Errata

**Document Title:** 5370B Universal Time Interval Counter Product Note 5370B-3

**Part Number:** 5952-7769

**Revision Date:** January 1985

---

**HP References in this Application Note**

This application note may contain references to HP or Hewlett-Packard. Please note that Hewlett-Packard's former test and measurement, semiconductor products and chemical analysis businesses are now part of Agilent Technologies. We have made no changes to this application note copy. The HP XXXX referred to in this document is now the Agilent XXXX. For example, model number HP8648A is now model number Agilent 8648A.

**About this Application Note**

We’ve added this application note to the Agilent website in an effort to help you support your product. This manual provides the best information we could find. It may be incomplete or contain dated information, and the scan quality may not be ideal. If we find a better copy in the future, we will add it to the Agilent website.

**Support for Your Product**

Agilent no longer sells or supports this product. You will find any other available product information on the Agilent website:

[www.agilent.com](http://www.agilent.com)

Search for the model number of this product, and the resulting product page will guide you to any available information. Our service centers may be able to perform calibration if no repair parts are needed, but no other support from Agilent is available.
High Throughput Picosecond Characterization of Pulse Parameters

5370B UNIVERSAL TIME INTERVAL COUNTER

Product Note 5370B-3
HEWLETT PACKARD
Contents

Introduction .................................................. 2
Applications .................................................. 3
The effects of instrument bandwidth on rise and fall
time measurements ........................................ 6
Counter or oscilloscope to characterize pulse parameters ............... 7
A sample program ............................................ 8
Operation .................................................... 24
Explanation of softkeys ...................................... 25
Miscellaneous ............................................... 26
Conclusion ................................................... 27
Introduction

A counter is particularly effective either when single shot events are measured, or when the time required to make a measurement is critical—such as in the production environment. This product note will show the use of a computer-controlled time interval counter to characterize pulse parameters with picosecond resolution. In addition, some powerful statistical routines will be presented to further enhance the characterization.

Obviously with this capability, the statistical nature of pulse parameters can be quickly analyzed. For example, drift of rise time with time or some external influence—such as temperature or voltage—can be rapidly characterized. Similarly, pulse width jitter statistics can be analyzed in seconds.

The 5370B Counter

The counter used here is the 5370B Time Interval Counter. This counter can measure single events such as rise time, fall time and pulse width to a typical resolution of 20 picoseconds rms, and can make successive measurements with only 165 μseconds delay between each measurement (approximately 6000 readings per second). (See 5370B data sheet for more details.)

The 5363B Probes

Used in these sample programs are the 5363B Time Interval Probes. Often necessary in time interval measurements, these probes provide high impedance, low capacitance connections to the device under test, thus minimizing loading effects. The probes also extend the voltage range over which trigger levels can be set with precision. (See 5363B data sheet for more details.)

The Series 200 Computer

The series 200 computers are powerful machines for scientific and engineering applications and are well suited for instrument control activities.

Many programming languages can be used. In this product note, the program is written in BASIC. For time critical functions, (such as the bin sorting for the histogram routine), some parts can be written in Pascal, subsequently compiled and then called by the BASIC program as CSUB's. (For more details refer to the appropriate Series 200 computer documentation.)

The program listed here will run on a 9816A, 9826A or 9836A with at least 250 kilobytes of program memory, BASIC 2.0 and Extensions 2.1. Graph2 _1 and the 9836C allow the graphs to be plotted in color.
Applications

There are many application areas where the following techniques can be extremely valuable. Examples are:

- Integrated Circuit Test System Timing Calibration
- Integrated Circuit Characterization
- Magnetic Disc Drive and Media Testing
- Digital Communications Systems Timing Analysis
- Pulse Generator Characterization

The predominant requirements are:

- The need to measure times with picosecond resolution
- The need to measure single shot or infrequently repetitive events
- The need for high measurement throughput such as in the production test environment
- The need to analyze the statistics of timing measurements
Disc Testing

By measuring the timing relationship between the read clock and the read data transition for many data bits, an estimate of the read error rate can be made. (See figure 1). The results of this margin test can be best observed in a histogram format. (See figure 2)

![Figure 1.](image1)

![Figure 2.](image2)

Digital Communications

The timing jitter accumulated over a communications link can be quickly assessed by measuring the bit-to-bit timing or the clock-to-bit edge timing. (See figure 3).

![Figure 3.](image3)

IC Characterization

The drift in, for example, rise time with time or temperature can be rapidly ascertained. By taking many rise time measurements, (as many as 6000 per second), over a period of time – perhaps as some external parameter is varied – a plot such as that illustrated in figure 4 can point out picosecond variations in rise time.

![Figure 4.](image4)
Pulse Generator Characterization

Pulse width jitter, a key specification in precision pulse generators, can be quickly quantified and the statistics visualized as shown in figure 5.

![Figure 5.]

Pulse Characterization

The traditional pulse parameters measured are illustrated in figure 6. Rise time is usually measured either from the 10% to 90% of peak-to-peak voltage levels or from 20% to 80% levels, as is fall time. Pulse width is generally measured at 50% of peak-to-peak voltage level.

![Figure 6.]

---

5
The Effects of Instrument Bandwidth on Rise and Fall Time Measurements

As the edge speeds of a waveform approach the bandwidth limitations of the measuring instrument, distortion occurs. The general case is shown in figure 7.

![Diagram showing undistorted waveform and distortion caused by bandwidth limitation.]

Figure 7.

To measure a signal's rise time accurately, the measurement system should have a rise time at least three times faster than that of the signal. Table 1 illustrates the effects of measuring a 2 nanosecond rise time signal with a range of instrument bandwidths. Similarly, table 2 shows the effects on a 5 nanosecond edge.

<table>
<thead>
<tr>
<th>Measuring Instrument Bandwidth / Rise Time</th>
<th>Measured Rise Time</th>
<th>Percent Error from 2 ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>250 MHz / 1.75 ns</td>
<td>2.66 ns</td>
<td>33%</td>
</tr>
<tr>
<td>350 MHz / 1.00 ns</td>
<td>2.24 ns</td>
<td>12%</td>
</tr>
<tr>
<td>500 MHz / 0.70 ns</td>
<td>2.12 ns</td>
<td>6%</td>
</tr>
<tr>
<td>1 GHz / 0.35 ns</td>
<td>2.03 ns</td>
<td>2%</td>
</tr>
</tbody>
</table>

Table 1. Bandwidth Limitation Errors Measuring a 2 ns Rise Time.

<table>
<thead>
<tr>
<th>Measuring Instrument Bandwidth / Rise Time</th>
<th>Measured Rise Time</th>
<th>Percent Error from 5 ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>250 MHz / 1.75 ns</td>
<td>5.30 ns</td>
<td>6.0%</td>
</tr>
<tr>
<td>350 MHz / 1.00 ns</td>
<td>5.10 ns</td>
<td>2.0%</td>
</tr>
<tr>
<td>500 MHz / 0.70 ns</td>
<td>5.05 ns</td>
<td>1.0%</td>
</tr>
<tr>
<td>1 GHz / 0.35 ns</td>
<td>5.01 ns</td>
<td>0.2%</td>
</tr>
</tbody>
</table>

Table 2. Bandwidth Limitation Errors Measuring a 5 ns Rise Time.
Counter or Oscilloscope to Characterize Pulse Parameters?

Which device to choose to characterize rise/fall time and pulse width is by no means obvious (assuming sufficient bandwidth is available). (Refer to tables 1 and 2.) There are several areas in which a counter makes a more effective analysis tool, and similarly others in which an oscilloscope is best. To help clarify, table 3 illustrates how choices can be made.

<table>
<thead>
<tr>
<th>Counter works best when:</th>
<th>Oscilloscopes generally need repetitive events</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Events are non repetitive or infrequent</td>
<td>(Because of their one shot nature, counters can make several thousand measurements per second)</td>
</tr>
<tr>
<td>• Measurement speed is important</td>
<td>(Once parameters are known, a counter can be set up to measure manually or automatically very simply)</td>
</tr>
<tr>
<td>• Operators have limited training</td>
<td>(Oscilloscope resolution is not a function of time measured)</td>
</tr>
<tr>
<td>• High accuracy and resolution is required for long time intervals</td>
<td></td>
</tr>
</tbody>
</table>

| Oscilloscope works best when:                                 | (Oscilloscopes allow the operator to integrate the “pattern” observed into data) |
| • Visualization of the waveform is important                  | (It is tricky to ascertain the 20%, 50% and 80% points on a waveform with a counter) |
| • Peak amplitude of the waveform is unknown                   | (Oscilloscopes are inherently more useful for voltage measurements) |
| • Voltage information must be gathered                       |                                               |

Table 3.

The rest of the time there is usually free choice between either device.
A Sample Program

Listed below is a sample program which uses many of the intrinsic capabilities of a Precision Time Interval Counter in characterizing pulse width, rise time and fall time.

The Equipment

The equipment configuration is shown in figure 8. It consists of a 5370B Time Interval Counter, 5363B Time Interval Probes, and a Series 200 computer which controls the measurement set-up and processes the results.

![Diagram of equipment configuration](image)

Figure 8.

The Program

The program will perform three major functions, demonstrating:

- Rapid measurements with fast data transfer
- Time vs elapsed time plot
- Histogram sort and plot

The code is listed with explanations and options, and is followed by operating instructions. The program is written in BASIC for clarity, but will run faster if the time critical sections are written in Pascal, compiled, and subsequently called as CSUB's.
PROGRAM "PULSTAT"  02/20/83  -- REV C. --

PROGRAM TO AUTOMATICALLY MEASURE AND CHARACTERIZE RISE/FALL TIMES
AND PULSE WIDTHS OF A SERIES OF DIGITAL PULSES.

SYSTEM REQUIRES ONE EACH OF THE FOLLOWING:  5370B, 5363B.

**80**

**VARIABLES**

**100** OPTION BASE 1

**110** DIM Counter$(50), Probe$(50), Generator$(50), Title$(50), Title1$(50), Title2$(50), Meas_id$(50)

**120** DIM Stat_min$(20), Stat_max$(20), Stat_mean$(20), Stat_dev$(20), X_values$(20)

**130** COM /Stats/ REAL Stat_min, Stat_max, Stat_mean, Stat_dev, Meas_time

**140** COM /Sample/ INTEGER Max_samples, Dnum_samples, Dresolution, Meas_duration

**150** INTEGER Counter, Prtr, Counter_addr, Probe_addr, Generator_addr, Plotter_addr, Disp_length

**160** INTEGER Resolution, Num_samples, Num_bins, Bin_num, X_pos, X_pos_limit, X_maj, Y_maj

**170** INTEGER Pw_auto_res, Pw_auto_samp, Rt_auto_res, Rt_auto_samp, Ft_auto_samp, Auto_sample/resolution values

**180** INTEGER False, True, T, F, Plot_yes, Plot_flag, T_plot_flag, Time_flag, flags

**190**

**200** ! **INITIAL**

**210** Counter = 1

**220** Prtr = 701

**230** False = 0

**240** True = 1

**250** F = False

**260** T = True

**270** Auto = T

**280** Single = T

**290** Contin = F

**300** Time_flag = F

**310** T_plot_flag = T

**320** Plot_yes = F

**330** Plot_flag = F

**340** Max_samples = 0

**350** Maxx = 3

**360** Minx = Maxx

**370** Step = 1

**380** PRINTER IS CRT

**390** Disp_length = 50  ! Allows the 9826A to be used - it has a 50 character display width

**400** Plotter_addr = 705

**410** Counter_addr = 703

**420** Probe_addr = 707

**430** Resolution = 10

**440** Dresolution = 1

**450** Resolution = Dresolution

**460** Stat_min = 0

**470** Stat_max = 0

**480** Stat_mean = 0

**490** Stat_dev = 0

**500** Stat_min$ = "MIN = "

**510** Stat_max$ = "MAX = "

**520** Stat_mean$ = "MEAN = "

**530** Stat_dev$ = "STD, DEV. = "

**540** Meas_id$ = "Put your measurement routine title here"

**550** X_gdu_max = 100 * MAX(1, Ratio)

**560** T_gdu_max = 100 * MAX(1, Ratio)

**570** Plot_left = 20 * X_gdu_max
580 Plot_right=.80*Y_gdu_max
590 Plot_bottom=.35*Y_gdu_max
600 Plot_top=.95*Y_gdu_max
610 !
620 ! ********** MAIN **********
630 !
640 Start: ! START OF PROGRAM
650 GINIT
660 GCLEAR
670 PRINT USING "@"
680 GOSUB Init_program
690 Num_arrays=5+3*8
700 Max_samples=MIN(ROUND(INT((VAL(SYSTEMS("AVAILABLE MEMORY")>>450D)/Num_arrays)),2),65530/5)
710 Dnum_samples=MIN(Max_samples,1000) !DEFAULT SAMPLE SIZE
720 Num_samples=Dnum_samples !INITIAL SAMPLE SIZE
730 GRAPHICS ON
740 PRINT USING "@"
750 ON KEY 0 LABEL "Pulse Width",13 GOSUB P_width
760 ON KEY 1 LABEL "Rise Time",13 GOSUB R_time
770 ON KEY 2 LABEL "Fall Time",13 GOSUB F_time
780 ON KEY 3 LABEL "Time Plot ON",14 GOSUB T_plot
790 IF Plot_yes THEN ON KEY 4 LABEL "Ext Plot OFF",14 GOSUB E_plot
800 ON KEY 5 LABEL "Chg Resolution",13 GOSUB Fix_resolution
810 ON KEY 6 LABEL "Chg # Samples",13 GOSUB Fix_samples
820 ON KEY 7 LABEL "DEMO ON",13 GOSUB Auto_demo
830 ON KEY 8 LABEL "Single ON",14 GOSUB Single_cont
840 ON KEY 9 LABEL "Quit",14 GOTO Quit
850 PRINT "Disp. Res.="; Resolution;"ps","Sample Size=";Num_samples
860 GINIT
870 GCLEAR
880 !
890 Wait_loop: DISP "Choose a key"
900 GOTO Wait_loop !LOOP HERE BETWEEN MEASUREMENTS
910 !
920 !
930 !
940 ! ********** EXEC MEASUREMENT SUBROUTINES **********
950 !
960 P_width: !DO HISTOGRAM OF PULSE WIDTH
970 Title$="Pulse Width"
980 PRINT Title$
990 Probe$="PGA+250UAR+250DS" !This is where trigger levels are set on the 5363B
1000 IF Auto THEN Resolution=Pw_auto_res
1010 IF Auto THEN Num_samples=Pw_auto_samp
1020 GOTO Do_it
1030 !
1040 !
1050 R_time:!DO HISTOGRAM OF PULSE RISE TIME
1060 Title$="Rise Time"
1070 Title$="Rise Time"
1080 PRINT Title$
1090 Probe$="PGA+50UAR+450US" !This is where trigger levels are set on the 5363B
1100 IF Auto THEN Resolution=Rt_auto_res
1110 IF Auto THEN Num_samples=Rf_auto_samp
1120 GOTO Do_it
1130 !
1140 F_time:!DO HISTOGRAM OF PULSE FALL TIME
1150 Title$="Fall Time"
1160
Title$="Fall Time"
PRINT Title$
Probes="PA+450DRA+050DS" ! This is where trigger levels are set on the 5363B
IF Auto THEN Resolution=Fi_auto_res
IF Auto THEN Num_samples=Fi_auto_samp
GOTO Do_it
!
Do_it: !COMPLETE THE MEASUREMENT
PRINT USING "@"
DISP "Setting up measurement ..."
Real_samples=Num_samples
Real_samples=Real_samples*5
Buffer_length=MIN(Real_samples,65500)
ALLOCATE Tivalue(Num_samples)!1i_data$ CONVERTED TO REAL VALUES
GOSUB Make_measure
IF NOT _plot_flag THEN Sort
CINIT
GOSUB Time_plot
IF NOT Plot_flag THEN Sort
GOSUB Ext_plot
GOSUB Time_plot
GOSUB Int_plot
Sort:
GOSUB Sort_data
GOSUB Get_stats
Num_bins=((Stat_max-Stat_min)*(1,E+12)/Resolution)+1
ALLOCATE INTEGER Hist(Num_bins)!DIMENSION LARGEST POSSIBLE Hist(*)&
ALLOCATE Hist_value(Num_bins)!VALUES ASSIGNED TO EACH BIN
GOSUB Compute_hist
CINIT
GOSUB Graph_it
IF NOT Plot_flag THEN Deal1
GOSUB Ext_plot
GOSUB Graph_it
GOSUB Int_plot
Deal1: DEALLOCATE Hist_value(&)
DEALLOCATE Hist(&)
DEALLOCATE Tivalue(&)
IF Auto THEN Wait
IF Contin THEN Do_it
INPUT "Do another measurement? (y/n)",Answer$
IF UPCS(Answer$[1,1])="Y" THEN Do_it
GOTO Finished
Wait: IF Contin THEN Finished
DISP "Hit CONTINUE to proceed."
PAUSE
Finished: RETURN
!
!******** DATA GATHERING AND DISPLAY SUBROUTINES CALLED BY EXECS ********
!
Make_measure: !MAKE Num_sample TI MEASUREMENTS
Open_files: ALLOCATE Tidata$(1:Num_samples)[5] !RAW TI DATA
ASSIGN @Counter TO Counter_addr:BYTE
ASSIGN @Buffer TO BUFFER [INT(Buffer_length)]
OUTPUT Probe_addr:Probe$
CLEAR @Counter
OUTPUT @Counter:"IN1"
OUTPUT @Counter:"SRSA15D1"
OUTPUT @Counter:"TRTA0"
OUTPUT @Counter:"TO0"
1760 OUTPUT "Counter: SS17B!" 'SEND FAST BINARY COMMAND
1770 ENTER @ COUNTER USING "-k"; Buf$ ! Used only for 5370A
1780 Total_time=0
1790 DISP "Making "$Title!$" measurements (FAST TI) ..."
1800 ON EDT @Counter, 15 GOSUB Stop_timer
1810 TRIGGER @Counter !BEGIN MEASUREMENT -Delete fo 5370A
1820 Start_time=TIMEDATE
1830 !
1840 !*** I/O MODE #1 *** approximately 700 readings/sec ***************
1850 !
1860 TRANSFER @Counter TO @Buffer; RECORDS Buffer_length/5, EOR (CONT 5, END)
1870 ! This is the most flexible mode. It uses EDI to divide the records, thus
1880 ! slowing the transfer, but allowing a sample size to be determined while
1890 ! the program is running.
1900 !
1910 !*** I/O MODE #2 *** approximately 5000 readings/sec ***************
1920 !
1930 ! TRANSFER @Counter TO @Buffer: WAIT
1940 ! This is the fastest mode. It requires that the 5370B or the HP-IB cable
1950 ! be modified so EDI is not asserted on the last byte of each 5 byte
1960 ! transfer. Due to uncertainty about the first 5 bytes output, this
1970 ! scheme will not work with an unmodified 5370A. Also it fills all of
1980 ! Tidata(*), so the sample size must be determined by program editing.
1990 !
2000 !*** I/O MODE #3 *** approximately 500 readings/sec ***************
2010 !
2020 ! ENTER @Counter USING "y"; Tidata(*)
2030 ! GOSUB Stop_timer ! Fake EDT
2040 ! This method provides slower I/O than the formatted transfer scheme,
2050 ! and is formatted to take care of EDI. It fills all of Tidata(*), so
2060 ! the sample size must be determined by program editing.
2070 !
2080 ! DISP "Dumping data buffer into string array ..."
2090 ! FOR I=1 TO Num_samples
2100 ! ENTER @Buffer USING "%8.5A"; Tidata$(I) !FILL TI_DATAS(*)
2110 ! NEXT I
2120 ! LOCAL @Counter !CLEAR FAST TI MODE
2130 ! DISP "Converting TI values from fast data format ..."
2140 ! Convert_tti(Tidata$(*) , Tval$(*) , (Num_samples) , (Resolution))
2150 ! ASSIGN @Buffer TO * !RELEASE MEMORY
2160 ! ASSIGN @Counter TO *
2170 ! DEALLOCATE Tidata$(*)
2180 ! PRINT USING "y"
2190 ! DISP ""
2200 ! RETURN
2210 !
2220 ! Stop_time: STOP_TIME=TIMEDATE
2230 ! Meas_time=(INT((Stop_time-Start_time)*100))/100
2240 ! Total_time=TOTAL_TIME*Meas_time
2250 ! Meas_time=TOTAL_TIME
2260 ! ABORT @Counter
2270 ! LOCAL @Counter !CLEAR FAST TI MODE; remove this line for
2280 ! multiple buffer transfers!
2290 ! RETURN
2300 !
2310 !
2320 !
2330 ! Time_plot: 1DO A PLOT OF SAMPLE VALUES VS. MEASUREMENT TIME
2340 ! GCLEAR
Time_flag=1
Title25=Title5&"VALUES VS. TIME"
PEN 1
GOSUB Do_titles
Xy_limits=INT(Meas_time*100)
X_maj=FNSel_ax(Xy_limits)
Xy_limits=INT((MAX(Tivalue(*))+MIN(Tivalue(*)))*1.E+12/Resoluti
on)*+.5)
IF Xy_limits<1 THEN Xy_limits=1
Y_maj=FNSel_ax(Xy_limits)
VIEWPORT Plot_left,Plot_right,Plot_bottom,Plot_top
FRAME
WINDOW 0,Meas_time*100,0,Xy_limits !SIZE/ASPECT RATIO OF GRAPH
AXES X_maj/5,Y_maj/5,0;X_maj,Y_maj
PENUP
WINDOW Plot_left,Plot_right,Plot_bottom,Plot_top
CLIP OFF
CSIZE 3.5,.6
LORG 6
MOVE (Plot_right-Plot_left)/2+Plot_left,Plot_bottom-.07*Y_gdu_max
X
LABEL "Time Into the Measurement (sec)"
LORG 4
LORG 6
LDIR 90
MOVE Plot_left-.13*X_gdu_max,Plot_top-Plot_bottom)/2+Plot_botto
m
LABEL Title1&" Values (ns)"
LDIR 0
WINDOW 0,100,0,Xy_limits !Y AXIS MUST MATCH GRAPH
PEN 3
LORG 8
FOR I=0 TO Xy_limits STEP Y_maj IPT VALUES ON Y AXIS
X_val=MIN(Tivalue(*))+I*(1.E-12)*Resolution
MOVE -.5,I !SMIDGEON TO THE LEFT
GOSUB Draw_param
NEXT I
LORG 6
WINDOW 0,Meas_time*100,0,20 !X AXIS MUST MATCH GRAPH
MOVE 0,-5 !SMIDGEON UNDERneath X AXIS
LABEL USING ",.K":0
MOVE Meas_time=100,-.5
LABEL USING ",.DDDD.DD":Meas_time
WINDOW 0,Meas_time*100,0,Xy_limits !SIZE/ASPECT RATIO OF GRAPH
CLIP ON
PEN 2
MOVE 0,0
Plot_ratio=(Num_samples/(Meas_time*100))
J=1
FOR I=1 TO Meas_time*100
DRAW I,(Tivalue(IN(J))+MIN(Tivalue(*)))*1.E+12/Resolution
J=J+Plot_ratio
NEXT I
PENUP
Time_flag=F
RETURN
Sort_data: DISP "Sorting data samples ..."
MAT SORT Tivalue
DISP ""
RETURN

Compute_stats(Ivalue(*),(Num_samples),(Resolution),Stat_min,Stat_max,Stat_mean,Stat_dev)

DISP ""
RETURN

Compute_hist: DISP "Filling histogram bins . . ."
FOR I=Stat_min TO Stat_max + Resolution / 2.E+12 STEP Resolution / 1.E+12
    Bin_num = Bin_num + 1
    Hist_value(Bin_num) = Hist_value(Bin_num) + Resolution / 2.E+12
    NEXT I
    Hist_hist = Hist_hist + 0
    Fill_hist(Ivalue(*), Hist_value(*), (Num_samples), Hist(*), (Num_bins))

DISP ""
RETURN

Graph_it: TITLE$="HISTOGRAM" !DRAW THE HISTOGRAM DISPLAY
GCLEAR
PEN 1
GOSUB Do_titles
GOSUB Plot_axes
GOSUB Label_axes
PEN 2
GOSUB Plot_hist
PEN 4
GOSUB Do_param
RETURN

!******************************************************************************

Graph_it SUBROUTINES

!******************************************************************************

Do_titles: SUBROUTINE TO PUT TITLES AND PARAMETER NAMES ON PLOT
VIEWPORT 0, X_gdu_max, 0, Y_gdu_max
LORG 6
CSIZE 5, .6
FOR I=Minx TO Maxx STEP Stepx
LORG 6
MOVE X_gdu_max / 2 + I, Y_gdu_max
LABEL Title$;
PEN_UP
LORG 4
MOVE X_gdu_max / 2 + I .8 !PLACE TITLE ABOVE SOFT KEYS
LABEL I$;
NEXT I
VIEWPORT .14 * X_gdu_max , .86 * X_gdu_max , .08 * Y_gdu_max , .13 * Y_gdu_max
WINDOW .14 * X_gdu_max , .86 * X_gdu_max , .08 * Y_gdu_max , .13 * Y_gdu_max
FRAME
IF Time_flag THEN T_done
VIEWPORT .08 * X_gdu_max , .92 * X_gdu_max , .13 * Y_gdu_max , .19 * Y_gdu_max
WINDOW .08 * X_gdu_max , .92 * X_gdu_max , .13 * Y_gdu_max , .19 * Y_gdu_max
FRAME
LORG 2
CSIZE 2.8..6
MOVE .08*X_gdu_max+4.,13*Y_gdu_max+4
LABEL Stat_min$;
MOVE .08*X_gdu_max+4.,13*Y_gdu_max+2
LABEL Stat_max$;
MOVE .08*X_gdu_max+60.,13*Y_gdu_max+4
LABEL Stat_mean$;
MOVE .08*X_gdu_max+60.,13*Y_gdu_max+2
LABEL Stat_dev$;

I_done: PENUP
RETURN
!
!
!
!
!
!
!

Plot_axes:!! SIZE/DRAW AXES ACCORDING TO HISTOGRAM VALUES
VIEWPORT Plot_left,Plot_right,Plot_bottom,Plot_top
WINDOW 0,Num_Bins,0.MAX(Hist(*)) SIZE/ASPECT RATIO OF GRAPH
LINE 1
X_major=FSel_ax(Num_bins)
XY_limits=MAX(Hist(*))
Y_major=FSel_ax(XY_limits)
AXES X_major/5,1,0,0,X_major,Y_major
PENUP
RETURN
!
!
!
!
!
!
!
Label_axes:!! SUBROUTINE TO PUT LABEL VALUES ON HISTOGRAM AXIS
VIEWPORT Plot_left,Plot_right,Plot_bottom,Plot_top
WINDOW Plot_left,Plot_right,Plot_bottom,Plot_top
CLIP OFF
CSIZE 3.5.,6
LORD 4
DEC
DIO 90
MOVE Plot_left-.13*X_gdu_max,(Plot_top-Plot_bottom)/2+Plot_top

LABEL "Occurences"
LDIR 0
LORD 6
MOVE (Plot_right-Plot_left)/2+Plot_left,Plot_bottom-.07*Y_gdu_max
LABEL Title$" Measurements (ns)"
WINDOW 0,100,0.MAX(Hist(*)) !Y AXIS MUST MATCH GRAPH
LORD 8
FOR I=0 TO MAX(Hist(*)) STEP Y_major !PUT VALUES ON Y AXIS
MOVE -.51!SMIDGEON TO THE LEFT
NEXT I
WINDOW 0,Num_bins,0.20 !X AXIS MUST MATCH GRAPH
LORD 6
!MIN X VALUE
X_val=Stat_min
X_pos=FSN.X_pos(X_val,Hist_value(*),Num_bins)
X_value$="min"
Value_flag=1
GOSUB Label_it
!MAX X VALUE
X_val=Stat_max
X_pos=FSN.X_pos(X_val,Hist_value(*),Num_bins)
X_value$="max"
GOSUB Label_it
4100 !MEAN X VALUE
4110 X_val=Stat_mean
4120 X_pos=FNX_pos(X_val,Hist_value(*),Num_bins)
4130 X_value$="mean"
4140 GOSUB Label_it
4150 Line=7
4160 Descent=.006*Y_gdu_max
4170 X_offset=0
4180 GOSUB Draw_line
4190 !LOW STANDARD DEVIATION BOUNDARY
4200 X_val=Stat_dev
4210 X_pos=FNX_pos(PRINT((Stat_mean*X_val)*1.E+12,LGT(Resolution))*1.1.E-12,Hist_value(*),Num_bins)
4220 Value_flag=F
4230 Line=4
4240 Descent=.003*Y_gdu_max
4250 IF Resolution<10 THEN X_offset=-.004*X_gdu_max
4260 GOSUB Draw_line
4270 !HIGH STANDARD DEVIATION BOUNDARY
4280 X_pos=FNX_pos(PRINT((Stat_mean*X_val)*1.E+12,LGT(Resolution))*1.1.E-12,Hist_value(*),Num_bins)
4290 Line=4
4300 Descent=.003*Y_gdu_max
4310 IF Resolution<10 THEN X_offset=.004*X_gdu_max
4320 GOSUB Draw_line
4330 PENUP
4340 RETURN
4350 !
4360 !Label_it: 1PUT LABEL ON X AXIS (WITH VALUE)
4380 WINDOW 0,Num_bins,0,20 !X AXIS MUST MATCH GRAPH, Y IS FIXED
4390 CLIP OFF
4400 MOVE X_pos,-1.3
4410 LABEL X_value$
4420 IF NOT Value_flag THEN Lbl_done
4430 MOVE X_pos,-.5
4440 GOSUB Draw_param
4450 Lbl_done: PENUP
4460 RETURN
4470 !
4480 Draw_line: LINE TYPE Line     !DRAW MEAN AND STD. DEV LINES
4490 WINDOW 0,Num_bins,0,MAX(Hist(*)!) BOTH AXES MUST MATCH GRAPH
4500 CLIP OFF
4510 PEN 4
4520 MOVE X_pos+X_offset,0
4530 DRAW X_pos+X_offset,MAX(Hist(*))
4540 PENUP
4550 !DRAW DESCENDER
4560 WINDOW 0,Num_bins,0,20 !X AXIS MUST MATCH GRAPH, Y IS FIXED
4570 CLIP OFF
4580 MOVE X_pos+X_offset,0
4590 DRAW X_pos+X_offset,-Descent
4600 PENUP
4610 LINE TYPE 1
4620 X_offset=0
4630 Descent=0
4640 RETURN
4650 !
4660 !Plot_hist: !SUBROUTINE TO PLOT THE HISTORAM OF THE DATA IN I:valor.
4670 VIEWPORT Plot_left,Plot_right,Plot_bottom,Plot_top
4680 PENUP
WINBUH U.Num_bins,0,MAT(Hist(*))  
LINE TYPE 1  
MOVE -2.0  
FOR I=1 TO Num_bins  
MOVE 1.0  
DRAW I.Hist(I)  
NEXT I  
PENUP  
RETURN  

ENTRY POINT FOR DISPLAY OF PARAMETER VALUES  
VIEWPORT .08*X_gdu_max+.92*X_gdu_max,.13*Y_gdu_max,.19*Y_gdu_max  
WINDOW .08*X_gdu_max,.32*X_gdu_max,.13*Y_gdu_max,.19*Y_gdu_max  
LINE TYPE 1  
LDOR 2  
CSIZE 2.8,.6  
MOVE .08*X_gdu_max+24,.13*Y_gdu_max+4  
X_val=Stat_min  
GOSUB Draw_param  
MOVE .08*X_gdu_max+24,.13*Y_gdu_max+2  
X_val=Stat_max  
GOSUB Draw_param  
MOVE .08*X_gdu_max+85,.13*Y_gdu_max+4  
X_val=Stat_mean  
GOSUB Draw_param  
MOVE .08*X_gdu_max+85,.13*Y_gdu_max+2  
LABEL USING "#,DDD.DDD";Stat_dev*1.E+9  
PENUP  
RETURN  

DRAW PARAMETERS WITH PROPER RESOLUTION  
SELECT INT(LGT(Resolution))  
CASE =0  
LABEL USING "#,DDD.DDD";X_val*1.E+9  
CASE =1  
LABEL USING "#,DDD.DD";X_val*1.E+9  
CASE =2  
LABEL USING "#,DDD.D";X_val*1.E+9  
CASE =3  
LABEL USING "#,DDD";X_val*1.E+9  
END SELECT  
PENUP  
RETURN  

RESOLUTION AND SAMPLE SIZE SUBROUTINES  

SUBROUTINE TO CHANGE THE MEASUREMENT RESOLUTION  
IF Auto THEN Resolution=Dresolution  
IF Auto THEN Fr_done  
PRINT USING "G"  
GCLEAR  
PRINT "Enter the desired resolution of the display"  
PRINT "in ps."
RES entry(Resolution)  
PRINT USING "G"  
PRINT "Disp. Res.=";Resolution;"ps","Sample Size=";Num_samp
RETURN
!

Fix_samples:!!SUBROUTINE TO SET/ADJUST THE NUMBER OF SAMPLES TAKEN FOR THE

HISTOGRAM PLOT

IF Auto THEN Num_samples=0num_samples

IF Auto THEN Fs_done

PRINT USING "@"

GCLEAR

PRINT "Enter the number of samples to be taken."

PRINT "NOTE: The max value shown below is a typical"

PRINT "maximum based upon the amount of memory"

PRINT "presently installed in the computer."

PRINT "with the number of samples set equal to"

PRINT "the number of histogram bins."

PRINT "The actual maximum may be lower or"

PRINT "higher depending upon the range of"

PRINT "samples taken and the displayed"

PRINT "resolution value chosen."

PRINT

Samp_entry(Num_samples)

PRINT USING "@"

PRINT "Disp. Res. =";Resolution;"ps","Sample Size=";Num_samples

RETURN

PLOTTER CONTROL SUBROUTINES

Ext_plot: GINIT

PLOTTER IS Plotter_addr,"HPGL"

Minx=1

Maxx=1

Stepx=1

RETURN

Int_plot: GINIT

PLOTTER IS 3,"INTERNAL"

Maxx=-2

Minx=Maxx

Stepx=.1

RETURN

FUNCTION KEY LABELS AND FLAG SUBROUTINES

Auto_demo: Auto=NOT (Auto)

SELECT Auto

CASE -F

ON KEY 7 LABEL " DEMO ON",13 GOSUB Auto_demo

GOTO Auto_done

CASE =T

ON KEY 7 LABEL "AUTODEMO ON",13 GOSUB Auto_demo

END SELECT

Set_auto_param: Res_entry(Pw_auto_res,"SET RESOLUTION OF THE PULSE WIDTH

HISTOGRAM")

Samp_entry(Pw_auto_samp,"SET SAMPLE SIZE FOR THE PULSE W

HISTOGRAM")

Res_entry(Rt_auto_res,"SET RESOLUTION OF THE RISE TIME FI

STOGRAM")

Samp_entry(Rt_auto_samp,"SET SAMPLE SIZE FOR THE RISE TIM
5040:  res_entry(Ft_auto_res, "SET RESOLUTION OF THE FALL TIME H:
5050:  S7H0GRAM")
Samp_entry(Ft_auto_samp,"SET SAMPLE SIZE FOR THE FALL TIM
5060:  E HISTOGRAM")
S560:  Contin=False
5870:  GOSUB Single_cont
5880:  Do_auto:
5890:  GOSUB P_width
5900:  GOSUB R_time
5910:  GOSUB F_time
5910:  GOTO Do_auto
5920:  Auto_done: RETURN
5930 !
5940 !
5950:  Single_cont: Contin=NOT (Contin)
5960:  ON KEY 8 LABEL "Cont ON",14 GOSUB Single_cont
5970:  IF NOT Contin THEN ON KEY 8 LABEL "Single ON",14 GOSUB Single_ cont
5980:  RETURN
5990 !
6000 :!
6010:  T_plot: T_plot_flag=NOT (T_plot_flag)
6020:  ON KEY 3 LABEL "Time Plot ON",14 GOSUB T_plot
6030:  IF NOT T_plot_flag THEN ON KEY 3 LABEL "Time Plot OFF",14 GOSUB T_p lot
6040:  RETURN
6050 !
6060 :!
6070:  E_plot: Plot_flag=NOT (Plot_flag)
6080:  ON KEY 4 LABEL "Ext Plot OFF",14 GOSUB E_plot
6090:  IF Plot_flag THEN ON KEY 4 LABEL "Ext Plot ON",14 GOSUB E_plot
6100:  RETURN
6110 :!
6120 :!
6130:  Init_program: !**** DEFINE EQUIPMENT SETUP ****************************
6140:  IF TRIM(SYSTEMS("SYSTEM ID"))="9836A" THEN Disp_length=80
6150:  PRINT TABXY(INT((Disp_length-LEN(Meas_id$))/2),1);Meas_id$
6160:  PRINT TABXY(1,3);"Equipment Required:"
6170:  PRINT TAB(3);"5370B Universal Time Interval Counter"
6180:  PRINT TAB(3);"5363B Time Interval Probes"
6190:  PRINT TAB(3);"9826/36 Desktop Computer"
6200:  PRINT TABXY(1,8);"This procedure is designed to show the"
6210:  PRINT "capabilities of the 5370B Universal Time Interval"
6220:  PRINT "Counter and 5363B Time Interval Probes in"
6230:  PRINT "measuring pulse stream characteristics. From these"
6240:  PRINT "measurements of pulse width or rise/fall time are"
6250:  PRINT "histogram is generated showing measured time"
6260:  PRINT "w, number of occurrences."
6270:  PRINT
6280:  PRINT "If not already done, be sure to calibrate the"
6290:  PRINT "5363B probes to ensure accurate time measurement."
6300:  INPUT "Counters HP-IB address? (default=703)";Counter_addr
6310:  PRINT TABXY(41,4);"HP-IB-";Counter_addr
6320:  ASSIGN #Counter TO Counter_addr
6330:  INPUT "II Probe's HP-IB address? (default=707)";Probe_addr
6340:  PRINT TABXY(41,5);"HP-IB-";Probe_addr
6350:  Ids$" graph title "
6360:  INPUT "Do you plan on using an HPGL PLOTTER (Y/N)?",Answer$
6370:  IF Answer$(1,1)="Y" OR Answer$(1,1)="y" THEN Yes_plot
6380:  GOTO Init_exit
6390:  Yes_plot: INPUT "Plotters' HP-IB address? (default=705)";Plotter_addr
6400:  Yes_plot=T
6410 Init_exit:  RETURN  
6420 !  
6430 !  
6440 Quit:! CLEAR GRAPHICS, VARIABLES, LOCAL HP-IB AND QUIT.
6450  GCLEAR  
6460  GINIT  
6470  PRINT USING "$"  
6480  DISP ""  
6490  ASSIGN @Buffer TO *  
6500  ASSIGN @Counter TO *  
6510  OFF KEY 0  
6520  OFF KEY 1  
6530  OFF KEY 2  
6540  OFF KEY 3  
6550  OFF KEY 4  
6560  OFF KEY 5  
6570  OFF INTR  
6580  CLEAR Counter_addr  
6590  CLEAR Probe_addr  
6600  LOCAL Counter_addr  
6610  LOCAL Probe_addr  
6620 END  
6630 !  
6640 !---------------------------- END OF MAIN PROGRAM -------------------------------
6650 !  
6660 !  
6670 !------------------- SUBPROGRAMS AND FUNCTIONS -------------------------------
6680 !  
6690 SUB Res_entry(INTEGER Resolution,OPTIONAL A_res$)  
6700 COM /Sample/ INTEGER Max_samples,Dnum_samples,Dresolution,Mresolution  
6710 ALLOCATE Resolution$[20]  
6720 PRINT "min=";Mresolution;"", max= 1000, default="";Dresolution  
6730  
6740 IF NPAR=1 THEN Get_res  
6750 PRINT A_res$  
6760 Get_res:  
6770 Resolution$=""  
6780 LINPUT Resolution$  
6790 Resolution$=UPCS(TRIM$(Resolution$))  
6800 SELECT Resolution$  
6810 CASE =""  
6820 Resolution=0  
6830 GOTO Set_res  
6840 CASE ="MIN"  
6850 Resolution=Mresolution  
6860 GOTO Fr_done  
6870 CASE ="MAX"  
6880 Resolution=9999  
6890 GOTO Set_res  
6900 END SELECT  
6910 Set_res:  
6920 IF Resolution<0 THEN Resolution=Dresolution  
6930 IF Resolution<1 THEN $130  
6940 IF Resolution>9999 THEN Resolution=1000  
6950 Fr_done: Resolution=10*INT(LGT(Resolution)) !INTEGER POWERS OF 10 ONLY  
6960 DEALLOCATE Resolution$  
6970 SUBEND  
6980 !  
6990 !  
7000 SUB Samp_entry(INTEGER Num_samples,OPTIONAL A_samp$)
CUM /Sample/ inIt&TTX Max_samples.Dnum_samples.Dresolution.Mresolution
ALLOCATE Num_samples$20
PRINT "max:"Max_samples," default="Dnum_samples
PRINT IF NPAR=1 THEN Get_samp
PRINT A_samp5
Get_samp: Num_samples$=""
INPUT Num_samples$
Num_samples$=UPC$(TRIM$(Num_samples$))
SELECT Num_samples$
CASE ="
    Num_samples=0
GOTO Set_samp
CASE ="MIN"
    Num_samples=1
GOTO Fs_done
CASE ="MAX"
    Num_samples=Max_samples
GOTO Fs_done
END SELECT
Num_samples=VAL(Num_samples$)
IF Num_samples>Max_samples THEN
    BEEP
    DISP " Entered value (";Num_samples;") is > max"
    WAIT 2
    Num_samples=Max_samples
END IF
Set_samp: IF Num_samples<0 THEN Num_samples=Dnum_samples
Fs_done: DEALLOCATE Num_samples$
PRINT USING "#"
SUBEND

***************

SUB Convert_t(tdata$(*),REAL Tvalue(*),INTEGER Array_size,Res)
CONVERT raw TI data in tdata$(*), to REAL time interval values, with a resolution, and passes back as the REAL array Tvalue(*).

I INTEGER I
Const1=2.16
Const2=2.8
Const3=2.17
Const4=2.18
FOR I=1 TO Array_size
    N=NUM(tdata$[I][4,4])#256+NUM(tdata$[I][5,5])
    Q=1
    IF NOT BIFIT(NUM(tdata$[I][1,1]),5) THEN Q=-1
    B=BIF(1+(NUM(tdata$[I][1,1]),3)*const1+NUM(tdata$[I][2,2])*const2+NUM(tdata$[I][3,3]))
    NEXT_I
    NEXT I
Next_samp: NEXT I
SUBEND

DEF FNX_pos(Value,Array(*),INTEGER Array_size)
RETURN the exact or next highest bin number belonging to Value in Array(*)
INTEGER Curr_pos, Next_pos, Low_limit, High_limit

Low_limit = 0
High_limit = Array_size + 1
Curr_pos = INT ((Array_size/2) + .51)

Another_x: SELECT Value
CASE = Array (Curr_pos)
   GOTO Found_x
CASE < Array (Curr_pos)
   High_limit = Curr_pos
   IF Curr_pos < 1 THEN
     IF Value > Array (Curr_pos - 1) THEN Found_x
   END IF
   Curr_pos = Curr_pos - INT ((High_limit - Low_limit) / 2)
   IF Curr_pos < 1 THEN Another_x
   Curr_pos = 1
   GOTO Found_x
CASE > Array (Curr_pos)
   Low_limit = Curr_pos
   IF Curr_pos * Array_size AND Value < Array (Curr_pos + 1) THEN Correct_x
   Curr_pos = Curr_pos + INT ((High_limit - Low_limit) / 2)
   IF Curr_pos * Array_size THEN Another_x
   Curr_pos = Array_size
   GOTO Found_x
END SELECT
Correct_x: Curr_pos = Curr_pos + 1
Found_x: RETURN Curr_pos
FEND

********************************************************

SUB Fill_hist (REAL Sample_array(*), Range(*), INTEGER S_array_size, Hist(*), H_array_size)

!Sample_array(*) is a real array of <S_array_size> size,
!containing MAT SORT(ed) values, low to high.
!The Range(*) array contains the quantized values for the histogram
! (ie, x axis) to which the samples in Sample_array must be matched.
!The Hist(*) array is filled by this routine according to the number
!of samples in Sample_array which fit each value of Range(*).

INTEGER I, Bin_number

Bin_number = 1
FOR I = 1 TO S_array_size
   Try_again: ! COUNT SAMPLE INTO FIRST BIN THAT IT CROSS THE THRESHOLD OF
   IF Sample_array(I) < Range (Bin_number) THEN
     Hist (Bin_number) = Hist (Bin_number) + 1
   GOTO Next_tivalue
   END IF
   Bin_number = Bin_number + 1
   IF Bin_number > H_array_size THEN Next_tivalue ! IN CASE ALL
   GOTO Try_again
   GOTO Try_again
   Next_tivalue: NEXT I
SUBEND
DEF FNsel ax(INTEGER Axis_limit)
  INTEGER Xy_maj
  Xy_maj=1
  SELECT Axis_limit !GET CORRECT MAJOR LABEL
  CASE >5000
    Xy_maj=1000
  CASE >1000
    Xy_maj=500
  CASE >500
    Xy_maj=100
  CASE >100
    Xy_maj=50
  CASE >50
    Xy_maj=10
  CASE >10
    Xy_maj=5
  END SELECT
RETURN Xy_maj
FNEND

SUB Compute_stat(Tvalue(*),INTEGER Array_size,Res,REAL Stat_min,Stat_max,Stat_mean,Stat_dev)
  !Compute min, max, mean, and standard deviation values of Tvalue(*),
  !trim to (Res) resolution, and pass back.
  INTEGER I
  REAL Std_dev,C
  Mean=0
  Std_dev=0
  A=0
  B=0
  Stat_min=Tvalue(I)
  Stat_max=Tvalue(I)
  FOR I=1 TO Array_size
    IF Tvalue(I)<Stat_min THEN Stat_min=Tvalue(I)
    IF Tvalue(I)>Stat_max THEN Stat_max=Tvalue(I)
    A=A*Tvalue(I)
    B=B+(Tvalue(I)*Tvalue(I))
  NEXT I
  Mean=A/Array_size
  C=Array_size
  Std_dev=SQRT((Array_size*B-A*A)/(C*(C-1)))
  Stat_mean=ROUND(Mean*1.E+12,LT(Res)*1.E-12)
  Stat_dev=ROUND(Std_dev*1.E+12,0)*1.E-12
SUBEND

! ************************************************** THE END **************************************************
**Operation**

After making 1000 measurements (or whatever sample size has been chosen) in the fast binary mode, the program converts the raw time interval data into real time interval numbers, and plots the samples in the time interval vs. time form. The samples are then sorted, processed for statistics and displayed a second time in histogram form (number of samples vs. time interval value).

The time interval measurements are pulse width, rise time, and fall time. These three measurements are obtained through appropriate and automatic programming of the START and STOP trigger levels on the 5363B probes.

**NOTE:**

The parameters which may be measured by the system are limited to a minimum value of 10 ns. With modifications, the system will measure and process time intervals below 10 ns. (See page 24 — Program Limits).

**Getting Started**

1) Connect up the equipment as shown in figure 8.
2) Power on the 5370B counter and the 5363B probes.
3) Calibrate the 5363B probes and verify correct equipment setup.
   a) 5363B probe START and STOP settings:
      A + 0.00
   b) 5370B counter START and STOP settings:
      Trigger Level . . . PRESET
      Slope . . . . . . rising edge
      Impedance . . . 50 ohm
      Aten . . . . . . + 1
      Coupling . . . DC
      SEP/COM . . . SEP
   c) Insert both probes into their respective calibration sockets on the front panel of the 5363B.
      Press the TIME ZERO/LEVEL switch DOWN and verify that the probes calibrate properly (see manual if necessary). Now press the switch up to TIME ZERO and verify a reading between 30 and 70 μs (very approximate). This shows that the system is operational at the hardware level.
4) Now make a test measurement on your signal to verify operation. Change the probe setting to make, for example, a risetime measurement — START 20% peak, STOP 80% peak. Put the A probe on the test point.
5) You should now see a reading of the rise time on the counter. If you’re not getting anything on the counter, use the trigger indication lights on the counter to localize which channel is not operating. Trigger lights can also be useful in determining the actual peaks of the output waveform.

If the setup is working to this point, then you are ready to load the software into the computer and run the program. Note that UNLESS THE SETUP IS WORKING HERE, THE PROGRAM WILL NOT RUN CORRECTLY, AND/OR MAY GIVE YOU ERRONEOUS RESULTS.
6) Power on the computer and load BASIC 2.0 with the 2.1 Extensions. Graph 2 _1 must also be loaded to support the color graphics.
7) Load the program from your disc into the computer and press RUN. The program will start up with a brief description and then ask you for 5363B and 5370B addresses. The default addresses are:
   5363B . . . 07
   5370B . . . 03
8) If you are using a plotter, indicate by entering ‘y’, followed by the plotter’s address (default is 05). The program is now ready to make measurements.
Explanation of Softkeys

Pulse Width
This key initiates a set of pulse width measurements. The number of measurements made is determined by the sample size you’ve chosen (default is 1000). (The 5363B probes are addressed to trigger at +2.50v in the sample program).

Rise Time
Same as Pulse Width – (except probes are addressed to trigger at +0.50v (START) and +4.50v (STOP) in the sample program).

Fall Time
Same as Rise Time – (with probes set to +4.50v (START) and +0.50v (STOP) in the sample program).

Time Plot ON/OFF
This key simply enables the time plot (i.e., time intervals vs. elapsed time). The plot is useful in showing relatively long term trends in the time intervals (or frequency) coming from a clock.

Chg Resolution
Pressing this key allows you to change the resolution of the time and histogram plots. Remember that the 5370B’s scheme is limited to 20 ps quantization error, (unless averaging is used), so 10 ps of plot resolution is about the finest to use. ‘ENTER’ a number, or press ‘CONTINUE’ or ‘ENTER’ to default.

Chg # Samples
You may adjust the number of samples from the default value of 1000 using this key. The maximum is set by the size of the buffer array ‘Tidata’. This means the program has to be stopped to increase the maximum. ‘ENTER’ a number, ‘min’ or ‘max’, or press ‘CONTINUE’ or ‘ENTER’ to default.

DEMO/AUTODEMO
Pressing this key puts the program into the AUTODEMO mode, which will cycle through all three measurement modes (pulse width, rise time, fall time). It requires a resolution and sample size for each mode.
‘AUTODEMO’ puts the program into continuous mode, and may be exited by pressing the key again.

Single ON/Cont ON
In ‘Single’, the program makes one set of measurements and pauses after the histogram has been drawn. ‘Cont’ will continually cycle, doing one measurement set after another without a prompt (whether in DEMO or AUTODEMO).

Ext plot OFF/ON
The status of this key tells the program whether or not to draw a (4 color) plot of the time or histogram plots (which ever one it just finished drawing on the screen) on an external plot device. If you want, for example, a plot of the histogram on a plotter, but do not want the time plot, then press this key at some point after the time plot has been drawn on the screen and before the histogram is finished on the screen. Press the key once – it may take a second or two for the calculator to change the label if it is sorting or filling histogram bins.
Note: If you want a plot of the time intervals vs. time, you must press this key before the time plot is finished on the CRT. This is because the sorting process throws away all timing information.

Quit
This key provides a means of exiting the program.
Miscellaneous

Program limits

Precision time intervals tend to be limited in duration. Because of this the measurement capability of the system has been limited to a maximum of 999ns and a minimum of 7 to 10 ns. This reduces the computation time and program complexity at the expense of user flexibility.

For longer time intervals you will have to do major modifications to the data processing subroutines. Shorter time intervals may be measured with a slight program modification and careful attention to measurement setup. Basically, you must:

1) add the ‘AR2’ command to line #1720 (± TI mode),

2) make sure that the start pulse arrives at the counter before the stop pulse, and

3) if you are trying to make measurements of 1ns or below (e.g. rise or fall time measurements), you should use some input device for the counter other than the 5363B probes. This is because the probes have a 350MHz bandwidth limit and may introduce significant errors into the measurement. (See table 1.)

Using a 5370A

A 5370A will work in place of the ‘B’ model, with certain conditions being met.

1) Line #1770 must be included as the binary bytes must be ‘synchronized’ in the 5370A.

2) Line #1810 must be deleted.

3) Realize that the 5370A has an older design input amplifier system, which will not give the performance of the newer designed one in the ‘B’ model, when looking at very short time interval measurements.

Using a 5363A

The ‘A’ model probes may be used directly in place of the ‘B’ model with no changes to the program.

Using a Series 200 other than the 9836C

The program was written to utilize the color graphics capabilities of the 9836C. It will run without modification on the 9836A, 9826A or 9816A so long as the memory is large enough to support BASIC 2.0, the BASIC EXTENSIONS 2.1, and about another 1/4 megabyte for program, variable and matrix storage.

Increasing the fast binary transfer rate

In its current configuration, the program is set up to transfer readings from the counter to the computer at roughly 700-800/second. This is because the TRANSFER process in the series 200 is interrupted by EOI being asserted, and EOI is asserted by the 5370B at the end of each 5 byte message. Servicing these interrupts slows the transfer. Avoiding the use of the TRANSFER construct, or the assertion of EOI, are the only ways to achieve the maximum data transfer rate.

A data transfer rate of greater than 5000/sec can be obtained by using a computer other than a series 200, by using a language other than BASIC (such as PASCAL, in which you construct your own TRANSFER statement), or by modifying the HP-IB connection to break the EOI line.

The upper ceiling of approximately 6000 readings/sec is set by the need for 165 microseconds dead time between measurements. During this time the counter is reading registers and transferring the data through the HP-IB port to the computer.
Measuring Time Intervals Larger Than 320 Microseconds In Fast Binary

In its normal measurement mode, the 5370B makes time interval measurements using three internal hardware registers; N0, N1 and N2, plus a fourth register in RAM which handles overflow of the N0 register. The microprocessor in the counter updates this fourth register as part of its data processing routine, and includes this in the time interval, frequency or period result on the LED display.

In the fast binary mode, the counter does no processing of the data and, hence, updating of the fourth register. Therefore, it is possible to get erroneous results by overflowing the N0 register. This will happen when you attempt to make time interval measurements larger than 327.680000 microseconds. (Equivalent to 2^16 x 5ns (16 bit register)).

You can circumvent this problem if you know the approximate length of the time interval to be measured, within 320 microseconds. The final time interval answer can then be computed as the following:

\[ TI (\text{ns}) = ((\text{integer number of times the counter will overflow}) \times \text{overflow value}) + (\text{current count in the counter}) \]
\[ = ((\text{expected } TI, \text{ in } \mu \text{s}) / (327.680000\mu \text{s})) \times (327680.00\text{ns}) + ((\text{N1}+2)/256 + \text{N0}) \times 5\text{ns} \]

Example:
The time interval to be measured is about 5ms (probably within 100\mu s);
\[ TI (\text{ns}) = (5\text{ms} / 327.680000\mu \text{s}) \times (327680.00\text{ns}) + \text{xxxxxx.xxxns} \]
\[ = (15) \times (327680.00\text{ns}) + \text{xxxxxx.xxxns} \]
\[ = 4.91520000 \times 10^{-3} + \text{xxxxxx.xxxns} , \]

(where xxxxx.xxx is the value transferred from the 5370B registers)

*Must be an integer

Conclusion

We have seen that, by virtue of its high single shot resolution and its great measurement speed, a counter has an important role to play in pulse characterization. This is particularly significant in the production test environment. Here, the time required to make measurements, and the pressures for automation in general, are well met by a counts strengths.

The 5370B was shown to be capable of greater than 5000 measurements per second, and each one of these with 20 picoseconds of resolution. Also illustrated were some useful statistical and graphical presentations of results, which enhance analysis of time phenomena.

There are many applications for such capabilities, but particularly noteworthy are Disc Testing, IC Characterization, IC Tester Calibration and Digital or Data Communications Testing.