Time Markers	•Off
Start Marker Position	••3.5us
Stop Marker Position	•+3.5us
Start Marker Edge Slope	•Positive
Stop Marker Edge Slope	•Negative
Start Marker Edge Number	•1
Stop Marker Edge Number	•1
Waveform Memories	•Off
SOURce for Store	•Channel 1
Selected Memory	•Memory 1
Pixel memories	•Off
Waveform Data	•0V
Pattern Duration	•10.0ns
Pattern (pattern edge mode)	•Ch1; Clock •Ch2; Dontcare •Trig3; Dontcare •Trig4; Dontcare
Pattern Present/Not Present	•Present
Holdoff Time (pattern edge mode)	•70.0ns
Arming Slope (time delayed mode)	•Negative
Arming Channel (time delayed mode)	•Channel 1
Time Delay (time delayed mode)	•20.0ns
Trigger Slope (time delayed mode)	•Positive
Trigger Channel(time delayed mode)	•Channel 1
Arming Slope (event delayed mode)	•Negative
Arming Chan. (event delayed mode)	•Channel 1
Events Delay (event delayed mode)	•1
Trigger Slope (event delayed mode)	•Positive
Trigger Chan.(event delayed mode)	•Channel 1
Functions 1&2	•OFF
Functions (definition)	•(Chan 1 - Chan 2)
Functions (volts/div)	•2.0 volt
Functions (offset)	•0.0 volts

RESET VALUES FOR THE HP-IB FOR THE 54100A/D

Service Request Mode	•Disabled (RQS OFF)
Service Request Mask	•Decimal 32546
Serial Poll Status Byte	•Clear
Error Queue	•Empty
WAVeform Format	•WORD
EOI	•ON
LONGform	•OFF
HEADer	•OFF
ARGument	•NUMERIC

Table 10-5. Reset Conditions

SAVE

command

This command saves an instrument setup in the specified save/recall register. It's action is the same as performing a SAVE operation from the front panel.

Command Syntax: SAVE[REGISTER]<d>

```
Example:  OUTPUT 707;"SAVE1"
```

SERial?

query

This query returns the instrument serial number as a quoted string.

Query Syntax: SERial?

```
Example:  OUTPUT 707:"SER?"
          ENTER 707; Ser$
          PRINT Ser$
```

SETup

command/query

This command sets up the 54100A/D according to the learn string. The query returns the learn string from the oscilloscope.

Command Syntax: SETup

```
Example:  OUTPUT 707;"SETUP ";Set$
```

Query Syntax: SETup?

```
Returned Format:  [SETup]<block type A>

Example:  DIM Set$[270]
          OUTPUT 707;"HEADER OFF EOI ON"
          OUTPUT 707;"SETUP?"
          ENTER 707 USING "-K";Set$
          OUTPUT 707;"SETUP ";Set$
```

NOTE

The logical order for this instruction would be to send the query first followed by the command at a time of your choosing. The query causes the learn string to be sent to the controller and the command causes the learn string to be returned to the 54100A/D.

SPOLl? | STB?

query

This query returns the status byte (the lower byte of the status word). This command is similar in operation to conducting a serial poll from the controller except that all bits in the byte returned by this query are dynamic and reflect the state of the instrument at the time of the query. Bits in the byte returned by a serial poll stay set if the require service message was sent and are cleared after a serial poll. This command is provided for use by controllers that have a limited HP-IB control capability. Using the serial poll is the preferred method of reading the status byte.

Query Syntax: { SPOLl | STB }?

```
Example:  OUTPUT 707;"STB?"
          ENTER 707;Stb$
          PRINT Stb$
```

STAtus

query

This query returns the instrument status word. The instrument status word is a 16-bit word which is returned as an integer, and contains information about the instrument conditions that set the ready bit in the status byte and/or generate a Require Service message. The upper 8 bits of the status word are known collectively as the ready byte, while the lower 8 bits correspond to the status byte sent during a serial poll. The STATus query is used to read the status word representing the current status of the 54100A/D. Unlike the response to serial poll, the conditions are dynamic, not latched. Therefore the status response reflects the current status. See paragraph 10-8 and Tables 10-1 and 10-2 for a complete description of the instrument status word.

A companion 16 bit word, the request mask, is used to specify both those conditions in the ready byte that set the ready bit in the status byte, and those conditions in the status byte that generate a Require Service message. The bits in the request mask have the same meanings as those in the instrument status word. The ready bit in the status byte is set when all of the conditions corresponding to the bits in the ready mask are true at the same time. This bit is actually set on the transition of the last condition to become true. The REQuest system command is used to specify the request mask.

Query Syntax: STATus?

```
Example:  OUTPUT 707;"STA?"
          ENTER 707;Sta$
          PRINT Sta$
```

STOP

command

This command causes the instrument to stop acquiring data for the active display on the CRT. The RUN command must be executed in order to restart the acquisition.

Command Syntax: STOP

Example: OUTPUT 707;"STOP"

STORe

command

This command allows you to move stored waveforms from one place to another internal to the instrument. This command has two parameters. The first is the source of the waveform which can be Channel 1 | 2, Function 1 | 2, or Memory 1 | 2 | 3 | 4. The second parameter is the destination of the waveform which can be Memory 1 | 2 | 3 | 4.

Command Syntax:
```
STORe {[CHANnel 1 | 1]
       [CHANnel 2 | 2 ]
       [FUNCtion 1 | 9 ]
       [FUNCtion 2 | 10 ]
       [MEMory 1 | 11 ]
       [MEMory 2 | 12 ]
       [MEMory 3 | 13 ]
       [MEMory 4 | 14 ]}<,>
      {[MEMory 1 | 11 ]
       [MEMory 2 | 12 ]
       [MEMory 3 | 13 ]
       [MEMory 4 | 14 ]}
```

Example: OUTPUT 707;"STORE CHANNEL2,MEMORY4"

TEST | TST

command

This command causes the instrument to perform a self-test. This is the same test that is executed when the instrument is powered up. The Tst bit in the Status Word (bit 13) will go to a 1 when the test is complete.

Command Syntax: {TEST | TST}

Example: OUTPUT 707;"TEST"

TRANsfer | XFER

command

This command allows the movement of waveform data from one of the waveform memories to one of the pixel memories. This command has two parameters; the first parameter is MEMORYn where n=1 through 4 and designates the source of the data as waveform memory 1,2,3, or 4, the second parameter is PLANEn where n=1 or 2 and designates the destination of the data as pixel memory 5 or 6.

If one of the pixel memories contains data and new data is written to that memory, the new data will be superimposed on the existing data.

NOTE

When using this command only the pixel data is transferred i.e., the waveform parameters are lost.

Command Syntax: TRANsfer [memory]<waveform memory #>,
[plane]<plane #>

<waveform memory #> ::= { 1 | 2 | 3 | 4 }
<plane #> ::= { 1 | 2 }

Example: OUTPUT 707;"TRANSFER MEMORY 3,PLANE 2"

TRG | GET

command

The instrument responds to this command in the same way it responds to the RUN system command and the GET bus command, (paragraph 9-16).

This command causes the instrument to acquire data for the active waveform display based on the timebase mode. If the time base mode is in SINGLE, the TRG command will cause the instrument to enable the trigger once, and display this data on the CRT. This is the same thing that happens when you press the front panel STOP\SINGLE key when the instrument has STOPPED.

If the timebase mode is AUTO or TRIGGERED, the TRG command will cause the instrument to enable the trigger repeatedly and display the acquired data on the CRT. This is the same thing that happens when you press the front panel RUN key. See MODE under the TIMEBASE subsystem in paragraph 10-13.

Command Syntax: {TRG | GET}

Example: OUTPUT 707;"TRG"

VIEW

command

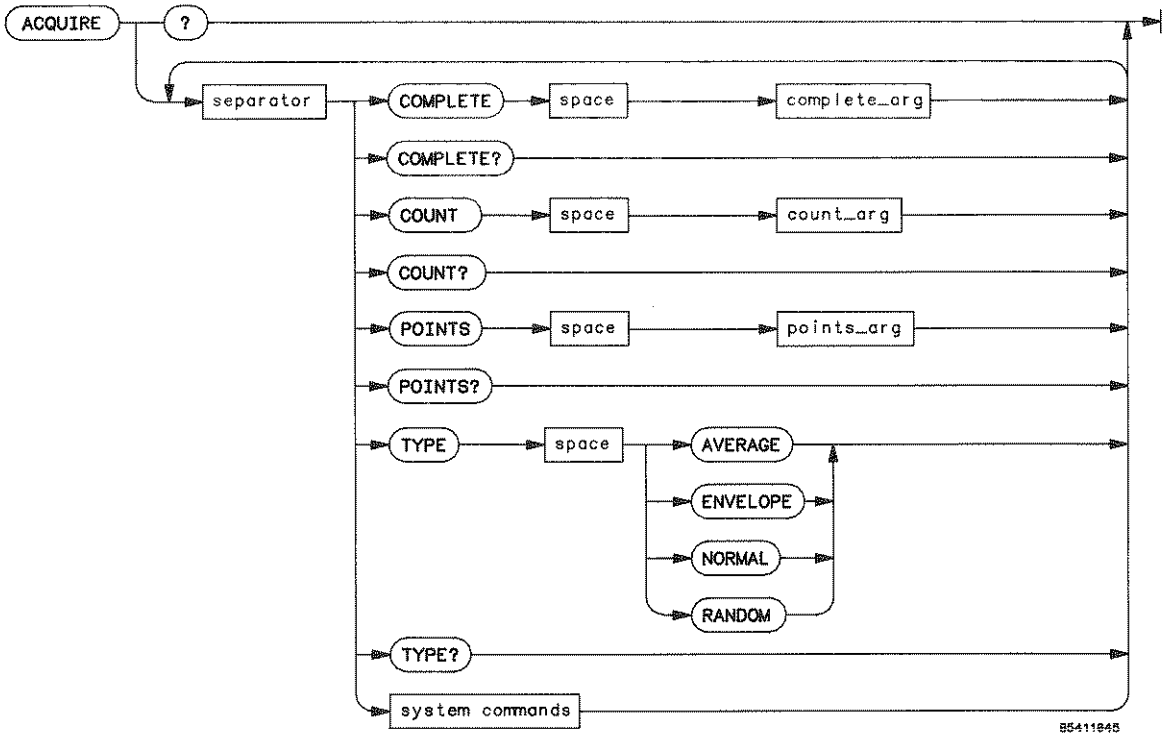
The VIEW command causes the instrument to turn on, (start displaying), an active channel, function, pixel memory or waveform memory. If you want to turn on an active display use the parameter **Channel 1|2.** If you want to turn on a pixel memory use the parameter **PLANE 1 | 2,** where plane 1 = pixel memory 5 and plane 2 = pixel memory 6. Using the VIEW MEMory { 1 | 2 | 3 | 4 } command in the split screen mode causes memories 1 and 3 to be displayed on the upper screen and memories 2 and 4 to be displayed on the lower screen.

Command Syntax: VIEW {[CHANnel { 1 | 2 }]
[PLANe { 1 | 2 }]
[FUNCtion { 1 | 2 }]
[MEMory { 1 | 2 | 3 | 4 }]}

Example: OUTPUT 707;"VIEW CHANNEL 1"

NOTES:

NOTES:



COMPLETE_ARG ::= An integer from 0 to 100, specifying, in percent, the number of buckets that must be filled before acquisition is considered complete.

COUNT_ARG ::= An integer from 1 to 2048 specifying the number of values to average for each point when in the averaged mode, and the number of values to use for each point when constructing the envelope.

POINTS_ARG ::= An integer specifying the number of points to be collected for each waveform record. Acceptable values are 128, 256, 500, 512, or 1024.

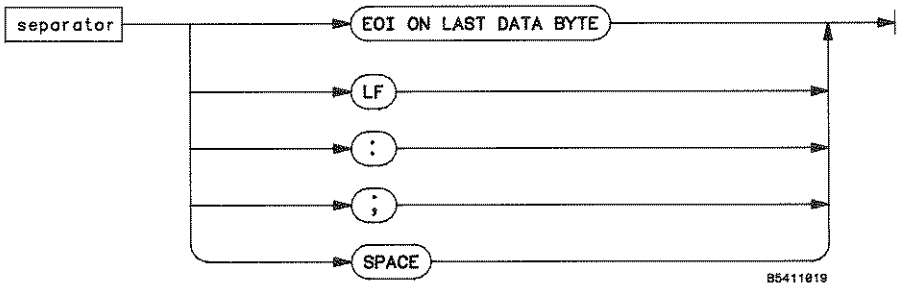


Figure 10-3. Acquire Subsystem Commands.

10-10 THE ACQUIRE SUBSYSTEM

The Acquire subsystem commands are used to setup conditions that are used when a DIGITIZE system command is executed. This subsystem is used to select the type of data, the number of points desired, the completion criteria, and the number of averages. See figure 10-3.

ACQuire

command/query

The ACQuire command selects the acquire subsystem as the destination for the commands that follow.

The ACQuire query responds with the settings of the acquire subsystem.

Command Syntax: ACQuire

Example: OUTPUT 707;"ACQUIRE"

Query Syntax: ACQuire?

Returned Format:
```
[ ACQuire <crlf> ]
[ TYPE ]<argument><crlf>
[ POINts ]<NR1>crlf>
[ COUNts ]<NR1><crlf>
[ COMPlete ]<NR1><crlf>
```

Example:
```
DIM Acquire$[70]
OUTPUT 707;"EOI ON"
OUTPUT 707;"ACQUIRE?"
ENTER 707 USING "-K";Acquire$
PRINT USING "K";Acquire$
```

COMPlete

command/query

This command specifies the completion criteria for an acquisition. The parameter determines what percentage of the time buckets need to be "full" before an acquisition is considered completed. If you are in the NORMAL mode the instrument only needs one data point in a time bucket to be considered full. In order for a time bucket to be considered full in the AVERAGED and ENVELOPE modes a specified number of data points (COUNT) must be acquired.

The parameter for this command has a range of 0 to 100 and indicates the the percentage of time buckets that have the required number of data points i.e., are considered "full" before the acquisition is considered complete. When the completion criteria is set to 0, only a single acquisition cycle will be performed, except for sweep speeds below 2.5 ns/div where either 1 or no acquisition cycles may be performed.

Command Syntax: COMPlete <NR1>

```
Example:  OUTPUT 707;"COMPLETE 85"
```

Query Syntax: COMPlete?

```
Returned Format:  [COMPlete] <NR1><crlf>

Example:  OUTPUT 707;"COMPLETE ?"
          ENTER 707;Complete$
          PRINT Complete$
```

COUNt | CNT

command/query

When the acquisition type is AVERAGE, this command specifies the number of values to be averaged for a particular time bucket before acquisition is considered complete for that bucket. When the acquisition type is ENVELOPE, this command specifies the number of values to be used in each time bucket when constructing the envelope. This command has no effect if the TYPE is NORMAL or RANDOM. The query returns the last specified count value. The COUNt parameter can be an integer from 1 to 2048.

Command Syntax: {COUNt | CNT} <NR1>

```
Example:  OUTPUT 707;"COUNT 1854"
```

Query Syntax: {COUNt | CNT}?

```
Returned Format:  [COUNt | CNT]<NR1><crlf>

Example:  OUTPUT 707;"CNT?"
          ENTER 707;Cnt$
          PRINT Cnt$
```

POINts | PNTS

command/query

This command specifies the number of points for each acquisition record. The command has one parameter and may be specified to be 128, 256, 500, 512 or 1024. 500 points is preferred if the acquired data is to be used for automatic measurements or function operands. There are two cases where the POINTS command has no affect:

For sweep speeds faster than 2 ns/div., the number of points is based on 10 ps resolution of the instrument's timebase. This means:

If (2.0 ns/div)>(time per div)>=(1.0 ns/div) then POINTS = 1000(500 if selected)
If (1.0 ns/div)>(time per div)>=(500 ps/div) then POINTS = 500
If (500 ps/div)>(time per div)>=(200 ps/div) then POINTS = 200
If (200 ps/div)>(time per div)>=(100 ps/div) then POINTS = 100

If the TYPE is RANDOM, the number of points is based on the number of complete data records (the points collected on each trigger) that can be gathered (cannot exceed 1024). The data acquisition hardware allows the data points gathered after a trigger event to vary with time per division and delay. It can also vary by one data point from one trigger event to the next. This makes it difficult to predict the number of points that will be gathered for any DIGITIZE command when the TYPE is RANDOM. Before the data is read from the instrument with the WAVEFORM DATA? query, the WAVEFORM POINTS? query may be used to determine the actual number of points collected.

The POINTS? query returns the last specified value.

Command Syntax: {POINts | PNTS}

```
Example:  OUTPUT 707;"PNTS 128"
```

Query Syntax: {POINts | PNTS}?

```
Returned Format:  [POINts] <NR1><crlf>

Example:  OUTPUT 707;"POINTS?"
          ENTER 707;Points$
          PRINT Points$
```

TYPE

command/query

This command lets you select the type of acquisition that is to take place when a DIGITIZE system command is executed. This command has one parameter and may be one of the following:

NORMAL

Last data value to be collected in each acquisition bucket. The data is returned to the controller as a series of voltage values that represent the evenly spaced data points on the CRT.

RANDOM

The Random mode simulates the way the instrument collects data for display on the CRT. This data is returned to the controller as a list of time-voltage pairs.

AVERAGE

The average of the data values collected in each acquisition bucket. The data is returned to the controller as a series of voltage values that represent the evenly spaced data points on the CRT.

ENVELOPE

The max and min value in each acquisition bucket. The data is returned to the controller as two lists of voltage values, the min values first then the max values.

When you change TYPE to AVERAGE the front panel display mode is changed to averaged. Changing TYPE to NORMAL, ENVELOPE, or RANDOM switches the front panel display mode to Normal.

Command Syntax: TYPE {[NORMAL | 1]
[AVERAGE | 2]
[ENVELOPE | 3]
[RANDOM | 4]}

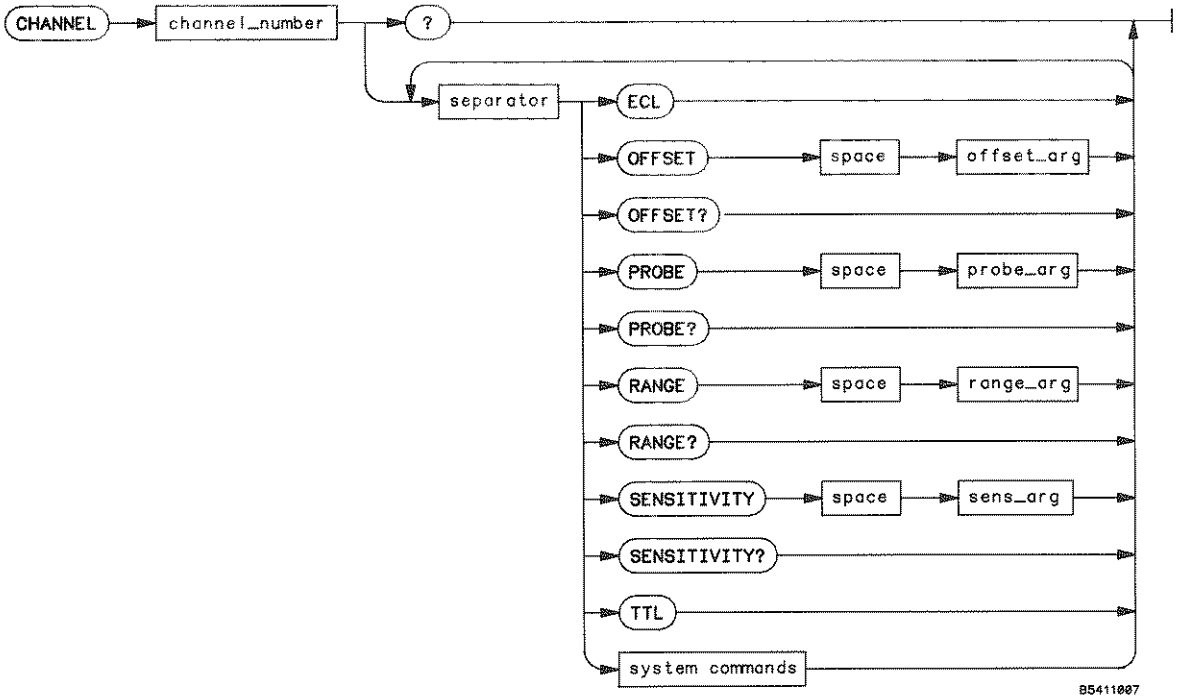
Example: OUTPUT 707;"ACQUIRE; TYPE RANDOM"

Query Syntax: TYPE?

Returned Format: [TYPE]<argument><crlf>

Example: OUTPUT 707;"TYPE?"
ENTER 707;Type$
PRINT Type$

NOTES:



CHANNEL_NUMBER = 1 or 2

OFFSET_ARG = A real number defining the voltage at the center of the voltage range smaller than 1.5 X voltage range.

PROBE_ARG = A real number from 1.0 to 1000.0 specifying the probe attenuation with respect to 1.

RANGE_ARG = A real number specifying the size of the acquisition window in volts. Acceptable values are 0.08, 0.16, 0.4, 0.8, 1.6, 4.0, 8.0. (With the probe attenuation ration set at 1:1)

SENS_ARG = A real number specifying the size of the acquisition window in volts/div. Acceptable values are 0.01, 0.02, 0.05, 0.1, 0.5, or 1.0 when the split screen display format is off. When the unit is in the split screen mode the acceptable values are: 0.02, 0.04, 0.1, 0.2, 0.4, 1.0, 2.0. (With the probe attenuation ratio set at 1:1)

Figure 10-4. Channel Subsystem Commands

10-11. CHANNEL SUBSYSTEM

The CHANNEL subsystem allows you to control the two input channels on the 54100A/D. Channel 1 and channel 2 are independently programmable for all functions. See Figure 10-4.

CHANnel | CH

command/query

This command allows you to select the vertical subsystem with the specified channel designated as the destination for the subsystem commands. The query responds with all the settings for the specified channel.

Command Syntax: {CHANnel | CH} { 1 | 2 }

```
Example:  OUTPUT 707;"CHANNEL 1"
```

Query Syntax: {CHANnel | CH} { 1 | 2 }?

```
Returned Format:  [CHANnel | CH]<NR1><crlf>
                  [PROBe]<NR3><crlf>
                  [RANGe]<NR3><crlf>
                  [OFFSet]<NR3><crlf>
                  [COUPling]<DC><crlf>

Example:  DIM Chan$[100]
          OUTPUT 707;"EOI ON"
          OUTPUT 707;"CHANNEL 2?"
          ENTER 707 USING "-K";Chan$
          PRINT USING "K";Chan$
```

ECL

command

This command sets the vertical range and offset and the trigger level for the selected channel for optimum viewing of ECL signals. The offset and trigger level are set to -1.30 volts and the range will be set to 1.6 volts.

Command Syntax: ECL

```
Example:  OUTPUT 707;"ECL"
```

OFFSet

command/query

This command allows you to set the voltage that is represented at center screen for the selected channel. The range of OFFSet is ± 1.5X RANGe of the selected channel.

Command Syntax: OffSet <OFFSET_ARG>

```
Example:  OUTPUT 707;"OFFSET 650E-3"
```

Query Syntax: OFFSet?

```
Returned Format:  [OFFSet] <NR3><crlf>

Example:  OUTPUT 707;"OFFSET?"
          ENTER 707;Offset$
          PRINT Offset$
```

PROBe

command/query

This command allows you to specify the probe attenuation factor for the selected channel. The range of the probe attenuation factor is from 1.0 to 1000.0. This command does not change the actual input sensitivity of the 54100A/D, it changes the reference constants that are used for scaling the display factors and for automatic measurements, trigger levels, etc.

Command Syntax: PROBe <PROBE_ARG>

```
Example:  OUTPUT 707;"PROBE 15.5"
```

Query Syntax: PROBe ?

```
Returned Format:  [PROBe]<NR3><crlf>

Example:  OUTPUT 707;"PROBE?"
          ENTER 707;Probe$
          PRINT Probe$
```

RANGe

command/query

This command allows you to define the full scale vertical axis of the selected channel. If you use a 1:1 probe attenuation factor the acceptable values for RANGe are: 0.08, 0.16, 0.4, 0.8, 1.6, 4.0, and 8.0. These values represent the full scale deflection factor of the vertical axis in volts. These values change as the probe attenuation factor is changed, e.g., if the probe attenuation factor is changed from 1:1 to 10:1 the Maximum RANGe value changes from 8 to 80 volts full scale. The query returns the current range setting.

Command Syntax: RANGe <RANGE_ARG>

Example:
```
OUTPUT 707;"RANGE 4"
```

Query Syntax: RANGe?

Returned Format: [RANGe]<NR3><crlf>

Example:
```
OUTPUT 707;"RANGE?"
ENTER 707;Range$
PRINT Range$
```

SENSitivity

command/query

This command allows you to specify the vertical deflection in volts/division as opposed to volts full scale as specified with the RANGe command. With the probe attenuation ratio set to 1:1 the allowable values for SENSitivity when you are using the single display format are 0.010, 0.020, 0.050, 0.100, 0.200, 0.500, and 1.000. All of these values represent volts/vertical division when using the grid graticule. The SENSitivity command takes the probe attenuation ratio into account so the SENSitivity value programmed should be the desired sensitivity at the probe tip. Using the RANGe command is a safer way of programming the vertical as it is independent of the DISPLAY FORMAT command i.e., (single or dual). The query returns the current sensitivity setting.

Command Syntax: SENSitivity <SENS_ARG>

Example:
```
OUTPUT 707;"SENSITIVITY 1"
```

Query Syntax: SENSitivity ?

Returned Format: [SENSitivity] <NR3><crlf>

Example:
```
OUTPUT 707;"SENS?"
INPUT 707;Sens$
PRINT Sens$
```

TTL

command

This command sets the vertical range and offset and the trigger level for the selected channel for optimum viewing of TTL signals. Offset and trigger level will be set to 1.6 volts and the range will be set to 8.0 volts.

Command Syntax: TTL

```
Example:  OUTPUT 707;"TTL"
```

NOTES:

NOTES:

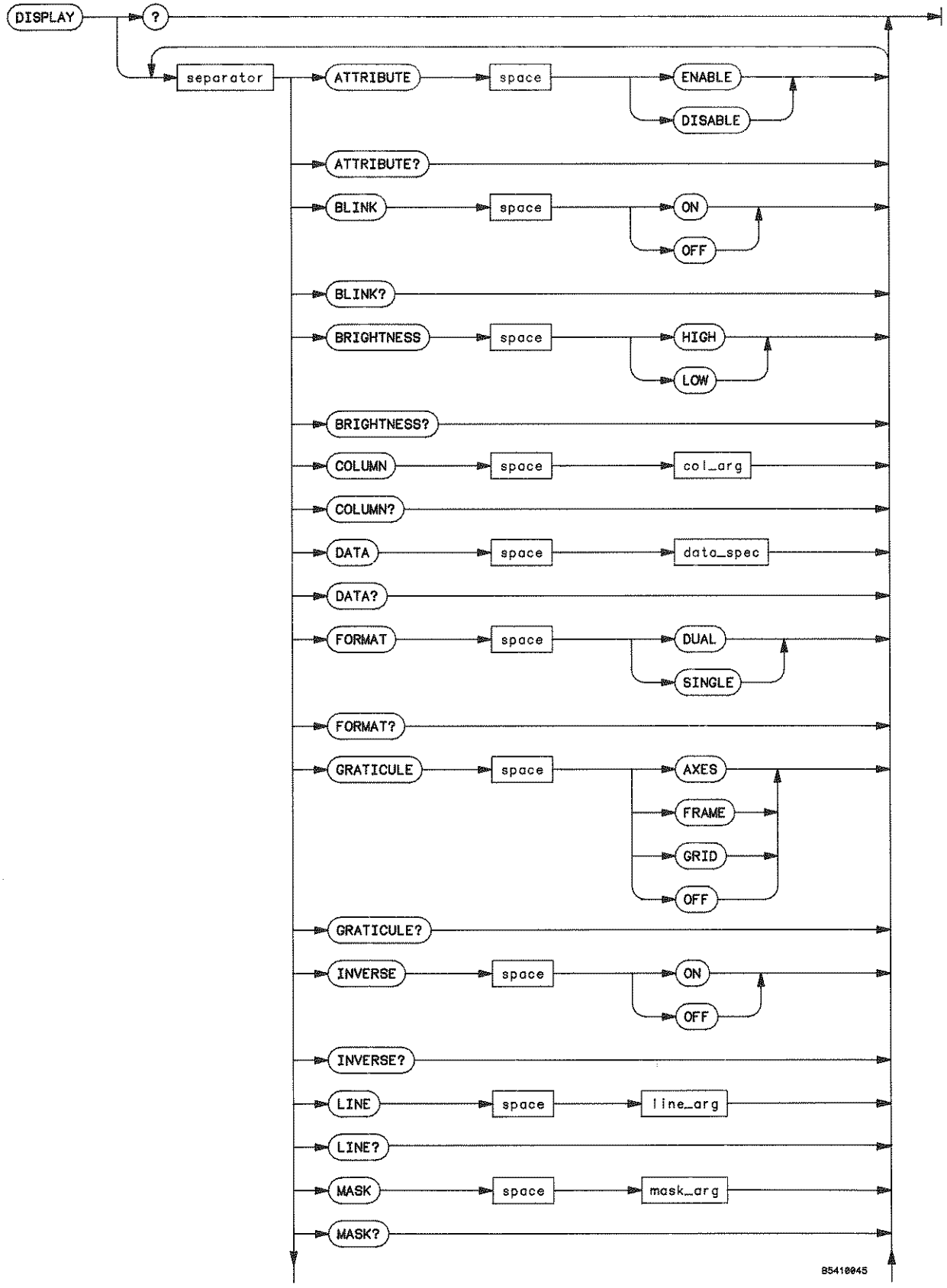


Figure 10-5. Display Subsystem Commands

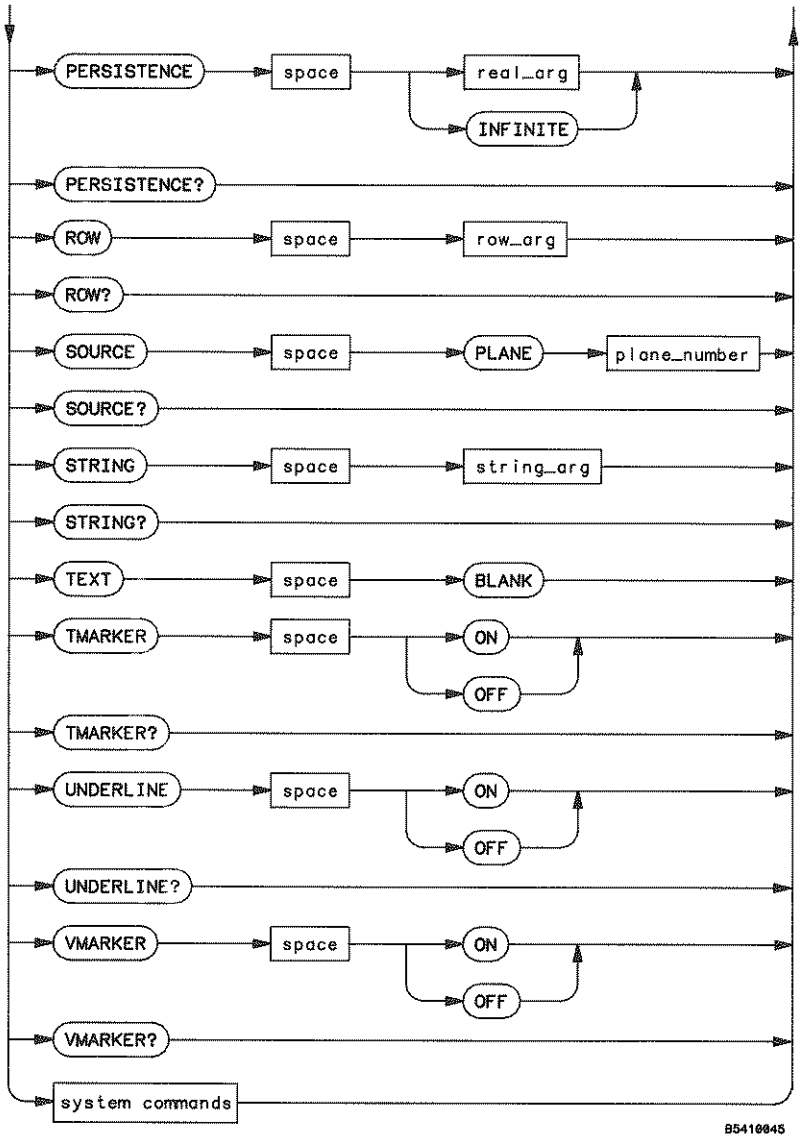


Figure 10-5. Display Subsystem Commands

DATA_SPEC = A block of data in #A format as defined in IEEE Std. 728-1982.

PLANE_NUMBER An integer from 0 to 3.

REAL_ARG = A real number from 0.2 to 10.0 in steps of 0.1.

COL_NUMBER = An integer from 0 to 63.

LINE_ARG = Any quoted string.

ROW_NUMBER = An integer from 0 to 22.

STRING_ARG = Any quoted String.

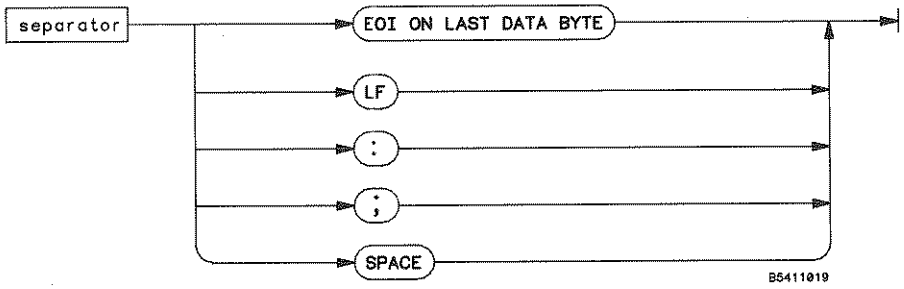


Figure 10-5. Display Subsystem Commands

10-12. DISPLAY SUBSYSTEM

The Display subsystem is used to control the display of data, markers, text and graticules. See Figure 10-5 for the syntax of the Display subsystem commands. The commands which control the display mode and number of averages are listed in the ACQUIRE subsystem as TYPE and COUNT.

DISPlay

command/query

This commands selects the display subsystem as the destination for the subsystem commands. The query returns all the parameters for this subsystem.

Command Syntax: DISPlay

Example: OUTPUT 707;"DISP"

Query Syntax: DISPlay?

Returned Format:
```
[ DISPlay ]<crlf>
[ FORMat ]{[ SINGLE | 1 ]
          [ DUAL | 2 ]}<crlf>
[ GRATicule ]{[ OFF | 0]
             [ GRID | 1 ]
             [ AXES | 2 ]
             [ FRAME | 3 ]}<crlf>
[ ROW ]<NR1><crlf>
[ COLumn ]<NR1><crlf>
[ ATTRibute ]{[ DISABLE | 0 ]
              [ ENABLE  | 1 ]}<crlf>
[ INVerse ]{[ OFF | 0 ]
           [ ON |  1 ]}<crlf>
[ BLINk ]{[ OFF | 0 ]
          [ ON  | 1 ]}<crlf>
[ UNDerline ]{[ OFF | 0 ]
              [  ON | 1 ]
[ BRIGhtness ]{[ LOW | 0 ]
               [ HIGH | 1 ]}<crlf>
[ VMARker ]{[ OFF | 0 ]
            [ ON  | 1 ]}<crlf>
[ TMARker ]{[ OFF | 0 ]
            [ ON  | 1 ]}<crlf>
[ PERSistence ]<NR3><crlf>
```

Example:
```
10 DIM Display$[500]
20 OUTPUT 707;"EOI ON"
30 OUTPUT 707;"DISPLAY?"
40 ENTER 707 USING "-K";Display$
50 PRINT USING "K";Display$
```

ATTRIBUTE BYTE

ATTRIBUTE MSB	NOT USED			HALFBRIGHT	UNDERLINE	BLINK	INVERSE LSB
128	64	32	16	8	4	2	1

When you want to set the attribute byte you must sum the binary values of the byte and send them via the HP-IB to the instrument. These selected attributes can be turned off/on by disabling/enabling the ATTRibute command. These attributes affect the text that is sent to the display of the instrument when you use the LINE or STRing commands. This example causes "HELLO" to be written in the upper left corner of the display using the inverse video and the blinking attributes.

```
OUTPUT 707;"DISPLAY TEXT BLANK ATTRIBUTE ENABLE"
OUTPUT 707 USING "8A,B,6A";"STRING """,128+2+1,"HELLO"""
```

Where 128 indicates that this is an attribute byte, 2 indicates the blink attribute, and 1 indicates the invert attribute. This could just as easily been output to the instrument as "131".

Figure 10-6. Attribute Byte

ATTRibute

command/query

This command controls embedded attributes in the strings that are sent with the DISPLAY, LINE or STRING commands. Refer to Figure 10-6 for more information. These text attributes include:

INVerse
UNDerline
BLINk
BRIGhtness

When this command is enabled the embedded attribute bytes in strings sent with the LINE or STRing commands will be used to override previously set attributes. The query returns the enable/disable state of the command.

Command Syntax: ATTRibute {[DISABLE | 0] [ENABLE | 1]}

```
Example:  OUTPUT 707;"ATTRIBUTE ENABLE"
```

Query Syntax: ATTRibute?

```
Returned format:  [ATTRibute]<argument><crlf>

Example:  OUTPUT 707;"ATTRIBUTE"
          ENTER 707;Attribute$
          PRINT Attribute$
```

BLINk

command/query

This command determines whether subsequent text sent with the DISPLAY LINE or STRING commands is to be written with the BLINk attribute, that is, when the text is displayed it will flash on and off. The query returns the state of the BLINk attribute.

Command Syntax: BLINk {[OFF | 0]
[ON | 1]}

```
Example:  OUTPUT 707;"BLINK ON"
```

Query Syntax: BLINk?

```
Returned Format:  [BLINk]<argument><crlf>

Example:  OUTPUT 707;"BLINK?"
          ENTER 707;Blink$
          PRINT Blink$
```

BRIGhtness

command/query

This command specifies whether subsequent text sent with the DISPLAY LINE or STRING commands is to be half bright or full bright. LOW or 0 provides half bright text and HIGH or 1 provides full bright text. The query returns the HIGH/LOW state of the BRIGhtness attribute.

Command Syntax: BRIGhtness {[LOW | 0]
[HIGH | 1]}

Example: OUTPUT 707;"BRIGHTNESS LOW"

Query Syntax: BRIGhtness?

Returned Format: [BRIGhtness]<argument><crlf>

```
Example:  OUTPUT 707;"BRIGHTNESS?"
          ENTER 707;Brightness$
          PRINT Brightness$
```

COLumn

command/query

This command specifies the starting column for subsequent STRING and LINE commands. The query returns the column where the next LINE or STRING will start.

Command Syntax: COLumn <COL_NUMBER>
<COL_NUMBER> ::= 0..63

Example: OUTPUT 707;"COLUMN 50"

Query Syntax: COLumn?

Returned Format: [COLumn]<NR1><crlf>

```
Example:  OUTPUT 707;"COLUMN?"
          ENTER 707;Column$
          PRINT Column$
```

DATA

command/query

The DATA command is used to write to or from one of the three pixel memory planes in the 54100A/D. The memory planes available are plane0 through plane2 and are specified by the DISPLAY SOURce command.

The DATA query causes the 54100A/D to output waveform data from the specified memory plane.

The DATA command is followed by a block of binary data that is transferred from the controller to a specific plane in the 54100A/D.

The data is in the form of 16032 bytes with four header bytes. The header contains:

```
<#> ::= (decimal 35)  = byte 1
<A> ::= (decimal 65)  = byte 2
        (decimal 62)  = byte 3
        (decimal 160) = byte 4
```

The third and fourth bytes make up a 16-bit integer whose value is decimal 16032, or the length of the binary block. This binary format complies with the "#A" Block Data Field in IEEE 728-1982.

Command Syntax: DATA <binary block type A>

Query Syntax: DATA?

```
Returned Format:  [DATA]<2sp><#><A><decimal 62><decimal 160>
                        <binary waveform data>
```

```
Example:  10   CLEAR 707
          20   DIM Plane$ [17000]
          30   OUTPUT 707;"HEADER ON EOI ON"
          40   OUTPUT 707;"DISPLAY SOURCE PLANE0 DATA?"
          50   ENTER 707 USING "-K";Plane$
          60   OUTPUT 707;"SOURCE PLANE1"
          70   OUTPUT 707 USING "-K";Plane$
          80   END
```

This example transfers data from the active display memory to the controller and then back to pixel memory 5 in the 54100A/D.

FORMat

command/query

This command allows you to turn the split screen mode on or off. FORMAT 1 turns the split screen mode off and uses 8 divisions for the full scale range. FORMAT 2 turns the split screen mode on and uses 4 divisions for the full scale range. The query returns the current number of display areas on the screen.

Command Syntax: FORMat { [SINGLE | 1]
[DUAL | 2]}

Example: OUTPUT 707;"FORMAT SINGLE"

Query Syntax: FORMat?

Returned Format: [FORMat]<argument><crlf>

Example: OUTPUT 707;"FORMAT?"
ENTER 707;Format$
PRINT Format$

GRATicule

command/query

This command allows you to determine the type of graticule that is displayed. The query returns the type of graticule displayed.

Command Syntax: GRATicule {[OFF | 0]
[GRID | 1]
[AXES | 2]
[FRAME | 3]}

Example: OUTPUT 707;"GRATICULE AXES"

Query Syntax: GRATicule?

Returned Format: [GRATicule]<argument><crlf>

Example: OUTPUT 707;"GRATICULE?"
ENTER 707;Grat$
PRINT Grat$

INVerse

command/query

This command sets inverse video on or off for subsequent DISPLAY LINE or STRING commands. The query responds with the on/off state of this command.

Command Syntax: INVerse {[OFF | 0]
[ON | 1]}

Example: OUTPUT 707;"INVERSE OFF"

Query Syntax: INVerse?

Returned format: [INVerse]<argument><crlf>

Example: OUTPUT 707;"INVERSE?"
ENTER 707;Inv$
PRINT Inv$

LINE

command/query

This command causes the string parameter to be written to the screen, starting at the location established by the ROW and COLUMN commands. Text may be written up to column 54. If the characters in the string parameter does not fill the line, the rest of the line is blanked. If the string is longer than the available space on the current line the excess characters will be discarded. In any case, ROW is incremented and COLUMN remains the same. The next LINE command will write on the next line of the display. After writing line 21, the last line in the display area, ROW is reset to 2. The query of this command outputs the quoted string at the current ROW and COLUMN values and causes ROW to be incremented by 1. The LINE command and query works on rows 2 through 21.

Command Syntax: LINE < any quoted string >

Example: OUTPUT 707:"LINE ""ENTER PROBE ATTENUATION"""

Query Syntax: LINE?

Returned Format: [LINE?] < quoted string ><crlf>

Example: DIM Line$[100]
Example: OUTPUT 707;"DISPLAY;ROW 12;COLUMN 14;LINE?"
ENTER 707;Line$
PRINT Line$

MASK

command/query

This command inhibits the instrument from writing to selected areas of the screen. Text sent over the HP-IB using the line and string commands is not effected by this command. The purpose of the command is to allow HP-IB text to be written anywhere on screen and to prevent the instrument from overwriting the text through its normal operation.

The mask parameter is an 8 bit integer in which each bit controls writing to an area of the screen. A 0 inhibits writing to the area represented by the bit, and a 1 enables writing to the area. Note: This command's parameters will not be reset with a RESET command.

Command Syntax: MASK <NR1>

```
Example:  OUTPUT 707;"MASK 254" ! Inhibits advisories only
```

Query Syntax: MASK?

```
Returned Format:  [MASK]<NR1><crlf>

Example:  OUTPUT 707;"MASK?"
          ENTER 707;Mask$
          PRINT Mask$
```

Bit	Mask Weight	Screen Area Effected
7	128	Not used.
6	64	Function Softkeys - Softkey labels on the right side of the display (rows 0-17, columns 56-63).
5	32	Menu Selection Softkeys - text on the bottom line of the display (row 22, columns 0-63).
4	16	Parameter Values - Text below the graticule (rows 18-21, columns 0-63)
3	8	Graticule Labels - text inside the graticule (rows 2-17, columns 0-54)
2	4	Value Label - displays value of selected knob function (row 1, columns 20-54).
1	2	Status Line - status information on the first two lines - (row 0, columns 0-63 and row 1, columns 0-19).
0	1	Advisory - Advisory and Error messages appear on row 15, columns 0-54.

PERSistence

command/query

This command sets PERSistence for the acquired signal on the display in the Normal display mode. The display mode is set to Normal when the ACQuire TYPE is NORMAL, ENVELOPE, or RANDOM. The parameter for this command is the keyword INFINITE or a real number from 0.2 to 10.0 representing the persistence in seconds. Any value greater than 10 seconds will set the PERSistence to infinite. The query returns the value of the current persistence value. If persistence is set to infinite the query response will be 1.1E+1

Command Syntax: PERSistence { NR2 | INFINITE }

Example: OUTPUT 707;"PERSISTENCE INFINITE"

Query Syntax: PERSistence?

Response Format: [PERSistence]<NR3><crlf>

```
Example:  DIM Pers$[30]
          OUTPUT 707;"PERSISTENCE?"
          ENTER 707;Pers$
          PRINT Pers$
```

ROW

command/query

The ROW command specifies the starting row on the CRT for subsequent STRING and LINE commands. The ROW number remains constant until another ROW command is received or it is incremented by the LINE command. The single parameter for this command is an integer from 0 to 22. The query returns the row that the next LINE or STRING will start on.

Command Syntax: ROW <row number>

<row number>::= 0..22

Example: OUTPUT 707;"ROW 10"

Query Syntax: ROW?

Returned Format: [ROW]<NR1><crlf>

```
Example:  OUTPUT 707;"ROW?"
          ENTER 707;Row$
          PRINT Row$
```

SOURce | SRC

command/query

This command allows you to specify the source or destination for the DISPLAY DATA query and command. The SOURce command has 1 parameter, PLANE0..PLANE3. The query returns the currently specified SOURce.

Command Syntax:

```
{ SOURce | SRC }
([PLANE0 | 0]  (active display)
 [PLANE1 | 1]  (pixel memory 5)
 [PLANE2 | 2]  (pixel memory 6)
 [PLANE3 | 3]) (graticule, markers,
                displayed pixel memories,
                and displayed waveform memories)
```

Example:

```
10   CLEAR 707
20   DIM Plane$ [17000]
30   OUTPUT 707;"HEADER ON EOI ON"
40   OUTPUT 707;"DISPLAY SOURCE PLANE0 DATA?"
50   ENTER 707 USING "-K";Plane$
60   OUTPUT 707;"SOURCE PLANE1"
70   OUTPUT 707 USING "K";Plane$
80   END
```

This example transfers data from the active display memory to the controller and then back to pixel memory 5 in the 54100A/D.

Query Syntax: { SOURce | SRC }?

Returned Format: [SOURce]<argument><crlf>

Example:

```
OUTPUT 707;"SRC?"
ENTER 707;Src$
PRINT Src$
```

STRing

command/query

This command allows you to write text to the CRT of the 54100A/D. The text will be written starting at the current ROW and COLUMN values. If the column limit is reached (63) the excess text is discarded. The query returns the text on the line defined by the ROW and COLUMN values.

Command Syntax: STRing <quoted string>

```
Example:  OUTPUT 707;"STRING ""INPUT SIGNAL TO CHANNEL 2"""
```

Query Syntax: STRing?

```
Example:  DIM Str$[90]
          OUTPUT 707;"STRING?"
          ENTER 707;Str$
          PRINT Str$
```

TEXT

command

This command allows you to blank the user text area on the CRT. The user text area includes rows 2 through 17, columns 0 through 54, and rows 18 through 21, columns 0 through 63. This command has only one parameter, BLANk or 2.

Command Syntax: TEXT { BLANk | 2 }

```
OUTPUT 707;"TEXT 2"
```

TMARker

command/query

This command allows you to turn the time markers on or off. The query tells you whether they are on or off.

Command Syntax: TMARKer{[OFF | 0]
[ON | 1]}

```
Example:  OUTPUT 707;"TMAR OFF"
```

Query Syntax: TMARker?

```
Returned Format:  [TMARker]<argument><crlf>

Example:  OUTPUT 707;"TMARKER?"
          ENTER 707;Tmar$
          PRINT Tmar$
```

UNDerline

command/query

This command lets you underline subsequent text sent with the DISPLAY, LINE or STRING commands. The query tells you whether the UNDerline attribute is on or off.

Command Syntax: UNDerline {[OFF | 0]
[ON | 1]}

Example: OUTPUT 707;"UNDERLINE ON"

Query Syntax: UNDerline?

Returned Format: [UNDerline]<argument><crlf>

Example: OUTPUT 707;"UNDERLINE?"
ENTER 707;Under$
PRINT Under$

VMARker

command/query

This command allows you to turn the voltage markers on and off. The query tells you whether they are on or off.

Command Syntax: VMARker {[OFF | 0]
[ON | 1]}

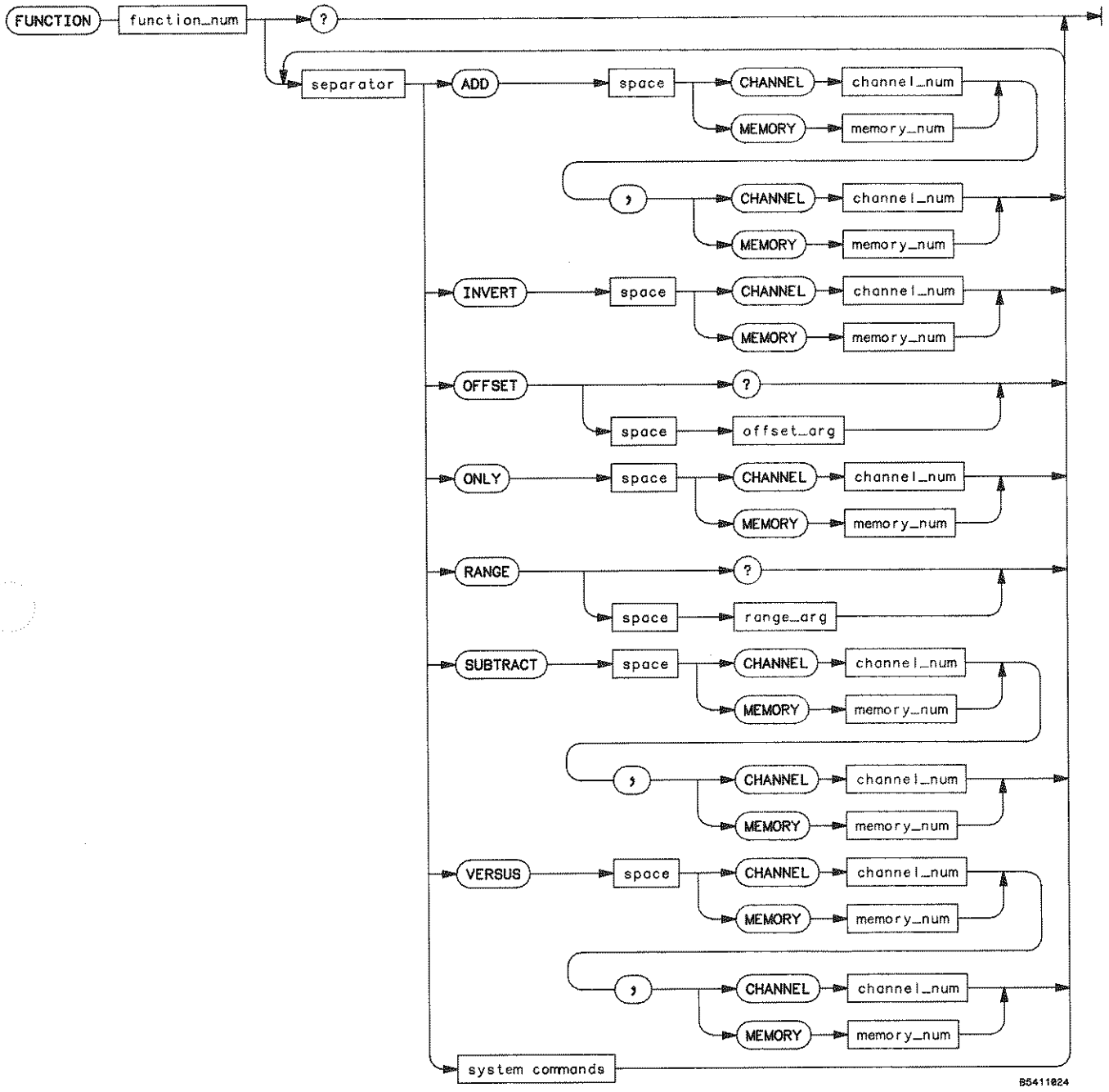
Example: OUTPUT 707;"VMARKER ON"

Query Syntax: VMARker?

Returned Format: [VMARker]<argument><crlf>

Example: OUTPUT 707;"VMARKER?"
ENTER 707;Vmark$
PRINT Vmark$

NOTES:



CHANNEL_NUM = An integer 1 or 2.
MEMORY_NUM = An integer 1 through 4.

Figure 10-7. Function Subsystem Commands

10-13. FUNCTION SUBSYSTEM

The Function subsystem allows you to define two functions using the displayed channels and/or the waveform memories as operands. The waveform operators are: ADDition, SUBTraction, INVErt, VERSus, and ONLY. The vertical scaling and offset and the display of these functions can be controlled remotely. See Figure 10-7 for a syntax diagram of the function subsystem commands.

When a function is first defined, it's initial vertical values are calculated with respect to the operands' vertical settings. The functions' range and offset may be changed using the range and offset commands. Changing any of the operands' vertical settings or redefining of the function will cause the the functions' vertical settings to be recalculated with respect to the new operand values. Any previously programmed vertical settings for the function will be lost.

The functions work on operands containing 500 points. If a function is defined and turned on using a memory operand which contains other than 500 points, the memory will be reformatted to 500 points. Also, memory operands that are in the RANDOM type will be reformatted to the NORMAL type with the number of points equal to 500.

FUNCtion

command/query

This command allows you to select the Function subsystem and define a waveform function. This command selects the function subsystem as the destination for the commands that follow. The query returns the definition of the selected function. Refer to Figure 10-7 for a syntax diagram of the Function subsystem commands.

Command Syntax: FUNCtion{ 1 | 2 }

Example: OUTPUT 707;"FUNCTION1 ADD CHANNEL1 CHANNEL2

Query Syntax: [FUNCtion]{ 1 | 2 }?

Returned Format:
```
[FUNCtion]{ 1 | 2 }<crlf>
{ADD | INVErt| ONLY | SUBTract | VERsus}
{[CHANnel 1 | 2] | [MEMory 1 | 2 | 3 | 4]}<,>
[{[CHANnel 1 | 2] | [MEMory 1 | 2 | 3 | 4]}]<crlf>
[OFFSet]<NR3><crlf>
[RANGe]<NR3><crlf>
```

Example:
```
DIM Fun$[300]
OUTPUT 707;"EOI ON"
OUTPUT 707;"FUNCTION1?"
ENTER 707 USING "-K";Fun$
PRINT USING "K";Fun$
```

ADD

command

The ADD command causes the unit to algebraically sum the two defined operands.

Command Syntax: ADD<operand1><,><operand2>

operand 1 & 2 ::= {channel 1 | channel 2 | memory 1 | memory 2 | memory 3 | memory 4}

Example: OUTPUT 707;"FUNCTION1 ADD MEMORY3,MEMORY4"

INVErt

command

This command allows you to invert the operand, that is channel 1 | 2, or memory 1 | 2 | 3 | 4. Note that the shortform of the command is INVE. The INVERSE command in the display subsystem uses the short form INV.

Command Syntax: INVErt<operand>

Example: OUTPUT 707;"FUNCTION2 INVERT MEMORY3"

OFFSet

command/query

The OFFSet command allows you to define the vertical voltage at center screen for the selected function. The query returns the voltage at center screen for the defined function.

Command Syntax: OFFSet<Offset_Arg>

Example: OUTPUT 707;"FUNCTION1 OFFSET .05"

Query Syntax: OFFSet?

Returned Format: [OFFSet]<NR3>

```
Example: DIM Off$[30]
         OUTPUT 707;"FUNCTION2 OFFSET?"
         ENTER 707;Off$
         PRINT Off$
```

ONLY

command

The ONLY command allows you to define a function as either channel 1 or 2, or memory 1, 2, 3, or 4. The ONLY command is useful for scaling channels and memories.

Command Syntax: ONLY<operand>

Example: OUTPUT 707;"FUNCTION1 ONLY MEMORY1"

RANGe

command/query

This command allows you to define the full scale vertical axis of a function's display.

Command Syntax: RANGe<Range_Arg>

Example: OUTPUT 707;"FUNCTION1 RANGE .01"

Query Syntax: RANGe?

Returned Format: [RANGe]<NR3>

```
Example:  DIM Range$[30]
          OUTPUT 707;"RANGE?"
          ENTER 707;Range$
          PRINT Range$
```

SUBTract

command

This command allows you to algebraically subtract one operand from another. Operand2 is subtracted from operand1.

Command Syntax: SUBTract <operand1><,><operand2>

```
operand1 & 2 ::= {channel1 |channel2 | memory1 |
                  memory2 | memory3 | memory4}
```

Example: OUTPUT 707;"FUNCTION2 SUBTRACT CHANNEL1,CHANNEL2"

(In this example channel 2 would be algebraically subtracted from channel 1.)

VERSus

command

This command allows X vs Y displays with two operands. The first operand defines the Y axis and the second defines the X axis. The Y axis range and offset is initially equal to the first operand's and can be adjusted using the range and offset commands in this subsystem. The X axis range and offset is always equal to that of the second operand. It can only be changed by changing the vertical settings of the second operand. This will also change the Y axis vertical sensitivity and offset.

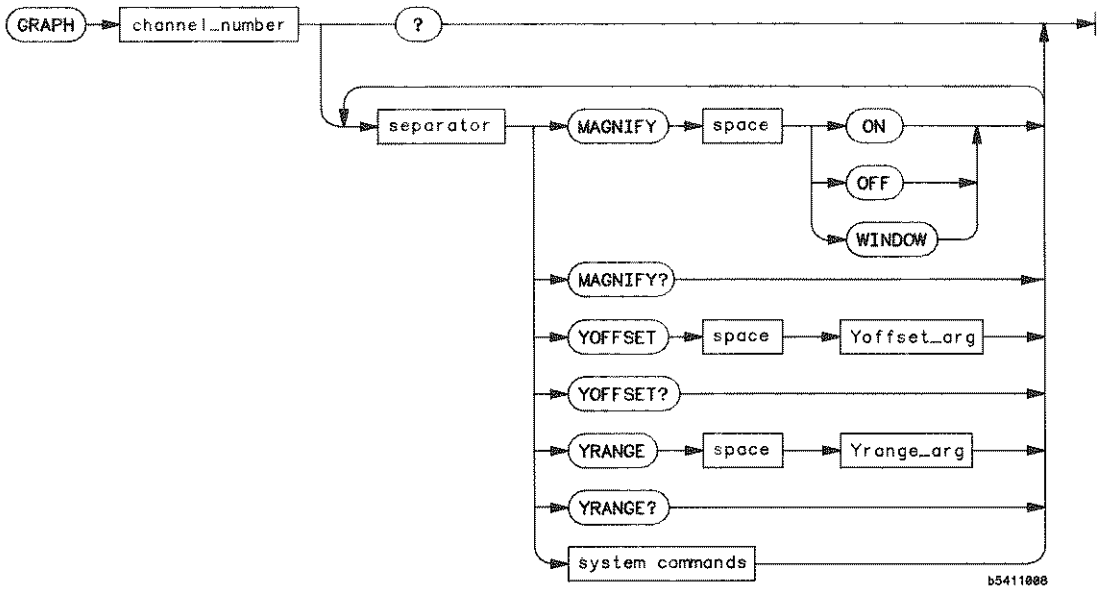
Command Syntax: VERSus<operand1><,><operand2>

operand1 & 2 ::= {channel 1 | channel 2 | memory 1 | memory 2 | memory 3 | memory 4}

Example: OUTPUT 707;"FUNCTION2 VERSUS CHANNEL1,MEMORY3"

NOTES:

NOTES:



YOFFSET_ARG = A real number less than or equal to the vertical range.

YRANGE_ARG = A real number between 1/16(vertical range) and the vertical range.

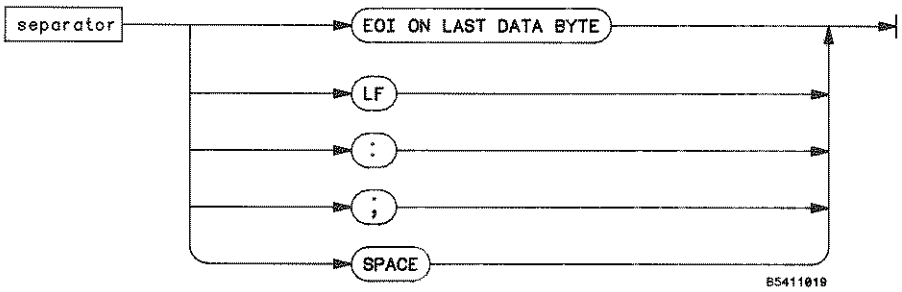


Figure 10-8. Graph Subsystem Commands

10-14. GRAPH SUBSYSTEM

The Graph subsystem allows you to control y-axis windowing, offset and the magnification for the two channels. See Figure 10-8 for a syntax diagram of the GRAPh subsystem commands.

GRAPh

command/query

This command allows you to select the graph subsystem and specify which input channel will be the destination for the graph subsystem commands that follow. The query responds with all the parameters in the subsystem. Graph 1 refers to the display for channel 1 and Graph 2 refers to the display for channel 2.

Command Syntax: GRAPh { 1 | 2 }

```
Example:  OUTPUT 707;"GRAPH1"
```

Query Syntax: GRAPh { 1 | 2 }?

```
Returned Format:  [GRAPh]<NR1><crlf>
                  [MAGNify]<argument><crlf>
                  [YOFfset]<NR3><crlf>
                  [YRANge]<NR3><crlf>

Example:  DIM Graph$[100]
          OUTPUT 707;"EOI ON"
          OUTPUT 707;"GRAPH1?"
          ENTER 707 USING "-K";Graph$
          PRINT USING "-K";Graph$
```

MAGNify

command/query

This command controls the MAGNify function for a specific channel. This command has one parameter: OFF, ON, or WINDOW. Off specifies that the channel will be displayed on the CRT in the unmagnified form. On specifies that the channel will be displayed in the magnified form. Window specifies that the channel will be displayed in the unmagnified form with the magnifier window displayed. The window is only displayed when the menu for the specified channel is selected.

Command Syntax: MAGNify {[OFF | 0][ON | 1][WINDOW | 2]}

```
Example:  OUTPUT 707;"MAGNIFY OFF"
```

Query Syntax: MAGNify?

```
Returned Format:  [MAGNify]<argument><crlf>

Example:  OUTPUT 707;"MAGNIFY?"
          ENTER 707;Mag$
          PRINT Mag$
```

YOFfset

command/query

This command allows you to control the voltage at the center of the magnify window. This voltage must be within the vertical range that is setup with the CHANNELn RANGE and OFFSET commands. The query returns the current value of YOFfset.

Command Syntax: YOFfset { NR1 | NR2 | NR3 }

```
Example:  OUTPUT 707;"YOFFSET 1E-3"
```

Query Syntax: YOFfset?

```
Returned Format:  [YOFfset]<NR3><crlf>

Example:  OUTPUT 707;"YOFFSET?"
          ENTER 707;Y$
          PRINT Y$
```

YRANge

command/query

This command allows you to control the size (in volts) of the magnify window. The combination of this command and the YOFfset command must define a window that is completely enclosed by the vertical range that is setup with the CHANneln RANGe and OFFSet commands. The query returns the current value of YRANge.

Command Syntax: YRANge { NR1 | NR2 | NR3 }

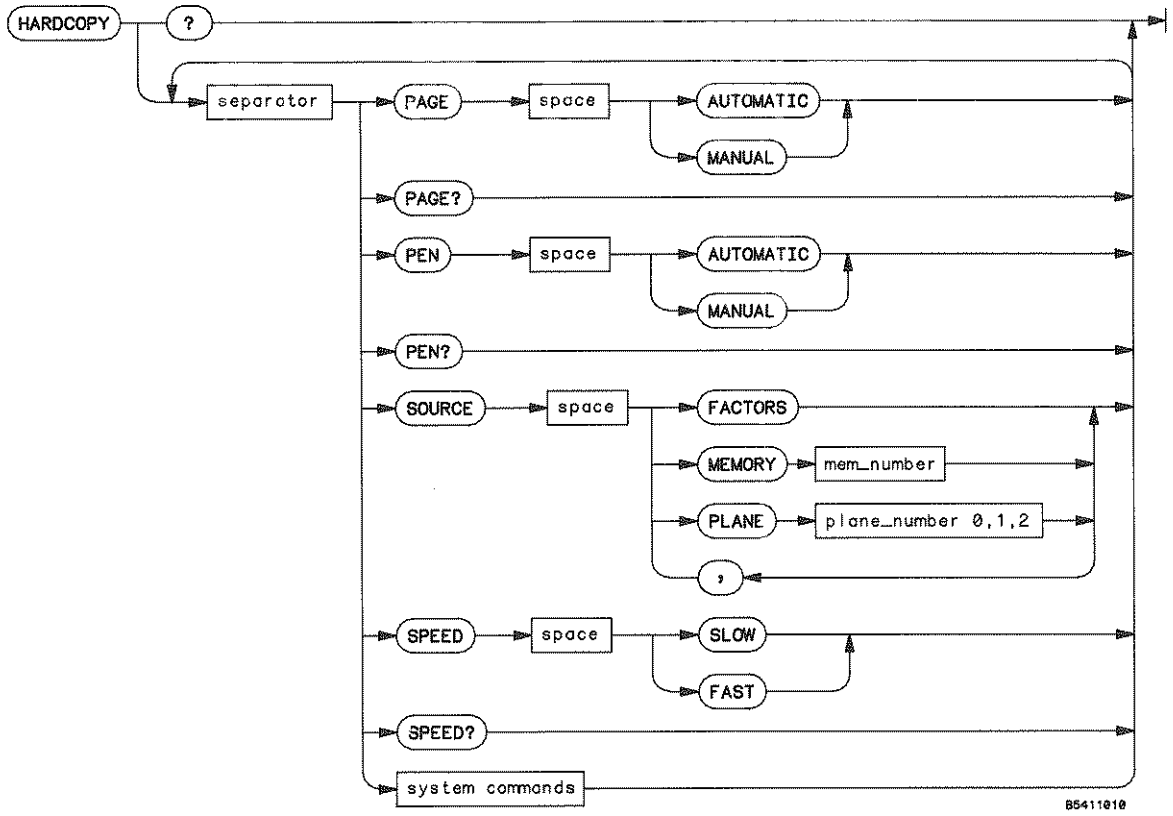
```
Example:  OUTPUT 707;"YRANGE .01"
```

Query Syntax: YRANge?

```
Returned Format:  [YRANge]<NR3><crlf>

Example:  OUTPUT 707;"YRANGE?"
          ENTER 707;Yr$
          PRINT Yr$
```

NOTES:



MEM_NUMBER = An integer from 1 to 4. **PLANE_NUMBER** = An integer from 0 to 2.

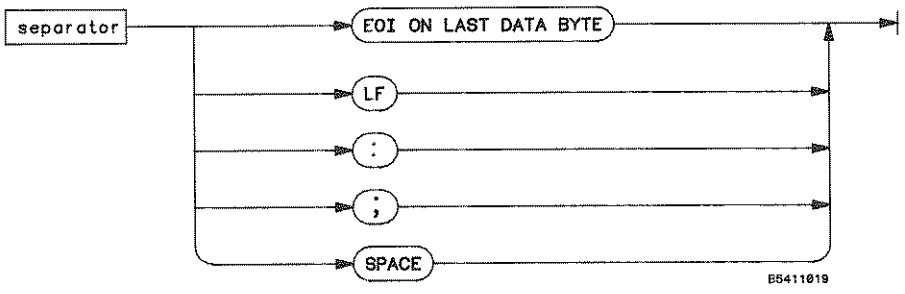


Figure 10-9. Hardcopy Subsystem commands

10-15. HARDCOPY SUBSYSTEM

The commands in the HARDcopy subsystem allow you to set various parameters used during the plotting and printing waveforms from the 54100A/D. Refer to Figure 10-9 for the syntax diagram of the HARDcopy subsystem commands.

HARDcopy

command/query

The HARDcopy command selects the hardcopy subsystem as the destination for the commands that follow.

Command Syntax: HARDcopy

```
Example:  OUTPUT 707;"HARDCOPY"
```

Query Syntax: HARDcopy?

```
Returned Format:  [HARDcopy]<crlf>
                  [PAGE]<argument><crlf>
                  [PEN]<argument><crlf>
                  [SPEed]<argument><crlf>

Example:  DIM Hard$[100]
          OUTPUT 707;"EOI ON"
          OUTPUT 707;"HARDCOPY?"
          ENTER 707 USING "-K";Hard$
          PRINT USING "-K";Hard$"
```

PAGE

command/query

The page command allows you to send a form feed after a hardcopy dump to a printer. During a hardcopy dump the 54100A/D ignores page boundaries. The query returns the current state of the page command parameter.

Command Syntax:

```
PAGE ([ MANual | 0 ]
      [ AUTOmatic | 1 ])

Example:  OUTPUT 707;"PAGE AUTO"
```

Query Syntax: PAGE?

```
Returned Format:  [PAGE]<argument><crlf>

Example:  OUTPUT 707;"PAGE?"
          ENTER 707;Page$
          PRINT Page$
```

PEN

command/query

The PEN command allows you to set the 54100A/D's pen control function. When this command is set to AUTOmatic the unit assigns the following pen numbers to these functions:

Pen #	Usage
1	Graticule and timebase factors
2	Channel 1 & 2 and associated factors
3	Waveform memories and associated factors
4	Pixel memories 5 & 6
5	Markers and delta measurement results

When the command is put in the MANual mode the plotter will not be instructed to select a pen when a plot is requested, at the completion of the plot an instruction will be sent to cause the plotter to put away the pen. The query returns the state of the pen control parameter.

Command Syntax: PEN {[MANual | 0]
[AUTOmatic | 1]}

Example: OUTPUT 707;"PEN AUTOMATIC"

Query Syntax: PEN?

Returned Format: [PEN]<argument><crlf>

```
Example:  OUTPUT 707;"PEN?"
          ENTER 707;Pen$
          PRINT Pen$
```

SOURce | SRC

command

The SOURce command specifies the source(s) to be output during a hardcopy dump. Commas should be used when specifying multiple sources.

```
Command Syntax:  { SOURce | SRC }
                 {[PLANe0 | 0]    (active display)
                  [PLANe1 | 1]    (pixel memory 5)
                  [PLANe2 | 2]    (pixel memory 6)
                 [FACTors | 4]    (scale factors)
                 [MEMory1 | 11]   (waveform memory 1)
                 [MEMory2 | 12]   (waveform memory 2)
                 [MEMory3 | 13]   (waveform memory 3)
                 [MEMory4 | 14]}  (waveform memory 4)
```

Example: OUTPUT 707;"SOURCE PLANE2,MEMORY1"

SPEed

command/query

The SPEed command allows you to specify the pen speed to be used during plotting. FAST is intended for use on normal paper and SLOW should be used when plotting transparencies. The query returns the current pen speed.

Command Syntax: SPEed {[SLOW | 0]
[FAST | 1]}

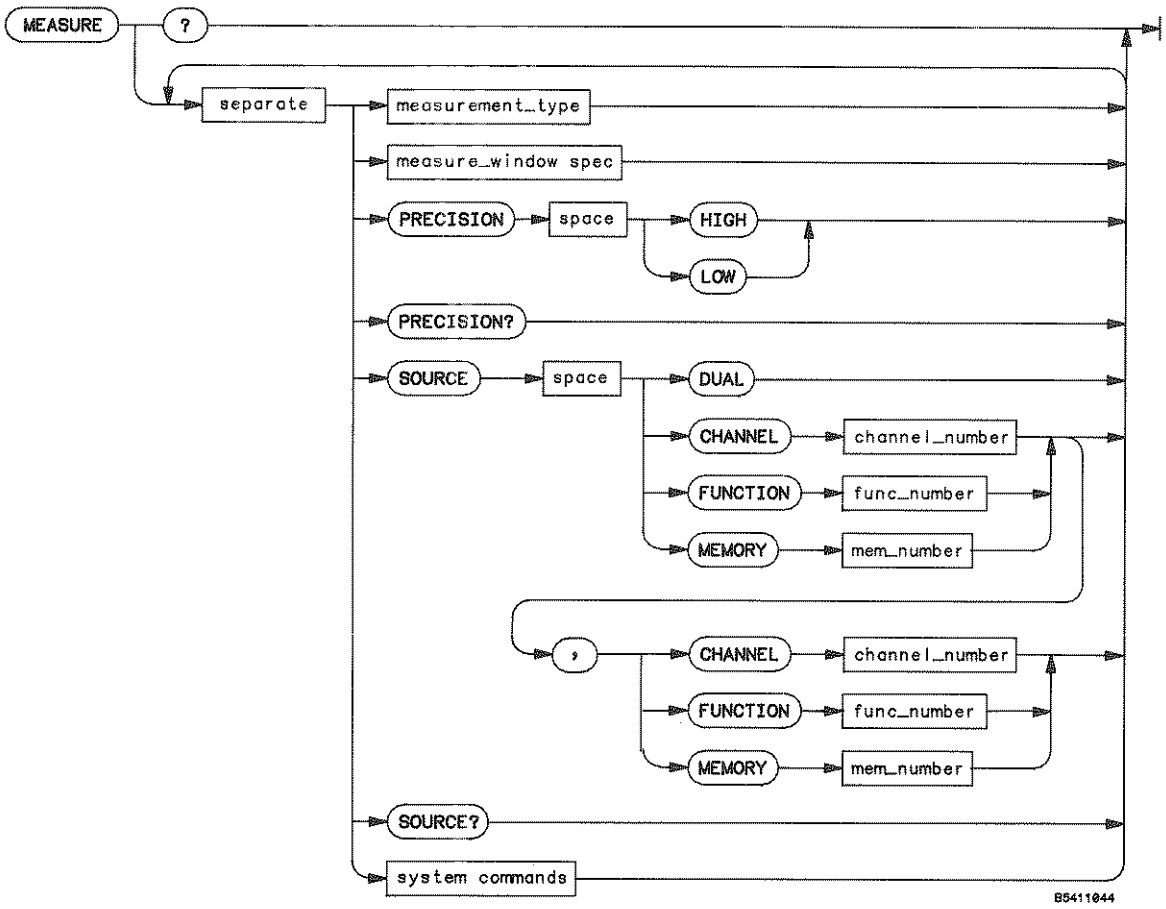
Example: OUTPUT 707;"SPEED FAST"

Query Syntax: SPEed?

Returned Format: [SPEed]<argument><crlf>

Example: OUTPUT 707;"SPEED?"
ENTER 707;Speed$
PRINT Speed$

NOTES:



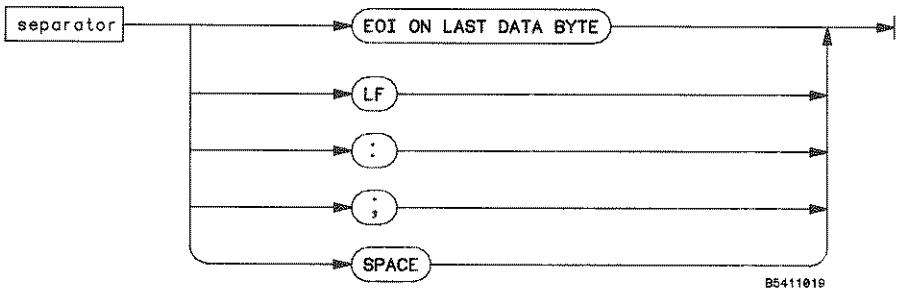


Figure 10-10. Measure Subsystem Commands

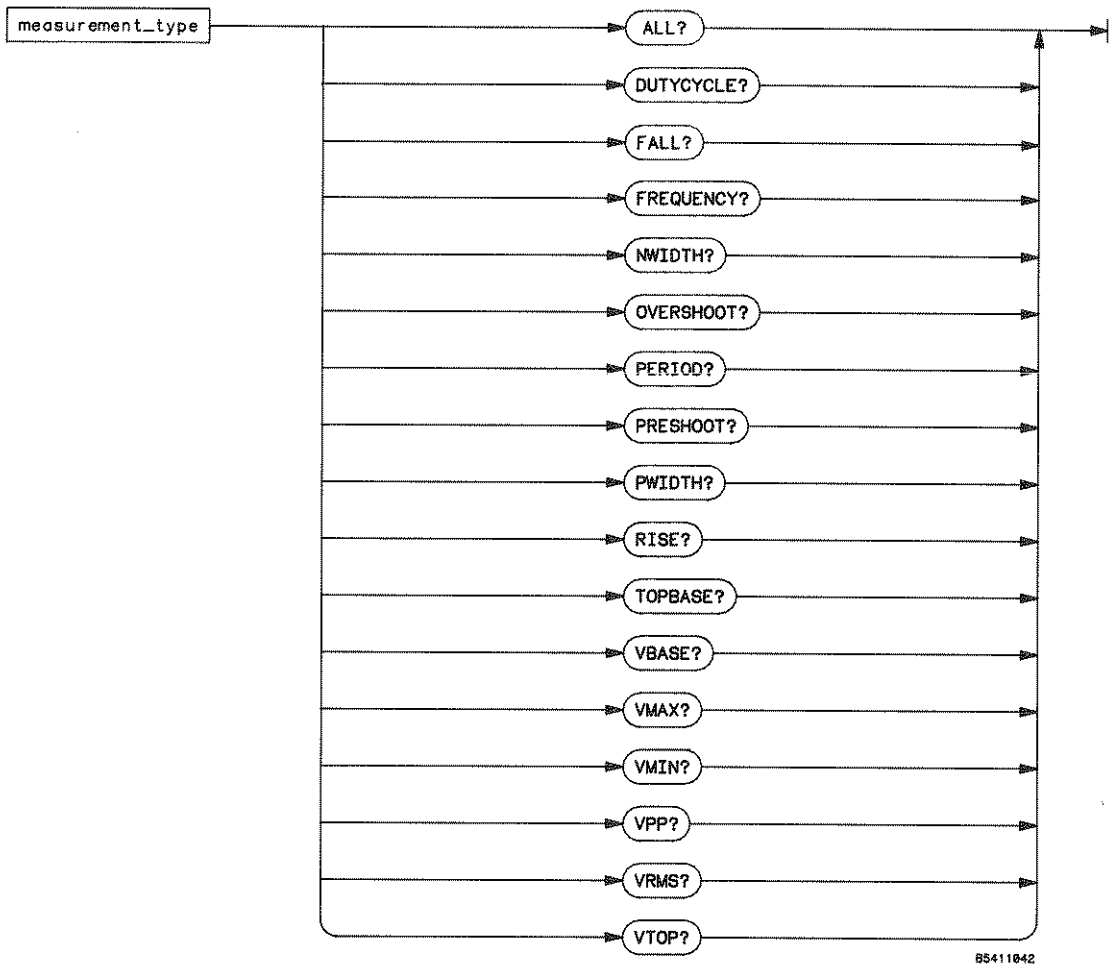


Figure 10-10. Measure Subsystem Commands

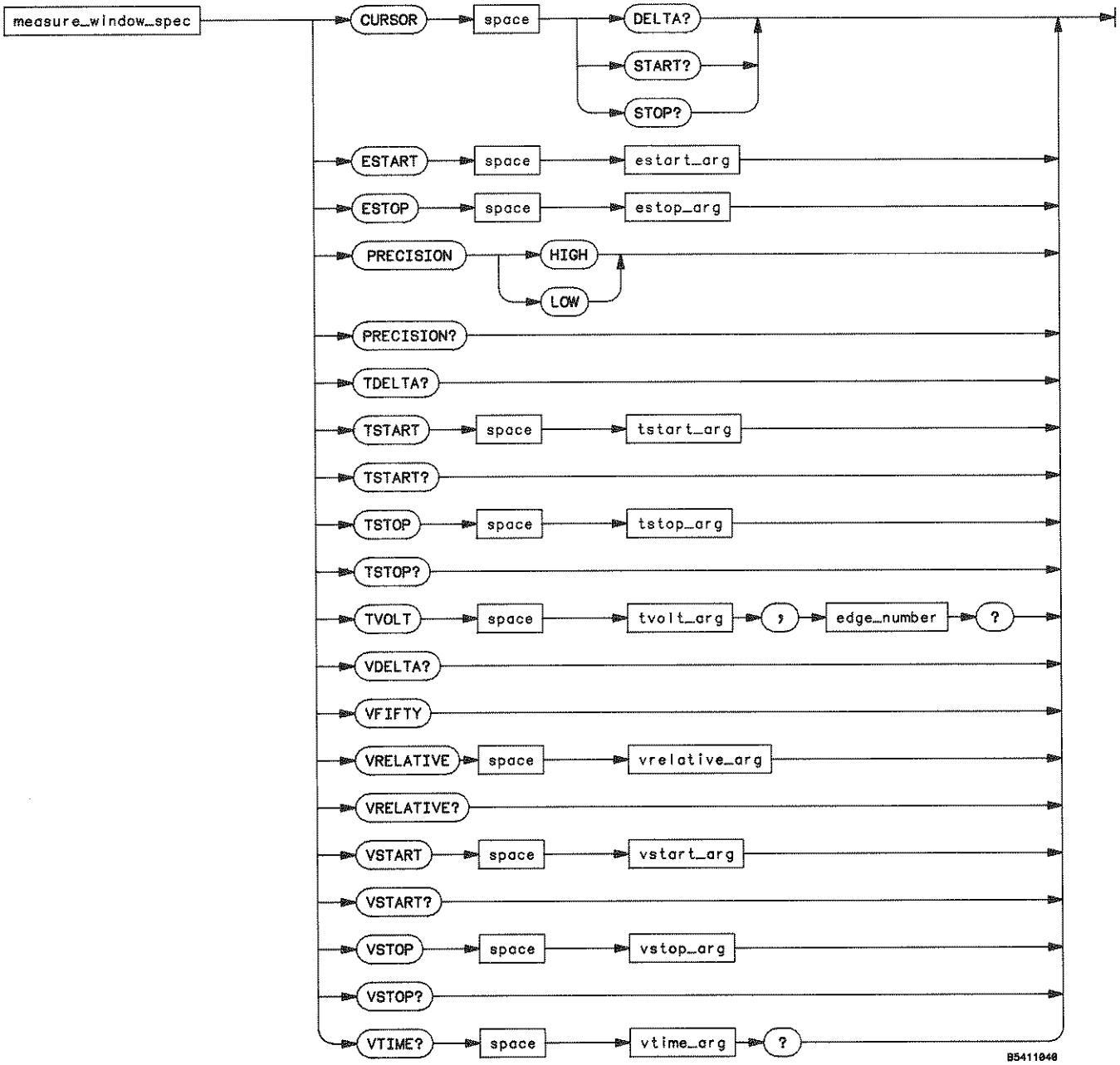


Figure 10-10. Measure Subsystem Commands

CHANNEL_NUMBER = An integer, 1 or 2.

EDGE_NUMBER = An integer from 1 to 100.

FUNC_NUMBER = An integer from 1 to 2.

MEM_NUMBER = An integer from 1 to 4.

VTIME_ARG = A real number that is within the horizontal display window.

VSTART_ARG = A real number <=2 X voltage range.

VSTOP_ARG = A real number <=2 X voltage range.

TSTART_ARG = A real number with the following restrictions:

```
Maximum is 60,000 X timebase range or 1.6sec, whichever is greater
If the delay reference is left
  then minimum is 200 ms or -(timebase range), whichever is smaller
Else if the delay reference is center
  then minimum is -5(timebase range) or -200 ms+(timebase range)
  whichever is smaller.
Else if the delay reference is right
  then minimum is 0 or -200 ms+10(timebase range),
  whichever is smaller.
```

TSTOP_ARG = A real number with the same restrictions as TSTART_ARG.

ESTART_ARG = An integer between 0 and 100.

ESTOP_ARG = An integer between 0 and 100.

VREL_ARG = An integer 0, 10, 20, 50.

Figure 10-10. Measure Subsystem Commands

10-16. MEASURE SUBSYSTEM

The commands in the MEASure subsystem allow you to make pulse parameter and voltage measurements. You may also make custom measurements using the voltage and time markers. Pulse parameter measurements are made on the left side of the display. If there isn't enough signal present on the display to make a measurement, 1E38 is returned. Measurements are made using previously specified PRECision. If PRECision is set to LOW, the waveform will not be expanded. If PRECision is set HIGH the unit will attempt to increase the precision of the measurement by making the sweep faster. Low precision measurements typically are accomplished faster than high precision measurements because of the additional time required for expansion. All predefined pulse parameter measurements cause an Auto Top-Base operation to be performed on the displayed signal. This operation determines the 10, 90, and 50 percent levels that are used to make the measurements. For more detailed information concerning the automated measurements refer to Appendix C. Refer to Figure 10-10 for a syntax diagram of the measure subsystem commands.

MEASure

command/query

The MEASure command selects the measure subsystem as the destination for the commands that follow. The query responds with selected measurement parameters.

Command Syntax: MEASure

Example: OUTPUT 707;"MEASURE"

Query Syntax: MEASure?

Returned Format:
```
[MEASure]<crlf>
[SOURce][CHANnel]<NR1>[FUNCtion]<NR1>
   [MEMory]<NR1><crlf>
[PRECision]<argument><crlf>
[VDELta]<NR3><crlf>
[VSTArt]<NR3><crlf>
[VSTOp]<NR3><crlf>
[TDELta]<NR3><crlf>
[TSTArt]<NR3><crlf>
[TSTOp]<NR3><crlf>
```

Example:
```
DIM Mea$[200]
OUTPUT 707;"EOI ON"
OUTPUT 707;"MEASURE?"
ENTER 707 USING "-K";Mea$
PRINT USING "K";Mea$
```

ALL?

query

This query makes as many measurements as possible on the displayed signal and buffers the answers for output over HP-IB. If the measurement cannot be made the instrument will respond with 1.00000E+38.

Query Syntax: ALL?

```
Returned Format:  [FREQuency]<NR3><crlf>
                  [PERiod]<NR3><crlf>
                  [PWIDth]<NR3><crlf>
                  [NWIDth]<NR3><crlf>
                  [RISE]<NR3><crlf>
                  [FALL]<NR3><crlf>
                  [TOPBase]<NR3><crlf>
                  [VPP]<NR3><crlf>
                  [PREShoot]<NR3><crlf>
                  [OVERshoot]<NR3><crlf>
                  [DUTYcycle]<NR3><crlf>
                  [VRMS]<NR3><crlf>
                  [VMAX]<NR3><crlf>
                  [VMIN]<NR3><crlf>
                  [VTOP]<NR3><crlf>
                  [VBASe]<NR3><crlf>
```

```
Example:  DIM All$[500]
          OUTPUT 707;"EOI ON"
          OUTPUT 707;"ALL?"
          ENTER 707 USING "-K";"All$"
          PRINT USING "K";All$
```

CURSor?

query

This query returns time and voltage values of the specified marker as an ordered pair of time/voltage values. If delta is specified the instrument returns the value of delta V and delta T. If start is specified the positions of Vmarker1 and the start marker are returned. If stop is specified the positions of Vmarker2 and the stop marker are returned.

Query Syntax:

```
CURSor ([ DELTA | 0 ]
        [ START | 1 ]
        [ STOP  | 2 ])?
```

```
Returned Format:  [CURSor]<NR3><,><NR3><crlf>
```

```
Example:  DIM Cursor$[30]
          OUTPUT 707;"CURSOR1?"
          ENTER 707;Cursor$
          PRINT Cursor$
```

DUTYcycle? | DUT?

query

This query causes the instrument to determine the duty cycle of the signal. The pulse width is measured at the 50% voltage point. The duty cycle is computed using the following formula:

duty cycle = (+pulse width/period) X 100

Query Syntax: { DUTYcycle | DUT }?

Returned Format: [DUTYcycle]<NR3><crlf>

```
Example:  DIM Dut$[30]
          OUTPUT 707;"DUTYCYCLE?"
          ENTER 707;Dut$
          PRINT Dut$
```

ESTArt

command/query

This command causes the instrument to position the start marker on the specified edge and slope at the voltage level corresponding to voltage marker 1. A positive integer looks for a specific positive transition and a negative integer looks for a specific negative transition through the voltage level equal to the voltage level of voltage marker 1. The query responds with the currently specified edge.

Command Syntax: ESTArt<NR1>

```
Example:  OUTPUT 707;"ESTART 2"
```

Query Syntax: ESTArt?

Returned Format: [ESTART]<NR1><crlf>

```
Example:  OUTPUT 707;"ESTART?"
          ENTER 707;Es$
          PRINT Es$
```

ESTOp

command/query

This command causes the instrument to position the stop marker on the specified edge and slope at the voltage level corresponding to voltage marker 2. A positive integer looks for a specific positive transition and a negative integer looks for a specific negative transition through the voltage level equal to the level of voltage marker 2. The query returns the currently specified edge.

Command Syntax: ESTOp

Example: OUTPUT 707;"ESTOP-2"

Query Syntax: ESTOP?

Returned Format: [ESTOP]<NR1><crlf>

```
Example:  OUTPUT 707;ESTOP?
          ENTER 707;Es$
          PRINT Es$
```

FALL?

query

This query causes the instrument to measure the fall time of the first falling edge whose 10% and 90% points are on screen using the formula:

fall time = time at 10% point - time at 90% point.

Query Syntax: FALL?

Returned Format: [FALL]<NR3><crlf>

```
Example:  OUTPUT 707;"FALL?"
          ENTER 707;Fall$
          PRINT Fall$
```

FREQuency?

query

This query causes the instrument to measure the frequency of the first complete period on screen using the 50% levels. The algorithm used is:

```
if first edge on screen is rising
     then
        frequency = 1/(time at second rising edge
                       - time at first rising edge)
     else
        frequency = 1/(time at second falling edge
                       - time at first falling edge)
```

Query Syntax: FREQuency?

```
Example:  DIM Freq$[30]
          OUTPUT 707;"FREQuency?"
          ENTER 707;Freq$
          PRINT Freq$
```

NWIDth? query

This query causes the instrument to measure the pulse width of the first negative pulse on screen using the 50% levels. The algorithm used is:

```
if first edge on screen is rising
     then
        width = (time at second rising edge
                 - time at first falling edge)
     else
        width = (time at first rising edge
                 - time at first falling edge)
```

Query Syntax: NWIDth?

Returned Format: [NWIDth]<NR3><crlf>

```
Example:  DIM Nwid$[30]
          OUTPUT 707;"NWIDTH?"
          ENTER 707;Nwid$
          PRINT Nwid$
```

OVERshoot? query

This query causes the instrument to measure the overshoot of a selected signal. Overshoot uses the first edge on screen using the following algorithm:

```
if the first edge on screen is rising
   then
      overshoot = Vmax - Vtop
   else
      overshoot = Vbase - Vmin
```

Query Syntax: OVERshoot?

Returned format: [OVERshoot]<NR3><crlf>

```
Example:  DIM Over$[30]
          OUTPUT 707;"OVERSHOOT?"
          ENTER 707;Over$
          PRINT Over$
```

PERiod? query

This command causes the instrument to measure the period of the first complete cycle on screen using the 50% level. The algorithm is:

```
if first edge on screen is rising
   then
      period = (time at second rising edge
                - time at first rising edge)
   else
      period = (time at second falling edge
                - time at first falling edge)
```

Query Syntax: PERiod?

Returned format: [PERiod]<NR3><crlf>

```
Example:  DIM Period$[30]
          OUTPUT 707;"PERIOD?"
          ENTER 707;Period$
          PRINT Period$
```

PRECision command/query

This command allows you to specify the precision that is used on subsequent measurements. When PRECision is set to HIGH the edges used for making a measurement are evaluated by making the sweep speed faster until the edge has a slope of approximately 45 degrees or the limit of the horizontal system has been reached. This increases the resolution of the measurement. When PRECision is set to LOW no horizontal expansion is accomplished. Low precision allows you increase measurement speed with the potential of reduced accuracy.

Command Syntax: PRECision {[LOW | 0]
[HIGH | 1]}

```
Example:  OUTPUT 707;"PRECISION LOW"
```

Query Syntax: PRECision?

Returned Format: [PRECision]<argument><crlf>

```
Example:  DIM Prec$[30]
          OUTPUT 707;"PREC?"
          ENTER 707;Prec$
          PRINT Prec$
```

PREShoot? query

This query causes the instrument to measure the preshoot of the selected SOURce. The PREShoot command uses the first edge on screen using the following algorithm:

```
if the first edge on screen is rising
   then
      preshoot = Vbase - Vmin
   else
      preshoot = Vmax - Vtop
```

Query Syntax: PREShoot?

Returned Format: [PREShoot]<NR3><crlf>

Example:
```
DIM Pres$[30]
OUTPUT 707;"PRESHOOT?"
ENTER 707;Pres$
PRINT Pres$
```

PWIDth? query

This query causes the instrument to measure the pulse width of the first positive pulse on screen using the 50% levels. The algorithm used is:

```
if first edge on screen is falling
   then
      width = (time at second falling edge
               - time at first rising edge)
   else
      width = (time at first falling edge
               - time at first rising edge)
```

Query Syntax: PWIDth?

Returned Format: [PWIDth]<NR3><crlf>

Example:
```
DIM Pw$[30]
OUTPUT 707;"PWIDTH?"
ENTER 707;Pw$
PRINT Pw$
```

RISE? query

This query causes the instrument to measure the rise time of the first rising edge whose 10% and 90% points are on screen using the formula:

rise time = (time at 90% point - time at 10% point)

Query Syntax: RISE?

Returned Format: [RISE]<NR3><crlf>

Example:
```
OUTPUT 707;"RISE?"
ENTER 707;Rise$
PRINT Rise$
```

SOURce | SRC command/query

This command selects the source(s) to be used for subsequent measurements. If the source is specified as CHANnel1 or CHANnel2, that channel will be used as the source for subsequent MEASure commands. For dual measurements, 2 parameters are specified after the source command. Vmarker 1 and the start marker will be assigned to the first and Vmarker 2 and the stop marker will be assigned to the second. If the keyword DUAL is used as the measurement source the markers will be assigned to chan 1 and 2 respectively. The marker measurement commands that work in dual measurements are: ESTART, ESTOP, TSTART TSTOP, TDELTA, VSTART, VSTOP, AND VDELTA.

Command Syntax:
```
{ SOURce | SRC }{[ DUAL | 0 ]          | [<,>
                [ CHANnel1 | 1 ]      | {[CHANnel1 | 1 ]
                [ CHANnel2 | 2 ]      |  [CHANnel2 | 2 ]
                [ FUNCtion1 | 9 ]     |  [FUNCtion1 | 9 ]
                [ FUNCtion2 | 10 ]    |  [FUNCtion2 | 10 ]
                [ MEMory1 | 11 ]      |  [MEMory1 | 11 ]
                [ MEMory2 | 12 ]      |  [MEMory2 | 12 ]
                [ MEMory3 | 13 ]      |  [MEMory3 | 13 ]
                [ MEMory4 | 14 ]}     |  [MEMory4 | 14 ]}]
```

Example:
```
OUTPUT 707;"SOURCE CHANNEL1;MEMORY1"
```

Query Syntax: { SOURce | SRC }?

Returned Format: [SOURce | SRC]<argument><crlf>

Example:
```
DIM Src$[50]
OUTPUT 707;"SRC?"
ENTER 707;Src$
PRINT Src$
```

TDELta?

query

This query returns the time difference between the start and stop time markers, that is;

Tdelta = Tstop - Tstart

Where Tstart is the time at the start marker and Tstop is the time at the stop marker.

Query Syntax: TDELta?

Returned Format: [TDELta]<NR3><crlf>

```
Example:  DIM Td$[30]
          OUTPUT 707;"TDELTA?"
          ENTER 707;Td$
          PRINT Td$
```

TOPBase?

query

This query returns the signal amplitude using the formula:

amplitude = Vtop - Vbase

Vtop and Vbase are located using a histogram of the voltage values of the waveform record. After a waveform record is collected the absolute min and max voltages are determined and a histogram of the voltage values is completed. Next, the waveform record is scanned to find the voltage values with the largest number of data points. If the maximum number of data points is greater than the limit criteria (approximately 5% of the maximum number of points in the record) that voltage level is used for the top or the base. If the limit criteria is not satisfied the absolute min, max values are used as the base and the top.

Query Syntax: TOPbase?

Returned Format: [TOPBase]<NR3><crlf>

```
Example:  DIM Top$[30]
          OUTPUT 707;"TOPBASE?"
          ENTER 707;Top$
          PRINT Top$
```

TSTArt

command/query

This command moves the start marker to the specified time with respect to the trigger time. The query returns the start marker position.

Command Syntax: TSTArt<start marker time>

```
Example:  OUTPUT 707;"TSTArt -.001"
```

Query Syntax: TSTArt?

```
Returned Format:  [TSTArt]<NR3><crlf>

Example:  DIM Ts$[30]
Example:  OUTPUT 707;"TSTART?"
          ENTER 707;Ts$
          PRINT Ts$
```

TSTOp

command/query

This command moves the stop marker to a specified time with respect to the trigger. The query returns the stop marker position.

Command Query: TSTOp<stop marker time><crlf>

```
Example:  OUTPUT 707;"TSTOP -1.0E-6"
```

Query Syntax: TSTOp?

```
Returned Format:  [TSTOp]<NR3><crlf>

Example:  DIM Ts$[30]
          OUTPUT 707;"TSTOP?"
          ENTER 707;Ts$
          PRINT Ts$
```

TVOLt?

query

When the TVOLt query is sent, the on screen signal is searched for the defined voltage level and transition. The time interval between the trigger event and this defined occurrence is returned as the response to this query.

The sign of <slope & occurrence> selects a rising(+) or falling(-) edge. The magnitude of this parameter defines the number of occurrences. For example, if <slope & occurrence> = +3 the on screen signal would be searched for the third occurrence of the specified voltage on a positive slope.

Query Syntax: TVOLt<voltage><,><slope & occurrence>?

Returned Format: [TVOLt]<NR3><crlf>

Example:
```
DIM Tvolt$[30]
OUTPUT 707;"TVOLT -.250,+3 ?"
ENTER 707;Tvolt$
PRINT TVolt$
```

VBASe?

query

This query returns the voltage level of the base of the waveform data. VBASe is determined by using a histogram of the voltage values of the waveform record. After a waveform record is collected the absolute min and max voltages are determined and a histogram of the voltage values is completed. Next, the waveform record is scanned to find the voltage values with the largest number of data points. If the maximum number of data points is greater than the limit criteria (approximately 5% of the maximum number of points in the record) that voltage level is used for the top or the base. If the limit criteria is not satisfied the absolute min, max values are used as the base and the top.

Query Syntax: VBASe?

Returned format: [VBASe]<NR3><crlf>

Example:
```
OUTPUT 707;"VBASE?"
ENTER 707;Base$
PRINT Base$
```

VDELta? query

This query returns the difference in voltage between voltage marker 1 & 2. That is:

VDELta = Marker2 - Marker1

Where Marker1 is the voltage at marker 1 and Marker2 is the voltage at marker 2.

Query Syntax: VDELta?

Returned Format: [VDELta]<NR3><crlf>

```
Example:  OUTPUT 707;"VDELta?"
          ENTER 707;Vdelta$
          PRINT Vdelta$
```

VFIFty command

For a single source this command sets the voltage markers at the 50% level. For dual source measurements Vmarker1 is sent to the 50% level of the first source and Vmarker 2 is set to the 50% level of the second source. For a single source, this command has the same effect as pressing the front panel Auto Top-Base key and then pressing the 50-50% key.

Command Syntax: VFIFty

```
Example:  OUTPUT 707;"VFIFTY"
```

VMAX? query

This query returns the absolute maximum voltage present at the selected source.

Query Syntax: VMAX?

Returned Format: [VMAX]<NR3><crlf>

```
Example:  OUTPUT 707;"VMAX?"
          ENTER 707;Vmax$
          PRINT Vmax$
```

VMIN?

query

This query returns the minimum voltage present on the selected source.

Query Syntax: VMIN?

Returned Format: [VMIN]<NR3><crlf>

```
Example:  OUTPUT 707;"VMIN?"
          ENTER 707;Vmin$
          PRINT Vmin$
```

VPP?

query

This query returns the peak-to-peak voltage computed using the formula:

Vpp = Vmax - Vmin

Where Vmax and Vmin are the maximum and minimum voltages present on the selected source.

Query Syntax: VPP?

Returned Format: [VPP]<NR3><crlf>

```
Example:  OUTPUT 707;"VPP?"
          ENTER 707;Vpp$
          PRINT Vpp$
```

VRELative

command/query

The VRELative command moves the voltage markers to defined percentage points of their last established positions. For example: after a TOPBase operation voltage marker 1 would be located at the base (0%) of the signal and voltage marker 2 would be at the top (100%) of the signal. If VRELative 10 command was executed, voltage marker 1 would be moved to the 10% level and voltage marker 2 would be moved to the 90% level the signal. VREL 100 would move the markers back to their original locations. VREL 50 would move both markers to the 50% point of their original positions. The query returns the current relative position of the markers i.e., 10, 20, 50, or 100.

Command Syntax: VRELative<percentage>

```
Example:  OUTPUT 707;"VRELative 20"
```

Query Syntax: VRELative?

```
Returned Format:  [VRELative]<NR1>

Example:  OUTPUT 707;"VREL?"
          ENTER 707;Vr$
          PRINT Vr$
```

VRMS? query

This query returns the RMS voltage of the selected SOURce. The RMS voltage is computed over one complete period using the formula:

$$V_{rms} = \left[\frac{1}{n} \sum_{j=1}^{j=n} V_j^2 \right]^{\frac{1}{2}}$$

M5410001

Where there are n time buckets in 1 period and Vj is the voltage at bucket j of the period data. Since it is rare for a period to fall precisely within an integral number of time buckets, the algorithm rounds to the nearest time bucket at the beginning and end and uses these as the limits.

Query Syntax: VRMS?

Returned Format: [VRMS]<NR3><crlf>

```
Example:  OUTPUT 707;"VRMS?"
          ENTER 707;V$
          PRINT V$
```

VSTArt command/query

This command moves voltage marker 1 to the specified voltage. The query returns the current voltage level of voltage marker 1.

Command Syntax: VSTArt<voltage level>

```
Example:  OUTPUT 707;"VSTART -.01"
```

Query Syntax: VSTArt?

Returned Format: [VSTArt]<NR3><crlf>

```
Example:  OUTPUT 707;"VSTART?"
          ENTER 707;Vs$
          PRINT Vs$
```

VSTOp

command/query

This command moves voltage marker 2 to the specified voltage. The query returns the current voltage level of voltage marker 2.

Command Syntax: VSTOp<voltage level>

```
Example:  OUTPUT 707;"VSTOP -.1"
```

Query Syntax: VSTOp?

```
Returned Format:  [VSTOp]<NR3><crlf>

Example:  OUTPUT 707;"VSTOP?"
          ENTER 707;Vstop$
          PRINT Vstop$
```

VTIMe?

query

This query returns the voltage at a time, this time is referenced to the trigger event and must be on screen. The time may be + or - (before or after the trigger event). This command functions on single valued waveform records only. If the time with respect to the trigger event is off screen 1E38 will be returned. If the time bucket of interest does not contain any voltage values, due to the completion criteria being less than 100%, the VTIMe value will be interpolated using linear interpolation between the closest points before and after the time bucket.

Query Syntax: VTIMe?

```
Returned Format:  [VTIMe]<NR3><crlf>

Example:  OUTPUT 707;"VTIME -.001?"
          ENTER 707;Vt$
          PRINT Vt$
```

VTOP?

query

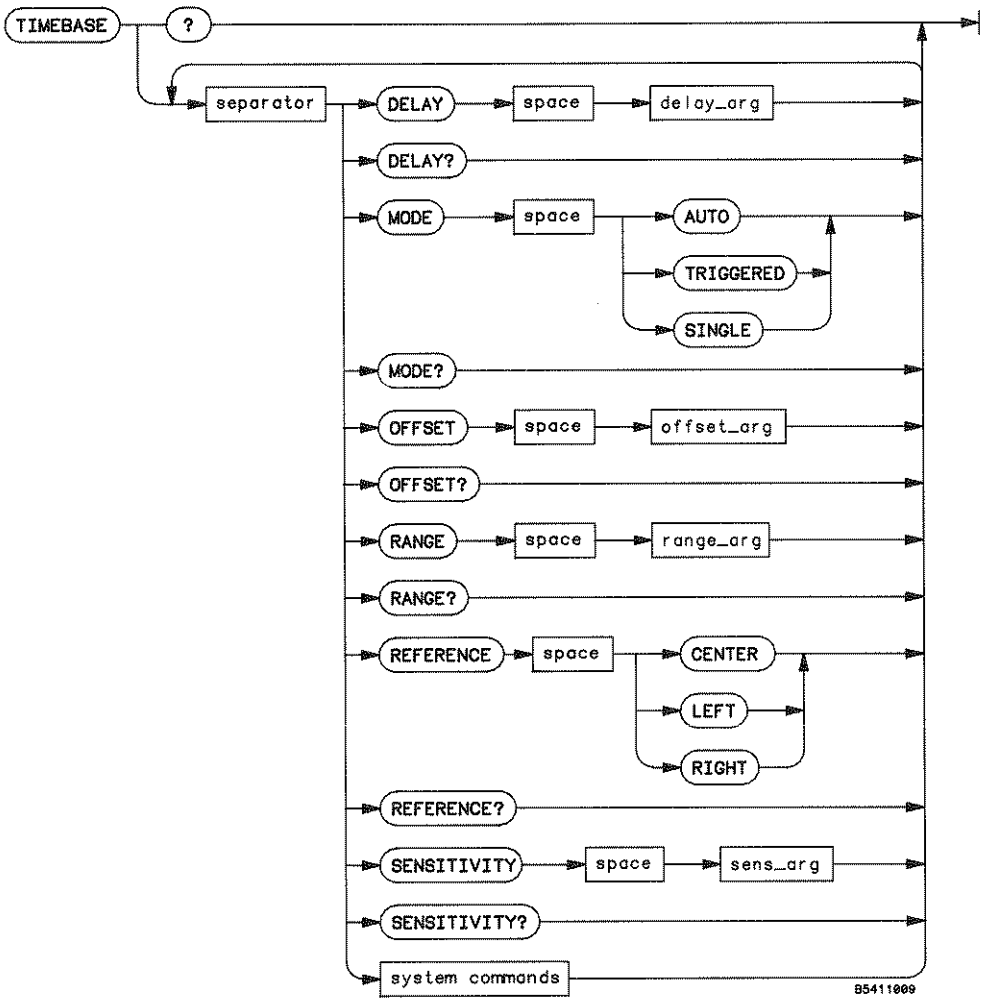
This command returns the voltage at the top of a waveform. VTOP is calculated by using a histogram of the voltages of the waveform record. After a waveform record is collected the absolute min and max voltages are determined and histogram of the voltage values is completed. Next, the waveform is scanned to find the voltage values with the largest number of data points. If the maximum number of data points is greater than the limit criteria (approximately 5%) of the maximum number of data points in the record, that voltage level is used for the top or the base. If the limit criteria is not satisfied the absolute min, max values are used as the base and the top.

Query Syntax: VTOP?

Returned format: [VTOP]<NR3><crlf>

Example:
```
OUTPUT 707;"VTOP?"
ENTER 707;Vtop$
PRINT Vtop$
```

NOTES:



RANGE_ARG = A real number with a range of 1 ns to 10 sec.

DELAY_ARG = A real number with the following restrictions:

```
Maximum delay is 60,000 X (timebase range) or 1.6 sec
   whichever is larger.
If the delay reference is left
   then minimum delay is 200 ms or -(timebase range)
   whichever is smaller.
Else if the delay reference is center
   then minimum delay is -5(timebase range) or
   -200 ms+5(timebase range) whichever is smaller.
Else if the delay reference is right
   then minimum delay is 0 or -200 ms+10(timebase range).
```

OFFSET_ARG = Same as DELAY_ARG.

SENS_ARG = A real number between 100 ps and 1 sec.

Figure 10-11. Timebase Subsystem Commands

10-17. TIMEBASE SUBSYSTEM

The Timebase Subsystem commands control the horizontal axis, "X axis", oscilloscope functions. See Figure 10-11 for a syntax diagram of the timebase subsystem commands.

TIMebase — command/query

The TIMbase command selects the timebase as the destination for the commands that follow. The query responds with all the settings of the timebase.

Command Syntax: TIMebase

```
Example:  OUTPUT 707;"TIMEBASE"
```

Query Syntax: TIMebase?

```
Returned Format:  [TIMebase]<crlf>
                  [MODE]<argument><crlf>
                  [RANGe]<NR3><crlf>
                  [DELay]<NR3><crlf>
                  [REFerence]<NR3><crlf>

Example:  DIM Time$[100]
          OUTPUT 707;"EOI ON"
          OUTPUT 707;"TIMEBASE?"
          ENTER 707 USING "-K";Time$
          PRINT USING "K";Time$
```

DELay | DLY — command/query

This command sets the timebase delay. This delay is the time interval between the trigger event and the on screen delay reference point. The query returns the current delay value.

Command Syntax: {[DELay][DLY]}<timebase delay>

```
Example:  OUTPUT 707;"DELAY 2E-3"
```

Query Syntax: {[DELay][DLY]}?

```
Returned Format:  [DELay]<NR3><crlf>

Example:  OUTPUT 707;"DELAY?"
          ENTER 707;Del$
          PRINT Del$
```

MODE

command/query

This command selects the timebase mode. If the AUTO mode is selected the unit will provide a baseline on the display in the absence of a trigger. If a signal is present but the instrument is not triggered the display will be unsynchronized but will not be a baseline. If the TRIGGERED mode is selected and no trigger is present the unit will not sweep, and the data acquired on the previous trigger will remain on-screen. The SINGLE mode causes the unit to make a single acquisition when the next trigger event occurs. The query returns the current mode.

Command Syntax: MODE{[AUTO | 0]
[TRIGGERED | 1]
[SINGLE | 2]}

Example: OUTPUT 707;"MODE SINGLE"

Query Syntax: MODE?

Returned Format: [MODE]<argument><crlf>

Example: OUTPUT 707;"MODE?"
ENTER 707;Mode$
PRINT Mode$

OFFSet

command/query

This command sets the timebase delay. This delay is the time interval between the trigger event and the on screen delay reference point. The query returns the current delay value. This command performs exactly the same function as the DELay command.

Command Syntax: OFFSet<timebase delay>

Example: OUTPUT 707:"OFFSET 1E-4"

Query Syntax: OFFSet?

Returned Format: [OFFSet]<NR3><crlf>

Example: OUTPUT 707;"OFFSET?"
ENTER 707;Offs$
PRINT Offs$

RANGe

command/query

This command defines the full scale horizontal time interval. RANGE = 10 X SENSITIVITY. The query returns the current range.

Command Syntax: RANGe<horizontal time interval>

Example: OUTPUT 707;"RANGE 1"

Query Syntax: RANGe?

Returned Format: [RANGe]<NR3><crlf>

```
Example:  OUTPUT 707;"RANGE?"
          ENTER 707;Range$
          PRINT Range$
```

REFerence

command/query

This command sets the delay reference to the left, center, or right side of the screen. The query returns the current delay reference.

```
Command Syntax:  REFerence {[ LEFT | 0 ]
                           [ CENTER | 1 ]
                           [ RIGHT | 2 ]}
```

Example: OUTPUT 707;"REFERENCE LEFT"

Query Syntax: REFerence?

Returned Format: [REFerence]<argument><crlf>

```
Example:  OUTPUT 707;"REFERENCE?"
          ENTER 707;Ref$
          PRINT Ref$
```

SENSitivity

command/query

This command sets the horizontal time/division to the defined value.
SENSitivity = RANGe/10. The query returns the current time/division.

Command Syntax: SENSITIVITY<time/division>

Example: OUTPUT 707;"SENSITIVITY 1E-7"

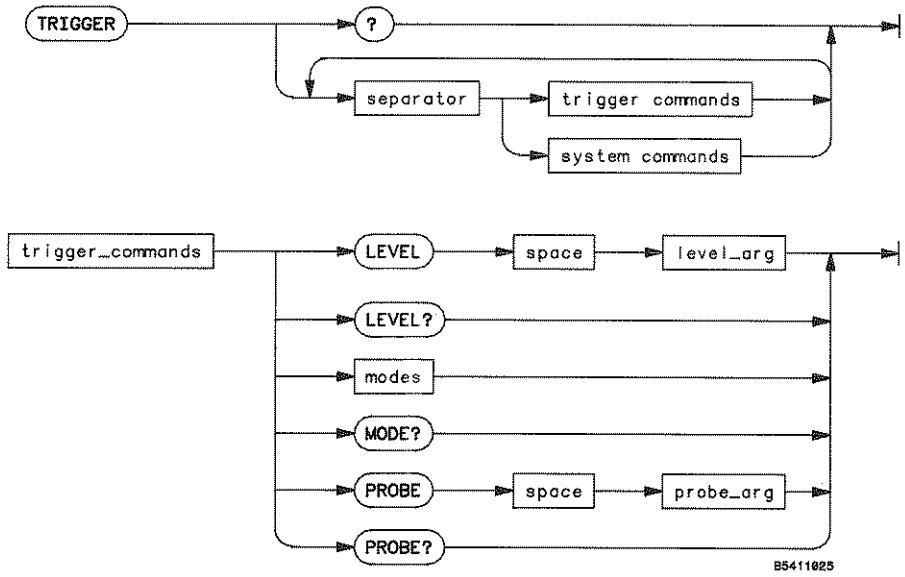
Query Syntax: SENSitivity?

Returned Format: [SENSitivity]<NR3><crlf>

```
Example:  OUTPUT 707;"SENSITIVITY?"
          ENTER 707;Sens$
          PRINT Sens$
```

NOTES:

NOTES:



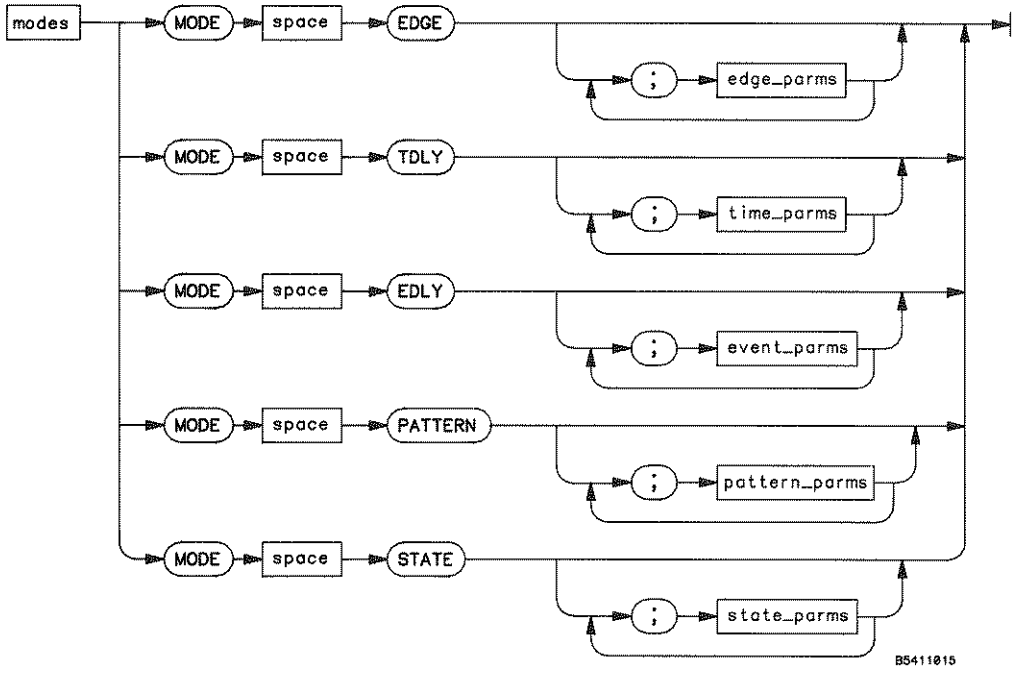
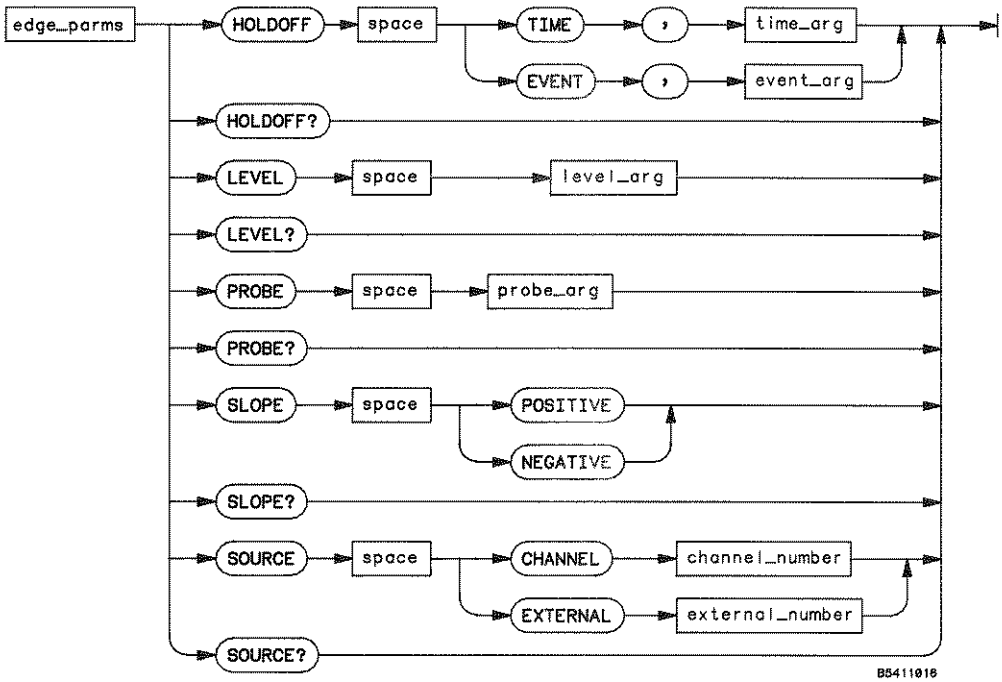


Figure 10-12 Trigger Subsystem Commands



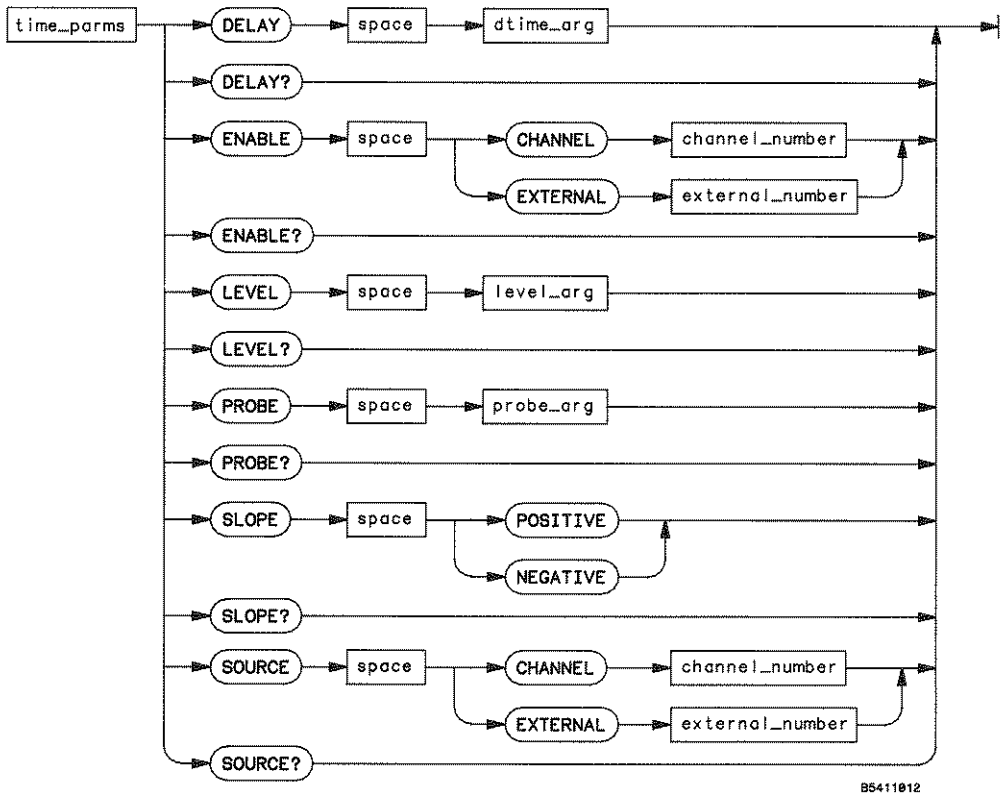


Figure 10-12 Trigger Subsystem Commands

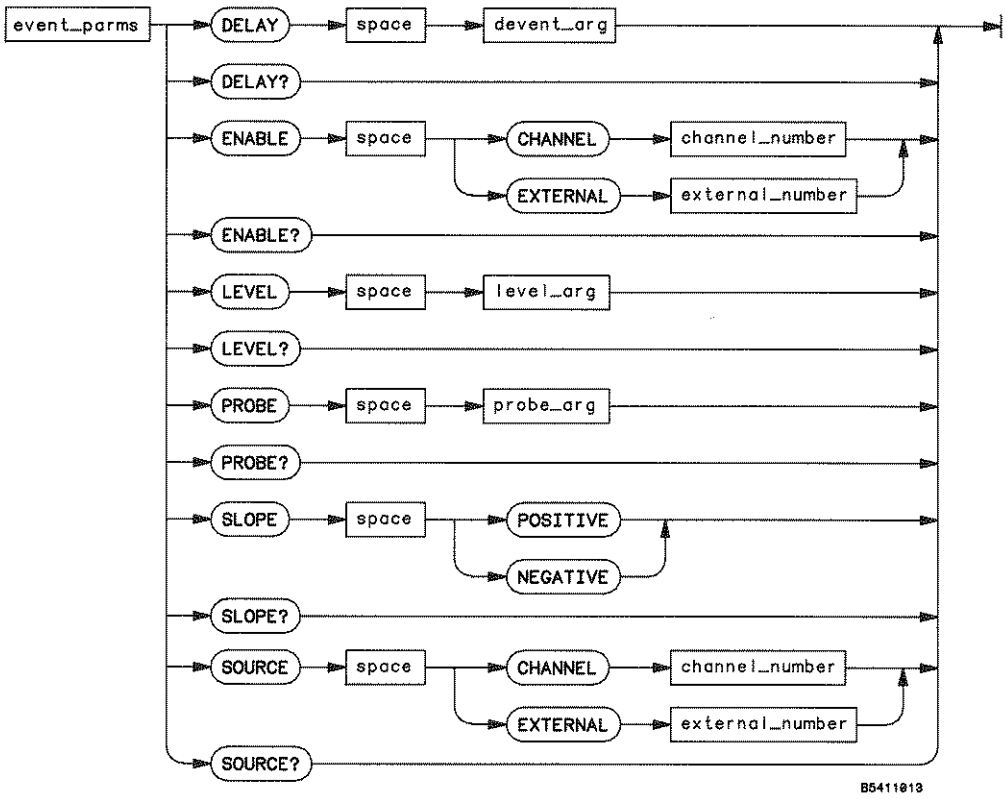


Figure 10-12 Trigger Subsystem Commands

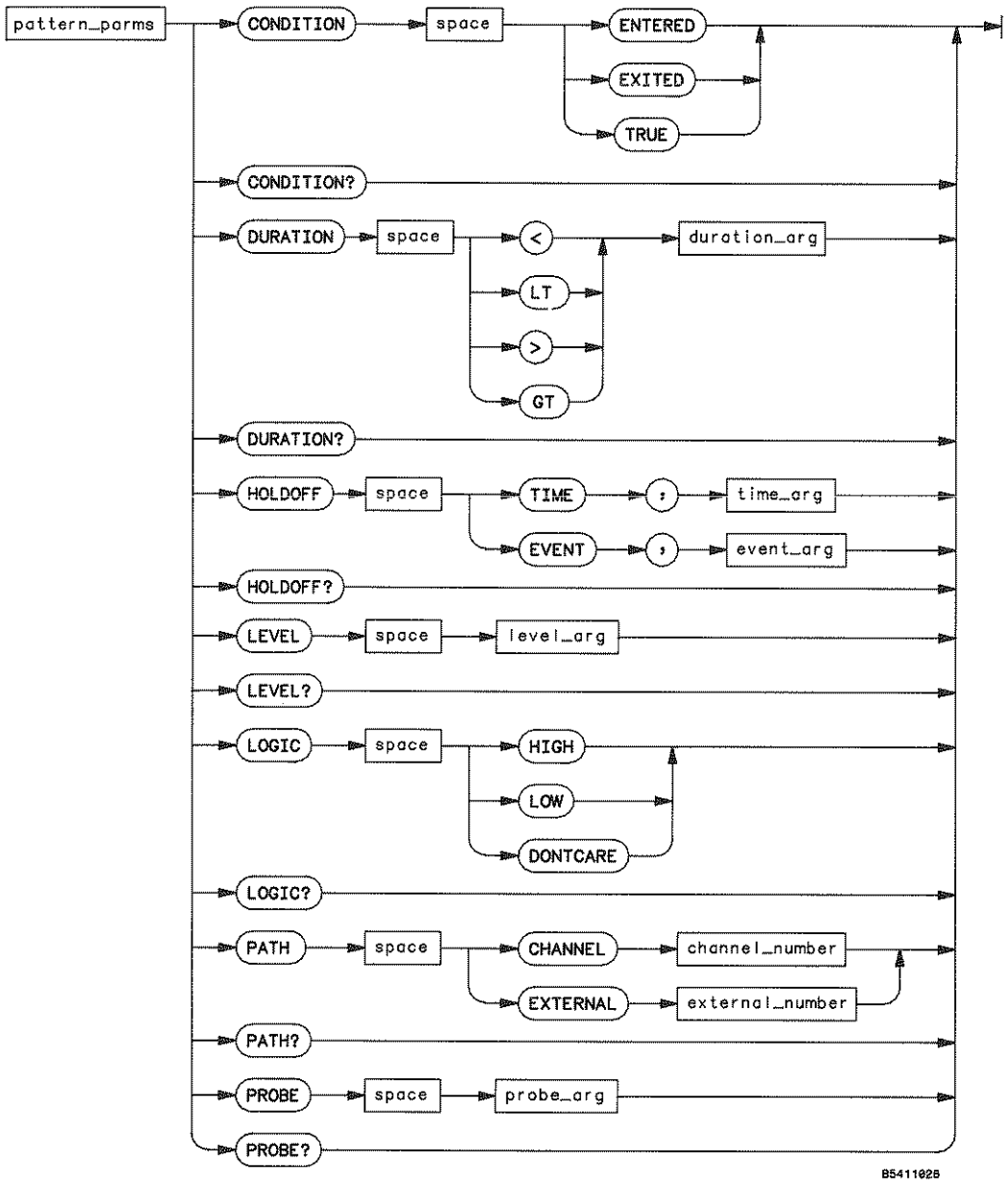


Figure 10-12 Trigger Subsystem Commands

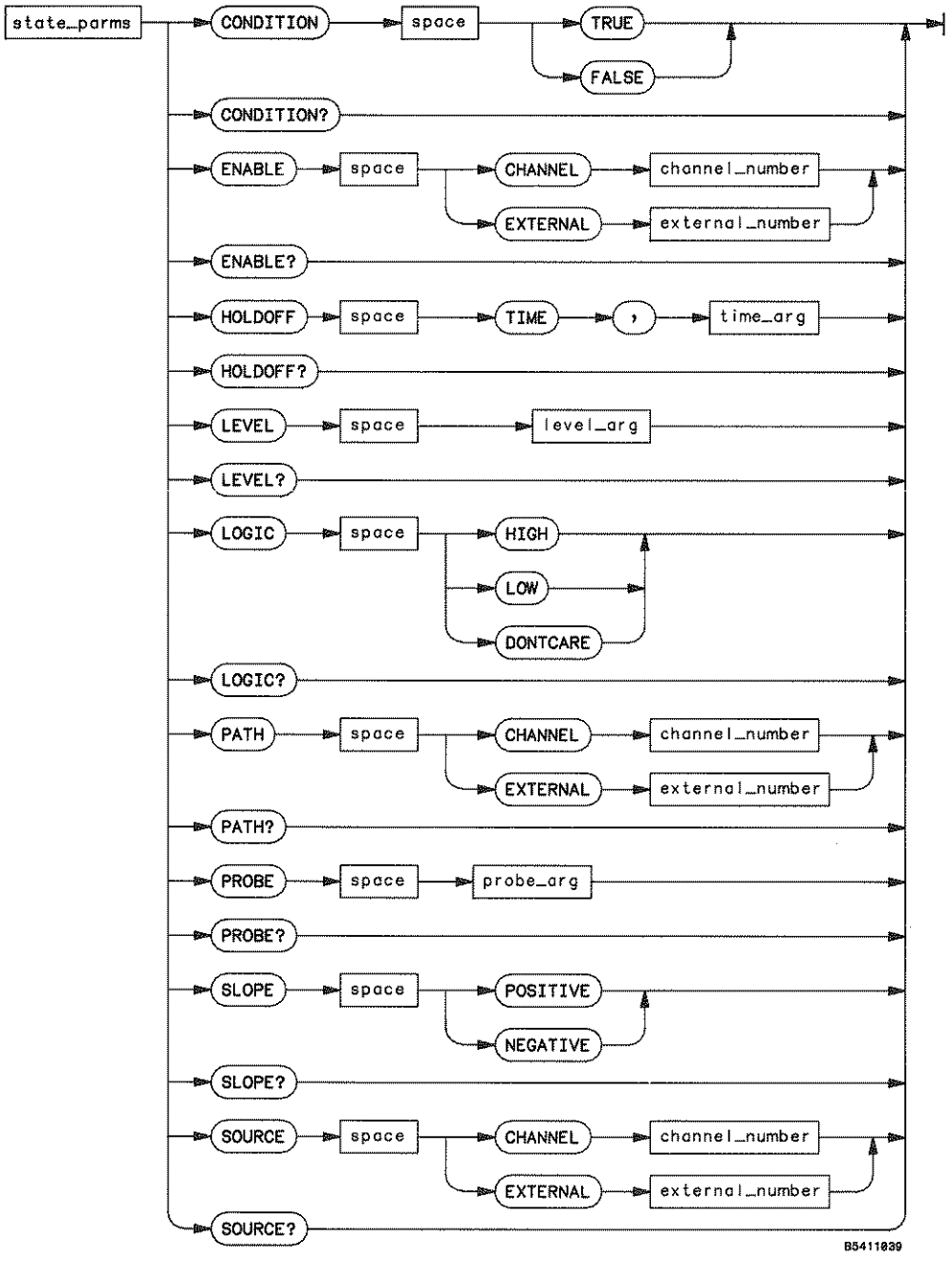


Figure 10-12 Trigger Subsystem Commands

PROBE_ARG = A real number from 1 to 1000.

LEVEL_ARG = A real number <=2 X voltage range.

CHANNEL_NUMBER = An integer 1 or 2.

EXTERNAL_NUMBER = An integer 1 or 2 for the 54100D, and 1 for the 54100A.

TIME_ARG = A real number from 70 ns to 670 ms.

EVENT_ARG = An integer from 2 to 67,000,000.

DURATION_ARG = A real number from 10 ns to 5 seconds.

DTIME_ARG = A real number 20 ns to 5 seconds.

DEVENT_ARG = An integer from 1 to 99,999,999.

Figure 10-12. Trigger Subsystem commands (continued).

NOTES:

10-18. TRIGGER SUBSYSTEM

The commands in the Trigger Subsystem are used to define the conditions for a trigger. The 54100A provides two trigger modes; EDGE mode and the PATTERN mode. The 54100D provides three additional trigger modes: EVENT DELAY mode, TIME DELAY mode, and the STATE mode.

In the edge mode each trigger source has an associated level, slope, and probe attenuation factor which are used when it is selected as a trigger source. These levels and probe attenuation factors are applicable to other modes, however, the slope will depend on the particular mode used.

The SOURCE, ENABLE and PATH commands are related in that they select the source for commands like LOGIC or LEVEL, however each is used in a slightly different way. The SOURCE command is used to specify the trigger source for the EDGE, STATE, TDLY, and EDLY modes. This is the source that the actual trigger is generated from. The ENABLE command is used in the TDLY and EDLY modes to specify the source that is used to qualify the trigger. The PATH command is used in the PATTERN and STATE modes to select a pattern element for setup.

Each individual trigger mode keeps track of the last referenced source and it is this source that is addressed by any SLOPE, LOGIC, etc. commands when that mode is re-entered.

See Figure 10-12 for Trigger Subsystem syntax diagram.

TRIGger

command/query

The trigger command selects the trigger subsystem as the destination for the trigger commands that follow. The query responds with the subsystem parameters for the current trigger mode.

Command Syntax: TRIGger

Example: OUTPUT 707;"TRIGGER"

Query Syntax: TRIGger?

Returned Format:

```
MODE EDGE<crlf>
SOURce<path name><crlf>
PROBe<NR1><crlf>
LEVel<NR3><crlf>
SLOPe<argument><crlf>
{[HOLDoff TIME]<NR3><crlf> |
 [HOLDoff EVENTS]<NR1><crlf>}

MODE PATTERN<crlf>
CONDition<argument><crlf>
DURation<argument><crlf>     (54100D only)
PATH<path name 1><crlf>
PROBe<NR3><crlf>
LEVel<NR3><crlf>
LOGic<argument><crlf>
PATH<path name 2><crlf>
PROBe<NR3><crlf>
LEVel<NR3><crlf>
LOGic<argument><crlf>
PATH<path name 3><crlf>
PROBe<NR3><crlf>
LEVel<NR3><crlf>
LOGic<argument><crlf>
PATH<path name 4><crlf>      (54100D only)
PROBe<NR3><crlf>                    "
LEVel<NR3><crlf>                    "
LOGic<argument><crlf>               "
{[HOLDoff TIME]<NR3><crlf> |
 [HOLDoff EVENTS]<NR1><crlf>}
```

(continued on next page)

TRIGger (cont'd)

```
MODE STATE<crlf>
CONDition<argument><crlf>
PATH<path name 1><crlf>
PROBe<NR3><crlf>
LEVel<NR3><crlf>
LOGic<argument><crlf>
PATH<path name 2><crlf>
PROBe<NR3><crlf>
LEVel<NR3><crlf>
LOGic<argument><crlf>
PATH<path name 3><crlf>
PROBe<NR3><crlf>
LEVel<NR3><crlf>
LOGic<argument><crlf>
SOURce<path name 4><crlf>
PROBe<NR3><crlf>
LEVel<NR3><crlf>
SLOPe<argument><crlf>
HOLDoff TIME<NR3><crlf>

MODE TDLY<crlf>
ENABle<path name 1><crlf>
PROBe<NR3><crlf>
LEVel<NR3><crlf>
SLOPe<argument><crlf>
DELay<NR3><crlf>
SOURCE<path name 2><crlf>
PROBe<NR3><crlf>
LEVel<NR3><crlf>
SLOPe<argument><crlf>

MODE EDLY<crlf>
ENABle<path name 1><crlf>
PROBe<NR3><crlf>
LEVel<NR3><crlf>
SLOPe<argument><crlf>
DELay<NR1><crlf>
SOURCE<path name 2><crlf>
PROBe<NR3><crlf>
LEVel<NR3><crlf>
SLOPe<argument><crlf>
```

(All responses on this page are for the 54100D)

(continued on next page)

TRIGger (cont'd)

```
Example:  DIM Trig$[700]
          OUTPUT 707;"EOI ON"
          OUTPUT 707;"TRIGGER?"
          ENTER USING "-K";Trig$
          PRINT USING "K";TRIG$
```

CONDition

command/query

This command/query is valid only when the trigger mode is PATTERN or STATE. In the PATTERN mode it specifies whether the trigger is generated on entry to the specified logic pattern or when exiting it. If TRUE is selected, DURATION is used as the qualifier for the PATTERN. In the STATE mode it specifies whether the trigger is generated when the pattern is present (true) or not present (false). The query returns the currently selected condition.

Command Syntax: CONDition

```
Example:  OUTPUT 707;"CONDITION EXIT"
```

Query Syntax: CONDition?

```
Returned Format:  [CONDition]{[ ENTER | 0 ]   (PATTERN mode)
                              [ EXIT  | 1 ]
                              [ TRUE  | 2 ]}

                  [CONDition]{[ FALSE | 0 ]   (STATE mode)
                              [ TRUE  | 1 ]}


Example:  OUTPUT 707;"CONDITION?"
          ENTER 707;Cond$
          PRINT Cond$
```

DELay | DLY

command/query

This command/query is valid only in the events delay (EDLY) or time delay (TDLY) modes (54100D only). In the time delay mode this command specifies the delay in seconds. In the events delay mode this command specifies the number of trigger events. The query returns the delay for the current mode.

Command Syntax: {DELay | DLY}{<event delay>|<time delay>}

```
Example:  OUTPUT 707;"DELAY 10"
```

Query Syntax: DELay | DLY?

```
Returned Format:  [DELay]{<NR1>|<NR3>}<crlf>

Example:  OUTPUT 707;"DELAY?"
          ENTER 707;Delay$
          PRINT Delay$
```

DURation

command/query

This command/query is valid only in the pattern mode (54100D only). It specifies the time limit (minimum time for ">", or maximum time for "<") a pattern must be present to generate a trigger. Pattern duration trigger is implicitly an "EXITED" condition, that is, the trigger coincides with the first event that makes the pattern false. The query returns the current selections for duration type and time.

Command Syntax:

```
DURation{[ LT | 1 | < ]
        [ GT | 2 | > ]}

Example:  OUTPUT 707;"DURATION LT 1E-3"
```

Query Syntax: DURation?

```
Returned Format:  [DURation]<argument><crlf>

Example:  OUTPUT 707;"DURATION?"
          ENTER 707;Dur$
          PRINT Dur$
```

ENABle

command/query

This command/query is valid in the STATE, TDLY, or EDLY modes (54100D only). It is used to specify the source that is to be used as the trigger enable, which is also the source for subsequent SLOPE and PROBE commands. The query returns the current trigger enable source of the present mode.

Command Syntax: ENABle{[CHANnel1 | 1]
[CHANnel2 | 2]
[EXTernal1 | 3]
[EXTernal2 | 4]}

Example: OUTPUT 707;"ENABLE CHANNEL3"

Query Syntax: ENABle?

Returned Format: [ENABle]<argument><crlf>

```
Example: OUTPUT 707;"ENABLE?"
         ENTER 707;Enable$
         PRINT Enable$
```

HOLDoff

command/query

This command allows you to specify the holdoff time in the EDGE, PATTERN, or the STATE modes, or the holdoff number of events in the EDGE or PATTERN modes. Each mode has its own holdoff parameters and "remembers" whether it was using holdoff by time or holdoff by events. The holdoff query is valid in the EDGE, PATTERN, or STATE modes and returns the current holdoff setting for the presently selected mode. Time holdoff ranges from 70 ns to 670 ms. Events holdoff ranges from 2 to 67,000,000 events.

Command Syntax: HOLDoff {[EVENT<# of events>]
[TIME<holdoff time>]}

Example: OUTPUT 707;"HOLDOFF EVENT 100"

Query Syntax: HOLDoff?

Returned Format: [HOLDoff]<current holdoff mode>
<holdoff value><crlf

```
Example: DIM Hold$[40]
         OUTPUT 707;"HOLDOFF?"
         ENTER 707;Hold$
         PRINT Hold$
```

LEVel | LVL

command/query

This command sets the trigger level of the selected SOURCE or PATH. The query returns the trigger level of the selected SOURCE or PATH.

Command Syntax: {[LEVel | LVL]} <trigger level>

```
Example:  OUTPUT 707;"LEVEL .1"
```

Query Syntax: {[LEVel | LVL]}?

```
Returned Format:  [LEVel]<NR3><crlf>

Example:  OUTPUT 707;"LEVEL?"
          ENTER 707;Level$
          PRINT Level$
```

LOGic

command/query

This command/query is valid in the STATE and PATTERN modes. The LOGic command is used to specify the relation between the signal and the predefined voltage level that must exist before, that part of the pattern is considered valid. If the signal on a selected source or path is greater than the trigger level that signal is considered HIGH. If it less than the trigger level it is considered LOW. The query returns the last specified logic level of the currently enabled source.

Command Syntax:

```
LOGic{[ LOW | 0 ]
      [ HIGH | 1 ]
      [ DONTCARE | 2 ]}

Example:  OUTPUT 707;"LOGIC DONTCARE"
```

Query Syntax:

```
LOGic?

Returned Format:  [LOGic]<argument><crlf>

Example:  DIM Log$[40]
          OUTPUT 707;"LOGIC?"
          ENTER 707;Log$
          PRINT Log$
```

MODE

command/query

This command allows you to select the trigger mode. The query returns the current trigger mode.

Command Syntax: MODE{[EDGE | 0]
[PATTERN | 1]
[STATE | 2] (54100D only)
[TDLY | 3] "
[EDLY | 4]} "

Example: OUTPUT 707;"MODE EDGE"

Query Syntax: MODE?

Returned Format: [MODE]<argument><crlf>

Example: OUTPUT 707;"MODE?"
ENTER YOU;Mode$
PRINT Mode$

PATH

command/query

This command/query is valid in the PATTERN and STATE modes. This command allows you to select a pattern bit as the source for future probe and logic commands. The query returns the current trigger source of the present mode.

Command Syntax: PATH {[CHANNEL1 | CHAN1 | 1]
[CHANNEL2 | CHAN2 | 2]
[EXTERNAL1 | EXT1 | 3]
[EXTERNAL2 | EXT2 | 4]} (54100D only)

Example: OUTPUT 707;"PATH CHANNEL3"

Query Syntax: PATH?

Returned Format: [PATH]<argument><crlf>

Example: DIM Path$[30]
OUTPUT 707;"PATH?"
ENTER 707;Path$
PRINT Path$

PROBe

command/query

This command specifies the attenuation factor for the last specified SOURCE or PATH for the current trigger mode. If the trigger source is also a channel, the last specified probe attenuation for that channel is the one used. See the CHANNEL PROBE command in Paragraph 10-10. The query returns the current source's probe attenuation factor.

Command Syntax: PROBe<attenuation ratio>

```
Example:  OUTPUT 707;"PROBE 10"
```

Query Syntax: PROBe?

```
Returned Format:  [PROBe]<NR3><crlf>

Example:  DIM Probe$[30]
          OUTPUT 707;"PROBE?"
          ENTER 707;Probe$
          PRINT Probe$
```

SLOPe

command/query

This command allows you to specify the trigger slope for the previously specified source. The query returns the current slope for the last selected source of the current mode.

Command Syntax:

```
SLOPe {[ NEGATIVE | 0 ]
       [ POSITIVE | 1 ]}

Example:  OUTPUT 707;"SLOPE POSITIVE"
```

Query Syntax:

```
OUTPUT 707;"SLOPE?"

Returned Format:  [SLOPe]<argument><crlf>

Example:  OUTPUT 707;"SLOPE?"
          ENTER 707;Slope$
          PRINT Slope$
```

SOURce | SRC

command/query

This command /query is valid in the EDGE, STATE, TDLY or EDLY modes and is used to specify the trigger source. This command also identifies the source for any subsequent SLOPe and PROBe commands. The query returns the current trigger source of the present mode.

Command Syntax: {[SOURce | SRC]} {[CHANnel1 | 1]
[CHANnel2 | 2]
[EXTernal1 | 3]
[EXTernal2 | 4]} (54100D only)

```
Example:  OUTPUT 707;"SOURCE CHANNEL1"
```

Query Syntax: {[SOURce | SRC]}?

Returned Format: [SOURce]<argument><crlf>

```
Example:  DIM Src$[30]
          OUTPUT 707;"SOURCE?"
          ENTER 707;Src$
          PRINT Src$
```

NOTES:

NOTES:

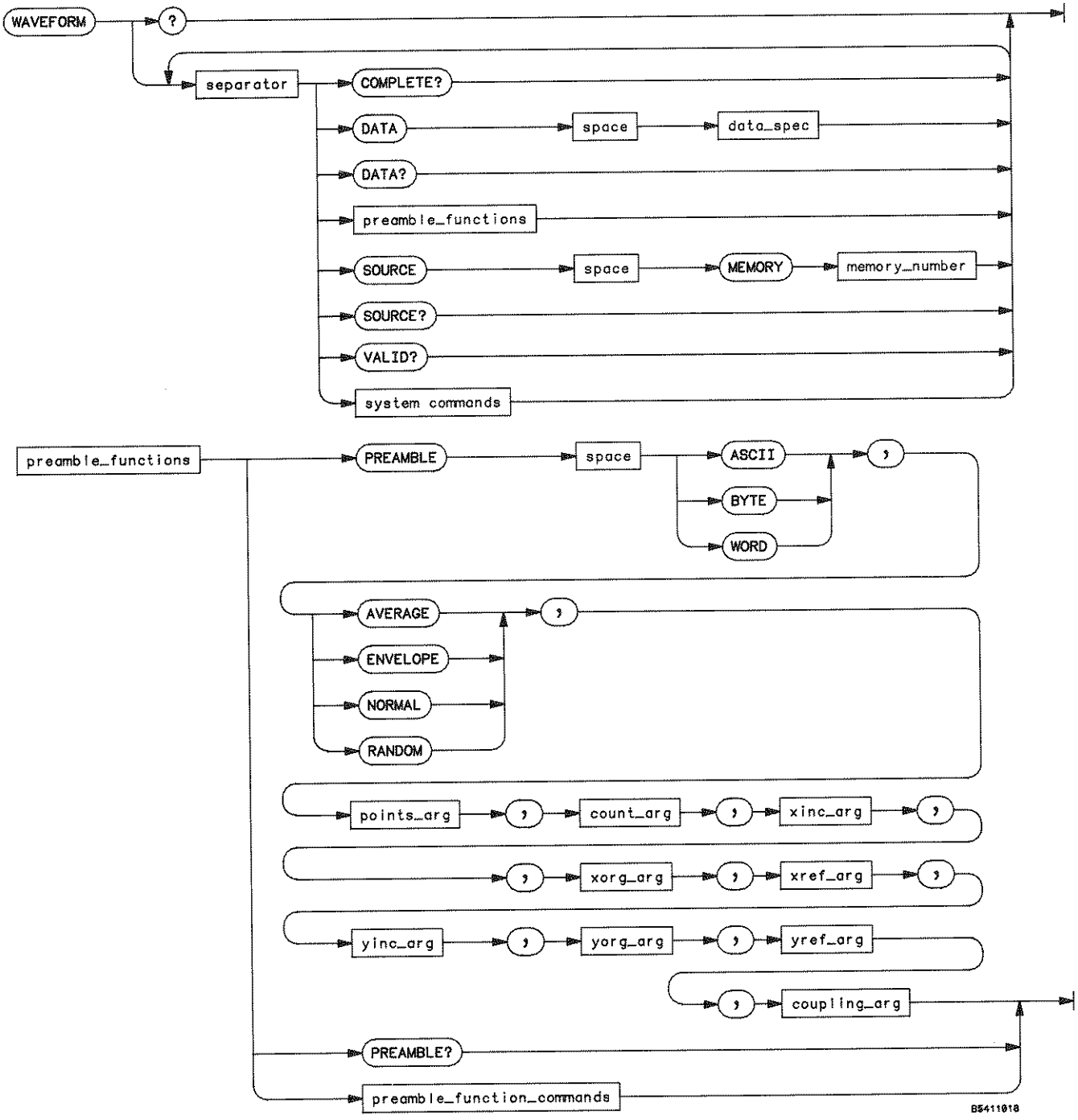


Figure 10-13. Waveform Subsystem Commands

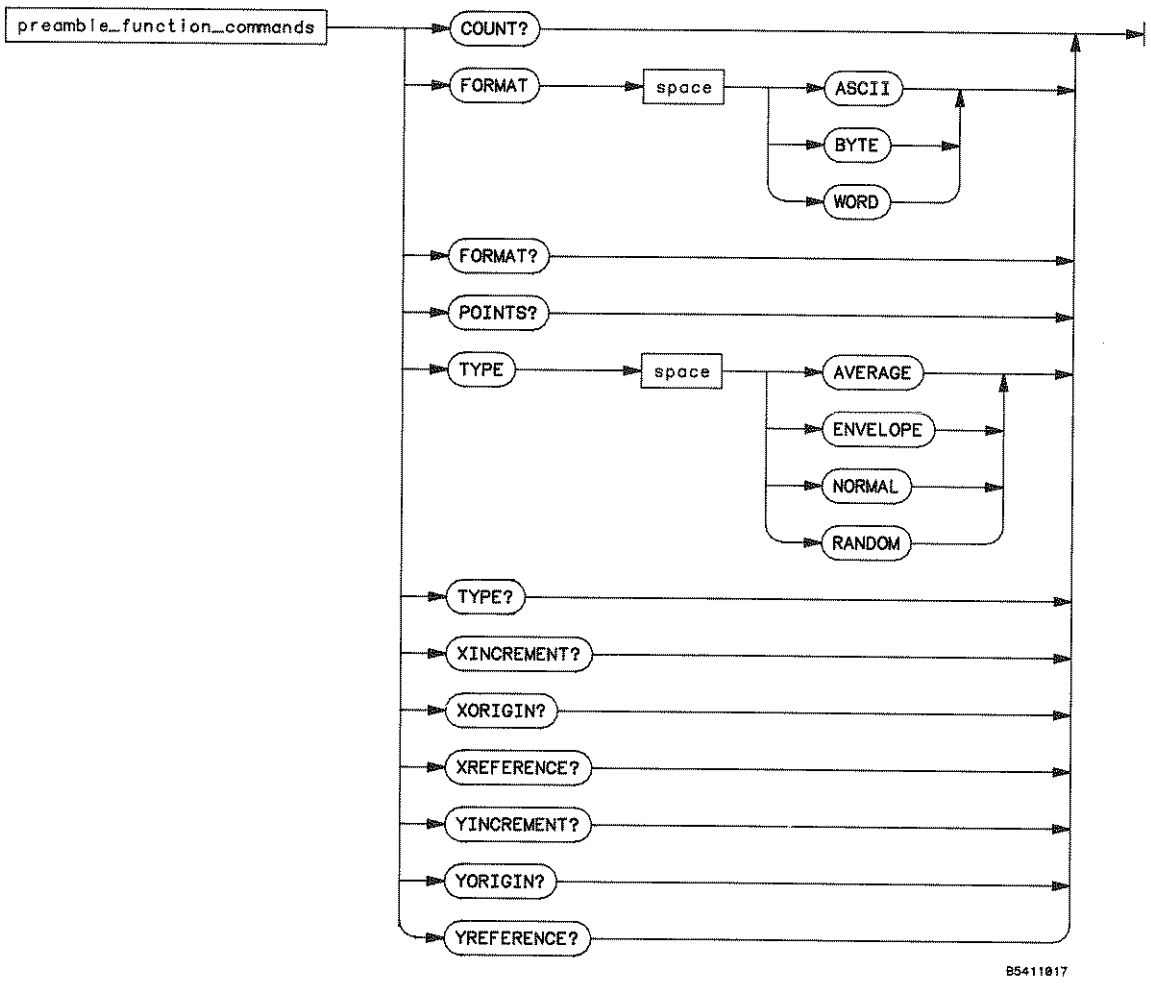


Figure 10-13. Waveform Subsystem Commands

MEMORY_NUMBER = An integer 1 through 4.

DATA_SPEC = A block of data in #A format as defined in IEEE Std. 728-1982.

POINTS_ARG = An integer = 128, 256, 500, 512, 1024.

COUNT_ARG = An integer from 1 to 2048.

XINC_ARG = A real number from 10 ps to 2 ms.

XORG_ARG = A real number with the following restrictions:
The maximum value is 60,000 X timebase range or 1.6 sec, which ever is greater.
If the delay reference is left then the minimum value is -200 ms or -(timebase range) whichever is smaller
Else if the delay reference is center
then the minimum value is the lesser of -5X(timebase range) and -200 ms + (10 X timebase range).
Else if the delay reference is right
then the minimum value is the lesser of 0 and -200 ms+(10Xtimebase range).

XREF_ARG = 0

YINC_ARG = A real number equal to (1/128) X voltage range.

YORG_ARG = A real number with a magnitude less than 1.5 X voltage range.

YREF_ARG = 64 for byte format; 16384 for word or ASCII format.

COUPLING = DCFIFTY or the integer 3.

Figure 10-13. Waveform Subsystem Commands

10-19. WAVEFORM SUBSYSTEM

The waveform subsystem commands are used to transfer waveforms to and from the four waveform memories. Waveform data consists of a preamble and a data record. The preamble contains scaling information useful for interpreting the data record while the data record contains the actual waveform data values.

Each element of the waveform preamble can be individually set or queried. This can cause you problems if not done judiciously, for example, setting POINTS in the preamble to a value different from the actual number of points in the waveform could result in inaccurate data being sent. For this reason only the query form of most of the preamble commands is documented here.

The actual values set in the preamble are determined when the DIGITIZE command is executed and are based on the settings of variables in the ACQUIRE subsystem. For more information on the DIGITIZE process and the ACQUIRE subsystem variables, see Section 10-8. For Syntax diagrams of the waveform subsystem commands see Figure 10-13.

The four waveform types are:

NORMAL:

Normal data consists of the last data point (hit) in each time bucket. This data is transmitted over HP-IB in a linear fashion starting with time bucket 0 and going through time bucket n-1, where n is the number returned by the WAVeform POINts query. Time buckets that don't have data in them return -1. Only the voltage values of each data point are transmitted, the time values correspond to the position in the data array. The first voltage value corresponds to the first time bucket on the left of the CRT and the last value corresponds to the next to last time bucket on the right side of the CRT.

AVERAGE:

Average data consists of the average of the first n hits in a time bucket, where n is the value returned by the WAVeform COUNt query. Time buckets that have fewer than n hits return the average of what data they do have. Again, if a time bucket doesn't have any data in it, it will return -1. This data is transmitted over the HP-IB in a linear fashion starting with time bucket 0 and proceeding through time bucket n-1, where n is the number returned by the WAVeform POINts query. The first value corresponds to a point at the left side of the screen and the last value is one point away from the right side of the screen.

ENVELOPE:

Envelope data consists of two arrays of data, one containing the minimum of the first n hits in each time bucket and the other consisting of the maximum of the first n hits in each time bucket, where n is the value returned by the WAVEform count query. If a time bucket does not have any hits in it then -1 is returned for both the minimum and maximum values. The two arrays are transmitted one at a time over the HP-IB linearly, starting with time bucket 0 (on the left side of the CRT) and proceeding through time bucket m-1, where m is the value returned by the WAVeform POINts query. The array with the minimum values is sent first. The first value of each array corresponds to the data point on the left side of the CRT. The last value is one data point away from the right side of the CRT.

When envelope data is acquired, the minimum and maximum data for channel 1 is stored in memories 1 and 3 respectively and the minimum and maximum data for channel 2 is stored in memories 2 and 4. Memories 1 and 2 are set to the envelope type and memories 3 and 4 are set to the normal type.

The data in the memories is transferred to a controller using the data query. The memory source for the transfer is specified by the waveform source command. When memory 1 is specified as the source, the minimum and maximum data from memories 1 and 3 is transferred over HP-IB, and when memory 3 is specified as the source, only the maximum data in memory 3 is transferred over HP-IB. Similarly, when memory 2 is specified as the source, the data from memories 2 and 4 is transferred over HP-IB, and when memory 4 is specified as the source only the maximum data in memory 4 is transferred over the bus.

If it is desirable to transfer only the data in memory 1 or 2, it can be accomplished by changing the memory 1 or memory 2 type from envelope to normal using the waveform type command.

Data is transferred into the instrument from a controller using the waveform data command. Envelope data can be transferred into memory 1 or memory 2 by specifying the type for the memories as envelope. The data is then transferred as two arrays. If memory 1 is specified as the source, the first array is transferred into memory 1 and the second array is transferred into memory 3. Similarly, if memory 2 is specified as the source, the first array is transferred into memory 2 and the second array is transferred into memory 4.

RANDOM:

Random data consists of all the data that can be gathered, but not to exceed 1024 points. This data is transmitted over the HP-IB in ordered time-voltage pairs. You should not use the BYTE format for this mode since the time bucket numbers range up to 500 and it is impossible to represent numbers this large in a byte without loss of precision. The time data is acquired using 500 time buckets. If ASCII and WORD format are used these time values are multiplied by 64 allowing values from 0 to 32,000. If BYTE format is used the allowed values are from 0 to 125, this demonstrates how precision is lost in the BYTE mode.

The three FORMATS that are used to transmit data over the HP-IB:

WORD:

Word formatted waveform records are transmitted using the binary block format (the #A format specified in IEEE STD. 728-1982). The character string "#A" is sent first, then followed by a two byte length value (16 bit binary) specifying the number of bytes to follow. The number of bytes is twice the number of words. The number of words is also the value returned by the WAVeform POINts query. This is followed by a sequence of bytes representing the data words, with the most significant byte of each word being transmitted first. The A/D conversion in the 54100A/D yields a 7 bit result and is contained in the upper half of the data words transmitted by the instrument. The lower byte contains zeros unless the TYPE was AVERAGE. In this case any increased resolution achieved through averaging will show up in the lower byte of the data. Values are always positive and between 0 and 32,767.

BYTE:

The BYTE formatted waveform records are transmitted over the HP-IB using the binary block format (the #A format specified in IEEE Std. 728-1982). The character string "#A" is sent first, then followed by a two byte length value (16 bit binary) specifying the number of bytes to follow. The number of bytes when the FORMat is BYTE is the same as the value returned by the WAVeform POINts query. BYTE formatted transfers run approximately twice as fast as WORD and ASCII transfers, but should be used with caution if there are any data values to be sent that are larger than decimal 127. If the data values have a larger range than 127 as is the case when TYPE is RANDOM,the values are shifted until they fit within a byte. For example, when TYPE is RANDOM, the X values normally range from 0 to 500. Trying to transmit RANDOM data in BYTE FORMAT results in the time bucket numbers being rescaled so that they range from 0 to 125. This lumps time buckets 0 through 3 into one x coordinate, time buckets 4 through 7 into the next X coordinate, etc.

ASCII:

ASCII formatted waveform records are transmitted one value at a time, separated by <cr><lf>s. The data values transmitted are the same as would be sent in the WORD FORMAT except that they are converted to an integer ASCII format (six characters) before being sent over HP-IB.

The data values can be converted to voltage and time values using the following formulas:

Voltage(j) = [(Yvalue(j) - Yreference)*Yincrement] + Yorigin

Time(j) = (j*Xincrement)+Xorigin (non-RANDOM type)

Time(j) = (Xvalue(j)*Xincrement) + Xorigin (RANDOM type)

Where Yvalue(j) is the value of the jth point and Yreference, Y increment, Yorigin, Xincrement, and Xorigin are the preamble values. In the RANDOM mode Xvalue(j) is the jth time point.

WAVeform

command/query

The WAVeform command selects the waveform subsystem as the destination for the waveform commands that follow. The query returns the waveform subsystem parameters. The COUPLing parameter is always DC or 1 depending on whether argument is set to alpha or numeric.

Command Syntax: WAVeform

```
Example:  OUTPUT 707;"WAVEFORM"
```

Query Syntax: WAVeform?

```
Returned Format:  [WAVeform]<crlf>
                  [SOURce]<specified source><crlf>
                  [VALid]<NR1><crlf>
                  [FORMat]<argument><crlf>
                  [TYPE]<argument><crlf>
                  [POINts]<NR1><crlf>
                  [COUNt]<NR1><crlf>
                  [XINCrement]<NR3><crlf>
                  [XORigin]<NR3><crlf>
                  [XREFerence]<NR1><crlf>
                  [YINCrement]<NR3><crlf>
                  [YORigin]<NR3><crlf>
                  [YREFerence]<NR1><crlf>
                  [COUPling]<argument><crlf>
                  [COMPlete]<NR1><crlf>

Example:  DIM Wav$[300]
          OUTPUT 707;"EOI ON"
          OUTPUT 707;"WAVEFORM?"
          ENTER 707 USING "-K";Wav$
          PRINT USING "K";Wav$
```

COMPlete?

query

This query returns the completion criterion that was used for the last acquisition to the currently selected memory from its preamble.

Query Syntax: COMPlete?

```
Returned Format:  [COMPlete]<NR1><crlf>

Example:  DIM Comp$[30]
          OUTPUT 707;"COMPLETE?"
          ENTER 707;Comp$
          PRINT Comp$
```

COUNt? | CNT?

query

This query returns the count field of the waveform preamble. The count field contains the number of averages if the TYPE is AVERAGED, or if the TYPE is ENVELOPE it contains the number of hits in each time bucket.

Query Syntax: {[COUNt | CNT]}?

Returned Format: [COUNt]<NR1>

```
Example:  OUTPUT 707;"COUNT?"
          ENTER 707;Count$
          PRINT Count$
```

DATA

command/query

This command causes the instrument to accept a waveform data record over the HP-IB from the controller and store it in the previously specified waveform memory. Note: The record format must match the format previously specified for the memory by it's preamble. The query returns the waveform record stored in the previously specified waveform memory.

Command Syntax: DATA

Example: OUTPUT 707;"DATA"

Query Syntax: DATA?

Returned format: [DATA]#A<binary block length in bytes>
<binary block><crlf>

(continued on next page)

DATA (cont'd)

The following program moves data from the 54100A/D to the controller and then back to the 54100A/D using the WAVEFORM DATA command and query. For this example program use the instrument's cal signal and connect it to channel 1.

```
10   CLEAR 707
20   OUTPUT 707;"RESET"
30   OUTPUT 707;"AUTOSCALE"
40   ASSIGN @Fast TO 707;FORMAT OFF
50   OUTPUT 707;"ACQUIRE"
60   OUTPUT 707;"TYPE ENVELOPE COUNT 256 COMPLETE 90"
70   OUTPUT 707;"DIGITIZE 1"
80   OUTPUT 707;"HEADER OFF"
90   OUTPUT 707;"WAVEFORM SOURCE MEMORY1 FORMAT WORD"
100  OUTPUT 707;"DATA?"
110  ENTER 707 USING "#,2A";Header$
120  ENTER 707 USING "#,W";Byte_len
130  Word_len=Byte_len/2
140  ALLOCATE INTEGER waveform(1:Word_len)
150  ENTER @Fast;Waveform(*)
160  DIM Preamble$[120]
170  OUTPUT 707;"LONGFORM OFF"
180  OUTPUT 707;"PREAMBLE?"
190  OUTPUT 707;"SOURCE MEMORY2"
200  ENTER 707;Preamble$
210  OUTPUT 707;"PREAMBLE";Preamble$
220  Header$="DATA #A"
230  OUTPUT 707 USING "#,7A,W";Header$;Byte_len
240  OUTPUT @Fast;Waveform(*)
250  OUTPUT 707;"TRANSFER MEMORY2,PLANE2"
260  OUTPUT 707;"VIEW PLANE2 BLANK CHANNEL1"
270  END
```

FORMat

command/query

This command allows you to set the data transmission mode for the waveform data points. When the ASCII mode is selected the data is sent as ASCII digits with each data value separated by a <crlf>. If the BYTE mode is selected the data is sent as eight bit integers, while a WORD formatted transfer transfers the data as 16 bit integers. The query returns the current transfer format for the previously specified memory.

Command Format; FORMat {[ASCII | 0]
[BYTE | 1]
[WORD | 2]}

Example: OUTPUT 707;"FORMAT WORD"

Query Syntax: FORMat?

Returned Format: [FORMat]<argument><crlf>

Example: OUTPUT 707;"FORMAT?"
ENTER 707;Format$
PRINT Format$

POINts? | PNTS?

query

This query returns the points value in the currently selected waveform preamble, which is the number of points acquired in the last DIGitize command to the selected waveform memory.

Query Syntax: {[POINts | PNTS]}?

Returned Format: [POINts]<NR1><crlf>

Example: OUTPUT 707;"POINTS?"
ENTER 707;Points$
PRINT Points$

PREamble

command/query

This command sends a waveform preamble to the selected waveform memory in the instrument. The query returns the previously specified memory.

Command Syntax: PREamble <preamble block>

```
<preamble block> ::= <format>,
                     <type>,
                     <points>,
                     <count>,
                     <xincrement>,
                     <xorigin>,
                     <xreference>,
                     <yincrement>,
                     <yorigin>,
                     <yreference>,
                     <coupling>
```

Query Syntax: PREamble?

```
Returned Format:  [PREamble]<format parameter>,
                            <type parameter>,
                            <points NR1>,
                            <count NR1>,
                            <xincrement NR3>,
                            <xorigin NR3>,
                            <xreference NR1>,
                            <yincrement NR3>,
                            <yorigin NR3>,
                            <yreference NR1>,
                            <coupling parameter>
```

Example: This example program uses both the command and query form of the key word.

```
10  DIM Points$[120]
20  OUTPUT 707;"HEADER OFF"
30  OUTPUT 707;"WAVEFORM"
40  OUTPUT 707;"PREAMBLE?"
50  ENTER 707;Points$
60  PRINT Points$
70  OUTPUT 707 USING "#,K";"PREAMBLE",Points$
```

SOURce | SRC

command/query

This command selects the memory that is to be used as the source in following waveform commands. The query returns the currently selected source.

Command Syntax: {[SOURce | SRC]} {[MEMory1 | 1]
[MEMory2 | 2]
[MEMory3 | 3]
[MEMory4 | 4]

Example: OUTPUT 707;"SOURCE MEMORY3"

Query Syntax: SOURce?

Returned format: [SOURce]<argument><crlf>

Example: OUTPUT 707;"SOURCE?"
ENTER 707;Src$
PRINT Src$

TYPE

command/query

This command sets the data type for the currently selected memory. The query returns the data type for the previously specified memory.

Command Syntax: TYPE {[INVALID | 0] (query response only)
[NORMal | 1]
[AVErage | 2]
[ENVelope | 3]
[RANDom | 4]}

Example: OUTPUT 707;"TYPE ENVELOPE"

Query Syntax: TYPE?

Returned Format: [TYPE]<argument><crlf>

Example: OUTPUT 707;"TYPE?"
ENTER 707;Type$
PRINT Type$

VALid? query

This query returns 0 if there is not data in the memory. If there is valid data in the previously selected memory the response will be 1.

Query Syntax: VALid?

Returned format; [VALid] ([0 | 1])

```
Example:  OUTPUT 707;"VALID?"
          ENTER 707;Va$
          PRINT Va$
```

XINCrement? query

This query returns the x-increment value currently in the preamble. This value is the time difference between consecutive data points for NORMAL, AVERAGED, or ENVELOPE data.

Query Syntax: XINCrement?

Returned Format: [XINCrement]<NR3><crlf>

```
Example:  DIM Xin$[30]
          OUTPUT 707;"XINCREMENT?"
          ENTER 707;Xin$
          PRINT Xin$
```

XORigin? query

The query returns the x-origin value currently in the preamble. This value is the time of the first data point in the memory with respect to the trigger point.

Query Syntax: XORigin?

Returned Format: [XORigin]<NR3><crlf>

```
Example:  DIM Xor$[30]
          OUTPUT 707;"XORIGIN?"
          ENTER 707;Xor$
          PRINT Xor$
```

XREFerence? query

This query returns the current x-reference value in the preamble. This value specifies the data point that is associated with the x-origin data values.

For example: If the x-reference is 0, the x-origin is -1 μs and the x-increment is 150 ns, then a data point whose x value is 45 would correspond to a time of (45-0)*150 μs + (-1 μs) or 6.749 ms after the trigger.

Query Syntax: XREFerence?

Returned Format: [XREFerence]<NR1><crlf>

```
Example:  DIM Xr$[30]
          OUTPUT 707;"XREFERENCE?"
          ENTER 707;Xr$
          PRINT Xr$
```

YINCrement? query

This query returns the y-increment value currently in the preamble. This value is the voltage difference between consecutive data points.

Query Syntax: YINCrement?

Returned Format: [YINCrement]<NR3><crlf>

```
Example:  DIM Yin$[30]
          OUTPUT 707;"YINCREMENT?"
          ENTER 707;Yin$
          PRINT Yin$
```

YORigin? query

This query returns the y-origin currently in the preamble. This value is the voltage at center screen.

Query Syntax: YORigin?

Returned Format: [YORigin]<NR3><crlf>

```
Example:  DIM Yor$[30]
          OUTPUT 707;"YORIGIN?"
          ENTER 707;Yor$
          PRINT Yor$
```

YREFerence? query

This query returns the current y-reference value in the preamble. This value specifies the data point where the y-origin occurs.

Query Syntax: YREFerence?

Returned Format: [YREFerence]<NR1><crlf>

```
Example:  DIM Yref$[30]
          OUTPUT 707;"YREFERENCE?"
          ENTER 707;Yref$
          PRINT Yref$
```

Note

*For example: if the y reference is 64, and the y-origin is 1.1 V and the y-increment is 150 mV, then a data point whose y value is 93 would corresponds to a voltage of (93 - 64)*150 mV + 1.1 V or 5.45 V.*

NOTES:

NOTES:

10-18. DATA MOVEMENT IN THE 54100A/D

Data in the 54100A/D may be moved inside the instrument between several different memories. Each of these memories has a specific function. These memories are diagrammed in Figure 10-14.

Fast ECL memory stores the data from the Analog to Digital converters for Channel 1 and Channel 2.

The active display memory contains the information that is currently being displayed on the 54100A/D's CRT. This memory is organized as a pixel array of 256 (vertical) by 501 (horizontal) bits or 16K bytes.

The waveform memories are the same waveform memories that are accessible from the front panel. These memories may contain up to 1024 data points. If the ACQuire TYPE is ENVELOPE, minimum and maximum data for channel 1 will be stored in memories 1 and 3 respectively, similarly, minimum and maximum data from channel 2 will be stored in memories 2 and 4.

The pixel memories are used to store copies of pictures from the active display memory. These two memories are organized the same way as the active display memory.

The HP-IB commands that control the movement of data in the 54100A/D are listed below and are shown in Figure 10-14 in the data paths that they control.

RUN causes the 54100A/D to acquire data for the active display memory.

MERGE copies the contents of the active display memory into either pixel memory 5 or 6.

TRANsfer allows the movement of data from any of the 4 waveform memories to either of the pixel memories.

DIGitize instructs the instrument to acquire data based on the conditions setup in the ACQuire subsystem. Data from channel 1 is placed in waveform memory 1 and data from channel 2 is placed in waveform memory 2, unless the ACQuire TYPE is ENVELOPE, then minimum and maximum data from channel 1 is stored in waveform memories 1 and 3 respectively, similarly, minimum and maximum data from channel 2 will be stored in waveform memories 2 and 4.

DATA may be moved to and from the waveform memories with the WAVeform DATA(?) commands.

DATA may also be moved to and from the pixel storage memories or the active display memory with DISPlay DATA(?) commands.

Measurements (MEASure subsystem) are performed using data from the fast ECL memory, when the source is an active channel or function. Measurements are performed using data from the waveform memories when the source is a memory.

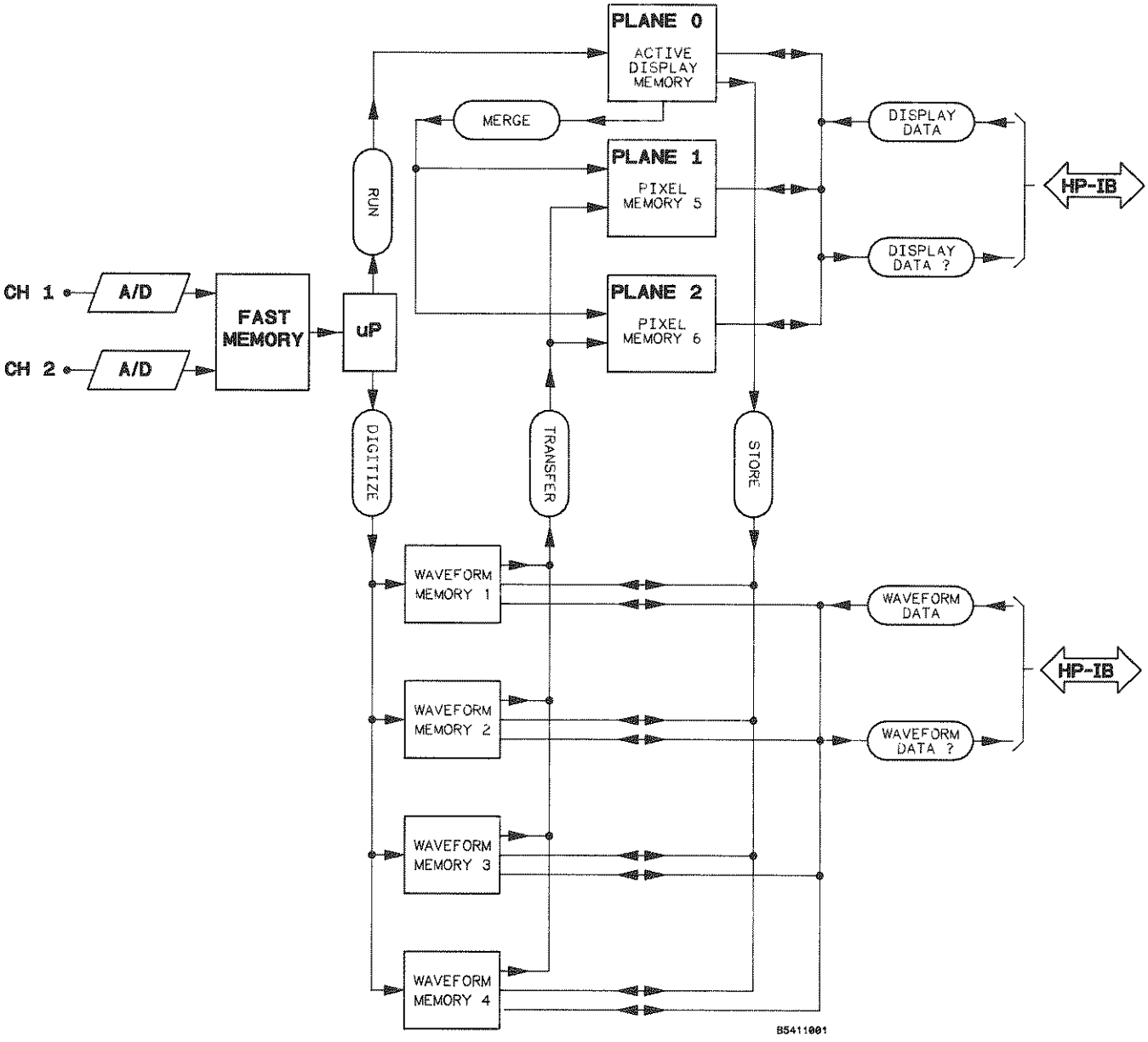


Figure 10-14. Data Flow

Appendix A
Example/Demo Programs

INTRODUCTION

This section contains example programs using the command set for the 54100A/D. In general, they use the longform of the command with alpha, (as opposed to numeric), arguments with each command using a separate output statement for clarity. To optimize speed, switch to concatenated shortform numerics.

Throughout these examples the 54100A/D is assumed to be at address 7, the hardcopy devices are assumed to be at address 1, and the system bus is at 700. The input signal used is the calibration signal available from the rear panel of the instrument connected to channel 1.

All programs were developed on an HP 200 series scientific computer using HP Basic 4.0. Several examples use the BASIC command "ENTER 2". This pauses program execution until the "ENTER" key is depressed on the controller. This is used to separate different blocks in the example for feature dramatizations, for user interaction, or to wait for the 54100A/D to finish something such as a hardcopy dump or an acquisition.

```
10    ! This sample program demonstrates some of the commands
20    ! used to set a vertical channel, in this case channel 1.
30    ! This program works well using the cal signal from the
40    ! rear panel of the instrument.  The program assumes that
45    ! the probe attenuation factor for channel 1 is 1.
50    CLEAR 707                          ! Device clear command,
60    !                                    initializes HP-IB registers.
70    !
80    OUTPUT 707;"AUTOSCALE"             ! Autoscales the unit
90    OUTPUT 707;"CHANNEL1"              ! Enter Ch1 subsystem
100   OUTPUT 707;"OFFSET 0.0"            ! Set offset to 0 volts
110   REAL Offset,Range
120   INTEGER J
130   Offset=0.                          ! Set offset variable to 0
140   FOR J=1 TO 11
150      OUTPUT 707;"OFFSET";Offset      ! Set next offset
160      Offset=Offset-.04
170      WAIT .3
180      NEXT J
190   !
200   !
210   !
```

```
220   OUTPUT 707;"AUTOSCALE"              ! Does as it says
230   OUTPUT 707;"RANGE .080"             ! Set vertical range to 80 mvolts (min)
240   !                                     Can also use "SENSITIVITY" for
250   !                                     volts/div
260    Range=.08                          ! Set range variable to minimum range
270   !                                     or maximum sensitivity
280    FOR J=1 TO 7
290       OUTPUT 707;"RANGE";Range        ! Set new range
300       Range=Range*2
310       WAIT .3
320       NEXT J
330   !
340   !
360    OUTPUT 707;"GRAPH1"                ! Enter GRAPH subsystem
370    OUTPUT 707;"YOFFSET -.1"           ! Set magnified offset
380    OUTPUT 707;"YRANGE 4.0"            ! Set magnified range
390    OUTPUT 707;"MAGNIFY WINDOW"        ! Turns on magnify window
400    Range=4.0
410    FOR J=1 TO 5
420       OUTPUT 707;"YRANGE";Range       ! Set new magnified range
430       Range=Range/1.5                 ! Automatically moves markers
440       WAIT .3                         ! to reflect new range
450       NEXT J
460    !
470    OUTPUT 707;"MAGNIFY ON"            ! Puts unit in the magnify mode.
480    LOCAL 707                          ! Returns the 54100A/D to local
490    END                                ! operation.
```

```
10    ! This is a sample program demonstrating the TIMEBASE
20    ! subsystem.  The rear panel calibration signal works well
30    ! with this program.
40    CLEAR 707                              ! Device clear command
50    !                                        initializes the HP-IB registers.
60    !
70    OUTPUT 707;"AUTOSCALE"                 ! Does as it says.
80    REAL Sens,Delay
90    INTEGER J
100   OUTPUT 707;"TIMEBASE"                  ! Enter TIMEBASE subsytem.
110   OUTPUT 707;"SENSITIVITY 200E-9"        ! Set timebase to 200 nsec/div.
120   !                                        Can also use "RANGE" for
130   !                                        second full screen.
140   !
150   OUTPUT 707;"DELAY 0.0 "                ! Set delay to 0.
160   OUTPUT 707;"REFERENCE LEFT"            ! Puts delay reference at left
170   !                                        side of graticule.
180   !
```

```
190   Delay=0.                                  ! Sets delay to 0.
200   FOR J=1 TO 25
210      OUTPUT 707;"OFFSET ";Delay             ! "OFFSET" = "DELAY".
220      WAIT .23
230      Delay=Delay-1.00E-7
240      NEXT J
250   !
260   !
270   !
280   OUTPUT 707;"AUTOSCALE"                    ! Does as it says.
290   Range=.080                                ! Sets full scale to 80 msec
300   !                                           i.e., 8 ms/div.
310   FOR J=1 TO 25
320       OUTPUT 707;"RANGE";Range              ! Next full scale range.
330       Range=Range/2
340       WAIT .4
350       NEXT J
360   !
370   WAIT 1
380   !
390   !
400   !       WATCH AND SEE HOW DATA IS ACQUIRED
410   !       NOTE THAT DATA POINTS ARE 25 ns APART
420   !       FOR EVERY ACQUISITION.
430   !
440   !
441   OUTPUT 707;"HEADER OFF"                   ! Turn off headers in query reply,
450   OUTPUT 707;"AUTOSCALE"                    ! Does as it says.
460   OUTPUT 707;"SENSITIVITY?"                 ! Ask for time/div.
470   ENTER 707;Sens                            ! Read time/div.
480   Sens=Sens/8
490   OUTPUT 707;"SENSITIVITY";Sens             ! Set faster sweep speed.
500   OUTPUT 707;"STOP"                         ! Halt acquisition (system command)
510   OUTPUT 707;"ERASE PLANE0"                 ! Clears the active display plane
520   !                                           This is a system command.
530   OUTPUT 707;"MODE SINGLE"                  ! Sets for single shot operation.
540   FOR J=1 TO 20
550         OUTPUT 707;"RUN"                    ! One acquisition.
560   !
570         WAIT 1
580         NEXT J
590   !
600   !
610   OUTPUT 707;"MODE TRIGGERED"               ! Puts unit in the triggered mode.
620         OUTPUT 707;"RUN"                    ! Starts acquisition.
630   END
```

```
10    !  This sample program demonstrates some of the commands in the
20    !  Hardcopy subsystem.  The service request is used to detect
30    !  when printing is complete.  The program assumes that a
40    !  graphics printer is used and its address is set to 1.
50    CLEAR 707
60    OUTPUT 707;"REQUEST 4112"              ! Request mask where:
70    !                                        4096 = Hardcopy complete, bit 12
80    !                                          16 = Ready bit - bit 4,
90    !                                               Set so bit 12 causes SRQ
100   OUTPUT 707;"REQUEST ON"                ! Enables scope's Service Request
110   ON INTR 7 GOTO 270                     ! Exit printing routine after SRQ
120   ENABLE INTR 7;2                        ! Enables SRQ on bus #7
130   DISABLE INTR 7                         ! Disables all interrupts on bus #7
140   OUTPUT 707;"HARDCOPY"                  ! Puts the scope in the HARDCOPY
150   !                                        subsystem.
160   OUTPUT 707;"PAGE AUTO"                 ! Sets the scope to automatically
170   !                                        output a formfeed
180   OUTPUT 707;"SOURCE PLANE0,FACTORS"     ! Selects the active display
190   !                                        (plane0) and the scale factors.
200   OUTPUT 707;"PRINT"                     ! Starts print buffering
210   SEND 7;UNT UNL                         ! Clears bus
220   SEND 7;LISTEN 1                        ! Tells printer to listen
230   SEND 7;TALK 7                          ! Puts the scope in talk mode
240   SEND 7;DATA                            ! Lowers the ATN line @ controller
250   ENABLE INTR 7                          ! Enables SRQ interrupt
260   GOTO 260                               ! Loops until printing is complete
270   A=SPOLL(707)                           ! Clear service request
280   END
```

```
10    !  This sample program demonstrates some of the commands in the
20    !  Hardcopy subsystem and the PLOT command.  It assumes that
30    !  the scope is at address 7, the printer is at address 1, and
40    !  that the system bus is 700.
50    CLEAR 707
60    OUTPUT 707;"HARDCOPY"                  ! Puts the scope in the HARDCOPY
70    !                                        subsystem.
80    OUTPUT 707;"PEN AUTO"                  ! Sets the 54100A/D to the auto
90    !                                        pen mode.
100   OUTPUT 707;"SOURCE PLANE0,FACTORS"     ! Selects the active display
110   !                                        (plane0) and the scale factors
120   !                                        for output.
130   !
140   OUTPUT 707;"PLOT"                      ! Outputs data to the plotter
150   SEND 7;UNT UNL                         ! Clears bus
160   SEND 7;LISTEN 1                        ! Tells printer to listen
170   SEND 7;TALK 7                          ! Sets scope to talk mode
180   SEND 7;DATA
```

```
190   WAIT 180                                ! Wait 3 minutes for transfer to
200   !                                         complete.
210   !                                         Note:  If programming, use the
220   !                                         SRQ capabilities of the 54100A/D
230   !                                         to determine when the transfer
240   !                                         is complete.
245   !                                         Attempting to program the
250   !                                         54100A/D while making a hardcopy
260   END                                     ! dump will cause errors.
```

```
10    ! This sample program demonstrates using the memories for
20    ! measurements and performing measurements on single shot
40    ! data.  Before running the program, connect the cal signal
50    ! on the rear panel of the instrument to the Channel 1 input.
70    !
80    ! Setup signal.
90    !
100   CLEAR 707                           ! Initialize HP-IB registers.
110   OUTPUT 707;"AUTOSCALE"              ! Scale the signal.
120   OUTPUT 707;"TIMEBASE RANGE 250E-9" ! Set to 25 nsecs/division.
130   !
140   ! Acquire single sweep of signal - 10 points.
150   !
160   OUTPUT 707;"ACQUIRE POINTS 500 "   ! Full screen is 500 points.
170   OUTPUT 707;"TYPE 1"                 ! Acquisition type is normal.
180   OUTPUT 707;"COMPLETE 0"             ! Acquire 1 sweep.
190   OUTPUT 707;"DIGITIZE CHANNEL 1"     ! Acquire Ch1 data to Memory 1.
200   OUTPUT 707;"BLANK CHANNEL 1"        ! Turn off Ch1.
210   !
220   ! Measure peak to peak voltage of memory 1.
230   !
240   OUTPUT 707;"MEAS SOURCE MEMORY1"   ! Set measurement source.
250   OUTPUT 707;"VPP?"                   ! Measure peak to peak voltage.
260   DIM Vpp$[20]
270   ENTER 707;Vpp$
280   PRINT Vpp$                          ! Print results.
290   END
```

```
10    !
20    !  This program demonstrates some of the learn string capabilities.
30    CLEAR 707
40    !
50    !
60    DIM Setting$[276]                !  266 = # of bytes in learn string
70    !                                   plus 10 = "SETUP #A**"
80    !                                   Where SET = header
90    !                                         #A = indicates binary block
100   !                                         ** = 2 byte integer = length
110   OUTPUT 707;"LONGFORM ON"         ! Use longform of mnemonics.
120   OUTPUT 707;"HEADER ON"           ! Tells the 54100A/D to precede the
130   !                                  learn string with a header.
140   OUTPUT 707;"EOI ON "             ! Tells the 54100A/D to output an EOI
150   !                                    with the last byte.
160   OUTPUT 707;"SETUP?"              ! This asks the scope for the learn string
170   ENTER 707 USING "-K";Setting$    ! Reads in header and string.  (-K tells
180   !                                  computer to treat CR & LF as data).
190   LOCAL 707                        ! Puts scope in local operation.
200   !
210   !
220   !
230   ENTER 2                           ! This allows you to change the scope's
240   !                                    setup.
250   !                                    Hit 'ENTER' to continue
260   !
270   !
280   OUTPUT 707;Setting$               ! Outputs the learn string and header.
290   !                                   Scope is reset to previous setup.
300   !
310   LOCAL 707                         ! Returns scope to local operation.
320   !
330   END
```

```
10    !
20    ! This sample program demonstrates more uses of the Service
30    ! Requests (SRQ's).  This set of instructions uses: Hardcopy
40    ! Done, Local, Front Panel Service, Ready bit and  Ready Masks.
50    ! The scope  will monitor the front panel for SRQ's and echo
60    ! any activity. Any Advisories or Acquisitions initiated
70    ! by the front panel will be disclosed. These examples assume
80    ! the scope to be at address 7 and the plotter to be at address
90    ! 1 on bus #7.
100   !
110   CLEAR 707
120   PRINTER IS 1                              ! Display is PRINT destination
130   DIM K$[80],A$[80]
```

```
140   !
150   ON INTR 7 GOSUB Srq_svc                ! Goto 'Srq_svc' on Service Request
160   ENABLE INTR 7;2                        ! Enables SRQ on bus #7
170   DISABLE INTR 7                         ! Disables all interuppts on bus #7
180   !
190   PRINT
200   OUTPUT 707;"RESET"                     ! Resets 54100A/D.
210   OUTPUT 707;"AUTOSCALE"                 ! Does as it says
220   OUTPUT 707;"ACQUIRE MODE AVERAGE"      ! Puts scope into averaged mode.
230   WAIT 4                                 ! Wait for data accumulation
240   INTEGER Rqsmask
250   Rqsmask=4096+16+4                      ! request mask
260   !                                        4096 = Hardcopy done - bit 12
270   !                                          16 = Ready - bit 4
280   !                                           4 = Front Panel Service - bit 2
290   !                                             Set so bit 12 causes an SRQ
300   OUTPUT 707;"REQUEST";Rqsmask           ! Send Request Mask
310   OUTPUT 707;"REQUEST ON"                ! Sets RQS - bit 6 (request mask)
320   OUTPUT 707;"LONGFORM ON"               ! Sets longform for headers.
330   OUTPUT 707;"HEADER ON"                 ! Sets headers on for queries.
340   Stat=SPOLL(707)                        ! Serial Poll scope, should be 0.
350   PRINT "Result of Serial Poll is ";Stat
360   !
370   Dump_flag=0
380   OUTPUT 707;"HARDCOPY SOURCE PLANE0,FACTORS" ! Selects active display and
390   !                                             scale factors for output.
400   OUTPUT 707;"PEN AUTO"                  ! Sets auto pen on.
410   OUTPUT 707;"PRINT"                     ! Starts Plot.
420   SEND 7;UNT UNL                         ! Turns off entire bus.
430   SEND 7;LISTEN 1                        ! Sets plotter to listen.
440   SEND 7;TALK 7                          ! Sets scope to talk.
450   SEND 7;DATA                            ! Lower ATN line @ controller
460   !
470   ENABLE INTR 7                          ! Enables interrupts on bus #7
480   !
490   IF Dump_flag=0 THEN
500      PRINT
510      PRINT " Waiting for hardcopy transfer to complete."
520      PRINT " Time availible for other tasks."
530      PRINT " !!!!    Bus is  NOT  availible      !!!!!"
540      WAIT 2
550      GOTO 490
560   END IF
570   GOTO 570
580   !
590   !
600   !
610   ! Service request interrupt routine,
620   !
630 Srq_svc:!
640   Stat=SPOLL(707)                        ! Perform serial poll
650   !                                        and clear SRQ.
660   INTEGER J
```

```
670   PRINT
680   PRINT "Service Request Status= ";Stat
690   !
700   !
710   IF BIT(Stat,0) THEN                          ! 54100A/D is not a controller.
720   PRINT "RQC should not be set - PROBLEM"      ! RQC cannot be set by 54100A/D
730   END IF
740   !
750   !
760   IF BIT(Stat,1) THEN                          ! RAM power failure
770   PRINT "PWR status has been set-WHY?"
780   END IF
790   !
800   !
810   IF BIT(Stat,2) THEN                          ! Front Panel Service
820     PRINT "FPS status has been set"
830     OUTPUT 707;"KEY?"                          ! Asks for key code.
840     ENTER 707;K$                               ! Reads key code.
850     OUTPUT 707;K$                              ! Outputs key code.
860     PRINT "      "&K$
870   END IF
880   !
890   !
900   IF BIT(Stat,3) THEN                          ! Local operation occurred.
910      PRINT "LCL operation has occured"
920   END IF
930   !
940   !
950   IF BIT(Stat,4) THEN                          ! Ready bit - only bit active
960      PRINT "Hardcopy Complete  !!"             ! in the Ready byte is Hardcopy
970      IF Dump_flag=0 THEN                       ! Complete.
980          SEND 7;UNT UNL
990          Dump_flag=1
1000     END IF
1010     PRINT "Now try pressing keys, they will echo from controller"
1020  END IF
1030  !
1040  !
1050  IF BIT(Stat,5) THEN                         ! Go read the errors.
1060    REPEAT
1070      OUTPUT 707;"ERR?"                       ! Asks for next error in queue.
1080      ENTER 707;A$                            ! Reads next error.
1090      PRINT "Error was : ";A$                 ! Prints next error.
1100    UNTIL VAL(A$[9,12])=0                     ! Until error queue is empty.
1110  END IF
1120  !
1130  !
1140  IF BIT(Stat,6) THEN                         ! A SRQ was generated by someone.
1150    OUTPUT 707;"REQUEST?"                     ! Asks for mask value.
1160    ENTER 707;A$                              ! Reads mask value.
1170    PRINT A$&" is the mask value"
1180  END IF
1190  !
```

```
1200  !
1210  IF BIT(Stat,7) THEN                    ! Advisory has been initiated
1220    OUTPUT 707;"DSP?"                    ! Asks for advisory.
1230    ENTER 707;A$                         ! Reads advisory.
1240    PRINT A$&" is the Advisory"          ! Prints advisory.
1250  END IF
1260  !
1270  !
1280  ENABLE INTR 7
1290  RETURN
1300  !
1310  END
```

```
10    ! This sample program demonstrates some of the commands in the
20    ! MEASURE Subsystem.
30    !
40    CLEAR 707
50    OUTPUT 707;"AUTOSCALE"               ! Does as it says
60    OUTPUT 707;"ACQUIRE TYPE NORMAL"     ! Sets display mode to Normal
70    OUTPUT 707;"DISPLAY"                 ! Selects DISPLAY Subsystem
80    OUTPUT 707;"TMARKER ON"              ! Turn on Tmarkers
90    !
100   !
110   OUTPUT 707;"MEASURE"                 ! Selects MEASURE subsystem
120   OUTPUT 707;"PRECISION HIGH"          ! Sets precision for maximum accuracy
130   OUTPUT 707;"SOURCE CHANNEL 1"        ! Channel 1 is the measurement source
140   OUTPUT 707;"VSTART -.2"              ! Sets voltage markers to
150   OUTPUT 707;"VSTOP -.2"               !  -200 mV.  This will be used as a
160   !                                       reference for the edge find
170   !                                       function.
180   INTEGER J
190   FOR J=1 TO 2
200         OUTPUT 707;"ESTART +";J        ! Find Jth positive edge
210         WAIT .75
220         OUTPUT 707;"ESTOP -";J         ! Find Jth negative edge
230         WAIT .75
240         NEXT J
250   !
260   ENTER 2                              ! This statement causes a pause in the
270   !                                       program, press ENTER on the control-
280   !                                       ler to continue.
290   !
300   !
310   REAL Delay,Offset
320   Delay=2.4E-6                         ! Initialize delay variable
330   Offset=.4                            ! Initialize offset variable
```

```
340  FOR J=1 TO 20
350       OUTPUT 707;"TSTART";Delay         ! Move time start marker
360       OUTPUT 707;"TSTOP";-Delay         ! Move time stop marker
370       OUTPUT 707;"VSTART";-.19-Offset   ! Move voltage start marker
380       OUTPUT 707;"VSTOP";-.19+Offset    ! Move voltage stop marker
390       Offset=Offset-.04
400       Delay=Delay-2.40E-7
410       NEXT J
420  !
430  ENTER 2                                ! Same as line 230.
440  !
450  !
460  OUTPUT 707;"PRECISION LOW"             ! Selects the PRECISION flag low.
470  !                                        Low precision uses previously
480  !                                        acquired data for measurements.
490  !                                        This allows faster completion of
500  !                                        measurements at the expense of some
510  !                                        accuracy and repeatability.
520  OUTPUT 707;"ALL?"                      ! Measure all parameters. They will be
530  !                                        displayed on scope, and are
540  !                                        availible over HP-IB.
550  !
560  !
570  !
580  ENTER 2                                ! Pause
590  !
600  !
610  OUTPUT 707;"RISE?"                     ! Measure RISE time using low
620  !                                        precision.  You set precision
630  !                                        flag earlier.
640  ENTER 2                                ! Pause
650  !
660  !
670  OUTPUT 707;"PRECISION HIGH"            ! Set PRECISION flag high.
680  OUTPUT 707;"RISE?"                     ! Measure precise RISE time.
690  !                                        Watch the display during this
700  !                                        measurement.
710  !
720  !
730  !
740  OUTPUT 707;"PRECISION LOW"             ! Sets PRECISION flag low.
750  LOCAL 707                              ! Puts the 54100A/D in local operation.
760  END
```

```
10    !
20    ! This sample program demonstrates some of the uses of
30    ! Service Requests (SRQ's).  This set of instructions uses
40    ! the Acquisition Done, Local, Front Panel Service,
50    ! Ready and Ready masks.  An acquisition that will produce
60    ! buffered results will be started.  When a SRQ is sent the
70    ! results will be read and displayed.  The scope will then
80    ! monitor the front panel keys using SRQ's and echo any
90    ! activated.   Any Advisories or Acquisitions initiated from
100   ! the front panel will be disclosed.
110   !
120   CLEAR 707
130   !
140   PRINTER IS 1                          ! Display is PRINT destination
150   DIM B$(1:16)[30],K$[80],A$[80]
160   !
170   ON INTR 7 GOSUB Srq_svc               ! Goto 'Srq_svc' on Service Request.
180   ENABLE INTR 7;2                       ! Enables SRQ on bus #7.
190   DISABLE INTR 7                        ! Disables all interrupts on bus #7.
200   !
210   PRINT
220   OUTPUT 707;"RESET"                    ! Resets 54100A/D.
230   OUTPUT 707;"AUTOSCALE"                ! Does as it says.
240   INTEGER Rqsmask
250   Rqsmask=1024+16+4                     ! Request mask where:
260   !                                       1024 = Acquisition done - bit 10.
270   !                                         16 = Ready - bit 4.
280   !                                          4 = Front Panel Service - bit 2.
290   !
300   OUTPUT 707;"REQUEST";Rqsmask          ! Sends Request Mask.
310   OUTPUT 707;"REQUEST ON"               ! Sets RQS - bit 6 in Request mask.
320   OUTPUT 707;"LONGFORM ON"              ! Turns on longform for headers.
330   OUTPUT 707;"HEADER ON"                ! Turns headers on for queries.
340   Stat=SPOLL(707)                       ! Serial Poll scope, should be 0
350   PRINT "Result of Serial Poll is ";Stat
360   !
370   Meas_flag=0
380   OUTPUT 707;"MEASURE "                 ! Enters MEASURE subsystem.
390   OUTPUT 707;"SOURCE CHANNEL 1"         ! Channel 1 is source for measurement
400   OUTPUT 707;"PRECISION HIGH; ALL?"     ! Sets PRECISION flag high and
410   !                                       measures all.
420   ENABLE INTR 7                         ! Enables interrupts on bus #7.
430   !
440   PRINT " Waiting for measurement to complete."
450   PRINT " Time availible for other tasks."
460   PRINT " Bus is available."
470   !
480   GOTO 480                              ! Loop until Service Request occurs.
490   !
500   !
510   !
520   ! Service Request Interrupt Routine
```

```
530   !
540 Srq_svc:                                  !
550   Stat=SPOLL(707)                         ! Performs a Serial Poll
560   !                                         and clears SRQ.
570   INTEGER J
580   PRINT
590   PRINT "Service Request Status= ";Stat
600   !
610   !
620   IF BIT(Stat,0) THEN                     ! Request Control - 54100A/D is
630      PRINT "RQC should not be set - PROBLEM" !  not a controller and
640   END IF                                  !  cannot send a RQC.
650   !
660   !
670   IF BIT(Stat,1) THEN                     ! RAM power failure.
680      PRINT "PWR status has been set.  WHY?"
690   END IF
700   !
710   !
720   IF BIT(Stat,2) THEN                     ! Front Panel Service.
730     PRINT "FPS status has been set"
740     OUTPUT 707;"Key?"                     ! Asks for key code.
750     ENTER 707;K$                          ! Reads key code.
760     OUTPUT 707;K$                         ! Outputs key code.
770     PRINT "     "&K$
780   END IF
790   !
800   !
810   IF BIT(Stat,3) THEN                     ! Local operation occurred
820      PRINT "LCL operation has occured"
830   END IF
840   !
850   !
860   IF BIT(Stat,4) THEN                     ! Ready bit - only bit active
870      PRINT "Acquisition done !"           ! in the Ready byte is Acq done
880      IF Meas_flag=0 THEN                  ! First time thru?
890           FOR J=1 TO 16                   !
900           ENTER 707;B$(J)                 ! Reads measurement results
910           PRINT B$(J)                     ! and prints them.
920           NEXT J
930           Meas_flag=1
940           PRINT
950           PRINT "Press some keys.  The key number will be printed out."
960      END IF
970   END IF
980   !
990   !
1000  IF BIT(Stat,5) THEN                     ! GO read the errors.
1010    REPEAT
1020      OUTPUT 707;"ERR?"                   ! Asks for next error in queue.
1030      ENTER 707;A$                        ! Reads error.
1040      PRINT "Error was : ";A$             ! Prints error.
1050    UNTIL VAL(A$[9,12])=0                 ! Until error queue is empty.
1060  END IF
```

```
1070  !
1080  !
1090  IF BIT(Stat,6) THEN                    ! A SRQ has been generated.
1100    OUTPUT 707;"REQUEST?"                ! Ask for mask value.
1110    ENTER 707;A$                         ! Reads mask value.
1120    PRINT A$&" is the mask value"        ! Prints mask value.
1130  END IF
1140  !
1150  !
1160  IF BIT(Stat,7) THEN                    ! Advisory has been initiated.
1170    OUTPUT 707;"OSP?"                    ! Asks for advisory.
1180    ENTER 707;A$                         ! Reads advisory.
1190    PRINT A$&" is the Advisory"          ! Prints the advisory.
1200  END IF
1210  !
1220  !
1230  ENABLE INTR 7                          ! SRQ disables interrupts.
1240  RETURN                                 ! This reenables them.
1250  !
1260  END
```

Appendix B
SOFTWARE DELAY CALIBRATION AND TRIGGER DELAY OPTIMIZATION

The trigger delay and channel to channel skew calibrations in the Cal menu on the 54100A/D are provided to null delay differences in the trigger and data acquisition paths of the trigger and the data. This would include acquisition time differences both internal and external to the instrument.

Channel to channel skew adjustment is used to change the placement of the channel 2 data relative to channel 1 so delay differences in the data acquisition paths do not introduce offsets in channel-to-channel time interval measurements.

Differences in internal delays as well as differences in external delays caused by dissimilar probes, cable lengths etc. can be nulled. This is done by injecting the cal signal at the probe tips or other desired points when performing the calibrations.

The trigger delay calibrations are used to position waveforms horizontally so that the zero reference corresponds with the trigger event. When both the internal and external delays have been compensated for, the instrument provides a timebase delay that is calibrated in an absolute sense to the trigger point. The timebase delay tells you where the display window is located relative to the trigger.

Trigger delay calibrations do not affect channel-to-channel measurements as the vertical inputs are always displayed relative to each other depending on the setting of the ch-to-ch Skew cal factor. As long as the Ch-to-Ch Skew cal factor is set correctly you can make accurate channel-to-channel measurements even if the trigger delay cal factors are not set correctly.

The delay calibration feature, a consequence of the 54100A/D's negative time and digital architecture, is convenient for referencing measurements to the probe tips, or other points, even if different types of probes or a probe multiplexer is used. To calibrate a given probe configuration inject a fast risetime signal at the probe tips, or wherever you want your measurement to be referenced and follow the instructions in the Cal menu. Refer to Section 6 for a discussion of the cal menu.

Once the cal procedure is completed the trigger edge will be displayed at the time-zero reference, and if you use an external trigger the time-zero reference will correspond to the time of the trigger event at the Trig 3 or 4 probe tip, or other point of interest. The cal factors are stored in the nonvolatile SAVE/RECALL registers. This allows the instrument to retain calibrations for up to 10 different probe or measurement configurations. By using the time interval measurements built into the 54100A/D, the display skew and trigger delay cal factors can be determined and programmed via the HP-IB.

The trigger delay calibrations compensate for delays to a first order approximation. Actual trigger delays, in addition to probe length, are function of signal characteristics such as risetime, amplitude, and rep. rate. If these signal characteristics are not the same when making measurements as they were during calibration, the trigger edges will be displaced from time-zero. The error, however, will be small and will rarely result in any confusion as to which edge is the trigger. For fast risetime signals (<3 ns) this displacement will be less than ±400 ps. Because of these second order affects it should not be assumed that the trigger edge is at precisely time-zero when making time interval measurements unless the edges are fast and 400 ps error can be tolerated. These effects apply only to trigger delays as channel-to-channel skew has no dependency on signal characteristics.

For signals with slower edge speeds, trigger hysteresis can cause a displacement from time-zero, however, compared to the sweep speed at which the signal would be viewed the displacement usually would be small. Trigger hysteresis on the 54100A/D is 1 minor division on channels 1 & 2 and 10 mV (with 50 ohm pods) on Trig 3 & 4. The trigger level is at the center of the hysteresis band and the trigger comparator actually begins to switch when the input voltage is 1/2 a minor division from the programmed threshold. This causes the actual threshold crossing to be displaced from time-zero by the amount of time it takes the signal to traverse 1/2 a minor division vertically. The direction of the displacement depends on the trigger slope. At sweep speeds where the signal appears as anything but a near-horizontal line, this displacement is not significant for viewing but can affect time interval measurements if ignored.

With trigger delay calibration captured signals can be correctly plotted relative to the time-zero reference with a small error caused by the second order effects. This applies for the Edge, State, Time-Delayed, and Event-Delayed modes. In the Pattern mode, however, the instrument does not know which input will provide the trigger and does not know which cal factor to use. In this case the average of Channel 1 and Channel 2 trigger delay cal factors is used as a compromise. This will result in a minimal displacement when the trigger edge comes from one of the vertical channels but a large displacement can result if the trigger comes from Trig 3 or 4.

Large delay differences in the signal paths for channel 1 and channel 2 will result in a large displacement, so it is desirable to match these paths as close to one another as possible if an accurate time-zero reference is needed in the pattern mode. Also of concern, when you are in the pattern mode is the relative skew between the inputs. This skew results from delay differences in the acquisition paths internal and external to the instrument. For example, when using the time qualified pattern, skew can cause the pattern true-time seen by the filter timer to be different than the actual time at the probe tips. Just how much skew, or differential delay, exists between paths is reflected in the trigger delay cal factors(assuming the instrument is calibrated) because the cal factor for each input is referenced to the same channel (channel 1). The difference in the cal factors is equal to the amount of skew in the trigger paths. A more negative cal factor means that the trigger path delay for that channel is longer.

The differential delay between channel 1 & 2 and between Trig 3 & 4 is usually less than 400 ps. This assumes the use of 54002A 50 ohm pods and is referenced to the BNC connectors.

The delay through Channel 1 & 2 is nominally 1.6 ns longer than the delay through Trig 3 & 4. This can cause erroneous pattern triggering unless the extra delay is compensated for by inserting extra delay by using longer cables on Trig 3 and Trig 4. Inserting delay(s) to reduce skew for pattern triggering will also reduce time-zero offset in the Pattern mode.

While skew is not of concern with respect to the time-zero reference in other than Pattern mode, it can affect the operation of the other modes. For example, the setup and hold times in the State mode can be different at the probe tips than they are at the instrument's inputs because of dissimilar probes or cable lengths. This is caused by the fact that the trigger circuitry operates on signals in real-time, thus ruling out software calibration.

Appendix C

Detailed Operation of the Automatic Measurements

INTRODUCTION

The automatic parameter and time interval measurements resident in the 54100A/D are designed to allow you to optimize measurement speed and accuracy for your application. Depending upon a number of factors such as display mode, number of averages, type of measurement, and state of the precision key or the PRECISION flag (HP-IB), the instrument uses different criteria for establishing how much data needs to be acquired for measurement.

DISPLAY MODE CONSIDERATIONS

The measurements are based on the data that is on the screen. The instrument maintains internal copies of the screen data, on which the measurements are made. If the unit is in the Normal mode it uses only the most recent y axis information in each time bucket to make the measurement.

The waveform area of the screen is 256 pixels high by 501 pixels wide. For most measurements 501 time buckets are used, however, if the sweep is faster than 500 ps/div the number of time buckets is equal to (10 * (time_per_div_/10 ps))+1. The reason for this is that the maximum time resolution for the 54100A/D is 10 ps.

Most of the measurements use 90% criterion. In the normal mode at least 90% of the time buckets have one data point. In the averaged mode the 90% criterion means that at least 90% of the time buckets have received N data points where N is the number of averages.

One method of trading measurement speed for accuracy is to increase the number of averages in average mode. The larger the number of averages, the more time that will be required to make a measurement, but also, the better the accuracy and repeatability.

FINE PRECISION AND COARSE PRECISION MEASUREMENTS

The 54100A/D performs two types of automatic measurements; coarse and fine (coarse precision and fine precision). Coarse measurements are made based on the data on screen. If there is insufficient data on the screen, then new data is acquired in order to make the measurement. Fine measurements begin with a course measurement to locate the edge(s). Each edge is then expanded to achieve maximum resolution.

The coarse measurements are: the voltage measurements, the front panel start on edge and stop on edge time interval measurements, Auto Top-Base, any measurement when the instrument is "Stopped", and all HP-IB measurements when the PRECISION flag is low.

The precise measurements are; the front panel Precise Edge Find measurement, the front panel parameter measurements when the instrument is "Running, and HP-IB time related measurements when the PRECISION flag is high.

The coarse measurements are as accurate as the precise measurements when the waveforms are fully expanded (edges at a 45 degree angle) and the display data is 90% complete.

COARSE MEASUREMENTS

As mentioned above, coarse measurements use the data on screen unless there is insufficient data. For front panel coarse measurements only 5 data points need to be present on the screen. For HP-IB coarse measurements the screen data must meet the 90% completion criterion. If the data is insufficient, the instrument acquires new data until the 90% completion criterion is met before the measurement is made.

Coarse measurements are faster but their accuracy, unlike precise measurements depends on the sweep speed. The front panel measurements can be made on very limited amount of data. This is important for single shot signals and low rep rate signals when only a limited amount of data has been acquired.

It is a good programming practice to clear the screen when the input signal is changed before making a coarse measurement. This is especially true when a high number of averages or a long persistence has been selected. (Note: in many cases the screen will be cleared automatically when an instrument setting is changed. The exceptions are changes in settings that don't effect the waveform e.g., moving the time or voltage markers.)

PRECISE MEASUREMENTS

Precise measurements in general, but not always, automatically rescale the timebase to expand signal edges for maximum resolution. This technique provides maximum accuracy and results that are independent of the initial sweep speed setting.

When a precise time interval measurement is made the instrument will first perform a coarse calculation to locate the edge(s) of interest. Next, for each edge, the display window will be positioned so that the edge of interest is at center screen and the sweep speed is increased causing the signal to be expanded on the horizontal axis. The instrument will continue to do this until one of three conditions is met: (1) The slope of the signal is 45 degrees or less. (2) The sweep speed equals 500 ps/div (maximum resolution). (3) Signal jitter at the current sweep speed makes it pointless to increase the sweep speed further. At each faster sweep speed a calculation is made to determine if any of these conditions are met. If so, expansion is stopped and the edge crossing time is determined.

If the sweep speed is 500 ps/div or faster or if the measurement requires only a single edge that already has less than 45 degrees slope as displayed, a precise measurement will revert to a coarse measurement.

LOCATING THE EDGE

The edges are measured at the point where the waveform edge crosses the voltage level. For the time interval measurements the level is defined by the voltage markers. For the parameter measurements, the level is defined by the measurement at 10%, 50%, or 90% level on the waveform relative to the top and base.

The measurement routine uses a dual threshold technique. The upper threshold is defined to be 2 A/D values above the level and the lower threshold is defined to be 2 A/D values below the level.

In terms of the screen, the threshold region is one minor division wide in the full screen mode and half a minor division in the split screen mode. The advantage of this technique is that edges can be determined accurately in the presence of a limited amount of noise.

The edge is determined by performing a linear regression (curve fit) on the points that fall inside the threshold region plus the first point below the threshold region and the first point above the threshold region. The edge crossing is defined as the point where the line generated by the linear regression crosses the level. In many cases, there are no points inside the threshold region and the linear regression becomes a linear interpolation between the two points above and below the threshold region. (e.g., nearly perpendicular edges).

A limitation of this technique is that there are cases which may result in the edge not being found, notably very small signals and measurements made near the top of waveforms.

INDEX

A PAGE

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y