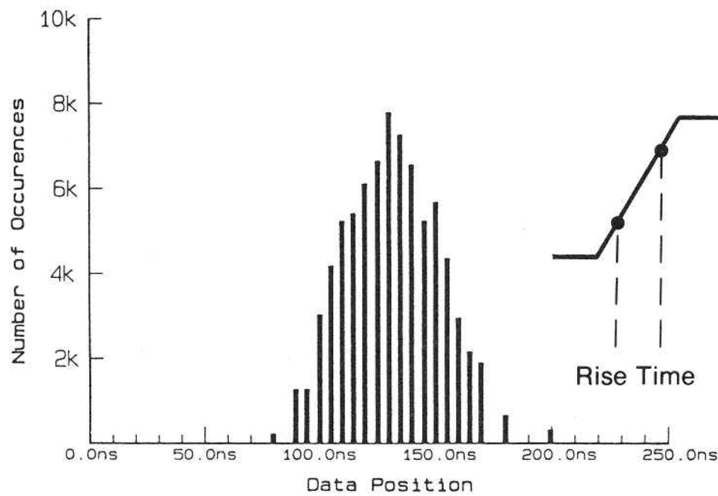
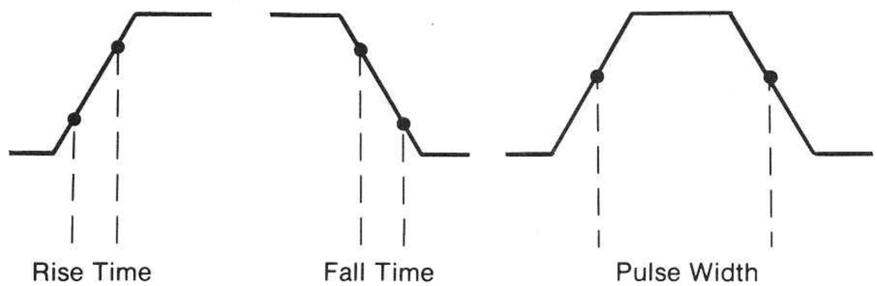
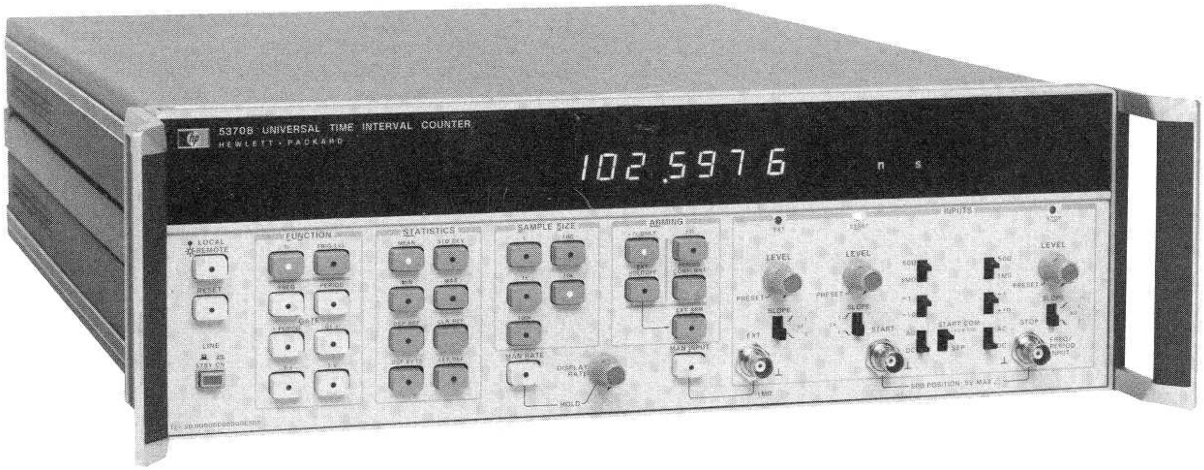
High Throughput Picosecond Characterization of Pulse Parameters







5370B UNIVERSAL TIME INTERVAL COUNTER

Product Note 5370B-3

HEWLETT PACKARD

Contents

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)

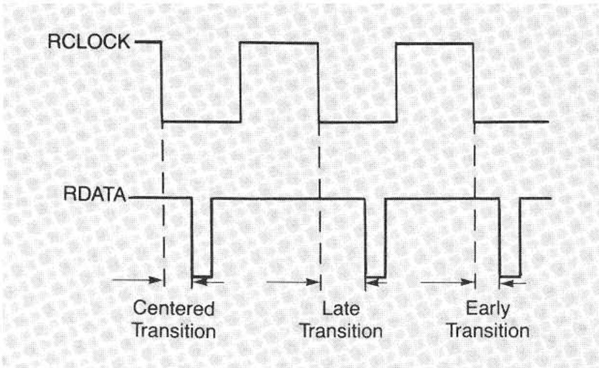


Figure 1.

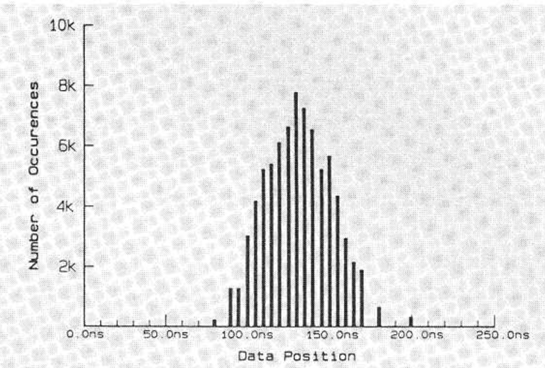


Figure 2.

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).

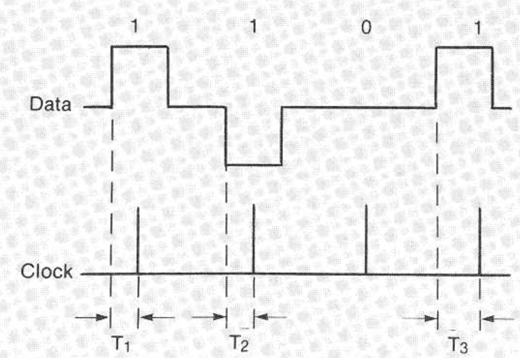


Figure 3.

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.

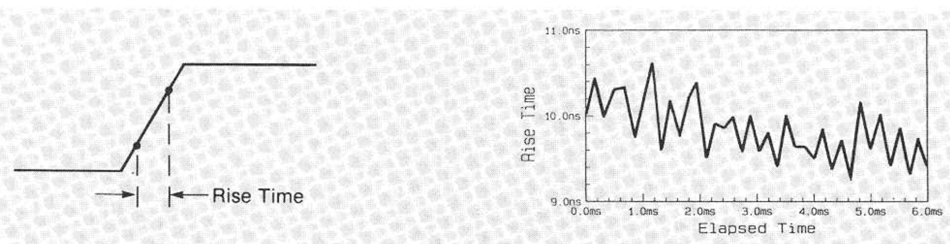


Figure 4.

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.

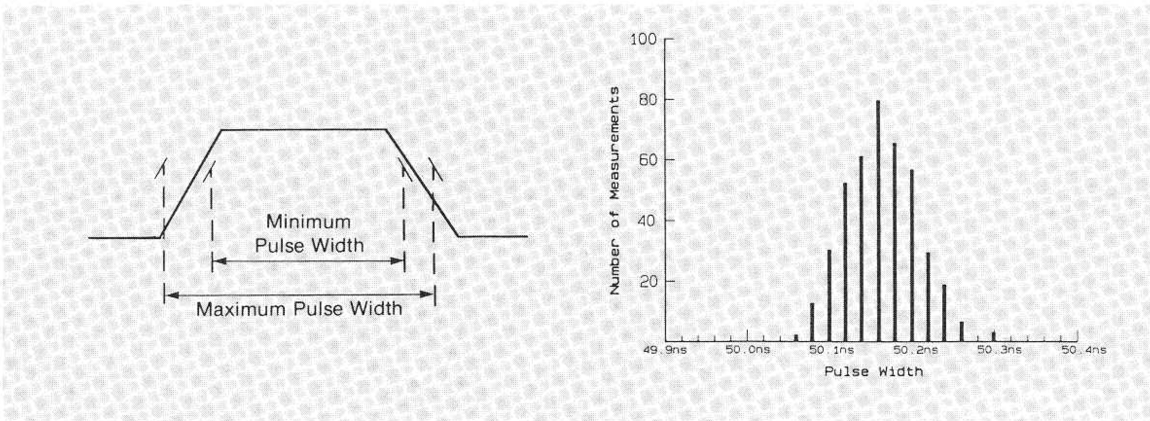


Figure 5.

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.

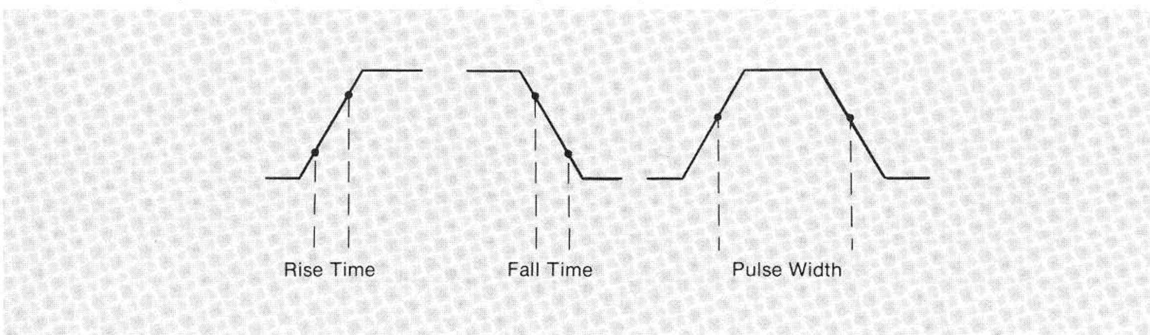


Figure 6.

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.

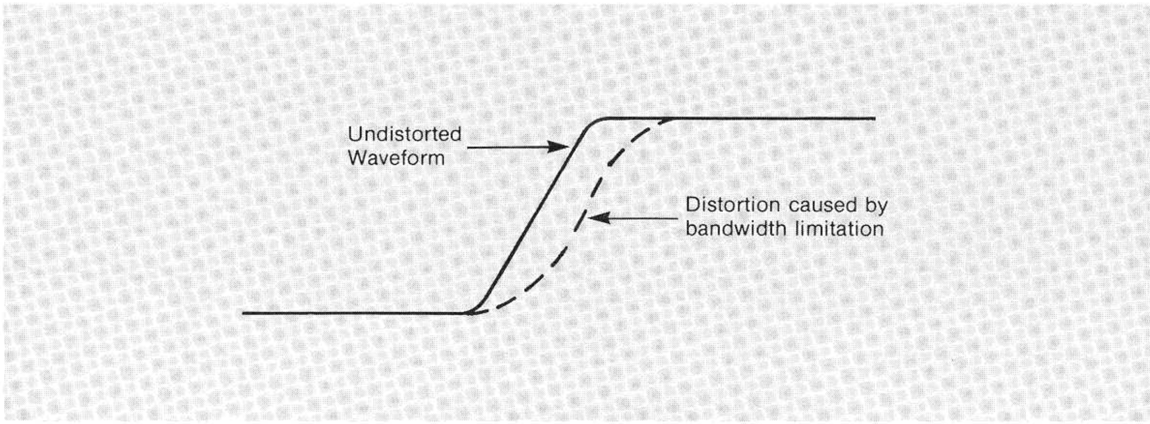


Figure 7.

To measure a signal's rise time accurately, the measurement system should have a rise time at least three times faster then 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.

Measuring Instrument Bandwidth / Rise Time	Measured Rise Time	Percent Error from 2 ns
250 MHz / 1.75 ns	2.66 ns	33%
350 MHz / 1.00 ns	2.24 ns	12%
500 MHz / 0.70 ns	2.12 ns	6%
1 GHz / 0.35 ns	2.03 ns	2%

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

Measuring Instrument Bandwidth / Rise Time	Measured Rise Time	Percent Error from 5 ns
250 MHz / 1.75 ns	5.30 ns	6.0%
350 MHz / 1.00 ns	5.10 ns	2.0%
500 MHz / 0.70 ns	5.05 ns	1.0%
1 GHz / 0.35 ns	5.01 ns	0.2%

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.

Counter works best when:	
• Events are non repetitive or infrequent	(Oscilloscopes generally need repetitive events)
• Measurement speed is important	(Because of their one shot nature, counters can make several thousand measurements per second)
• Operators have limited training	(Once parameters are known, a counter can be set up to measure manually or automatically very simply)
• High accuracy and resolution is required for long time intervals	(Counter resolution is not a function of time measured)
Oscilloscope works best when:	
• Visualization of the waveform is important	(Oscilloscopes allow the operator to integrate the "pattern" observed into data)
• Peak amplitude of the waveform is unknown	(It is tricky to ascertain the 20%, 50% and 80% points on a waveform with a counter)
• Voltage information must be gathered	(Oscilloscopes are inherently more useful for voltage measurements)

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 instrinsic 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.

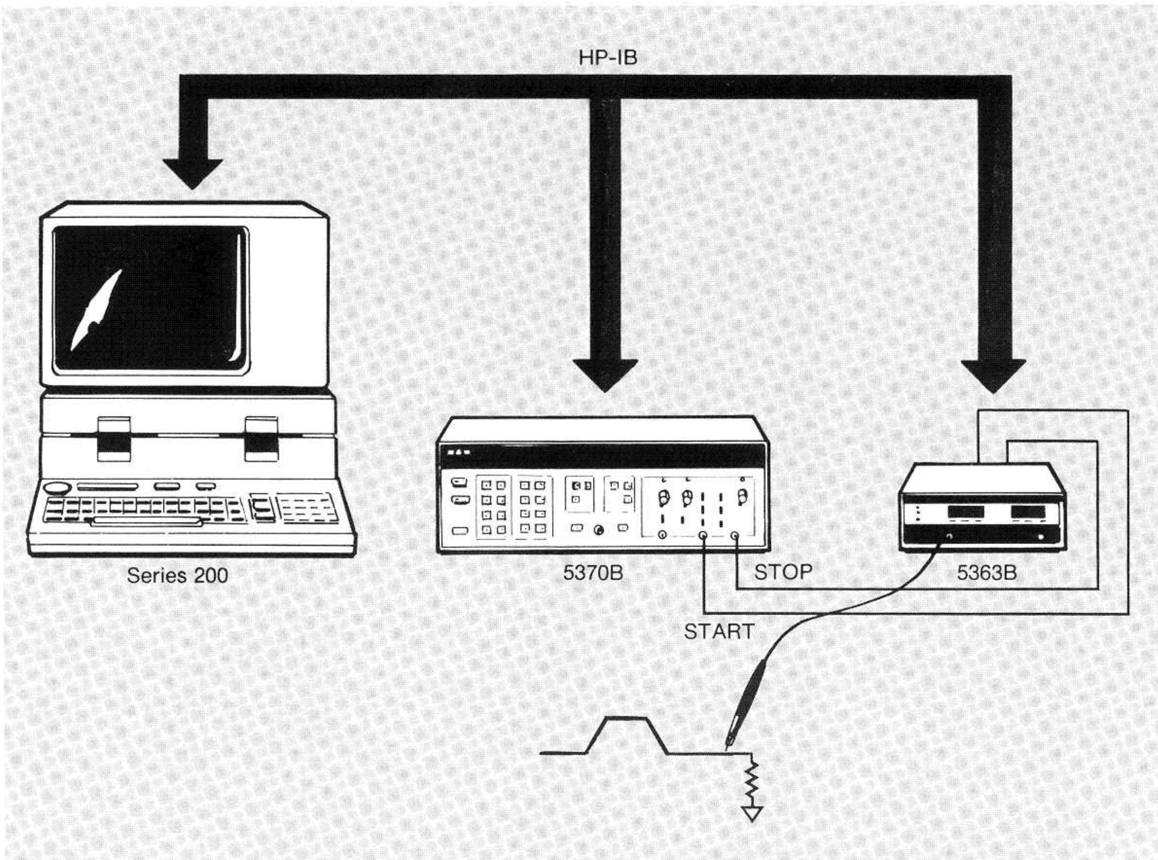


Figure 8.

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.

```
10!*******PROGRAM "PULSTAT" ************************ 9/20/83 *** REV C. *******
20!
30!       PROGRAM TO AUTOMATICALLY MEASURE AND CHARACTERIZE RISE/FALL TIMES
40!       AND PULSE WIDTHS OF A SERIES OF DIGITAL PULSES.
50!
60!       SYSTEM REQUIRES ONE EACH OF THE FOLLOWING:  5370B, 5363B.
70!
80!
90!****************************** VARIABLES ************************************
100   OPTION BASE 1
110 DIM Counter$[50],Probe$[50],Generator$[50],Title$[50],Title1$[50],Title2$[5
],Meas_id$[50]
120 DIM Stat_min$[20],Stat_max$[20],Stat_mean$[20],Stat_dev$[20],X_value$[10]
130 COM /Stats/ REAL Stat_min,Stat_max,Stat_mean,Stat_dev,Meas_time
140 COM /Sample/ INTEGER Max_samples,Dnum_samples,Dresolution,Mresolution
150 INTEGER Crt,Prtr,Counter_addr,Probe_addr,Generator_addr,Plotter_addr,Disp_l
ngth
160 INTEGER Resolution,Num_samples,Num_bins,Bin_num,X_pos,Xy_limits,X_maj,Y_maj
170 INTEGER Pw_auto_res,Pw_auto_samp,Rt_auto_res,Rt_auto_samp,Ft_auto_res,Ft_au
o_samp !Auto_sample/resolution values
180 INTEGER False,True,T,F,Plot_yes,Plot_flag,T_plot_flag,Time_flag  !Flags
190 !
200 ! ***************************INITIALIZATION *******************************
210 Crt=1
220 Prtr=701
230 False=0
240 True=1
250 F=False
260 T=True
270 Auto=F
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 Stepx=.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 Dresolution=10     !INITIAL (DEFAULT) RESOLUTION OF DISPLAYED RESULTS IN ps
440 Mresolution=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 Y_gdu_max=100*MAX(1,1/RATIO)
570 Plot_left=.20*X_gdu_max
```

```
580 Plot_right=.80*X_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(PROUND(INT((VAL(SYSTEM$("AVAILABLE MEMORY"))-4500)/Num
_arrays),2),65500/5)
710        Dnum_samples=MIN(Max_samples,1000)  !DEFAULT SAMPLE SIZE
720        Num_samples=Dnum_samples            !INITIAL SAMPLE SIZE
730        GRAPHICS ON
740        PRINT USING "@"
750        ON KEY 0 LABEL "Pulse Width",13 GOSUB P_width
760        ON KEY 1 LABEL "Rise Time",13 GOSUB R_time
770        ON KEY 2 LABEL "Fall Time",13 GOSUB F_time
780        ON KEY 3 LABEL "Time Plot ON",14 GOSUB T_plot
790        IF Plot_yes THEN ON KEY 4 LABEL "Ext Plot OFF",14 GOSUB E_plot
800        ON KEY 5 LABEL "Chg Resolution",13 GOSUB Fix_resolution
810        ON KEY 6 LABEL "Chg # Samples",13 GOSUB Fix_samples
820        ON KEY 7 LABEL " DEMO ON",13 GOSUB Auto_demo
830        ON KEY 8 LABEL "Single ON",14 GOSUB Single_cont
840        ON KEY 9 LABEL "Quit",14 GOTO Quit
850        PRINT "Disp. Res.=";Resolution;"ps","Sample Size=";Num_samples
860        GINIT
870        GCLEAR
880 !
890 Wait_loop:  DISP "Choose a key"
900             GOTO Wait_loop        !LOOP HERE BETWEEN MEASUREMENTS
910 !
920 !
930 !
940 ! ********************** EXEC MEASUREMENT SUBROUTINES *********************
950 !
960 P_width: !DO HISTOGRAM OF PULSE WIDTH
970           Title$="PULSE WIDTH "
980           Title1$="Pulse Width"
990           PRINT Title$
1000          Probe$="PGA+250URA+250DS"  ! This is where trigger levels are set
                                           on the 5363B
1010          IF Auto THEN Resolution=Pw_auto_res
1020          IF Auto THEN Num_samples=Pw_auto_samp
1030          GOTO Do_it
1040!
1050 R_time:!DO HISTOGRAM OF PULSE RISE TIME
1060         Title$="RISE TIME "
1070         Title1$="Rise Time"
1080         PRINT Title$
1090         Probe$="PGA+050URA+450US"   !  This is where trigger levels are set
                                           on the 5363B
1100         IF Auto THEN Resolution=Rt_auto_res
1110         IF Auto THEN Num_samples=Rt_auto_samp
1120         GOTO Do_it
1130!
1140 F_time:!DO HISTOGRAM OF PULSE FALL TIME
1150         Title$="FALL TIME "
```

```
1160        Title1$="Fall Time"
1170        PRINT Title$
1180        Probe$="PGA+450DRA+050DS"   !  This is where trigger levels are set
                                           on the 5363B
1190        IF Auto THEN Resolution=Ft_auto_res
1200        IF Auto THEN Num_samples=Ft_auto_samp
1210        GOTO Do_it
1220 !
1230 Do_it: !COMPLETE THE MEASUREMENT
1240        PRINT USING "@"
1250        DISP "Setting up measurement ..."
1260        Real_samples=Num_samples
1270        Real_samples=Real_samples*5
1280        Buffer_length=MIN(Real_samples,65500)
1290        ALLOCATE Tivalue(Num_samples) !Ti_data$ CONVERTED TO REAL VALUES
1300        GOSUB Make_measure
1310        IF NOT T_plot_flag THEN Sort
1320        GINIT
1330        GOSUB Time_plot
1340        IF NOT Plot_flag THEN Sort
1350        GOSUB Ext_plot
1360        GOSUB Time_plot
1370        GOSUB Int_plot
1380 Sort:  GOSUB Sort_data
1390        GOSUB Get_stats
1400        Num_bins=((Stat_max-Stat_min)*(1.E+12)/Resolution)+1
1410        ALLOCATE INTEGER Hist(Num_bins) !DIMENSION LARGEST POSSIBLE Hist(*)
1420        ALLOCATE Hist_value(Num_bins) !VALUES ASSIGNED TO EACH BIN
1430        GOSUB Compute_hist
1440        GINIT
1450        GOSUB Graph_it
1460        IF NOT Plot_flag THEN Deall
1470        GOSUB Ext_plot
1480        GOSUB Graph_it
1490        GOSUB Int_plot
1500 Deall: DEALLOCATE Hist_value(*)
1510        DEALLOCATE Hist(*)
1520        DEALLOCATE Tivalue(*)
1530        IF Auto THEN Wait
1540        IF Contin THEN Do_it
1550        LINPUT "Do another measurement? (y/n)",Answer$
1560        IF UPC$(Answer$[1,1])="Y" THEN Do_it
1570        GOTO Finished
1580 Wait:  IF Contin THEN Finished
1590        DISP "Hit CONTINUE to proceed."
1600        PAUSE
1610 Finished:   RETURN
1620 !
1630 !
1640 !******** DATA GATHERING AND DISPLAY SUBROUTINES CALLED BY EXECS **********
1650 !
1660 Make_measure: !MAKE Num_sample TI MEASUREMENTS
1670 Open_files:   ALLOCATE Tidata$(1:Num_samples)[5]         !RAW TI DATA
1680               ASSIGN @Counter TO Counter_addr;BYTE
1690               ASSIGN @Buffer TO BUFFER [INT(Buffer_length)]
1700               OUTPUT Probe_addr;Probe$
1710               CLEAR @Counter           !INITALIZE  COUNTER
1720               OUTPUT @Counter;"IN1"
1730               OUTPUT @Counter;"SRSA1S01"
1740               OUTPUT @Counter;"TRTA0"
1750               OUTPUT @Counter;"TO0"
```

```
1760              OUTPUT @Counter;"SS1TB1" !SEND FAST BINARY COMMAND
1770              !ENTER @ COUNTER USING "-K";Buf$   ! Used only for 5370A
1780              Total_time=0
1790              DISP "Making "&Title1$&" measurements (FAST TI) ..."
1800              ON EOT @Counter,15 GOSUB Stop_timer
1810              TRIGGER @Counter          !BEGIN MEASUREMENT -Delete fo 5370A
1820              Start_time=TIMEDATE
1830 !
1840 !*** I/O MODE #1 *** approximately 700 readings/sec *********************
1850 !
1860              TRANSFER @Counter TO @Buffer;RECORDS Buffer_length/5,EOR (COU
NT 5,END)
1870 !  This is the most flexible mode.  It uses EOI to divide the records, thus
1880 !  slowing the transfer, but allowing a sample size to be determined while
1890 !  the program is running.
1900 !
1910 !*** I/O MODE #2 *** approximately 5000 readings/sec ********************
1920 !
1930 !            TRANSFER @Counter TO @Buffer;WAIT
1940 !
1950 !  This is the fastest mode.  It requires that the 5370B or the HP-IB cable
1960 !  be modified so EOI is not asserted on the last byte of each 5 byte
1970 !  transfer.  Due to uncertainty about the first 5 bytes output, this
1980 !  scheme will not work with an unmodified 5370A.  Also it fills all of
1990 !  Tidata(*), so the sample size must be determined by program editing.
2000 !
2010 !*** I/O MODE #3 *** approximately 500 readings/sec *********************
2020 !
2030 !            ENTER @Counter USING "Y";Tidata(*)
2040 !            GOSUB Stop_timer                    ! Fake EOT
2050 !
2060 !  This method provides slower I/O than the formatted transfer scheme,
2070 !  and is formatted to take care of EOI.  It fills all of Tidata(*), so
2080 !  the sample size must be determined by program editing.
2090 !
2100              DISP "Dumping data buffer into string array ..."
2110              FOR I=1 TO Num_samples
2120              ENTER @Buffer USING "%,5A";Tidata$(I) !FILL TI_DATA$(*)
2130              NEXT I
2140              LOCAL @Counter             !CLEAR FAST TI MODE
2150              DISP "Converting TI values from fast data format ..."
2160              Convert_ti(Tidata$(*),Tivalue(*),(Num_samples),(Resolution))
2170              ASSIGN @Buffer TO * !RELEASE MEMORY
2180              ASSIGN @Counter TO *
2190              DEALLOCATE Tidata$(*)
2200              PRINT USING "@"
2210              DISP ""
2220              RETURN
2230 !
2240 Stop_timer:  Stop_time=TIMEDATE
2250              Meas_time=(INT((Stop_time-Start_time)*100))/100
2260              Total_time=Total_time+Meas_time
2270              Meas_time=Total_time
2280              ABORTIO @Counter
2290              LOCAL @Counter !CLEAR FAST TI MODE: remove this line for
                                   multiple buffer transfers!
2300              RETURN
2310 !
2320 !
2330 Time_plot: !DO A PLOT OF SAMPLE VALUES VS. MEASUREMENT TIME
2340            GCLEAR
```

```
2350            Time_flag=T
2360            Title2$=Title$&"VALUES VS.TIME"
2370            PEN 1
2380            GOSUB Do_titles
2390            Xy_limits=INT(Meas_time*100)
2400            X_maj=FNSel_ax(Xy_limits)
2410            Xy_limits=INT(((MAX(Tivalue(*))-MIN(Tivalue(*)))*1.E+12/Resoluti
on)+.5)
2420            IF Xy_limits<1 THEN Xy_limits=1
2430            Y_maj=FNSel_ax(Xy_limits)
2440            VIEWPORT Plot_left,Plot_right,Plot_bottom,Plot_top
2450            FRAME
2460            WINDOW 0,Meas_time*100,0,Xy_limits   !SIZE/ASPECT RATIO OF GRAPH
2470            AXES X_maj/5,Y_maj/5,0,0,X_maj,Y_maj
2480            PENUP
2490            WINDOW Plot_left,Plot_right,Plot_bottom,Plot_top
2500            CLIP OFF
2510            CSIZE 3.5,.6
2520            LORG 6
2530            MOVE (Plot_right-Plot_left)/2+Plot_left,Plot_bottom-.07*Y_gdu_ma
x
2540            LABEL "Time Into the Measurement (sec)"
2550            LORG 4
2560            DEG
2570            LDIR 90
2580            MOVE Plot_left-.13*X_gdu_max,(Plot_top-Plot_bottom)/2+Plot_botto
m
2590            LABEL Title1$&" Values (ns)"
2600            LDIR 0
2610            WINDOW 0,100,0,Xy_limits         !Y AXIS MUST MATCH GRAPH
2620            PEN 3
2630            LORG 8
2640            FOR I=0 TO Xy_limits STEP Y_maj  !PUT VALUES ON Y AXIS
2650            X_val=MIN(Tivalue(*))+I*(1.E-12)*Resolution
2660            MOVE -.5,I                       !SMIDGEON TO THE LEFT
2670            GOSUB Draw_param
2680            NEXT I
2690            LORG 6
2700            WINDOW 0,Meas_time*100,0,20      !X AXIS MUST MATCH GRAPH
2710            MOVE 0,-.5                       !SMIDGEON UNDERNEATH X AXIS
2720            LABEL USING "#,K";0
2730            MOVE Meas_time*100,-.5
2740            LABEL USING "#,DDDD.DD";Meas_time
2750            WINDOW 0.Meas_time*100,0,Xy_limits   !SIZE/ASPECT RATIO OF GRAPH
2760            CLIP ON
2770            PEN 2
2780            MOVE 0,0
2790            Plot_ratio=(Num_samples/(Meas_time*100))
2800            J=1
2810            FOR I=1 TO Meas_time*100
2820            DRAW I,((Tivalue(INT(J))-MIN(Tivalue(*)))*1.E+12)/Resolution
2830            J=J+Plot_ratio
2840            NEXT I
2850            PENUP
2860            Time_flag=F
2870            RETURN
2880 !
2890 !
2900 Sort_data:  DISP "Sorting data samples ..."
2910             MAT SORT Tivalue
2920             DISP ""
```

```
2930              RETURN
2940 !
2950 !
2960 Get_stats:  DISP "Computing statistics ..."
2970              Compute_stat(Tivalue(*),(Num_samples),(Resolution),Stat_min,Sta
t_max,Stat_mean,Stat_dev)
2980              DISP ""
2990              RETURN
3000 !
3010 !
3020 Compute_hist: DISP "Filling histogram bins ... "
3030               Bin_num=0
3040               FOR I=Stat_min TO Stat_max+Resolution/2.E+12 STEP Resolution/
1.E+12
3050               Bin_num=Bin_num+1
3060               Hist_value(Bin_num)=I+Resolution/2.E+12
3070               NEXT I
3080               MAT Hist= Hist*(0)
3090               Fill_hist(Tivalue(*),Hist_value(*),(Num_samples),Hist(*),(Num
_bins))
3100               DISP ""
3110               RETURN
3120 !
3130 !
3140 Graph_it: Title2$=Title$&"HISTOGRAM"      !DRAW THE HISTOGRAM DISPLAY
3150           GCLEAR
3160           PEN 1
3170           GOSUB Do_titles
3180           GOSUB Plot_axes
3190           PEN 3
3200           GOSUB Label_axes
3210           PEN 2
3220           GOSUB Plot_hist
3230           PEN 4
3240           GOSUB Do_param
3250           RETURN
3260!
3270!
3280!********************** "Graph_it" SUBROUTINES **********************
3290 !
3300 Do_titles:!SUBROUTINE TO PUT TITLES AND PARAMETER NAMES ON PLOT
3310           VIEWPORT 0,X_gdu_max,0,Y_gdu_max
3320           LORG 6
3330           CSIZE 5,.6
3340           FOR I=Minx TO Maxx STEP Stepx
3350           LORG 6
3360           MOVE X_gdu_max/2+I,Y_gdu_max
3370           LABEL Title2$
3380           PENUP
3390           LORG 4
3400           MOVE X_gdu_max/2+I,8   !PLACE TITLE ABOVE SOFT KEYS
3410           LABEL Id$;
3420           NEXT I
3430           VIEWPORT .14*X_gdu_max,.86*X_gdu_max,.08*Y_gdu_max,.13*Y_gdu_max
3440           WINDOW .14*X_gdu_max,.86*X_gdu_max,.08*Y_gdu_max,.13*Y_gdu_max
3450           FRAME
3460           IF Time_flag THEN T_done
3470           VIEWPORT .08*X_gdu_max,.92*X_gdu_max,.13*Y_gdu_max,.19*Y_gdu_max
3480           WINDOW .08*X_gdu_max,.92*X_gdu_max,.13*Y_gdu_max,.19*Y_gdu_max
3490           FRAME
3500           LORG 2
```

```
3510            CSIZE 2.8,.6
3520            MOVE .08*X_gdu_max+4,.13*Y_gdu_max+4
3530            LABEL Stat_min$;
3540            MOVE .08*X_gdu_max+4,.13*Y_gdu_max+2
3550            LABEL Stat_max$;
3560            MOVE .08*X_gdu_max+60,.13*Y_gdu_max+4
3570            LABEL Stat_mean$;
3580            MOVE .08*X_gdu_max+60,.13*Y_gdu_max+2
3590            LABEL Stat_dev$;
3600 T_done:    PENUP
3610            RETURN
3620 !
3630 !
3640 Plot_axes:! SIZE/DRAW AXES ACCORDING TO HISTOGRAM VALUES
3650           VIEWPORT Plot_left,Plot_right,Plot_bottom,Plot_top
3660           WINDOW 0,Num_bins,0,MAX(Hist(*))    !SIZE/ASPECT RATIO OF GRAPH
3670           LINE TYPE 1
3680           X_maj=FNSel_ax(Num_bins)
3690           Xy_limits=MAX(Hist(*))
3700           Y_maj=FNSel_ax(Xy_limits)
3710           AXES X_maj/5,1,0,0,X_maj,Y_maj
3720           PENUP
3730           RETURN
3740!
3750 !
3760 !
3770 Label_axes:!SUBROUTINE TO PUT LABEL VALUES ON HISTOGRAM AXIS
3780             VIEWPORT Plot_left,Plot_right,Plot_bottom,Plot_top
3790             WINDOW Plot_left,Plot_right,Plot_bottom,Plot_top
3800             CLIP OFF
3810             CSIZE 3.5,.6
3820             LORG 4
3830             DEG
3840             LDIR 90
3850             MOVE Plot_left-.13*X_gdu_max,(Plot_top-Plot_bottom)/2+Plot_botto
m
3860             LABEL "Occurences"
3870             LDIR 0
3880             LORG 6
3890             MOVE (Plot_right-Plot_left)/2+Plot_left,Plot_bottom-.07*Y_gdu_ma
x
3900             LABEL Title1$&" Measurements (ns)"
3910             WINDOW 0,100,0,MAX(Hist(*))          !Y AXIS MUST MATCH GRAPH
3920             LORG 8
3930             FOR I=0 TO MAX(Hist(*)) STEP Y_maj  !PUT VALUES ON Y AXIS
3940             MOVE -.5,I                           !SMIDGEON TO THE LEFT
3950             LABEL USING "#,K";I
3960             NEXT I
3970             WINDOW 0,Num_bins,0,20               !X AXIS MUST MATCH GRAPH
3980             LORG 6
3990 !MIN X VALUE
4000             X_val=Stat_min
4010             X_pos=FNX_pos(X_val,Hist_value(*),Num_bins)
4020             X_value$="min"
4030             Value_flag=T
4040             GOSUB Label_it
4050 !MAX X VALUE
4060             X_val=Stat_max
4070             X_pos=FNX_pos(X_val,Hist_value(*),Num_bins)
4080             X_value$="max"
4090             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$="mean"
4140            GOSUB Label_it
4150            Line=7
4160            Descent=.006*Y_gdu_max
4170            X_offset=0
4180            GOSUB Draw_line
4190 !LOW STANDARD DEVIATION BOUNDARY
4200            X_val=Stat_dev
4210            X_pos=FNX_pos(PROUND((Stat_mean-X_val)*1.E+12,LGT(Resolution))*1
.E-12,Hist_value(*),Num_bins)
4220            Value_flag=F
4230            Line=4
4240            Descent=.003*Y_gdu_max
4250            IF Resolution<=10 THEN X_offset=-.004*X_gdu_max
4260            GOSUB Draw_line
4270 !HIGH STANDARD DEVIATION BOUNDARY
4280            X_pos=FNX_pos(PROUND((Stat_mean+X_val)*1.E+12,LGT(Resolution))*1
.E-12,Hist_value(*),Num_bins)
4290            Line=4
4300            Descent=.003*Y_gdu_max
4310            IF Resolution<=10 THEN X_offset=.004*X_gdu_max
4320            GOSUB Draw_line
4330            PENUP
4340            RETURN
4350 !
4360 !
4370 Label_it:  !PUT LABEL ON X AXIS (WITH VALUE)
4380            WINDOW 0,Num_bins,0,20    !X AXIS MUST MATCH GRAPH, Y IS FIXED
4390            CLIP OFF
4400            MOVE X_pos,-1.3
4410            LABEL X_value$
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 DECSENDER
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_offst=0
4630              Descent=0
4640              RETURN
4650 !
4660 !
4670 Plot_hist:   !SUBROUTINE TO PLOT THE HISTOGRAM OF THE DATA IN Tivalue.
4680              VIEWPORT Plot_left,Plot_right,Plot_bottom,Plot_top
```

```
4690              WINDOW 0,Num_bins,0,MAX(Hist(*))    !MUST MATCH GRAPH
4700              LINE TYPE 1
4710              MOVE -2,0
4720              FOR I=1 TO Num_bins
4730              MOVE I,0
4740              DRAW I,Hist(I)
4750              NEXT I
4760              PENUP
4770              RETURN
4780 !
4790 !
4800 Do_param:      ! ENTRY POINT FOR DISPLAY OF PARAMETER VALUES
4810                VIEWPORT .08*X_gdu_max,.92*X_gdu_max,.13*Y_gdu_max,.19*Y_gd
_max
4820                WINDOW .08*X_gdu_max,.92*X_gdu_max,.13*Y_gdu_max,.19*Y_gdu_
ax
4830                LINE TYPE 1
4840                LORG 2
4850                CSIZE 2.8,.6
4860                MOVE .08*X_gdu_max+24,.13*Y_gdu_max+4
4870                X_val=Stat_min
4880                GOSUB Draw_param
4890                MOVE .08*X_gdu_max+24,.13*Y_gdu_max+2
4900                X_val=Stat_max
4910                GOSUB Draw_param
4920                MOVE .08*X_gdu_max+85,.13*Y_gdu_max+4
4930                X_val=Stat_mean
4940                GOSUB Draw_param
4950                MOVE .08*X_gdu_max+85,.13*Y_gdu_max+2
4960                LABEL USING "#,DDD.DDD";Stat_dev*1.E+9
4970                PENUP
4980                RETURN
4990 !
5000 Draw_param:  !DRAW PARAMETERS WITH PROPER RESOLUTION
5010              SELECT INT(LGT(Resolution))
5020                CASE =0
5030                  LABEL USING "#,DDD.DDD";X_val*1.E+9
5040                CASE =1
5050                  LABEL USING "#,DDD.DD";X_val*1.E+9
5060                CASE =2
5070                  LABEL USING "#,DDD.D";X_val*1.E+9
5080                CASE =3
5090                  LABEL USING "#,DDD";X_val*1.E+9
5100              END SELECT
5110              PENUP
5120              RETURN
5130 !
5140 !
5150 !******************RESOLUTION AND SAMPLE SIZE SUBROUTINES **************
5160 !
5170 Fix_resolution: !SUBROUTINE TO CHANGE THE MEASUREMENT RESOLUTION
5180                 IF Auto THEN Resolution=Dresolution
5190                 IF Auto THEN Fr_done
5200                 PRINT USING "@"
5210                 GCLEAR
5220                 PRINT "Enter the desired resolution of the display"
5230                 PRINT "in ps."
5240                 Res_entry(Resolution)
5250                 PRINT USING "@"
5260                 PRINT "Disp. Res.=";Resolution;"ps","Sample Size=";Num_samp
les
```

```
5270                RETURN
5280 !
5290 !
5300 Fix_samples:!SUBROUTINE TO SET/ADJUST THE NUMBER OF SAMPLES TAKEN FOR THE
5310              !HISTOGRAM PLOT
5320              IF Auto THEN Num_samples=Dnum_samples
5330              IF Auto THEN Fs_done
5340              PRINT USING "@"
5350              GCLEAR
5360              PRINT "Enter the number of samples to be taken."
5370              PRINT "NOTE: The max value shown below is a typical"
5380              PRINT "      maximum based upon the amount of memory"
5390              PRINT "      presently installed in the computer,"
5400              PRINT "      with the number of samples set equal to"
5410              PRINT "      the number of histogram bins. "
5420              PRINT "      The actual maximum may be lower or"
5430              PRINT "      higher depending upon the range of"
5440              PRINT "      samples taken and the displayed"
5450              PRINT "      resolution value chosen."
5460              PRINT
5470              Samp_entry(Num_samples)
5480              PRINT USING "@"
5490              PRINT "Disp. Res.=";Resolution;"ps","Sample Size=";Num_samples
5500              RETURN
5510 !
5520 !
5530 !******************** PLOTTER CONTROL SUBROUTINES ************************
5540 !
5550 Ext_plot:   GINIT
5560             PLOTTER IS Plotter_addr,"HPGL"
5570             Minx=1
5580             Maxx=1
5590             Stepx=1
5600             RETURN
5610 !
5620 Int_plot:   GINIT
5630             PLOTTER IS 3,"INTERNAL"
5640             Maxx=.2
5650             Minx=-Maxx
5660             Stepx=.1
5670             RETURN
5680 !
5690 !
5700 !******** FUNCTION KEY LABELS AND FLAG SUBROUTINES ***********************
5710 !
5720 Auto_demo: Auto=NOT (Auto)
5730             SELECT Auto
5740               CASE =F
5750                 ON KEY 7 LABEL " DEMO ON",13 GOSUB Auto_demo
5760                 GOTO Auto_done
5770               CASE =T
5780                 ON KEY 7 LABEL "AUTODEMO ON",13 GOSUB Auto_demo
5790             END SELECT
5800 Set_auto_param:   Res_entry(Pw_auto_res,"SET RESOLUTION OF THE PULSE WIDTH
HISTOGRAM")
5810                   Samp_entry(Pw_auto_samp,"SET SAMPLE SIZE FOR THE PULSE WI
DTH HISTOGRAM")
5820                   Res_entry(Rt_auto_res,"SET RESOLUTION OF THE RISE TIME HI
STOGRAM")
5830                   Samp_entry(Rt_auto_samp,"SET SAMPLE SIZE FOR THE RISE TIM
E HISTOGRAM")
```

```
5840                    Res_entry(Ft_auto_res,"SET RESOLUTION OF THE FALL TIME HI
STOGRAM")
5850                    Samp_entry(Ft_auto_samp,"SET SAMPLE SIZE FOR THE FALL TIM
E HISTOGRAM")
5860                    Contin=False
5870                    GOSUB Single_cont
5880 Do_auto:           GOSUB P_width
5890                    GOSUB R_time
5900                    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$(SYSTEM$("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 a"
6250               PRINT "histogram is generated showing measured time"
6260               PRINT "vs. number of occurances."
6270               PRINT
6280               PRINT "If not already done, be sure to calibrate the"
6290               PRINT "5363B probes to ensure accurate TI 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 "TI Probe's HP-IB address ? (default=707)",Probe_addr
6340               PRINT TABXY(41,5);"HP-IB=";Probe_addr
6350               Id$=" graph title "
6360               INPUT "Do you plan on using an HPGL PLOTTER (Y/N)?",Answer$
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               Plot_yes=1
```

```
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=";Dresolutio
n
6730                PRINT
6740                IF NPAR=1 THEN Get_res
6750                PRINT A_res$
6760 Get_res:       Resolution$=""
6770                LINPUT Resolution$
6780                Resolution$=UPC$(TRIM$(Resolution$))
6790                SELECT Resolution$
6800                  CASE =""
6810                    Resolution=0
6820                    GOTO Set_res
6830                  CASE ="MIN"
6840                    Resolution=Mresolution
6850                    GOTO Fr_done
6860                  CASE ="MAX"
6870                    Resolution=9999
6880                    GOTO Set_res
6890                END SELECT
6900                Resolution=VAL(Resolution$)
6910 Set_res:       IF Resolution<=0 THEN Resolution=Dresolution
6920                IF Resolution<1 THEN 5190
6930                IF Resolution>9999 THEN Resolution=1000
6940 Fr_done:  Resolution=10^INT(LGT(Resolution))  !INTEGER POWERS OF 10 ONLY
6950           DEALLOCATE Resolution$
6960           SUBEND
6970 !
6980 !                        **********************
6990 !
7000 SUB Samp_entry(INTEGER Num_samples,OPTIONAL A_samp$)
```

```
7010 COM /Sample/ INTEGER Max_samples,Dnum_samples,Dresolution,Mresolution
7020     ALLOCATE Num_samples$[20]
7030             PRINT "max=";Max_samples;", default=";Dnum_samples
7040             PRINT
7050             IF NPAR=1 THEN Get_samp
7060             PRINT A_samp$
7070 Get_samp:   Num_samples$=""
7080             LINPUT Num_samples$
7090             Num_samples$=UPC$(TRIM$(Num_samples$))
7100             SELECT Num_samples$
7110               CASE =""
7120                 Num_samples=0
7130                 GOTO Set_samp
7140               CASE ="MIN"
7150                 Num_samples=1
7160                 GOTO Fs_done
7170               CASE ="MAX"
7180                 Num_samples=Max_samples
7190                 GOTO Fs_done
7200             END SELECT
7210             Num_samples=VAL(Num_samples$)
7220             IF Num_samples>Max_samples THEN
7230             BEEP
7240             DISP " Entered value (";Num_samples;") is > max"
7250             WAIT 2
7260             Num_samples=Max_samples
7270             END IF
7280 Set_samp:   IF Num_samples<=0 THEN Num_samples=Dnum_samples
7290 Fs_done:    DEALLOCATE Num_samples$
7300             PRINT USING "@"
7310             SUBEND
7320 !
7330 !                     ************************
7340 !
7350 SUB Convert_ti(Tidata$(*),REAL Tivalue(*),INTEGER Array_size,Res)
7360 !
7370 !Converts raw TI data in Tidata$(*) to REAL time interval values, trims to
     (Res) resolution, and passes back as the REAL array Tivalue(*).
7380 !
7390               INTEGER I
7400               Const1=2^16
7410               Const2=2^8
7420               Const3=2^17
7430               Const4=2^18
7440               FOR I=1 TO Array_size
7450               N=NUM(Tidata$(I)[4,4])*256+NUM(Tidata$(I)[5,5])
7460               Q=1
7470               IF NOT BIT(INT(NUM(Tidata$(I)[1])),5) THEN Q=-1
7480               B=BINAND(NUM(Tidata$(I)[1]),3)*Const1+NUM(Tidata$(I)[2,2])*Co
nst2+NUM(Tidata$(I)[3,3])
7490               IF B>=Const3 THEN B=B-Const4
7500               Counter_ti=(B/256+N*Q)*5.E-9 !5370B 'FAST BINARY' TI EQUATION
7510               Tivalue(I)=PROUND((Counter_ti)*1.E+12,LGT(Res))/1.E+12
7520 Next_samp:    NEXT I
7530               SUBEND
7540 !
7550 !                     *********************
7560 !
7570 DEF FNX_pos(Value,Array(*),INTEGER Array_size)
7580 !Return the exact or next highest bin number belonging to Value in Array(*)
```

```
7590            INTEGER Curr_pos,Next_pos,Low_limit,High_limit
7600            Low_limit=0
7610            High_limit=Array_size+1
7620            Curr_pos=INT((Array_size/2)+.51)
7630 Another_x:   SELECT Value
7640                CASE =Array(Curr_pos)
7650                  GOTO Found_x
7660 !
7670                CASE <Array(Curr_pos)
7680                  High_limit=Curr_pos
7690                  IF Curr_pos>1 THEN
7700                  IF Value>Array(Curr_pos-1) THEN Found_x
7710                  END IF
7720                  Curr_pos=Curr_pos-INT((High_limit-Low_limit)/2)
7730                  IF Curr_pos>1 THEN Another_x
7740                  Curr_pos=1
7750                  GOTO Found_x
7760 !
7770                CASE >Array(Curr_pos)
7780                  Low_limit=Curr_pos
7790                  IF Curr_pos<Array_size AND Value<Array(Curr_pos+1) THEN Co
rrect_x
7800                  Curr_pos=Curr_pos+INT((High_limit-Low_limit)/2)
7810                  IF Curr_pos<Array_size THEN Another_x
7820                  Curr_pos=Array_size
7830                  GOTO Found_x
7840              END SELECT
7850 Correct_x: Curr_pos=Curr_pos+1
7860 Found_x:   RETURN Curr_pos
7870            FNEND
7880 !
7890 !                              ***********************
7900 !
7910 SUB Fill_hist(REAL Sample_array(*),Range(*),INTEGER S_array_size,Hist(*),H_
array_size)
7920 !
7930 !Sample_array(*) is a real array of <S_array_size> size,
7940 !containing MAT SORT(ed) values, low to high.
7950 !The Range(*) array contains the quantized values for the histogram
7960 !(ie,x axis) to which the samples in Sample_array must be matched.
7970 !The Hist(*) array is filled by this routine according to the number
7980 !of samples in Sample_array which fit each value of Range(*).
7990 !
8000              INTEGER I,Bin_number
8010              Bin_number=1
8020              FOR I=1 TO S_array_size
8030 Try_again:   ! COUNT SAMPLE INTO FIRST BIN THAT IT CROSSES THE THRESHOLD OF
8040                IF Sample_array(I)<Range(Bin_number) THEN
8050                  Hist(Bin_number)=Hist(Bin_number)+1
8060                    GOTO Next_tivalue
8070                END IF
8080                Bin_number=Bin_number+1
8090                IF Bin_number>H_array_size THEN Next_tivalue ! IN CASE ALL
                                                                   ZERO'S
8100                GOTO Try_again
8110 !
8120 Next_tivalue: NEXT I
8130              SUBEND
8140 !
8150 !                              ***********************
8160 !
```

```
8170 DEF FNSel_ax(INTEGER Axis_limit)
8180          INTEGER Xy_maj
8190          Xy_maj=1
8200          SELECT Axis_limit  !GET CORRECT MAJOR LABEL
8210            CASE >5000
8220              Xy_maj=1000
8230            CASE >1000
8240              Xy_maj=500
8250            CASE >500
8260              Xy_maj=100
8270            CASE >100
8280              Xy_maj=50
8290            CASE >50
8300              Xy_maj=10
8310            CASE >10
8320              Xy_maj=5
8330          END SELECT
8340          RETURN Xy_maj
8350          FNEND
8360 !
8370 !                          ************************
8380 !
8390 SUB Compute_stat(Tivalue(*),INTEGER Array_size,Res,REAL Stat_min,Stat_max,S
tat_mean,Stat_dev)
8400 !
8410 !Compute min, max, mean, and standard deviation values of Tivalue(*),
      trim to (Res) resolution, and pass back.
8420 !
8430               INTEGER I
8440               REAL Std_dev,C
8450               Mean=0
8460               Std_dev=0
8470               A=0
8480               B=0
8490               Stat_min=Tivalue(1)
8500               Stat_max=Tivalue(1)
8510               FOR I=1 TO Array_size
8520               IF Tivalue(I)<Stat_min THEN Stat_min=Tivalue(I)
8530               IF Tivalue(I)>Stat_max THEN Stat_max=Tivalue(I)
8540               A=A+Tivalue(I)
8550               B=B+(Tivalue(I)*Tivalue(I))
8560               NEXT I
8570               Mean=A/Array_size
8580               C=Array_size
8590               Std_dev=SQR(ABS((Array_size*B-A*A)/(C*(C-1))))
8600               Stat_mean=PROUND(Mean*1.E+12,LGT(Res))*1.E-12
8610               Stat_dev=PROUND(Std_dev*1.E+12,0)*1.E-12
8620               SUBEND
8630 !
8640 ! ************************ 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
      Atten . . . . ÷ 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 plotters 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 quatization 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* (eg, 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 ¼ 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 327.68000 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 (ns) = ((integer number of times the counter will overflow) x overflow value)) + (current count in the counter)
= ((expected TI, in μs) / (327.68000μs))* x (327680.00ns) + ((N1N2)/256 + N0) x 5ns

Example:

The time interval to be measured is about 5ms (probably within 100μs);

TI (ns) = (5ms / 327.68000μs)* x (327680.00ns) + xxxxxx.xxns
= (15) x (327680.00ns) + xxxxxx.xxns
= $4.91520000 \times 10^{-3}$ + xxxxxx.xxns,

(where xxxxxx.xx 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 counters 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.

HEWLETT PACKARD

For more information, call your local HP Sales Office or nearest Regional Office: **Eastern** (201) 265-5000; **Midwestern** (312) 255-9800; **Western** (213) 970-7500; **Canadian** (416) 678-9430. Ask the operator for instrument sales. Or write Hewlett-Packard, 1501 Page Mill Road, Palo Alto, CA 94304. **In Europe:** Hewlett-Packard S.A., 7, Rue du Bois-du-Lan, P.O. Box, CH-1217 Meyrin 2, Geneva, Switzerland. **In Japan:** Yokogawa-Hewlett-Packard Ltd., 29-21, Takaido-Higashi 3-chome, Suginami-ku, Tokyo 168.

02-5952-7769

PRINTED IN U.S.A.