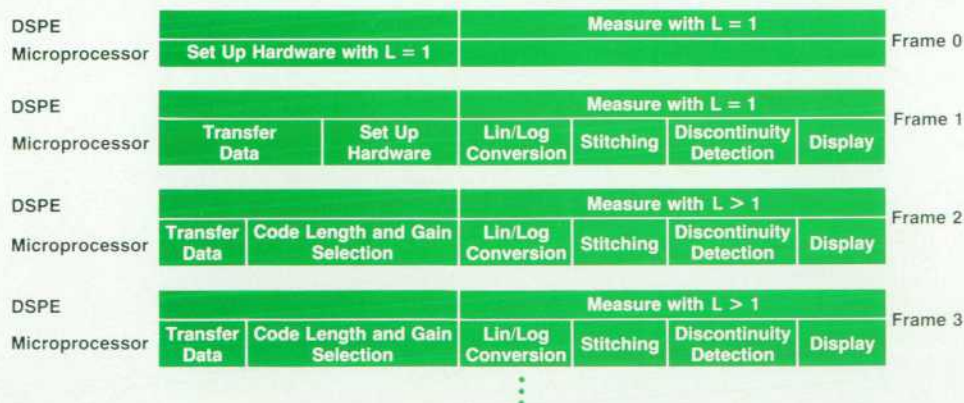
Fig. 7. *Relative timing of the tasks of the measurement firmware and the tasks of the DSPE.*

tected, the firmware will probe the fiber with a single pulse.

Since the ADC clip level and the receiver noise level are known, this information, together with the calculated noise reduction, can be used to determine the signal-to-noise ratio (SNR) of every measured point of the fiber of interest. When the SNR of the beginning of the fiber (leftmost part of the display) guarantees that the measured values lie within a ±0.05-dB boundary, this part of the fiber will not be measured anymore, while measurements continue on only the part of the fiber that has an insufficient SNR.

This means that the nonlinearities that forced single-pulse measurements before are of no concern for subsequent measurements. As long as there are no non-linearities in the remainder of the fiber that cause ADC clipping, the remainder of the fiber can be measured with longer codes (L > 1).

Constructing the fiber impulse response of parts that are measured with different combinations of code length L and laser gain G is referred to as stitching. Fig. 9 illustrates the stitching procedure.

Discontinuity Detection

The sidelobe suppression that works perfectly in theory has its limitations in the real world. According to the theory of the complementary code technique, the sum of the two autocorrelation functions for codes A and B equals an ideal Dirac delta function of height 2L. In the real code correlation OTDR, the codes used to probe the fiber are only close approximations of ideal A and B codes. Finite rise and fall

times of the laser driver, ringing, and droop effects cause ones in the code to be only near-ones. These nonideal effects result in imperfect sidelobe cancellation. The peak-to-sidelobe ratio is about 35 dB.

Sidelobes occur in an area of width ±L/2 around the ideal impulse. This means that instead of a delta function of height 2L the fiber is probed with a delta function of height 2L and associated sidelobes of width ±L/2 around the peak.

At strong reflections the sidelobes can interfere with the backscatter. A measurement firmware algorithm detects
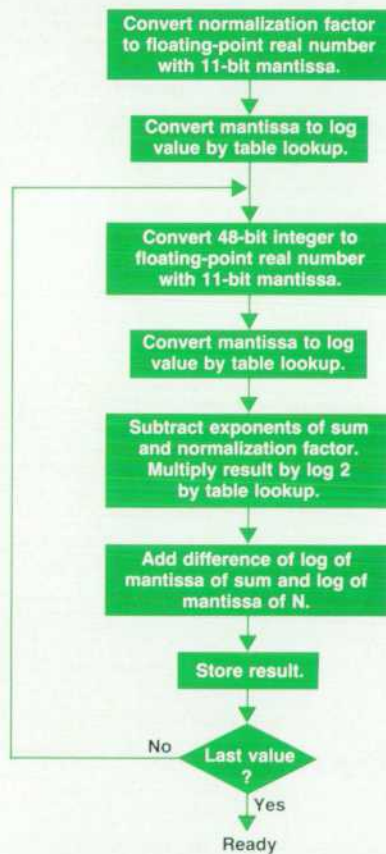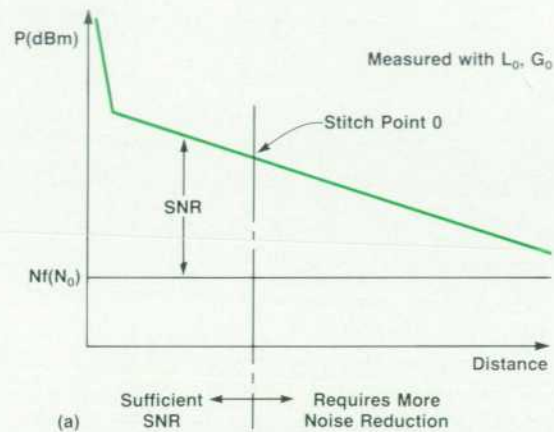
Nf(N_i) = Noise Floor After N_i Averages
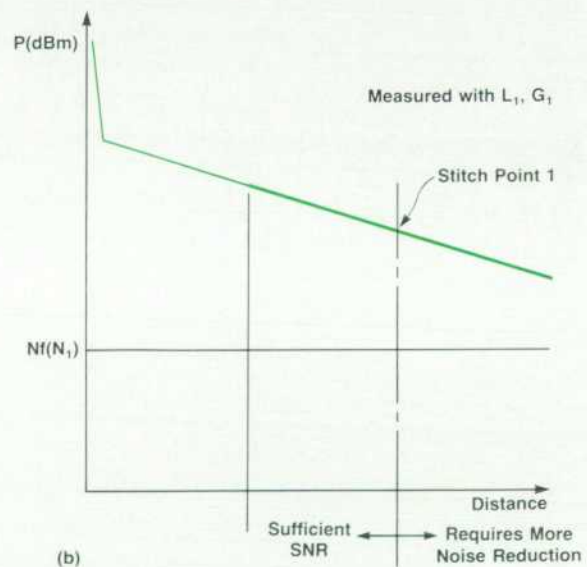
Fig. 9. *Stitching is the process of constructing the fiber impulse response of parts that are measured with different combinations of code length and gain. (a) Measurements are made on the entire fiber and averaged until the signal-to-noise ratio (SNR) is sufficient to guarantee accurate values (within ±0.05 dB) for the beginning of the fiber. The first stitch point (0) is the point where the SNR becomes insufficient. (b) The code length and/or gain can then be increased to measure the next section of the fiber. Returns from the first section, which may now be large enough to cause ADC clipping, are ignored. The process is repeated until the entire fiber has been measured accurately.*

Fig. 8. *Flow chart of the lin/log conversion routine.*
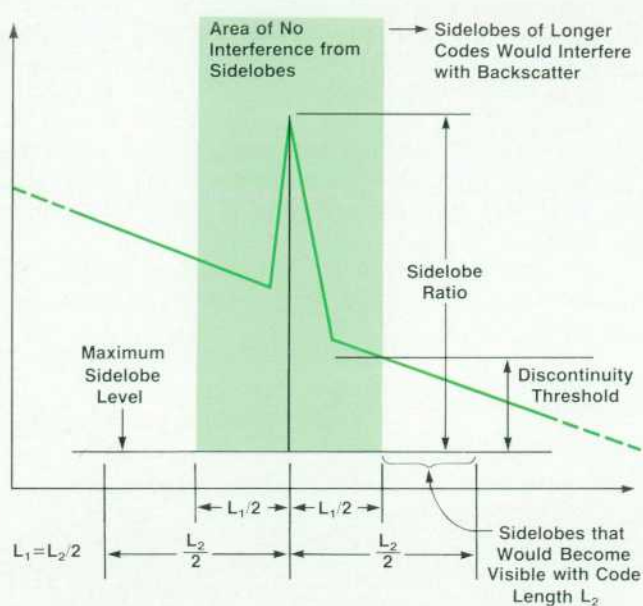
Fig. 10. *Limitation of code length by strong reflections.*

discontinuities by comparing each measurement sample with the samples from a range L/2 on either side of it. When the difference is too large, the sidelobes associated with that sample can interfere with the backscatter. If this is the case, the code length is halved and the check is performed again. If there are still sidelobe problems, the code length is halved again, and so on. The algorithm determines the maximum allowable code length for measuring the selected part of the fiber. Fig. 10 shows the areas where sidelobes would be visible if no precautions were taken.

Code Length and Gain Selection

The maximum code length with which the fiber can be probed is not necessarily the code length determined by the discontinuity detection algorithm, since the fiber response to this code might cause clipping in the ADC. On the other hand, maximizing the SNR requires that the maximum possible probe energy be used.

To establish this trade-off, another algorithm probes the fiber with a test shot, using the code length L determined by the discontinuity detection algorithm and minimum gain G. Any clipping in the ADC during the measurement is latched in a hardware register that can be read by the microprocessor. If no clipping occurs, the gain is increased and another test shot is fired. If clipping occurs, the code length is reduced for the next test shot. The algorithm determines the maximum L and G to optimize the power coming back to the receiver without ADC clipping.

Conclusions

The pipelined structure of the measurement firmware, together with the signal processing hardware, allows very efficient data throughput with only a small amount of overhead. Limitations in code length because of front reflections or excessive discontinuities are reduced by stitching a trace together using parts that are measured with code lengths and gains that are optimally adapted to the fiber to be measured.

Acknowledgments

Optical Time-Domain Reflectometer User Interface Design

The firmware built into the HP 8145A OTDR is the invisible intelligence that makes a complex instrument easy to use.

by Joachim Vobis

THE HP 8145A OPTICAL Time-Domain Reflectometer described in this issue is a highly sophisticated machine. Its hardware and its theory of operation are difficult to understand. On the other hand, most users want to see an interface that is self-explanatory and helps them do everything from simple to complex tasks easily.

The creation of a friendly user interface for both local and remote control is the task of a large portion of the firmware built into the HP 8145A. The firmware can be divided into five parts, as shown in Fig. 1:

- A central data base holds the machine's states in ten registers (0-9) and the measurement results in thirteen internal registers (0-12) and 114 registers in an external plug-in module. Setting register 0 holds the current parameter settings and measurement register 0 holds the current measurement.
- The keyboard/display process provides input and output between the data base and the front panel. This part of the firmware implements the local human interface.
- The HP-IB process does the same for remote control, conforming to IEEE 488.1 and 488.2. This process also dumps measurement results to a plotter or an HP Think-Jet printer without an external computer.
- The measurement process executes the measurement by reading setting register 0, programming the hardware and processing the data received. It stores its results in measurement register 0. This part of the firmware is described in the article on page 29.
- The check process implements test and verification routines such as the self-test during power-on.

The largest piece of firmware is the keyboard/display process. While it was possible to develop the other parts mostly in the lab, the human interface was grown in an interactive way. Successive prototypes were shown to customers. The feedback allowed us to optimize the human interface for the requirements of typical applications, and to add features wanted for advanced tasks. Because users can be divided into three groups, three levels of use are offered:

- Standard use. The user wants to select the distance range and run the measurement. Results and markers are obtained directly from the display or by creating a hard copy on a printer or plotter.
- Very easy use. Most of the hard keys and softkeys can be disabled (blanked) by HP-IB commands. Typically only the keys to recall settings and to start and stop the measurement are left. A user can recall the proper settings and execute the measurement, but can't do any-

thing wrong by accidently pressing an unwanted buttom.
- Sophisticated use. For documentation, special measurements, and comparison of measurement results, a set of features is available to the skilled user. Some of the features are not offered by other OTDRs, so they are designed to be sufficiently self-explanatory that the user can become acquainted with them without a lot of training or reference to manuals.

Front-Panel Interface

The hard keys on the front panel are divided into four groups that provide the functions most often needed (Fig. 2). The **START**, **STOP**, and **CONT**inue keys control the execution of measurements. The digit keys together with the knob and the vernier allow numerical entry to set or modify parameters. The marker keys activate different markers to calculate distances, attenuation, and losses. **STO** stores an instrument state in one of the setting registers (1-9) and **RCL** recalls it. **RCL 0** can be used for reset (load factory-defined setting) and **RCL ENTER** rescales the X and Y axes of the display to show the complete measurement result.

Fig. 2 shows a typical CRT image. An X-Y grid contains the measured curve. The grid's Y scale (offset and dB/div) is on the left side and the X scale (start and stop position) is directly under the grid. Below the X scale, a measurement bar shows the measurement range and the calibration values.

In the upper right corner is a readout of marker values. The left and right markers are used to measure distances, losses (dB), and attenuation (dB/km, least-squares approx-
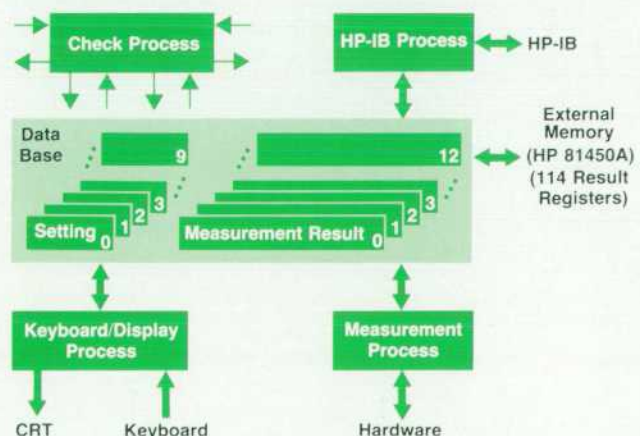
Fig. 1. *HP 8145A OTDR firmware structure.*

imation). The splice marker can be used automatically (by default) or manually to calculate the loss of splices or connectors.

Softkeys

The right side of the display contains the softkey labels, which offer functions not provided by the hard keys. Because there are many more tasks than softkeys, the softkey functions are arranged hierarchically, the most important on the highest level and the less important in one of the sublevels.

The main level is split into two halves. The first half controls the X and Y axes (start or center position, distance span, dB/div, and vertical offset) while the second half of the main level manipulates the measurement parameters (continuous averaging or refresh mode, auto or manual pulse width, 1300 or 1540 nm wavelength). In the continuous average mode, all modifications are editor operations, that is, changing a value affects only the display and not the measurement. This can be easily demonstrated with the X scaling: changing the start/center position or the span has a zooming effect, but the measurement as shown in the bar at the bottom runs with the same distance range. Pressing **START** to initiate a measurement forces the edited values to become the actual parameters for the hardware.

Submenus

The submenus allow the user to set the fiber characteristics, memory functions, hard-copy features, and display and HP-IB options.

Fiber characteristics include the refractive index, defined as the ratio of the speed of light in a vacuum to the speed of light in the glass fiber, and the ratio of the fiber length to the cable length (windings can make the mechanical length of a cable shorter than the optical length seen by an OTDR).

The memory functions are used to assign an identification label (up to 39 characters) to a measurement result and store it together with the corresponding setting in one of the measurement result registers. Internally, the HP 8145A has 12 of these registers. An additional 114 curves can be stored in the HP 81450A, a nonvolatile memory module which plugs into the rear panel of the OTDR and can be exchanged at any time. The curves can be recalled to be analyzed (zoomed), dumped out, backed up via the HP-IB, or used for comparisions (see below).

The dump menu provides one-keystroke hard copies from the CRT to a ThinkJet printer or an HP-GL plotter without the need of an external computer. If a user has made a lot of measurements and stored them, PRINT FROM/ TO or PRINT ALL can be used to print them.

The display and HP-IB options in the lowest softkey level set brightness, HP-IB address, curve style (dots or lines),
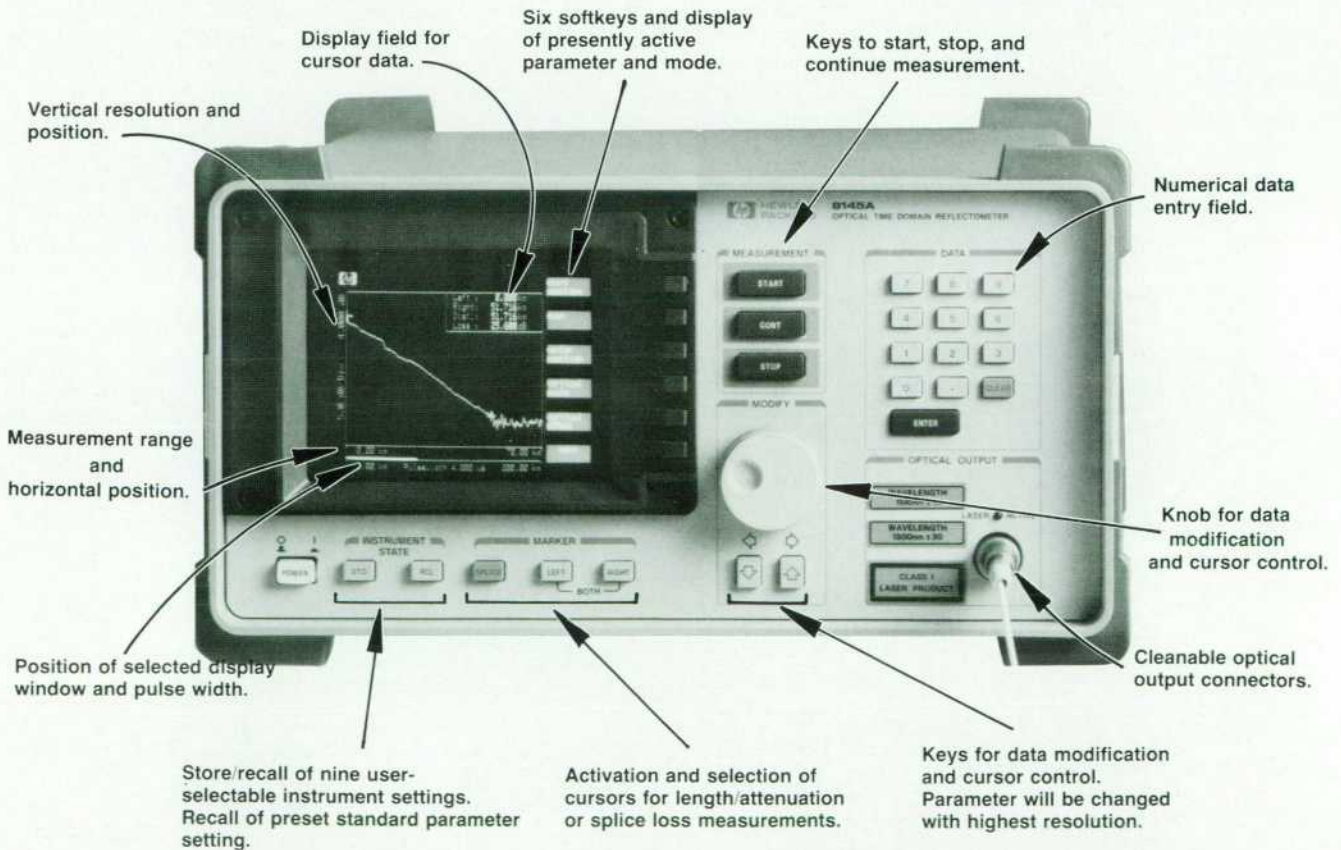
Fig. 2. *Human interface for local control.*

and the distance unit. The distance unit can be set to kilometers, feet, or miles.

Comparing Curves

One of the HP 8145A's advanced features is curve comparisions. The COMPARE MODE softkey causes the OTDR to take an older measurement result as a reference and compare the current measurement or another stored trace with it (see Fig. 3). Comparing two curves is very helpful when:
- A fiber must be characterized at more than one wavelength
- A fiber must be compared with a curve showing the specification limit (quality assurance)
- A fault must be found by comparing a current measurement with an OTDR curve recorded when the fiber was inspected after installation
- Small changes of fiber characteristics must be detected (research and development tasks, environmental tests, aging of fiber/splice/connector losses, sensor characteristics, etc.)

Many experiments were done to see if it would be better to calculate and display the difference of the two traces or to show the original waveforms. Because of the familiarity of OTDR curves, we decided to take the latter approach. It is not as natural for the human mind to interpret the calculated difference as it is to look for parallelism of the two original traces. In noisy regions it is hard to tell whether a peak is a noise peak or a real difference. Using the vertical offset controls to move one curve over or close to the other provides much more meaningful results.

HP-IB Capabilities

The functions offered in the human interface can be restricted as mentioned above. Fig. 4 shows an example in which some of the softkeys are blanked by HP-IB commands.

All normal programming of the OTDR can be done remotely over the HP-IB. The command and query syntax, the data formats, and the status reporting conform to IEEE 488.2. The mnemonics used for commands can be given in a short or a long form. For example, the string OFFS has the same meaning as OFFSET and STARTPOS? can be abbreviated to STAR?. The length unit used in messages can be meters, miles, or feet.

Measured data can be read remotely in the following ways:
- The command POINTS? returns the number of curve points, or zero if no measurement result is available.
- YVALUE? <i> returns the dB value of point i of the curve. This method is slow, but easy to use because it returns only one real number in ASCII format. Furthermore, the command YVALUE <i>, <dB> overrides the value measured to build a synthetic curve (e.g., a specification limit curve) as a reference curve for the compare mode.
- DATA? returns a long ASCII string representing the measured curve as a list of points. It is medium-fast and good for computers not supporting binary transfers. Many BASIC programmers with little experience in complicated I/O commands use this mnemonic to get the data.
- BDATA? transfers the entire measurement result register in binary format, including the settings and all bytes needed internally. The command BDATA ... sends this block back to the OTDR.

A useful property of the HP-IB command pair BDATA? and BDATA ... is the possibility of uploading measurement results to a computer and restoring them back to the OTDR. Downloading and uploading use binary block transfers. This method is important because of its high transfer speed (about 0.7 second per curve). It is useful for building data
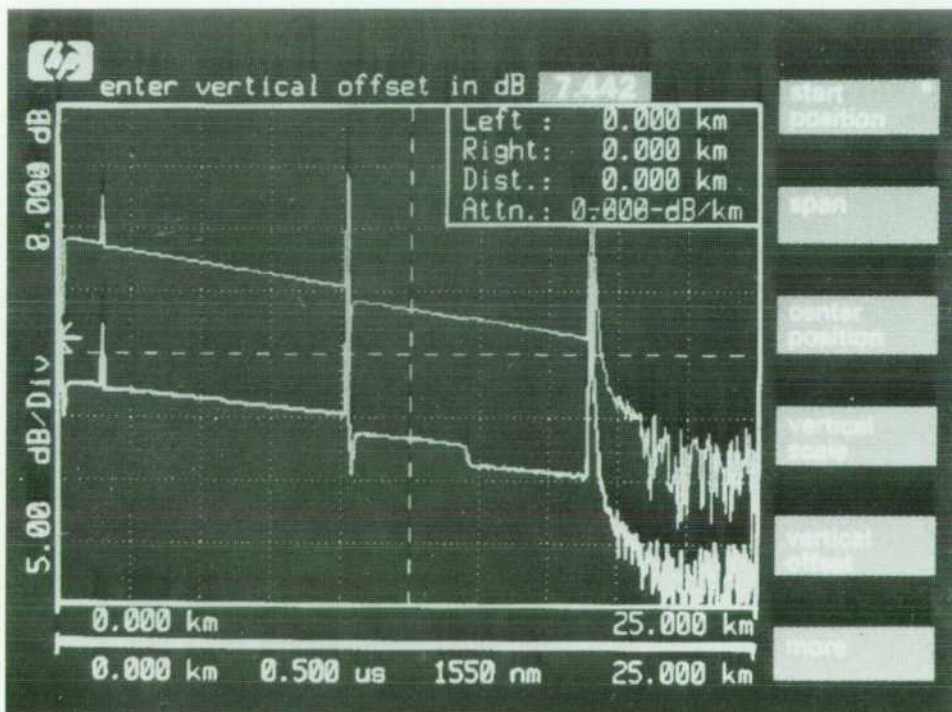
Fig. 3. *Compare mode is useful for comparing a stored curve with the current measurement or another stored curve.*
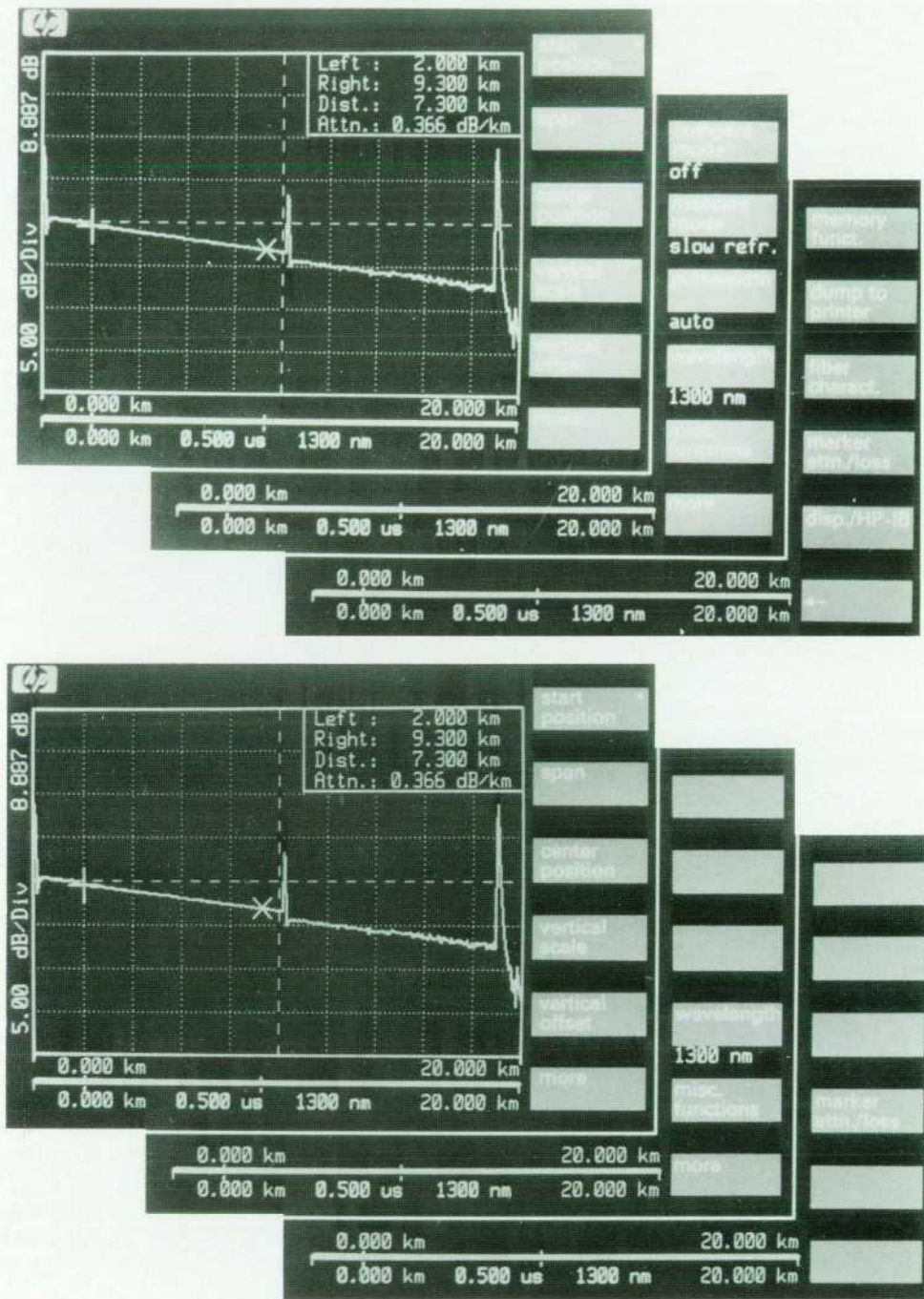
Fig. 4. *Menu example with no blanked softkeys (top) and with eight selectively blanked keys.*

bases of measurements in fiber optic networks or for reloading curves for comparison purposes (finding faults, testing aging, etc.).

Acknowledgments

Printing on Plain Paper with a Thermal Inkjet Printer

An understanding of "plain papers" and how their variability affects performance played a key role in the development of the HP DeskJet printer.

by Steven J. Bares

THE HP DESKJET PRINTER[1] is the first low-cost inkjet printer to produce true letter-quality print on plain paper. Intended for office use, the DeskJet printer was designed specifically to meet customer demand for products that use available office papers. Approximately three million tons of paper is produced and distributed annually to offices in the U.S.A. alone. This paper is made on various types of machinery using many different processes and materials. A major challenge during the development phase was to construct a printhead that would produce reliable print quality over such a wide range of papers. To meet this challenge, the design team needed to identify and obtain samples of the types of papers available to the user and to understand the chemical and physical variability of these papers. This included obtaining information about the paper industry's raw materials, manufacturing processes, and distribution networks. This paper summarizes the results of these studies. Topics covered include the types of papers found in the office, the properties of these papers, and the effect of these properties on inkjet print quality.

What Is Plain Paper?

To design a printhead that produces consistent letter-quality print on plain papers, the development team needed to obtain paper samples for testing during the design phase. However, simply calling a paper company and requesting samples of plain paper will result in an incredulous response. The properties of all papers are radically different and depend entirely on the paper's intended use. Hence, there are no "plain" papers, only papers that are sold into a particular environment.

In U.S. offices, thousands of papers can be found, manufactured by over 75 paper companies. To conduct a complete study of all papers bought by thousands of distributors and sold to literally millions of offices nationwide would involve a vast expenditure of resources. Our first major dilemma in selecting papers for testing was to understand the processes by which business papers are brought from manufacturers into offices. This complex and dynamic distribution network is diagrammed in Fig. 1. We began by focusing our efforts on identifying high-market-share, manufacturer-labeled office papers—only one of the possible channels for paper distribution. It was felt that this approach would identify the office papers having the largest overall market share while minimizing the expenditure of resources.

Laboratory testing of over 150 of these selected papers revealed a wide range of properties for two general types of office papers, which we will call correspondence and copier papers. Correspondence papers are rough, have a large amount of starch on the surface (are hard sized), are typically acidic, and usually contain cotton fiber as well as traditional wood pulp. The relatively homogeneous nature of these correspondence papers simplified the task of designing a compatible printhead.

The specifications for copier papers, which represent over 88% of the paper tonnage shipped to U.S. offices, are well-documented and conform to an industry standard. Given our favorable experience with nonstandardized correspondence papers, we had high expectations of rapidly achieving excellent print quality on copier papers as well. However, when we tested copier papers, we found that their only homogeneous property pertinent to inkjet printing was their surface smoothness. Other potentially important factors affecting inkjet print quality (discussed later) showed significant variations.

The next step toward the goal of obtaining consistent print quality on office papers was to select a small subset
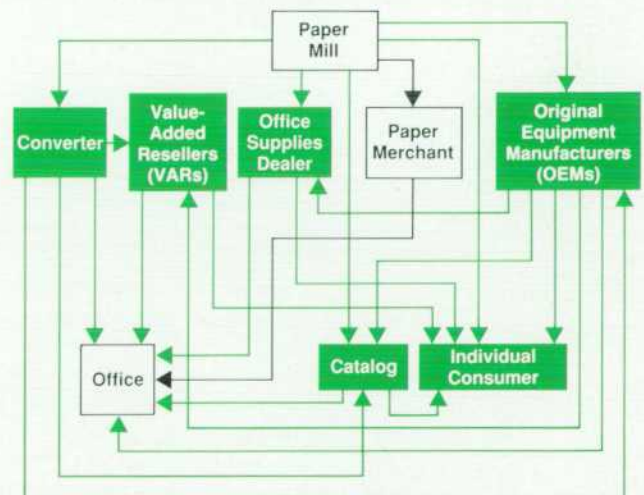
Fig. 1. *Schematic of the distribution network used by manufacturers to get papers into the office. Market research at Hewlett-Packard focused on manufacturer-labeled papers sold by paper mills to paper merchants who supply paper to businesses.*

for printhead optimization from the original 150 identified by market research. The criteria used to consolidate these papers were driven by both the need to cover the broadest market representation and the desire to include the widest ranges of print quality, fiber content, filler content, pH, type of internal sizing, smoothness, and many other variables. This work resulted in the selection of eight representative papers, which were purchased and used throughout the design phase. After selection of the eight-paper set and before the optimization phase, sufficient amounts of each paper sample were purchased so that paper consistency was maintained and only the ink and printhead design varied. The names of these eight papers and some of their properties considered important to the design of the DeskJet printhead are shown in Table I.

Similar research into European office papers was performed in parallel with the work reported above. This research focused on office papers in Germany, France, England, Switzerland, Holland, and Sweden. Because of the expense of market research, we limited our study to these countries and assumed that the information collected was representative of the office papers of all Western Europe. Our preliminary analysis of European office papers, obtained before the completion of the entire study, guided our choice of the eight-paper representative set for the U.S.A. Our analysis of European papers showed that they, like the office papers found in the U.S., displayed a wide range of properties, the major difference being a higher number of alkaline papers (pH above 7) in Europe. We determined that the range of properties covered by the U.S. eight-paper set was sufficient to represent its European counterparts. Hence, our optimization of the DeskJet printhead was performed on the eight-paper set and it was assumed that these papers typified those found in both U.S.
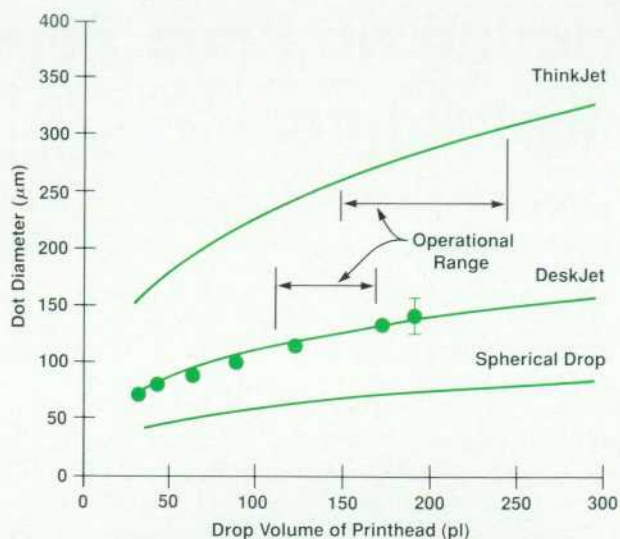
Fig. 2. *Plots of dot diameter versus printhead drop volume for the ThinkJet and DeskJet printers. Data for the ThinkJet printer was obtained using HP ThinkJet paper. The curve shown for the DeskJet printer is determined by averaging dot-diameter data taken from eight copier and bond papers chosen by Hewlett-Packard to represent the office market. The curve labeled Spherical Drop is the in-flight diameter of ink drops at each drop volume calculated assuming spherical geometry. For both curves, the solid lines are obtained from a least-squares fit of the data to a cube-root model.*

and European offices. No detailed studies of office papers found in other countries were performed, although samples of papers from around the world were occasionally tested.

Table I

Representative Set of Papers and Selected Properties Used for Optimization of DeskJet Printhead and Ink

Paper	Manufactured in	Paper Grade	Paper pH	Degree of Sizing	Type of Filler	Surface Smoothness (s)	Fiber Type	Average Fiber Length (mm)	Herculese Test (s)	Porosity (s/ml)
Neenah CLASSIC Laid	Wisconsin	#1 Sulfite	4	Hard	Kaolin Clay $Al_2O_3 \cdot 2SiO_2 \cdot 2H_2O$	370	Birch, Aspen, Pine	0.69	139.8	32.6
Gilbert Bond	Wisconsin	#1 Premium	4	Hard	Kaolin Clay	300	Birch, Aspen, Pine, Spruce	0.75	78.8	21.4
Hammermill Fore-9000	Washington	#4 Copier	5	Variable	Kaolin Clay	144	Douglas Fir, Spruce	1.80	73.3	12.8
Ardor Xerocopy	Arkansas	#4 Copier	6	Variable	Talc $3M_gO \cdot 4SiO_2 \cdot H_2O$	139	Birch, Aspen, Pine	1.02	50.6	13.8
Ardor Bond	Arkansas	#1 Writing	7	Variable	Kaolin Clay and Chalk	172	Birch, Aspen, Pine	1.01	9.5	15.1
Champion Datacopy	Alabama	#4 Copier	5	Variable	Kaolin Clay	157	Birch, Aspen, Pine	0.96	74.5	16.8
IPCO High-Speed	New York	#4 Copier	8	Variable	Chalk $CaCO_3$	143	Birch, Beech, Pine	1.25	41.9	9.6
Xerox 4024 DP	Foreign	#4 Copier	6	Variable	Talc	130	Eucalyptus	0.68	261.9	13.0

Print Quality and Plain Papers

Selection of the representative office paper set allowed the design team to optimize the printhead on a small test bed with confidence that the papers reflected the majority of office papers. To obtain the most consistent print quality, the ink and printhead were chosen to yield the smallest variation of printed dot diameters over as many of these papers as possible. Vision system analysis as described in an earlier article[2] provided the best means for reproducibly measuring printed dots on the paper.

We have found that a plot of printed dot diameter versus printhead drop volume is the best means of illustrating the variation of print quality over the papers used for a particular inkjet printer. Drop volumes are calculated by measuring the mass of ink after firing a known number of drops into a container. Printheads with different drop volumes are obtained as a result of normal manufacturing variations during production. A plot of this type for the DeskJet printer, using dot diameters averaged over the entire eight-paper test set and printed with the final DeskJet ink is shown in Fig. 2. For comparison, a similar plot for the ThinkJet printer using its special paper is also shown, along with the calculated in-flight diameter of an ink drop at each drop volume, assuming a spherical drop geometry.

Only printheads that have drop volumes in a specified range are supplied for use in the ThinkJet and DeskJet printers. This operational range is noted for both these printers in Fig. 2. In the ThinkJet printer, the special paper and the wider customer tolerance for variations in draft-mode print quality allow a wide operational range to be used. For the DeskJet printer, the operational range is much narrower. This constraint on the drop volume range translates into small variations of print quality from printhead to printhead and ensures that consistent print quality is obtained on the broadest possible range of office papers.

In the DeskJet printer, comparison of the dot diameters obtained at any drop volume with the diameter calculated assuming a spherical drop measures the amount of spread the drop undergoes after it strikes the paper. Comparison of this spread factor with that obtained with the ThinkJet printer provides some insight into one of the most important of the properties of paper that determine inkjet print quality. To optimize the print quality of the ThinkJet printer
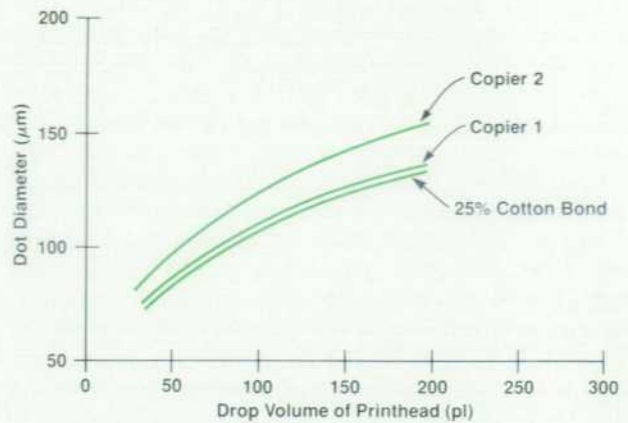
Fig. 3. *Plots of dot diameter versus printhead drop volume for three office papers on the DeskJet printer.*

a paper that yielded a spread factor near four was needed. To get such an absorbent paper that also yielded dots that were as circular as possible required a special paper, since most office papers don't have the desired properties. To achieve homogeneous print quality with the DeskJet printer on office papers, the spread factor has been reduced significantly below that of the ThinkJet printer. This reduction in the DeskJet printer's spread factor is obtained by ink formulation.

The difference in spread factor between the ThinkJet and the DeskJet printers is a direct measure of the relative dependence of each printer's print quality on its media. Even with the final printhead design and ink formulation, optimal print in the DeskJet printer is obtained with a spread factor of about two. Because the spread factor is greater than one, the ink drop must interact with the paper. Therefore, there is still some dependence on paper, and consequently, there is a potential for papers to have unacceptable print quality. The range of spread factors observed on copier papers was the primary difficulty in finding an ink that prints consistently over all papers of this type. Even with the optimized printhead and ink, there are some copier papers that print unacceptably because of this range of surface physiochemistry in office papers.
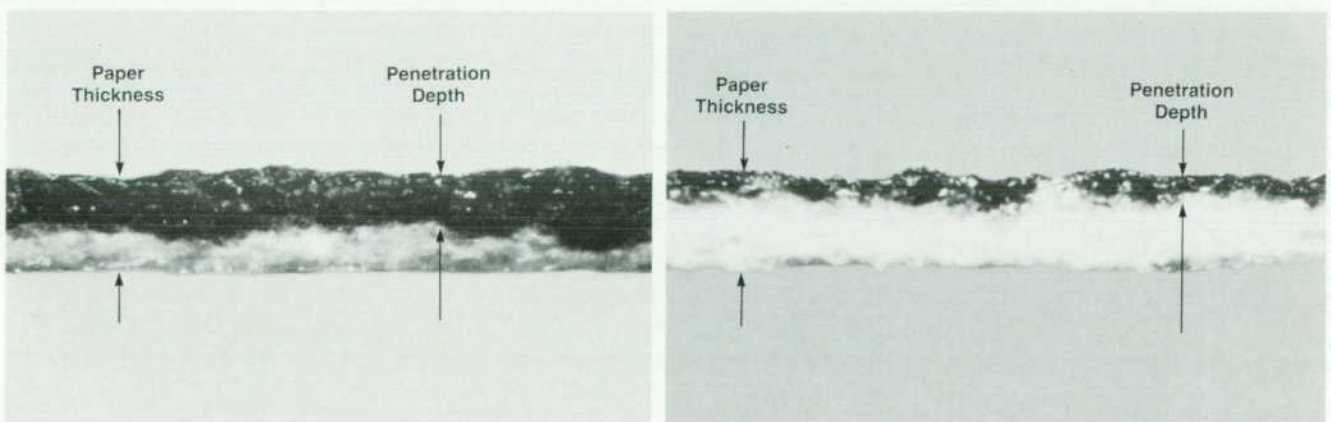
Fig. 4. *Cross sections of printed papers showing depth of ink penetration on a paper with good print quality (right) and one with poor print quality (left).*

Papers that display poor print quality in the DeskJet printer can be readily characterized by a more detailed examination of a plot of dot diameter versus printhead drop volume. Such a study is shown in Fig. 3 for three typical office papers printed with the DeskJet printer. The curves labeled Copier 1 and 25% Cotton Bond are representative of most copier and correspondence papers found in the office using the current DeskJet ink and printhead design. Some copier papers yield curves that lie significantly above most other office papers. The characteristic curve labeled Copier 2 is one example. Papers with characteristic curves that lie above most others yield poor print quality with the DeskJet printer. This is because the higher absorbency allows too much spread of the ink drop across the paper surface. Much of our effort to understand paper has been centered around understanding the causes of poor print quality in these few papers.

In papers that print poorly with the DeskJet printer, too much penetration occurs, not only across the paper surface as shown above, but also into the bulk of the paper. This is demonstrated in Fig. 4, which shows cross sections of papers printed at 300 dpi. In this figure, a cross section of copier paper that displays good print quality is shown on the right. In this case the penetration depth is less than 20% of the paper thickness. Compare this penetration depth with a copier paper that yields poor quality, as shown in the photograph on the left of Fig. 4. Here, penetration is greater than 50% of the paper thickness.

Based on these types of studies it is clear that to understand what causes some papers to print poorly in the DeskJet printer, we must understand differences in ink penetration dynamics at plain paper surfaces. A simple scanning electron microscope or optical comparison of the surfaces of two papers displaying radically different print quality shows that the surfaces appear very similar. Analysis of fiber length and fiber types in many examples of papers shows little correlation with observed print quality. Initially, it was surprising to find that the obvious properties that relate to the ink absorption process, such as fiber length, fiber type, porosity, and smoothness, did not show any correlation with the observed inkjet print quality. This set of somewhat counterintuitive observations can best be explained by considering the manufacturing process for office papers, the operations that control the amount of penetration, and the traditional uses for office papers.

Controlling Fluid Penetration

Papermaking is the art of taking wood pulp and converting it into sheets with some desired set of physical properties. Office papers are characterized by their limited ability to accept ink into the paper surface. This property is very important if the papers are used with roller-ball pens or in printing presses. The ability to accept an aqueous fluid such as the ink in the DeskJet printhead is the dominant property of paper affecting inkjet print quality.

A simplified schematic view of a papermaking operation is shown in Fig. 5. In the modern papermaking process, a dilute (0.5%) mixture of wood fibers and various additives in water is pumped onto a moving wire or belt. In part, the additives limit the hydrophilic character of the wood fibers. As the wire moves along, water drains through the mesh. After much of the water has drained, the sheet leaves the wire and enters the press section for continued dewatering. To impart further fluid resistance, an additional hydrophobic material, typically starch, is placed on the paper surface in the size press. The effect on print quality of increasing the hydrophobicity of the paper surface at the size press in the DeskJet printer is demonstrated in Fig. 6. In this experiment, print quality is compared for sized and unsized sheets of standard copier paper on the DeskJet printer. Print quality is greatly improved by the addition of the size-press starch. Our studies show that the chemical interactions between the starched paper surface and the aqueous inks dominate the ink/paper interaction. This explains why no macroscopic or other obvious difference could be observed between papers with good and bad print quality in high-magnification studies.

Increasing the hydrophobicity of the paper surface improves print quality in the DeskJet printer by reducing the penetration rate of the ink. Correspondence papers are all heavily sized to give them good writing properties and a stiff, high-quality feel.[3] Indeed, it is the similarity in the expected uses of these papers that drives their need to have low and consistent penetration rates. Fortunately, this property also yields reproducible print quality on the DeskJet printer. Copier papers are designed to be penetrated by toners at high temperature, making the dynamics of fluid penetration in these papers almost irrelevant to their use. It is no surprise, then, that fluid penetration rates in these papers vary over a broad range.

By optimizing on the eight-paper test set, an ink formulation and printhead design were achieved that yield consistent print quality on virtually all of the correspondence papers we tested and most copier papers as well. However, even with the optimized final design, papers with curves like that of Copier 2 in Fig. 3 still yield unacceptable print quality. We have shown that small increases in surface hydrophobicity, easily obtained by adding small amounts of starch at the size press during the papermaking process, improve the print quality in these aberrant papers. With the success of the DeskJet printer, much of our recent work has focused on finding specifications for penetration rate that can be used by paper companies to help make their office papers conform to the needs of the DeskJet printer.
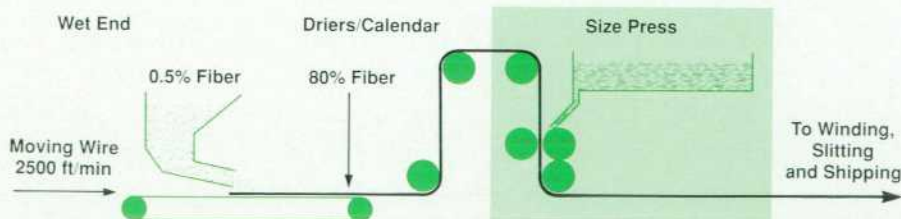
Fig. 5. *Schematic of the office paper production method. Increasing the hydrophobic character of the paper by means of additives in the wet end and starch from the size press has the most impact on DeskJet print quality.*
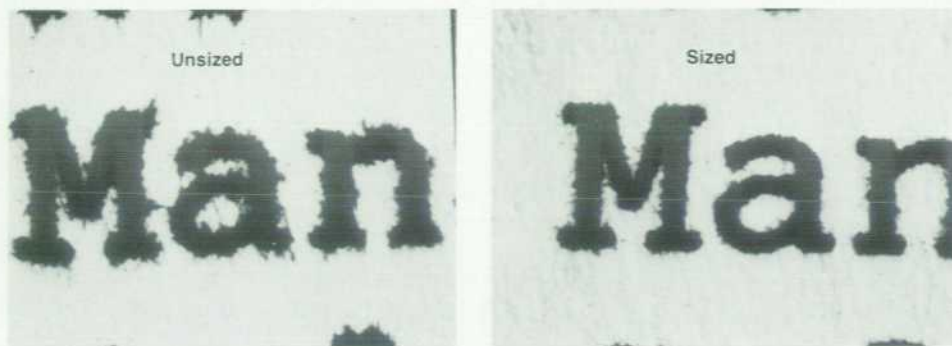
Fig. 6. *Photographs of print made on a standard office copier paper before and after starch addition at the size press.*

Ink Penetration Rates

Techniques for examining fluid penetration into papers were developed long ago in the paper industry. However, techniques for measuring penetration rates of inks into paper surfaces under the dynamic conditions pertinent to inkjet printing have only been developed in recent years. One technique for measuring such rapid penetration rates was introduced by Bristow in 1967.[4] Lyne and Aspler[5] demonstrated the applicability of this technique to inkjet printing in their 1984 study of the mechanisms of fluid penetration into paper surfaces. A commercial instrument based on the Bristow concept was introduced by Noram Instrument Company (Point Claire, Quebec) in 1984.

The measurement of the volume of fluid that penetrates into a paper surface in a limited time forms the basis of the Bristow technique. The instrument consists of a stationary ink headbox and a paper strip mounted on a rotating wheel, as shown in Fig. 7a. When the headbox and the paper touch, ink is transferred to the paper. The amount of ink that transfers is dependent on the contact time between the paper and the ink. The contact time is calculated from the tangential velocity of the rotating wheel and the width of the slit in the ink headbox. The quantity of liquid per unit area that transfers to the paper in time t can be described by:[3]

$$V(t) = V(t=0) + K_a t^{1/2} \qquad (1)$$

Here $V(t=0)$ is the amount of liquid that penetrates into the paper surface before absorption by capillary forces begins. K_a is the Lucas-Washburn penetration rate and contains all the ink physiochemical parameters, ink/paper interaction terms, and terms describing the paper's pore structure. To get the penetration rate K_a, ink is metered into the headbox and trace areas are measured at selected wheel speeds. This data is plotted on a square-root time scale and K_a is extracted by regression analysis.

Using the Bristow apparatus (Paprican Bristow Tester, Noram Instruments) and DeskJet ink as penetrant, penetration rates were measured on a wide variety of office papers. Examples of this data for three office papers are shown in Fig. 7b. These curves show that for short contact times, ink penetration does not occur. This initial delay time is observed only when considering penetration of aqueous fluids into papers. In this region, ink is transferred only because it fills the gaps in the paper surface. The intercept, therefore, is an alternate measure of the paper's bulk topology or smoothness. When the contact time is sufficiently long, penetration occurs with a rate proportional to the square root of the contact time, as given by equation 1. Penetration rates are calculated by least-squares regression of the data in the region after the delay time.

To correlate penetration rates of paper with print quality, reproducible standards for letter-quality print were needed. Unfortunately, quantitative methods to measure print quality in a fashion that correlates with consumer preferences are not yet available. Hewlett-Packard has attempted to design a reproducible, consumer-based rating system. To rate print quality, a committee of individuals considered
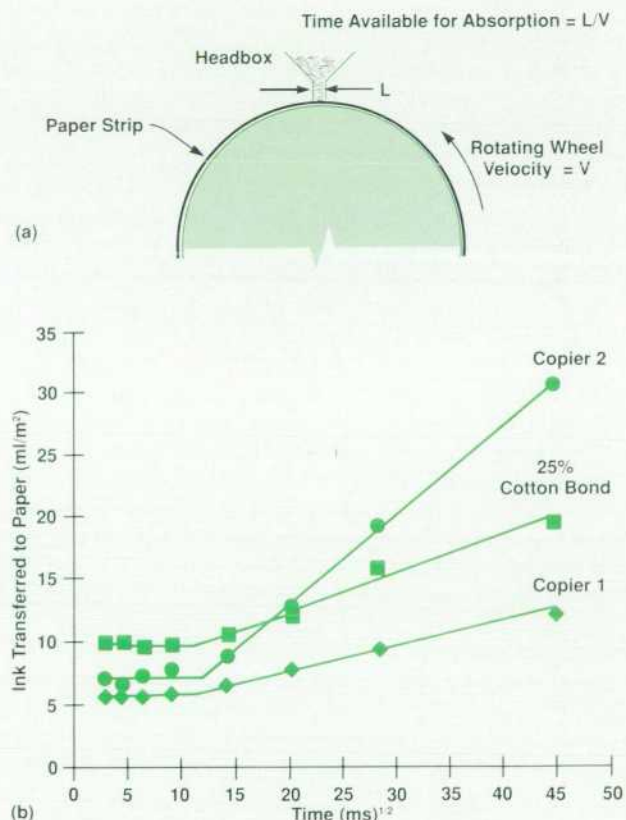
Fig. 7. *(a) Schematic diagram of the Bristow apparatus for measuring fluid penetration under dynamic conditions. (b) Penetration data for three representative office papers. Measurements were made using the Noram Bristow tester using ink from the DeskJet printer as penetrant. Penetration rates are obtained by least-squares regression of data in the positive-slope portion of each curve.*
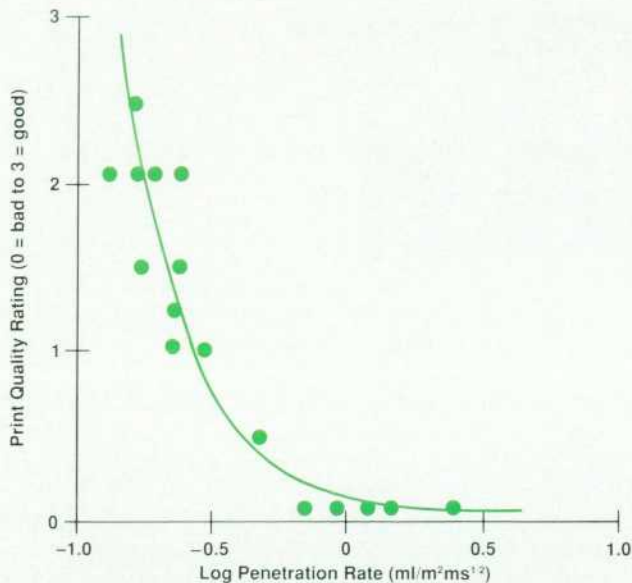
Fig. 8. *Print quality rating for DeskJet printer output on copier papers plotted versus the penetration rate obtained from the Bristow apparatus. The solid curve, shown for clarity, shows degenerating print quality with increasing ink penetration rate.*

to be typical office users is employed. Each committee member evaluates a sample's print quality on a scale of 0 to 3, with 0 = unacceptable, 1 = acceptable, 2 = good, and 3 = excellent. The print samples used for those tests incorporate both text and graphics. The print quality rating from the committee for any test sample is an average of the perceptions of the committee members. We have found that the results of this rating system are reliable. Rating errors are approximately ±0.5 for ratings above acceptable, and ±0.1 for papers below acceptable.

Fig. 8 shows the print quality ratings for sixteen standard copier papers plotted against their penetration rates obtained from the Bristow Tester. Copier papers were selected for this experiment because, in general, their look and feel are reasonably consistent. The sixteen copier papers chosen for this experiment were selected based on their uniformity of color, stiffness, opacity, and smoothness to eliminate such variables from the assessment of print quality. Rapid penetration rates yield large, irregularly shaped dots and poorer print quality as attested by print quality ratings less than one for such papers. Acceptable print quality was

observed in those papers with penetration rates below 0.4 $ml/m^2ms^{1/2}$. Because of the experimental error in the Bristow apparatus, penetration rates could not be used to distinguish acceptable from good and good from excellent papers reproducibly. However, papers with ink penetration rates above 0.4 $ml/m^2ms^{1/2}$ yield below-acceptable print quality (a rating less than 1) in all cases.

Summary

Simply put, we have learned that there is no such thing as plain paper. By linking timely market research into the design phase of the DeskJet printhead we were able to design a product that meets the difficult goal of printing on office papers. The variation of penetration rates of ink into office papers was identified to be the most difficult obstacle to overcome in designing the ink and printhead. The Bristow apparatus serves to specify the penetration rate and will help design teams make future products. With our technical studies of office papers complete, we are now in a position to help suppliers make papers that perform optimally on the DeskJet printer and future products.

Acknowledgments

References

1. *Hewlett-Packard Journal*, Vol. 39, no. 5, October 1988, pp. 51-98.
2. M.C. Huth, et al, "CIM and Machine Vision in the production of Thermal Inkjet Printheads," *ibid*, pp. 91-98.
3. M.B. Lyne, "Wetting and the Penetration of Aqueous Liquids," *Handbook of Physical and Mechanical Testing of Paper and Paperboard*, R.E. Mark, editor, Vol. 2, Marcel Dekker, Inc., 1984, Chapter 18.
4. J.A. Bristow, "Liquid Absorption into Paper during Short Time Intervals," *Svensk Papperstid*, Vol. 70, no. 19, 1967, p. 623.
5. M.B. Lyne and J.S. Aspler, "Wetting and the Sorption of Water by Paper under Dynamic Conditions," *TAPPI Journal*, Vol. 12, 1982.

Host Independent Microprocessor Development Systems

A new architecture makes it possible to use this family of emulators with workstations, mainframes, or personal computers. The cabling technology and chassis design inprove performance and usability.

by Arnold S. Berger

HP 64700 SERIES EMULATORS are a new family of products representing the latest stage in the evolution of the HP 64000 Microprocessor Development System, introduced in 1979. The original family of products, which is still going strong, was designed around a cluster of development stations sharing a local disc memory and printer. This cluster uses an optimized version of the HP-IB as its LAN core, and is a very efficient hardware and software integration environment for small engineering teams. In fact, many of today's concepts in distributed computing networks were first applied when the HP 64000 system was introduced.

As microprocessors became more complex, the development environment shifted to larger teams of software engineers working on small to medium-size mainframe computers, such as the Digital Equipment Corporation VAX™ and the HP 9000 Series 500. Hewlett-Packard responded to this trend with the introduction, in 1985, of the HP 64000 Hosted Development System. This system permits an existing HP 64000 cluster to be linked to the mainframe computer with the high-speed HP-IB file transfer capability, while the software cross-development tools remain resident on the host mainframe. Individual HP 64000 development stations can be operated under remote control via the terminals on the host machine.

However, the industry trends in software engineering were pointing towards more open software development environments. The large software development teams, now writing software for 32-bit microprocessors with address ranges measured in gigabytes, needed environments that could best be serviced by HP-UX (Hewlett-Packard's version of AT&T's UNIX® operating system), while the small teams were moving towards software development on HP Vectra-class computers.

Hewlett-Packard responded to the needs of the large teams with the introduction, in 1986, of the HP 64000-UX family of hardware and software-based products. Work was also in progress to develop a family of products that could satisfy the needs of the single engineer or small development team, without the proprietary technology of the original HP 64000 Logic Development System. We recognized that the IBM PC/AT and the Microsoft MS-DOS® operating system had become de facto computing standards for the engineering community and that a product offering was needed for these systems. In addition, we had the enviable position of being able to start with a clean slate, using the knowledge gained through the evolution of the original families of products.

Host Independence

During the investigation phase of this project, the entire design team went on the road to visit customers. Both our installed base of customers and customers who did not have any of the HP 64000 products were visited. In all, some 35 companies who design products containing embedded microprocessors were surveyed to determine the set of features that were considered most appropriate for a personal-computer-based microprocessor development system. The HP 64000-PC Personal Integration Environment, based on the HP 64700 Series emulators/analyzers, is the result of our effort to bring a microprocessor development system to the personal-computer-based engineering team.

Early in the HP 64700 product definition phase we realized that expanding the product definition to include the requirement of host independence would free the product from dependence on a particular computer or operating
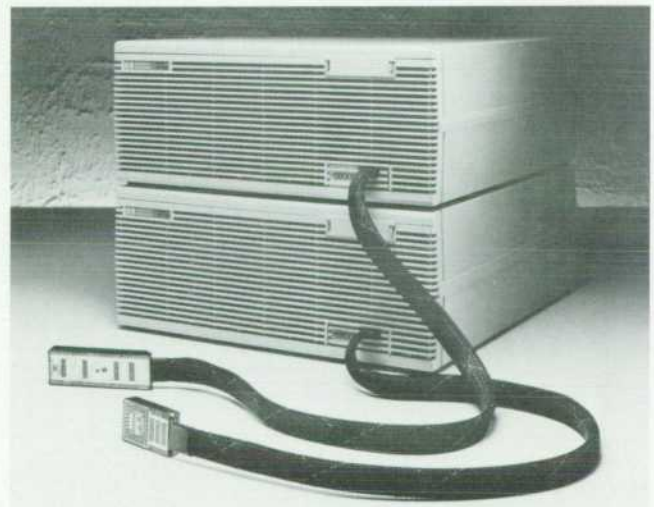
Fig. 1. *These HP 64742A Emulators for the MC68000 microprocessor are members of the HP 64700 Series. They are available with DIP or PGA cabling. HP 64700 emulators are host independent and can operate with workstations or personal computers.*

system and broaden the environment in which it could function effectively. At the lowest level, host independence implies a totally self-contained microprocessor emulation vehicle. The emulator contains its own local controller, which has sufficient on-board operating software to relieve the remote computer of all base-level control functions. All commands to the new HP 64700 Series emulators are constructed from simple ASCII character strings.

The emulator communicates with its controller over a standard-protocol, serial network. The world beyond its RS-232-D/RS-422 port is understood to be an 80-character-by-24-line "dumb" terminal. File transfers using industry standard formats are accepted. Thus, any host computer (or terminal) can control an HP 64700 Series product, making it useful for a variety of tasks, from code development through field service applications.

Host independence also means that the product can be used as easily with workstations such as the HP 9000 Series 350, with mainframe computers such as the DEC VAX, and with personal computers, such as the HP Vectra PC. Any cross-development software that produces an industry standard file format can create object files for use by HP 64700 systems.

The initial product offering consisted of four products: the HP 64742A/L 68000 Emulator, the HP 64753A/L Z80 Emulator, the HP 64764A/L 80186 Emulator, and the HP 64765A/L 80188 Emulator.

Follow-on products include emulators for the 8086/8088 microprocessor families and popular microcontrollers. Table I is a listing of microprocessors and microcontrollers supported at the time this article appears. Fig. 1 is a photo of the HP 64742A Emulator.

Table I
HP 64700 Series Emulators

Model	Processor	Type
64742	68000	General-Purpose 16-Bit
64745	68010	General-Purpose 16-Bit
64753	Z80	General-Purpose 8-Bit
64762	8086	General-Purpose 16-Bit
64763	8088	General-Purpose 16-Bit
64764	80186	General-Purpose 16-Bit
64765	80188	General-Purpose 16-Bit
64771	80C196	Microcontroller 16-Bit
64786	TMS 32020	Digital Signal Processor
64787	TMS 320C25	Digital Signal Processor
64764C	80C186	General-Purpose 16-Bit
64765C	80C188	General-Purpose 16-Bit
64731	V25	Microcontroller 16-Bit
64733	H16	General-Purpose 16-Bit
64735	647180X	Microcontroller 8-Bit

Emulation

Before going any further it might be instructive to review what an emulator is and how it operates. Fig. 2 is a block diagram of the major components of the emulation system. Every emulator contains a duplicate of the processor that is being emulated, which is called the target processor. The system being designed around the target processor is called the target system. The emulation processor resides on the

run control board within the emulator and is connected to the processor socket in the target system via line driver buffers and a target system cable. The emulation processor, typically a high-speed version of the target system processor, forms the core of the run control board. Surrounding the emulation processor are basically three functional circuit blocks:
- Run control
- Memory and memory mapping
- Analysis bus interface.

The run control circuitry controls the operation of the emulation processor so that the user can always maintain control of the emulator. Since each microprocessor has a different internal architecture, the run control circuitry must be tailored to provide the interface to the host control system. Run control affects operations such as single-stepping, sharing interrupt signals with the target system, signaling the analyzer, keeping the host controller aware of the status of the emulation processor, and so on.

The emulator contains its own internal emulation memory, which can be used as a substitute for the memory, RAM, and ROM that will eventually be in the target system. This memory can be located anywhere in the address space of the emulation processor. Mapping of the memory and its designation as RAM, ROM, or guarded are handled by the mapping circuitry, which provides a translation function between the address being output by the emulation processor and the address being supplied to the emulation memory array.

The analysis bus interface provides the link between the signals traversing the address, data, and status buses of the emulation processor, the emulation bus analyzer, and the state/timing analyzer. The analyzers will be discussed later in this article.

Two other system blocks, the host controller and the analysis system, are not shown in Fig. 2. The host controller works with the run control system, memory subsystem, and analysis system to provide the coordination of all the complex tasks going on within the emulator. It interfaces with the user to convert commands to proper operational
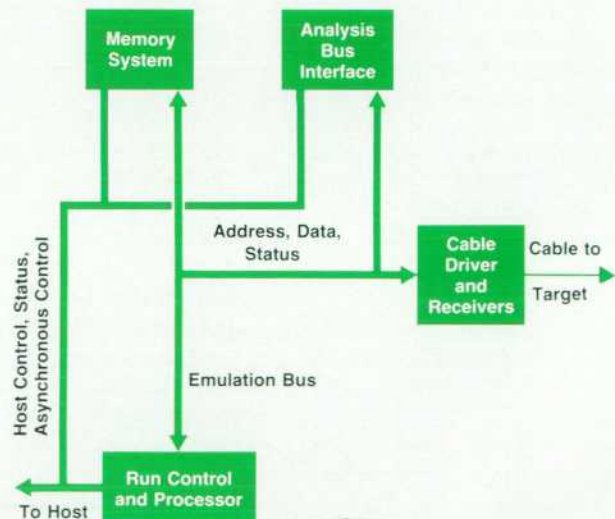
Fig. 2. *Major components of the emulation system.*

sequences. The host controller also interfaces independently with the state/timing analyzer, emulation bus analyzer, and memory subsystems. This enables the host controller to communicate with emulation memory and read and write to it while the emulation processor continues to run the user's code. Likewise, the state/timing analyzer or the emulation bus analyzer can be set up or unloaded without stopping the emulator. This type of architecture is called dual-bus, referring to the two independent processor buses (emulation and host control) within the emulator.

Design Goals

As stated earlier, the entire R&D team undertook an extensive series of customer visits. The purpose of these visits was to focus on the features that were most important to users of microprocessor emulation systems. The two most important features were ease of use and transparency.

Every engineer wants products that are easy to use. The key for a design team is to make the product operate in a straightforward manner without compromising the functionality needed to solve complex measurement problems. Functionality, at least for development systems, has two aspects. First, the emulator must provide a comprehensive set of system analysis and processor control functions. Second, as important as the feature set is the transparency of the emulator. An emulator is supposed to behave as if the target microprocessor chip is plugged into the socket in the user's target system. When the performance of the emulator differs from how the target microprocessor would behave, then the transparency of that emulator is compromised.

The HP 64700 emulators have the same high degree of transparency as their predecessors. For the initial HP 64700 products, in-circuit operation is full-speed, with zero wait states in the target system or emulation memory. For the Z80, this is 10 MHz, and for the 80186/88 and the 68000, 12 MHz.

Special care was taken to match the timing and parametric performance of each emulator to those of the target processor. Operation from power-on or system reset is consistent with the target processor chip. Overall, care was taken to see that when differences do occur, as they must in some design environments, the HP 64700 Series performs in a manner consistent with a user's expectations.

To meet the ease of use and transparency design goals

for the HP 64700 Series, it was necessary for the design team to address three critical areas: a new architecture for the emulation subsystem, a new chassis configuration, and new cabling technology.

System Architecture

Many possible architectures can be employed in emulator design. The HP 64700 Series uses a dual-bus architecture (Fig. 3), with a foreground or background monitor to permit the user to control the microprocessor emulation in the target system.

The dual-bus architecture has been the heart of the HP 64000 family of products since its introduction in 1979. What makes the architecture of the HP 64700 Series new is the addition of the hybrid monitor. The hybrid monitor, residing either in foreground or background, is a major departure from the original HP 64000 family design.

The monitor is the control program that the emulation microprocessor is running when it is not explicitly running the user's program. The monitor program is very similar to the small programs often supplied with single-board computers. Typically 2K to 3K bytes in size, the monitor handles reading and modifying memory, internal registers, I/O ports, and so on.

Single-stepping and software breakpoints are also handled by the monitor program. While the use of the monitor is easily understandable, its integration within the emulator plays an important role in the emulator's range of useful applications.

Complex applications typically require the use of a foreground monitor. This monitor is a block of code that runs in the same address space (foreground) as the user's own program. It is linked with the user's run-time code at assembly time so that when control of the emulation processor is switched from the user program to the monitor program, real-time system events can still be serviced. These events, such as multiple interrupts and watchdog timers, are very difficult to handle properly with other monitor implementations. The foreground monitor is often the architecture of choice for complex multitasking environments.

Background monitors are easier to use when trying to get a new design off the ground. The monitor is present in a background or shadow memory, and control is transferred to this monitor by remapping (called "jamming") the monitor instructions from the background address space directly to the emulation processor. This type of monitor
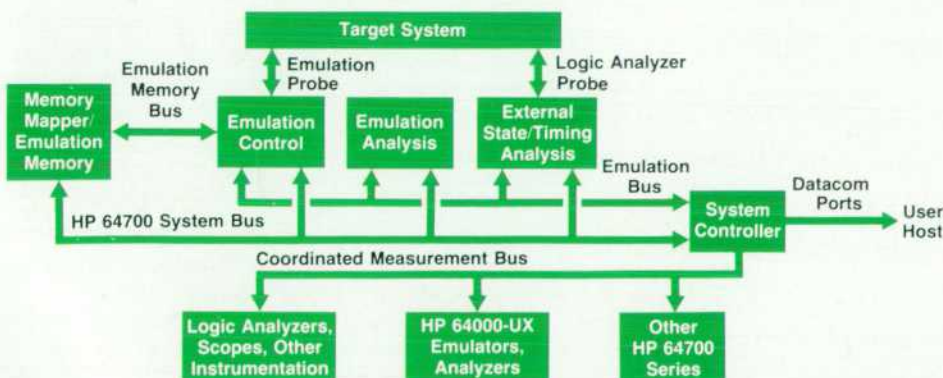
Fig. 3. HP 64700 emulator system architecture.

makes no assumptions about the user's environment. The emulation processor can always enter the monitor when instructed to do so. Problems arise from this architecture when it becomes necessary to keep a user's target system alive while the emulator is running in background. The new HP 64700 emulators, such as those for the 8018X and 68000 processors, solve this problem by offering the best of both worlds. Once the user answers a simple configuration question, either the emulation system will operate with the monitor in background, or by assembling and linking the monitor with the run-time code, the user will have the advantages of a foreground monitor. A more complete discussion of foreground and background monitors can be found in reference 1.

Cabling

In keeping with the twin goals of transparency and ease of use, the design of the cable to the target system became a major focus. The cable has to be small, light, flexible, and long, so the user can plug it into a target system where the microprocessor socket is not readily accessible. The signal that starts at the emulator and exits at the target system must look like the signal that would be present if the target processor chip itself were present. Since these cables experience a great deal of mechanical stress over the life of the product and its application in target systems, they have to be rugged.

We approached this design task in two ways. A member of the team did an experimental and theoretical study of the electrical performance needed for the emulator to meet its design goal. The objective was to make the signal fidelity in the cable essentially independent of the length of the cable. Another team was working with an outside cable manufacturer to realize the mechanical features needed in the cable.

The length of the cable is important because in the majority of real-world target systems, the microprocessor socket is not easily accessible to the emulation cable, and it isn't always possible, either mechanically or electrically, for the user to have a convenient extender card with the processor socket easily accessible. Since the transit time through the cable is set by the speed of light, the length of the cable for each emulator is set by the maximum operating speed of the processor being emulated and its timing requirements.

A cable design evolved that borrows heavily from existing oscilloscope technology. However, an oscilloscope typically has between two and eight signals to deal with. The Motorola 68000 microprocessor has 64 pins, of which roughly 60 must maintain high signal fidelity. The cable technology is considered proprietary, so further details cannot be given here. However, the results are easily demonstrated.

Fig. 4 illustrates the signal fidelity achievable with the new cable technology. Fig. 4a is the trace from a signal propagating along a standard ribbon-type cable, 12 inches long, of the type commonly used for connection to a target system. The upper trace is the source end and the lower trace is the exit end of the cable. It is clear that the lower trace exhibits significant undershoot and ringing. Signals such as this would probably lead to erroneous data and

inconsistent performance, since correct operation would depend upon how much design margin was available in the user's target system. Fig. 4b is a 1.5-meter length of the controlled-impedance cable used in the HP 64700 Series. The signal fidelity is vastly superior, although some degree of rise time degradation does occur. Since each cable used within the HP 64700 Series is optimized for full-speed operation of the processor in the target system, the effects of cable propagation delay and rise time degradation can be minimized. Also, the clock signals to and from the processor are buffered at the target end of the cable with a small, surface-mounted line driver circuit. In processors such as the Z80, 8086, and 68000, this is not absolutely essential since these processors do not have their own internal oscillator circuitry. The 80186 processor does have an on-chip oscillator that would not function at all if it needed to drive the added capacitance of the cable. In this case, the buffer circuitry provides the oscillator function and drives a clean signal back to the emulator. With this design a Z80 emulator can operate at 10 MHz with no wait states with a one-meter-long cable. The shortest cable of the initial products is 18 inches long, and is designed to be used with an 80186/80188 processor operating at clock speeds up to 16 MHz.

Achieving these levels of performance results in a cable that is expensive to manufacture and test. This would translate to customer dissatisfaction with the product if this

(a)

Ch. 1	=	2.000 volts/div	Offset	=	2.000 volts
Ch. 2	=	2.000 volts/div	Offset	=	0.000 volts
Timebase	=	10.0 nsec/div	Delay	=	30.000 nsec

(b)

Ch. 1	=	2.000 volts/div	Offset	=	2.000 volts
Ch. 2	=	2.000 volts/div	Offset	=	0.000 volts
Timebase	=	10.0 nsec/div	Delay	=	30.000 nsec
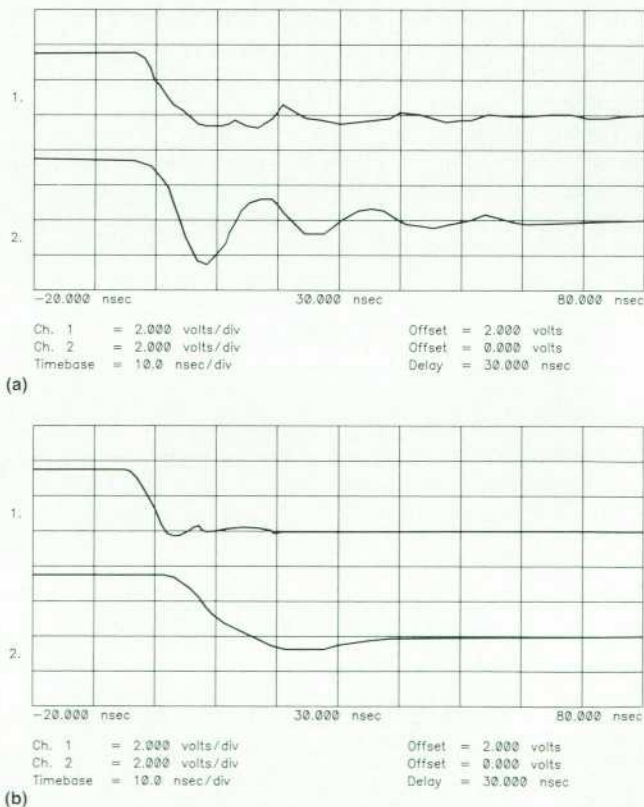
Fig. 4. (a) Signals at the source end (upper trace) and exit end (lower trace) of a standard ribbon-type cable 12 inches long. (b) Signals at the source end (upper trace) and exit end (lower trace) of a 1.5-meter length of the controlled-impedance cable used in the HP 64700 Series.

performance led to a mechanically fragile cable design. Considerable effort also went into making the cable mechanically rugged and flexible. During the product's development, a special test fixture was designed to flex the cable through thousands of bending cycles, typical of how these cables are stressed in real environments.

Chassis

During our customer visits we found that one of our customers had suspended an HP 64000 Development System from the ceiling by steel cables to get it close to the target system processor, which was located in the topmost circuit card in a six-foot-high equipment bay. The HP 64000 is about the size of a large workstation computer and this was obviously not optimal for mechanical, safety, and convenience reasons. Thus, we saw the need for a chassis that could be easily brought close to the system under development.

The chassis for HP 64700 Series emulators is designed for minimum size. Holes are placed around the perimeter of the chassis to hold two handles, which can be used for standing or suspending the emulator closer to the target system (see Fig. 5). These handles, dubbed the "skyhooks" by a member of the design team, permit the emulator to adjust to any desired mechanical configuration.

Data Transfer

Since the HP 64700 Series uses an industry-standard serial interface structure, the speed at which files and information can be exchanged between the emulator and the host computer is a significant issue. The HP 64700 can transmit and receive files at burst rates up to 450 kilobaud using the synchronous RS-422 channel (imbedded in an otherwise standard RS-232-D port) with hardware handshaking. However, raw bandwidth is useless unless it can be coupled with an equally fast method of loading the files into the processor's memory space. This file transfer link has been optimized so that a realizable system transfer rate in the range of 100 to 150 kilobaud is obtained. This includes the overhead of file formats and file error correction software. Actual file transfer rate measurements have been made. A 256K-byte file was repeatedly downloaded to the emulation memory of each of the HP 64700 emulators. Table II lists the transfer times for three of the initial products.

Table II

HP 64700 Series Emulator 256K-Byte File Transfer Time

Emulator	Time (seconds)
HP 64753A (Z80)	14
HP 64764A (80186)	20
HP 64742A (68000)	26

However, it should noted that this is a special 8-bit binary transfer mode. Files in ASCII or hexadecimal format would load at approximately one half to two thirds of these rates. Buffered interface circuits are currently being designed for use with the HP 9000 Series 300 and HP Vectra Personal Computers (IBM PC/AT compatibles) so that the maximum download speed capabilities can be realized.

Operating Modes

An additional RS-232-D interface is also included with the emulators. This second port allows the emulator to be operated, together with a terminal, from a single drop of a mainframe computer, or in conjunction with a PROM programmer. The HP 64700 emulator can be placed in a pass-through mode so that the terminal can effectively communicate with the host computer. When the appropriate escape sequences are given, the emulator will intercept and execute those commands intended for it. The transparent mode also allows several emulators to be serially connected to each other and to a single host control port. After each emulator is assigned a unique serial address code, it can then be individually controlled by the host.

An emulator can be used in a stand-alone mode, that is, without target hardware. Used this way, it is much like a single-board computer. The stand-alone mode is a very efficient way to develop the embedded code that the final product will eventually be controlled by, even though it may be too early in the design cycle for any real target system hardware. Eventually, of course, it becomes necessary to begin turning on the target system hardware and integrating it with the various software modules that have been and still are being developed for it. At this point the emulator must be able to plug into the target system and substitute for the microprocessor that will be used in the final product.

Analysis

An important feature of the HP 64700 Series is the power of its internal emulation bus analysis and external state and timing analysis. The heart of the analysis system is the logic analyzer on a chip used in the HP 165X family of logic analyzers.[2] Internal analysis of the emulation processor's address, data, and status buses is a standard feature, and an additional 16 external channels of 25-MHz state analysis and 100-MHz timing analysis are available as an option. The external channels can be either tightly or loosely coupled to the internal channels, depending

Fig. 5. Handles attach to the HP 64700 emulator chassis for standing or suspending the emulator close to the target system.

upon the requirements of the measurement.

The features and applications of the HP 64700 internal analyzer include:

- Eight levels of sequencing for complex program flow tracking
- Address, data, or status range resources
- Prestore queue for variable access tracking
- Time tagging for instruction execution measurements
- 1024-state-deep memory (512 states with time tagging)
- Powerful set of store qualification resources
- Selectable trigger resources
- Code coverage memory for reliability metrics.

Environments

An emulator is only one part of a microprocessor development system. Also required is a comprehensive software environment that will support the microprocessor of choice. The HP 64000-PC Personal Integration Environment provides the required environment. Since the hardware is host independent, any assembler or compiler that can output one of several industry-standard object file formats, such as Intel Hexadecimal File Format, Motorola Hexadecimal S-Record File Format, Extended Tektronix Hexadecimal File Format, and Hewlett-Packard Absolute File Format, can be used. However, to access the complete development system environment the user is encouraged to use software tools that also produce compatible symbol table information for use by HP-approved, high-level debuggers and HP-supported interfaces.

The supported interfaces, initially targeted for HP Vectra Personal Computers and HP 9000 Series 300 workstations, provide a window environment for code development and hardware/software integration. On the HP Vectra PC, the window environment is coupled with an inverted-tree command line structure. The inverted-tree structure is the interface most familar to users of the popular spreadsheet programs. This structure allows complex measurement sequences to be constructed from ever-descending levels of

options until the final structure of the command is established. At each level a brief explanation line is provided for each command option. The user selects the option possibilities using the cursor keys or the first letter of the command option. This type of interface is often called noun-driven, since only noun descriptors are used to traverse the tree.

Windows can be set up and customized for various information displays. Register information, memory contents, high-level source code, assembly code, and analyzer trace information can coexist in different windows and be called up for full display. In addition, a terminal window can be invoked. This window allows direct access to the host independent, ASCII command set of the HP 64700 system. Fig. 6 shows this user interface for an MC68000 emulator (HP 64742A).

One of the most challenging chores for the user is the proper structuring of the triggering, tracing, and storing conditions required to build complex measurement structures. The window interface opens up to reveal a straightforward menu construct. Simply filling in the menu fields automatically builds the analyzer's measurement sequence.

On HP 9000 Series 300 Computers the interface is modeled after the original HP 64000 syntax-directed softkey structure. This consistency permits an HP 64700 Series emulator to operate from the same environment as a companion HP 64000-UX emulator.

Multiprocessor Emulation

Multiprocessor systems are becoming the norm, rather than the exception, in the design of products with embedded microprocessors. Industrial robots, telecommunications networks, and aircraft are just a few examples of such applications. The HP 64700 Series contains a special measurement and control structure called the coordinated measurement bus, or CMB. The CMB interface is a 9-pin D-type connector on the rear of the instrument. Up to 32 emulators
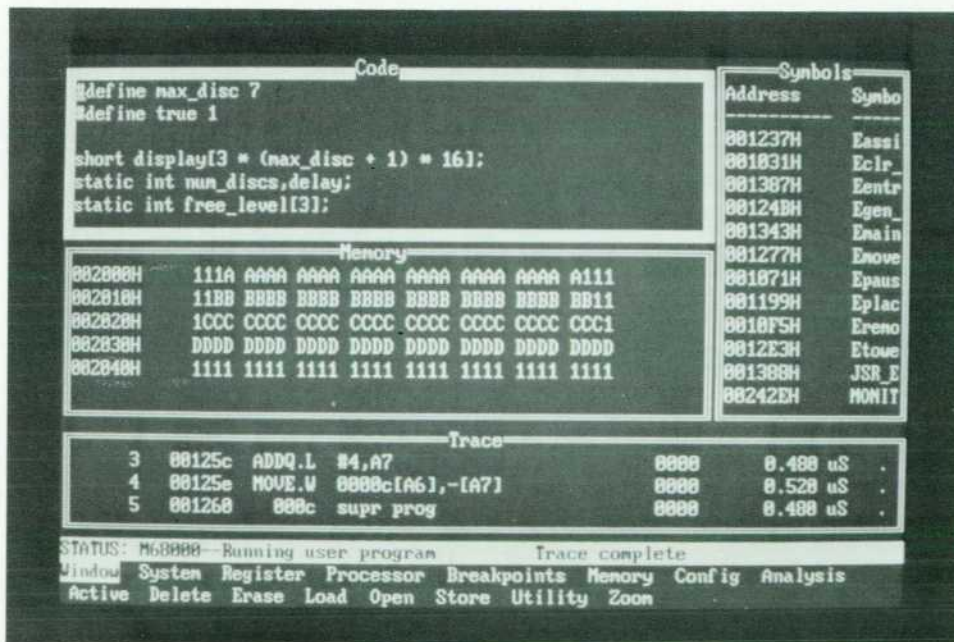
Fig. 6. *User interface for an HP 64742A Emulator, showing the window capability.*
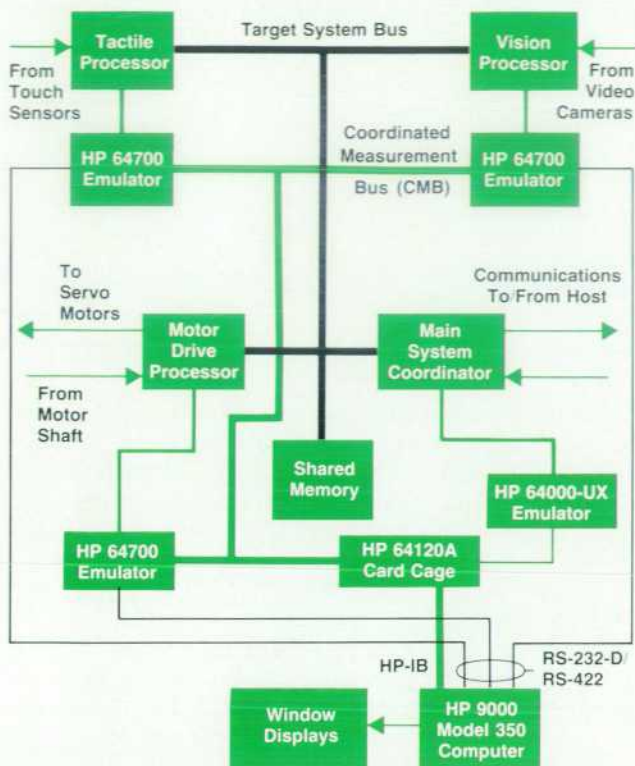
Fig. 7. *Emulation in a robotics application of a multiprocessor array.*

can be connected via this port, using inexpensive insulation-displacement ribbon cable, with a total separation distance of 50 meters. The CMB allows any grouping of emulators to participate in a coordinated measurement. This measurement can include synchronous starts and stops, internal or external analyzers coupled to multiple trigger sources, or triggering by activity occurring at another, remote emulator. Each emulator can be controlled by the same host computer or by independent hosts. Thus, for example, in a system with multiple processors accessing a common memory, fault conditions leading to an erroneous data transfer can be traced by triggering the analyzers of the two emulators at the detection of the fault, using the external state/timing analyzer to look at bus activity, and breaking the emulators to the background state to prevent any system or code corruption.

Fig. 7 is a simplified schematic diagram of one possible multiprocessor configuration, a complex industrial robotics application. The robot has communications links that allow it to share status and task programming information with other computers. Usually, satellite processors handle the peripheral chores, while a powerful central processor such as a 32-bit 68020 coordinates overall operation. In the example shown, each peripheral processor (such as an 80186) is replaced by a separate HP 64764A/L Emulator. These emulators are linked to each other via the CMB and to a 68020 emulator via the CMB/IMB link in the HP 64120A card cage. The entire system is controlled by an HP 9000 Model 350 workstation computer, and each emulator appears as a session on the high-resolution display. The HP 64120A card cage is linked to the Model 350 via the HP-IB (IEEE 488/IEC 625), and the HP 64700 emulators communicate via the RS-232-D/RS-422 serial link. With this configuration, some or all of the processors can be started or stopped simultaneously. Program flow events occurring in one part of the system can trigger analyzers somewhere else and perhaps shut down the system to prevent damage from, say, a robot arm out of control.

Acknowledgments

References

1. A.S. Berger, "Foreground/Background Monitor Increases Development-System Transparency," *Personal Engineering and Instrumentation News*, Vol. 5, no. 2, 1988, p. 39.
2. *Electronic Products*, August 15, 1987.

Host Independent Emulator Software Architecture

Built into the firmware of the HP 64700 Series host independent emulators is an entire microprocessor development system.

by William A. Fischer, Jr.

FOR A MICROPROCESSOR EMULATOR, the system software is a significant portion of the overall design. HP 64700 Series emulators have, for the first time, an entire microprocessor development system built into firmware. This firmware lays the groundwork for future emulator design.

When a silicon manufacturer releases a new microprocessor, it is important that emulation support be available in a timely fashion. The present HP 64000-UX Advanced Integration Environment offers more than 40 different microprocessor emulators. A flexible software architecture was developed for the HP 64700 Series that also supports many different emulators. This article will discuss the HP 64700 software architecture. It will be shown how the software architecture improves the emulation development environment and how users benefit with easy-to-use, flexible emulation interfaces.

Software Architecture Overview

The HP 64700 software has a layered architecture. The layers consist of the processor specific drivers, the generic firmware, the terminal interface, the RS-232-D/RS-422 communication channel, the programmatic interface, and the host interface (see Fig. 1). The software layers have well-defined interfaces that permit easy communication between layers. These interfaces facilitated design and implementation without unnecessary concern for the detailed design of the other software layers.

The system design started with the terminal interface. The requirement for host independence (see article, page 45) necessitated development of an easy-to-use interface that connects to a simple ASCII terminal. The needs of the terminal interface user were some of the key system design concerns.

The terminal interface is supported by the generic and processor-specific firmware layers. The generic firmware layer is used by all emulation products. The processor-specific layer customizes the emulation product for a specific microprocessor. The terminal interface firmware is primarily generic, but commands for specific processors can be added.

The user interface residing on the host computer makes use of the terminal interface to perform its functions. Some terminal interface commands use an option mode to improve system performance by reducing the information passed over the RS-232-D/RS-422 communication channel.

The programmatic interface is the layer between the terminal interface and the host user interface. It resides on the host computer and consists of a library of C function calls that access the entire functionality of the emulation system. The programmatic interface is ported to multiple hosts and allows connection of user interfaces of different flavors including high-level software debuggers.

Terminal Interface

The terminal interface provides for communication between the emulation system and the outside world. Terminal interface commands are designed to be atomic, each performing only one function. Command names are one to five characters in length and form easily remembered mnemonic abbreviations of the emulation functions. Command options provide extra flexibility for the command functions. Command operands are used when additional data is required by the function.

For example, to display registers the simple command reg is used. With no operands, the command displays the entire register set. A single register can be displayed by following the command with an operand, the register name, as in reg pc. To modify a register the command might be used as follows: reg pc = 100h.

The terminal interface command set consists of nearly 100 commands. Almost half of these commands are used to access the functionality of the emulation analyzer (see article, page 45). The other commands provide functionality
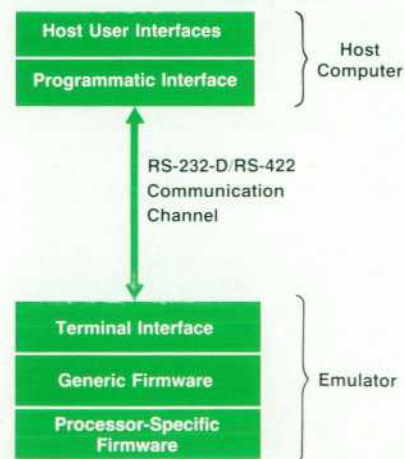
Fig. 1. *The HP 64700 emulator software has a layered architecture.*

for memory display, software breakpoints, emulation memory and target memory loading, stepping, running, and system functions.

To help the user with the sheer number of commands that must be remembered, the HP 64700 terminal interface contains an extensive help system. The help system provides detailed descriptions and uses for all commands. User access to the commands is simple and hierarchical so that users can work their way from generic help, through a listing of command groups, and finally to more thorough help for an individual command. See Fig. 2 for a display of help commands.

Terminal interface system commands are also available for displaying the time and date, controlling the communication port, and running performance verification tests. The HP 64700 Series also allows the user to define system macros, groupings of commands that can be executed by a user-defined name.

Firmware Structure

The firmware consists of generic core code and a set of processor-specific drivers and functions. This division allows easy partitioning of firmware tasks. The generic core code is universal. Only the processor-specific functions are developed for each new emulator. This minimizes the overall development time for a new microprocessor emulator.

The goals of the firmware design were to create maximum flexibility in the generic core code, total isolation of the processor-specific functions, and minimization of the processor-specific functions. Maximum flexibility allows the system to accommodate emulation of most conceivable types of processors. The total isolation of processor-specific

```
R>help

  help  - display help information

    help <group>      - print help for desired group
    help -s <group>   - print short help for desired group
    help <command>    - print help for desired command
    help              - print this help screen

  --- VALID <group> NAMES ---
    gram     - system grammar
    proc     - processor specific grammar

    sys      - system commands
    emul     - emulation commands
    *        - all command groups


R>help emul

  emul - emulation commands
  ---------------------------------------------------------------------------
  b......break to monitor  cov....coverage        mo.....modes
  bc.....break condition   cp.....copy memory      r......run user code
  bnct...bnct signal       dump...dump memory      reg....registers
  bp.....breakpoints       es.....emulation status rst....reset
  cf.....configuration     io.....input/output     rx.....run at CMB execute
  cim....copy target image load...load emul memory s......step
  cmb....CMB interaction   m......memory           ser....search memory
  cmbt...cmbt signal       map....memory mapper     x......emit CMB execute


R>help m

  m - display or modify processor memory space

    m <addr>                 - display memory at address
    m -d<dtype> <addr>       - display memory at address with display option
    m <addr>..<addr>         - display memory in specified address range
    m -dm <addr>..<addr>     - display memory mnemonics in specified range
    m <addr>..               - display 128 byte block starting at address A
    m <addr>=<value>         - modify memory at address to <value>
    m -d<dtype> <addr>=<value>   - modify memory with display option
    m <addr>=<value>,<value>     - modify memory to data sequence
    m <addr>..<addr>=<value>,<value> - fill range with repeating sequence

  --- VALID <dtype> MODE OPTIONS ---
    w - display size is 2 byte(s)
    b - display size is 1 byte(s)
    l - display size is 4 byte(s)
    m - display processor mnemonics
```

Fig. 2. *A typical HP 64700 hierarchical help facility display.*

functions means that the firmware for a given emulator resides by itself in ROM on the processor-specific hardware board and that the functions are separately compiled. Most of the functionality of the firmware is in the generic code, reducing the quantity of processor-specific code needed for a new emulator.

The generic commands can be extended. An emulator may need a special command not developed in the original design. The processor-specific firmware can insert a new command into the generic command set or a host interface can download new commands via RS-232-D/RS-422. This makes the firmware extremely flexible for future but yet unknown emulation requirements.

Separating the generic core functionality from the processor-specific functionality means that the basic emulation functionality is handled by the generic core code, which makes calls to low-level drivers that are resident in the emulator ROM. The generic core code handles the parsing of commands and the basic emulator control. Calls are made to the processor-specific drivers whenever communication with the emulation hardware is needed to complete a function. For example, the user enters a command to display memory (m 0..20h). The generic core code parses the command and passes the ASCII string (0..20h) to an emulation function that converts the string to an actual address. The generic core code then passes the address to a processor-specific function that interacts with the emulation hardware to read memory and return the memory values. The generic core code will then display the value in the proper format.

The processor-specific functions are accessed through a table consisting of function calls. Other tables provide information about processor-specific registers, configuration items, the memory mapper, and error code information.

The data structures are also separated into generic and processor-specific pieces of information. Because of processor differences, each requires processor-specific data structures. Some of these processor differences include address, data, and registers. Functions that must differ for each emulator include emulation status, configuration, and miscellaneous functions.

Address

Physical address ranges differ widely with processor types. For example, the Zilog Z80 microprocessor has a 16-bit address range, the Intel 8086 has a 20-bit address range, and the Motorola MC68000 has a 24-bit address range. In addition, address mnemonics differ. The Intel family uses a segment:offset structure while the Motorola family uses a linear address space. Also, some processors like the MC68000 and the TMS 320C25 use the concept of

multiple address spaces. For example, the MC68000 uses function codes to distinguish between user and supervisor space. These function codes become a logical extension to the address. See Fig. 3 for specific examples.

To handle the varying complexities of processor addresses, the concept of an address object was developed. An address object is a processor-specific structure that represents a location in the processor's address space. The address object structure contains all the pertinent information about that processor's address (see Fig. 4). The generic firmware never handles a specific address type but instead only passes pointers to these address objects. The lower-level processor-specific drivers interpret the address, translating ASCII string to address object, address object to ASCII string, address object to physical address, physical address to address object, or program counter to address object as appropriate. Adding an integer to an address object is another lower-level driver function.

Data

Data sizes also vary between processors. The MC68000 can access data lengths of 1, 2, or 4-byte words. The Intel 8086 can access data as 1 or 2 bytes but the data must be displayed in a swapped order compared with the MC68000. Since most processors access data as multiples of single bytes, data input and display are handled generically as an array of bytes. The ordering of the byte data for display and interpretation is left up to the processor-specific functions to specify for the particular processor being emulated.

It is also important to control the type of cycle that is used to access the data. The 8086, for example, can access data as bytes or words, and it may be necessary for peripheral devices to see the proper type of data access. The mode (mo) command is used to select the access mode for data. The options of this command are processor-specific and are handled in the drivers that access the data. These processor-specific access modes are defined in one of the processor-specific tables.

Similarly, it is important to display the data in the form that is most understandable to the user. Typical ways of displaying data are in bytes, words, or mnemonics. Again, the mode command is used to set the default for these processor-specific attributes. Alternatively, commands such as memory (m) and search memory (ser) permit the user to input an override option to change the display mode for that command and the default.

Registers

Every processor has its own set of registers. The register names vary, as do their lengths and functions. In addition, the registers may be read only, write only, or read-write.

```
Processor        Address Example      Address Type

Z80              1000                 linear
MC68000          10000@sp             linear with function codes
80186            1000:23a5            segment:offset
TMS320C25        1000@p               linear with function codes

note: @ indicates function code follows
      sp = supervisor program space for the MC68000
      p = program space for TMS320C25
```

Fig. 3. *Address representations for various processors.*

Some can reside in memory and others in I/O space.

Processor-specific tables are created for each emulator that fully define the valid register names, sizes, read/write status, and register grouping information.

Configuration

Each emulator requires a different set of configuration parameters. These configure the emulation hardware to provide the type of functionality that the user wants. Users need to decide if they want to use the target system clock or the internal emulation clock, or they might need to switch between a foreground monitor and a background monitor (see article, page 45). These configuration parameters are extremely important for normal functionality, and provide the flexibility for an emulator to work in a wide variety of target systems.

Each emulator establishes its own table of function calls for configuration parameters. The table contains the configuration name, the function to be called for that configuration, and the configuration help functions.

Status

An emulator must be able to control the transition to various states and determine the state in which it is currently residing. An emulator has three basic states: running user code, running in monitor, and static reset. The method each emulator uses to enter these various states can vary greatly from a simple toggle of a control bit located at a defined address in the controller's memory space to multiple writes to the hardware.

Determining the current status of an emulator may be as simple as reading a single hardware status bit or as complex as interrogating the monitor for detailed information. Processor-specific functions control the state transitions and read the current system state.

```
Address Object Length = 3
Word 1 = high 8 bits of address
Word 2 = low 16 bits of address
Word 3 = function code
```

Fig. 4. *Address object structure for the MC68000 processor.*

Miscellaneous

Each processor has its own instruction mnemonics. An inverse assembler is provided in firmware for each emulator. Also, functions are provided for loading the emulation monitor, the code that actually communicates with the target processor.

Host Interfaces

Host interfaces are programs resident on the host computer that provide the user's view of the HP 64700 Series emulators. Because emulation control firmware is resident in ROM on the emulator, it is reasonable and efficient to develop multiple host interfaces for each emulator.

Currently, host interfaces are supported for two different hosts: the HP 9000 Series 300 and the HP Vectra PC and IBM PC/AT compatibles. Two different interfaces are supported on the Series 300: an emulator interface using the same syntax-directed softkey design as the original HP 64000 and a new high-level software debugger. The interface on the HP Vectra PC provides a friendly emulation interface with a flexible windowing display system (see Fig. 6 on page 50). A timing analyzer interface is provided on the HP Vectra PC using a graphical display (see Fig. 5). All host interfaces provide a syntax-directed command structure, displays of emulator output, and an integrated symbolic package.

Programmatic Interface

The programmatic interface is the starting point for all host software. The programmatic interface is a library of C


```
Sample Period [100 ns] 1.000 us/div       510.00 ns x to time trig
Magnification [ 4x]      magnify about [x]  0.0 us x to o
cursor moves  [x]                           10.00 ns/clk
```

STATUS: M68000 -- Running in Monitor   Trace Complete
Display   Begin   Halt   Spec   System   Utility
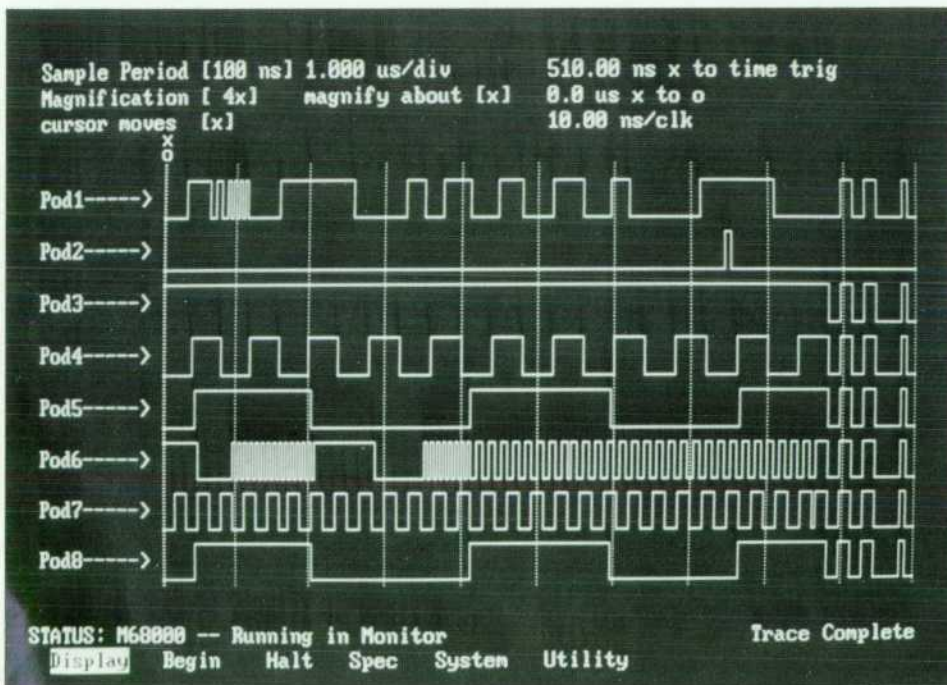
Fig. 5. *HP 64700 timing analyzer interface for the HP Vectra Personal Computer.*

function calls. A host interface program gains access to the emulation system via these calls.

The programmatic interface provides functionality for memory and I/O accesses, run control, register display and modification, software breakpoints, the coordinated measurement bus, and error message control. The programmatic interface hides the emulator's functional idiosyncrasies from the host interfaces. Thus, the designers of host interfaces do not need to understand the complete emulator behavior, greatly simplifying their task.

The programmatic interface also handles command and error message synchronization. The HP 64700 emulators will only respond with information such as data, an error message, or a status message in response to an entered terminal command. Updates of information are also returned after a null command (a simple carriage return). The only time the user receives information without requesting it is on power-up, when the copyright message and emulator hardware configuration and firmware revisions are sent. This methodology greatly simplifies the handling of information returned from the emulator.

The information returned from the HP 64700 Series can be either synchronous or asynchronous. A memory command, for example, returns the contents of the requested memory locations. If an invalid address is entered, an error message is generated and displayed. Both of these information types are synchronous with the last command entered.

A software breakpoint command (bp) establishes a condition that could result in the generation of an asynchronous message. The software breakpoint command replaces an instruction in the target memory space with the processor's software trap instruction. When that trap instruction is executed in the user's program, the control of the target processor is transferred to the emulation monitor program. The emulation control firmware, executing on the control processor, determines that a software breakpoint occurred, and a message is displayed indicating this. The software breakpoint will most likely occur long after the original software breakpoint command was entered. Because of this, the software breakpoint message will be returned with data from another command. This type of message is termed asynchronous.

The status and error commands are separated into synchronous and asynchronous buffers by the programmatic interface. Host interfaces can access these two buffers for display and error evaluation purposes.

The programmatic interface, like the firmware, uses the concept of an address object. The processor-specific dependencies are manipulated by a library that together with an expression evaluator and a symbol package evaluates all expressions of address objects. This permits the programmatic software to handle the expression in a generic fashion.

The programmatic interface also provides substantial control of the HP 64700 emulation analyzer. The terminal interface handles analyzer control with a large set of elemental commands. Simple analyzer control is easy, but setting up a complex analysis sequence requires the user to allocate all the analyzer resources manually. This requires an intimate understanding of the analyzer hardware and its programming model. The programmatic interface uses an expression tree with a validation function to determine if the analyzer contains enough resources to perform the measurement.

The programmatic interface also includes the communication port drivers. For MS-DOS these port drivers control the standard RS-232-D/RS-422 port on the HP Vectra PC and provide high-speed communication at up to 460 kilobaud using a new buffered interface card (HP 64037A). A communication port configuration file is read to associate a logical emulator name with a physical communication port on the host computer.

The PC Interface

The host user interface for PCs provides the user with a friendly windowing environment. Commands are entered by single keystrokes or command selection using the cursor keys. The PC interface takes the commands from the user and builds a set of data structures. These data structures are passed to the programmatic interface. The programmatic interface validates the data structures and generates the appropriate terminal interface commands. The commands are passed to the port drivers which send them to the HP 64700 emulator via an RS-232-D/RS-422 communication channel.

The data generated by the HP 64700 Series is received via the RS-232-D/RS-422 drivers. It is then processed by the programmatic interface and the data is put into data structures. The PC interface translates the data structures into ASCII for display in one of the windows.

Summary

The layered software interface for the HP 64700 emulators has succeeded in reducing the development time for new emulation products. The software contains well-defined interfaces that simplify software communication between layers. The terminal interface layer provides a host independent interface that is also used by all the host interfaces. Incorporating the entire emulation functionality in firmware has permitted easy development of more than one host interface, providing an emulation system that can be used as a hardware/software integration tool, a high-level software debugger, or a timing analyzer.

Acknowledgments

Expanded Memory for the HP Vectra ES Personal Computer

This memory subsystem provides high-performance expanded memory and extended memory support for HP Vectra Personal Computer applications while maintaining compatibility with industry standards.

by Gary W. Lum, Milton J. Lau, and Wesley H. Stelter

THE VECTRA ES EXPANDED MEMORY CARD is a memory subsystem consisting of a hardware card and a software driver for the HP Vectra ES Personal Computer. The key design objectives for this subsystem were to:
- Conform to industry standards, including LIM EMS 3.2 and 4.0 (see box, page 61)
- Provide a high-performance expanded memory system to support Microsoft® Windows and the NewWave environment
- Provide a high-performance memory system to support extended memory applications (see box, page 62, for an explanation of extended and expanded memory).

The design challenge was to provide superior performance and yet remain compatible. Both LIM EMS 4.0 and 3.2 define a protocol for using expanded memory, but EMS 3.2 includes support of function calls 10 and 11, which dictate a specific memory architecture, and several major applications make use of them. These software functions define not only the memory cards' I/O port addresses, but the I/O data bit assignments as well. The memory card had to conform to these definitions—a high-performance memory system that was not register-compatible with the LIM EMS 3.2 specification would not meet our objectives. How does one contribute within such a strict definition of compatibility?

The project team began by addressing the level of compatibility required by end-users. Clearly, full support of LIM EMS 4.0 was a requirement for Microsoft Windows, and because some major industry applications use LIM EMS 3.2 functions 10 and 11, supporting these was also necessary. Extended memory support was required for support of new operating systems, such as OS/2. None of these applications specifies a particular mechanical configuration of a memory card, or a maximum CPU or memory clock speed. Research showed that customers wanted at least 1M bytes of additional memory and required any upgrades to be user-installable. Industry trends showed that a maximum configuration of 6M to 8M bytes would be competitive in the next generation of memory cards. Therefore, we required full compatibility with LIM EMS 3.2 and 4.0, but had some freedom in the mechanical configuration, memory configuration, and electrical performance of the card.

Given these constraints, the ES expanded memory card

definition evolved into a card that provides the following features:
- 100% compatibility with LIM EMS 4.0 and 3.2
- Daughter board configuration that does not take up a standard Vectra I/O slot.
- Up to 8M bytes of expanded or extended memory
- User-installable memory upgrades
- All 8M bytes of memory running at the same memory speed as the Vectra ES motherboard memory
- EMS 4.0-specific hardware for superior expanded memory performance.

Each of these features adds capability to the Vectra ES Computer, yet does not compromise the required compatibility models. A proprietary motherboard/memory-card interface was defined that makes the 8M-byte memory array appear to the 80286 CPU as motherboard memory. Since the new LIM EMS 4.0 function calls do not specify a hardware architecture, proprietary high-performance circuitry was designed to assist these calls. Surface mount logic ICs and the latest DRAM technology allow both the new logic and an 8M-byte memory array to fit on a single card.

Hardware Architecture

The ES expanded memory card is partitioned into three major blocks (Fig. 1): a single 8M-byte DRAM array, extended memory logic, and expanded memory logic. The extended and expanded logic blocks are two independent memory systems. Separate extended and expanded mem-
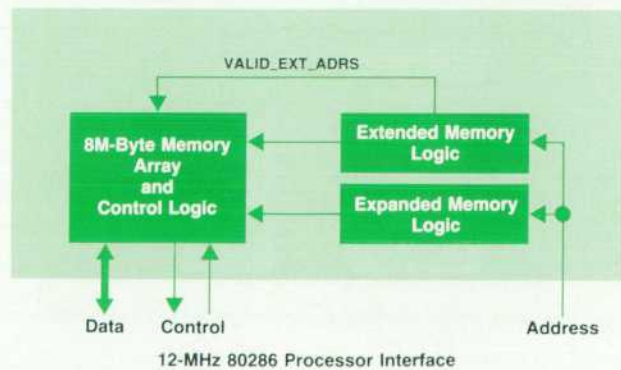
Fig. 1. *Block diagram of the Vectra ES expanded memory card.*

ory addresses are generated each memory cycle, and one is chosen depending on the type of cycle (defined by the state of the highest microprocessor address lines).

The 8M-byte DRAM array is organized as four independent 2M-byte banks (the reason for this is explained later). Each bank is 18 bits wide—16 bits of data and 2 parity bits—which allows a minimum memory configuration of 2M bytes, with upgrades in 2M-byte increments. On-board logic handles all timing and refreshing for the memory array. Each bank can be assigned to either expanded or extended memory via DIP switches, so the user can specify a combination of both kinds of memory.

The extended memory logic is straightforward (Fig. 2). Comparators and DIP switches define a starting address and an ending address on any 1M-byte boundary. These signals generate VALID_EXT_ADRS, which is used in a lookup table to determine which of the four physical memory banks is used. Any memory bank not assigned in the lookup table is considered to be expanded memory, allowing a combination of both extended and expanded memory. DIP switches define all extended memory parameters, since extended memory must be defined at system power-up, before any software driver can configure the card.

The expanded memory logic, shown in Fig. 3, is considerably more complex. Four logically independent banks of eight map tables each,* bank-switching logic, and a set of I/O registers make up this block. The I/O registers are used to configure the bank-switching logic and update map table contents. Once the map tables are programmed, they begin managing the expanded memory pages whenever memory addresses are presented to the board.

Since the card is register-compatible with EMS 3.2, both the I/O port locations and the bit definitions were predefined. To provide EMS 4.0 with the full mapping capability it requires (it allows programs to address up to 32M bytes of expanded memory), a map table of 64 entries was defined. Each entry corresponds to a 16K-byte block in the MS®-DOS 1M-byte address space. With this structure, any open 16K-byte block can be mapped with expanded memory, which gives applications the ability to manipulate much larger amounts of expanded memory than the 64K bytes available in EMS 3.2. When expanded memory is used to backfill user memory, an operating environment can manage several concurrent applications by simply

*As explained later, there are eight tables per bank so the card can run eight tasks before requiring software mapping.
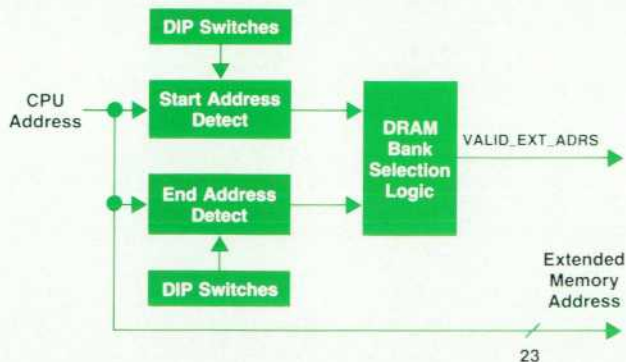
mapping in or out additional pages, rather than moving programs off to disc. Special I/O decoding is used to place the EMS 3.2-compatible I/O ports at their required addresses, while making the additional 60 EMS 4.0-compatible map table entries available via proprietary I/O ports.

Map Table Operation

A map table entry consists of a page enable bit and a 7-bit page offset. The page enable bit determines whether the current table entry is active. It is necessary when multiple expanded memory cards are present in a system. The page offset selects 1 of 128 16K-byte expanded memory blocks, which limits each card to a maximum of 2M bytes. The basic operation of the map table is shown in Fig. 4. When a memory address is presented to the map table, the appropriate entry is selected. If the page enable bit is set, then the corresponding expanded memory page is accessed; otherwise, the bank-switching logic will not respond. In a system that has two or more expanded memory cards, only one page enable bit can be set for a given 16K-byte block; otherwise, memory contention occurs.

Because we wanted to support 8M bytes, we needed a way to select one of the possible 512 16K-byte expanded memory blocks. The most straightforward approach was to define the page offset to be 9 bits, which would select 1 of 512 expanded memory blocks.

However, this would make the I/O registers incompatible with LIM EMS 3.2. Therefore, we decided upon a more complex addressing scheme, which makes the card look like four 2M-byte expanded memory cards, as shown in Fig. 5. Each of the four memory banks is defined as a logical LIM EMS 3.2 board, each capable of addressing its respective 128 16K-byte expanded memory blocks. The bank-switching logic simulates the parallel memory decoding of four boards and generates a single address to the 8M-byte

Fig. 2. Extended memory logic.
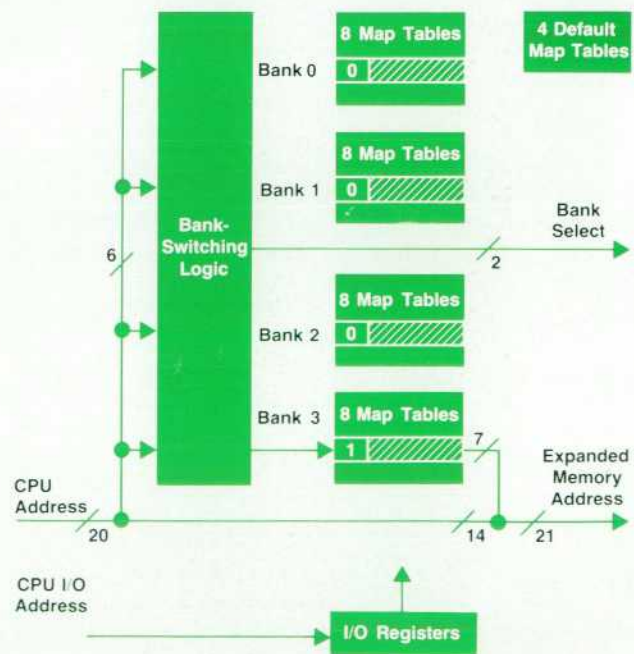
Fig. 3. Expanded memory logic.

memory array. To any expanded memory application, the ES expanded memory card looks like four independent 2M-byte memory cards.

To improve EMS 4.0 performance within the EMS 3.2 compatibility model, new circuitry was developed. The resulting capability is a superset of EMS 3.2, incorporating multiple map tables, independent control of these map tables, and a microprocessor interface that allows 12-MHz memory operation anywhere within the 16M-byte address range. Each of these capabilities is usable by software drivers that know of their existence, such as the Vectra ES expanded memory manager. Applications that write directly to I/O registers are not affected by the additional circuitry.

Multiple map tables work on the same principle as microprocessor register sets. Rather than storing in memory an image of an active register set when invoking a new task, additional copies of the register sets are available, and can

be selected using a single instruction. Using data from the HP Windows development group, it was decided to implement eight map tables, allowing the card to run eight tasks before requiring software mapping. Unique to bank 0 on the expanded memory card are four additional default map tables. These nonvolatile map tables are used at power-on, and store different configurations of backfilled memory. DIP switches are used to select the appropriate backfill configuration. All map tables are controlled by two proprietary I/O registers, MTACTIVE and MTACCESS. MTACTIVE defines which map table is used at execution time. MTACCESS defines which map table is accessible, and therefore updated, through the appropriate (and EMS 3.2-compatible) I/O registers. The independent control allows a current task to continue executing while a new task is being set up.

The Vectra ES motherboard memory is a 16-bit-wide array operating at 12 MHz, with one wait state (249-ns cycle time). The objective of the ES expanded memory card
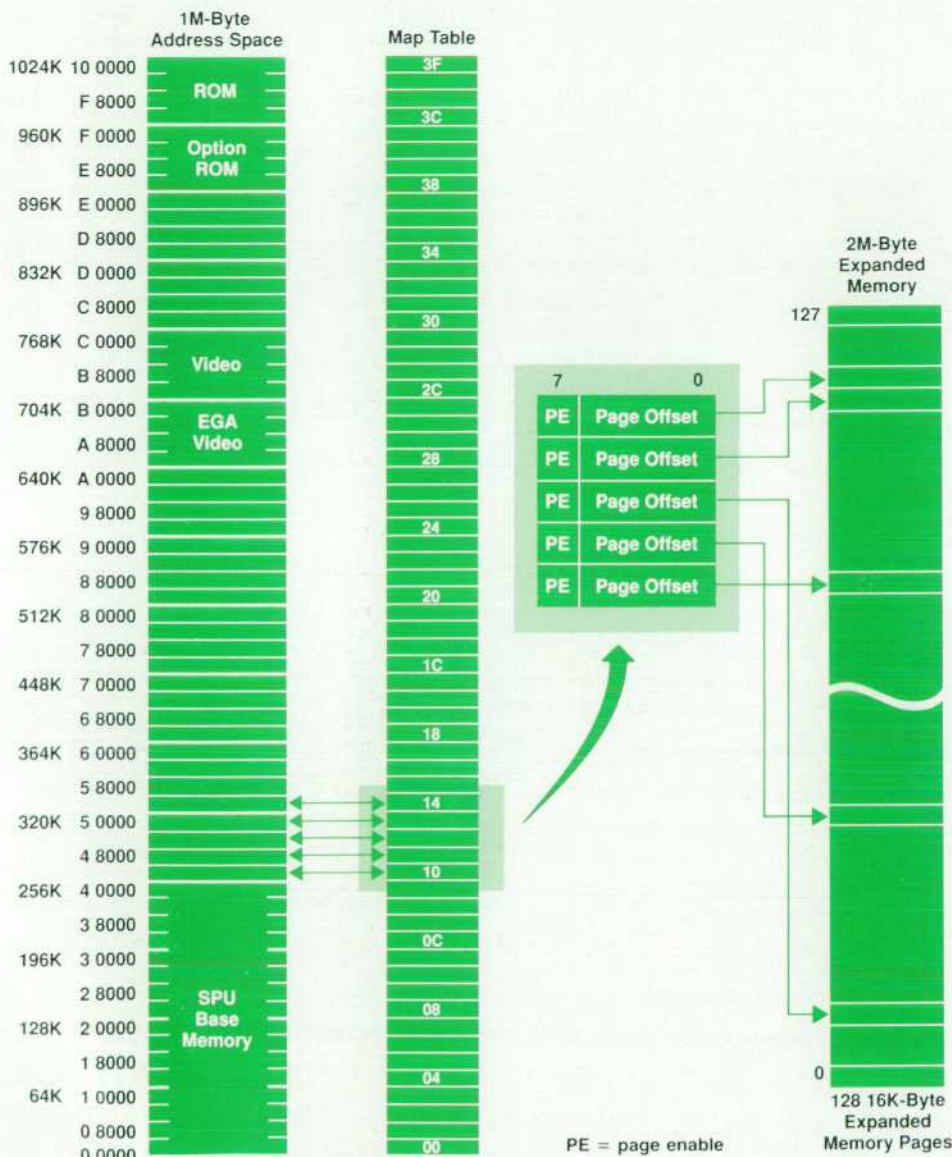
Fig. 4. Map table operation.

was to operate its 8M-byte memory array at the same speed and memory width. Because of timing limitations in the ES/12's industry-standard backplane, all memory accesses are limited to 8 MHz with one wait state. Furthermore, backplane memory can only be accessed as 16-bit words if an entire 128K-byte address range is programmed for 16-bit accesses, again because of timing restrictions on the industry-standard backplane. Because much of the 640K-to-1M-byte DOS address range runs as 8-bit accesses, a 128K-byte block may not exist, and consequently all backplane memory accesses may only be 8-bit. Quick calculations showed that 8-bit accesses at 8 MHz with one wait state would not meet our performance goals. It became clear that the ES expanded memory card could not operate over the standard backplane, and a new proprietary connector was defined which allows access to the local 80286 buses and control signals. The particular signals and buses used are:

- The local 80286 data bus to provide 20 ns of additional data setup time for a write to the expanded memory card.
- The local 80286 address bus to provide decoding 40 ns earlier than available over the backplane. It is mainly access to this bus that makes 16-bit, 12-MHz memory operation possible.
- Memory commands (MEM_READ, MEM_WRITE) directly from the 80286 to generate memory timing signals (RAS, CAS). This allows standard 100-ns DRAM to be used.

The use of local 80286 buses required new circuitry to be added to the ES/12 motherboard. Because both motherboard memory and the expanded memory card reside on the local buses, direction control of data and address buffers needed to be modified when memory accesses were initiated by either DMA devices or backplane masters.

Hardware Implementation

The Vectra ES expanded memory card is roughly the size of a standard Vectra I/O card. Two proprietary edge connectors are intentionally offset from standard backplane connectors so users are mechanically prevented from installing the ES expanded memory card in a standard I/O slot.

The 8M-byte memory is implemented using 1M-byte single in-line memory modules (SIMMs). Each SIMM has nine 1M × 1-bit, 100-ns CMOS DRAMs mounted on a small printed circuit board. The supporting circuitry is implemented in discrete TTL and PALs. High-speed (25-ns) SRAM is used to implement the 2048 map table entries required for expanded memory. The nonvolatile map tables are stored in high-speed PROM. Surface mount technology is used wherever possible. Roughly 85% of the components are surface mounted.

Software Driver Architecture

The Vectra ES expanded memory manager is a software driver. It must be loaded before the expanded memory card can be used. It is designed to be 100% compatible with the LIM 3.2 and LIM 4.0 specifications.

All code was developed in 80286 assembly language under the MS-DOS environment. The code is written as a standard character device driver and is installed during normal initialization of MS-DOS. Although it does not perform any character I/O operations, it is identified with a character device name, EMMXXXX0, and remains as a typical terminate, stay resident program. The size of the resident code is dynamic and varies with the amount of expanded memory available in the memory pool. This achieves efficient use of conventional memory to allow other huge applications to run. After initialization, the typical size of the memory manager is about 8K bytes when managing 4M bytes of expanded memory.

The memory manager consists of three separate pieces of code:
- Initialization routines
- LIM 3.2 memory management functions
- LIM 4.0 memory management functions.

Initializing the expanded memory card requires five tasks. The first is scanning the 1M-byte address range for

Fig. 5. An application's view of the ES expanded memory card.
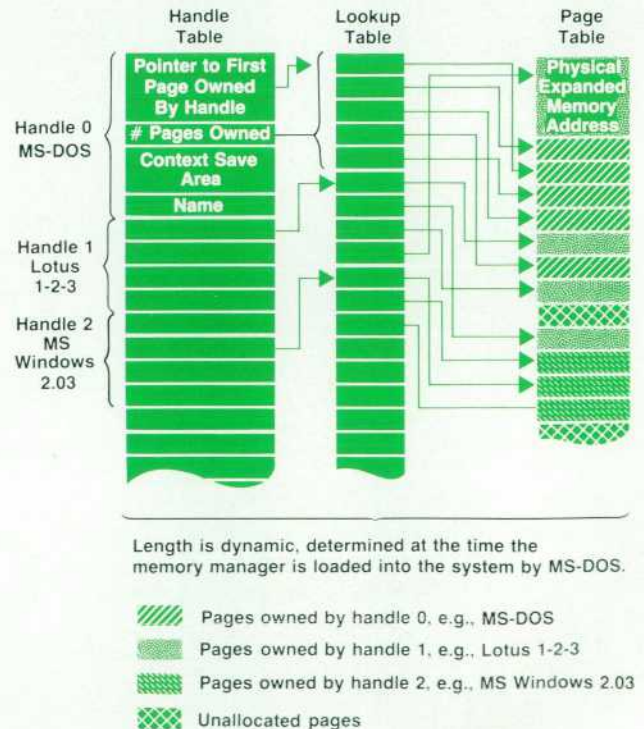
Length is dynamic, determined at the time the memory manager is loaded into the system by MS-DOS.

Pages owned by handle 0, e.g., MS-DOS

Pages owned by handle 1, e.g., Lotus 1-2-3

Pages owned by handle 2, e.g., MS Windows 2.03

Unallocated pages

Fig. 6. Memory manager data structures.

unused 16K-byte blocks. These empty spaces are areas in which pages of expanded memory can be mapped. The spaces need not be contiguous and every available space from the end of base memory through option ROM is scanned. The algorithm uses a nondestructive read/write scheme to check for RAM and the industry-standard AA55 checksum for ROM. In cases where known areas of memory space are being used for memory mapped I/O, an exclude option on the driver command line prevents searching these blocks.

The second initialization task is setting up the map tables. To allow expanded memory to be mapped into various 16K-byte holes, the hardware needs to be told what pages can be mapped. This is achieved by writing the expanded memory page number into a map table. On power-up the default map table is read from PROM and copied into one of the eight SRAM map tables. During this procedure, the expanded memory pages used to backfill MS-DOS memory from 256K bytes to 640K bytes are marked and excluded from the the expanded memory pool.

The third initialization task is identifying memory types. Conventional memory on the expanded memory card is identified by reading the default map table. The remaining memory is either extended or expanded, and is determined by using an overlapping algorithm. An ID byte is written on every 16K-byte interval in extended memory. Then, using expanded memory, every 16K-byte page is read. If the expanded memory page contains the extended memory ID byte, then the page is extended memory. Pages being used as extended memory are marked so they do not get allocated into the memory pool. The read/write test for extended memory is nondestructive so that any extended memory applications such as VDISK.SYS will not be destroyed.

The fourth initialization task is a memory test. The RAM test is used to verify the memory's read/write ability. Any 16K-byte page that contains a bad memory location is marked and not entered in the memory pool. This procedure is also necessary during power-on to initialize the DRAM parity checking circuitry.

Data Structure

The final initialization task is data structure setup. The memory manager uses three data structures to control the expanded memory pool (see Fig. 6). These are the handle table, the lookup table, and the page table.

The handle table is used to store information about handles and the pages they own. A handle is merely a token used by the memory manager to identify an entry in this table. When applications request pages, a handle is returned back to them so that when future requests to manipulate the pages are made, the owner of the pages can be identified. One handle is returned for every request to allocate pages from the memory manager. Each entry consists of a pointer to the first page owned by the handle, the number of pages owned by the handle, the context save area, and the name of the handle (8 bytes of ASCII).

The lookup table contains the N consecutive logical page translation mappings that belong to a handle. Each element is considered a logical page and contains a pointer to a page table entry or a null entry if unused.

The page table contain two fields. The first field is used

to indicate that the page has been allocated and the second field contains the physical address of the expanded memory page being used. The mapping of a page is finished when this address is written into an entry in the currently active map table.

The handle table is adjustable and can be programmed by the user to allow up to 64 handles. The lookup and page tables are dynamically determined by the amount of expanded memory available. The more expanded memory added to the card, the larger the data structure. This approach eliminates wasted space for unused entries because of nonexistent expanded memory. Since the majority of memory manager requests are allocating, reallocating, deallocating, and mapping expanded memory, a data structure is needed that handles the requests in an efficient manner. At first a linked list was considered because it provides a fast means of reallocating and deallocating memory pages; however, it would not be efficient for mapping pages. An important realization was that almost all memory manager activities after an application has been allocated expanded memory involve mapping those pages so the application can use them. Therefore, a lookup table was chosen over a linked list because it is much faster for translating and mapping logical pages. Speed in mapping pages

---

LIM EMS 3.2 and 4.0

Lotus Development Corporation, Intel Corporation, and Microsoft Corporation jointly defined the original *LIM Expanded Memory Specification* (EMS) in 1984. This specification, at revision 3.2, allows programs with an EMS 3.2 driver to use up to 8M bytes of expanded memory for data storage. LIM EMS 3.2 defines a rather limited bank-switching scheme, allowing only four 16K-byte blocks of expanded memory to be present at any one time. These four 16K-byte blocks, or pages, must be contiguous, meaning that a single 64K-byte open space must be available for bank switching. This 64K-byte page window must reside above user memory, typically at segments C800H, CC00H, D000H, and D400H. RAM discs, print spoolers, and certain programs (such as Lotus® 1-2-3® or AutoCAD™) make use of this additional storage. Early versions of LIM EMS include software function calls that allow an application to write directly to the I/O ports of the expanded memory card. These calls are not documented in revision 3.2, but they are supported.

In August 1987, LIM EMS 4.0 was announced. With 4.0, several limitations of version 3.2 were removed. EMS 4.0 allows programs to address up to 32M bytes of expanded memory, code as well as data to be stored in expanded memory, and the functionality of multitasking, which allows multiple applications, RAM discs, and spoolers to run simultaneously in expanded memory. Any open 16K-byte block in the 1M-byte address space can be filled with expanded memory.

EMS 4.0 can run on expanded memory cards designed only for EMS 3.2; only a new driver is required. Such a system, however, will not make use of all the power of EMS 4.0, because the new function calls can be accessed only by hardware cards that have specific EMS 4.0 circuitry, such as backfill capability or multiple map tables. Map tables are the tool used by the expanded memory manager driver to manage the bank switching of expanded memory.

is a priority, since operating systems, device drivers, and interrupt subroutines also make calls to the memory manager. Consider the following algorithm to map a logical page using a linked list data structure:

```
while ( lookup→logical_page != logical_page ) do {
        lookup = lookup→next /* traverse */
}
/* map the page */
out( MAP_TABLE, page_table[lookup→logical_page])
```

The translation from logical page to physical address would have required an average of 64 traversals down the linked list data structure for a 2M-byte card. This number would double for every 2M bytes added to the system.

Using the lookup table data structure, the algorithm is reduced to:

```
/* map a page */
out( MAP_TABLE, page_table[ lookup[ logical_page ] ] )
```

The translation from logical page to physical address requires only two reads.

Expanded versus Extended Memory

Three kinds of memory can exist in an MS-DOS personal computer: conventional, extended, and expanded. The physical memory devices used are identical in all cases. Only the access mode differs.

Conventional memory resides in the linear address space of an Intel microprocessor (8088, 8086, 80286, or 80386) running MS-DOS. This memory can be accessed directly by the 20 address lines of an 8088/6. Of this 1M-byte address space, the lowest 640K bytes is allocated as conventional memory (also referred to as user or base memory). Memory space between 640K bytes and 1M bytes is designated by the computer architecture for video, hard disc, option ROMs, and other system utilities, but there are normally areas in this 384K-byte memory that are not used.

Extended memory is the linearly mapped memory above the 1M-byte memory space. In a virtual-mode 80286 or 80386, up to 15M bytes of extended memory can be accessed. Because MS-DOS does not support virtual mode, extended memory is limited to specially written utilities such as RAM discs and disc caches. An 8088/6 cannot access any extended memory.

Expanded memory is a method that allows any Intel microprocessor running MS-DOS to access more than 1M bytes of memory. This is accomplished by bank switching, in which 16K-byte blocks (or pages) of expanded memory are mapped into some or all of the empty spaces that occur in the 1M-byte memory space. An expanded memory specification (EMS) describes the specific software calls that manipulate the bank switching.

Most personal computers allow a portion of their base memory (which normally resides on the motherboard) to be disabled, usually from 256K bytes to 640K bytes. Although the disabled memory is inaccessible, it allows the more powerful expanded memory to replace, or backfill, the disabled base memory. This capability is particularly useful for supporting LIM EMS 4.0.

Expanded Memory Access

The LIM 4.0 functions provide an interface for operating systems and/or applications to access expanded memory. Each function is optimized for speed. The memory manager is also reentrant so device drivers and interrupt service routines can use expanded memory. An application requesting service from the memory manager would follow these steps:

1. Request from the memory manager M logical pages of expanded memory. The memory manager will scan the page table entries for unallocated pages. When they are found the lookup table is updated with M consecutive entries pointing to the allocated pages. The number of pages requested and a pointer to the first logical page are stored in the handle table. A handle representing an entry to the handle table is returned to the application.

2. Request the page frame address. There are always at least four consecutive 16K-byte page frame windows available for mapping in the 1M-byte address space (to maintain compatibility with LIM 3.2). The memory manager will return the segment address of the first 16K-byte page frame.

3. Request to save the current page map. Since other applications may have already mapped expanded memory into the page frame, the current mapping must be preserved. The mapping is saved in the context save area of the handle table.

4. Request to map a page. The application will give the memory manager a handle, a logical page, and a physical page. From the handle and logical page, a physical address of the expanded memory page will be translated. The map table entry given by the physical page will be written with this physical address, thus completing the mapping.

5. At this time the application can now read, write, and/or execute code at the page frame address where the page is currently mapped.

6. Request to unsave a page mapping. When the application has completed its task with the mapped page it was using, it must replace the original page mapping before leaving. The memory manager will replace the old mappings that were stored in the context save area of the handle table.

7. Request to deallocate pages. The memory manager will deallocate pages belonging to the application by resetting the allocated bit for each page in the page table. The lookup table entries will then be cleared and adjusted for the deallocated pages. Each active handle will have its pointer to the first logical page entry adjusted to the lookup table. Other applications can then reuse the expanded memory pages returned to the pool.

Performance Data

The data collected on ES expanded memory card compatibility and performance has been excellent. We have not found any application or utility using LIM EMS 3.2 or 4.0 that does not run on the ES expanded memory card. In comparisons of a Vectra ES/12 with the ES expanded memory card and an ES/12 with a conventional expanded memory card, the following data has been taken:

Load Time from Disc to Memory
(seconds)

Application	ES Expanded Memory Card	Conventional Card
Windows Paint	4.3	8.1
Lotus® 1-2-3®	3.9	5.8

Given these results, we believe we have met our initial design objectives. The performance differences in the above data are attributable to the ES card's 16-bit memory access anywhere in the address space, 12-MHz operation anywhere in the address space, use of the eight hardware map tables, and optimized driver design.

Conclusions

The development of products for the industry-standard marketplace is deceivingly difficult. Because "industry compatible" is not always a cleanly written specification, part of the design process is determining what aspects of industry compatibility need to be considered. The challenge for product developers is to identify where contributions can be made within the nebulous compatibility model and then provide added value without compromising that model. The ES expanded memory card, with its high-speed memory system, large memory size, and adherence to industry standards, is an example of added value within the industry-standard marketplace.

References

1. *Lotus/Intel/Microsoft Expanded Memory Specification*, Version 4.0, October 1987.
2. T. Mirecki, "Expandable memory," *PC Tech Journal*, Vol. 4, no. 2, February 1986.
3. J. A. Lefor and K. Lund, "Reaching into expanded memory," *PC Tech Journal*, Vol. 5, no. 5, May 1987.

HEWLETT-PACKARD JOURNAL

Volume 39 January 1988 through December 1988

PART 1: Chronological Index

PART 2: Subject Index

PART 3: Product Index

PART 4: Author Index

Mayotte, Mark E.	Feb.	Proehl, Kraig A.	June	Steinfield, Steven W.	Aug.
McElfresh, David K.	Aug.			Stelter, Wesley H.	Dec.
McClelland, Don	Oct.	Quist, Robert D.	Oct.	Steranka, Frank M.	Aug.
McMahon, William T.	Oct.			Stoffel, John	Aug.
Melton, Gerod C.	June	Randel, Annette	Oct.		
Meyer, Jeffrey W.	Apr.	Rasmussen, Steve O.	Oct.	Ta, Chuong Cam	Aug.
Millard, Joe K.	June	Read, Sherry L.	June	Tesler, Joel D.	Oct.
Miranda, Rafael F.	Apr.	Read, Timothy R.C.	June	Thysell, Peter A.	Apr.
Montijo, B. Allen	June	Reese, Paul S.	Feb.	Togami, Chris	Oct.
Morris, Peter C.	Aug.	Regazzi, John R.	Apr.	Trego, Mick	Aug.
Mostafa, Hatem E.	Aug.	Regelson, Elaine C.	Feb.	Tribolet, David C.	Aug.
Motta, Ricardo J.	Aug.	Rehder, Wulf D.	June	Trueba, Kenneth E.	Oct.
		Rhoads, W. Wistar	Aug.	Tu, Chin-Wang	Aug.
Nazarathy, Moshe	Dec.	Rhodes, John D.	Oct.		
Neff, David J.	Oct.	Richter, Kenneth A.	Apr.	Unkrich, Mark A.	Feb.
Nelson, Nancy W.	Feb.	Rivoir, Jochen	Dec.	Upham, Herbert L.	Apr.
Nevarez, Ruben	Aug.	Roessler, Peter M.	Aug.		
Newton, Steven A.	Dec.	Rosenberg, Steven T.	Aug.	Van de Poll, Richard R.	Oct.
Nichols, Claude W.	Oct.	Rudaz, Serge L.	Aug.	Vobis, Joachim	Dec.
Nichols, Douglas C.	Feb.	Ruder, Jim L.	Aug.		
Nicol, Robert	Oct.	Ruska, David W.	June	Wagner, Kimberly S.	Oct.
Nigro, Stephen J.	Aug.	Russon, Virgil K.	June	Walmsley, Simcoe, Jr.	Feb.
				Walsh, William J.	Aug.
O'Day, Vicki	Aug.	Sagun, Wilfredo T.	Apr.	Wang, Scott W.	Oct.
Oldenburg, Glen	Aug.	Sayed, Mohamed M.	Apr.	Ward, Jeff	Oct.
Olsson, Kjell A.	Apr.	Scheffelin, Joseph E.	Aug.	Watkins, Marvin L.	Apr.
		Schultz, Philip C.	Aug.	Watson, Paul B.	Apr.
Page, Richard W.	Feb.	Scott, Perry E.	Oct.	Way, Peter	June
Palmer, Donald J.	Aug.	Selensky, Ronald J.	Aug.	Widder, John A.	Oct.
Peetz, Bruce E.	Feb.	Sischka, Franz	Dec.	Wield, P. Jeffrey	Aug.
Perlmutter, Paul Q.	Oct.	Skarke, Donald D.	June	Wiesmeier, Ed	Aug.
Pinkernell, David W.	Oct.	Smith, Andrew N.	Apr.		
Pless, Wilfried	Dec.	Smith, James C.	Aug.	Yamada, Ronald T.	Apr.
Plouse, Lynn M.	June	Snyder, Wayne L.	Aug.	Yano, Bruce	Oct.
Pon, Rayman W.	Feb.	Sorden, James L.	Feb.	Yee, Grace T.	Apr.
Poulton, Ken	June	Sperry, William R.	Aug.	Young, Ronald W.	Feb.
Pritchard, Thomas B.	Oct.	Splitter, Randolph N.	Aug.		

Authors

6 ⎯ Advanced Optical TDR

Franz Sischka

Born in Stuttgart, West Germany, Franz Sischka earned his Diplom Ingenieur degree (1979) and doctoral degree (1984) from the University of Stuttgart. After completing his studies in 1979, he joined the Institute of Network and System Theory of the University. During this time, he worked on a new and easily calculable modeling scheme for bipolar transistors and on the design of low-power amplifiers. Franz joined HP in 1984 and since has focused his professional interests on fiber optic instruments. He was a project leader for the HP 8145A OTDR.

Michael Fleischer-Reumann

With HP since 1980, Michael Fleischer-Reumann is an R&D project manager in the lightwave section of the Böblingen Instrument Division. He has been project manager or project leader for a number of fiber optics products, including the HP 8152A Optical Average Power Meter, HP 8154B LED Source, and HP 8158B Optical Attenuator, and he contributed to the hardware design of the HP 8112A 50-MHz Pulse Generator. A patent describing pulse generator timing is based on his ideas. In 1986, he assumed resonsibility for OTDRs. Michael has written or coauthored several previous articles for the HP Journal. He was born in Essen, Germany, and received his Diplom Ingenieur degree from the Ruhr University of Bochum. He and his wife live in Gechingen. In his spare time, Michael teaches electronics at a Stuttgart college. He also enjoys backpacking, mountain hiking and bicycling, white-water kayaking, and playing guitar.

14 ⎯ Complementary Correlation OTDR

Franz Sischka

Author's biography appears elsewhere in this section.

Moshe Nazarathy

Moshe Nazarathy received his BSc and DSc degrees in electrical engineering from the Technion Institute of Technology, Israel. In the ensuing two years, he held a postdoctoral appointment with the Information Systems Laboratory at Stanford University. Moshe came to HP in 1984 and joined the staff of the instrument and photonics laboratory. His design contributions to the HP 8145A OTDR include the signal

processing algorithms. A number of other projects he has worked on include fiber optic signal processing circuits and test instrumentation, solid-state lasers, and ultrafast optoelectronic modulation and sampling techniques.

Steven A. Newton

As a project manager at HP Laboratories, Steve Newton directed the design contributions of his organization to the OTDR project and helped invent the instrument's signal processing system. Born in Teaneck, New Jersey, he received his BS degree in physics from the University of Massachusetts (1976), and his MS (1978) and PhD (1983) degrees in applied physics from Stanford University. He came to HP on a part-time basis in 1978 and worked on optical design, optical data storage, and metal vapor lasers. Since he joined the HP Laboratories staff full-time in 1983, Steve has focused on research on single-mode fiber components, circuits, and measurement systems. He is now project manager of the fiber optics team in the instruments and photonics laboratory. He is named inventor or coinventor on seven patents and has authored or coauthored over 30 papers and articles. Steve and his wife live in Belmont, California. His outside interests include sports and music.

22 ⎯ Optical TDR Components

Siegfried Gross

A project engineer in the fiber optics development section of the Böblingen Instrument Division, Sigi Gross joined HP in 1984. As a member of the team that developed the HP 8145A OTDR, his main responsibility was design of the laser driver circuits. His Diplom Ingenieur degree is from the Berufsakademie Stuttgart. He, his wife, and his small daughter live in Sindelfingen, West Germany. His hobbies include several kinds of sports, choir music, and photography.

Robin Giffard

A native of the island of Guernsey, U.K., Robin Giffard received his undergraduate and graduate degrees in physics from Oxford University. Since he came to HP in 1980 as a development engineer, he has worked on a variety of projects, including trapped-ion frequency standards and laser interferometer circuitry. On the HP 8145A OTDR, his design objectives focused on the receiver chain. Robin is a resident of Los Altos, California, is married, and has two small children. In his off-hours, he enjoys reading and music.

Jürgen Beck

Born in Tübingen, West Germany, Jürgen Beck joined HP's Böblingen Instrument Division in 1984. He first performed experimental investigations of the optical system for the HP 8145A OTDR, using the results for his diploma thesis. After receiving his Diplom Ingenieur degree from the University of Stuttgart, he continued work on the OTDR project, developing the mechanical configuration of the instrument. He later became responsible for the optical system. Jürgen, his wife, and his son live in Tübingen. Among his favorite pastimes are traveling, skiing, sailing, and riding his motorcycle.

29 ⎯ OTDR Data Processing

Wilfried Pless

After receiving his Diplom Ingenieur degree in 1983 from the Ruhr University in Bochum, West Germany, Willy Pless joined HP as an R&D engineer. His first assignment was with a team developing the HP 8151A Optical Pulse Power Meter, one of HP's first fiber optics instruments. His main responsibility in the development of the HP 8145A OTDR included the signal processing software. Willy was born in Westphalia, is married, and has three children. He is building a house, which monopolizes most of his spare time.

Jochen Rivoir

Jochen Rivoir received his Diplom Ingenieur degree in 1983, after studying electronics engineering with the emphasis on system design at the University of Karlsruhe. He came to HP in 1985. On the HP 8145A OTDR project, Jochen's varied contributions encompass the processor circuit board, the DSPE circuit board, and firmware elements. He enjoys swimming, skiing, singing, and playing guitar, but a newborn daughter demands all his and his wife's attention.

35 ⎯ OTDR User Interface

Joachim Vobis

Joachim Vobis is a native of Heidelberg, West Germany. He joined HP after receiving his Diplom Ingenieur degree from the University of Karlsruhe. His initial assignments included the design of fast analog circuits for photonic instruments, but he soon transferred to a team of software designers where

he assumed responsibility for the entire firmware package for the HP 8145A OTDR. Joachim is married and has a son and two daughters. Among his leisure activities are swimming, hiking, backpacking, and amateur radio.

39 ═══ Plain Paper Printing ═══

Steven J. Bares

Steve Bares carried the engineering responsibilities for the technical studies of plain papers reported in this issue of the HP Journal and acted as liaison to marketing for all plain-paper market research. He continues to be involved in ink developments for future inkjet printers and in matching HP's inkjet products with the capabilities of the paper industry. Before coming to HP in 1985, Steve worked on experimental and theoretical studies of laser desorption and high-power laser/surface interactions in a postdoctoral appointment at Exxon Research and Engineering Company. His BS degree in chemistry is from the California State University at Humboldt (1979), and his PhD degree in physical chemistry (1983) and his MBA degree are from Oregon State University. Steve has published five papers on laser studies and has submitted a number of patent disclosures describing inkjet inks, dyes, and paper testing systems. Steve was born and raised in California, is married and has two children. He lives in Corvallis, Oregon. His leisure activities include playing guitar, international travel, political history, and racquetball.

45 ═══ Host Independent Emulators ═══

Arnold S. Berger

Arnie Berger was hardware project manager and leader of the original design and definition team for the HP 64700 Series Emulators. His past design responsibilities include the storage CRT for the HP 1727A Oscilloscope and the HP 54300A Probe Multiplexer, and he was the project manager for the HP 64120A Card Cage. His professional experience before joining HP in 1979 includes basic research in nondestructive testing as a senior staff scientist at Ford Motor Company and research on crystalline defect interactions in metal at Argonne National Laboratory. Arnie's BS degree (1966) and PhD degree (1971) are in materials science and are both from Cornell University. He has written some 25 papers and oral presentations, two journal articles, and one prior HP Journal article. He is a member of the American Physical Society and an authority on techniques for the measurement of electrical resistivity at cryogenic temperatures. Arnie was born in Brooklyn, New York, is married and has a child. He and his wife, who is a software design engineer in HP's Electronic Design Division, live in Colorado Springs, Colorado. His avocations include bicycling, woodworking, running, and restoring old HP instruments.

52 ═══ Emulator Software ═══

William A. Fischer, Jr.

During development of the HP 64700 Emulator, Bill Fischer was project manager in charge of software design. He has since become section manager with responsibility for Intel and Motorola emulation products. Previously, he has worked as technical support engineer and product marketing engineer. In the ten years before he joined HP in 1984, Bill was an engineer at the Hamilton Standard Division of United Technologies, designing automated test equipment for the space shuttle. He holds a BSEE degree (1973), an MSEE degree (1978), and a master's degree in management (1980), all from Rensselaer Polytechnic Institute. He has authored and coauthored a number of papers and articles, mainly about software project management. Bill was born in Attleboro, Massachusetts, and lives with his wife and four children in Colorado Springs, Colorado. He enjoys running and playing basketball.

57 ═══ Expanded Memory ═══

Gary W. Lum

As lead hardware engineer, Gary Lum's contribution to the NewWave development was the expanded-memory card hardware architecture for the HP Vectra ES Personal Computer. He is now project manager for advanced PC development. Since joining HP in 1979, he has worked on the logic design for HP 2627A and HP 2626A terminals, IC design for the HP Touchscreen II Personal Computer, and on HP Vectra PC development. Gary's BS degree in electrical engineering is from the University of California at Berkeley (1979) and his MS degree is from Stanford University (1981). He was born in Syracuse, New York, is married, and lives in Santa Clara, California. He is interested in film and film history, photography, and gardening.

Milton J. Lau

R&D engineer Milton Lau served as software project leader for development of the HP Vectra ES expanded-memory manager and RAM disc, and for debugging the expanded-memory card. Among his past projects were the operating system and firmware for the HP 150 Touchscreen PC and BIOS firmware for the HP Vectra PC. Before joining HP in 1984, he was an R&D engineer for a small company, developing

firmware for terminals compatible with HP and Digital Equipment Corporation systems. Milton's BSCS degree is from California State University, San Francisco (1983). He was born in San Francisco, California, and lives in Milpitas, California. His favorite leisure activities include basketball, skiing, and flying remote-controlled gliders.

Wesley H. Stelter

With memory system design and architecture his special interests, Wes Stelter's work experience as development engineer includes design for the HP Vectra and Vectra ES Personal Computers. On the NewWave project, he was responsible for the expanded-memory board for the Vectra ES PC. He attended Cogswell College and received a BSET degree in 1977, the year he also joined HP. Wes was born in San Francisco, where he serves as assistant scoutmaster in the Boy Scouts. He's married, has a child, and makes his home in San Bruno, California. For recreation, Wes enjoys hiking, marksmanship, history, and house reconstruction.

74 ═══ Redfield-Kunz Generalization ═══

Ulrich H. Haeberlen

Ulrich Haeberlen's involvement with the discipline of nuclear magnetic resonance goes back to the years 1967 to 1969, when he worked in John Waugh's NMR laboratory at the Massachusetts Institute of Technology. When he later returned to his native Germany, Ulrich joined the Max Planck Institute for Medical Research in Heidelberg to establish a research group focusing on solid-state NMR. He attended the Technische Hochschule in Stuttgart, where he earned his diploma and PhD degree. Ulrich's association with HP dates back to the mid-sixties, as he likes to recall, when an HP RF double-balanced mixer he was using greatly impressed him with its design and performance characteristics.

Alexander Keller

The theoretical and mathematical aspects of molecular motion in solid-state nuclear magnetic resonance have been Alex Keller's focal interests in the past three years. He was born in Germany and studied at the Heidelberg University, where he earned his PhD degree. Although Alex' primary professional concern is with mathematics, there are occasional ventures into experimental physics, one of which produced his contribution to use of the HP 5180A Waveform Recorder described in this issue of the HP Journal.

Generalization of the Redfield-Kunz Treatment of Quadrature Phase Time Data

A prescription is given to compute the complex Fourier transform spectrum from quadrature phase time data when the x and y signals are sampled neither simultaneously nor alternately. This case applies to the sampling scheme of the HP 5180A Waveform Recorder.

Alexander Keller and Ulrich H. Haeberlen

CLASSICAL, DISCRETE, QUADRATURE PHASE Fourier transformation stipulates that the analog signals $x(t)$ and $y(t)$ to be Fourier transformed are sampled and digitized simultaneously and that all time increments are equal. Digitization is often done by two-channel waveform recorders or equivalent instruments. Many of these sample the two input signals $x(t)$ and $y(t)$ alternately and not simultaneously. These instruments also do not store the data in separate sets $\{x_i\}$ and $\{y_i\}$, $i = 0, ..., N-1$ with $N = 2^n$ for some integer n, but rather in a single consecutive set $\{x_0, y_0, x_1, y_1, ..., x_{N-1}, y_{N-1}\}$. Redfield and Kunz[1] have indicated an elegant way to handle the computation of the Fourier transform or spectrum in this situation. The data set $\{x_0, ..., y_{N-1}\}$ is multiplied, element by element, with $+1, -1, -1, +1, +1, -1, -1, ...$. The discrete cosine (sine) Fourier transform of the resulting set $\{x_0, -y_0, -x_1, y_1, x_2, ...\}$ gives, without any further data manipulation, the quadrature phase absorption (dispersion) spectrum with the center or zero frequency automatically in the middle of the respective partial spectrum.

Hewlett-Packard offers a two-channel waveform recorder (the HP 5180A) which is very well-suited for digitization of quadrature phase time signals of many kinds, for which the Fourier transform, or spectrum, is ultimately to be computed. The data obtained in pulsed nuclear magnetic resonance (NMR) spectroscopy is just one example. The problem is that the HP 5180A samples input signals $x(t)$ and $y(t)$ neither simultaneously nor alternately. Whatever digitization rate is chosen, $y(t)$ is always sampled 100 ns later than $x(t)$. This problem could be cured with a delay line and appropriate compensation of the losses in the RF part of the quadrature phase receiver, but it is preferable to seek the solution on the computational side. It is the purpose of this communication to indicate the appropriate steps.

To explain the procedure we apply these steps to a test signal $x(t) = \cos(2\pi ft/T)$, $y(t) = \sin(2\pi ft/T)$ with f being an integer and $T = N\Delta t$ the total measuring time, where Δt is the sampling interval. T is thus an integer multiple of the period of the test signal and we can clearly formulate what we want to see in the resulting spectrum: the absorption part should have a line with some normalized amplitude at the position $\nu_k = k\Delta\nu$ with $k = +f$ and $\Delta\nu = 1/T$, and zeros everywhere else. All entries of the dispersion part should be zero. The signal is sampled as shown in Fig. 1, where the 100-ns difference in the sampling times for $x(t)$ and $y(t)$ is generalized to an interval dt.

The stored data consists of the set $\{s_q\}$ with

$$s_q = x_{2l} \rightarrow \cos(2\pi fl/N) \text{ for } q = 2l \text{ even}$$
$$= y_{2l+1} \rightarrow \sin(2\pi fl/N) \text{ for } q = 2l+1 \text{ odd}$$

where $l = 0, ..., N-1$. The arrows indicate the result in the case of our test signal. The Redfield-Kunz-treatment leaves us with the set $\{r_q\}$ with

$$r_q = r_{2l} = (-1)^l x_{2l} \rightarrow (-1)^l \cos(2\pi fl/N), \ q = 2l$$
$$= r_{2l+1} = -(-1)^l y_{2l+1} \rightarrow -(-1)^l \sin(2\pi fl/N), \ q = 2l+1.$$

The Fourier transform or complex spectrum $g(\nu_k)$ of the set $\{r_q\}$ consists of the set of N complex numbers

$$g_k = \sum_{q=0}^{2N-1} r_q e^{-2\pi iqk/2N} \ , \quad k = 0, ..., N-1$$

$$= \sum_{l=0}^{N-1} \{r_{2l} + r_{2l+1} e^{-i\pi k/N}\} e^{-2\pi ilk/N}. \tag{1}$$

We now insert our test signal into equation 1 and consider first the Redfield-Kunz case $\varepsilon = dt/\Delta t = \frac{1}{2}$.

$$g_k = \sum_{l=0}^{N-1} \{\cos(2\pi fl/N) - \sin(2\pi f(l+\frac{1}{2})/N) e^{-i\pi k/N}\}(-1)^l e^{-2\pi ilk/N} \tag{2}$$

$$= \frac{1}{2} \sum_{l=0}^{N-1} (-1)^l \{(1 + ie^{-i\pi(k-f)/N}) e^{-2\pi il(k-f)/N} + (1 - ie^{-i\pi(k+f)/N}) e^{-2\pi il(k+f)/N}\}.$$

Expressing i by $e^{i\pi/2}$ and $(-1)^l$ by $e^{i\pi l}$ leaves us with

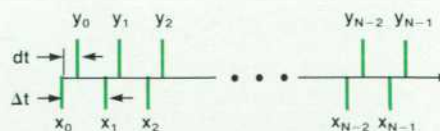**Fig. 1.** *Sampling of quadrature phase signals by waveform recorders.* $\varepsilon = dt/\Delta t = \frac{1}{2}$ *corresponds to the Redfield-Kunz case. dt = 100 ns corresponds to the HP Model 5180A.*

$$g_k = \frac{1}{2} \sum_{l=0}^{N-1} \{(1 + e^{-i\pi(k-f-N/2)/N})e^{-2\pi il(k-f-N/2)/N}$$
$$+ (1 - e^{-i\pi(k+f-N/2)/N})e^{-2\pi il(k+f-N/2)/N}\}. \quad (3)$$

Since

$$\sum_{l=0}^{N-1} e^{-2\pi il(k-k')/N} = N\delta_{k,k'}. \quad \text{modulo } N$$

we get

$$g_k = \frac{N}{2} \{(1 + e^{-i\pi(k-f-N/2)/N})\delta_{k,N/2+f} + (1 - e^{-i\pi(k+f-N/2)/N})\delta_{k,N/2-f}\}. \quad (3)$$

Note that both exponentials are equal to one whenever the Kronecker symbols are nonzero. Therefore,

$$g_k = N\delta_{k,N/2+f}. \quad (4)$$

This is the Redfield-Kunz result: a test signal of the type we have chosen gives a single line at the desired position, i.e., f steps to the right of the center in the absorption spectrum, whereas the dispersion spectrum (imaginary parts of the g_k) is zero throughout the entire range of k. If we had chosen $x(t) = \sin(2\pi ft/T)$ and $y(t) = \cos(2\pi ft/T)$ as test signals we would have obtained a single nonzero entry in the dispersion part of the complex spectrum.

We now turn to the general or HP case where $\varepsilon = dt/\Delta t \neq \frac{1}{2}$. With the identical sequence of steps in the treatment of the data, i.e., Redfield-Kunz multiplication and Fourier transformation as above, we obtain

$$g_k = \sum_{l=0}^{N-1} \{\cos(2\pi fl/N) - \sin(2\pi f(l+\varepsilon)/N)\}(-1)^l e^{-2\pi ilk/N} \quad (2')$$

$$= \frac{1}{2} \sum_{l=0}^{N-1} \{(1 + e^{-i\pi(k-2\varepsilon f - N/2)/N})e^{-2\pi il(k-f-N/2)/N}$$
$$+ (1 - e^{-i\pi(k+2\varepsilon f - N/2)/N})e^{-2\pi il(k+f-N/2)/N}\} \quad (2a')$$

which are the modified equations 2 and 2a. Instead of equation 3 we get

$$g_k = \frac{N}{2} \{(1 + e^{-i\pi(k-2f\varepsilon-N/2)/N})\delta_{k,N/2+f}$$
$$+ (1 - e^{-i\pi(k+2f\varepsilon-N/2)/N})\delta_{k,N/2-f}\}. \quad (3')$$

Applying the action of the Kronecker symbols to the exponentials gives

$$g_k = \frac{N}{2} \{(1 + e^{-i\pi(k-N/2)(1-2\varepsilon)/N})\delta_{k,N/2+f}$$
$$+ (1 - e^{-i\pi(k-N/2)(1-2\varepsilon)/N})\delta_{k,N/2-f}\}. \quad (5)$$

The problem with the HP waveform recorders is now evident: for $\varepsilon \neq \frac{1}{2}$ the exponentials are not equal to one, which means that our test signal produces nonzero entries in the dispersion part of the spectrum as well as a "ghost line" at $k = N/2 - f$ in the absorption part. We also recognize that the ghost lines are small in the central region of the spectrum where $k \approx N/2$ whereas they become large near the edges, where $k \approx 0$ and $k \approx N-1$. An example for $\varepsilon = 0.25$ and $f = 3N/8$ is shown in Fig. 2a.

To "exorcise" the ghosts we begin by defining a set (h_k) by
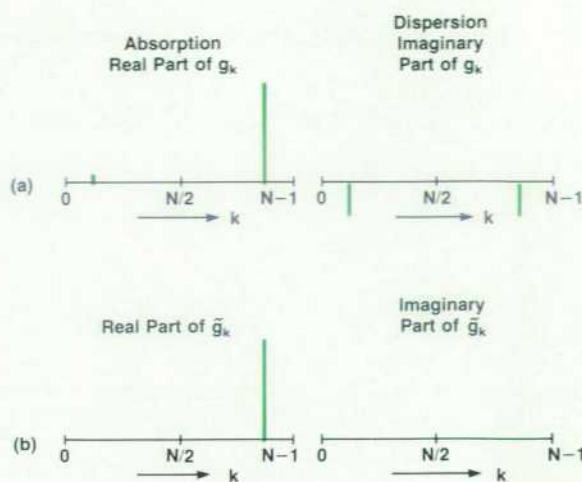

Fig. 2. *Absorption and dispersion spectra obtained with Red-field-Kunz treatment of data in the case of a test signal $x = \cos(2\pi ft/T)$, $y = \sin(2\pi ft/T)$ with $f = 3N/8$ and $\varepsilon = \frac{1}{4}$. (a) is the result before applying equation 7. Note the ghost line in the left half of the absorption spectrum, and the two ghost lines in the dispersion part of the spectrum. (b) is the result after applying equation 7.*

$$h_k \equiv g_{N-k}$$

$$= \frac{1}{2} \sum_{l=0}^{N-1} \{(1 + e^{-i\pi(N/2-k-2\varepsilon f)/N})e^{-2\pi il(N/2-k-f)/N}$$
$$+ (1 - e^{-i\pi(N/2-k+2\varepsilon f)/N})e^{-2\pi il(N/2-k+f)/N}\}.$$

This expression for h_k follows immediately from equation 2a'. Again introducing Kronecker symbols gives

$$h_k = \frac{N}{2} \{(1 - e^{i\pi(k-N/2)(1-2\varepsilon)/N})\delta_{k,N/2+f}$$
$$+ (1 + e^{i\pi(k-N/2)(1-2\varepsilon)/N})\delta_{k,N/2-f}\}. \quad (6)$$

By comparing equations 5 and 6 we see immediately that

$$g_k + h_k^* = g_k + g_{N-k}^* = N(\delta_{k,N/2+f} + \delta_{k,N/2-f})$$

and

$$g_k - h_k^* = g_k - g_{N-k}^* = N(\delta_{k,N/2+f} - \delta_{k,N/2-f}) e^{-i\pi(k-N/2)(1-2\varepsilon)/N}$$

from which we obtain our final result

$$\tilde{g}_k \equiv \frac{1}{2}\{(g_k + g_{N-k}^*) + (g_k - g_{N-k}^*)e^{i\pi(k-N/2)(1-2\varepsilon)/N}\}$$

$$= N\delta_{k,N/2+f} \quad (7)$$

which replaces equation 4, valid for the Redfield-Kunz-case $\varepsilon = \frac{1}{2}$ in the case of the use of an HP 5180A or equivalent waveform recorder. The digitization scheme of these waveform recorders can thus be taken care of by this simple prescription: treat the data exactly as in the Redfield-Kunz-case where the x and y samples are taken alternately. This gives the set $\{g_k\}$. Then use these g_k to compute the set $\{\tilde{g}_k\}$ according to equation 7. The real (imaginary) part of this set represents the desired discrete absorption (dispersion) spectrum.

Reference

1. A.G. Redfield and S.D. Kunz, *Journal of Magnetic Resonance,*
Vol. 19, 1975, p. 250.

CHANGE OF ADDRESS: To subscribe, change your address, or delete your name from our mailing list, send your request to Hewlett-Packard Journal, 3200 Hillview Avenue, Palo Alto, CA 94304 U.S.A. Include your old address label, if any. Allow 60 days.

5953-8572