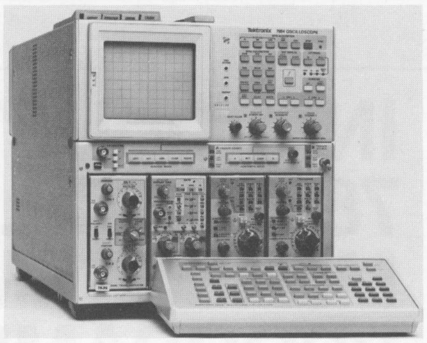
TEK 7854

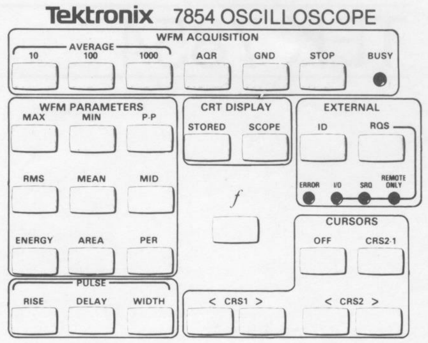
APPLICATION NOTE
42AX-4682

INTRODUCTION TO 7854 OSCILLOSCOPE MEASUREMENT AND PROGRAMMING TECHNIQUES

Tektronix®
COMMITTED TO EXCELLENCE

Introduction to 7854 Oscilloscope Measurement and Programming Techniques





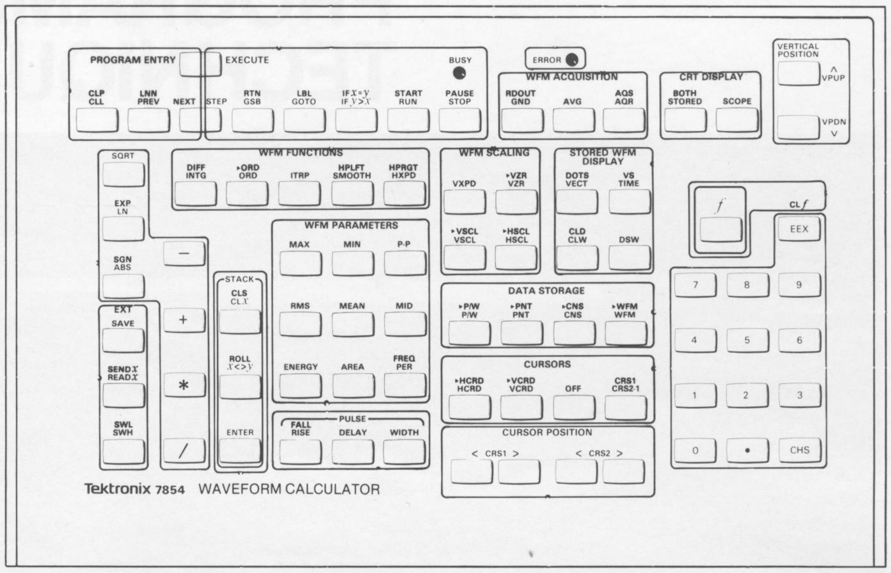


Figure 2. Measurement key pads on the 7854 oscilloscope

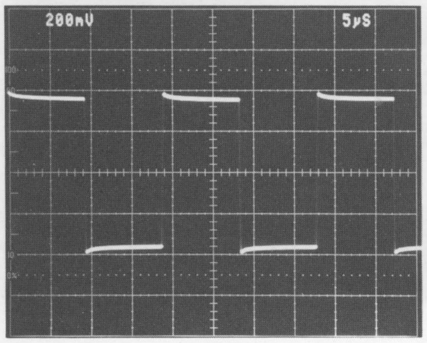
The TEKTRONIX 7854 Oscilloscope is an instrument designed to open new vistas in bench-top measurements. As a general-purpose measurement tool, the 7854 is first of all a high-performance oscilloscope with a 400 megahertz bandwidth (when used with the 7A19 or 7A29 Amplifier plug-in). The 7854 contains all of the standard features of the TEKTRONIX 7000-Series line of oscilloscopes. But these features address only the traditional aspects of oscillography.

The greater vistas of the 7854 come from its waveform digitizing capabilities and its microprocessor-based measurement features.

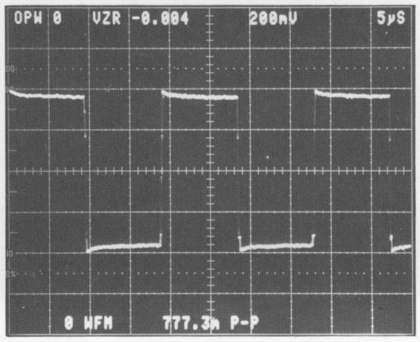
Waveform storage is no longer accomplished by employing special phosphors on the CRT screen. The 7854 digitizes waveforms by taking amplitude samples of the input signal. These signal samples are then stored in the 7854's digital memory, from which they can be recalled for clear, crisp display on the CRT screen (see figure 1). This storage and recall process eliminates the blooming and fading problems of older screen storage methods. But more importantly, 7854 digitizing and storage techniques give you waveform data with resolutions equivalent to 0.01 display divisions vertically and as much as 0.01 display divisions horizontally. That means at least an order of magnitude improvement in data collection over traditional visual techniques.

Even more important, though, is the 7854's ability to repeatably pick specific data points from waveforms. Take peak-to-peak measurements as a simple example. With a standard oscilloscope, the waveform display is positioned to coincide with a convenient graticule line. Then you start counting divisions, rounding to tenths of a division, and finish by multiplying by a scale factor. (Try it with the waveform in figure 1a. Do you get about 780 millivolts?) With the 7854, pressing the P-P button yields the peak-to-peak amplitude of the stored waveform. An internal microprocessor takes care of all the division counting and scaling (down to hundredths instead of tenths of a division), then displays the final result with the stored waveform as shown in figure 1b. And the microprocessor does it the same way every time.

That's just a simple example of capabilities. As indicated in figure 2, the 7854 has a variety of push-button routines available for computing most standard waveform parameters. And what's more, these routines can even be combined into program sequences that can be stored and called up to perform multi-stepped analysis tasks.



a. Standard real-time oscilloscope display



b. 7854 display of stored waveform with computed data

Figure 1. Compare the displays in a and b for information provided. The real-time display shows the waveform with scale factors. The stored waveform display includes the same information plus the vertical zero reference (VZR is 0.004 divisions below center) and internally computed waveform data such as peak-to-peak value (777.3 mV shown at bottom of Figure b).

Basics of Waveform Storage

Waveforms displayed on the 7854 Oscilloscope can be stored in a digital memory similar to those used in calculators and computers. However, before a waveform can be stored, it must be converted to a digital format. This is accomplished by digitizing the waveform.

Digitizing, in the simplest sense, is a matter of finding the amplitude value of a waveform at a single point in time. For example, if you display a waveform on an oscilloscope screen and determine its amplitude value at midscreen by counting display divisions and multiplying by the vertical scale factor, you have essentially digitized the waveform at a single point. Similarly, the displayed waveform's vertical values could be determined at every horizontal division. For a 10-division graticule, this would give you 11 points on the waveform(including time zero) which could be entered into a table (stored) for future use or reference.

The 7854 digitizes waveforms essentially along those same lines. The difference is that it does it on a much finer and more accurate scale than is offered by an oscilloscope display graticule.

Digitizing with the 7854 is accomplished electronically over a vertical range corresponding to five divisions above and five divisions below the center of the display. Furthermore, the digitizer can resolve each major vertical division into an average of 102.3 minor divisions or levels. And, in the default mode, this is done at 512 points equally spaced in time, with point zero corresponding to the left side of the oscilloscope display, and point 511 corresponding to the right side. Optionally, digitizing can be reduced to 128 horizontal points per waveform or increased to 1024 horizontal points. This is done by entering the desired number of points and then pressing the f > P/W keys on the Waveform Calculator.

All of the points obtained by digitizing a waveform's vertical values are stored in memory in a tabular format. Also stored along with this "table of values" are the vertical and horizontal scale factors and zero-referenced information associated with that waveform. All of this data is referenced by a waveform address designated as 0 WFM, and waveform digitizing and storage is always directed initially to that address.

Digitizing and storing a waveform is accomplished by pressing one of the store keys, AQR or AVERAGE 100 for two examples. AQR causes a full complement of points (a single copy of the waveform) to be digitized and stored in 0 WFM. The AVERAGE or AVG keys, however, cause multiple copies of the waveform to be taken and an average copy to be computed from these. This process is referred to as signal averaging, and its benefit is that random components (noise, jitter, and other zero-mean phenomena) associated with waveforms tend to be averaged toward zero. In other words, signal averaging repetitive waveforms improves their signal-to-noise ratio.

It should be pointed out that signal averaging requires an additional memory area for working space. Waveform storage address 1 WFM is automatically used for this, and 0 WFM is used for the final results of averaging.

Besides 0 WFM and 1 WFM, there are also a number of other waveform storage areas available. These areas are designated 2 WFM, 3 WF M, 4 WFM, etc. Once a wave form has been stored by AQR or AVG, it should be transferred from 0 WFM to another area if that waveform data will be needed after other waveform acquisitions. Copying a waveform to one of these other areas, 2 WFM for example, is done by first keying in 0 WFM and then keying in 2 f >WFM.

The actual number of waveform storage areas that can be addressed is determined by both the memory option installed and the number of points per waveform. Further information is provided in the accompanying Memory Format Table.

Once a waveform is acquired or copied into a memory address, it can be recalled for display or operation on by the Waveform Calculator functions. This is done by first keying in the address of the waveform (I WFM, 2 WFM, or whatever) and then keying in the function (RMS, MEAN, or whatever) to be performed on the data residing at the selected address.

Memory Format

	Standard				Option 2D				Option 0D
Points Per Waveform*	128	256	512	1024	128	256	512	1024	512
Max. No. of Waveforms	16	8	4	2	40	20	10	5	1
Max. No. of Constant Registers	50				100				0
Max. No. of Prog. Commands plus lines	920				2000				0

* Unless otherwise selected, default value is 512 at power-up.

To explore multi-stepped analysis tasks further, consider the problem of measuring pulse parameters.

Storage is the First Step

Before a pulse or any other waveform can be analyzed, it must be stored or held in position long enough to be examined. Traditionally, synchronous triggering and phosphor retention have served to freeze waveform images on oscilloscope CRTs for observation.

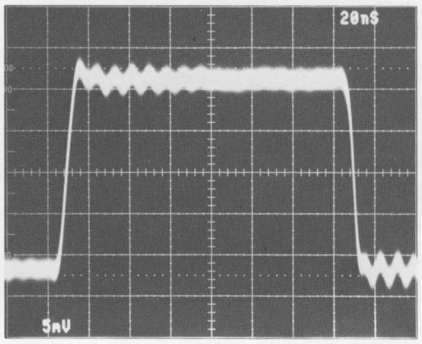
The same tradition is followed with the 7854 when it is operated in the scope display mode. In fact, that is the first step in digital storage with the 7854: put it in SCOPE mode and obtain a traditional display of the waveform to be stored. Just press the SCOPE button on either measurement key pad and operate the 7854 as a normal oscilloscope.

Once a display of the desired waveform has been obtained, press the "f" key (shift function) and then the BOTH key. The

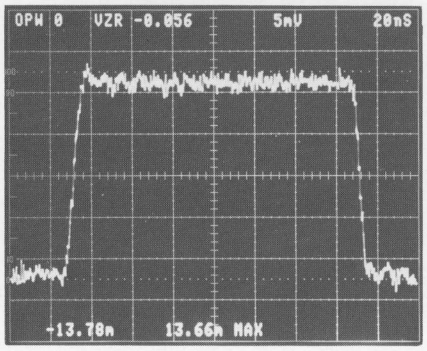
BOTH function allows you to observe both the real-time waveform and the waveform data stored in the OPW (operational waveform) storage area.

At this point, it is possible to press the AQR (acquire repetitive) button to digitize and store the waveform. However, for many measurements (such as the maximum or minimum value of a waveform) it is necessary to have first established a zero-reference for the waveform. In oscilloscope measurements, this is the same as grounding the probe or input, establishing a reference by vertically positioning the ground trace to correspond to one of the display division lines, and then making measurements relative to that zero-reference line. With the 7854, the operation of storing a ground or zero reference is similar. The input is grounded. Then, instead of adjusting trace position, the GND button on either measurement key pad is pressed. Pressing the GND button enters 128 samples of the ground signal and computes their mean value relative to the center horizontal graticule line. Until a new reference is acquired, this computed mean value is established as the ground or zero reference for all waveform data subsequently stored with the AQR button.

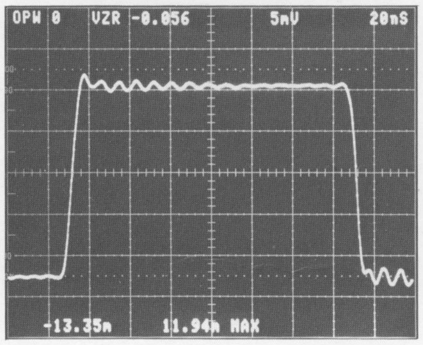
If you've augmented the preceding discussion of storage by looking up button locations on the key pad illustrations in figure 2, you've probably noticed several buttons other than GND and AQR in the waveform acquisition section. For one, there is AQS (acquire single-sweep) which is the shift function of AQR. AQS is a special-purpose operation to sequentially digitize single-shot waveforms and is detailed in the 7854 Oscilloscope Operators Manual, TEKTRONIX part number 070-2873-00. The other buttons (AVERAGE 10, 100, and 1000 on the mainframe and AVG on the Waveform Calculator) are of more general use in acquiring repetitive waveforms. These buttons invoke a signal averaging process that reduces additive noise on waveforms as they are stored. Details on this averaging process and the basic processes



a. Noisy waveform displayed in SCOPE mode



b. Noisy waveform stored by AQR button



c. Noisy waveform stored by AVG 1000

Figure 3. The above CRT photos show three methods of acquiring a noisy waveform. In a, measurements require visual estimation of center trace. In b, storage by the AQR button includes noise components which are included in any measurements (e.g., MIN and MAX at bottom of display). In c, storage by averaging 1000 acquisitions of the signal results in a waveform with greatly reduced noise and measurements occur on just the waveform instead of the waveform plus noise (compare MIN and MAX results in c to those in b).

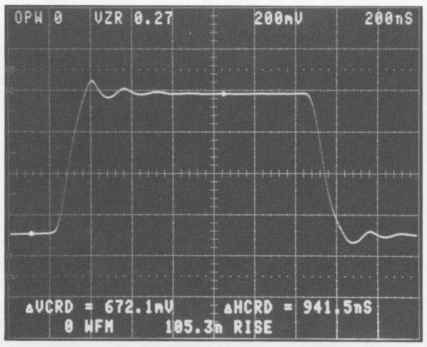
of waveform digitizing and storage are discussed herein under the heading "Basics of Waveform Storage." For now, however, it is sufficient to just be aware of the potential benefit of signal averaging. This benefit is illustrated in figure 3 which shows that signal averaging is often essential to improving measurement results.

Let Your Cursors Do the Counting

The minimum and maximum waveform values displayed at the bottoms of the CRT photos in figures 3b and 3c were obtained by pressing the MIN and MAX buttons on the measurement key pad. The minimum function, when activated by the MIN button, simply searches the array of stored waveform values (the OPW) for the minimum value. Once found, this value is displayed in the X-register position. Pushing the MAX button causes the stored waveform's values (the OPW) to be searched for its maximum value. This maximum value is then displayed at the X-register position, and what was previously stored in the X-register (in this case the minimum) is shifted to the Y-register.

In the same manner, the peak-to-peak value (P-P), root-mean-square value (RMS), average value (MEAN), median or midvalue (MID), etc., of the OPW can be computed and displayed simply by pressing the appropriate button. However, it should be pointed out that these buttons or functions by themselves operate on the totality of OPW values. That is, RMS or MEAN compute strictly from the set of stored waveform values without regard to what portion or shape of signal that collection of values might represent. So if you wish to compute the mean or RMS value of just a selected portion of the displayed waveform (over one cycle of a signal or over a pulse while excluding extraneous baseline data) the button functions must be limited to that desired portion. The 7854 cursors let you do this limiting or selecting. When the cursors are on, the button functions operate only on the waveform data spanned or located between the cursors.

Figure 4 shows a stored pulse with the 7854 cursors turned on and positioned to span the rising portion of the pulse. In this particular photo, the trace intensity has been subdued for better visibility of the cursors, both of which appear as bright dots on the waveform trace (one near the left edge and one near midscreen).



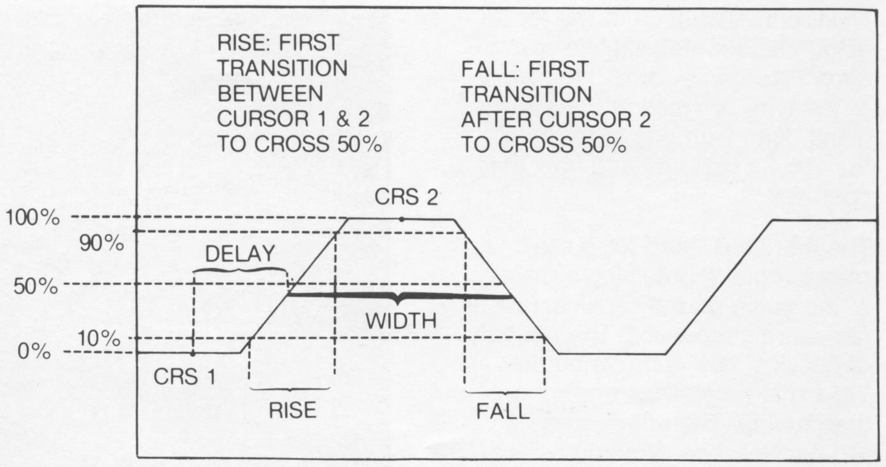
```
OPW 0   VZR 0.27        200mV        200nS
 ΔVCRD = 672.1mV     ΔHCRD = 941.5nS
      0 WFM      105.3n RISE
```

Figure 4. Stored waveform with the cursor dots spanning the rising transition. Cursor displacement is displayed in the first line at the bottom of the CRT. Rise time, bottom center, is computed simply by pressing the RISE button.

Turning the cursors on while viewing a stored waveform is simply a matter of pressing the CRS2-1 button. When the cursors are brought up, their displacement from each other is also computed and displayed at the bottom of the screen. This is illustrated in figure 4 where the cursors are positioned 672.1 millivolts apart vertically (ΔVCRD) on the waveform and 941.5 nanoseconds apart horizontally (ΔHCRD). Moving either cursor left or right along the trace is accomplished with the <CRS1> or <CRS2> buttons. Each cursor movement is reflected by an appropriate change of displacement values in the readout. So, for relative measurements on waveforms like pulse height in figure 4, where you used to have to do a lot of finger pointing and division counting, you can now just let the cursors do the counting for you.

Rise time, Fall Time, Delay, and Width

Besides being set for pulse height (ΔVCRD), the cursors in figure 4 are also set for another standard pulse measurement, rise time. In figure 4, the RISE button has been pressed to compute the 10 to 90 percent rise time of the transition spanned by the cursors. The result, 105.3 nanoseconds, is displayed in the X-register position at the bottom of figure 4 preceding RISE.



RISE: FIRST TRANSITION BETWEEN CURSOR 1 & 2 TO CROSS 50%
FALL: FIRST TRANSITION AFTER CURSOR 2 TO CROSS 50%
CRS 2
100%
90%
DELAY
50%
WIDTH
10%
0%
CRS 1
RISE
FALL

Figure 5. Defining pulse parameters with the cursors: The cursor 1 position defines the pulse base (0-percent level), and cursor 2 defines the pulse top (100 percent level). From these levels, all other levels are computed for use in defining times of rise, fall, delay, and width.

Fall time can also be computed and displayed by pressing the "f" key and then FALL (shift function of RISE). This button sequence causes the 10 to 90 percent fall on the transition to the right of cursor 2 to be computed. Also, without changing the cursor positions, delay and width can be computed by pressing the DELAY and WIDTH buttons. This sequence causes both pulse parameters to be computed along the 50 percent level as indicated in figure 5. As each value is computed by pressing a button, the result appears in the X-register display (causing the previously computed result to be shifted to the Y-register display).

Programming Your Measurements

Suppose you need to make the same measurements over and over on a large number of waveforms. The push-button sequences discussed thus far can reduce time requirement and increase measurement accuracy. After all, positioning cursors and pressing the RISE button is much faster than the old method of counting divisions.

But, for even more speed in completing a measurement sequence, the Waveform Calculator button operations can also be entered into 7854 memory as a program sequence. As an example of how this is done and how it can further speed and simplify a measurement process, consider some of the details of writing a simple program for pulse measurement and analysis.

In preparing to write any program, the first step is to define the major operations to be performed. For example, to use the 7854 to measure pulse rise time, fall time, delay, and width, the following steps must be completed:

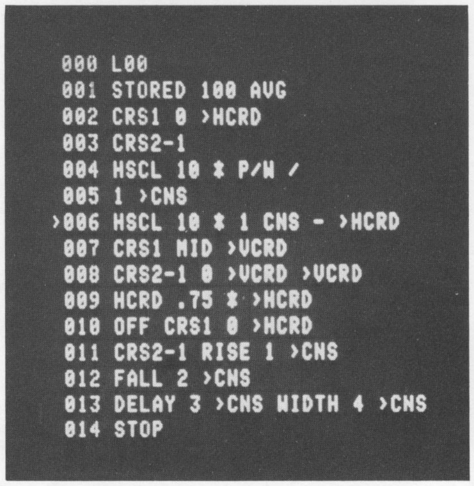
1. obtain a display of the pulse on the oscilloscope
2. digitize and store the waveform
3. position cursor 1 on the pulse base (0-percent level) preceding the first transition
4. position cursor 2 on the pulse top (100-percent level) after the first transition
5. initiate the RISE, FALL, DELAY, and WIDTH functions

Next, consider each of the foregoing steps in detail. How can each step be performed in terms of the type or types of waveforms being dealt with and in terms of the 7854's features and how they operate?

The first step listed for pulse measurement, obtaining a display of the pulse on the oscilloscope, is fairly straightforward. The majority of oscilloscope controls on the 7854 and its plug-ins are not programmable. Therefore, standard oscilloscope procedures are used to obtain a reasonable real-time display of the pulse to be measured.

The next step, digitizing and storing the waveform, offers several choices. One choice is to push the sequence of buttons discussed previously for storage. Or, with the exception of probe grounding for zero reference acquisition, the operation can be programmed. Or, if you don't need a zero or ground reference, the whole operation of digitizing and storage can be done under program control. This latter case is a valid one for pulse measurement analysis since a zero reference is not necessary to defining the relative levels used in computing rise time, fall time, delay, and width. So, digitizing and storage of a pulse can be completely programmed. This is done in line 001 listed in figure 6a, which is a full listing of a pulse analysis program.

For those not familiar with the Reverse Polish Notation (RPN) of stack-oriented calculators, the keystroke sequences for the various calculations listed in figure 6 may be confusing. To minimize this confusion, read the information under the heading "Introduction to RPN" before attempting to interpret the program listing shown in figure 6a.



```
 000 L00
 001 STORED 100 AVG
 002 CRS1 0 >HCRD
 003 CRS2-1
 004 HSCL 10 * P/W /
 005 1 >CNS
>006 HSCL 10 * 1 CNS - >HCRD
 007 CRS1 MID >VCRD
 008 CRS2-1 0 >VCRD >VCRD
 009 HCRD .75 * >HCRD
 010 OFF CRS1 0 >HCRD
 011 CRS2-1 RISE 1 >CNS
 012 FALL 2 >CNS
 013 DELAY 3 >CNS WIDTH 4 >CNS
 014 STOP
```

a. Program listing

LISTING LINE NO.	PROGRAM ENTRY KEYSTROKE SEQUENCES
000	f LNN 0 0 NEXT
001	STORED 1 0 0 AVG NEXT
002	f CRS1 0 f >HCRD NEXT
003	CRS2-1 NEXT
004	HSCL 1 0 * P/W / NEXT
005	1 f >CNS NEXT
006	HSCL 1 0 * 1 CNS – f >HCRD NEXT
007	f CRS1 MID f >VCRD NEXT
008	CRS2-1 0 f >VCRD f >VCRD NEXT
009	HCRD • 7 5 * f >HCRD NEXT
010	OFF f CRS1 0 f >HCRD NEXT
011	CRS2-1 RISE 1 f >CNS NEXT
012	f FALL 2 f >CNS NEXT
013	DELAY 3 f >CNS WIDTH 4 f >CNS NEXT
014	STOP NEXT

b. Keystroke sequence for entering the program

Figure 6. 7854 program to automate program analysis

The program listing shown in figure 6a was entered by first putting the 7854 into the program entry mode. Program entry is accomplished with the front-panel PROGRAM ENTRY/EXECUTE button (top left of the Waveform Calculator). This button switches the system from one mode to the other each time it is pressed.

With the 7854 in the program entry mode, the keystroke sequences shown in figure 6b are those necessary to enter the pulse analysis program. The first sequence, corresponding to line number 000, is simply for program labeling. Its entry is terminated by pressing the NEXT button, which automatically assigns a line number in sequence and prepares the system for entry of the next line.

Introduction to RPN

RPN (Reverse Polish Notation) is the arithmetic system used in the 7854 Oscilloscope. This is a system common to stack-oriented calculators, and it is distinguished by a protocol that requires entry of operands before specification of the function to be performed.

As a simple demonstration of RPN, consider adding two numbers, 57 and 24 for example. Pressing 5 and 7 on the numeric keypad causes 57 to be entered into the X-register of the stack. Entry of the number is then terminated by pressing the ENTER key. Next, 24 is keyed in, causing 57 to be pushed from the X-register into the Y-register. Finally, the + function is pressed. This causes the contents of the Y- and X-registers to be shifted off of the stack (to the right in the accompanying figure) into the calculator area; the result of the calculation is then pushed back onto the stack (to the left) into the X-register.

Notice in the above sequence that the ENTER key was not used after entering the number 24. The + function, or any other function entry (–,*,SQRT, etc.), performs both the operation of terminating current numeric entry as well as the mathematical operation specified.

Thus, it is only necessary to use the ENTER key for signaling completion of a numeric entry that is going to be followed immediately by another numeric entry, as in the case of entering 57 and then 24.

REGISTER STACK CONTENTS

W	T	Z	Y	X	KEYSTROKES
0WFM	14	–.97	1.8	3.145	STACK CONTENTS FROM PREVIOUS OPERATIONS
14	–.97	1.8	3.145	57	5 7 ENTER
–.97	1.8	3.145	57	24	2 4
0	–.97	1.8	3.145	81	+

In line 001, the line for storing the pulse, the first keystroke is to put the 7854 into the stored display mode so that the results of waveform storage can be viewed. The next keystrokes cause entry of the command to acquire the waveform by signal averaging it 100 times. For more or less signal averaging, just key in the desired number instead of the 100 indicated in the listing of figure 6a.

The final keystroke to enter line 001 is the NEXT required to terminate the line. Notice that NEXT is simply a program entry operation and therefore does not appear in the final program listing.

The next major operation after digitizing and storage is to position cursor 1 on the pulse base (0-percent level) preceding the first transition. This is quite easy to do when it is stipulated that the pulse must be acquired so that the left edge (time zero) of the stored trace will be the pulse base (see figure 7). Given this stipulation, the keystrokes in line 002 will cause cursor 1 to be turned on and moved on the trace to the horizontal location representing time zero. Specifically, "f CRS1" causes cursor 1 to be turned on, "0" causes the value zero to be entered into the X-register, and "f >HCRD" causes the cursor to be moved along the trace to the horizontal time location currently specified in the X-register.

The next operation, positioning cursor 2 on the 100-percent level of the pulse, is somewhat more involved. However, it is instructive to go through the sequence both from the standpoint of understanding the pulse analysis routine and from the standpoint of understanding how the cursors can be manipulated individually or together for any other kind of analysis. As an aid to this understanding, table 1 contains some general rules governing cursor positioning by 7854 commands.

Since a less than ideal pulse shape is often the case in pulse analysis, the example pulse illustrated in figure 7 includes overshoot and ringing. This condition means that the maximum value of the pulse generally will not correspond to the 100-percent value normally used to define rise and fall times.

To aid in determining where the pulse top is, it is assumed that any ringing will be fully settled by the time the pulse reaches 75-percent of its duration. As illustrated in figure 7, this location, indicated there as "top," is the point that cursor 2 needs to be positioned to under program control. Program lines 002 through 010 cause this cursor positioning to occur.

Besides being used to place cursor 1 at the assumed 0-percent level, line 002 also serves to place cursor 1 into a known or home position. For the purposes of the program, that home position is the left edge of the stored trace. The next step turns on cursor 2 (line 003). Cursor 2 is then put into its known or home position, the right edge of the stored trace.

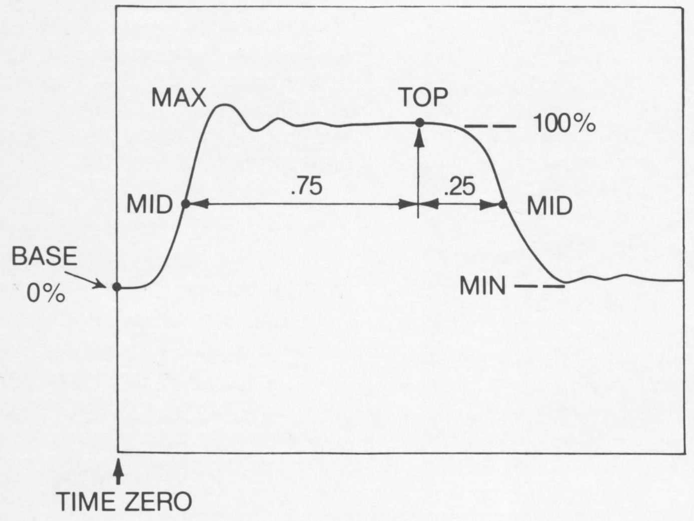


Figure 7. Defining locations on the acquired pulse for automatic cursor positioning with a 7854 keystroke routine

Table I

Rules for Cursor Positioning by >HCRD and >VCRD

* Cursors can only be positioned on the stored trace (OPW); attempts at off-trace positioning result in error conditions.
* >HCRD causes horizontal positioning according to the value stored in the X-register.
* >VCRD causes vertical positioning according to the value stored in the X-register.

Cursor 1 On (f CRS1)

* Positioning by >HCRD is relative to the right side of the stored trace (time zero).
* Positioning by >VCRD is determined by a search to the right of the current cursor position for a vertical value equal to that in the X-register; if the value isn't found by the time the right edge is reached, the search continues from the left edge of the display up to the original cursor position.
* All waveform operations (e.g., peak-to-peak, min, max, etc.) are confined to stored trace values to the right of and including the cursor 1 position.

Cursor 1 and 2 On (CRS2-1)

* >HCRD causes cursor 2 to be positioned relative to cursor 1 by the amount stored in the X-register.
* Positioning by >VCRD is determined by a search to the right of the current cursor 2 position for a vertical displacement from cursor 1 that equals the value stored in the X-register; if the displacement isn't found by the time the right edge of the trace is reached, the search continues from the cursor 1 position back to the original cursor 2 position.
* All waveform operations (e.g., peak-to-peak, min, max, etc.) are confined to the stored trace values spanned by the cursor positions.

In terms of time data points, the right edge of the stored trace is ten times the horizontal scale factor. But since cursor 2 moves relative to cursor 1, which has already been placed on the first point (time zero), the differential time needed for cursor 2 positioning is ten times the horizontal scale factor less one time increment. Actual computation of this differential time begins in line 004 where the horizontal time per waveform point is computed. The result is then stored in 1 CNS (line 005) and then used in line 006 to compute the differential time necessary to position cursor 2 on the last point of the stored trace. The last command in line 006 causes this cursor 2 positioning to occur.

At this stage, both cursors have been established at home positions. Cursor 1 is on the left-most point of the stored trace, and cursor 2 is on the right-most point.

Line 007 begins positioning of the cursors to span the first pulse transition. This operation is started by calling up cursor 1, computing the middle vertical value of the pulse, and then positioning cursor 1 to that point. This sequence puts cursor 1 generally near the center of the pulse's first transition.

The next line, 008, positions cursor 2 at the same vertical level on the second transition. The operation is done by first calling up cursor 2, entering a value of zero into the X-register, and then causing a double search for that differential vertical value. Since cursor 2 was previously positioned at the rightmost point on the stored trace, the first search begins at the cursor 1 position. This is the first incidence of zero vertical differential, and cursor 2 is moved to overlap the cursor 1 position. The next search is to the right of that and finds the next incidence of zero differential. Assuming that pulse ringing does not equal or exceed the mid vertical value, the second occurrence of zero differential is on the second transition. Thus, the second search places cursor 2 on the second transition at a level equal to the level of cursor 1. The resulting horizontal difference between cursors (HCRD) provides a reasonable estimate of pulse width and is used in line 009 to place cursor 2 on the pulse top.

With cursor 2 in position, the cursors can be turned off and cursor 1 brought up again by itself for positioning on the assumed pulse base at time zero. This is done in line 010. Then, with both cursors finally positioned to span the transition, they can be brought up together to compute pulse parameters. Rise time, fall time, delay, and width are computed in lines 011-013 and the values stored in constant storage locations 1 CNS through 4 CNS.

When the above program has been keyed in, the 7854 is returned to the execute mode. This is done by pressing the PROGRAM ENTRY/EXECUTE button on the Waveform Calculator keyboard. When the 7854 display changes from program lines to a waveform display, the instrument is in the execute mode.

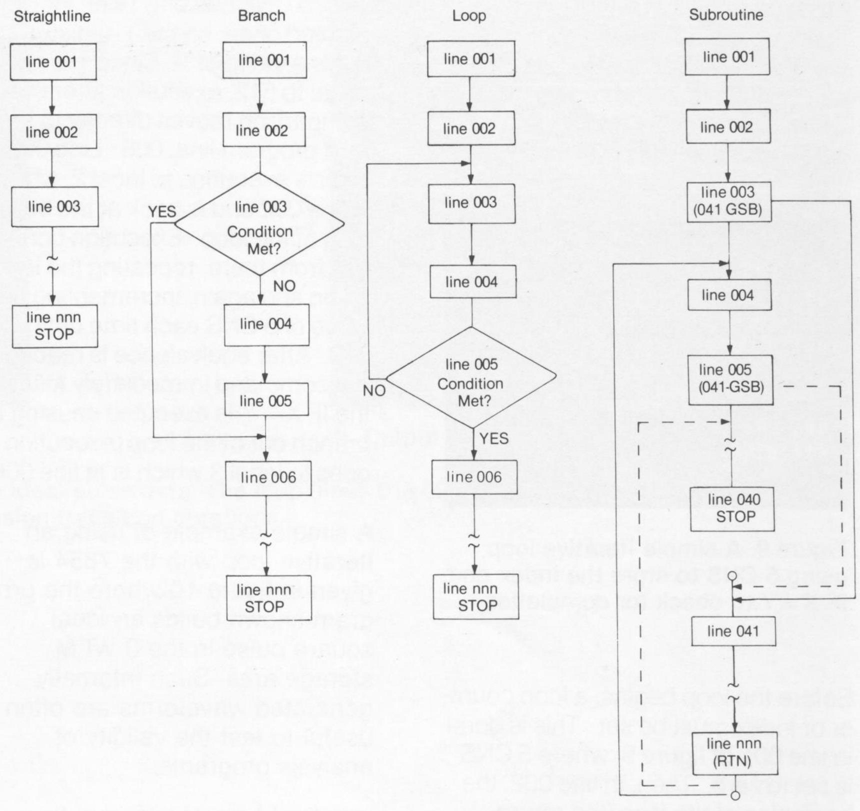


Figure 8. Program flow can be straightline or altered conditionally or unconditionally by several programming devices including branches, loops, and subroutines.

To use the pulse analysis program, simply press SCOPE and obtain a display of the pulse you want to analyze. Then press f START on the Waveform Calculator. This initiates program execution causing the displayed pulse to be stored, the cursors to be placed, and the four pulse parameters to be computed and stored, all without pressing any additional buttons. To display any of the results in the X-register area, simply press the constant number and the CNS button. For example, to display fall time in the X-register area, press 2 CNS.

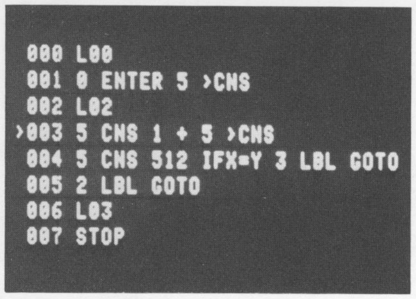
To analyze another pulse with the program, return the 7854 to the oscilloscope display mode by pressing SCOPE. Then adjust the 7854 for a display of the new pulse and start the program again by pressing f START.

Other Programming Techniques

The pulse analysis program used in the example follows a form which can be referred to as a straightline program. In other words, the program executes one line after the other in numeric order. However, other types of program flow are often necessary to solve more complex problems. Some flow types are shown in figure 8. In sophisticated applications, it is not unusual for all of these to occur one or more times in a single program.

The 7854 Oscilloscope with the Waveform Calculator provides capabilities for all of the flow types illustrated in figure 8. For branching or looping, the conditional test can be done with either the IF Y > X or IF X = Y functions. These two functions test the contents of the X and Y registers for the indicated condition (> or =). If the condition is met, program execution goes to the command immediately following the IF. If not, execution drops to the next program line.

As an example of usage, consider constructing an iterative loop with the IF X = Y function. Such a loop for counting to 512 is shown in figure 9, where the looping or iterations span lines 002 through 005.



```
 000 L00
 001 0 ENTER 5 >CNS
 002 L02
>003 5 CNS 1 + 5 >CNS
 004 5 CNS 512 IFX=Y 3 LBL GOTO
 005 2 LBL GOTO
 006 L03
 007 STOP
```

Figure 9. A simple iterative loop using 5 CNS to store the index and IF X = Y to check for completion

Before the loop begins, a loop counter or index must be set. This is done in line 001 of figure 9, where 5 CNS is set to zero. Then, in line 002, the beginning of the loop (the return point) is labeled as L02 (press f LNN 02). The next line, 03, increments the index (5 CNS) by 1, and the following line checks to see if the count has reached 512 yet.

The check is performed by first pushing the value of 5 CNS onto the stack. This is followed by the loop terminating value, 512. The value of 5 CNS is now in the Y-register, and 512 is in the X-register. IF X = Y compares the register contents for equivalence and directs the next program move accordingly. On the first execution of the loop, 5 CNS has only been incremented once, so the Y-register holds a value of 1. Since 1 is not equal to 512, execution after the comparison moves directly to the next program line, 005. Line 005 directs execution to label 2, which is line 002 and is back at the beginning of the loop. Execution continues from there, repeating the loop again and again, incrementing the value of 5 CNS each time until it is 512. After equivalence is reached, the command immediately following the IF X = Y is executed causing a branch out of the loop (execution goes to label 3 which is at line 006).

A simple example of using an iterative loop with the 7854 is given in figure 10 where the program shown builds an ideal square pulse in the 0 WFM storage area. Such internally generated waveforms are often useful to test the validity of analysis programs.

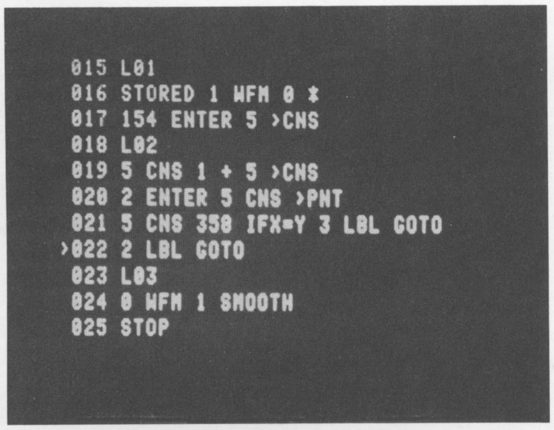
The final type of program flow shown in figure 8 is that using a subroutine. A subroutine is extremely useful where the same calculation must be used at several different places in a program. Instead of keying in the same operation at each place of need, it can be entered once as a subroutine. This method saves keystrokes during program entry as well as conserves program storage space.

Typically, subroutines are located after the body of the main program, as illustrated in figure 8. In all cases, each subroutine must be terminated with an RTN (return) function.

Each time the operations of a subroutine are needed by the main program, they can be called by a GSB (go to subroutine) command. The GSB causes execution to jump from the main program to the first subroutine line. The subroutine executes, and then the RTN at the end of the subroutine causes a jump back to the main program line immediately following the GSB.

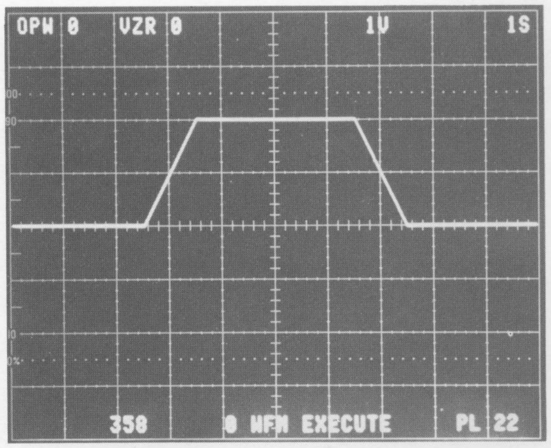
Interfaced for Extended Capabilities

The Waveform Calculator features of the 7854 Oscilloscope literally reduce most common waveform measurements to a matter of simply pressing a few buttons. And some features — like RMS and MEAN — add measurement capabilities not usually associated with oscilloscopes. The capability of storing key-strokes as a program lets you reduce many extensive measurement sequences to a matter of simply pressing a few buttons to get the program started. It's like stepping up from a slide rule to a sophisticated calculator...plus more.



```
 015 L01
 016 STORED 1 WFM 0 *
 017 154 ENTER 5 >CNS
 018 L02
 019 5 CNS 1 + 5 >CNS
 020 2 ENTER 5 CNS >PNT
 021 5 CNS 350 IFX=Y 3 LBL GOTO
>022 2 LBL GOTO
 023 L03
 024 0 WFM 1 SMOOTH
 025 STOP
```

a. Pulse generation routine



b. Output of the routine

Figure 10. Using an iterative loop to generate ideal pulse data. The loop (lines 018-022) generates the square pulse, and the 1 SMOOTH (line 024) gives it one-division transition durations.

Besides its internal computational and programming features, the 7854 Oscilloscope also offers GPIB* compatibility, with a GPIB interface included as standard equipment. This feature makes it ready to connect to any GPIB compatible controller (desk-top calculator or minicomputer). Plus, the 7854 contains enhancements to ease transfer of digitized waveforms over the GPIB. This means you can quickly transform your 7854 Oscilloscope from a bench-top analysis tool to a front-end processing unit in a larger waveform analysis system. When that happens, the extent of your waveform measurement and analysis capabilities becomes almost limitless.

*GPIB is the acronym for the General Purpose Interface Bus which has been implemented in the 7854 in accordance with IEEE Standard 488-1978.

For further information, contact:

U.S.A., Asia, Australia, Central & South America, Japan
Tektronix, Inc.
P.O. Box 4828
Portland, OR 97208
Phone: 800/547-6711
Oregon only 800/452-6773
Telex: 910-467-8708
Cable: TEKTRONIX

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

Canada
Tektronix Canada Inc.
P.O. Box 6500
Barrie, Ontario L4M 4V3
Phone: 705/737-2700

Tektronix sales and service offices 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

4/81 42AX-4682