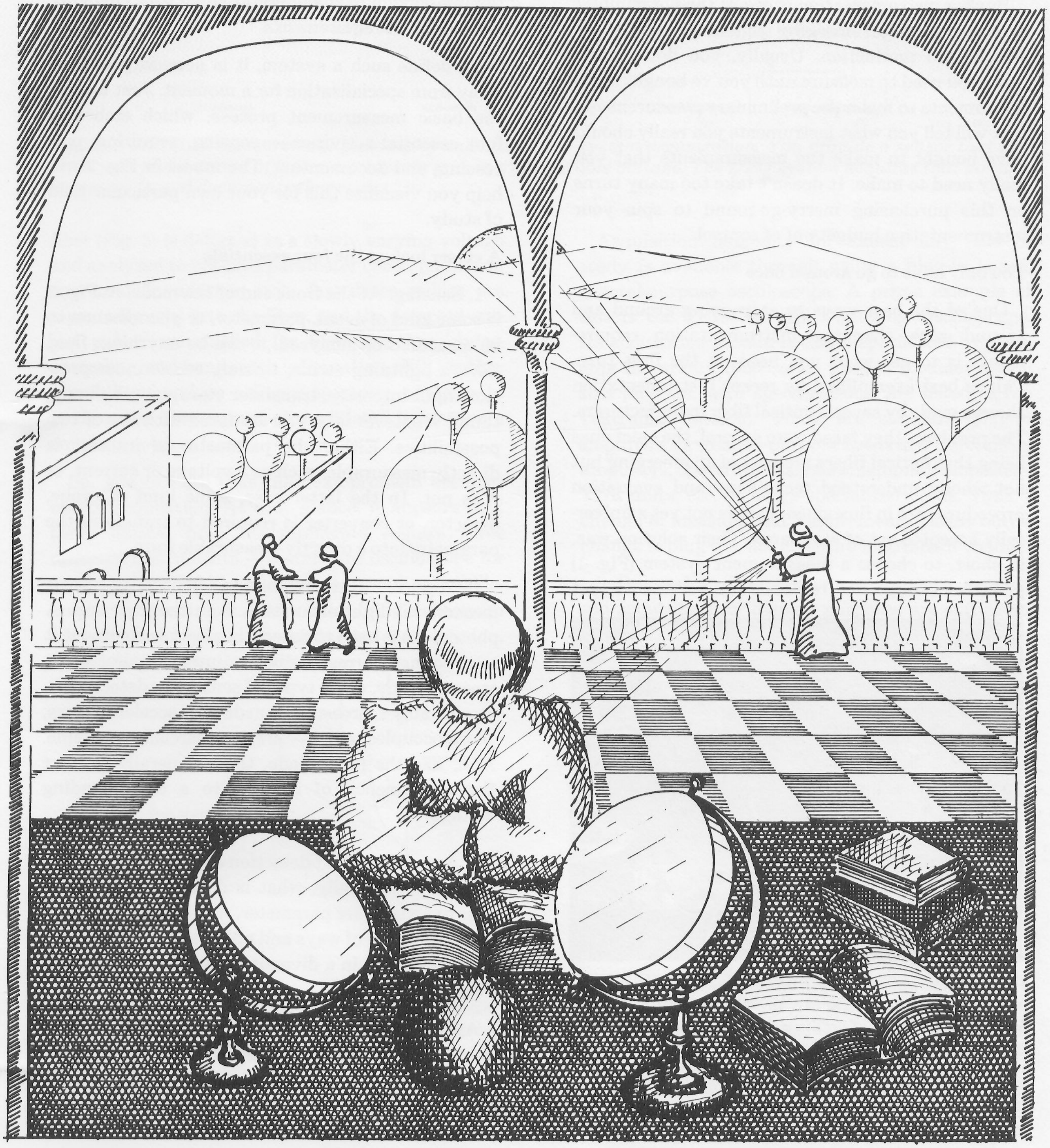
Vol. 3 No. 4 SUMMER 1978

Tektronix® COMMITTED TO EXCELLENCE

HANDSHAKE

Newsletter of the Signal Processing Systems Users Group

INSIDE:



Optical Measurements Enter a New Era

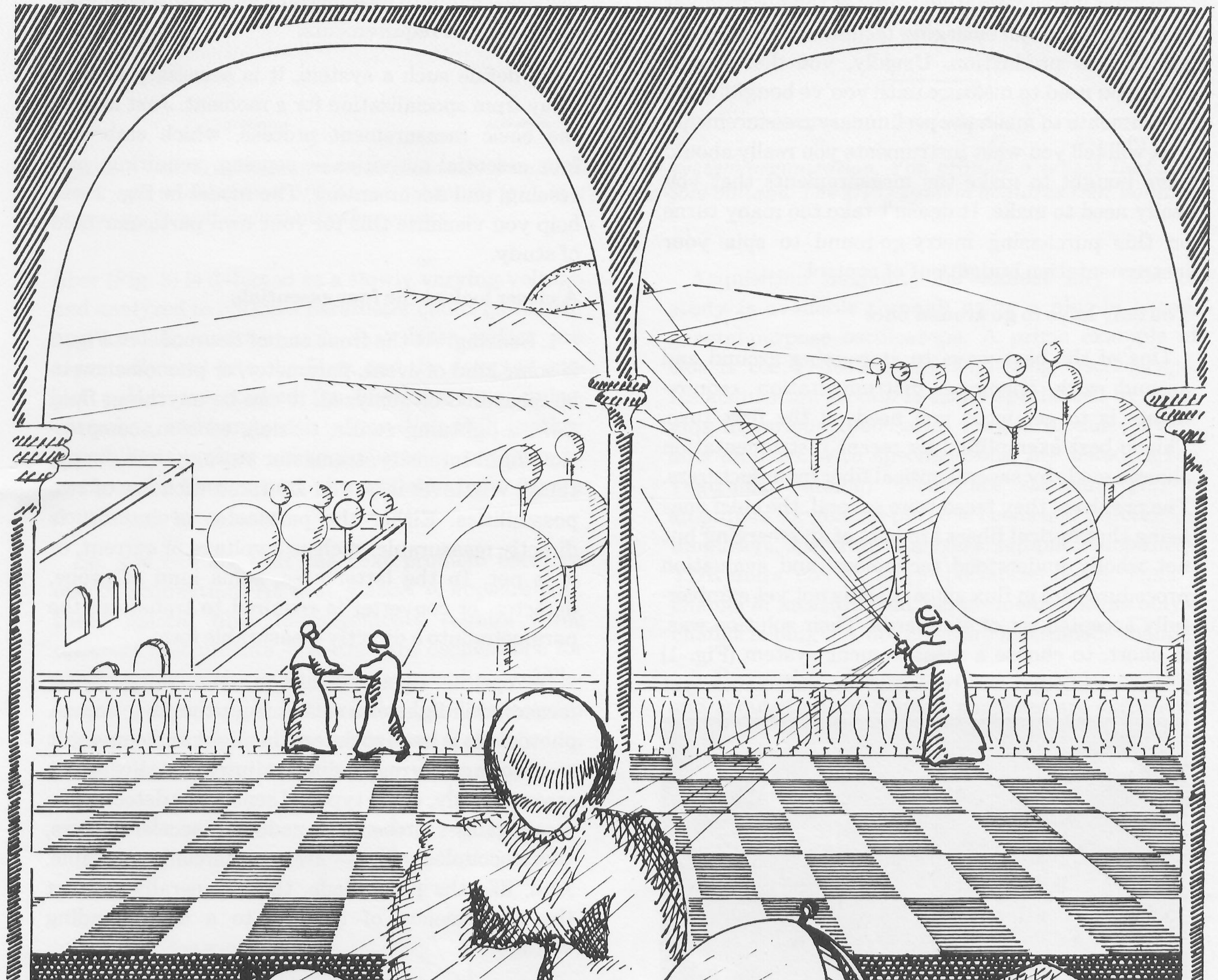
Vol. 3 No. 4 SUMMER 1978

Tektronix®
COMMITTED TO EXCELLENCE

HANDSHAKE

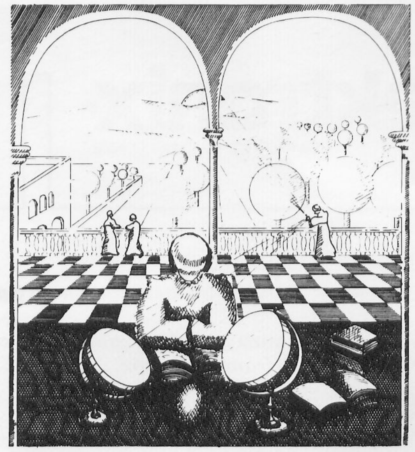
Newsletter of the Signal Processing Systems Users Group

INSIDE:



Optical Measurements Enter a New Era

Getting off of the instrumentation merry-go-round . . .



fiber optics success story points the way

Buying instrumentation is one of the big gambles in trying to ride an emerging technology through research into production. Usually, you don't know what you need to measure until you've bought some instruments to make the preliminary measurements that will tell you what instruments you really should have bought to make the measurements that you really need to make. It doesn't take too many turns on this purchasing merry-go-round to spin your instrumentation budget out of control.

You only have to go around once

One of the best ways to stop going around and around with changing instrumentation requirements is to get what you need on the first turn. This is best exemplified by recent instrumentation choices made by several optical fiber manufacturers. The problems they faced were several, the chief ones being that optical fibers are part of an emerging but not wholly understood technology and evaluation procedures are in flux since there is not yet a universally accepted set of standards. Their solution was, in short, to choose a measurement system (Fig. 1) powerful enough for research measurements, yet flexible enough to change right along with changing measurement requirements.

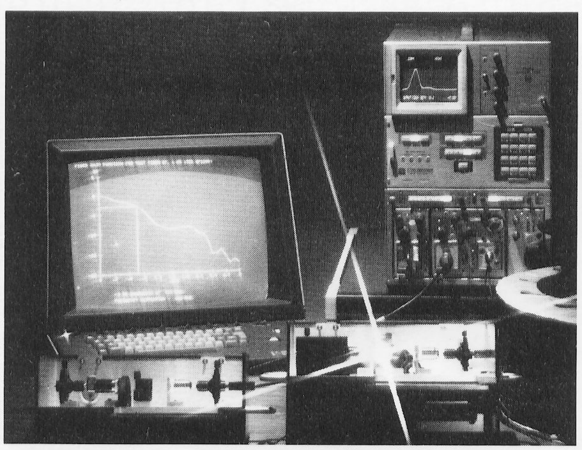


Fig. 1. With appropriate front-end sensors and plug-ins, the Digital Processing Oscilloscope (DPO) system is capable of solving measurement problems in most any field of study. Here, the DPO system is shown at an optical bench where its power and flexibility lend themselves to either complex research analyses or repetitive production evaluation of optical fibers.

To define such a system, it is necessary to step away from specialization for a moment. Just look at the basic measurement process, which embodies four essential activities — sensing, acquiring, processing, and documenting. The model in Fig. 2 will help you visualize this for your own particular field of study.

A closer look at the four essentials

1. **Sensing.** At the front end of the model in Fig. 2 is some kind of event, parameter, or phenomenon to be measured or analyzed. it can be anything: fluid flow, a lightning strike, torsion, tension, compression, light intensity, transistor storage time — anything. Whatever it is, you are faced with one of two possibilities. Either the parameter of interest is directly measurable, such as a voltage or current, or it is not. In the latter case, some kind of probe, detector, or converter is required to transform the parameter into a directly measurable form.

In optical fiber evaluation, the thing being measured is light emanating from the fiber end. A photodiode is used to detect this light and convert it to a voltage corresponding to intensity. For other areas of study, other types of sensors or detectors — strain gauges, pressure transducers, accelerometers, thermocouples, etc. — are commercially available. And, like the photodiode, these generally convert the phenomenon of interest to a corresponding voltage.

2. **Acquisition.** But detection or sensing is just the first step. Generally, what is detected varies with time or some other parameter. Also, it may be detected in a variety of ways and under a variety of conditions, resulting in a diversity of voltage waveforms from the detector.

Again, using optical fiber testing as an example, light is launched into the fiber in different ways for different tests. In one test, light from a continuous source is used. Then the intensity variation across the cone of light projecting from the other end of the

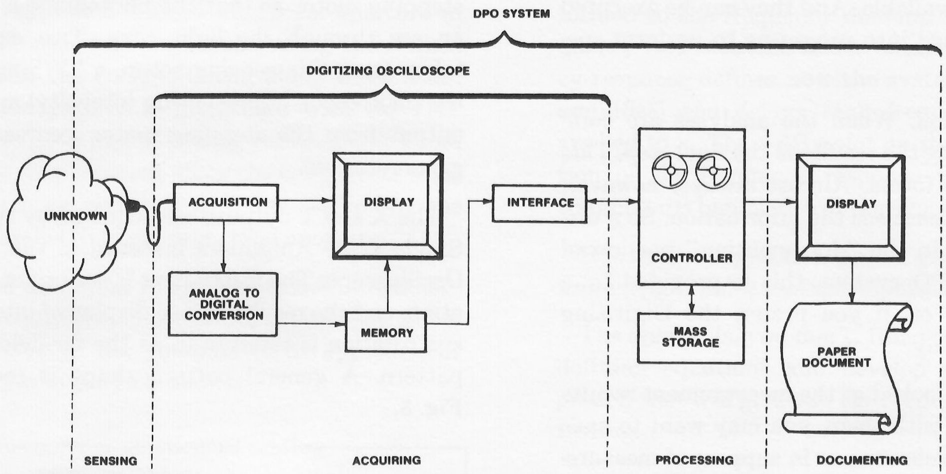


Fig. 2. A universal measurement philosophy underlies DPO system organization. You provide a sensor that converts your phenomenon of interest, whatever it is, to a measurable voltage. The DPO system acquires that voltage, processes it, and documents the results.

fiber (Fig. 3) is detected as a slowly varying voltage and analyzed to obtain a parameter called numerical aperture. In another test, fast laser pulses are repeatedly sent into the fiber, and the detected output pulses are studied for fiber rise-time, pulse-spreading, and bandwidth information. Still another possibility involves using a spectrum analyzer to analyze the fiber output of modulated light from a continuous-wave laser.

So, you can see that the next problem becomes one of acquisition. At first glance, it appears that these optical fiber measurements require three different instruments — a standard oscilloscope for studying intensity variation, a sampling oscilloscope for capturing fast laser pulses, and a spectrum analyzer for looking at the modulated transmission. This is not the case, however.

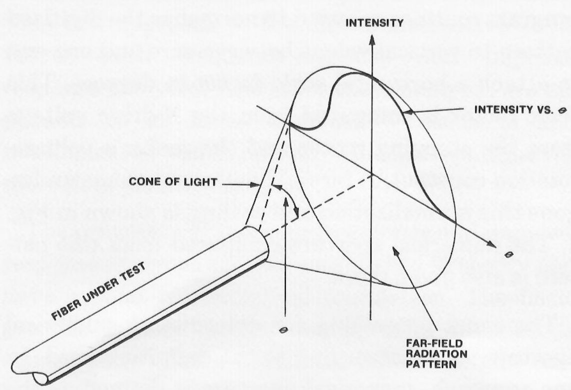


Fig. 3. Numerical aperture is obtained by computing the sine of the half angle, θ, of the light emanating from the fiber end. The half angle is taken from the far-field intensity pattern and is often defined as the angle between maximum intensity and some percentage of maximum, with 50 percent being a common choice.

Acquisition flexibility for almost any field of study is available through using a plug-in based, general-purpose oscilloscope. A prime example of this is the TEKTRONIX Digitizing Oscilloscope. Its acquisition section is easily specialized by plugging in various 7000-Series Plug-in modules (available plug-ins, with specifications, are listed in the Tektronix Catalog). There are standard plug-in amplifiers for general-purpose needs and differential amplifiers, spectrum analyzers, sampling amplifiers, TDR units, etc. for more specialized needs. Thus, a change in measurement requirements means only a change in plug-ins, not an entire instrument change.

3. **Processing.** For some measurements, sensing and acquisition for display purposes are sufficient. But more often, the phenomenon of interest can only be detected and acquired in terms of secondary information. These secondaries must then be manipulated and analyzed — processed — in order to arrive at the primary parameter of interest. Except for simple operations, this means that a processor and some waveform processing software are almost always necessary. So processing power becomes the third system requirement.

The Digitizing Oscilloscope meets this third requirement by virtue of containing an analog-to-digital converter and an interface to a minicomputer-type controller. It converts sensed and acquired waveforms to a digital format (a waveform array). Then the array of waveform data is transferred to a controller for processing. This processing is done with TEK SPS BASIC software, which performs mathematical operations on entire waveform arrays. A variety of operations — addition, subtraction, integration, differentiation, fast Fourier transformation, etc. — are available. And they can be executed singly or combined into programs to perform specialized analyses.

4. **Documentation.** When the analyses are complete, documenting the results is the final step. This can take several forms. Almost always, however, you'll need to at least see the information. So a display is specified in the "documenting" portion of Fig. 2. In the DPO system, this is provided by a graphic terminal or, if you prefer, the Digitizing Oscilloscope display.

Once you have looked at the measurement results and are satisfied with them, you may want to save them for further reference or in support of measurement methodology. This saving can be done either in digital form on the mass storage device shown under "processing" in Fig. 2 or by recording onto paper. Either path can be taken in the DPO system, with the latter paper-copy path available when a Hard Copy Unit is attached to the graphic terminal. An example hard copy of an optical fiber specification sheet, derived and formatted by software, is shown in Fig. 4.

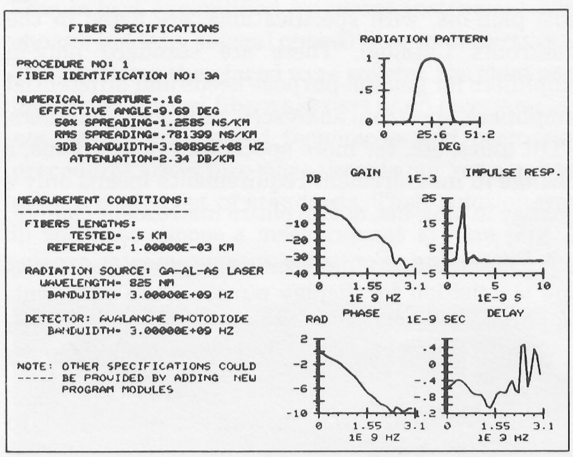


Fig. 4. Optical fiber specification sheet computed by TEK SPS BASIC software and output by the TEK SPS BASIC Graphics package. Such documentation is an integral part of the total measurement process.

Putting it to work for answers

Let's go back to the numerical aperture problem illustrated in Fig. 3 and put all four basic steps together to get a numerical aperture value. A test setup encompassing the first two steps — sensing and acquisition — is shown in Fig. 5.

Sensing is done by a photodiode mounted at the end of a rotating arm. The arm is axially aligned with the fiber undergoing test and is driven by a stepping motor so that the photodiode is swung in an arc through the light cone. Two outputs are taken from this sensing setup, a "Y" output from the photodiode (representing intensity) and an "X" output from the stepping motor (representing degrees rotation).

The X and Y outputs are acquired by two 7A15A Single-Trace Amplifiers installed in the Digitizing Oscilloscope. The instrument is set up for X-Y operation, and the resulting X-Y display of intensity versus rotation is referred to as the far-field radiation pattern. A general pattern shape is indicated in Fig. 5.

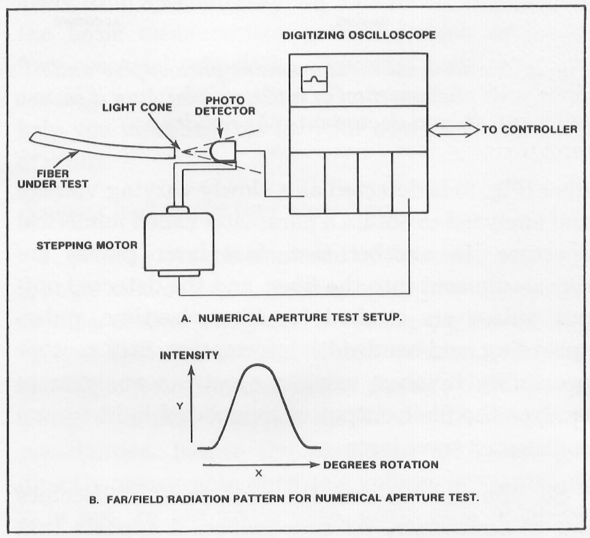


Fig. 5. Sensing and acquisition steps for determining numerical aperture.

As a prelude to the third step — processing — the far-field radiation pattern is digitized and transferred to the controller. Then, in the controller, BASIC program routines are used to normalize the digitized pattern to vertical values between zero and one and to attach a horizontal scale factor in degrees. This scale factor is computed from the X-drive voltage from the stepping motor and the motor's voltage-rotation constant. A far-field pattern that has undergone this normalization and scaling is shown in Fig. 4. The numerical aperture computed from this pattern is also given there.

The exact processing for determining numerical aperture varies according to the definition used. In one approach, numerical aperture is defined as the sine of the angle at which intensity has dropped to some percentage of maximum. A simple search along the specified percentage level with the cross function (CRS) of TEK SPS BASIC finds the angle, which is then converted to numerical aperture by computing its sine with the SIN function.

Another approach defines numerical aperture in terms of the cone half angle subtending the central 90 percent of the energy. Going along these lines, the radiation pattern is integrated with the INT command to get a plot of energy versus angle. From there, it's a simple search along the ten-percent level to find the angle, which is then operated on by the sine function (SIN) to get numerical aperture.

Whatever definition you choose to follow, the sensing and acquisition instruments remain the same. You just change the computational approach of your program to get your answer.

Reaching for more sophisticated testing

Numerical aperture is no doubt a useful parameter, but the greater concern today seems to be with optical cable bandwidth and rise time. This emphasis is understandable since these two parameters provide the primary evidence of transmission capabilities. Also, they are probably the most difficult to determine by traditional methods.

Several techniques have been used to indicate optical fiber transmission capabilities. The simplest of these is to repetitively inject fast-rise laser pulses into the cable. Then, assuming the input pulse rise time is much shorter than that of the cable, the output pulse rise time becomes a fair indication of cable rise time. Another approach involves comparing the width of an input pulse to the width of the resulting output pulse and computing a figure known as pulse spreading.

Any of these techniques can be accomplished with greater speed and precision by using a signal processing system. However, with signal processing power at hand, a more definitive approach is available. This new approach — the digital alternative — involves both frequency-response and impulse-response analyses. And since these analyses are applicable to all linear systems — electrical and mechanical as well as optical — the concept deserves more than passing mention.

The principles of frequency-response and impulse-response analyses are well established in theory and have a solid mathematical foundation. The basic premise is that an impulse or delta function (a square pulse of zero width and infinite height) contains all frequencies at constant magnitudes of unity. If such a pulse could be applied to a linear, time-invariant system (an optical cable for example), the resulting output would be the impulse response and would fully characterize the system. The rise time of the impulse response is the system rise time. And, when the impulse response is Fourier transformed to the frequency domain, the system's frequency-response function is obtained. This frequency response defines how the system attenuates (or amplifies) and delays each frequency component applied to it. The 3-dB point on the magnitude portion of the frequency response is the commonly sought 3-dB bandwidth of the system.

But where, in practice, do you get a zero-width, infinite-height pulse for true impulse testing?

The answer is, you don't. But you can look at the defining equations and develop an alternate approach that will give you the results of true impulse testing.

The first equation to look at is in the time domain. It is composed of functions of time and states that the output of a system or device is equal to its impulse response convolved with the input waveform. Or, in mathematical terms,

$$p_o(t)=p_i(t)*h(t)$$

where $p_o(t)$ and $p_i(t)$ are cable output and input pulses as functions of time, h(t) is the cable's impulse response, and * denotes the mathematical operation of convolution.

Having to deal with convolution, however, makes the equation in the time domain somewhat unappealing. But, by transforming the equation to the frequency domain with the Fourier transform, it becomes much more pleasant to work with. In the frequency domain, the equation becomes

$$P_o(f)=P_i(f)H(f)$$

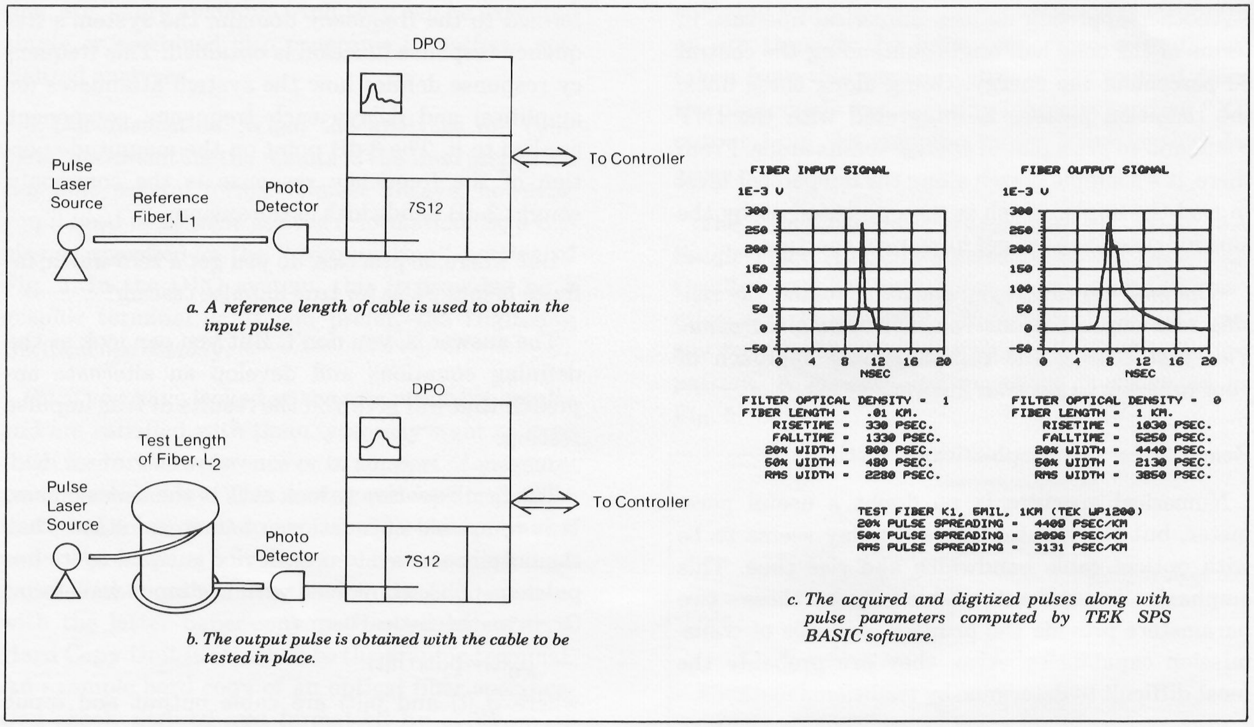
where $P_o(f)$ and $P_i(f)$ are the frequency-domain functions for the output and input pulses and H(f) is the frequency-response function. This is a simple algebraic relation, involving only multiplication, and it can be rearranged to give

$$H(f)=P_o(f)/P_i(f).$$

And, from H(f), you get a frequency-by-frequency description of how the cable affects any input in terms of attenuation and phase shift. Also, by Fourier transforming H(f) back to the time domain, you get h(t), the impulse response, from which you can obtain cable rise time.

The steps in actually doing this with a signal processing system are really quite simple, First, you start with sensing and acquisition, as indicated in Fig. 6. For optical fibers, a laser repetitively pulses the cables used in the test, and the transmitted pulses are detected with a photodiode. Because of the fast rise pulses involved (hence high-frequency content), a sampling plug-in is used to expand the Digitizing Oscilloscope's acquisition capabilities.

Getting off of the instrumentation merry-go-round . . .



a. A reference length of cable is used to obtain the input pulse.

b. The output pulse is obtained with the cable to be tested in place.

c. The acquired and digitized pulses along with pulse parameters computed by TEK SPS BASIC software.

Fig. 6. Acquiring input and output date for frequency-response and impulse-response analyses of optical cable.

The 7S12 Sampling Plug-in indicated in Fig. 6, when used with the Digitizing Oscilloscope, provides 0.4 picosecond resolution on its fastest range (20 picoseconds per division). "Sampling plug-ins push Digitizing Oscilloscope into the gigahertz region," an article appearing elsewhere in this issue, contains more details on sampling techniques.

Once the input, $p_i(t)$, and output, $p_o(t)$, pulses have been acquired and converted to waveform arrays by the Digitizing Oscilloscope, they are transferred to the controller for processing. TEK SPS BASIC contains an FFT (fast Fourier transform) algorithm that allows you to transform the pulse arrays to the frequency domain. So, by using the FFT, $p_i(t)$ and $p_o(t)$ are converted to $P_i(f)$ and $P_o(f)$. Then, it's simply a matter of ratioing these transform results to get H(f), the frequency response. And, to complete the analysis, H(f) is inverse transformed with the FFT algorithm to get h(t), the impulse response. Example results obtained by this process are documented in Fig. 4, under the radiation pattern.

In the example results, the gain function is the magnitude portion of H(F). Shown below this gain function, is the phase portion of H(f), and next to it is a delay function obtained by differentiating the phase function. The impulse response, obtained by inverse transforming H(f) to the time domain, is shown just to the right of the gain function.

Getting off the merry-go-round

For those interested in more details about evaluating optical fibers with a DPO system, we recommend Tektronix Concept Note AX-3903, entitled "Keeping Pace with Changing Needs in Optical Fiber Evaluation." This Concept Note describes a variety of measurements made with a single DPO system. In each case, measurement specialization is achieved, not through a major instrument change, but by simply changing sensors, plug-ins, or more often just by changing the processing approach to evaluation. To get your copy of this Concept Note, just check the appropriate box on the reply card in this issue of HANDSHAKE and return the card to us.

For those interested in fields other than fiber optics, Tektronix has a variety of Concept and Application Notes covering different measurement needs. Some are listed on the HANDSHAKE reply card. Others can be found in the "Tektronix Listing of Free Publications," also available by checking the reply card.

One of these publications might just be the helping hand that will get you off the instrumentation merry-go-round!

By Bob Ramirez, HANDSHAKE Staff

Sampling plug-ins push Digitizing Oscilloscope into the gigahertz region

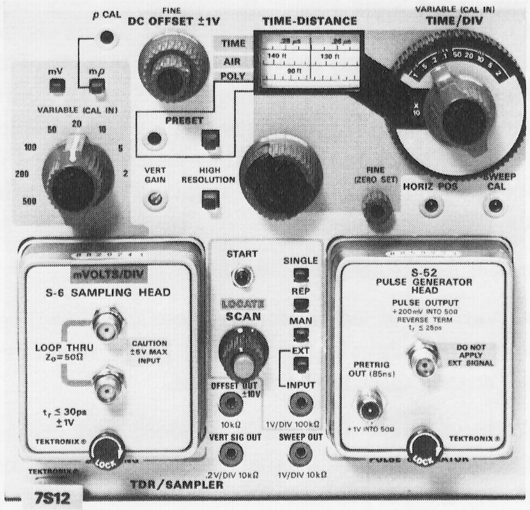


Fig. 1. The 7S12 TDR/Sampler, for use with TEKTRONIX 7000-Series oscilloscopes and the Digitizing Oscilloscope, extends scope bandwidth up to 14 gigahertz for repetitive signals. Also, with the pulse generator head shown, the 7S12 operates as a calibrated time domain reflectometer (TDR).

Suppose you have to measure gigahertz signals... and you only have a 200 megahertz oscilloscope.

Is there a way out of this seeming impasse?

The answer is yes, if you are working with repetitive signals. By using sampling techniques, oscilloscope bandwidths can be expanded into the gigahertz region, giving you the power to measure most high-frequency, repetitive signals. Normally, this requires the purchase of a sampling oscilloscope. However, if your 200 megahertz scope happens to be a TEKTRONIX 7000-Series Oscilloscope or the TEKTRONIX Digitizing Oscilloscope, it can be inexpensively converted by sliding a sampling unit into the plug-in slots.

For either gigahertz or picosecond work — such as evaluating optical fibers, microwave components, etc. — three sampling plug-ins are available for use with the Digitizing Oscilloscope. These are the TEKTRONIX 7S12, 7S11, and 7S14. The 7S12 (Fig. 1) contains its own time base and is used for general single-channel sampling as well as time domain reflectometry. The 7S11 is used as a companion unit, installed next to the 7S12, when dual-channel operation is required.* The bandwidth and rise-time figures for each of these sampling plug-ins depend upon the sampling head used. The sampling heads contain the signal input circuitry — the sampler, preamplifier, etc. They plug into the sampling unit and can be interchanged according to your bandwidth and input impedance needs. Figure 2 lists the available sampling heads with their specifications. Or, in lieu of these, the 7S14 Sampling Unit can be used. It has a self-contained time base and two self-contained sampling heads for dual-trace operation. These have 50-ohm inputs with DC-to-1 gigahertz bandwidths and 350 picosecond or less rise times.

SAMPLING HEAD	INPUT Z	BANDWIDTH (3 dB DOWN)	RISE TIME
S1	50 ohm	DC to 1 GHz	350 ps or less
S2	50 ohm	DC to 4.6 GHz	75 ps or less
S3A	100 K, 2.3 pF	DC to 1 GHz	350 ps or less
S4	50 ohm	DC to 14 GHz	25 ps or less
S5	1 M, 15 pF	DC to 350 MHz	1 ns or less
S6	50 ohm	DC to 11.5 GHz	30 ps or less

Fig. 2. Sampling heads for the 7S11 and 7S12 plug-ins.

Time conversion is the key

There are several ways that sampling techniques are implemented for fast-response measurement needs. However, they are all directed toward the same goal — to convert rapid changes to slower speeds that can be digitized or at least observed via an oscilloscope. The idea is similar to the slow-motion instant replays that are so popular with televised sporting events.

The block diagram in Fig. 3a is useful in explaining how this time conversion or slowing of time is achieved. It should be noted that this is a general block diagram and not attributable to any specific instrument. Specific approaches to time conversion may vary slightly from the following general approach, which uses sequential sampling.

The sequential process starts in Fig. 3a with the high-speed, repetitive signal to be measured applied at the signal input. A fast sweep generator is triggered by this signal. Also, a slow sweep gener-

**The 7S11 could also be used as a single-channel unit with the 7T11 Sampling Sweep Unit acting as a time base. However, the 7T11 is not wholly compatible with the Digitizing Oscilloscope, so this combination is not recommended for use with Waveform Processing systems.*

ator is triggered in synchronism with the fast sweep. These two sweeps and the ability to compare them are the key to the time-conversion process.

During the time when both the fast and slow sweeps are running up, a sample pulse generator is triggered. In Fig. 3, this triggering occurs at those times when the fast sweep and slow sweep are of equal amplitude, and the resulting trigger pulse is applied to the sample-and-hold circuit.

The sampler is usually a very fast diode which turns on momentarily, allowing the signal to pass to a memory or holding device. The speed of the diode is such that only a very short segment or "point" on the signal is passed to the hold circuit. The hold circuit charges quickly to that sampled value, then holds it until the next sample is taken. As can be seen from the waveforms sketched in Fig. 3b, the holding of each sample effectively stretches or slows it to a duration much longer than the time the sampling diode is on. The duration of this held sample is sufficient to create a single dot on the oscilloscope display.

The vertical location of the displayed dot is determined by the sample's value (signal amplitude at the sample point). Its horizontal location on the display is determined by the slow sweep, which provides horizontal drive for the oscilloscope. The dot's location, relative to its position on the sampled input signal, is maintained by the sequencing and comparison of the fast and slow sweeps. And, although the slow ramp provides horizontal drive for the oscilloscope, it is the fast ramp that determines the time scale for the display.

The result of all this is a sequence of displayed dots, sampled from many repetitions of the signal, that depicts a single repetition of the display. This is shown in the bottom waveform in Fig. 3b. Figure 4 shows an actual oscilloscope display of a sampled waveform. Most sampling units include options to smooth such raw "dotted-in" displays into a continuous, single-trace display.

Connecting the dots for a smooth display

A variety of circumstances conspire to keep "repetitive" signals from being truly repetitive. Chief among these is additive noise.

Additive noise is usually random and creates amplitude variations on the signal being measured. The randomness of these added variations keeps one repetition of the signal from looking exactly like the next. Also, to sample and view the signal with an oscilloscope, it is necessary to trigger off of some fixed point on the signal to obtain a stable display. But, with random noise, there is no stable trigger point. Triggering, then, tends to be jittery, and a shaky display of the signal results. Combined with this, too, is the fact that no circuit is perfect and, therefore, some degree of jitter is introduced to the signal by its generating circuits. Then there is a final, but small, contribution by the normal jitter of oscilloscope triggering circuits.

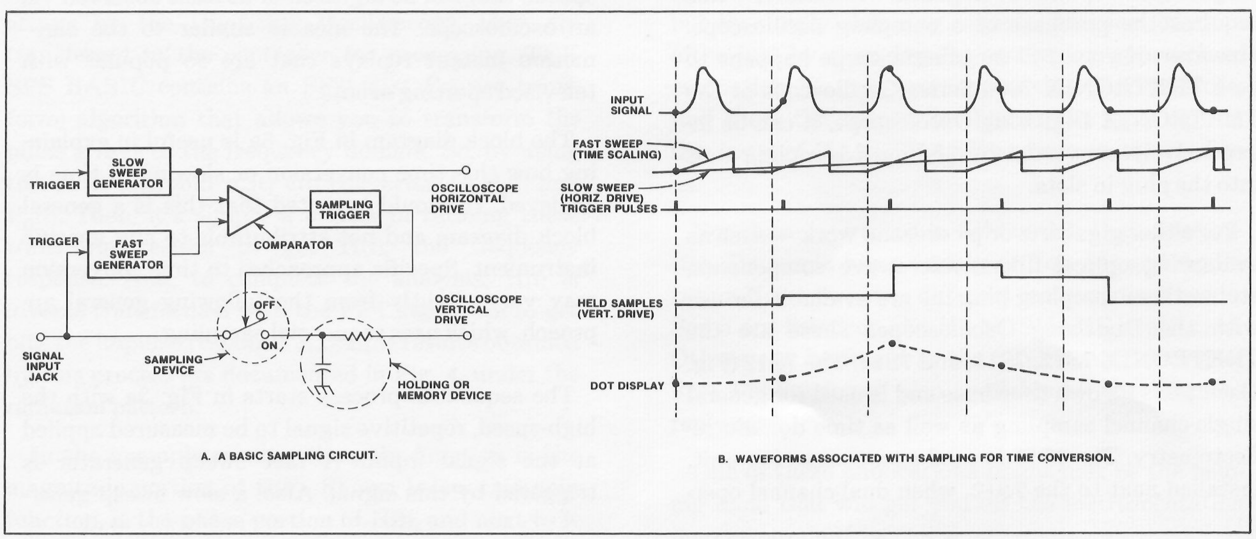


Fig. 3. Principles of sequential sampling for time conversion of fast signal to slow displays.

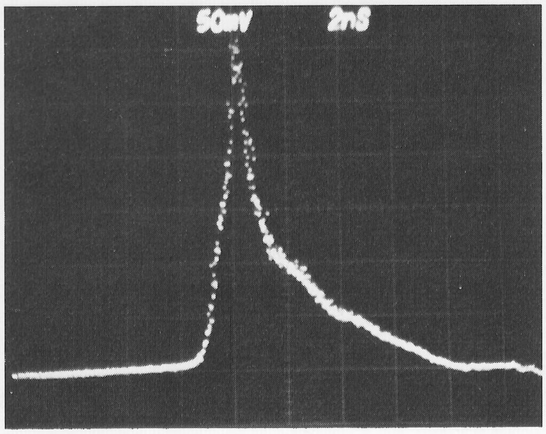


Fig. 4. Display of a detected laser pulse after transmission through an optical fiber. The pulse was captured with a 7S12 sampling unit used in conjunction with a TEKTRONIX Digitizing Oscilloscope.

In sampling measurements, time jitter and amplitude noise reflect themselves as a swath of dots as opposed to the ideal, single, well-ordered string of dots. This is because each repetition of the signal changes slightly; thus, the sample points change slightly from one horizontal sweep to the next.

To cope with these noisy signals, most sampling units use some kind of smoothing or resolution enhancement. Generally, this has the effect of filtering out high frequencies to gain a smooth signal display. This action and its results stem from the substantial high-frequency content of most noise and jitter. And, applied judiciously, smoothing or filtering is more than adequate for measurements made visually from an oscilloscope display. There is a limitation, though. Filtering is not discriminatory. It cuts out all high frequencies — those that make up fast signal transitions as well as those associated with noise and jitter.

Where signal processing is available, a statistically purer approach to removing noise can be used. This approach is signal averaging, and it is discriminatory on the basis of randomness rather than frequency. By averaging many repeated acquisitions of the signal, random noise and jitter components approach a mean value — zero for most random occurrences — while the stable, repetitive signal is reinforced or "lifted out of the noise" without adding any distortions. With the Digitizing Oscilloscope and TEK SPS BASIC software, many averages can be taken quickly with substantial improvements (30 decibels and more!) in signal-to-noise ratio. And you do not affect the high-frequency components that belong to the signal.

By Bob Ramirez, HANDSHAKE Staff

New literature from Tektronix

New technical notes available

Nonlinear Least-Squares using the Gauss-Newton and Marquardt Algorithms

Experimental data generally suffers from random errors, or noise, generated within the process or produced in the act of measuring the data. The least-squares method is one approach to estimating the parameter values that is relatively insensitive to noise.

Alternative Methods of measuring Optical Fiber Attenuation with the Digital Processing Oscilloscope

This note reviews the alternative methods of the fiber attenuation measurement and demonstrates the Tektronix solution to these various measurement requirements.

Getting the most out of TEK BASIC GRAPHICS

Why not fit it on one page?

If you have seen the feature article in this issue of HANDSHAKE, "Getting off of the instrumentation merry-go-round . . .", you probably noticed that the fiber optics test results were automatically documented on a single page. We felt you might be interested in the routine used to format this specification sheet. The program was developed by Chris Flatau of TEK Ltd. and makes extensive use of several Graphic Package commands.

Running the program as listed in Fig. 2 outputs the bare sheet in Fig. 1. There are six data files that must be filled and five flags (K1 thru K5) that must be made nonzero by the calling program in order for the sheet to resemble the figure in the feature article.

The specification sheet program is made up of a series of short routines, each introduced by an explanatory REM statement. Line 680, for example, introduces the segment which composes the section of the sheet devoted to "Phase" and "Delay" data. The VIEWPORT command (lines 750 and 790) allocates space on the display screen for the two graphs: Phase and Delay. The SETGR command (lines 760 and 800) sets up the graticule for the GRAPH command. Omitting the keyword WINDOW from the SETGR command allows the GRAPH command's autoscaling to determine the "best fit" presentation of the data. The Phase data to be graphed is stored in file "PHADAT.FIB"; Delay data in "DELDAT.FIB". If the Phase and Delay tests were run, making the flag (K2 at line 720) nonzero, lines 730 to 820 are executed. Here the files are read (lines 740 and 780) and the results graphed. If the tests were not run, K2 remains zero and "NOT SPECIFIED" is output on the sheet in place of the graphs.

A similar pattern is followed within each program section devoted to formatting a graph. An area of the display screen is reserved for a particular graph with the VIEWPORT command. The SETGR command ensures that the GRAPH command will use that viewport and determines the graticule. Then, if the test has been run, the graph is made, if not, a "NOT SPECIFIED" message is inserted to indicate the lack of data.

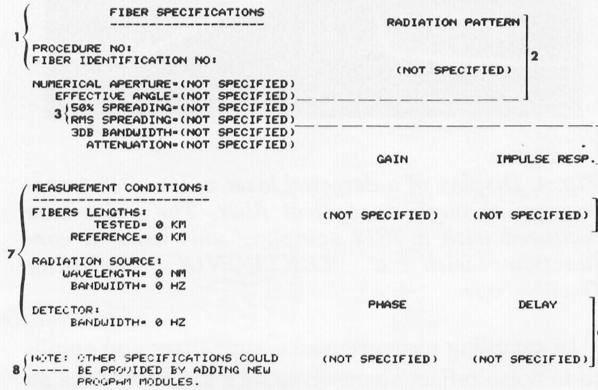


Fig. 1. Specification sheet routine output without test data. A filled specification sheet is shown in this issue with the article entitled "Getting off of the instrumentation merry-go-round . . ."

Two commands handle the work of displaying the many standard messages which appear on the sheet. SMOVE shifts the writing beam to a specific location, in graphic device units, so the PRINT command can display the message (lines 170, 270, 540, etc.) The abilities of the PRINT command alone are sufficient to handle setting up the tabular format for the messages occupying the left margin of the spec. sheet.

Some interesting characteristics of the PRINT command are worth pointing out. The first PRINT in line 50 is used to output a carriage return and line feed instead of the blank line you might expect. (The carriage return and line feed were suppressed by the semicolon delimiter following the PRINT statement in line 40.) The second PRINT, then, in line 50 outputs a blank line. A similar situation occurs in line 1080, where a semicolon again ends the PRINT statement. In this case, however, the PRINT in line 1090 is used to add a message following on the same line as the dashes done by line 1070. The TAB function is frequently used to horizontally align the many messages. Instead of using TAB, blank spaces could have been entered within the quotation marks of the PRINT command, but it is not always easy to remember how many blank spaces have been input. This is clear when TAB is used.

```
   10 REM OPTICAL FIBER SPECIFICATION SHEET
   20 INITG
   30 PRINT TAB(10);"FIBER SPECIFICATIONS"
1  40 PRINT TAB(10);\FOR I=1 TO 20\PRINT "-";\NEXT I
   50 PRINT\PRINT\PRINT "PROCEDURE NO: ";PN$
   60 PRINT "FIBER IDENTIFICATION NO: ";F1$
   70 PRINT
   80 REM
   90 REM PRINTING NUMERICAL APERTURE AND GRAPHING RAD. PAT.
   100 REM
   110 IF K5=0 THEN GOTO 140
   120 OPEN #1 AS DX1:"NAPDAT.FIB" FOR READ\READ #1,A1,A1$,NU\CLOSE #1
   130 A2$=STR(A1)&A1$\N2$=STR(NU)\GOTO 150
   140 A2$="(NOT SPECIFIED)"\N2$=A2$
   150 PRINT "NUMERICAL APERTURE=";N2$
   160 PRINT TAB(3);"EFFECTIVE ANGLE=";A2$
   170 SMOVE 640,748\PRINT "RADIATION PATTERN"
2  180 DELETE A,AA\WAVEFORM A IS AA(511),SA,HA$,VA$
   190 IF K4=0 THEN GOTO 270
   200 OPEN #1 AS DX1:"RADDAT.FIB" FOR READ\READ #1,A\CLOSE #1
   210 VA$=""
   220 VIEWPORT 683,863,600,720
   230 WINDOW 0,512*SA,MIN(AA),MAX(AA)
   240 SETGR VIEW,WIND,GRAT 4,4,TICS 2,2
   250 GRAPH A
   260 GOTO 290
   270 SMOVE 655,660\PRINT "(NOT SPECIFIED)"
   280 REM
   290 REM PRINTING PULSE SPREADING RESULTS
   300 REM
   310 IF K1=0 THEN GOTO 350
   320 OPEN #1 AS DX1:"DISRES.FIB" FOR READ\READ #1,P5,PR,P$\CLOSE #1
   330 P5$=STR(P5)&P$\PR$=STR(PR)&P$
3  340 GOTO 360
   350 P5$="(NOT SPECIFIED)"\PR$=P5$
   360 SMOVE 0,592
   370 PRINT TAB(5);"50% SPREADING=";P5$
   380 PRINT TAB(5);"RMS SPREADING=";PR$
   390 REM
   400 REM PRINTING AND DISPLAYING GAIN PARAMETERS
   410 REM
   420 SMOVE 623,500\PRINT "GAIN"
   430 SMOVE 837,500\PRINT "IMPULSE RESP."
   440 IF K2=0 THEN GOTO 520
   450 DELETE A,AA\WAVEFORM A IS AA(NS),SA,HA$,VA$
   460 OPEN #1 AS DX1:"GADATA.FIB" FOR READ\READ #1,A,B3,B3$,AN,AN$
   470 CLOSE #1
   480 VIEWPORT 563,743,320,460
4  490 SETGR VIEW,GRAT 4,4,TICS 2,0
   500 GRAPH A
   510 B3$=STR(B3)&B3$\AN$=STR(AN)&AN$\GOTO 540
   520 SMOVE 525,400\PRINT "(NOT SPECIFIED)"
   530 B3$="(NOT SPECIFIED)"\AN$=B3$
   540 SMOVE 0,548\PRINT TAB(5);"3DB BANDWIDTH=";B3$
   550 PRINT TAB(7);"ATTENUATION=";AN$
   560 REM
   570 REM DISPLAYING IMPULSE RESPONSE
   580 REM
   590 IF K3=0 THEN GOTO 660
   600 DELETE A,AA\WAVEFORM A IS AA(511),SA,HA$,VA$
   610 OPEN #1 AS DX1:"RESDAT.FIB" FOR READ\READ #1,A\CLOSE #1
5  620 VIEWPORT 803,983,320,460
   630 SETGR VIEW,GRAT 4,4,TICS 2,0
   640 GRAPH A
   650 GOTO 680
   660 SMOVE 795,400\PRINT "(NOT SPECIFIED)"
   670 REM
   680 REM DISPLAYING PHASE AND DELAY
   690 REM
   700 SMOVE 609,240\PRINT "PHASE"
   710 SMOVE 879,240\PRINT "DELAY"
   720 IF K2=0 THEN GOTO 830
   730 DELETE A,AA\WAVEFORM A IS AA(NS),SA,HA$,VA$
   740 OPEN #1 AS DX1:"PHADAT.FIB" FOR READ\READ #1,A\CLOSE #1
   750 VIEWPORT 563,743,60,200
   760 SETGR VIEW,GRAT 4,4,TICS 2,0
6  770 GRAPH A
   780 OPEN #1 AS DX1:"DELDAT.FIB" FOR READ\READ #1,A\CLOSE #1
   790 VIEWPORT 803,983,60,200
   800 SETGR VIEW,GRAT 4,4,TICS 2,0
   810 GRAPH A
   820 GOTO 850
   830 SMOVE 525,140\PRINT "(NOT SPECIFIED)"
   840 SMOVE 795,140\PRINT "(NOT SPECIFIED)"
   850 DELETE A,AA\RELEASE "GRAPH","VIEWPORT","SETGR","SMOVE"
   860 REM
   870 REM PRINTING MEASUREMENT CONDITIONS
   880 REM
   890 SMOVE 0,450
   900 PRINT "MEASUREMENT CONDITIONS:"
   910 FOR I=1 TO 23\PRINT "-";\NEXT I
   920 PRINT
   930 PRINT "FIBERS LENGTHS:"
   940 PRINT TAB(8);"TESTED=";UL;" KM"
7  950 PRINT TAB(5);"REFERENCE=";RL;" KM"
   960 PRINT
   970 PRINT "RADIATION SOURCE: ";RS$
   980 PRINT TAB(4);"WAVELENGTH=";WV;" NM"
   990 PRINT TAB(5);"BANDWIDTH=";RB;" HZ"
   1000 PRINT
   1010 PRINT "DETECTOR: ";DE$
   1020 PRINT TAB(5);"BANDWIDTH=";DB;" HZ"
   1030 REM
   1040 REM PRINTING A NOTE
   1050 REM
   1060 PRINT\PRINT
   1070 PRINT "NOTE: OTHER SPECIFICATIONS COULD"
8  1080 FOR I=1 TO 5\PRINT "-";\NEXT I
   1090 PRINT " BE PROVIDED BY ADDING NEW"
   1100 PRINT TAB(6);"PROGRAM MODULES."
   1110 RETURN
```

Fig. 2. Program listing for fiber optics specification sheet.

Before returning to the calling program, line 850 deletes the waveform and array and releases the nonresident commands used to build the specification sheet. We assume the calling program would appreciate the extra memory space.

By Walt Robatzek, HANDSHAKE Staff.
Based on a program by Chris Flatau, Applications Engineer, TEK Ltd.

Signal Processing Systems users' application program library

If one of the following programs interests you, a listing and any available support literature will be sent to you — free of charge. Order your program by title from SPS Software Support at the address shown below, or contact your local Tektronix representative. For a copy of the latest index of programs in the library, check the appropriate box on the reply card in this issue of HANDSHAKE.

Remember, if you have a TEK BASIC program you would like to share with other Signal Processing System users, you may enter it into the library by sending the program listing to:

Tektronix, Inc.
SPS Software Support
94-319
P.O. Box 500
Beaverton, OR 97077

Outside the USA, send your programs to SPS Software Support via your local Tektronix representative. Please include with your program a short description of what it does and how it does it. We would also like to know about any special data conditions, instruments, or software package requirements. The memory requirements for running the program would also be very helpful.

TEK 31 PROGRAMMABLE CALCULATOR ABSTRACTS

Ultrasonic Velocity Testing of Nodular Iron

Instruments Required: Tek 31 Programmable Calculator, DPO, two 7D15 Counter/timers, 7A26 Dual-Trace Amplifier, 7B53A Dual Time Base, Dual Delayed Trigger Unit, AC Switching Unit, pulser/receiver, and transducer crystals.

Listing Length: 2000 steps.

This program (which uses TEK 31 keystroke programming) tests nodular iron cast parts for size and composition quality by timing ultrasonic pulse transmission through the part and the water transfer medium.

TEK SPS BASIC PROGRAM ABSTRACTS

Evaluation of Optical Fiber Transmission Characteristics

Instruments Required: DPO, 7S12 sampling plug-in, S4 or S6 sampling head, and S53 trigger recognizer head.
Software Packages Required: DPO Driver, Signal Processing, and Graphics.
Listing Length: 1700 lines.

This program is set up to automatically acquire optical fiber test signals (laser pulse and broadband lamp) via a Digitizing Oscilloscope. The acquired signals are then used to compute pulse attenuation, pulse spreading, fiber frequency response, fiber impulse response, spectral attenuation, and numerical aperture.

Static Performance Testing of Digitizers

Instruments Required: None, but stored instrument data is required.
Software Packages Required: Graphics.
Listing Length: 960 lines.

Using stored instrument data, this program computes the static performance of analog-to-digital converters or digitizers. It does this by estimating analog-to-digital transition points. A variety of static performance analysis options are provided.

Dynamic Performance Testing of Digitizers

Instruments Required: None, but stored instrument data is required.
Software Packages Required: Graphics and Signal Processing.
Listing Length: 1620 lines.

This analysis sequence uses two programs to determine digitizer dynamic performance. The first program in the sequence operates on stored output data from the digitizer to synthesize the digitizer's analog input. The synthesis is done by least-squares fitting an appropriate model to the digitizer output data. Then, key statistics describing the digitizer's dynamic performance are computed and stored. The second program provides several options for further analyses of the stored performance statistics.

Simple Exponential Decay Curve Fitting

Instruments Required: None.
Software Packages Required: None.
Listing Length: 570 lines.

This program uses a Gauss-Newton, nonlinear, least-squares algorithm to fit simple exponential decay models to data.

Multiple Exponential Decay Curve Fitting

Instruments Required: None.
Software Packages Required: None.
Listing Length: 700 lines.

This program fits decay models consisting of either a single exponential or the sum of several exponentials to the decay data. The algorithm used is a Marquardt, nonlinear, least-squares algorithm.

Interfacing issues

The 4051/7912AD — an automatic waveform acquisition system

When you need waveform processing but don't need the speed of standard waveform processors, you'll like the economics of interfacing the TEKTRONIX 7912AD Programmable Digitizer to the TEKTRONIX 4051 Graphic System. The connection is direct, thanks to the IEEE 488 bus, also called the General Purpose Interface Bus or GPIB.

The 7912AD, with its 7A16P Programmable Amplifier and 7B90P Programmable Time Base plug-ins, is a fully programmable waveform digitizer. Input sensitivities, offset, sampling rate — all important front-panel controls can be operated either manually or under program control from the 4051.

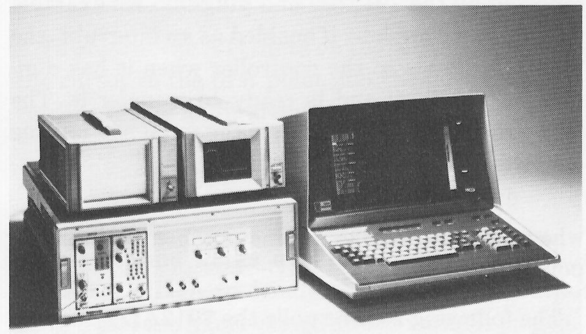
The 4051, itself, provides general waveform processing capabilities. It is a popular, small controller that includes a programmable calculator, graphic display, magnetic tape drive, and resident BASIC programming language. If you need more or want to go faster, there's the possibility of an RS-232-C link to a host computer.

Putting the 4051/7912AD combination to work does require some applications programming, however. If you like to do it yourself, here's some ideas on how to go about it. Complete details on operation and programming can be found in both the 7912AD and 4051 manuals.

Getting started

The techniques that follow can be used as the basis for subroutines to control the 7912AD and its plug-ins and to acquire data. For an automatic system that permits operator interaction from the keyboard, the main program can allow these subroutines to be called either by the program or by a keystroke on the 4051 user-definable keys. The main program should be written to wait or loop in a busy state with these keys enabled when the processor is not otherwise occupied.

To restore front-panel control of the 7912AD and plug-ins, the program has only to stop or the operator to press the break key, since the 4051 then releases the remote enable line of the IEEE 488 bus. Front-panel control can also be restored by sending the go-to-local command to the 7912AD.



A simple main program (monitor) need only be:

```
2 WAIT
3 GO TO 2
```

with line 1 used to jump to an initializing subroutine where interrupts from the user-definable keys and instruments are enabled by SET KEY and ON SRQ program statements. Other subroutines should begin by disabling the user-definable keys with a SET NOKEY statement. All subroutines should end with SET KEY RETURN statements.

A typical system

The system used to develop these routines is one that might be used at either a research or a production site to acquire, process, store, and display data. Besides the 4051 system controller and the 7912AD with programmable plug-ins, two monitors display acquired waveforms: a TV monitor for a real-time display of the 7912AD input and an XYZ monitor for a refreshed display of the digitized waveform once it is stored in the 7912AD data memory.

To configure this system, the 7912AD primary address switches were set for 1 (00001 binary). The secondary address switches were also set for 1 (00001 binary), making the 7912AD device 1 in the program. The plug-ins then assumed the next higher secondary addresses, making the amplifier (7A16P) device 2 and the time base (7B90P) device 3. Any primary address could have been chosen other than zero, which is reserved for the 4051.

Variables P and S are used in the program for the primary and secondary address. These variables should be set to the value selected by the address switches during the initializing subroutine.

A serial poll

A serial poll routine is needed for system control. The first occasion arises at power-up when the 7912AD and plug-ins assert the service request (SRQ) line to report they are on-line. This routine can be used to handle other conditions and errors reported by the 7912AD and plug-ins. An example is operation-complete; if enabled as an interrupt, the 7912AD signals to the controller when it has digitized a waveform and is ready to transfer it. The program should normally leave the serial poll routine enabled by the program statement:

```
ON SRQ THEN X
```

where X is the beginning line number of the serial poll routine.

The following routine polls the 7912AD and plug-ins, returning the device number and the value of the status byte of the first device found asserting SRQ. If no device is asserting SRQ, it returns a zero for the device number and status.

```
POLL Q1,Q2;P,S;P,S+1;P,S+2
PRINT "DEVICE ";Q1;" STATUS = ";Q2
```

The device number, Q1, is the number of the device in the list that is found to be asserting SRQ, counting the first device in the list as number 1, the next as number 2, and so on. If the secondary address switches are set for 1, then the secondary address matches the device number, Q1.

Of course, the interrupt routine would normally be longer to handle the interrupt condition according to user needs. To do this, the routine can test Q1 to decode which instrument is asserting SRQ and then branch to a routine to handle that instrument. This routine can then test Q2 to decode the interrupt condition and branch to another routine that handles that condition.

With the simple monitor described above, an existing SRQ hangs the system. It must be cleared by calling the serial poll routine at runtime as many times as is necessary to read all SRQ conditions.

Reading data

Although the 7912AD is useful for viewing very fast transients, its main claim to fame is acquiring such waveforms as data and transferring the data over the IEEE 488 bus for processing. So let's demonstrate this.

Some understanding of the data is useful before proceeding. Data are recovered from the scan converter by detecting the trace written on a diode matrix in a manner similar to plotting the X-Y coordinates of points on an oscilloscope trace. Rather than a single pair of X-Y coordinates to represent a data point, however, two Y values are detected for each X: one Y represents the top of the trace at that point and another Y represents the bottom of the trace. The distance between these two Y values is caused by trace width, which varies with waveform shape, sweep speed, waveform repetition rate, and writing beam intensity.

This is not the messy situation it might appear to be, however, thanks to an internal bit-slice microprocessor system. This system has several routines in firmware that can massage the data before transferring it over the IEEE 488 bus. One routine, ATC (average-to-center) is especially handy with a smaller controller such as the 4051. The ATC routine converts the raw data to a single-valued function by summing the top and bottom trace values and filling in any missing points.

To acquire an ATC waveform, send the appropriate commands as shown in the following 4051 BASIC statement:

```
PRINT @P,S:"DIG DAT;MODE TV;ATC"
```

Graticule intensity should previously have been set to zero. If not, the above line should be preceded by:

```
PRINT @P,S:"GRI 0"
```

This prevents the graticule from being written on the target and digitized with the waveform. If the graticule is to be digitized for comparison to the waveform, it should not be part of the data processed by the ATC routine. Instead it should be digitized and read out separately.

The digitize/average statement above initiates a digitize operation on the next time-base sweep, then resets the instrument mode for a TV display of the input waveform. If the waveform is repetitive, the TV monitor and XYZ monitor both display the waveform, the TV in real-time and the XYZ as a plot of ATC-processed data.

Although the ATC waveform has the simplest format and reduces much of the burden of further processing, it may not be the ideal data for all applications. For this reason, the full internal data array as well as the data arrays that result from several other internal processing routines can be read out over the bus. Because these other arrays are not as well adapted for use with the 4051 as for other controllers, they are not discussed further in this article. Instead, let's proceed to read and display the ATC waveform.

The ATC data array contains 512 values that represent the amplitude of the waveform beginning at time zero (sweep start) and continuing in time order (from left to right as viewed on a monitor). The array is transferred in the 7912AD block-binary format. This format comprises a parser, a byte count, the data values, a checksum, and the 7912AD message unit delimiter.

The binary block begins with the parser, an ASCII percent sign (31 decimal), and is followed by two bytes that represent the number of bytes to follow in the block. For a simple input program, these bytes can be ignored. If used, the percent sign could be a parser to cause the program to branch to a binary block input routine, and the byte count could initialize both the input array size and an error-checking routine. For our purposes, however, we can dimension the array to 512 and do without error checking.

The 512 values begin on the fourth byte and are transmitted as byte pairs, first a high byte and then a low byte. The low byte contains the lower eight bits of the value and the high byte contains the higher-order bits with unused bits set to zero. The 4051 RBYTE statement converts these binary bytes to decimal, requiring some simple arithmetic to assemble the whole value from the two bytes. To do the arithmetic, multiply the high byte by 256, because it represents the value of bits nine and above in the 16-bit binary word, and add the value of the low byte.

Following the 512 data values, the 7912AD sends a checksum and the ASCII code for simicolon — each a single byte. For our purposes, let's handshake, but ignore, both the checksum and simicolon.

Thus, the routine needs only to follow these steps: set up an array, send the 7912AD read command and talk address, read but throw away three bytes, read the ATC values to fill the array, read but throw away the last two bytes, and untalk the 7912AD. This routine can also be used to read another 7912AD data array, the signal-averaged array acquired with the DIG SA command.

```
DIM A(512)
PRINT @P,S:"READ ATC"
WBYTE @64+P,96+S:
RBYTE X1,X2,X3
FOR I=1 TO 512
RBYTE V1,V2
A(I)=(V1*256+V2)/2
NEXT I
RBYTE X1,X2
WBYTE @95:
```

Displaying the data

To see the waveform, let's graph the data on the 4051 screen. The first step is to create a viewing area and define the limits of the data. A graticule can be simulated by a box with tic marks around the viewing area. The commands to do this are:

```
VIEWPORT 10,110,10,90
WINDOW 1,512,1,512
AXIS 51.2,64,1,1
AXIS 51.2,64,512,512
```

To present the ATC data on the 4051 screen, move the cursor to the first point and then draw the succeeding points in the array:

```
MOVE 1,A(1)
FOR N=2 TO 512
DRAW N,A(N)
NEXT N
```

This routine works well for averaged data, but must be altered to handle other 7912AD output arrays that are multi-valued and of variable length.

The data plot can be scaled by reading in the scale factors as discussed later in this article.

Putting the program in programmability

Most front-panel controls on the 7912AD and the plug-ins can be operated under remote control using set commands. The programmable functions and their command mnemonics are listed in the 7912AD and plug-in manuals. The following routine can be used to enter a set command at the 4051 keyboard and transfer it to the 7912AD or plug-ins. When the device number is requested, either 1, 2, or 3 should be entered: 1 for the 7912AD, 2 for the 7A16P, or 3 for the 7B90P.

```
PRINT "ENTER DEVICE #"
INPUT D
PRINT "ENTER SET COMMAND"
INPUT A$
PRINT @P,S+D-1:A$
```

Some examples of set commands for the 7912AD are:

MODE DIG (set to digital operating mode)
MAI 102 (set main intensity to 1/10 full scale)
OPC ON (enable SRQ when waveform digitized)

Some examples of plug-in set commands are:

V/D 0.01 (set 7A16P volts/division to 10 mV)
T/D 1. E-9 (set 7B90P time/division to 1 nS)
POS -3 (set 7A16P offset three divisions below graticule center)
RIN HI (set 7A16P input impedance to 1 megohm)
SRC EXT (set 7B90P for external triggering)

The current setting of any programmable control can be read by a query command. A routine to send a query command and print the response follows:

```
PRINT "ENTER DEVICE #"
INPUT D
```

The 4051/7912AD — An automatic waveform acquisition system

```
PRINT "ENTER QUERY COMMAND"
INPUT A$
PRINT @P,S+D-1:A$
INPUT @P,S+D-1:A$
PRINT A$
```

This routine sends the query command entered on the 4051 keyboard, makes the selected device a talker, inputs the response, then prints it on the 4051 screen. The result is the query printed on the screen with the answer printed just below it.

Queries can be used to determine the scale factors from either the 7912AD or its plug-ins. To find the vertical scale factor (volts/division), query either the 7912AD with VS1? or the 7A16P with V/D? To find the horizontal scale factor (time/division), query either the 7912AD with HS1? or the 7B90P with T/D?

Queries can also be used to determine the settings of other programmable functions, the identification of the instrument (ID?), and the error, if any, corresponding to the last status byte reported by the 7912AD.

All programmable settings of a plug-in can be acquired with the single query, SET?. The above query routine must be modified, however, to handle the response. In the case of the plug-ins, the 4051 interprets each setting in the response as a message. The input statement should be put in a loop to read all settings of each plug-in (eight for the 7A16P and ten for the 7B90P). A print statement can be made part of the loop to display each setting on the 4051 screen.

In the case of the 7912AD, the 4051 interprets the response of the 7912AD to SET? as a single string. Since the string is too long for a single 4051 line, the input string variable size must be previously dimensioned to 120 and the string broken into several lines for printing.

Thanks to the common use of the IEEE 488 bus, the 4051, 7912AD, 7B90P, and 7A16P can be easily connected and can be programmed to play together as an automated waveform acquisition system. Although the programming is not trivial, it is in high-level, English-like BASIC statements.

If you would like more information about this system, call your nearest Tektronix field office or representative.

By Jim Kassebaum, Tektronix Software Engineer and Jim Kimball, HANDSHAKE Staff

The Interfacing Issues Column is dedicated to informing you of instrument combinations possible through various types of interfacing schemes, with the current emphasis being on the IEEE 488 bus, also called the General Purpose Interface Bus (GPIB). Since this is your forum, we would like to hear of your successful efforts to interface instruments in uncommon or interesting combinations.

Edited by Walt Robatzek *Graphics by Bernard Chalumeau*

HANDSHAKE is published quarterly by the SPS Documentation Group. Permission to reprint material in this publication may be obtained by writing to:
HANDSHAKE Editor
Group 157
Tektronix, Inc.
P.O. Box 500
Beaverton, Oregon 97077
HANDSHAKE is provided free of charge by Tektronix, Inc., as a forum for people interested in digital signal processing. As a free forum, statements and opinions of the authors should be taken as just that — statements and opinions of the individual authors. Material published in HANDSHAKE should not be taken as or interpreted as statement of Tektronix policy or opinion unless specifically stated to be such.

AX 4029

HANDSHAKE
SPS Documentation Group MS 94-384
TEKTRONIX, INC.
P.O. Box 500
Beaverton, OR 97005

2270783
CLARK FOLEY
39-327
IF YOU HAVE MOVED, PLEASE
CALL EXT. 6378