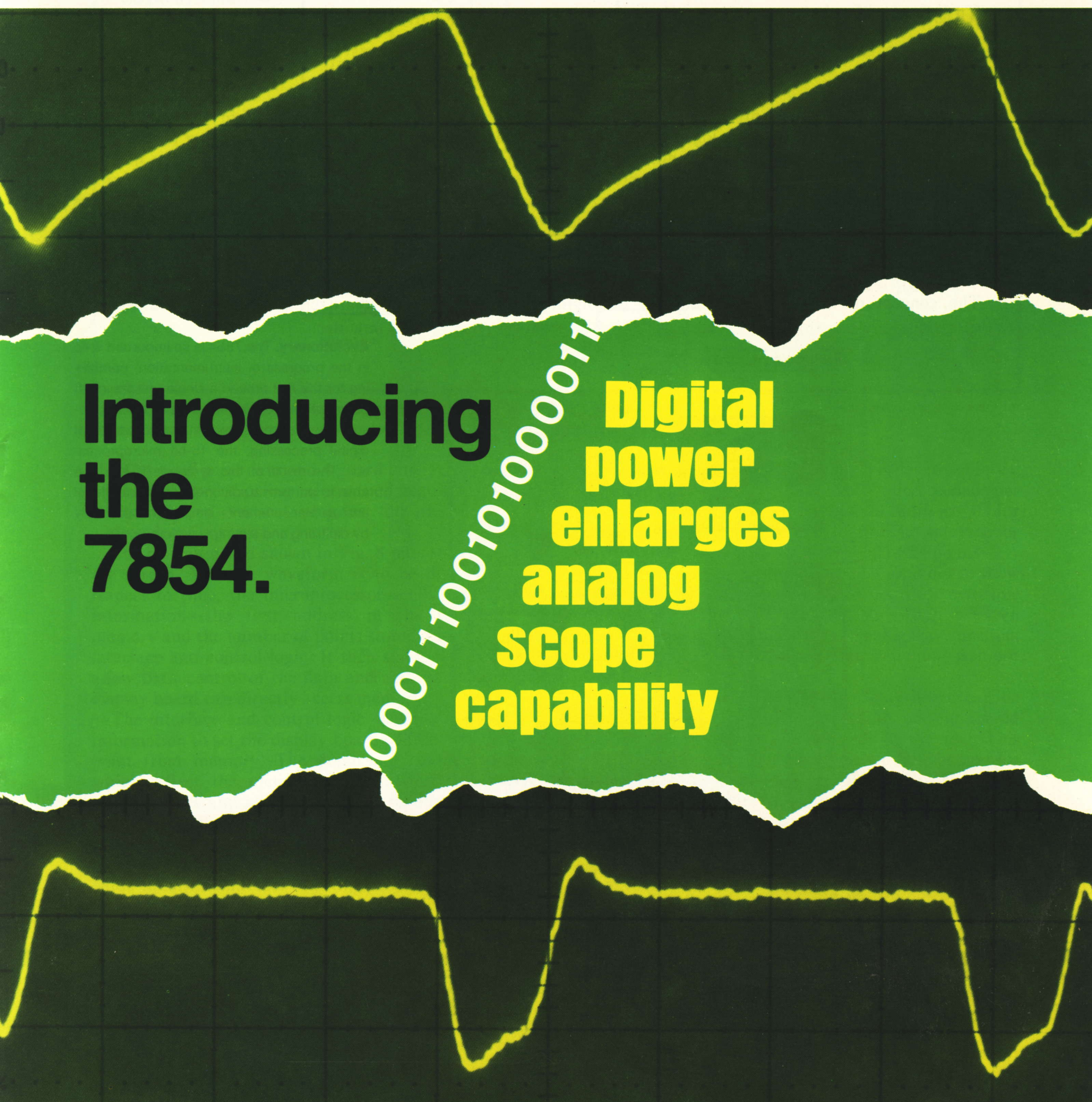
MARCH 13, 1980

On-board digital processing
refines scope measurements.

50 YEARS
A McGRAW-HILL PUBLICATION
Electronics®



On-board digital processing refines scope measurements

General-purpose unit with 400-MHz bandwidth stores waveforms digitally; its calculatorlike keyboard treats them as operands for one-keystroke functions like integration, differentiation, and smoothing—and it's programmable

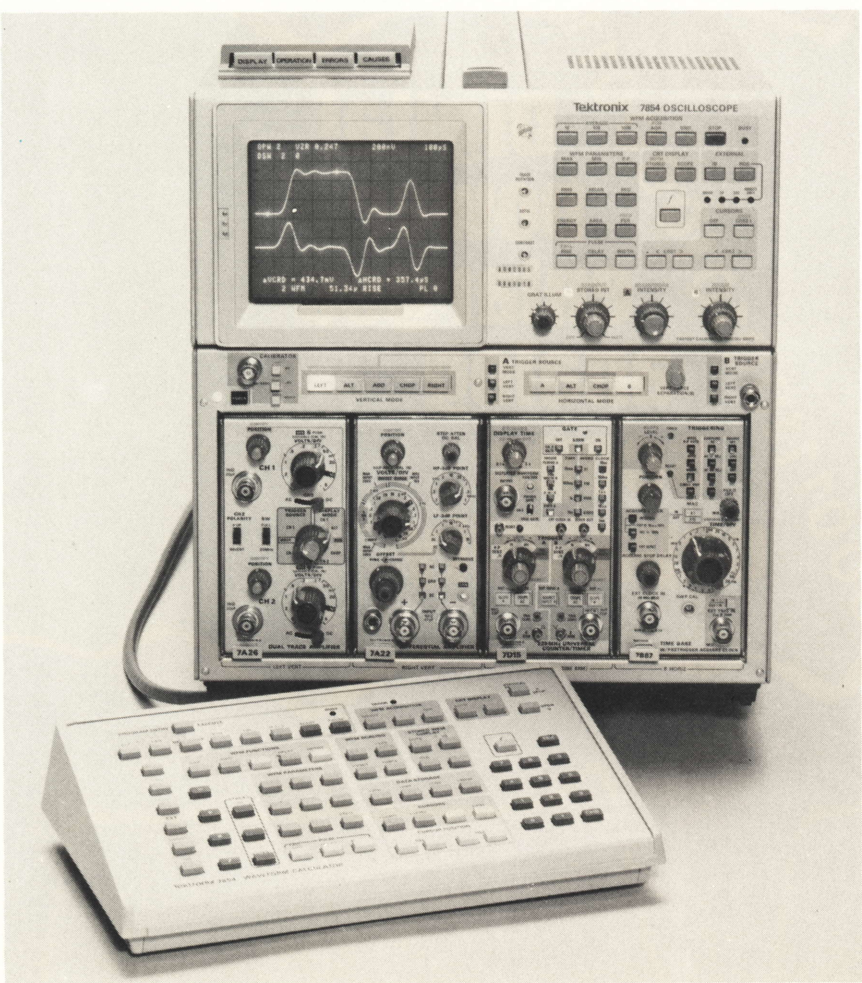
by Val Garuts and Jim Tallman, *Tektronix Inc., Beaverton, Ore.*

□ The long-awaited truly intelligent oscilloscope has finally arrived. This benchtop analog instrument easily measures, compares, and transforms entire waveforms digitally and thus precisely, as if they were simple, single-valued parameters.

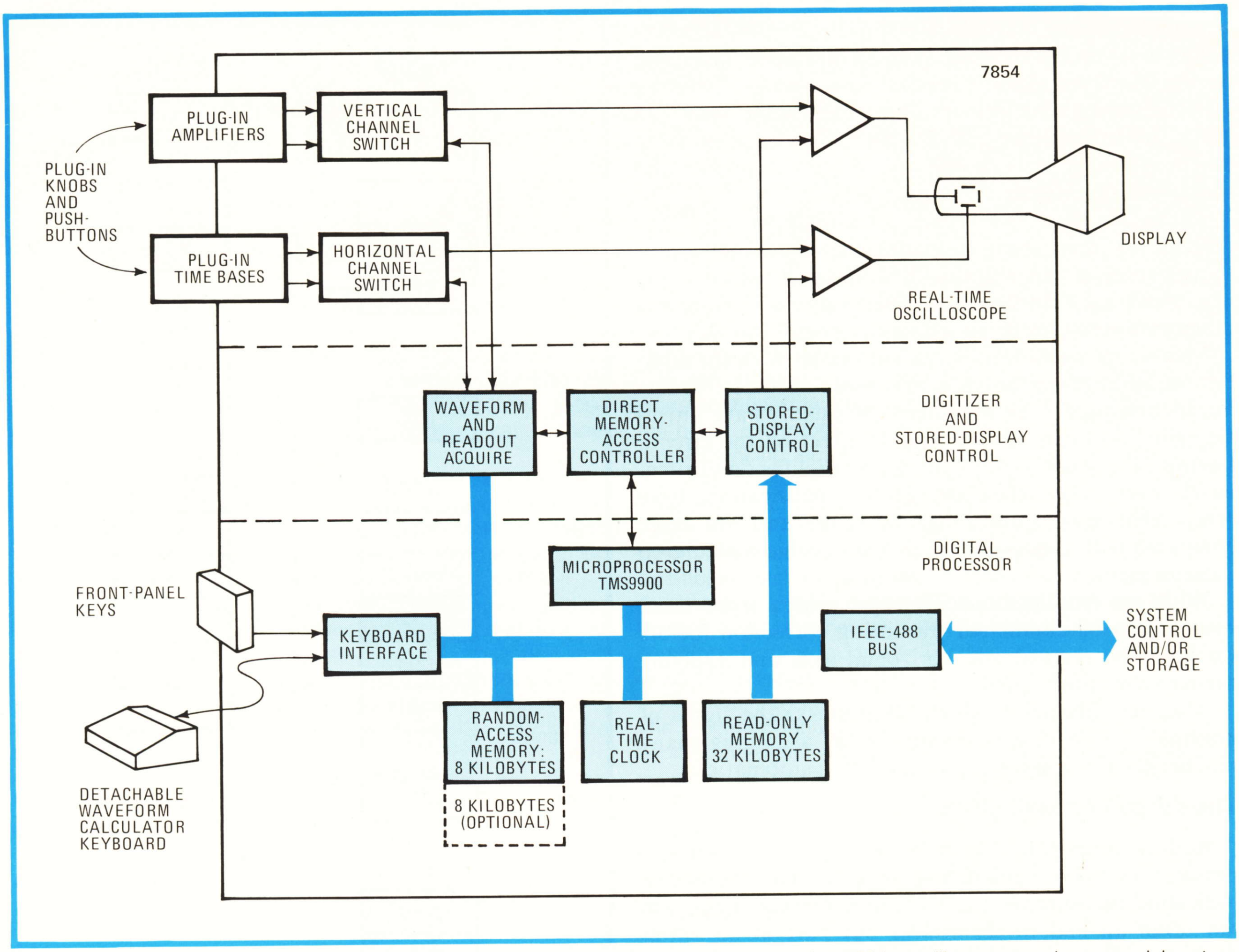
The Tektronix model 7854 is an evolutionary step in oscillography, combining in a single unit, to a degree never before achieved, the data density of analog waveform displays and the power of digital processing.

The concept of an oscilloscope able to do digital waveform processing is not new; in the early 1970s, designers at another company were trying to build such an instrument but ran up against the problem of trying to design digital instruments before the invention of the right tools. They turned their attention to making those tools and succeeded, but their original goal was not achieved until now.

For systems use, Tektronix developed a digitizing oscilloscope in 1972. It is a combination 7704A oscilloscope and P7001 digitizer designed for use with an



Evolutionary. The 7854 is an important step in the progress of instrumentation, combining for the first time in a single unit the most powerful of analog instruments, the oscilloscope, with the present standard-bearer of digital processing, the 16-bit microprocessor. The result of this synthesis is a scope able to perform traditional and untraditional analog measurements simply and precisely by digitizing and storing waveforms.



1. Three tiers. Circuits in the top section of the functional block diagram shown let the 7854 act like a general-purpose laboratory oscilloscope, those in the middle let it capture and display waveforms like a digital storage scope, and those at the bottom let it compute.

external minicomputer and works at frequencies to 175 megahertz or 14 gigahertz with a plug-in sampler. A total system costs anywhere from \$27,000 to \$40,000.

The goal of providing a design tool that could fit on a lab bench—with the power of a minicomputer and a scope in one package at a reasonable cost—could not be met until the advent of the 16-bit microcomputer. Only with a 16-bit processor could an instrument be made adequately "intelligent"—able to quickly perform complex tasks at the push of a single button, a paramount goal for lab use. Yet even after such processors began to become available, adding digital power to an analog scope was not a simple task.

A scope's scope

An understanding of the significance of the 7854 and why it was difficult to design involves putting scope measurements into perspective—seeing what measurements are now made with scopes and what kinds users would make if they could.

All oscilloscopes deal with waveforms, or two-dimensional entities that consist of a practically infinite number of values and in the majority of instances are repetitive. The scope not only presents all waveform values simultaneously, but also does so in such a way that the relationship between them is instantly visually apparent.

But though it may be easy to visualize relationships and values immediately with a scope, quantifying them is another matter. Any measurement results in a number or a yes/no decision.

There are, therefore, many numerical values that the user must often take from the scope's screen. Some typical measurements are maximum, minimum, and midpoint of a signal. Rather than try to eyeball these readings, a user can make them more accurately on the 7854 by simply pushing a single key. The value of the selected parameter then appears on the screen.

Other typical measurements usually involve more than simply reading a number directly off the screen. To measure the peak-to-peak value, frequency, or period of a wave, the user has had to read one or more points from the screen and then do some mental arithmetic (addition, subtraction, multiplication, or division) to get the desired number. With the 7854, all he or she need do is position a couple of cursors to define the measurement locale and push one key to get an accurate, decimal readout on the screen.

Pulse parameters are generally tough to measure with an ordinary oscilloscope. To measure the rise or fall time of a pulse, a user first has to estimate the 10% and 90% points between the base and plateau of a pulse, find them on the displayed signal's vertical axis, and estimate the time between them from the horizontal axis. With the 7854, this measurement is also performed by positioning two cursors and pressing one key, as are pulse-width and -delay measurements.

There are also less typical measurements that are difficult to read from a scope now. The root-mean-square value is one, and the 7854 makes it with a single key. Pulse area, energy, and mean value of a wave are other parameters that can also be measured this way.

Measuring waveforms is not only easier with the addition of intelligence, but it is also more accurate. Transducers and probes for measurement usually have intrinsic errors and can have a loading effect on the circuit. Further, the scope's own calibrated amplifiers and time-bases, even when they are within specification, have errors that vary from range to range and are large compared with the resolution that is possible with digital measurements.

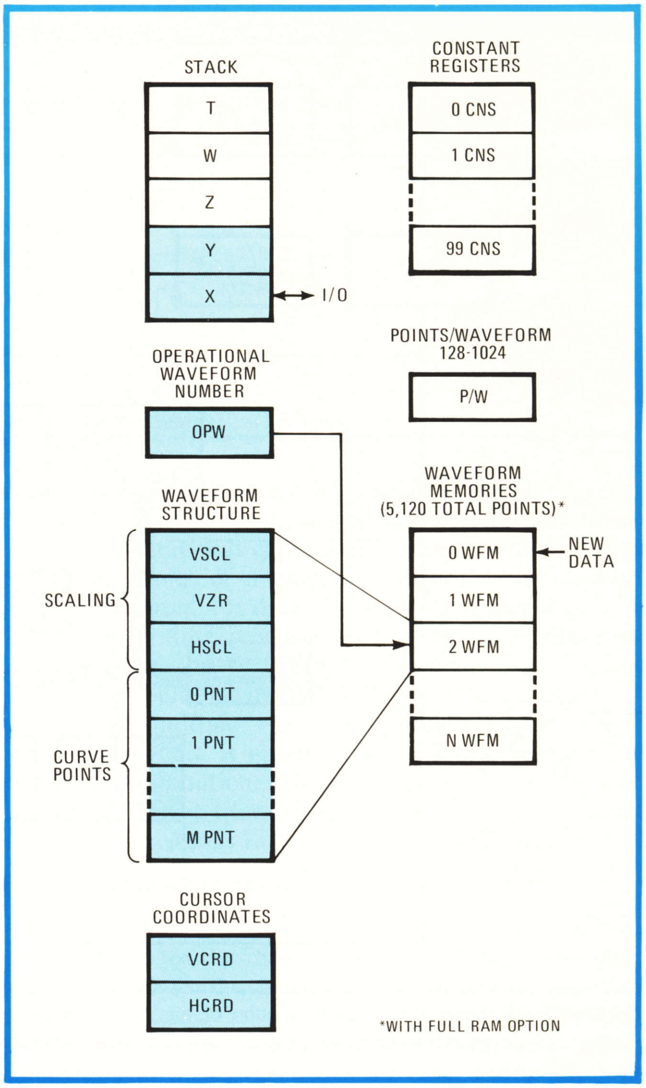
With an intelligent oscilloscope, errors from these sources can be eliminated by storing correction factors for both the vertical and horizontal axes and applying them to measured signals.

Also, random noise, which often masks the signal of interest, can be largely eliminated by averaging many repetitions of the input, a process that is automatic.

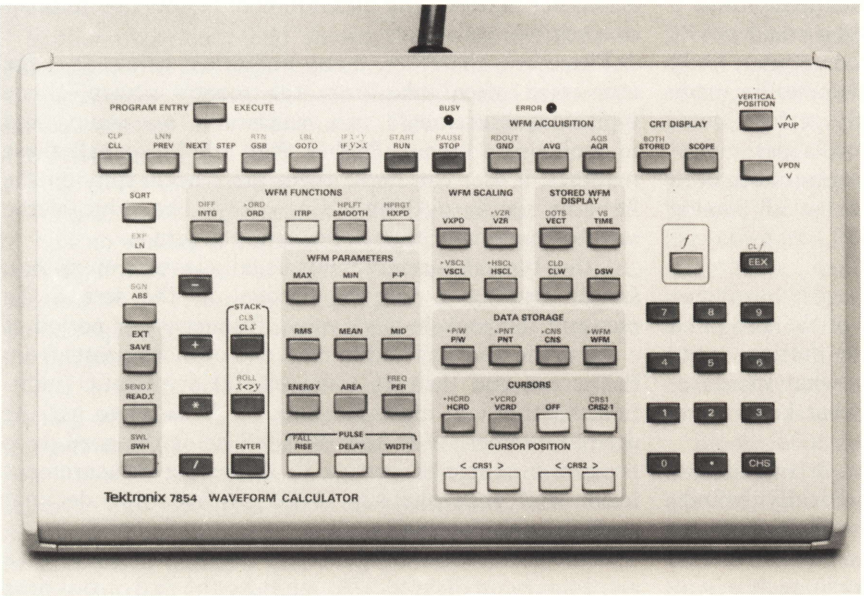
The scope of possibilities

Adding intelligence to a scope, however, opens up possibilities beyond enhancing the ease and accuracy of individual measurements. Waveform measurements are not often an end in themselves but part of an entire process, such as system design or incoming inspection. An intelligent scope can refine an entire process, such as circuit evaluation, not just single measurements.

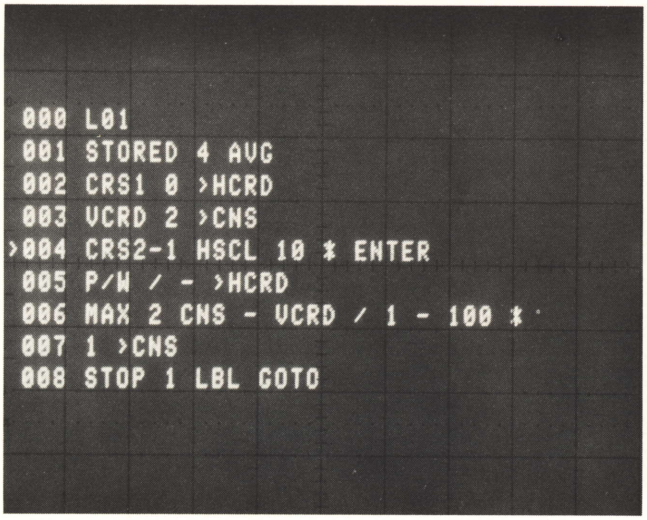
Circuits are usually designed conceptually using ideal resistors, capacitors, flip-flops, and other components, all of which follow known mathematical laws. With the 7854, a designer can calculate and store the ideal



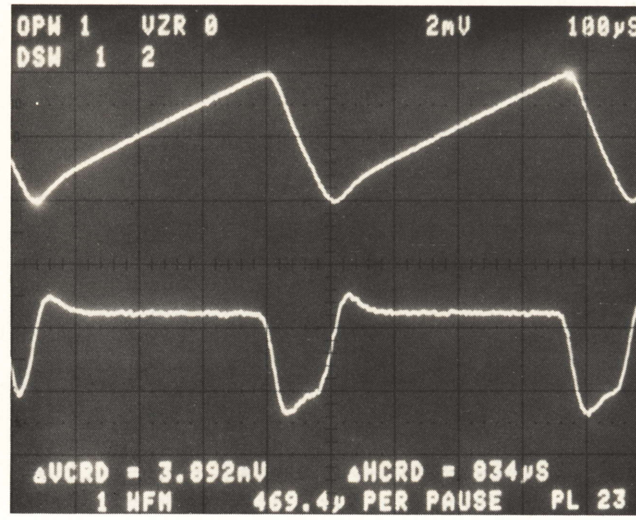
2. Registers. To process waveforms, the 7854 provides a register structure like scientific calculator's, with stack registers for operations and storage registers for constants and waveform data. The contents of the shaded registers appear on the scope's display.



3. Waveform calculator. The specially designed detachable keyboard, like the scope's registers, is based on scientific calculator practice, adapted to handle waveforms. Groups of related waveform operators and programming functions have been set off by shading; operators are at left, operands at right.



4. From the calculator. Measurement programs like the one shown here are written using the waveform calculator. Each line of a program executes separately, so results can be displayed with the program status. Labels are used for branching.



5. From storage. Four lines on digitally stored waveform display tell user what is happening. The top line says which wave is being worked on (1) and its scaling; next, which waves are shown. Last line gives measurement results (469.4-μs period) and program status.

circuit's response to typical waveforms (sine wave, pulse train, ramp, etc.).

The output of this process can simplify the building of the real circuit. The ideal response can be stored in the scope, a real circuit built, and the response of that circuit compared with that of the ideal. Real components can be tweaked and the circuit output reexamined to optimize the real design.

The intelligent scope can be used in an automated test equipment environment. In incoming inspection and quality control, for example, a test equipment operator now visually checks an actual waveform to see that it falls within certain well-defined tolerances. Tolerance indications are usually placed over the scope screen, using anything from grease pencils to expensive custom overlays. Because trace width, parallax, and display distortion can affect the reading, the operator must be highly proficient.

But an intelligent scope can store tolerance waveforms and automate comparison so that the operator can perform complex tests in go/no-go fashion. Further, with a general-purpose interface bus, it can put out the results for further analysis or record keeping.

Making possibilities real

In designing its new scope, Tektronix adopted a philosophy it calls progressiveness. The idea was not to shock the user by introducing a completely new and unusual instrument, unrecognizable as an oscilloscope to any engineer or technician. Rather, the user must be able to progress from the familiar to the unfamiliar easily. Applying the progressive philosophy to the 7854 meant that total redesign was avoided.

When used for real-time display of analog signals, the 7854 operates exactly like an oscilloscope in every respect. It accepts any of the existing 7000-series plug-ins: amplifiers, timebases, counters, timers, and spectrum analyzers, to name a few. It may not be a simple instrument, but it presents a familiar, friendly face to the user, letting him or her use it even without an understanding of its digital capabilities.

This friendliness was kept by functionally segmenting the scope into three basic subsystems: the real-time oscilloscope, the waveform-acquisition (digitizer) and stored-display control subsystem, and the digital processor, shown in Fig. 1. As shown, the familiar, real-time controls remain separate from those that bring the digital processing capabilities into play in conformance with the basic functional design philosophy.

Unlocking digital doors

Deciding how to give access to the processing capability was a major challenge. There were many choices to make: what language to use, what functions to provide, which algorithms to use, how to arrange the keyboard and label keys, and where to put information on the display. The progressive philosophy again provided the answer.

Access to the scope's digital capabilities was on three levels, to match the proficiency levels of various users. On the first level are the front panel keys that call into play some of the processing capabilities. They correspond functionally with those measurements called typical or would-be typical earlier: maxima and minima, peak-to-peak, pulse rise and fall time, rms, and so on. Since to measure a signal digitally it must first be digitized, there are keys to acquire the waveform digitally. Likewise, there are the keys for the cursors needed to set up those measurements.

For the second level, a separate, detachable keyboard was designed that would let a design engineer working at a bench more fully utilize the scope's capability. This part of the functional design influenced the other levels significantly and proved the most challenging.

For several reasons, the keyboard was built around a calculator, or key-stroke, language rather than a popular programming language like Basic. First, extending a language like Basic so that it would be able to perform waveform operations would dilute its familiarity. Using a key-stroke language, each function is represented by a single labeled key. The system's capabilities are easy to learn and remember. Another advantage of a key-stroke language is that it permits one-finger push-button operation, which is more at home in the lab than two-handed typing.

RPN's advantages

Having narrowed the field to a calculatorlike language, there were two choices: reverse Polish notation (RPN) or algebraic syntax. Here the choice was relatively easy—RPN had all the advantages.

RPN is a syntax-free language—each operation depending only on the operator and operand—that allows great freedom in tailoring a set of functions for waveform operation. Each operation is performed independently of all others and provides a result immediately. Thus, the name of the operation just performed and the result of that operation can be displayed together. Further, if a function cannot be executed because of an improper operand (the natural logarithm of a negative number, for example), the cause can be, and is, automatically frozen on the display to alert the operator. This capability eliminates the confusion that can occur when a long arithmetic expression results in an obviously wrong answer.

Further forcing the choice is the fact that more engineers are familiar with RPN. Several forms of the algebraic notation are used in calculators, and though they differ on how to perform simple arithmetic functions, they do manage to agree on how to perform one-operand functions—they all use RPN.

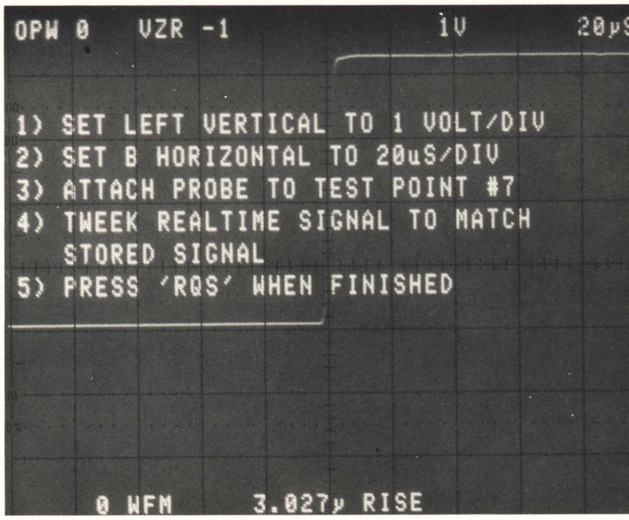
The backbone of an RPN calculator is its data register structure, and so the hardware design criteria (about which more will said later) called for an abundance of them (Fig. 2). The stack and constant registers needed for any calculator were provided, as were waveform registers. Here waveforms are recorded in terms of the vertical scale factor (VSCL), vertical zero with respect to ground (VZR), horizontal scale factor (HSCL), and the digitized curve points, in that order.

So that users can trade off the number of waveforms stored for the resolution with which they are recorded, the number of digitized points per waveform (P/W) was made selectable. A user can pick 128-, 256-, 512-, or 1,024-point digitizing and that selection is stored in the P/W register.

Another pair of registers stores either the vertical and horizontal coordinates of a single displayed cursor (VCRD and HCRD, respectively) or the difference in coordinates between two displayed cursors (ΔVCRD and ΔHCRD).

As with any other calculator, the operands and the result of the operation should go into the stack. Though this procedure works perfectly well for constants, an entire waveform would take up more room than the stack could economically provide. For this reason the waveform number only is put in the stack as a pointer, and individual points are fetched from waveform memory as they are required.

To handle the results of the operation, a rule was adopted that new waveform data would be transferred into waveform memory 0. The rule was broadened to include all new memory data, even that acquired by digitizing, and since two waveforms can be digitized simultaneously, waveform memory 1 was also set aside. After an operation or acquisition is complete, the digitized waveform can be transferred to another memory area for more permanent storage.



6. From afar. For an ATE environment, the 7854 can accept information and display instructional text like that shown through its IEEE-488 interface. Its ability to store tolerances simplifies testing, increases accuracy, and eliminates need for screen overlays.

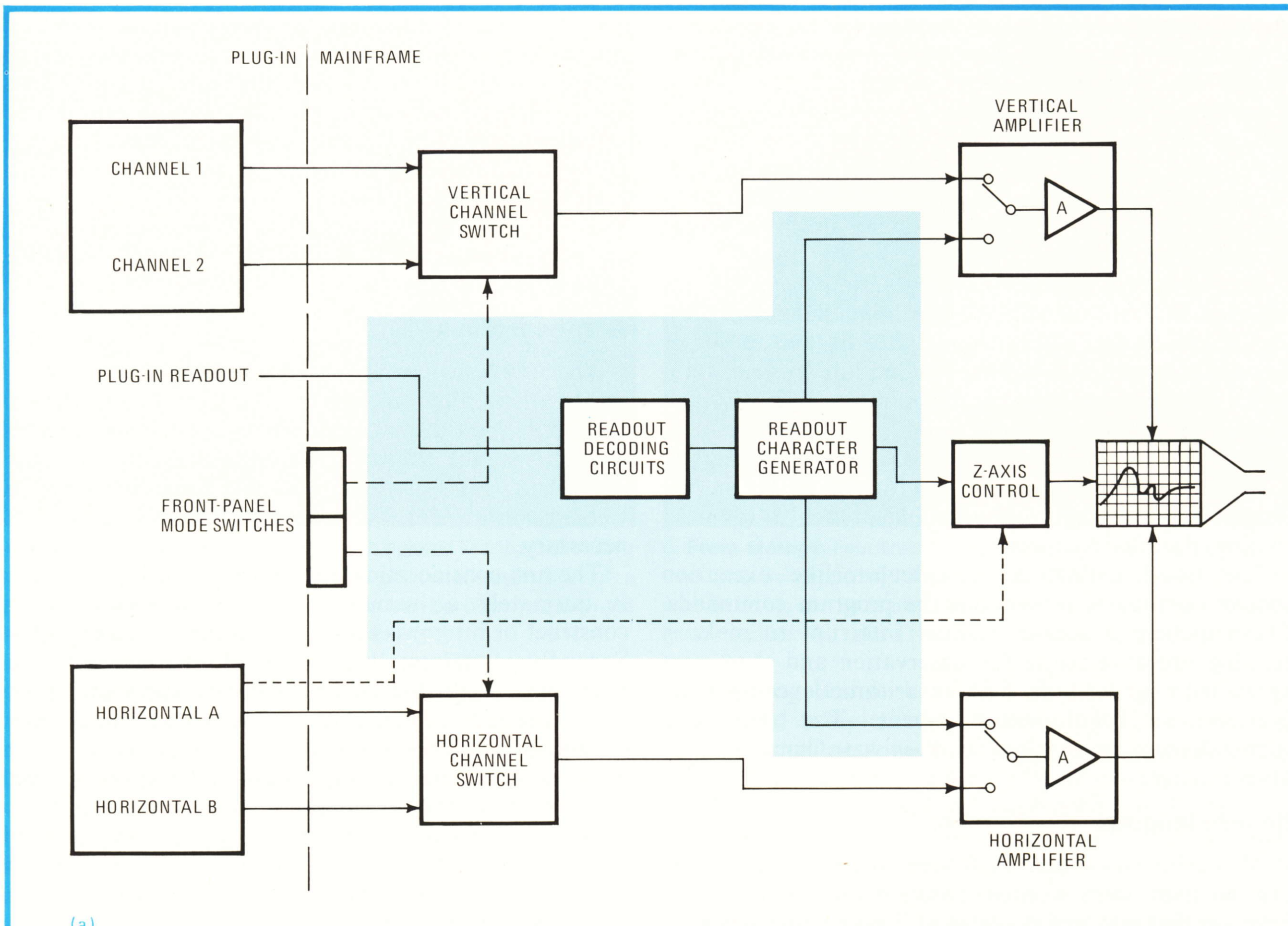
The calculator concept also led to the introduction of the operational waveform (OPW): the waveform whose number has appeared most recently in the X register is designated the operational waveform by transferring its number to the OPW register too. Several operations are often performed in sequence on the same waveform operand, and designating the OPW waveform as the default operand minimizes key strokes. The OPW waveform is also kept on the display to prevent it from going blank and to give the measurement cursors a place to reside.

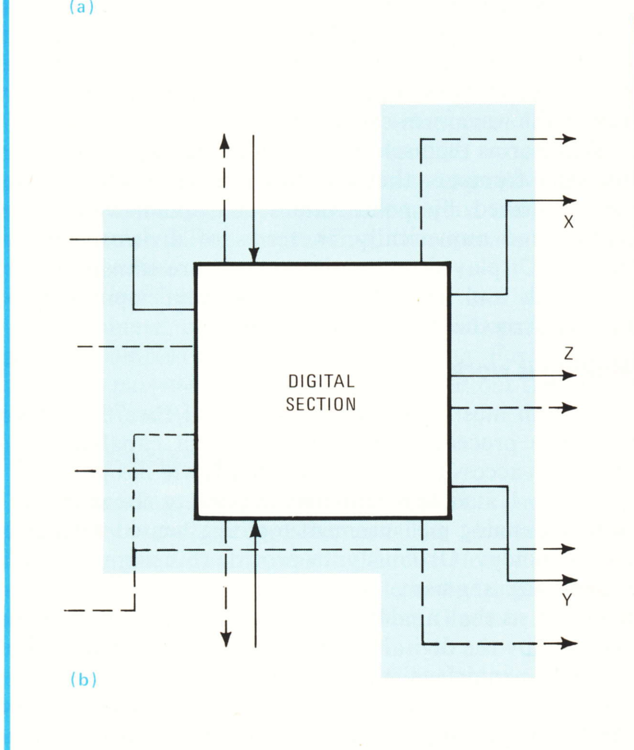
In addition to the register structure, the algorithms that could be provided to operate on waveforms strongly influenced the calculator design. The choice of algorithms and their implementation was by interpolation.

In implementing any measurement or operation on a digitized waveform, some assumption is made about what happens to the waveform between the sampled, digitized points—an assumption voiced in the interpolation method.

When samples are taken very frequently, the choice of interpolation method is less critical; waveforms tend to be regular in their wanderings, except in infrequent or specialized cases. Therefore a simple linear (connect-the-dots) interpolation method was implemented in the 7854. Easily visualized by the user, it matches the linear vector nature of the stored waveform display.

Using this method, algorithms for various measurements and operations can be based on previous visual techniques. Period and frequency measurements, for example, are simply the difference or reciprocal of the





7. A-d conversion. Careful modification of channel switches, amplifiers, and Z-axis control retained scope's familiar analog operation while permitting control by digital section (b), which replaces shaded area in (a) of waveform capture, storage, and display.

differences in time between successive crossings of a reference level (usually the horizontal graticule on an oscilloscope).

With the functional attributes of language register organization and types of algorithms determined, several groups of command keys became obvious, as seen in the waveform-calculator keyboard shown in Fig. 3.

Choices

One group acquires waveforms and chooses which information—real-time, stored, or both—is put on the cathode-ray-tube display. To control the display of the stored waveform, other keys choose between dot and connected—or vector—displays, between waveform-versus-time and waveform-versus-waveform displays, and between clearing the entire screen or just the waveforms. Keys are also provided to vary vertical and horizontal scaling and to position stored signals on the display.

The data-storage key set permits entry of constants, points per waveform, and waveform numbers using the numeric keys. Keys for controlling the number and position of cursors are beneath. Single-key measurement keys are provided too, and there are keys to control the register stack.

There are simple arithmetic keys and some commonly used algebraic functions like natural log and exponential. There is a group of functions that relate solely to waveforms—integration, differentiation, smoothing, and linear interpolation (for constructing waveforms from a few points). Horizontal expansion was included here, rather than with the display control keys, because it causes a new waveform to be computed from part of the old and the remainder to be lost.

Practical application of these individual functions, or commands, always involves performing them in sequence, that is, as a program. So one group facilitates generation of programs.

The waveform calculator's programming operation is modeled on that of other programmable calculators. Contiguous line numbers are assigned to lines of one or more commands, and editing is done line by line. Individual keys move a line-pointing cursor a line at a time or slew it through many lines.

Since line numbers can change as a program is edited, a facility was provided to label target points in the program so that the program could branch to different routines during execution.

The usual collection of calculatorlike execution control commands rounds out the program commands. These include a simple manual interrupt to make a running program pause for observation and then start again with no loss, as well as arithmetic comparison operators for conditional statements. The latter have been enhanced so that they work on waveforms as easily as on constants.

No new languge

All keyboard commands (except the edit commands) can be used in a written program, and any written program that is stored operates as if each command were entered manually. This not only makes it easier to debug or patch a program, but it also means that the user does not have to learn a new language to write a program—a progressive approach. A typical program is illustrated in Fig. 4.

The arrangement of all command keys minimizes hand motion. Although key-stroke sequence cannot be predicted for certain, the typical sequence with RPN is operand-operator. This led to grouping operands on the right and operators on the left, which keeps the next key in an operation visible to the user. Related keys are visually united by shading, and different colors and shapes distinguish functional groups.

Labeling individual keys was an important consideration. There was room on the panel for two- or three-word function descriptions, but it was an iron-clad design rule that command names—whether they appeared on the keyboard, on an IEEE-488–compatible controller, or on the scope's own display—would always appear in the same form. As a result, brief mnemonics were used.

The display in Fig. 5, generated by a stored program, contains the operands, function status, and results of the operation. The top line shows the number of the operational waveform—OPW 1, in this case—its offset from the vertical zero reference, and the scale settings. The second lists the waveform numbers of both of the signals displayed.

On the bottom of the screen, the difference between the positions of the two cursors' coordinates is shown. Below the coordinates are the contents of stack registers Y and X (waveform 1 and the period of that waveform, respectively). Next to the register contents are the program status (PAUSE) and line number (23). Thus, using the keyboard mnemonics, the display provides the complete story of the scope's program status.

With intelligent instruments becoming part of larger systems in many applications today, this third level of scope application evolved naturally. Accordingly, it was the obvious next step to provide the 7854 with an IEEE-488 bus interface.

At the system level

The 7854 will transmit and receive three types of information: program commands, data (waveforms or constants), and display text. Though the interface standard defines the control signals and hardware interconnection for bus systems, it does not define the language to be used in interdevice communications, so rules were necessary.

The first consideration was that any machine interface is ultimately a human interface—somebody has to construct or interpret the messages that flow through it. Secondly, to maintain the concept of progressiveness, the language used at the keyboard should apply identically to the interface. And finally, any output used as an input should re-create the original state it was derived from. This last consideration was necessary for operation with nonintelligent mass storage.

So it was decided that a program put onto the GPIB would be an ASCII representation of that program as it appeared on the display—command mnemonics separated by spaces—with line numbers replaced by line terminators to satisfy the final rule. The same mnemonics when received as input are handled by the 7854 as if they came from the keyboard, so the GPIB can press any key on the waveform-calculator.

Waveforms themselves are transmitted and received in the same format as they are stored in memory, as scaling data followed by point values, but point values are transmitted numerically, in terms of divisions on the display. Displayed text and constants are transmitted as displayed, and any EIA-compatible text input can be displayed on the screen also (Fig. 6).

Making it work

As with most modern design, that of the 7854 was a top-down procedure, with measurement capability and the user's access to it forcing the hardware design.

In Fig. 1, it is apparent that in concept the scope was to be an analog measurement tool augmented with digital technology. Obviously, to provide this augmentation, the analog segment of the scope would have to be modified so that analog signals acquired by it could be accessed by the digital subsystem. Keeping to the goal of a familiar interface, the analog portion of the scope is very similar to other members of the 7000-series—the 7904 general-purpose single-beam dual-trace oscilloscope in particular.

Figure 7 shows the modifications of the mainframe to accommodate digital processing. An additional output port and control switch was added to both the vertical and horizontal channel so the signal could be acquired,

Digitizing with display in mind

Like other 7000-series laboratory oscilloscopes, the 7854 was to operate in real time with any of the numerous plug-ins built for that familiar family. This meant that the new scope would have to digitize an input regardless of what plug-in was used. To provide this capability, Tektronix devised a display-oriented random-sampling technique.

For this technique, the display is regarded as consisting of 128, 256, 512, or 1,024 horizontal and 1,024 vertical locations, as shown at the left of the figure below.

In a real-time scope, these display locations are addressed with analog inputs from the horizontal and vertical plug-ins, which are amplified and used to drive the deflection plates. To capture the signal digitally, those signals are fed by the horizontal and vertical channel switches to separate sample-and-hold circuits.

A free-running clock controls the sampling process, turning on a Schottky diode bridge that lets the signal charge the sample-and-hold capacitor. Samples are taken at the clock's frequency so they generally appear to be random with respect to the signal.

The sample-and-hold circuits simultaneously acquire the vertical and horizontal deflection voltages and do so at each clock cycle. The values from the horizontal and vertical sample-and-hold capacitors are multiplexed in succession to a successive-approximation a-d converter, which digitizes the value and transfers it to the appropriate output latch.

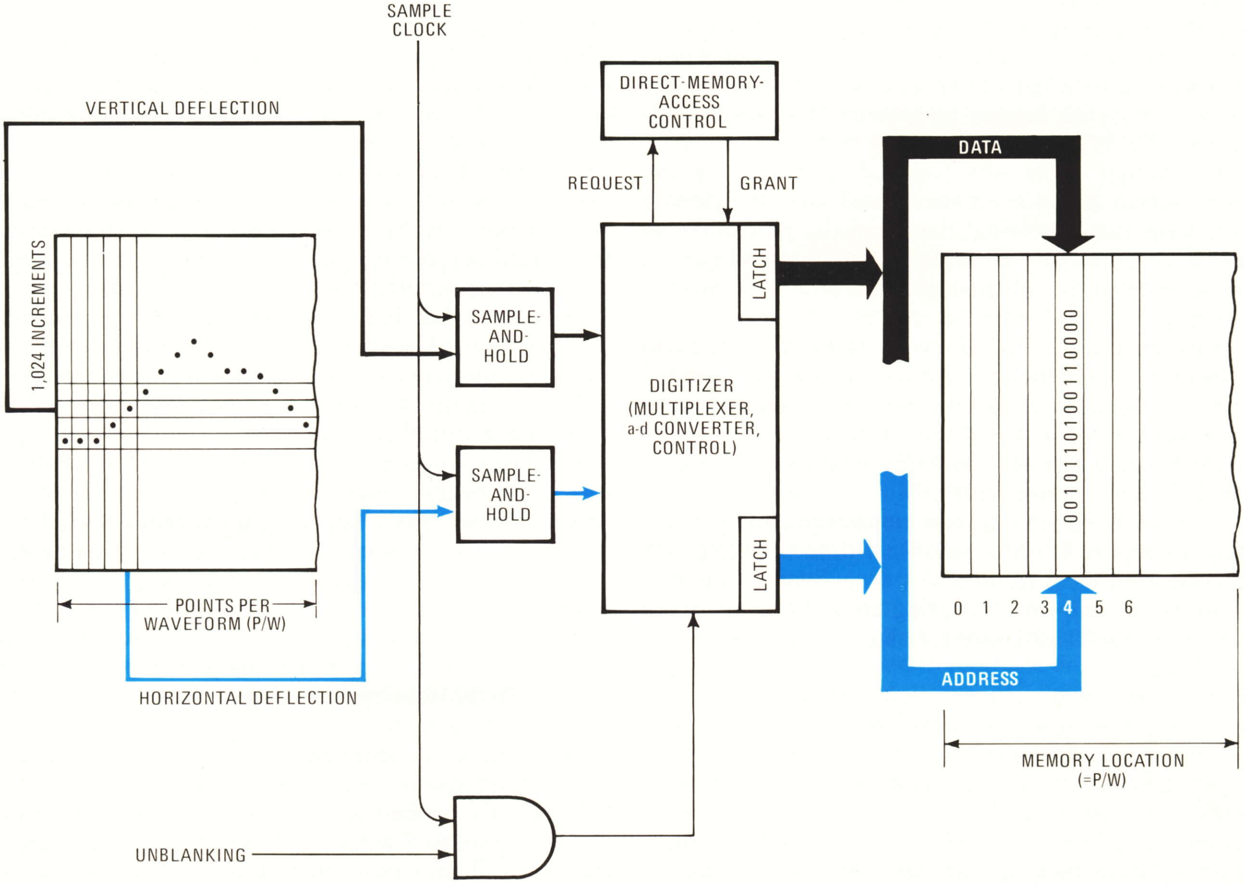
Successive approximation was selected as the conversion method since it was not too expensive to implement, yet provided sufficiently rapid conversion. Dual-slope conversion would have been too slow, and a flash (or parallel) conversion technique would have required too much space, power, and money. A multiplexing scheme was chosen because, though dual converters would have improved the total conversion rate by 30%, they would have increased the a-d conversion cost by almost 100%.

It takes 1 microsecond for the digitizer to perform a single 10-bit conversion. Both horizontal and vertical signals must be converted, which takes 2 μs, and the information transferred to memory, which takes 1.5 μs. This gives an overall digitizing rate of 3.5 μs.

In the storage mode, the conversion process takes place continuously regardless of the horizontal location of the beam. Therefore, to prevent the storage of a retrace, the unblanking signal of the scope is used to gate the digitized signals onto their respective buses. It also initiates a direct-memory-access request.

The 10 bits that result from digitizing the horizontal signal are used to form part of the storage address. Additional bits are supplied by the microprocessor to designate a block of memory addresses for waveform storage and, together with the horizontal bits, form the complete storage address. To this address, the 10-bit result of the vertical signal conversion is written.

After each address is filled, a flag is set to indicate this. The process continues, with samples taken on each repetition of the signal until a minimum of 99% of the allocated waveform memory locations are filled. So memory now contains the vertical values of the waveform in the horizontal sequence in which they were displayed.



MEMORY-ADDRESS CODING SCHEME																	
Points/ waveform	Address-word bit source																Address range (hexadecimal)
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1,024	R	R	R	R	R	D	D	D	D	D	D	D	D	D	D	0	A000 → A7FE
512	R	R	R	R	R	R	D	D	D	D	D	D	D	D	D	0	A000 → A3FE
256	R	R	R	R	R	R	R	D	D	D	D	D	D	D	D	0	A000 → A1FE
128	R	R	R	R	R	R	R	R	D	D	D	D	D	D	D	0	A000 → A0FE

R = bits assigned by processor to designate memory range D = bits from digitized horizontal signal

and, similarly, an input port and control switch to the horizontal and vertical amplifier so that stored signals could be displayed.

The new scope was to accept all 7000-series plug-ins, and those plug-ins have always provided readouts of their control settings. The digital section needs these to give correct scaling information on stored signals, so they are routed to that section.

Front-panel mode switches (vertical left, right, alternate, chop and add; horizontal A, B, alternate, and chop) were also routed through and controlled by the digital section to make them programmable. In ATE environments, this provides a certain degree of control over the input format. Providing remote control of the horizontal and vertical channel ranges was also considered, but the cost and complexity of such a feature would not have been in keeping with a laboratory instrument and would have required redesign of the plug-ins. In view of the number of plug-ins in user inventories this was considered particularly unadvisable.

Equal design effort was required in modifying the analog portion and adding the digital section. Hard to satisfy were two functional design goals: providing the high-resolution display from the stored data and storing waveforms up to the full analog bandwidth—400 MHz—of the scope.

Earlier experience had shown that 8-bit digitization was insufficient for high resolution; 10 bits were needed to characterize accurately both the horizontal and vertical coordinates of a waveform. This would provide a maximum resolution of 1 in 1,024 in each axis or 1 in 1,048,576 for the entire screen.

With a 10-bit digitizing scheme chosen, the question of how to realize it remained. Real-time sampling, the technique used most for digital storage scopes, was out, because to do real-time sampling on a 400-MHz signal would require a 10-bit converter able to work at 1 GHz at the minimum.

The answer was a new digitizing scheme called display-oriented random sampling. In this scheme, samples are taken on the vertical and horizontal channels for different points in the waveform each time it repeats, until at least 99% of the points have been digitized. The digitized horizontal coordinate becomes an address in memory and the digitized vertical coordinate is the data stored there. The implementation of this is discussed in "Digitizing with display in mind."

The choice of 10-bit digitization, along with the functional, computational capabilities to be provided, determined the choice of microprocessor. An 8-bit machine could not be used, for data would have to be broken into 2 bytes each time it was stored or operated on, thus slowing down the digitizing, computing, and display process. It would not provide sufficient address capability to store multiple waveforms, programs, and algorithms, nor would it provide the word width to ensure accurate computation. Therefore a 16-bit device was needed and at the time design began, such a device was available from Texas Instruments: the TMS 9900.

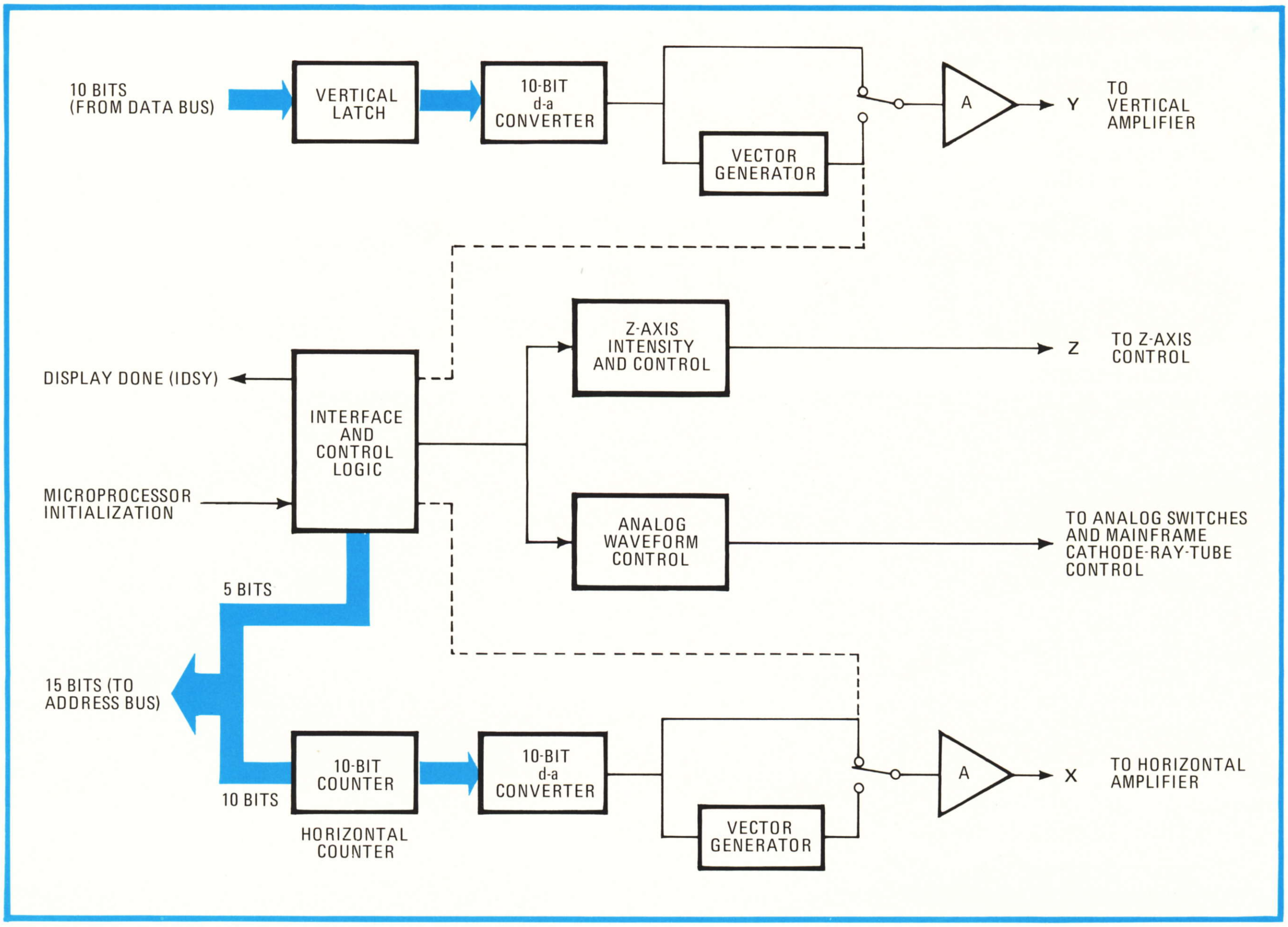
Serendipitous processor

The features of that processor were particularly advantageous for the 7854. It had 16 vectored interrupts, which permitted the fast switching between different operations needed to react to changes in instrument status, such as the receipt of a GPIB command or the completion of an assigned display task. Using RAM to switch to a new set of working registers while saving the old required only that a workspace pointer be set. It had 16 general-purpose registers that eliminated the need to save and load accumulators, speeding processing.

In addition, the TMS 9900 has request and grant lines that can be used to clear the data and address buses for direct memory access. Thus waveform data can be stored and retrieved quickly, reducing total digitizing time and providing fast display, respectively.

Firmware for the system resides in 32 kilobytes of ROM; field-programmable logic arrays and PROMs can be used for software patches. The system's RAM board, organized as 4,096 words of 16 bits, using 1,024-by-4-bit static RAMs, can be expanded to 8 kilowords when more waveform/program storage is needed. The 7854 can be configured with backup power for transportation, and codes entered into RAM when line power is removed are checked to be sure memory integrity is retained when line power is restored.

Blocks of RAM addresses for waveform data storage are assigned by the processor in response to the user's specification of points per waveform and waveform number, as shown in Table 1. Vertical coordinates are then entered into those addresses, with overrange bits filling the first two places in the 16-bit word, followed by



8. Bits to pixels. To translate stored waveforms into a display, the display board's interface and control logic is initialized by the processor for the area where the waveform is stored, and then calls the waveform point by point by incrementing a 10-bit counter.

the 10 bits of point data and 4 guard bits (zeros) to make up the full word. Scaling information acquired from the horizontal and vertical plug-ins fills the first three address locations.

The display board shown in Fig. 8 puts stored data on the screen. When a waveform is to be displayed with respect to time, the microprocessor sends initialization information—the first address of the waveform in memory and the number of points stored—to the board's interface and control logic. It then starts the display to allow DMA control of the data and address buses so the display board can directly access memory.

The interface and control logic uses the initialization information to set the display board counter and pull the point from memory. The point and counter settings, which reflect the memory location, are converted to analog values and used to drive the scope's vertical and horizontal deflection plates, respectively. After the point is displayed, the counter is incremented and the next point fetched. This process continues until the counter value equals the number of points per waveform supplied at initialization. Then the interface and control logic signal the processor that the display is complete, and the processor resumes control of the buses. If multiple waveforms are displayed, the process is repeated for each.

A problem that had to be solved in the display board design was how to display a real-time waveform while showing multiple stored waveforms, in order to achieve a flicker-free display. With a budget of 8 microseconds per dot and a minimum refresh time of 20 milliseconds for flicker-free display, 2,500 dots can be displayed, or approximately two stored waveforms of 1,024 points.

The actual time the beam must be on for a dot to show on the screen is 5 μs, which leaves 3 μs for a real-time waveform. But the time required to switch between stored and analog display, write the analog wave portion, and allow for settling between that switching is more than 3 μs.

To allow multiple displays, the display board can be initialized by the processor to show odd-numbered memory contents on one scan and even on the next. This increases the time for real-time display by the budget for the omitted dot, 8 μs, to a total of 11, which is more than sufficient.

Displays of one stored waveform against another can also be generated by the board, in which case two waveform points are called consecutively from memory to drive the deflection amplifiers. This increases the time to generate a dot to 20 μs. The display also controls cursor display, halting the counter increment at the cursor's horizontal location when given that information by the processor. The vertical coordinate at the time the counter is stopped and repeated, creating a brighter spot at that location on the screen. □

For the address of your nearest Tektronix Field Office, contact:

U.S.A.
Tektronix, Inc.
P.O. Box 1700
Beaverton, OR 97075
Phone: 800/547-1512
Oregon only 800/644-9051
503/644-0161
Telex: 910-467-8708
Cable: TEKTRONIX

Africa, Europe, Middle East
Tektronix International, Inc.
European Marketing Centre
Postbox 827
1180 AV Amstelveen
The Netherlands
Telex: 18312

Asia, Australia, Canada, Central & South America, Japan
Tektronix, Inc.
Americas/Pacific
P.O. Box 1700
Beaverton, OR 97075
Telex: 910-467-8708
Cable: TEKTRONIX

Tektronix offices to serve you around the world:
Argentina, Australia, Austria, Belgium, Bolivia, Brazil, Canada, Chile, Colombia, Costa Rica, Denmark, East Africa, Ecuador, Egypt, El Salvador, Federal Republic of Germany, Finland, France, Greece, Hong Kong, Iceland, India, Indonesia, Iraq, Israel, Italy, Ivory Coast, Japan, Jordan, Korea, Kuwait, Lebanon, Malaysia, Mexico, Morocco, The Netherlands, New Zealand, Norway, Pakistan, Panama, Peru, Philippines, Portugal, Republic of South Africa, Saudi Arabia, Singapore, Spain, Sri Lanka, Sudan, Surinam, Sweden, Switzerland, Syria, Taiwan, Thailand, Turkey, Tunisia, United Kingdom, Uruguay, Venezuela, Zambia.

Tektronix®
COMMITTED TO EXCELLENCE

AX-4440