Agilent 54753A and 54754A TDR Plug-in Modules Programmer's Guide

Agilent Technologies

Notices

Manual Part Number

54754-90014

Edition

July 2004
Printed in Malaysia

Agilent Technologies, Inc.
Digital Signal Analysis Division
1400 Fountaingrove Parkway
Santa Rosa, CA 95403, USA

Warranty

The material contained in this document is provided "as is," and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

TTechnology Licenses

MS-DOS ® is a U.S. registered trademark of Microsoft Corporation.

Restricted Rights Legend

If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as "Commercial computer software" as defined in DFAR 252.227-7014 (June 1995), or as a "commercial item" as defined in FAR 2.101(a) or as "Restricted computer software" as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to Agilent Technologies' standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Safety Notices

CAUTION

Caution denotes a hazard. It calls attention to a procedure which, if not correctly performed or adhered to, could result in damage to or destruction of the product. Do not proceed beyond a caution sign until the indicated conditions are fully understood and met.

WARNING

Warning denotes a hazard. It calls attention to a procedure which, if not correctly performed or adhered to, could result in injury or loss of life. Do not proceed beyond a warning sign until the indicated conditions are fully understood and met.

Introduction

54750A/83480A Mainframes

For controlling 54753A and 54754A TDR plug-in modules that are installed in a 54750A or 83480A mainframes, refer to the following books:

- This manual, the *54753A and 54754A TDR Plug-in Modules Programmer's Guide.*
- *54753A and 54754A Programmer's Quick Reference*.
- *54750A Oscilloscope and 83480A Analyzer Programmer's Guide* for information about programming the oscilloscope.
- *54753A and 54754A User's Guide* for general information about the operation of the TDR plug-in module.

86100-Series Mainframes

For 54753A and 54754A TDR plug-in modules that are installed in an 86100-series mainframe, refer to the following books:

- 86100-series programmer's guide.
- 86100-series online help.

Notational Conventions

`computer type.` All characters appearing in `computer type` are key words and must be entered exactly as shown.

CAPITAL LETTERS. Capital letters indicate the short form of a command. The actual command is not case sensitive and can be entered in upper or lower case.

`<>` Angular brackets enclose words or characters that symbolize a program code parameter or an GPIB command.

`::=` "is defined as." For example, <A> ::= <B> indicates that <A> can be replaced by <B> in any statement containing <A>.

`|` "or." Indicates a choice of one element from a list. For example, <A> | <B> indicates <A> or <B> but not both.

`. . .` An ellipsis (trailing dots) indicate that the preceding element may be repeated one or more times.

`[ ]` Square brackets indicate that the enclosed items are optional.

`{ }` When several items are enclosed by braces, one, and only one of these elements may be selected.

Definitions

`d` ::= A single ASCII numeric character, 0-9.

`n` ::= A single ASCII non-zero, numeric character, 1-9.

`<NL>` ::= Newline or Linefeed (ASCII decimal 10).

`<sp>` ::= space character.

`<white_space>` ::= space(s) and tabs.

Contents

1 **Introduction**

2 **Root Level Commands**

BLANk 2–7
COMMents 2–8
DIGitize 2–8
MENU 2–10
STORe:WAVEform 2–11
VIEW 2–12

3 **Acquire Commands**

BEST 3–3

4 **Calibrate Commands**

SAMPlers 4–4

5 **Channel Commands**

OFFSet 5–4
RANGe 5–5
SCALe 5–6
TDRSkew 5–8
UNITs 5–9

6 **Disk Commands**

LOAD 6–4
STORe 6–5

7 **Display Commands**

ASSign 7–6

8 Function Commands

ADD 8–6
BWLimit 8–6
DIFFerentiate 8–7
DIVide 8–8
FFTMagnitude 8–8
INTegrate 8–9
INVert 8–10
MAGNify 8–10
MAXimum 8–11
MINimum 8–12
MULTiply 8–12
SUBTract 8–13
VERSus 8–14

9 Histogram Commands

Histogram Commands 9–2
WINDow:SOURce 9–5

10 Limit Test Commands

SWAVeform 10–8

11 Marker Commands

MODE 11–6
PROPagation 11–6
REFerence 11–8
X1Y1source 11–9
X2Y2source 11–10
XUNITs 11–11
YUNITs 11–12

12 Mask Test Commands

AMASk:SOURce 12–10
SWAVeform 12–11
SCALe:SOURce 12–12

13 Measure Commands

DELTatime 13–11
DUTycycle 13–12
FALLtime 13–13
FREQuency 13–15
NWIDth 13–16
OVERshoot 13–18
PERiod 13–19
PREShoot 13–20
PWIDth 13–22
RISetime 13–23
SOURce 13–24
TEDge 13–26
TMAX 13–27
TMIN 13–29
TVOLt 13–30
VAMPlitude 13–31
VAVerage 13–32
VBASe 13–34
VLOWer 13–35
VMAX 13–36
VMIDdle 13–37
VMIN 13–38
VPP 13–40
VRMS 13–41
VTIMe 13–43
VTOP 13–44
VUPper 13–45

14 TDR Commands

RESPonse:CALibrate 14–5
RESPonse:CALibrate:CANCel 14–5
RESPonse:CALibrate:CONTinue 14–5
PRESet 14–6
RATE 14–7
RESPonse 14–8
RESPonse:HORizontal 14–9
RESPonse:HORizontal:POSition 14–10
RESPonse:HORizontal:RANGe 14–11
RESPonse:RISetime 14–12
RESPonse:TDRDest 14–14
RESPonse:TDRTDT 14–15
RESPonse:TDTDest 14–16
RESPonse:VERTical 14–18
RESPonse:VERTical:OFFSet 14–19
RESPonse:VERTical:RANGe 14–20
STIMulus 14–21

15 Waveform Commands

SOURce 15–5

16 Waveform Memory Commands

SAVe 16–4

1

Introduction

Introduction

The 54753A and 54754A Programmer's Guide contains descriptions of the remote programming commands that apply to the TDR plug-in modules.

2

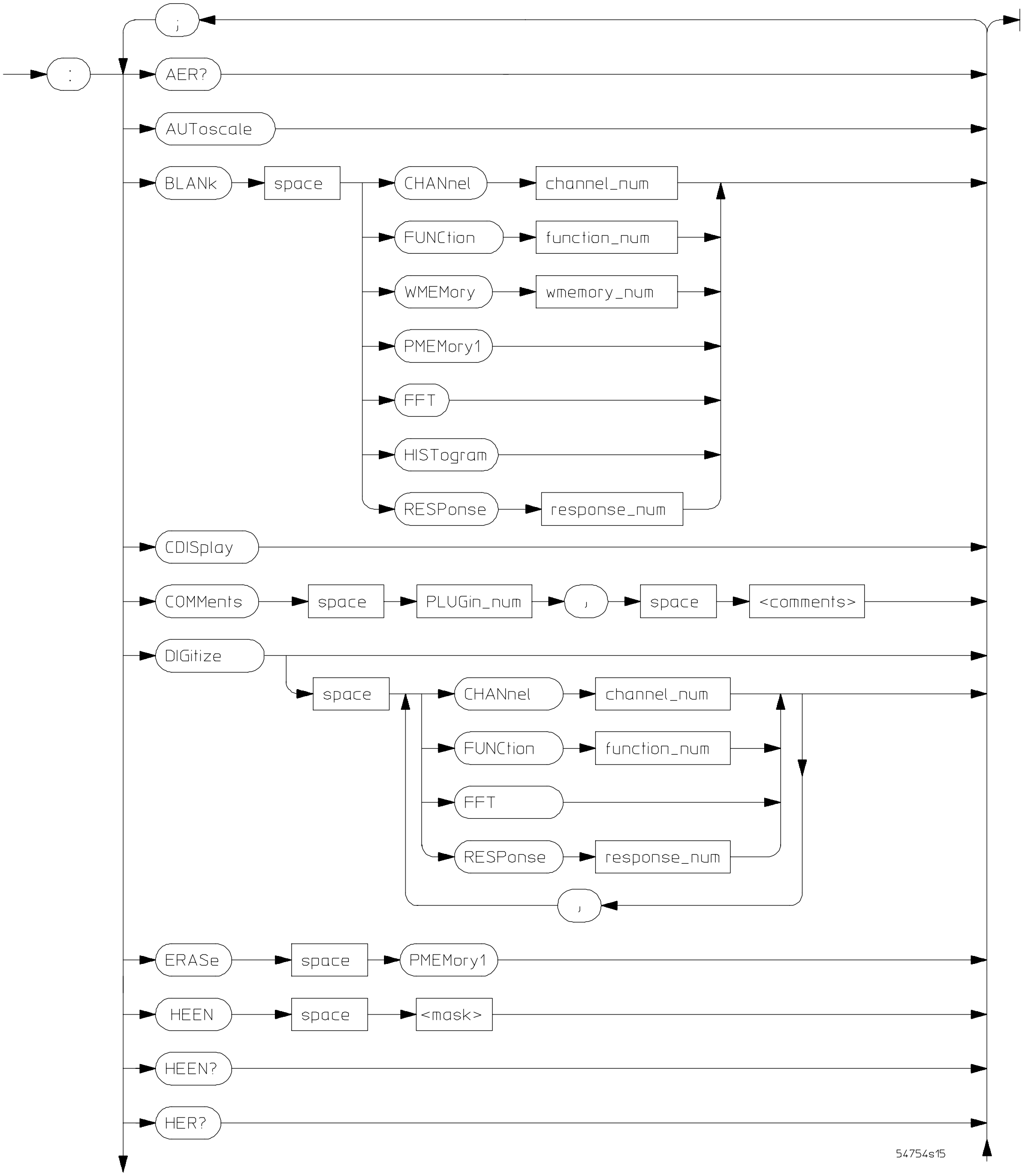
Root Level Commands

Root Level Commands

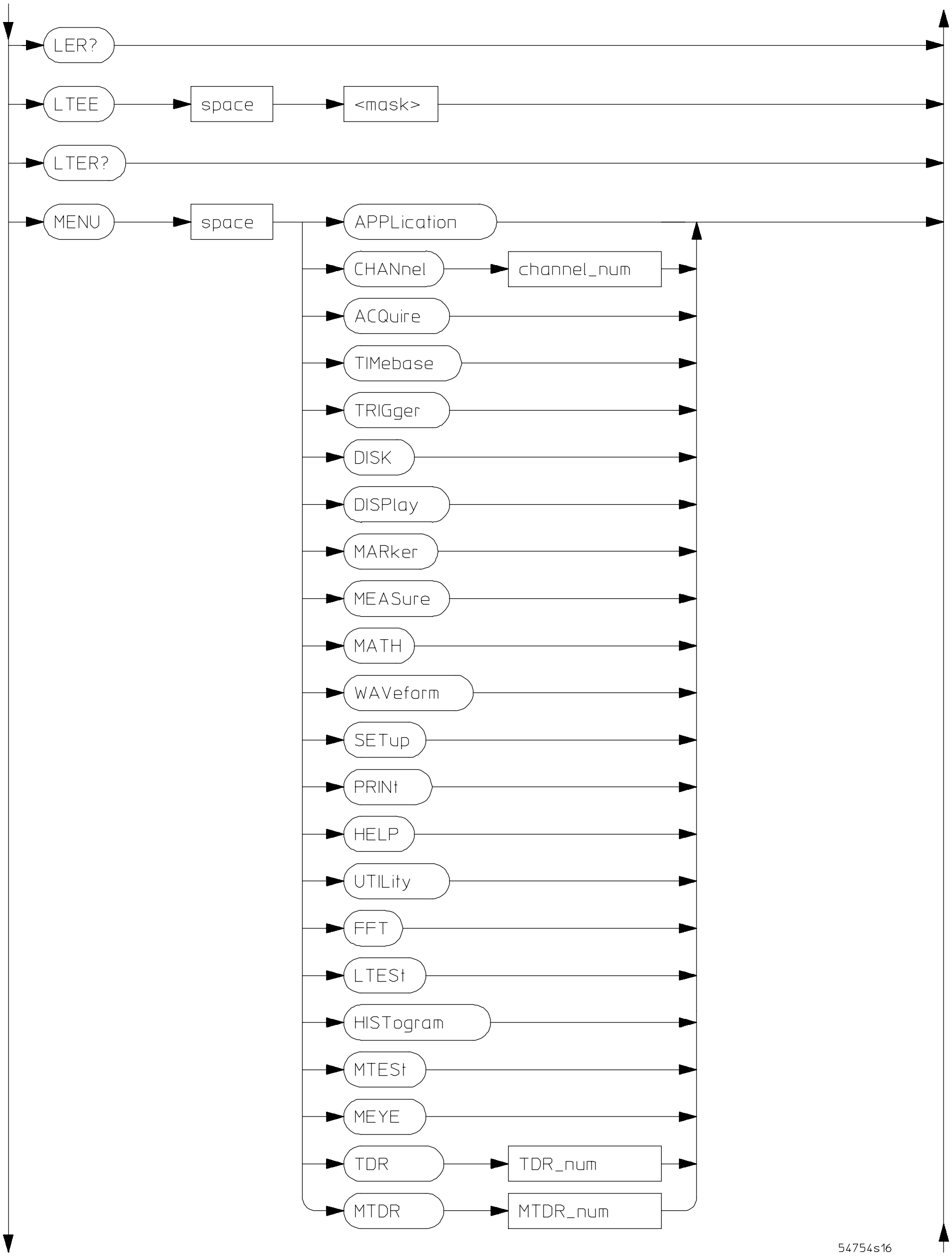
Root level commands control many of the basic operations of the oscilloscope; most of these can also be selected by pressing the labeled keys on the front panel. These commands are always recognized by the command parser if they are prefixed with a colon, regardless of the current tree position. After executing a root level command, the parser is positioned at the root of the command tree.

The TDR plug-in modifies the operation of the following root level commands:

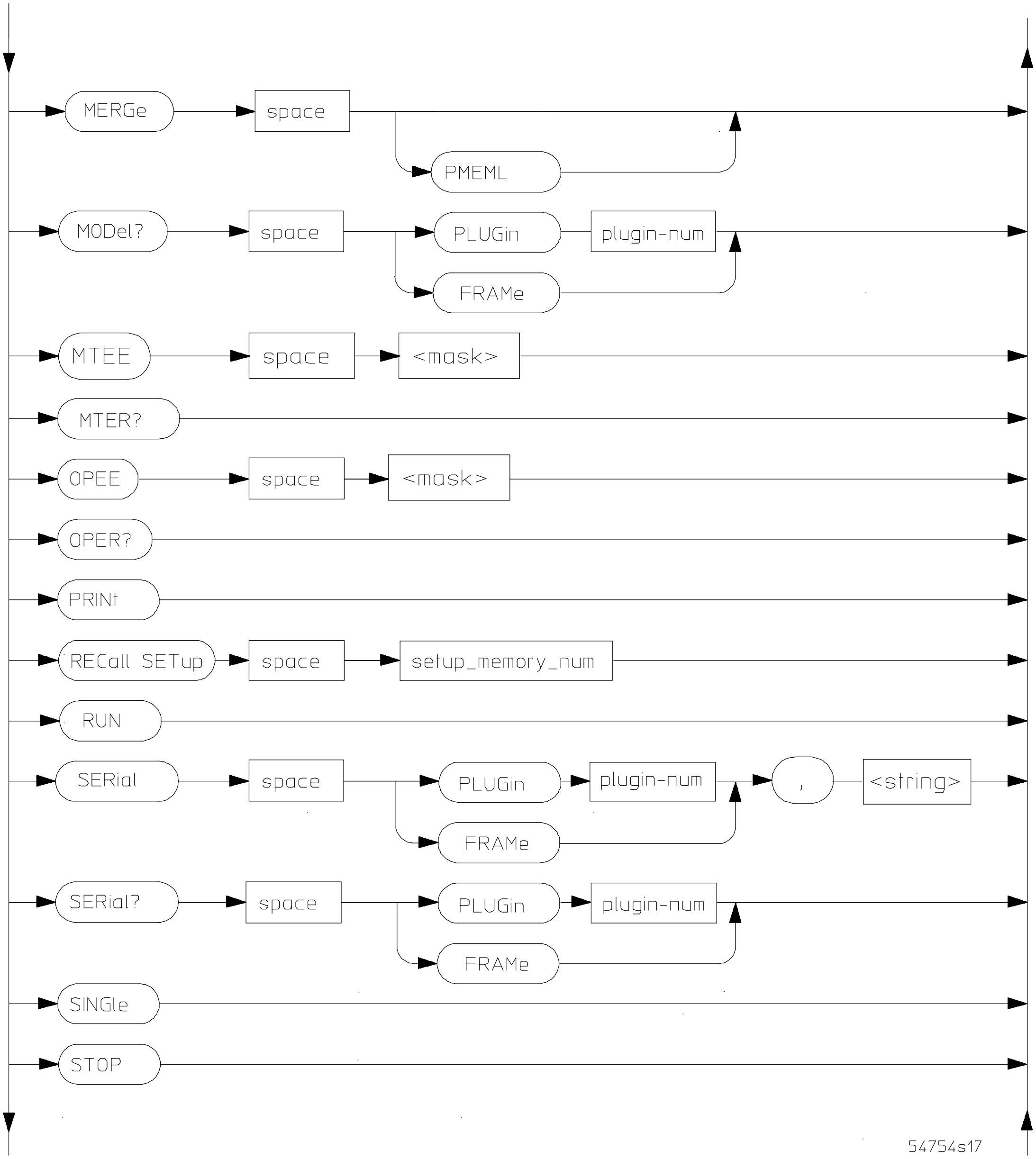
- BLANk
- COMMents
- DIGitize
- MENU
- STORe:WAVeform
- VIEW



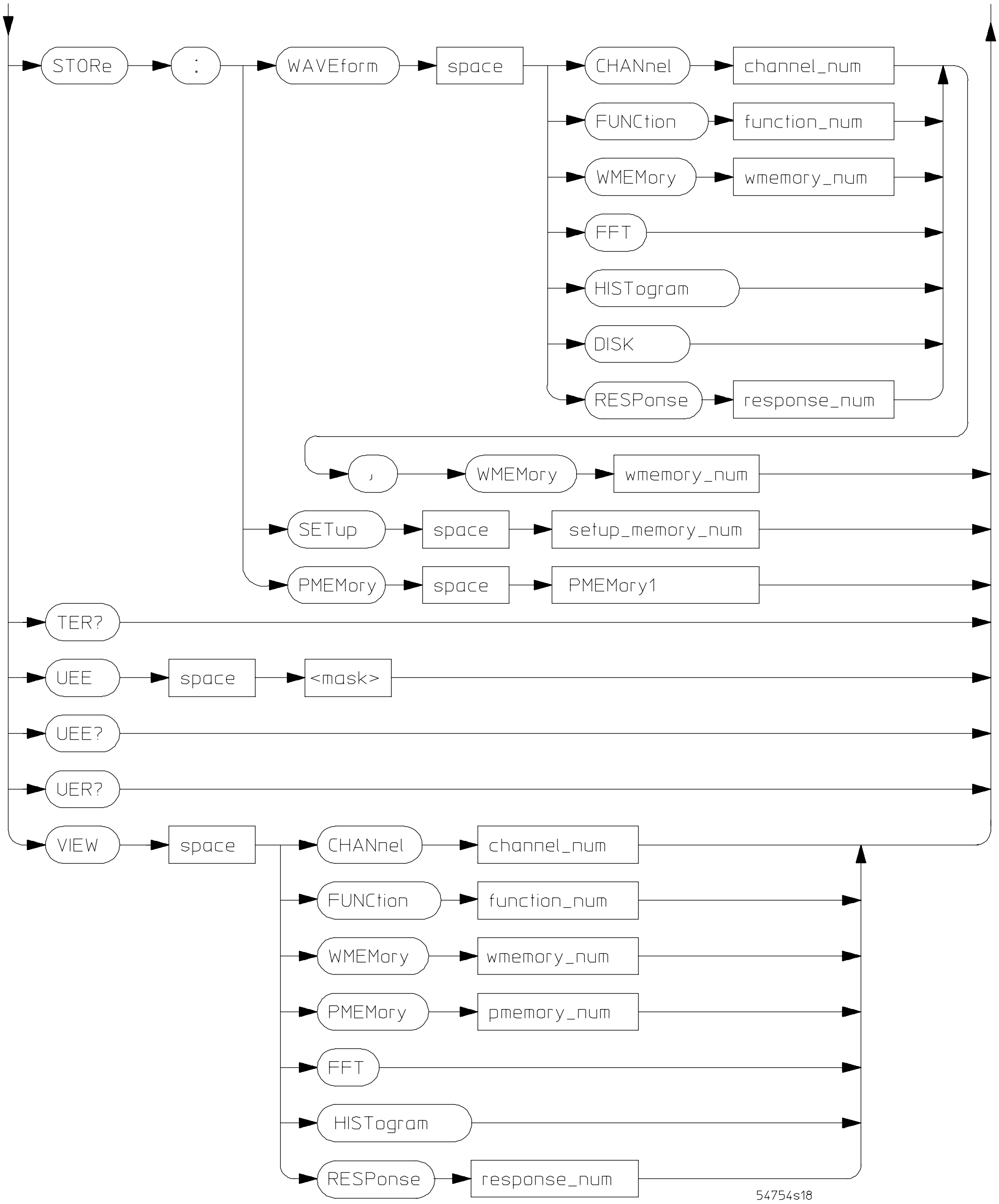
Root Level Command Syntax Diagram



Root Level Command Syntax Diagram (continued)



Root Level Command Syntax Diagram (continued)



Root Level Command Syntax Diagram (continued)

BLANk

Command

```
:BLANk {CHANnel<number> | WMEMory<number> |
FUNCtion <number> | FFT | RESPonse<number> |
HISTogram | PMEMory1}
```

Turns off an active channel, function, waveform memory, pixel memory, FFT, histogram, or TDR response. The VIEW command turns them on.

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For pixel memories (PMEMory): 1. For TDR response: 1, 2, 3, or 4.

Example

The following example turns off TDR response 1.

```
10 OUTPUT 707;":BLANK:RESPONSE1"
20 END
```

COMMents

Command

```
:COMMents {PLUGin<number>, <comments>}
```

Sets the comments field for the plugin. This field is used to describe options included in the plugin, or for user commands about the plugin.

`<number>` Represents the plugin slot numbered 1 through 4.

`<comments>` Represents the unquoted string.

Example

The following example turns off TDR response 1.

```
10 OUTPUT 707;":COMMENTS PLUGIN1"
20 END
```

Query

```
:COMMents? PLUGin<number>
```

Returns the comments field associated with the plugin.

DIGitize

Command

```
:DIGitize [<waveform_name>][,<waveform_name>]
```

Invokes a special mode of data acquisition that is more efficient than using the RUN command. This command initializes the selected channels, functions, responses, or FFT to “unacquired,” then acquires them according to the current instrument settings. When all signals are completely acquired, the instrument is stopped.

If channel, function, response, or FFT parameters are specified, then these are the only waveforms acquired. To speed acquisition, these waveforms are not displayed and their display state is “off.” Subsequent to the digitizing operation, the waveform display may be turned on, if desired. Other sources are turned off and their data are invalidated.

You use the ACQuire subsystem to set up conditions, such as TYPe and COUNt, for the next DIGitize command. Use the WAVeform subsystem commands to specify how data is transferred from the instrument to the controller and how to interpret the data.

NOTE

Even though digitized waveforms are not displayed, the full range of measurement and math operators may be used on them.

If the DIGitize command is issued without parameters, the digitize operation is performed on the channels, functions, responses, and FFT that were acquired with a previous digitize, run, or single operation. In this case, the display state of the acquired waveforms is not changed. Because the command executes more quickly without parameters, this form of the command is useful for repetitive measurement sequences. Also, this mode can be used if it is desirable to view the digitized results because the display state of the digitized waveforms is not affected.

The data acquired with the DIGitize command are placed in the normal channel, function, response, or FFT memories just as when acquired via the RUN command.

`<waveform_name>` {CHANnel<number> | FUNCtion<number> | FFT | RESPonse<number>}

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For responses: 1, 2, 3, or 4.

Example

The following example uses the DIGitize command to acquire data on response 1 and function 2.

```
10 OUTPUT 707;":DIGITIZE RESPONSE1,FUNCTION2"
20 END
```

MENU

Command

```
:MENU {APPLication | CHANnel<number> | ACQuire |
TIMebase | TRIGger | DISK | DISPlay | MARker |
MEASure | MATH | WAVeform | SETup | PRINt |
HELP | UTILity | FFT | LTESt | HISTogram |
MTESt | MEYE | TDR<number> | MTDR<number>}
```

`<number>` For channels, indicates the slot (1 through 4) in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced.
For TDR, controls the TDR/TDT menus for channels 2 and 4.
For MTDR, controls the TDR/TDT menus for channels 2 and 4.

Example The following example allows you to control the TDR menu for channel 2.

```
10 OUTPUT 707;":MENU TDR2"
20 END
```

STORe:WAVEform

Command

```
:STORe:WAVEform {CHANnel<number> |
FUNCtion<number> | WMEMory<number> | HISTogram |
FFT | RESPonse<number>},{WMEMory<number>}
```

Copies a channel, function, stored waveform, histogram, TDR response, or FFT to a waveform memory. The first parameters specifies the source of the copy and can be any channel, function, waveform memory, histogram, TDR response, or the FFT. The second parameter is the destination of the copy, and can be any waveform memory.

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3 or 4. For responses: 1, 2, 3, or 4.

Example

The following example copies response 1 to waveform memory 3.

```
10 OUTPUT 707;":STORE:WAVEFORM RESPONSE1,WMEMORY3"
20 END
```

VIEW

Command

```
:VIEW {CHANnel<number> | WMEMory<number> |
FUNCtion <number> | FFT | RESPonse<number> |
HISTogram | PMEMory1}
```

Turns on a channel, function, pixel memory, waveform memory, TDR response, histogram, or FFT. Use the BLANk command to turn these off.

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4. For pixel memories (PMEMory): 1.

Example

The following example turns on TDR response 1.

```
10 OUTPUT 707;":VIEW RESPONSE1"
20 END
```

3

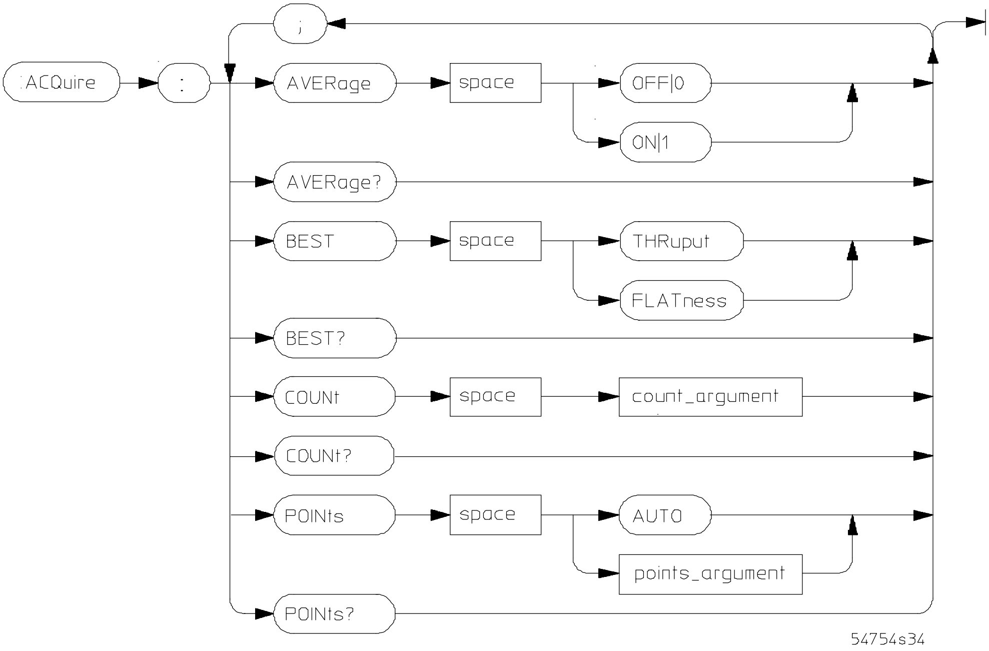
Acquire Commands

Acquire Commands

The ACQuire subsystem commands set up conditions for executing a DIGitize root level command to acquire waveform data. The commands in this subsystem select the type of data, the number of averages, and the number of data points.

The TDR plug-in adds the following ACQuire subsystem command:

- BEST



Acquire Subsystem Commands Syntax Diagram

BEST

Command

```
:ACQuire:BEST {THRuput | FLATness}
```

When averaging is enabled with ACQuire:AVERage, the FLATness option improves the step flatness by using a signal processing algorithm within the instrument. You should use this option when performing TDR measurements or when step flatness is important. The THRuput option improves the instrument's throughput and should be used whenever best flatness is not required.

Example

The following example sets the instrument to best step flatness:

```
10 OUTPUT 707;":ACQUIRE:BEST FLATNESS"
20 END
```

Query

```
:ACQuire:BEST?
```

The :ACQuire:BEST? query returns the current acquisition algorithm setting.

Returned Format

```
[:ACQuire:BEST]{THRuput | FLATness}<NL>
```

Example

The following example obtains the current setting of the acquisition algorithm from the instrument, stores it in the variable Best, then prints the contents of the variable to the controller's screen:

```
10 DIM Best$[50]       !Define variable
20 OUTPUT 707;":ACQUIRE:BEST?"
30 ENTER 707;Best$
40 PRINT Best$
50 END
```

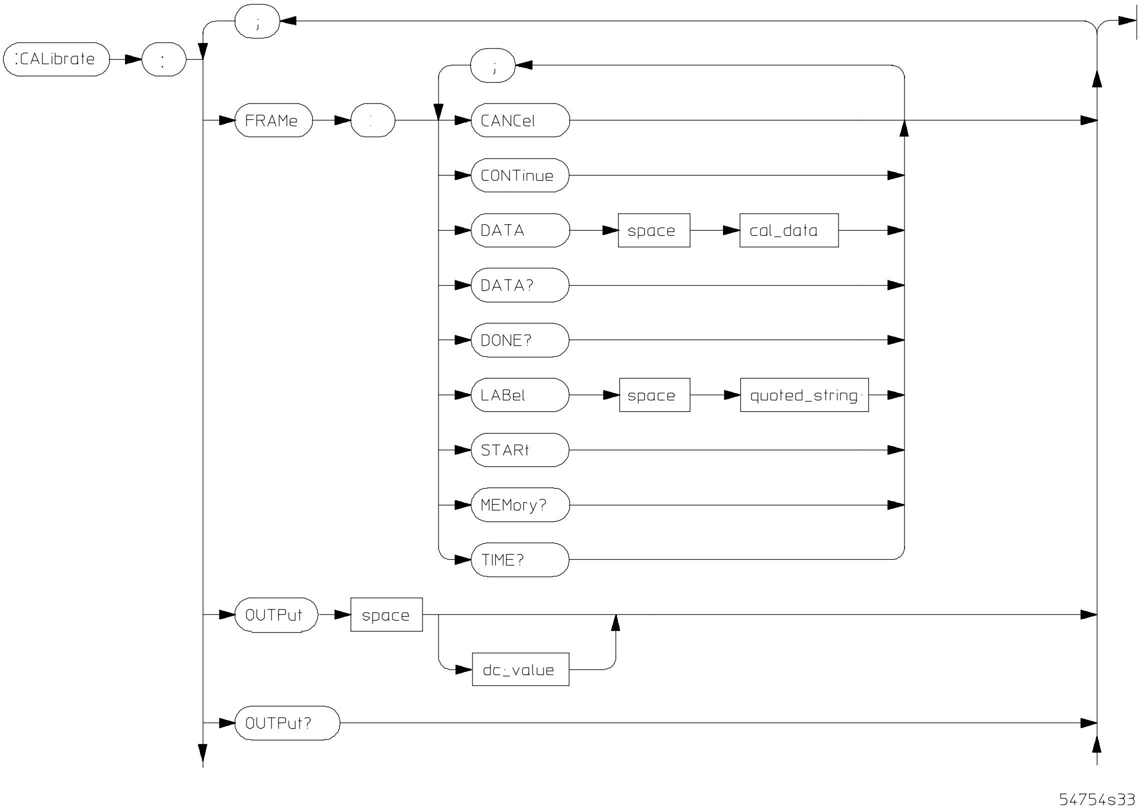
4

Calibrate Commands

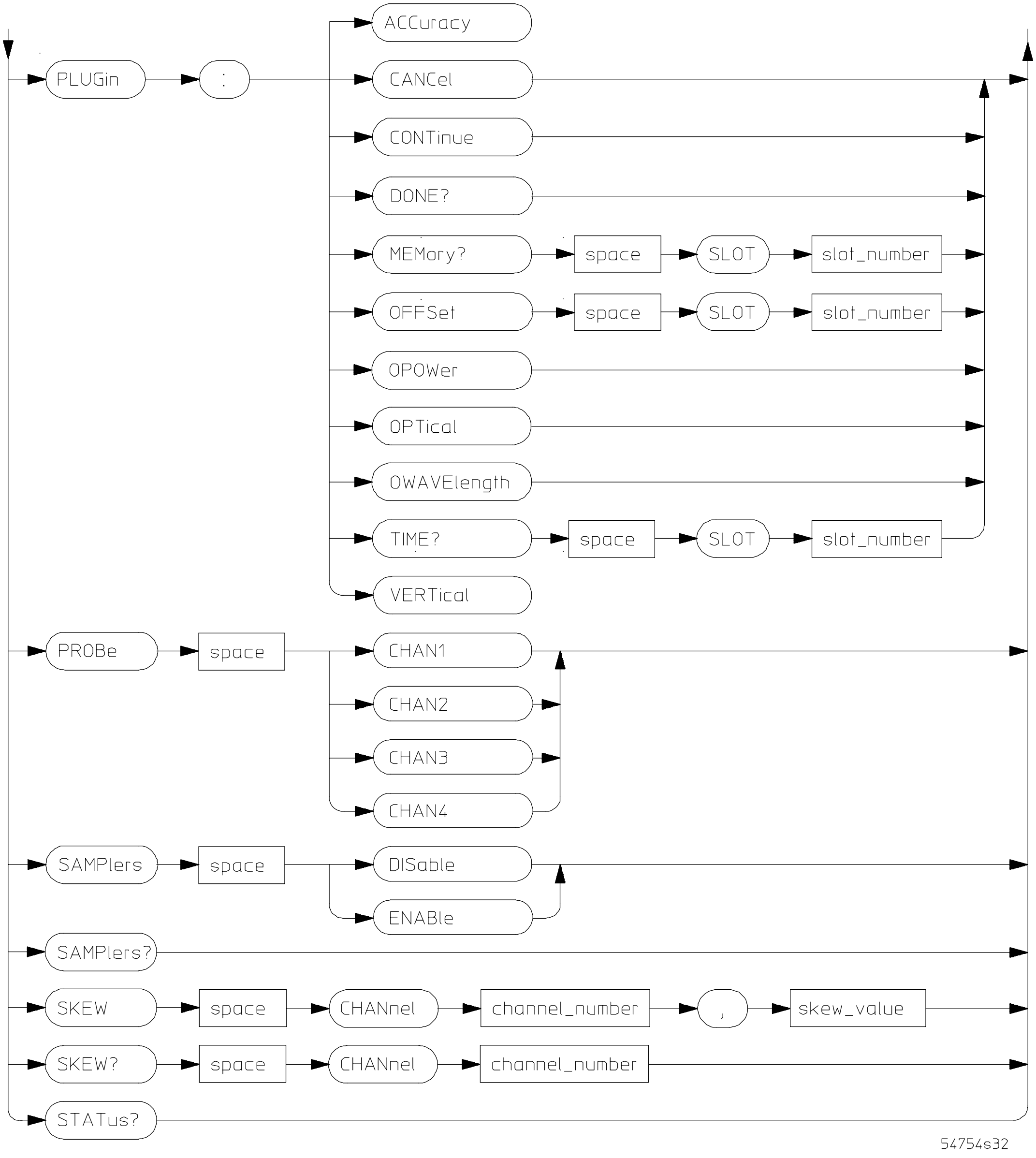
Calibrate Commands

The commands in the Calibration subsystem initiate instrument calibration over the GPIB. Firmware revision A.02.00 adds the following calibration command to the calibration subsystem:

- SAMPlers



Calibration Subsystem Syntax Diagram



Calibration Subsystem Syntax Diagram (continued)

SAMPlers

Command

```
:CALibrate:SAMPlers {DISable | ENABle}
```

Enables or disables the samplers in the plug-in.

Example

The following example enables sampler calibration for the TDR plug-in.

```
10 OUTPUT 707;":CALIBRATE:SAMPLERS ENABLE"
20 END
```

Query

```
:CALibrate:SAMPlers?
```

Returns the current calibration enable/disable setting.

Returned Format

```
[:CALibrate:SAMPlers]{DISable | ENABle}<NL>
```

Example

The following example gets the current setting for sampler calibration, stores it in the variable Sampler, and prints the contents of the variable to the screen.

```
10 DIM Sampler$[50]                    !Define variable
20 OUTPUT 707;":CALIBRATE:SAMPLERS?"
30 ENTER 707;Sampler$
40 PRINT Sampler$
50 END
```

5

Channel Commands

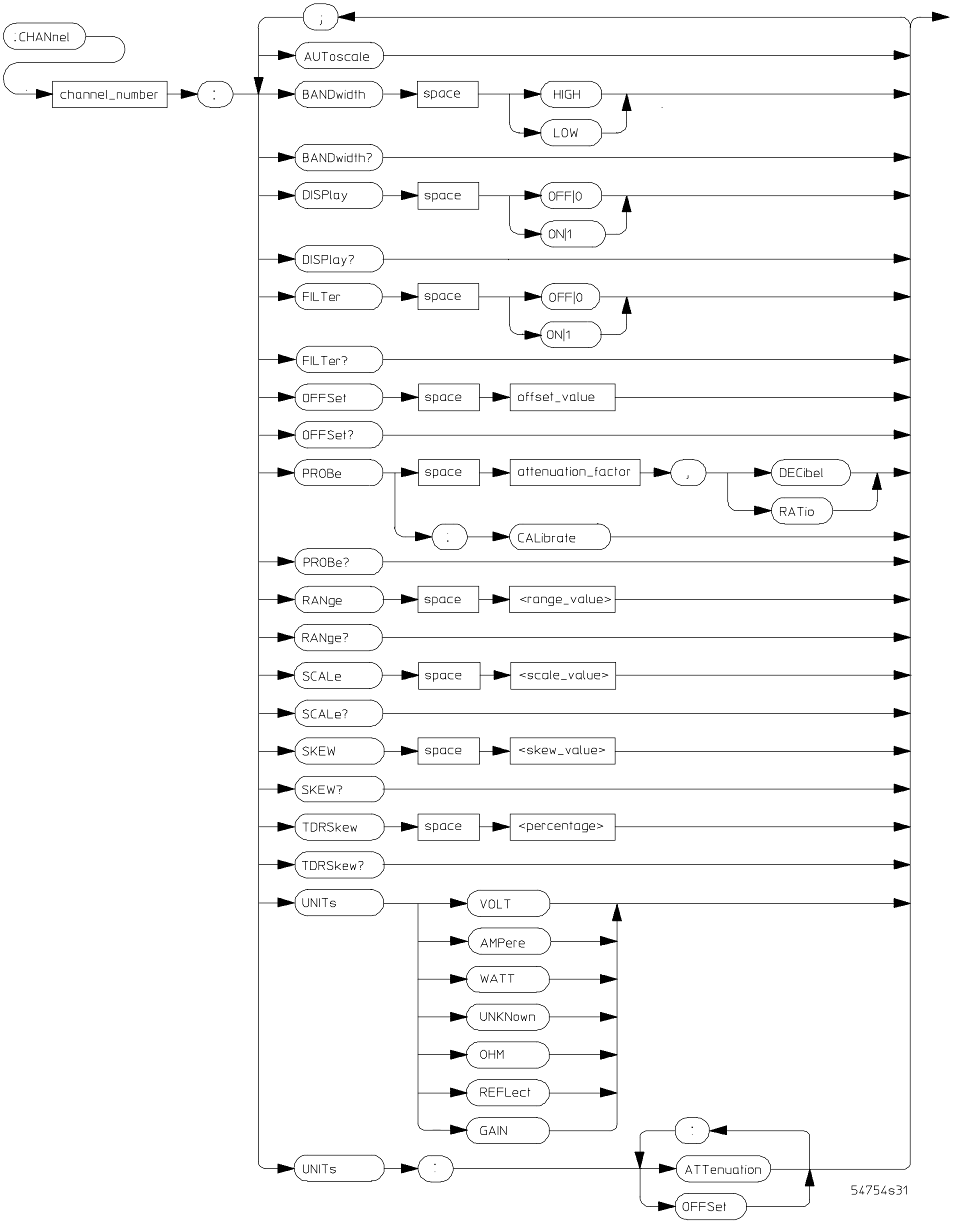
Channel Commands

The CHANnel subsystem commands control all vertical (Y axis) functions of the oscilloscope. The options for the channel subsystem commands vary depending on which plug-in you are using.

The channel displays may be toggled on and off with the root level commands VIEW and BLANk.

The TDR plug-in modifies the operation of the following channel commands:

- OFFSet
- RANGe
- SCALe
- TDRSkew
- UNITs



Channel Subsystem Commands Syntax Diagram

OFFSet

Command

```
:CHANnel<number>:OFFSet <offset_value>
```

Sets the voltage that is representated at center screen (two divisions from the bottom for optical channels) for the selected channel. Offset parameters are plug-in, probe, and vertical scale dependent.

For TDR and TDT applications, when the TDR stimulus is set to differential or common mode, or when OHM, REFLect, or GAIN units are selected, the instrument will change offset to magnify offset. This command is used to set the magnify offset as well as the offset.

`<number>` An integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced.

`<offset_value>` Offset value at center screen. Usually expressed in volts, but could be in other measurement units, such as amperes, if you have specified other units using the CHANnel:UNITs command.

Example The following example sets the offset for channel 1 to 125 mV:

```
10 OUTPUT 707;":CHANNEL1:OFFSET 125 MV"
20 END
```

Query

```
:CHANnel<number>:OFFSet?
```

The query returns the current offset value for the specified channel.

Returned Format

```
[:CHANnel<number>:OFFSet]<offset_value><NL>
```

Example

The following example places the offset value of channel 1 into the string variable Offset$, then prints the contents of that variable to the controller's screen:

```
10 DIM Offset$[50]
20 OUTPUT 707;":CHANNEL1:OFFSET?"
30 ENTER 707;Offset$
40 PRINT Offset$
50 END
```

RANGe

Command

```
:CHANnel<number>:RANGe <range_value>
```

Defines the full-scale vertical axis of the selected channel. It sets up acquisition and display hardware to display the waveform at a given range scale. The values represent the full-scale deflection factor of the vertical axis in volts. These values change as the probe attenuation factor is changed.

For TDR and TDT applications, when the TDR stimulus is set to differential or common mode, or when OHM, REFLect, or GAIN units are selected, the instrument will change scale to magnify scale. This command is used to set the magnify range as well as the range.

`<number>` An integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced.

`<range_value>` Full-scale voltage of the specified channel number. Available ranges are dependent on the plug-in and attenuation factor.

Example

The following example sets the full-scale range for channel 1 to 500 mV:

```
10 OUTPUT 707;":CHANNEL1:RANGE 500 MV"
20 END
```

Query

```
:CHANnel<number>:RANGe?
```

The query returns the current full-scale vertical axis setting for the selected channel.

Returned Format

```
[:CHANnel<number>:RANGe]<range_value><NL>
```

Example

The following example places the range value of channel 1 into the string variable Range$, then prints the contents of that variable to the controller's screen:

```
10 DIM Range$[50]
20 OUTPUT 707;":CHANNEL1:RANGE?"
30 ENTER 707;Range$
40 PRINT Range$
50 END
```

SCALe

Command

```
:CHANnel<number>:SCALe <scale_value>
```

Defines the vertical scale of the channel in units per division. This command is the same as the front-panel channel scale.

For TDR and TDT applications, when the TDR stimulus is set to differential or common mode, or when OHM, REFLect, or GAIN units are selected, the instrument will change scale to magnify scale. This command is used to set the magnify scale as well as the scale.

`<number>` An integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced.

`<scale_value>` Vertical scale of the channel in units per division.

Example

The following example sets the scale for channel 1 to 500 mV:

```
10 OUTPUT 707;":CHANNEL1:SCALE 500 MV"
20 END
```

Query

```
:CHANnel<number>:SCALe?
```

The query returns the current scale setting for the specified channel.

Returned Format

```
[:CHANnel<number>:SCALe]<scale_value><NL>
```

Example

The following example places the scale value of channel 1 into the string variable Scale$, then prints the contents of that variable to the controller's screen:

```
10 DIM Scale$[50]
20 OUTPUT 707;":CHANNEL1:SCALE?"
30 ENTER 707;Scale$
40 PRINT Scale$
50 END
```

TDRSkew

Command

```
:CHANnel<number>:TDRSkew <percent> [%]
```

This command sets the TDR skew for the given channel. The TDR skew control moves the TDR step relative to the trigger position. The control may be set from -100 to 100 percent of the allowable range. This command is only applicable to TDR channels.

`<number>` An integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced.

`<percent>` A number between -100 and 100, used to set the step position.

Example

The following example sets the TDR skew for channel 1 to 20%:

```
10 OUTPUT 707;":CHANNEL1:TDRSKEW 20"
20 END
```

Query

```
:CHANnel<number>:TDRSkew?
```

The query returns the current TDR skew setting for the specified channel.

Returned Format

This query returns the TDR skew value in percent of allowable range from -100 to 100 percent. This command is only applicable to TDR channels. The returned format is a real number.

UNITs

Command

```
:CHANnel<number>:UNITs {VOLT | AMPere | WATT |
UNKNown | OHM | REFLect | GAIN}
```

Allows you to work in vertical units other than volts. The units command changes the name of the Y-axis units from VOLT to AMPere, WATT, or UNKNown. The units are implied for other pertinent channel commands, such as RANGe and OFFSet.

For TDR and TDT applications, you can also select the units OHM, REFLect, and GAIN. OHM and REFLect may be selected for TDR channels, while GAIN may be selected for TDT channels. You must establish the normalization and reference plane to select these units.

`<number>` An integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced.

Example

The following example sets the units for channel 1 to ohms:

```
10 OUTPUT 707;":CHANNEL1:UNITS OHM"
20 END
```

Query

```
:CHANnel<number>:UNITs?
```

The query returns the current units setting for the specified channel.

Returned Format

```
[:CHANnel<number>:UNITs]{VOLT | AMPere | WATT | UNKNown | OHM |
REFLect | GAIN}<NL>
```

Example

The following example places the vertical units for the specified channel into the string variable Units$, then prints the contents of that variable to the controller's screen:

```
10 DIM Units$[50]
20 OUTPUT 707;":CHANNEL:UNITS?"
30 ENTER 707;Units$
40 PRINT Units$
50 END
```

6

Disk Commands

Disk Commands

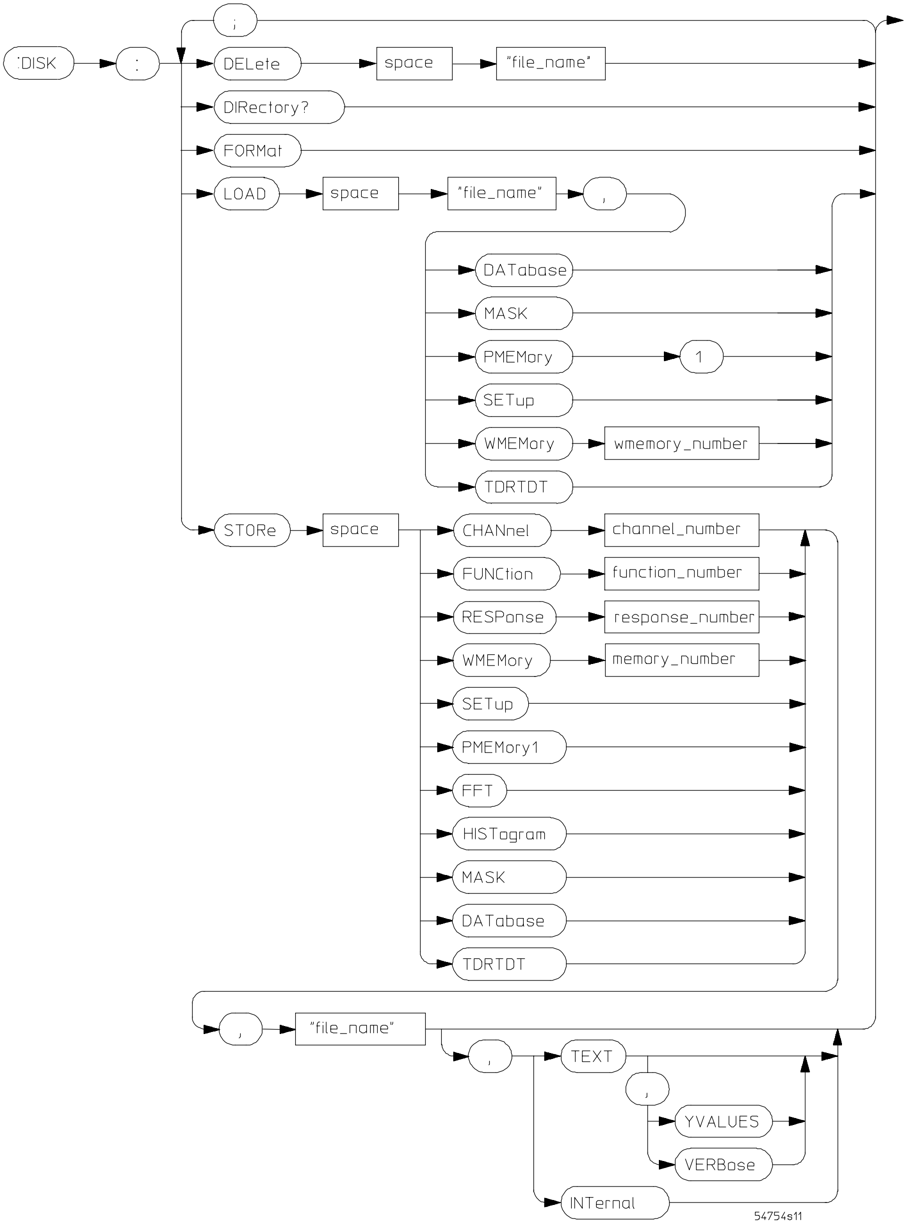
The DISK subsystem commands perform the disk operations as defined under the disk menu of the front panel. This allows storage and retrieval of waveforms, setups and pixel memory, as well as formatting the disk.

The filenames used are compatible with MS-DOS. They consist of up to 8 characters for the name with a 0 to 3 character extension, separated by a "." (dot). File names are all upper case but can be entered in either upper or lower case; they will be forced to upper case internally. Valid characters are [A-Z, 0-9, _ (the underscore character)].

> **NOTE**
> The filename must be enclosed in quotes in the disk commands.

The TDR plug-in modifies the operation of the following disk commands:

- LOAD
- STORe



Disk Subsystem Command Syntax Diagram

LOAD

Command

```
:DISK:LOAD "<filename>", <source>
```

Loads a setup, waveform, database, mask, TDR/TDT calibration, or pixel memory from the disk. The filename does not include a suffix. The suffix supplied by the instrument depends on the source and file format specified. The TDRTDT option is a file type choice used to load TDR/TDT calibration values into the instrument.

`<source>` {DATabase | MASK | PMEMory 1 | SETup | WMEMory<number> | TDRTDT}

`<filename>` Is an MS-DOS compatible file name up to 8 characters long.

`<number>` Represents waveforms 1, 2, 3, or 4.

Example

The following example loads the file "CALIBR" which contains the TDR/TDT calibrations:

```
10 OUTPUT 707;":DISK:LOAD "CALIBR", TDRTDT
20 END
```

STORe

Command `:DISK:STORe <source>, "<filename>"[,<format>]`

Stores a setup, waveform, database, mask, TDR response, TDR/TDT calibration, or pixel memory to the disk. The filename does not include a suffix. The suffix is supplied by the instrument depending on the source and file format specified. The database may only be saved in internal format. The TDRTDT option is a file type choice used to store the instrument's TDR/TDT calibration values.

`<source>` {CHANnel<number> | FUNCtion<number> | WMEMory<number> | SETup | PMEMory1 | HISTogram | DATabase | FFT | MASK | RESPonse<number> | TDRTDT}

`<number>` For channels: 1, 2, 3, or 4, optionally followed by A or B. For functions: 1 or 2. For waveforms: 1, 2, 3, or 4. For responses: 1, 2, 3, or 4.

`<filename>` Is an MS-DOS compatible file name up to 8 characters long.

`<format>` {TEXT [,YVALues | VERBose] | INTernal}

> NOTE
>
> The format field is for waveforms. The default is INTernal. In TEXT mode, y values may be specified so that only Y values are stored. VERBose is the default in which Y values and the waveform preamble are stored. See the WAVeform Commands chapter in the 54750/83480 Programmer's Guide for information on converting data to values.

Example

The following examples stores TDR response 4 in the file named NEWRESP.RSP:

```
10 OUTPUT 707;":DISK:STORE RESPONSE4,""NEWRESP.RSP"""
20 END
```

7

Display Commands

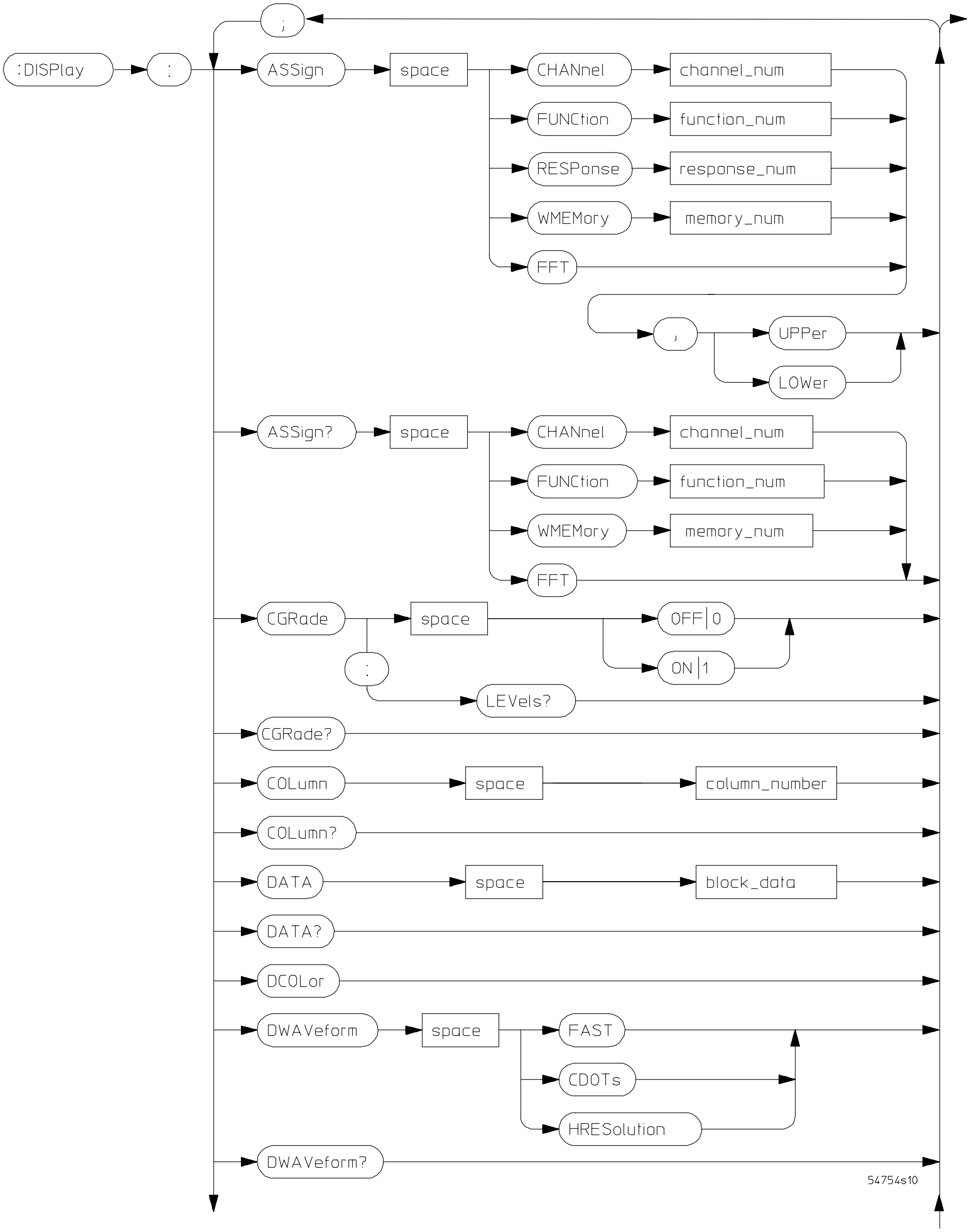
Display Commands

The DISPlay subsystem controls the display of data, markers, text, and graticules, and also controls the use of color.

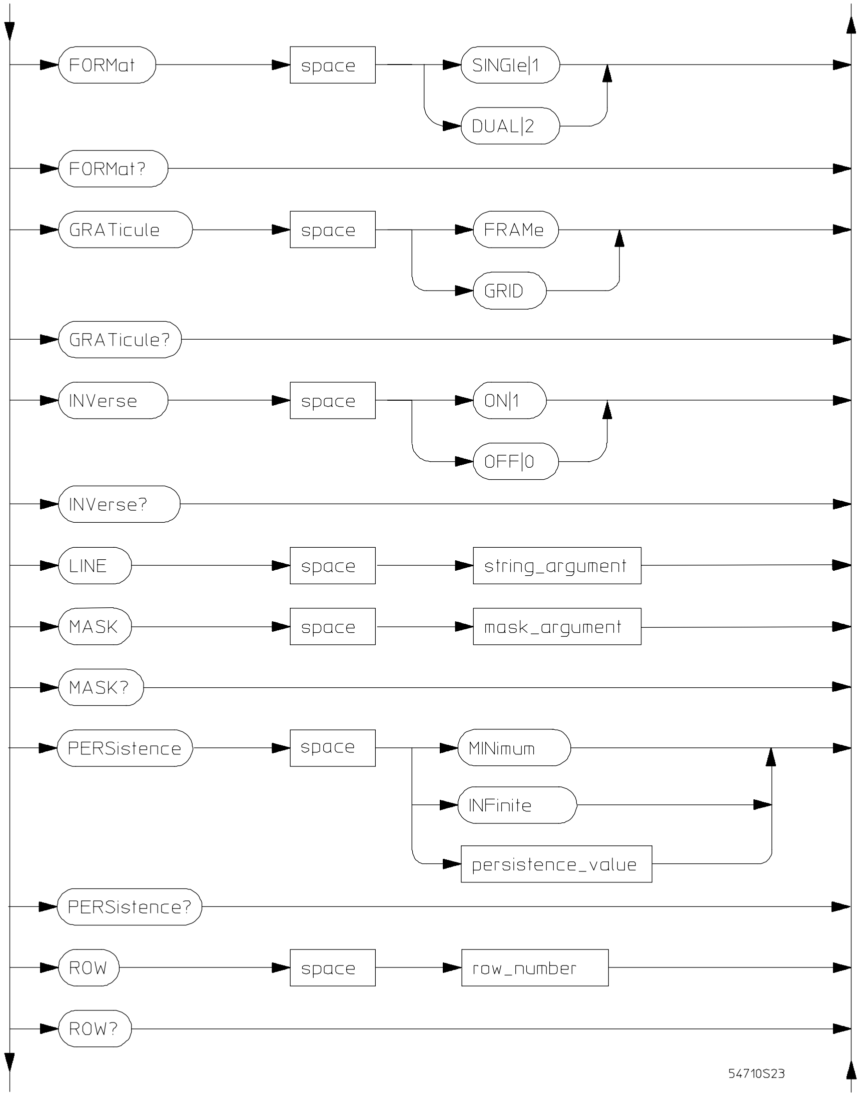
The display mode is selected by the ACQuire:TYPe command. The number of averages is selected by the ACQuire:COUNt command.

The TDR plug-in module modifies the operation of the following DISPlay subsystem command:

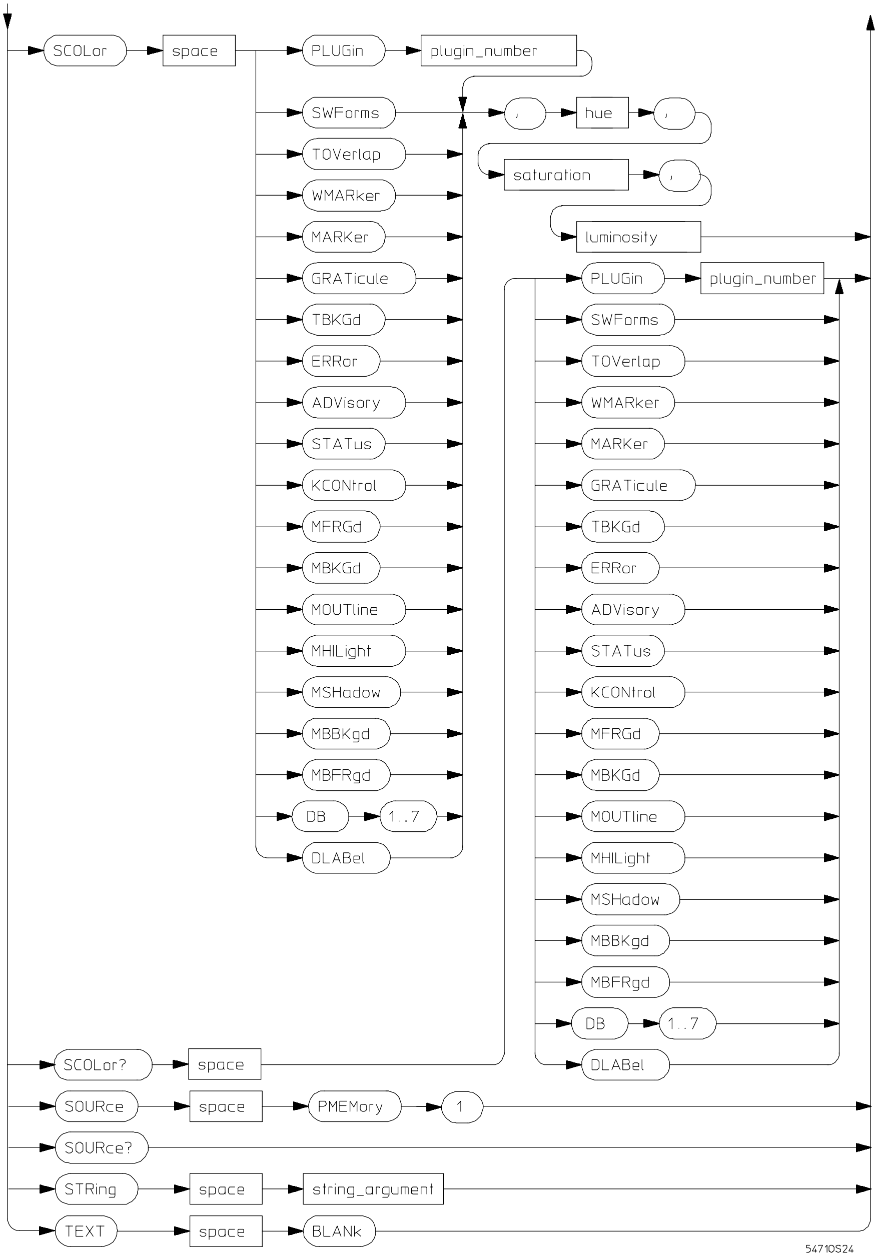
- ASSign



Display Subsystem Command Syntax Diagram



Display Subsystem Command Syntax Diagram (continued)



Display Subsystem Command Syntax Diagram (continued)

ASSign

Command

```
:DISPlay:ASSign {CHANnel<number> |
FUNCtion<number> | RESPonse<number> |
WMEMory<number> | FFT},{UPPer | LOWer}
```

Assigns the specified channel, waveform, function, TDR response, or FFT to a particular portion of the waveform area on the screen. This command has no effect when the graticule format is single.

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

Example

The following example assigns TDR response 1 to the upper portion of the oscilloscope screen.

```
10 OUTPUT 707;":DISPLAY:ASSIGN RESPONSE1,UPPER"
20 END
```

Query

```
:DISPlay:ASSign? {CHANnel<number> |
FUNCtion<number> | RESPonse<number> |
WMEMory<number> | FFT}
```

The query returns the portion of the waveform area where the specified waveform is displayed or was last displayed if it is not currently on screen.

Returned Format

```
[:DISPlay:ASSign? {CHANnel<number> | FUNCtion<number> |
RESPonse<number> | WMEMory<number> | FFT}]{UPPer | LOWer}<NL>
```

Example

The following example returns the portion of the waveform area where the TDR response 1 waveform resides to the string variable Setting$, then prints it to the controller's screen.

```
10 DIM Setting$[50]     !Dimension variable
20 OUTPUT 707;":DISPLAY:ASSIGN? RESPONSE1"
30 ENTER 707;Setting$
40 PRINT Setting$
50 END
```

8

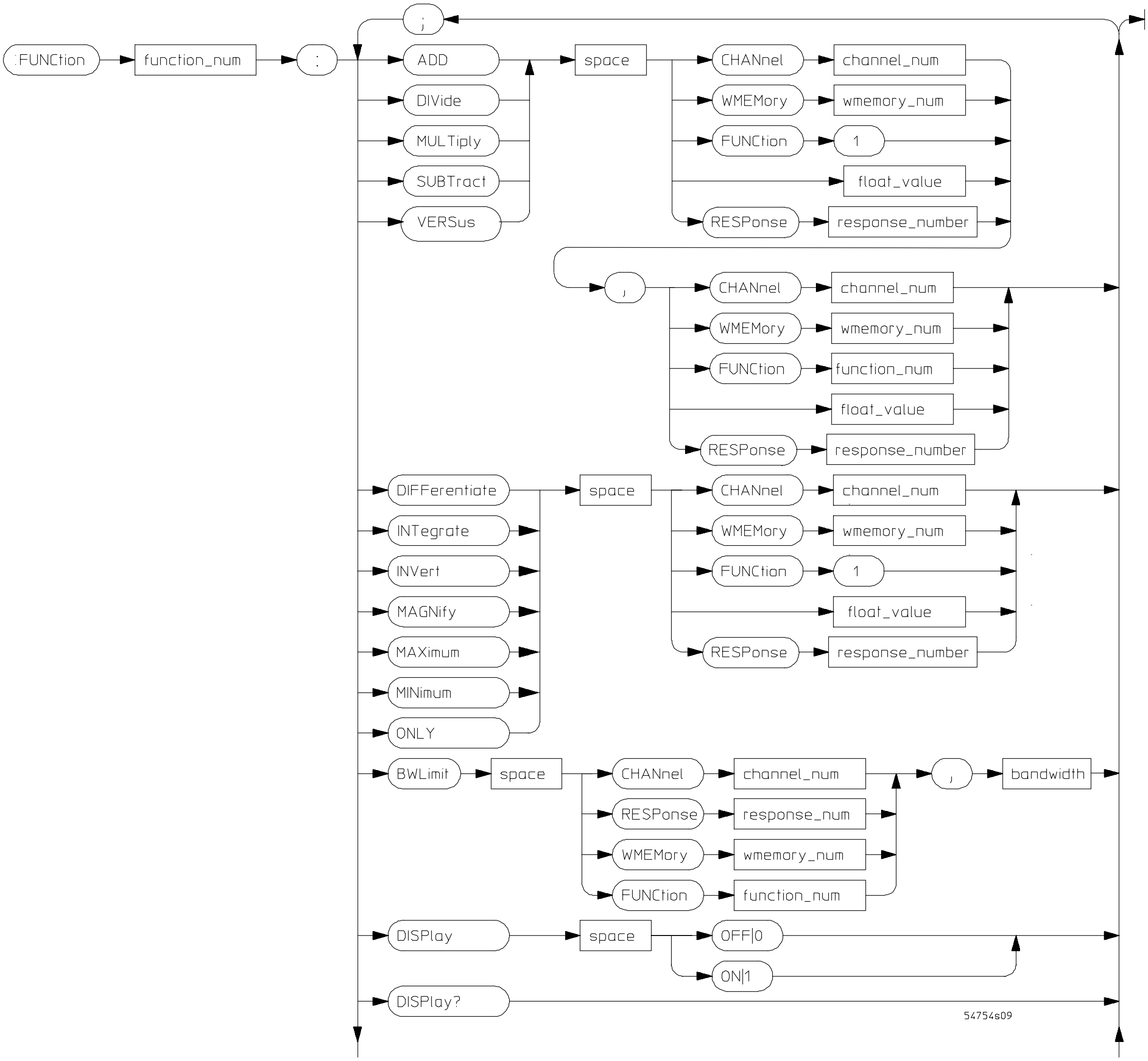
Function Commands

Function Commands

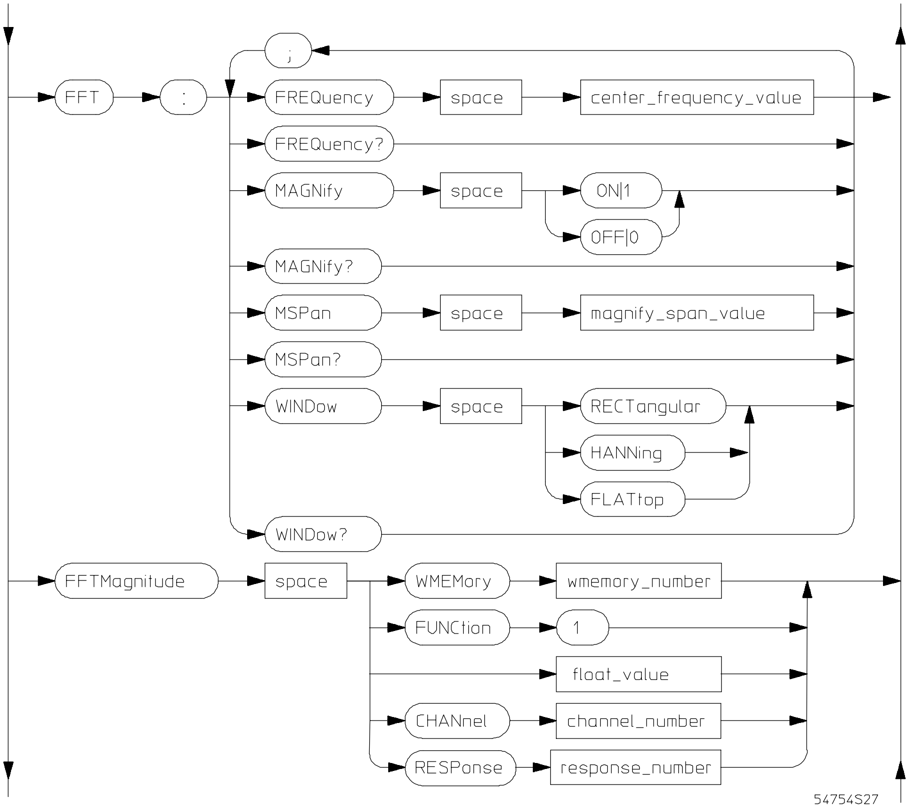
The FUNCtion subsystem defines two functions, function 1 and function 2. The operands of these two functions can be any installed channels in the oscilloscope, TDR responses, waveform memories 1 through 4, a function definition, or a constant.

The A.02.00 firmware revision to the FUNCtion subsystem modifies the following commands:

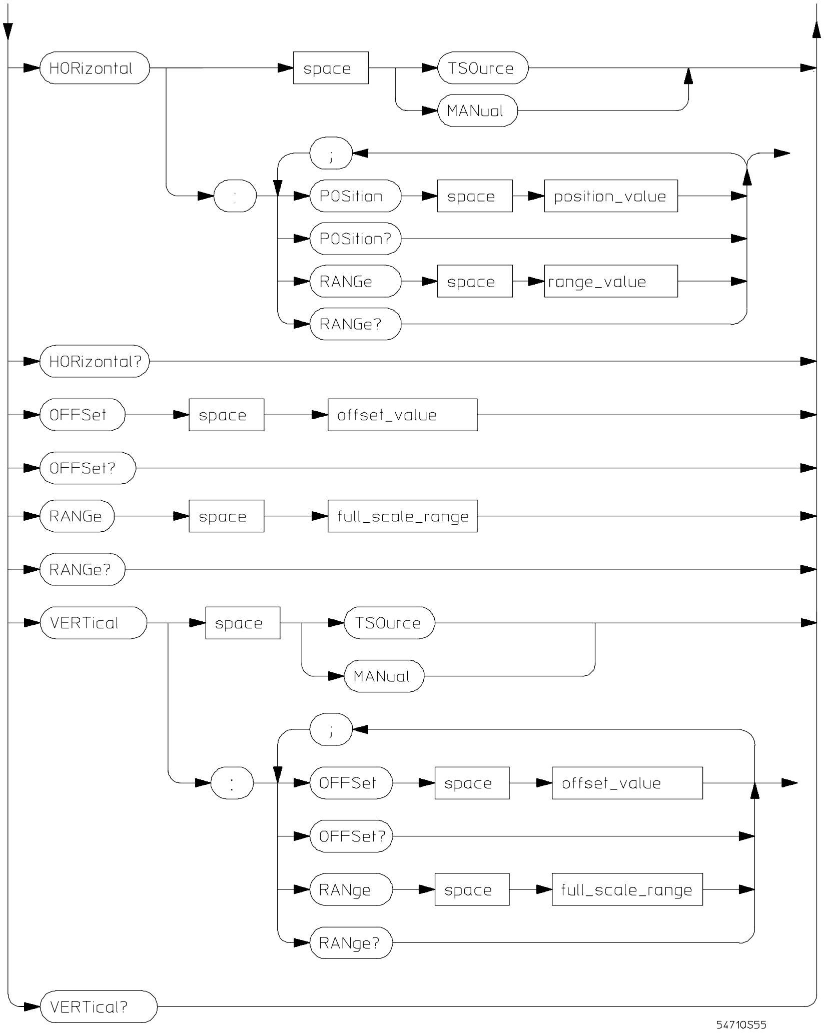
- ADD
- BWLimit
- DIFFerentiate
- DIVide
- FFTMagnitude
- INTegrate
- INVert
- MAGNify
- MAXimum
- MINimum
- MULTiply
- SUBTract
- VERSus



Function Subsystem Command Syntax Diagram



Function Subsystem Command Syntax Diagram (continued)



Function Subsystem Command Syntax Diagram (continued)

ADD

Command

```
:FUNCtion{1 | 2}:ADD <source>[,<source>]
```

Defines a function that takes the algebraic sum of two defined operands.

`<source>` {CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number> | <float_value>}

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

`<float_value>` A real number.

Example

The following example sets up function 1 to add channel 1 to channel 2:

```
10 OUTPUT 707;":FUNCTION1:ADD CHANNEL1,CHANNEL2"
20 END
```

BWLimit

Command

```
:FUNCtion{1 | 2}:BWLimit {{CHANnel<number> |
FUNCtion<number> | RESPonse<number> |
WMEMory<number>}, <bandwidth>}
```

Provides a bandwidth limit filter function. You can use this function on waveforms to limit the bandwidth of the signal. In TDR or TDT applications where the accuracy of normalization is not required, you can use this to simulate a bandlimited signal through a device under test.

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

`<bandwidth>` Real number indicating frequency limit for the specified function.

Example

The following example defines function 1 as a bandwidth limit for TDR response 1 of 100 MHz:

```
10 OUTPUT 707;":FUNCTION1:BWLIMIT RESPONSE1,100 MHZ"
20 END
```

DIFFerentiate

Command

```
:FUNCtion{1 | 2}:DIFFerentiate {CHANnel<number> |
FUNCtion<number> | RESPonse<number> |
WMEMory<number> | <float_value>}
```

Computes the discrete derivative of the defined operand's waveform.

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

`<float_value>` A real number.

Example

The following example sets up function 1 to take the discrete derivative of the signal on channel 2:

```
10 OUTPUT 707;":FUNCTION1:DIFFERENTIATE CHANNEL2"
20 END
```

DIVide

Command `:FUNCtion{1 | 2}:DIVide <source>[,<source>]`

Defines a function that divides the first operand by the second operand.

`<source>` `{CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number> | <float_value>}`

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

`<float_value>` A real number.

Example The following example sets up function 1 to divide the signal on channel 1 by the signal in waveform memory 4:

```
10 OUTPUT 707;":FUNCTION1:DIVIDE CHANNEL1,WMEMORY4"
20 END
```

FFTMagnitude

Command `:FUNCtion{1 | 2}:FFTMagnitude {CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number> | <float_value>}`

Defines a function that computes the FFT magnitude.

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

`<float_value>` A real number.

Example

The following example computes the FFTmagnitude for channel 1:

```
10 OUTPUT 707;":FUNCTION1:FFTMAGNITUDE CHANNEL1"
20 END
```

INTegrate

Command

```
:FUNCtion{1 | 2}:INTegrate {CHANnel<number> |
FUNCtion<number> | RESPonse<number> |
WMEMory<number> | <float_value>}
```

Defines a function that computes the integral of the defined operand's waveform.

<number> For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

<float_value> A real number.

Example

The following example sets up function 1 to compute the integral of waveform memory 3:

```
10 OUTPUT 707;":FUNCTION1:INTEGRATE WMEMORY3"
20 END
```

INVert

Command

```
:FUNCtion{1 | 2}:INVert {CHANnel<number> |
FUNCtion<number> | RESPonse<number> |
WMEMory<number> | <float_value>}
```

Defines a function that inverts the defined operand’s waveform by multiplying by -1.

<number> For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

<float_value> A real number.

Example

The following example sets up function 1 to invert the signal on channel 1:

```
10 OUTPUT 707;":FUNCTION1:INVERT CHANNEL1"
20 END
```

MAGNify

Command

```
:FUNCtion{1 | 2}:MAGNify {CHANnel<number> |
FUNCtion<number> | RESPonse<number> |
WMEMory<number> | <float_value>}
```

Defines a function that is a copy of the operand. The magnify function is a software magnify. No hardware settings are altered as a result of using this function. It is useful for scaling channels, another function, TDR/TDT responses and memories with the RANGE and OFFSET commands in this subsystem.

Magnify performs the same operation as the "ONLY" operator, and is the preferred operator of the two. "ONLY" is included in this instrument for compatibility with previous instruments.

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

`<float_value>` A real number.

Example

The following example creates a function (function 1) that is a magnified copy of channel 1:

```
10 OUTPUT 707;":FUNCTION1:MAGNIFY CHANNEL1"
20 END
```

MAXimum

Command

```
:FUNCtion{1 | 2}:MAXimum {CHANnel<number> |
FUNCtion<number> | RESPonse<number> |
WMEMory<number> | <float_value>}
```

Defines a function that computes the maximum value of the operand waveform in each time bucket.

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

`<float_value>` A real number.

Example

The following example sets up function 1 to compute the maximum of each time bucket for channel 1:

```
10 OUTPUT 707;":FUNCTION1:MAXIMUM CHANNEL1"
20 END
```

MINimum

Command

```
:FUNCtion{1 | 2}:MINimum {CHANnel<number> |
FUNCtion<number> | RESPonse<number> |
WMEMory<number> | <float_value>}
```

Defines a function that computes the minimum of each time bucket for the defined operand's waveform.

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

`<float_value>` A real number.

Example

The following example sets up function 1 to compute the minimum of each time bucket for channel 1:

```
10 OUTPUT 707;":FUNCTION1:MINIMUM CHANNEL1"
20 END
```

MULTiply

Command

```
:FUNCtion{1 | 2}:MULTiply <source>[,<source>]
```

Defines a function that algebraically multiplies the first operand by the second operand.

`<source>`

```
{CHANnel<number> | FUNCtion<number> |
RESPonse<number> | WMEMory<number> |
<float_value>}
```

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

<float_value> A real number.

Example The following example defines a function that multiplies channel 1 by waveform memory 1:

```
10 OUTPUT 707;":FUNCTION1:MULTIPLY CHANNEL1,WMEMORY1"
20 END
```

SUBTract

Command

```
:FUNCtion{1 | 2}:SUBTract <source>[,<source>]
```

Defines a function that algebraically subtracts the second operand from the first operand.

<source> {CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number> | <float_value>}

<number> For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

<float_value> A real number.

Example The following example defines a function that subtracts waveform memory 1 from channel 1:

```
10 OUTPUT 707;":FUNCTION1:MULTIPLY CHANNEL1,WMEMORY1"
20 END
```

VERSus

Command `:FUNCtion{1 | 2}:VERSus <source>[,<source>]`

Defines a function for an X-versus-Y display. The first operand defines the Y axis and the second defines the X axis. The Y-axis range and offset are initially equal to that of the first operand and can be adjusted with the RANGE and OFFSET commands.

`<source>` `{CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number> | <float_value>}`

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

`<float_value>` A real number.

Example

The following example defines function 1 as an X-versus-Y display. Channel 1 is the X axis and waveform memory 2 is the Y axis:

```
10 OUTPUT 707;":FUNCTION1:VERSUS WMEMORY2,CHANNEL1"
20 END
```

9

Histogram Commands

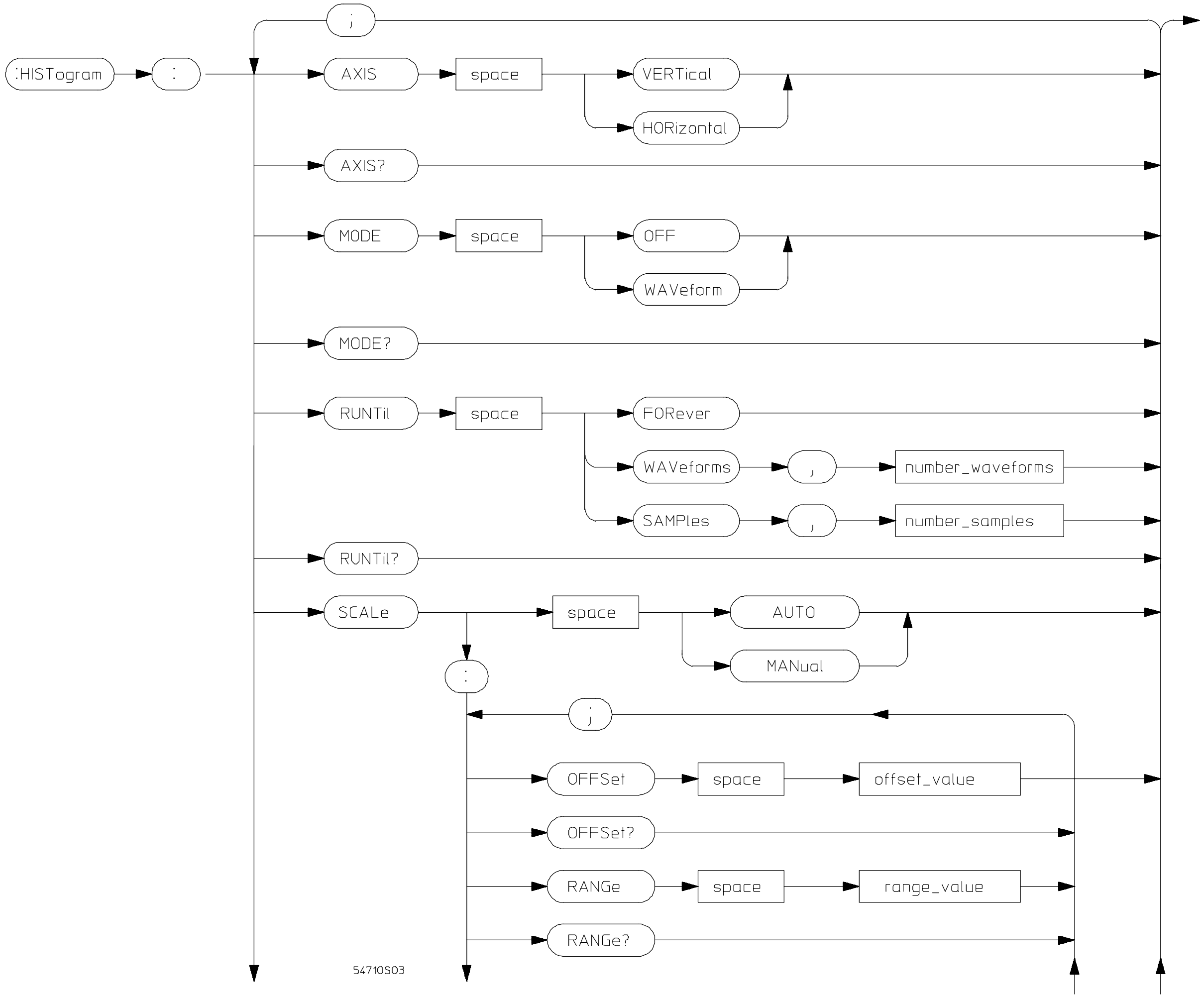
Histogram Commands

The HISTogram commands and queries control the histogram features of the oscilloscope. A histogram shows the probability distribution of acquired data within a user-defined histogram window. You can display the histogram vertically for voltage measurements or horizontally for timing measurements.

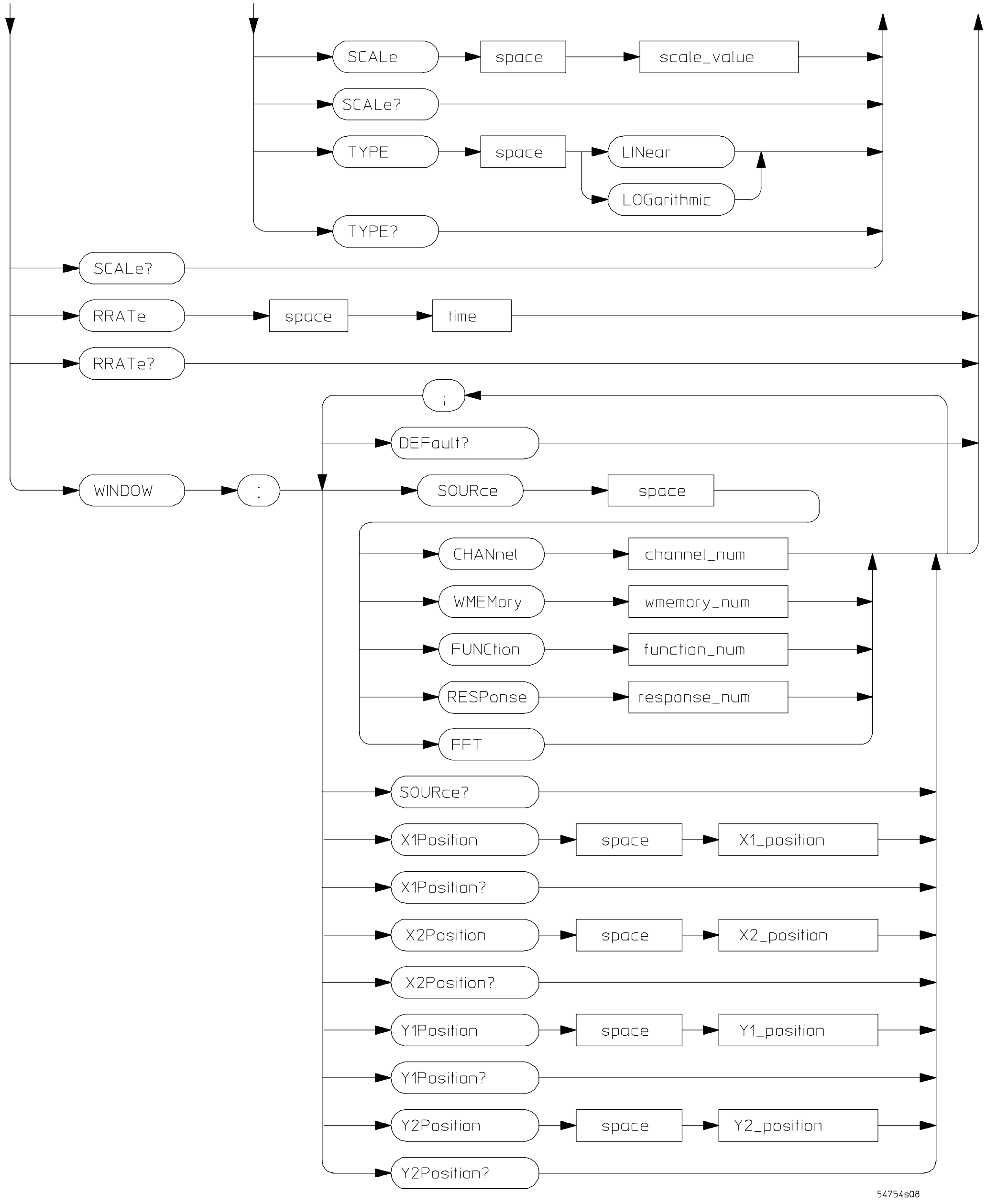
The most common use for histograms is measuring and characterizing noise or jitter on displayed waveforms. Noise is measured by sizing the histogram window to a narrow portion of time and observing a vertical histogram that shows the noise on the waveform. Jitter is measured by sizing the histogram window to a narrow portion of voltage and observing a horizontal histogram that shows the jitter on an edge.

The TDR plug-in modifies the operation of the following HISTogram subsystem command:

- WINDow:SOURce



Histogram Subsystem Command Syntax Diagram



Histogram Subsystem Command Syntax Diagram (continued)

WINDow:SOURce

Command

```
:HISTogram:WINDow:SOURce {CHANnel<number> |
FUNCtion<number> | RESPonse<number> |
WMEMory<number> | FFT}
```

Selects the source of the histogram window. The histogram window will track the source's vertical and horizontal scale.

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR response: 1, 2, 3, or 4.

Example

The following example sets the histogram window's source to TDR response 1:

```
10 OUTPUT 707;":HISTOGRAM:WINDOW:SOURCE RESPONSE1"
20 END
```

Query

```
:HISTogram:WINDow:SOURce?
```

The query returns the currently selected histogram window source.

Returned Format

```
[:HISTogram:WINDow:SOURce]{CHANnel<number> | FUNCtion<number>
| RESPonse<number> | WMEMory<number> | FFT}<NL>
```

Example

The following example returns the result of the window source query and prints it to the controller's screen.

```
10 DIM Winsour$[50]     !Dimension variable
20 OUTPUT 707;":HISTOGRAM:WINDOW:SOURCE?"
30 ENTER 707;Winsour$
40 PRINT Winsour$
50 END
```

10

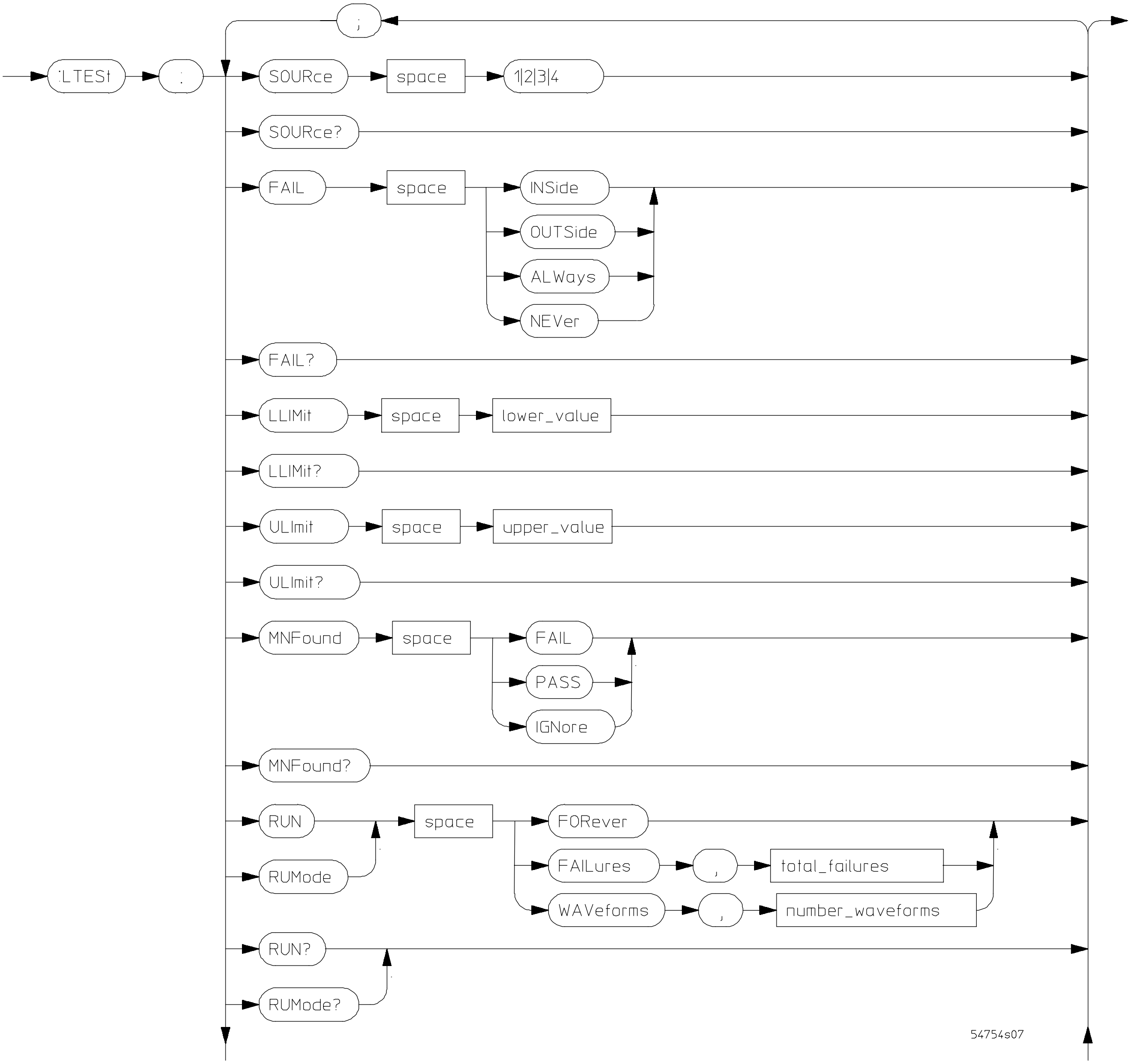
Limit Test Commands

Limit Test Commands

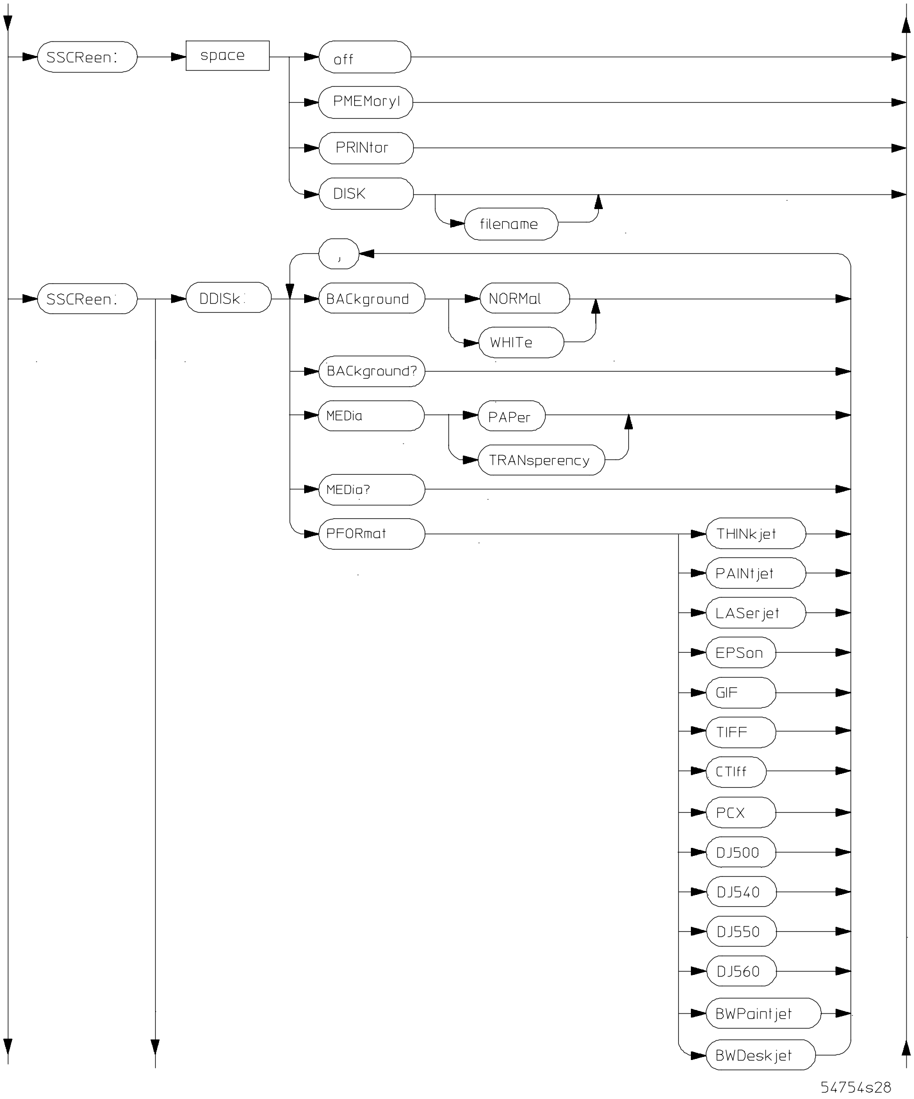
The Limit Test commands and queries control the limit test features of the oscilloscope. Limit testing automatically compares measurement results with pass or fail limits. The limit test tracks up to four measurements. This command subsystem also controls the actions that are taken when a test fails.

The TDR plug-in modifies the operation of the following Limit Test subsystem command:

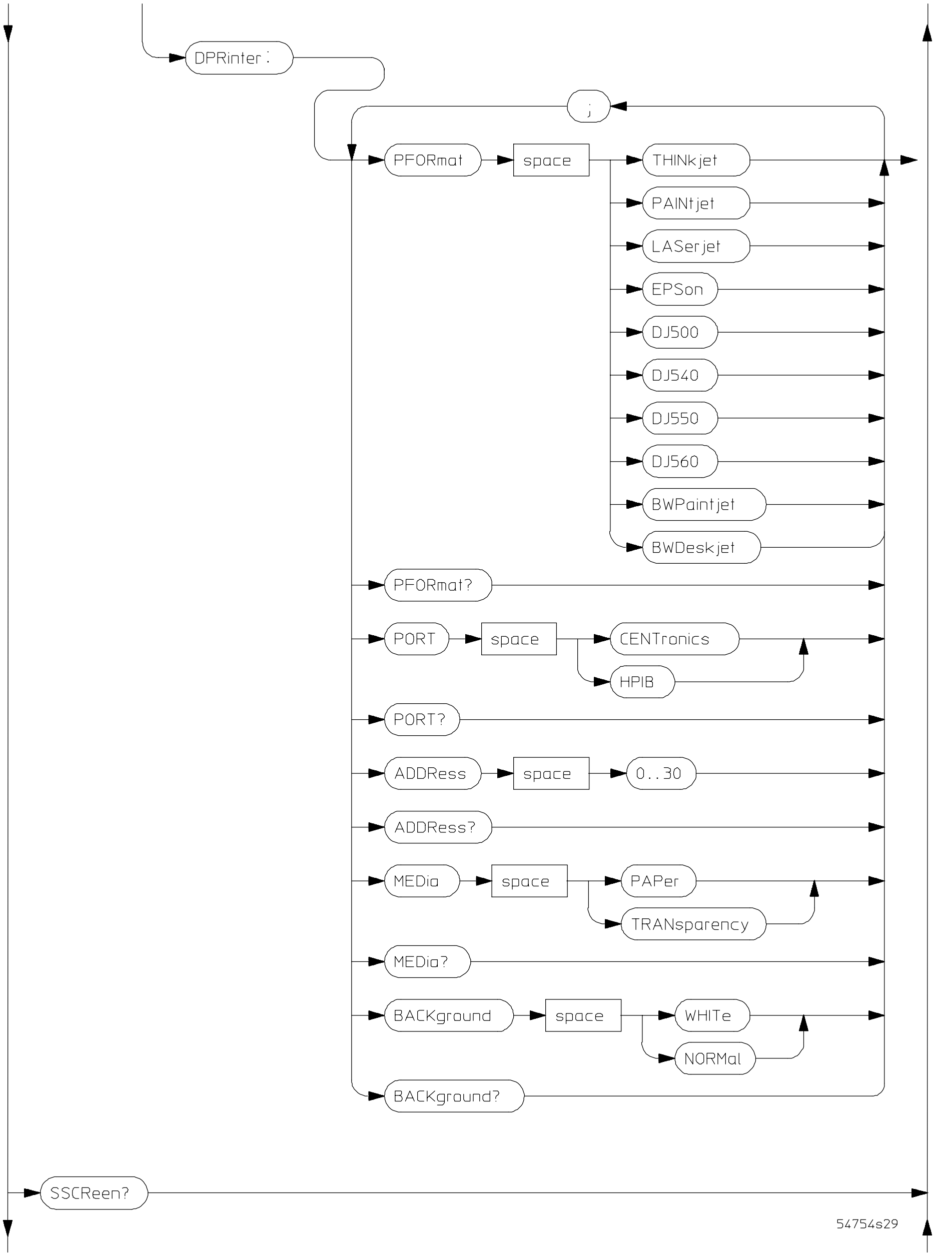
- SWAVeform



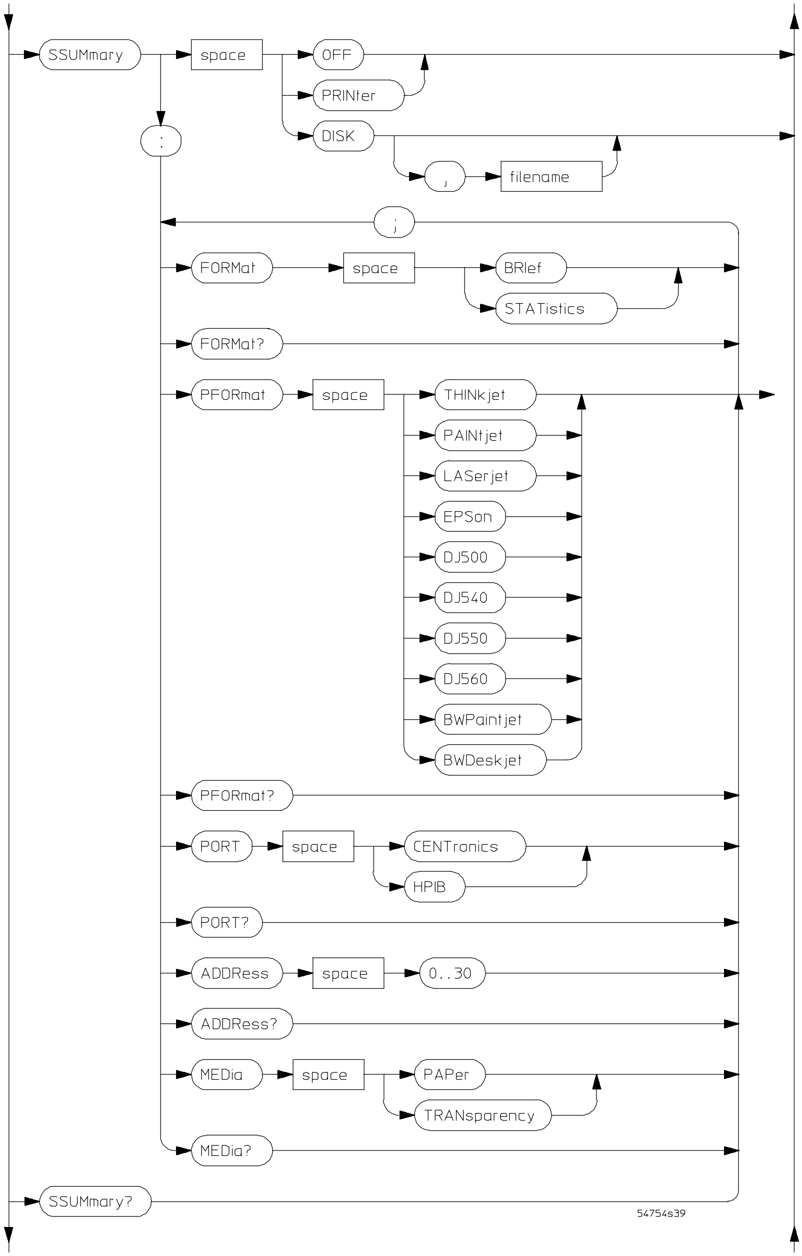
Limit Test Subsystem Command Syntax Diagram



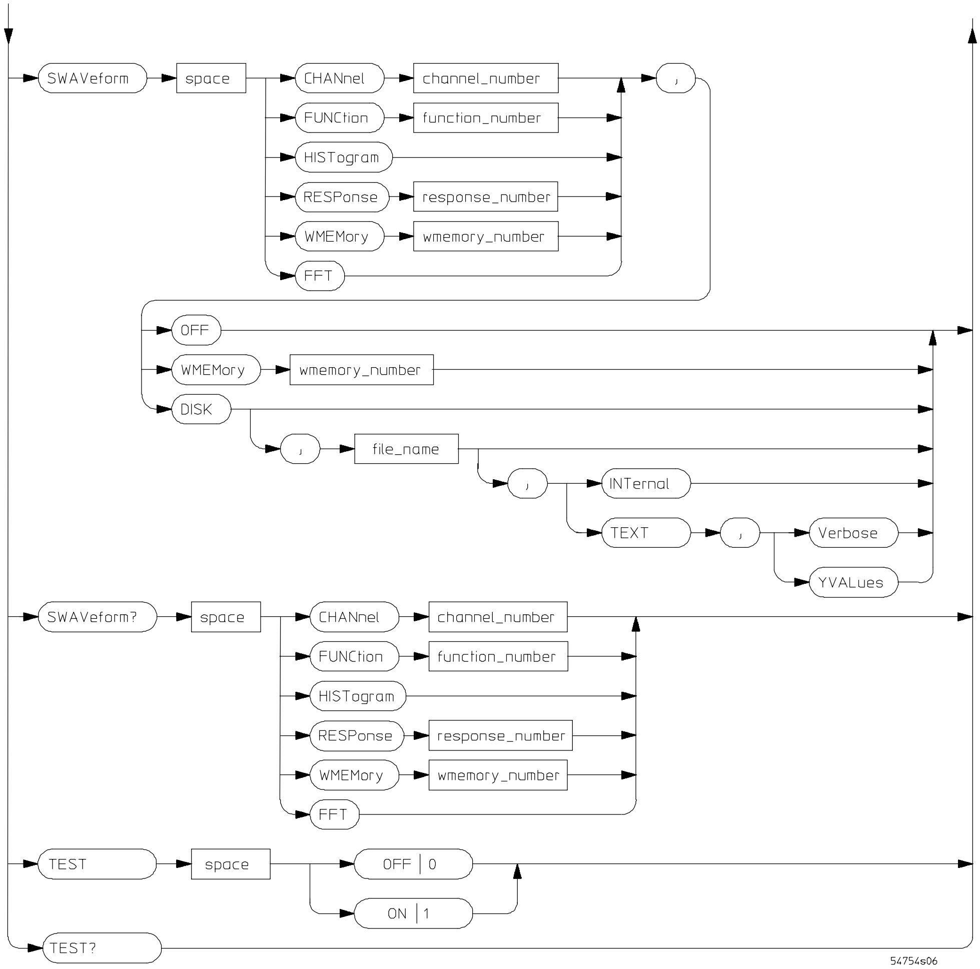
Limit Test Subsystem Command Syntax Diagram (continued)



Limit Test Subsystem Command Syntax Diagram (continued)



Limit Test Subsystem Command Syntax Diagram (continued)



Limit Test Subsystem Command Syntax Diagram (continued)

SWAVeform

Command

```
:LTESt:SWAVeform
<source>,<destination>,[,<filename>[,<format>]]
```

Saves waveforms from a channel, function, histogram, TDR response, or waveform memory in the event of a failure detected by the limit test. Each waveform source can be individually specified, allowing multiple channels or functions to be saved to disk or waveform memories. Setting a particular source to OFF removes any waveform save action from that source.

`<source>` {CHANnel<number> | FUNCtion<number> | HISTogram | RESPonse<number> | WMEMory<number>}

`<destination>` {OFF | WMEMory<number> | DISK}

When DISK is selected, a file name is required.

`<filename>` A descriptive file prefix consisting of up to four characters. If no filename is specified, the prefix defaults to CH1A..CH4B, FN1, FN2, FFT, HIST, MEM1..MEM4, RSP1..RSP4.

`<format>` {TEXT [,YVALues | VERBose] | INTernal}

Example

The following example turns off saving of waveforms from TDR response 1.

```
10 OUTPUT 707;":LTEST:SWAVEFORM RESP1,OFF"
20 END
```

Query

```
:LTESt:SWAVeform? <source>
```

The query returns the current save waveform configuration.

Returned Format

```
:LTESt:SWAVeform <source>,<destination>,[,<filename>[,<format>]]
```

Example

The following example retrieves the information about where TDR response 1 will be saved, then displays it on the controller screen:

```
10 DIM SWAV$[50]
20 OUTPUT 707;":LTEST:SWAVEFORM? RESPONSE1"
30 ENTER 707;SWAV$
40 PRINT SWAV$
50 END
```

11

Marker Commands

Marker Commands

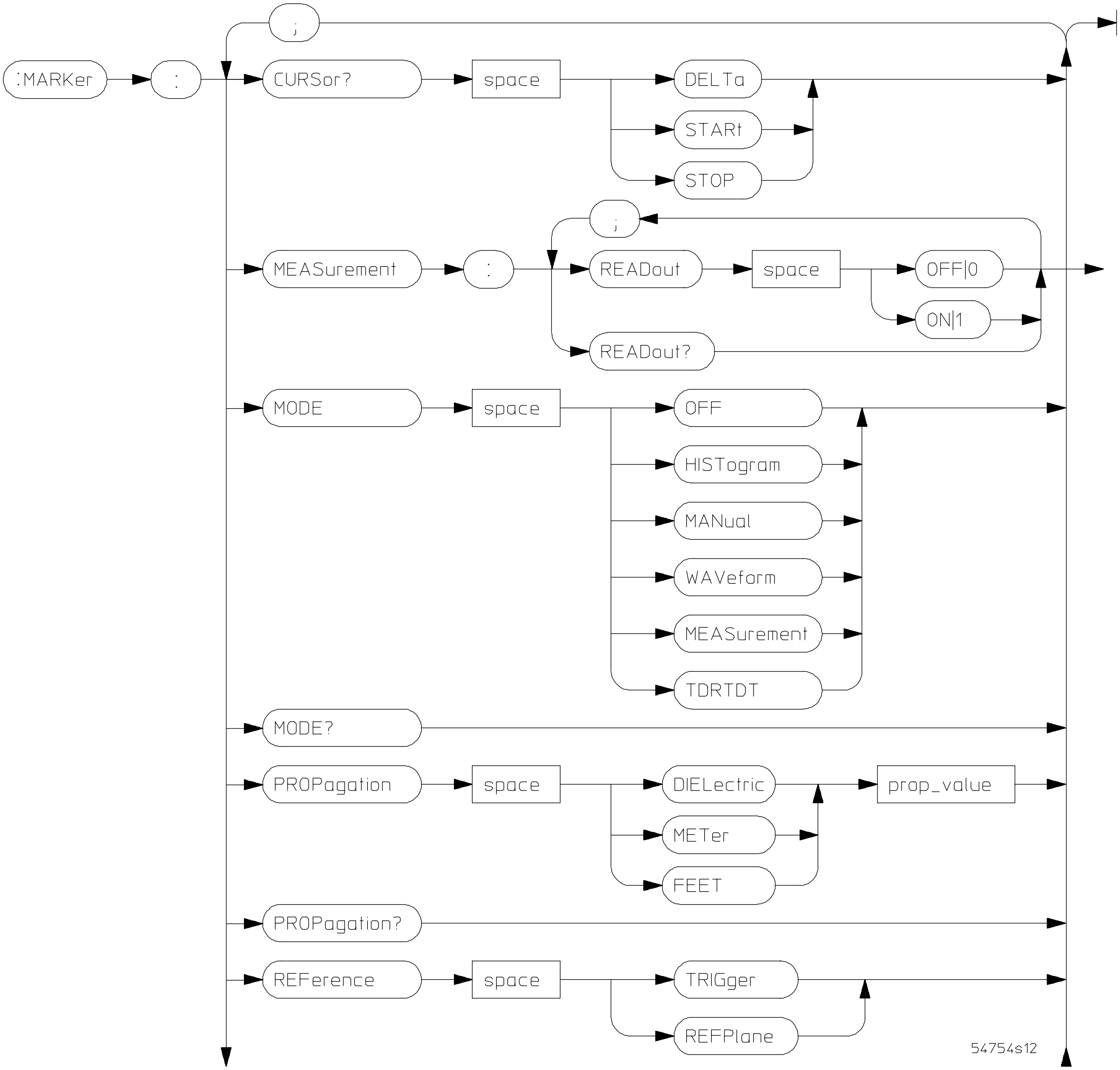
The commands in the MARKER subsystem are used to specify and query the settings of the time and distance (X-axis) and the current measurement units (Y-axis) markers. The X-axis measurement units are typically set using the CHANnel:XUNITs command. The Y-axis measurement units are typically set using the CHANnel:YUNITs command.

The TDR plug-in adds the following commands to the Marker subsystem:

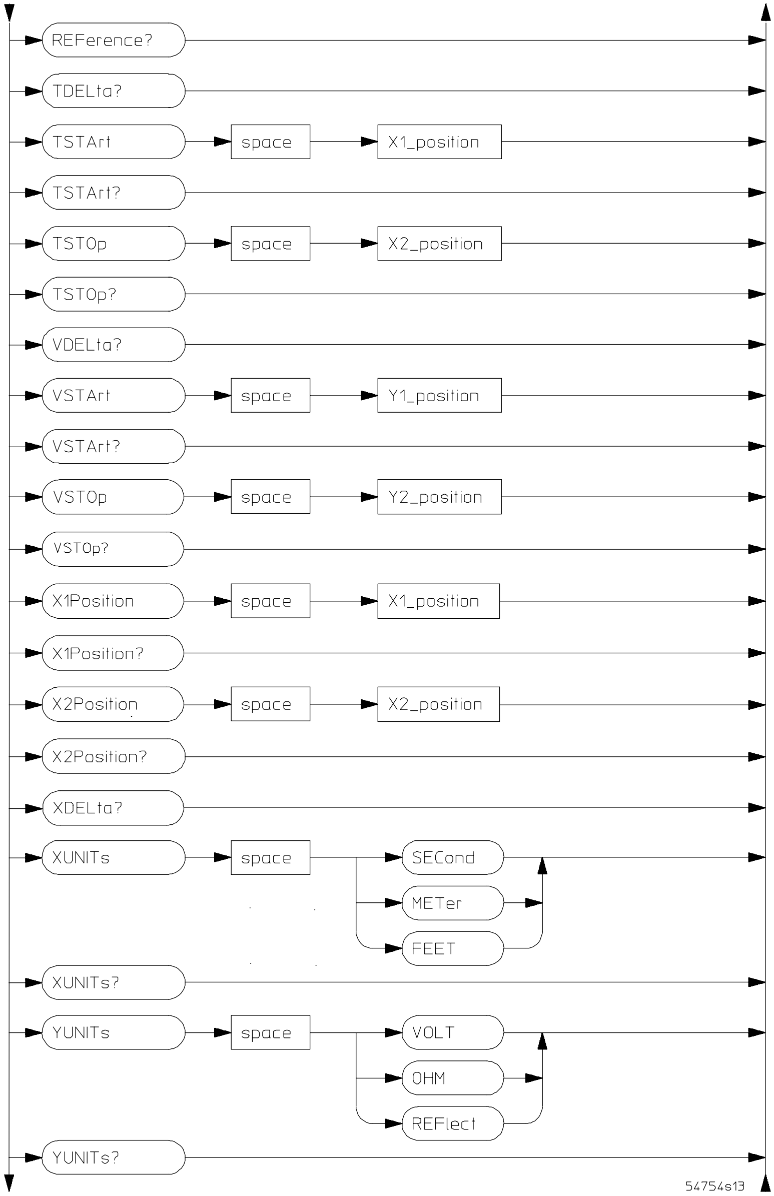
- PROPagation
- REFerence
- XUNITs
- YUNITs

The TDR plug-in modifies the operation of the following Marker subsystem commands:

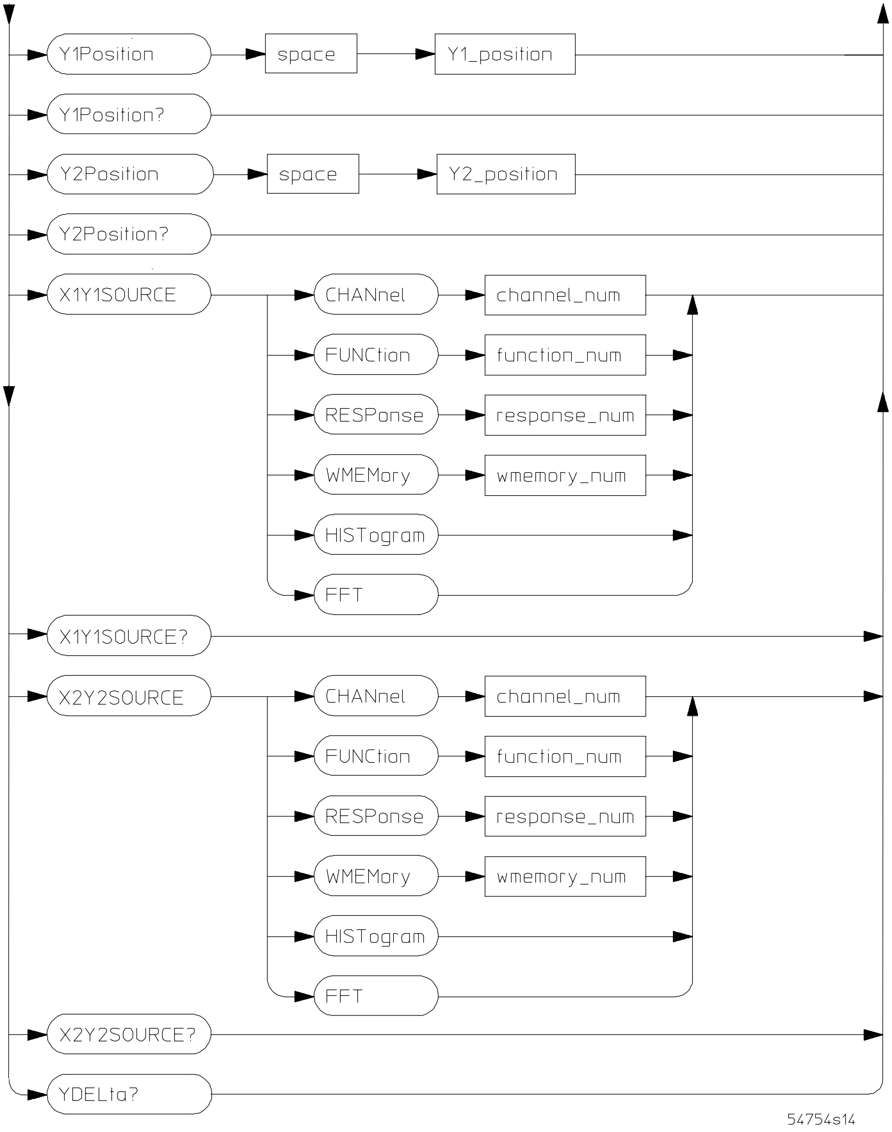
- MODE
- X1Y1source
- X2Y2source



Marker Subsystem Command Syntax Diagram



Marker Subsystem Command Syntax Diagram (continued)



Marker Subsystem Command Syntax Diagram (continued)

MODE

Command

```
:MARKer:MODE {OFF | MANual | WAVeform |
MEASurement | HISTogram | TDRTDT}
```

Sets the marker mode.

Example

The following example sets the marker mode to TDR/TDT.

```
10 OUTPUT 707;":MARKER:MODE TDRTDT"
20 END
```

Query

```
:MARKer:MODE?
```

The query returns the current marker mode.

Returned Format

```
[:MARKer:MODE]{OFF | MANual | WAVeform | MEASurement | HISTogram
| TDRTDT}<NL>
```

Example

The following example puts the current marker mode into the string variable Selection$, then prints the contents of the variable to the controller's screen.

```
10 DIM Selection$[50]          !Dimension variable
20 OUTPUT 707;":MARKER:MODE?"
30 ENTER 707;Section$
40 PRINT Selection$
50 END
```

PROPagation

Command

```
:MARKer:PROPagation {DIELectric | METer | FEET},
<propagation>
```

Sets the propagation velocity for TDR and TDT measurements. The propagation may be specified as a dielectric constant or in meters per second or feet per second. The value is used to determine the distance from the reference plane in TDR and TDT marker measurements.

> NOTE
>
> To ensure accurate marker measurements, you must ensure that the propagation value is accurate, that the units are set correctly (:MARKer:XUNITs), and that the correct reference plane is selected (:MARKer:REFerence).

`<propagation>` Dielectric constant or propagation value. You must specify one of the modifiers DIELectric, METer, or FEET.

Example

The following example sets the propagation to 100,000 feet per second.

```
10 OUTPUT 707;":MARKER:PROPAGATION FEET, 1E5"
20 END
```

Query

```
:MARKer:PROPagation?
```

The query returns the current propagation value.

Returned Format

```
[:MARKer:PROPagation]<propagation> {DIELectric | METer |
FEET}<NL>
```

Example

The following example gets the propagation value from the instrument, puts it into the variable Prop$, then displays the contents of the variable on the controller’s screen.

```
10 DIM Prop$[20]          !Declare variable
20 OUTPUT 707;":MARKER:PROPAGATION?"
30 ENTER 707;Prop$
40 PRINT Prop$
50 END
```

REFerence

Command

```
:MARKer:REFerence {TRIGger | REFPlane}
```

Specifies the marker reference for TDR and TDT style markers. If the reference is TRIGger, then all horizontal axis marker measurements are made with respect to the trigger point. If the reference is REFPlane, then all horizontal axis marker measurements are made with respect to the reference plane. You must perform a normalization and reference plane calibration before using a reference plane reference. This feature is available only for TDR and TDT applications.

Example

The following example sets the markers to indicate all horizontal axis measurements with respect to the trigger:

```
10 OUTPUT 707;":MARKER:REFERENCE TRIGGER"
20 END
```

Query

```
:MARKer:REFerence?
```

Returns the current reference setting.

Returned Format

```
[:MARKer:REFerence]{TRIGger | REFPlane}<NL>
```

Example

The following example puts the current reference setting into the variable Ref$, then displays the contents of the variable on the controller's screen.

```
10 DIM Ref$[20]                      !Declare variable
20 OUTPUT 707;":MARKER:REFERENCE?"
30 ENTER 707;Ref$
40 PRINT Ref$
50 END
```

X1Y1source

Command

```
:MARKer:X1Y1source {CHANnel<number> |
FUNCtion<number> | RESPonse<number> |
WMEMory<number> | HISTogram | FFT}
```

Sets the source for the X1 and Y1 markers.

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B indicating which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR response: 1, 2, 3, or 4.

Example

The following example selects TDR response 1 as the X1Y1 source.

```
10 OUTPUT 707;":MARKER:X1Y1SOURCE RESPONSE1"
20 END
```

Query

```
:MARKer:X1Y1source?
```

The MARKer:X1Y1source query returns the current X1Y1 source.

Returned Format

```
[:MARKer:X1Y1source] {CHANnel<number> | FUNCTion<number>  |
RESPonse<number> | WMEMory<number> | HISTogram | FFT}
```

Example

The following example returns the current X1Y1 source selection to the string variable Selection$, then prints the contents of the variable to the controller’s screen.

```
10 DIM Selection$[50]      !Dimension variable
20 OUTPUT 707;":MARKER:X1Y1SOURCE?"
30 ENTER 707;Selection$
40 PRINT Selection$
50 END
```

X2Y2source

Command

```
:MARKer:X2Y2source {CHANnel<number> |
FUNCtion<number> | RESPonse<number> |
WMEMory<number> | HISTogram | FFT}
```

Sets the source for the X2 and Y2 markers.

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B indicating which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR response: 1, 2, 3, or 4.

Example

The following example selects TDR response 1 as the X2Y2 source.

```
10 OUTPUT 707;":MARKER:X2Y2SOURCE RESPONSE1"
20 END
```

Query

```
:MARKer:X2Y2source?
```

The MARKer:X2Y2source query returns the current X2Y2 source.

Returned Format

```
[:MARKer:X2Y2source] {CHANnel<number> | FUNCTion<number> |
RESPonse<number> | WMEMory<number> | HISTogram | FFT}
```

Example

The following example returns the current X2Y2 source selection to the string variable Selection$, then prints the contents of the variable to the controller’s screen.

```
10 DIM Selection$[50]      !Dimension variable
20 OUTPUT 707;":MARKER:X2Y2SOURCE?"
30 ENTER 707;Selection$
40 PRINT Selection$
50 END
```

XUNITs

Command

```
:MARKer:XUNITs {SECond | METer | FEET}
```

Sets the units for horizontal display in TDR and TDT applications. The units may be in seconds, meters, or feet relative to the trigger or reference plane. The marker mode must be TDRTDT to use this feature. See the :MARKer:REFerence command for information on setting the reference point.

Example

The following example sets the horizontal display units to meters:

```
10 OUTPUT 707;":MARKER:XUNITS METER"
20 END
```

Query

```
:MARKer:XUNITs?
```

Returns the current marker units setting.

Returned Format

```
[:MARKer:XUNITs]{SECond | METer | FEET}<NL>
```

Example

The following example puts the current marker units setting into the variable Units$, then displays the contents of that variable on the controller’s screen.

```
10 DIM Units$[20]
20 OUTPUT 707;":MARKER:XUNITS?"
30 ENTER 707;Units$
40 PRINT Units$
50 END
```

YUNITs

Command

```
:MARKer:YUNITs {VOLT | OHM | REFlect}
```

Sets the units for vertical display in TDR and TDT applications. The units may be in volts, ohms, or % reflection. The marker mode must be TDRTDT to use this feature.

Example

The following example sets the vertical display units to ohms:

```
10 OUTPUT 707;":MARKER:YUNITS OHM"
20 END
```

Query

```
:MARKer:YUNITs?
```

Returns the current marker units setting.

Returned Format

```
[:MARKer:YUNITs]{VOLT | OHM | REFlect}<NL>
```

Example

The following example puts the current marker units setting into the variable Units$, then displays the contents of that variable on the controller's screen.

```
10 DIM Units$[20]
20 OUTPUT 707;":MARKER:YUNITS?"
30 ENTER 707;Units$
40 PRINT Units$
50 END
```

12

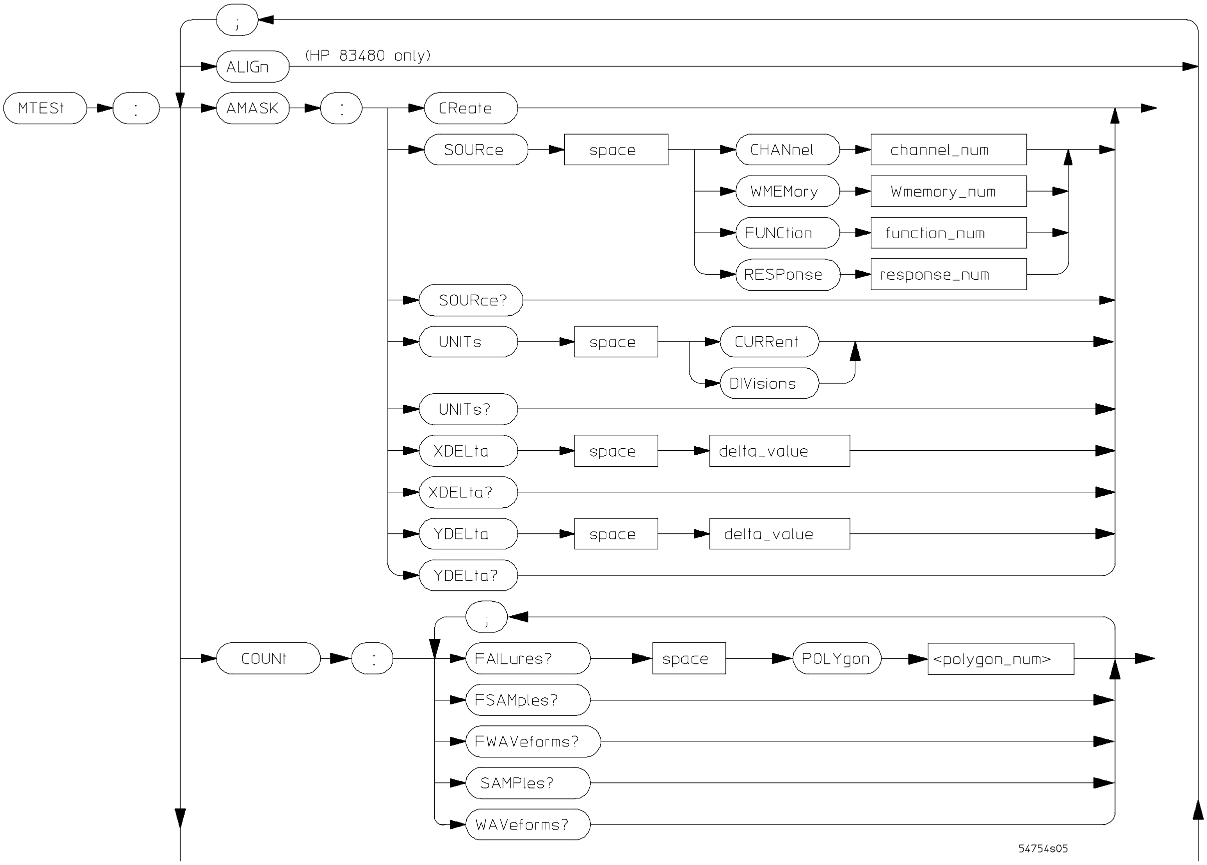
Mask Test Commands

Mask Test Commands

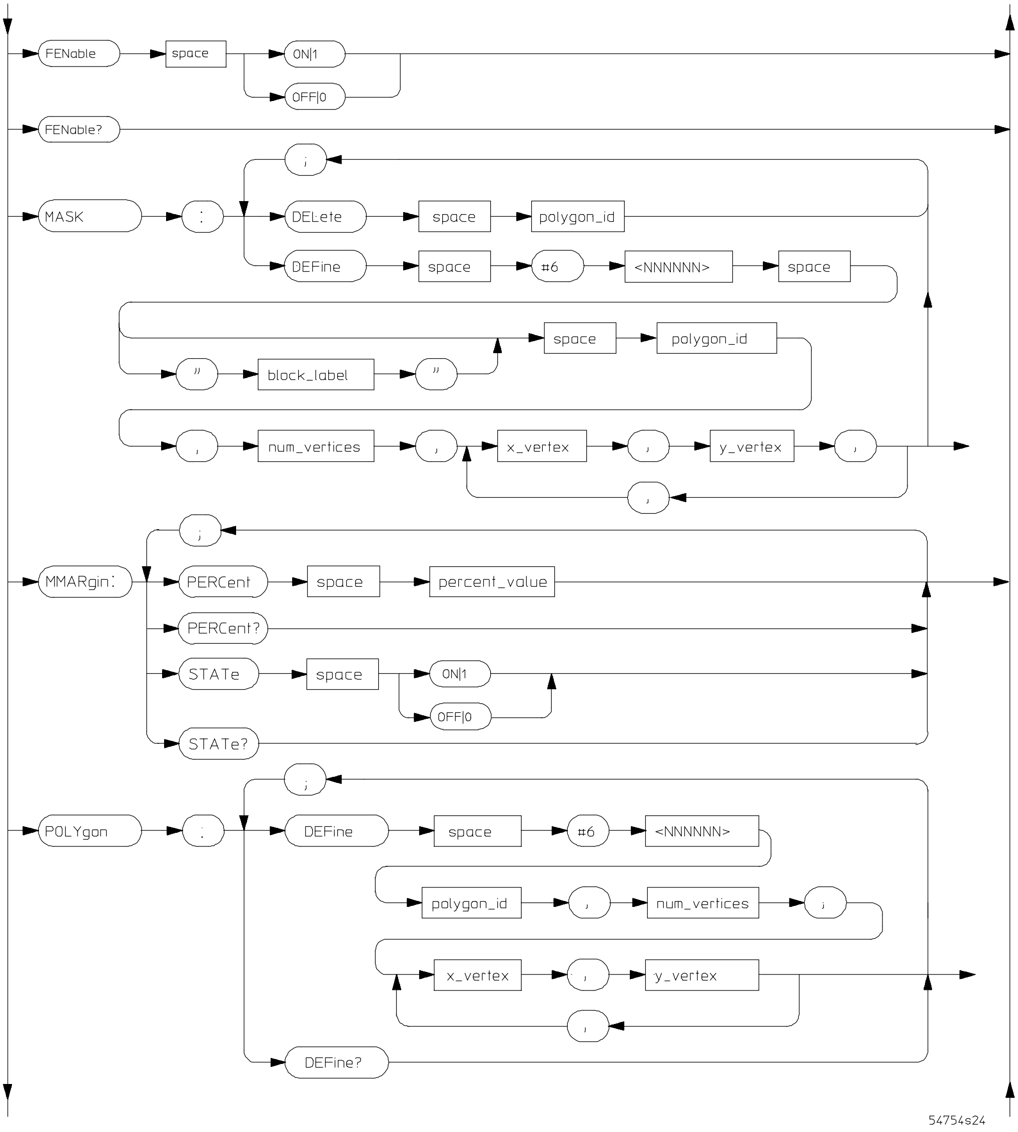
The Mask Test commands and queries control the mask test features. Mask testing automatically compares measurement results with the boundaries of a set of polygons that you define. Any waveform or sample that falls within the boundaries of one or more polygons is recorded as a failure.

The TDR plug-in modifies operation of the following Mask Test subsystem commands:

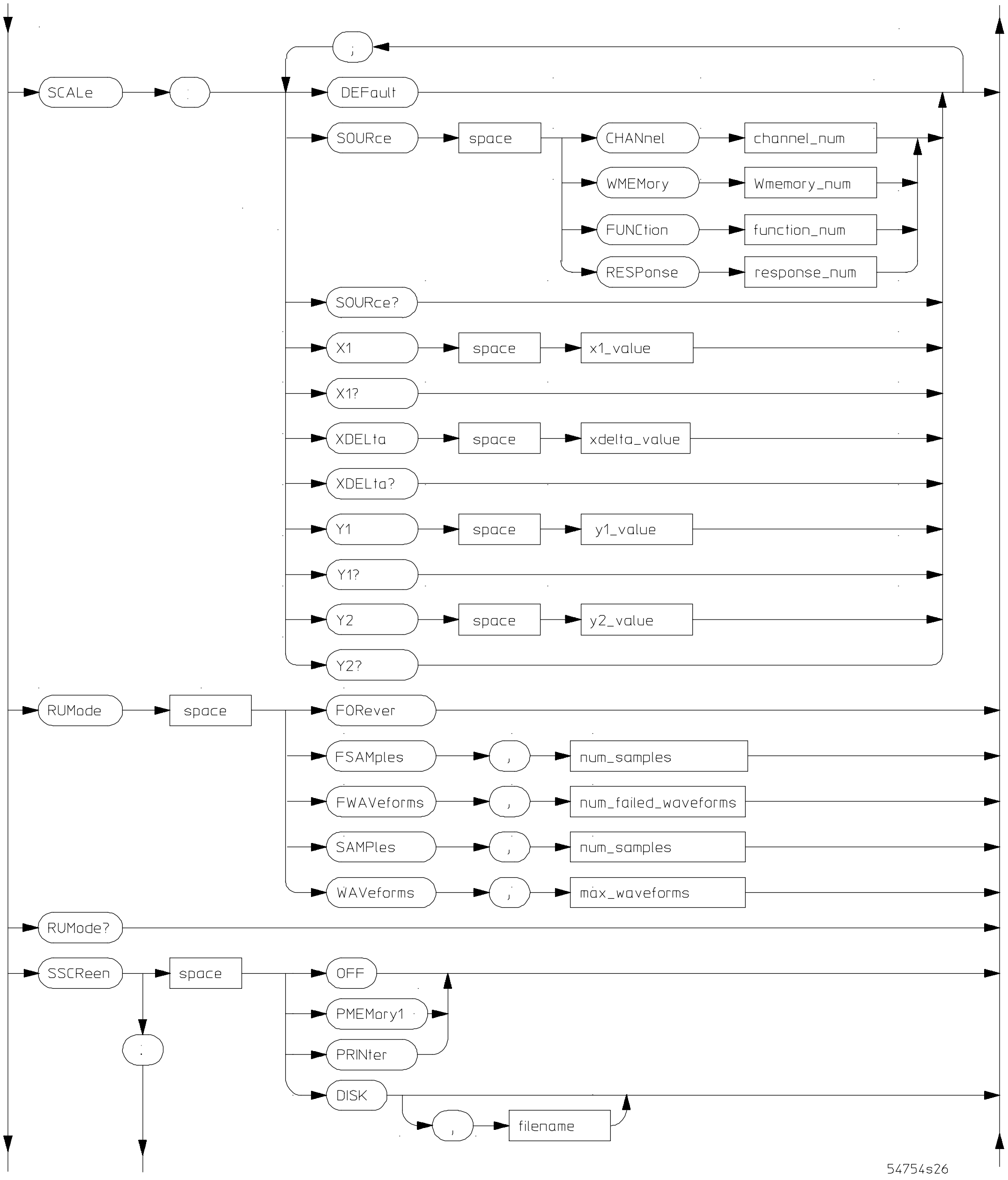
- AMASk:SOURce
- SCALe:SOURce
- SWAVeform



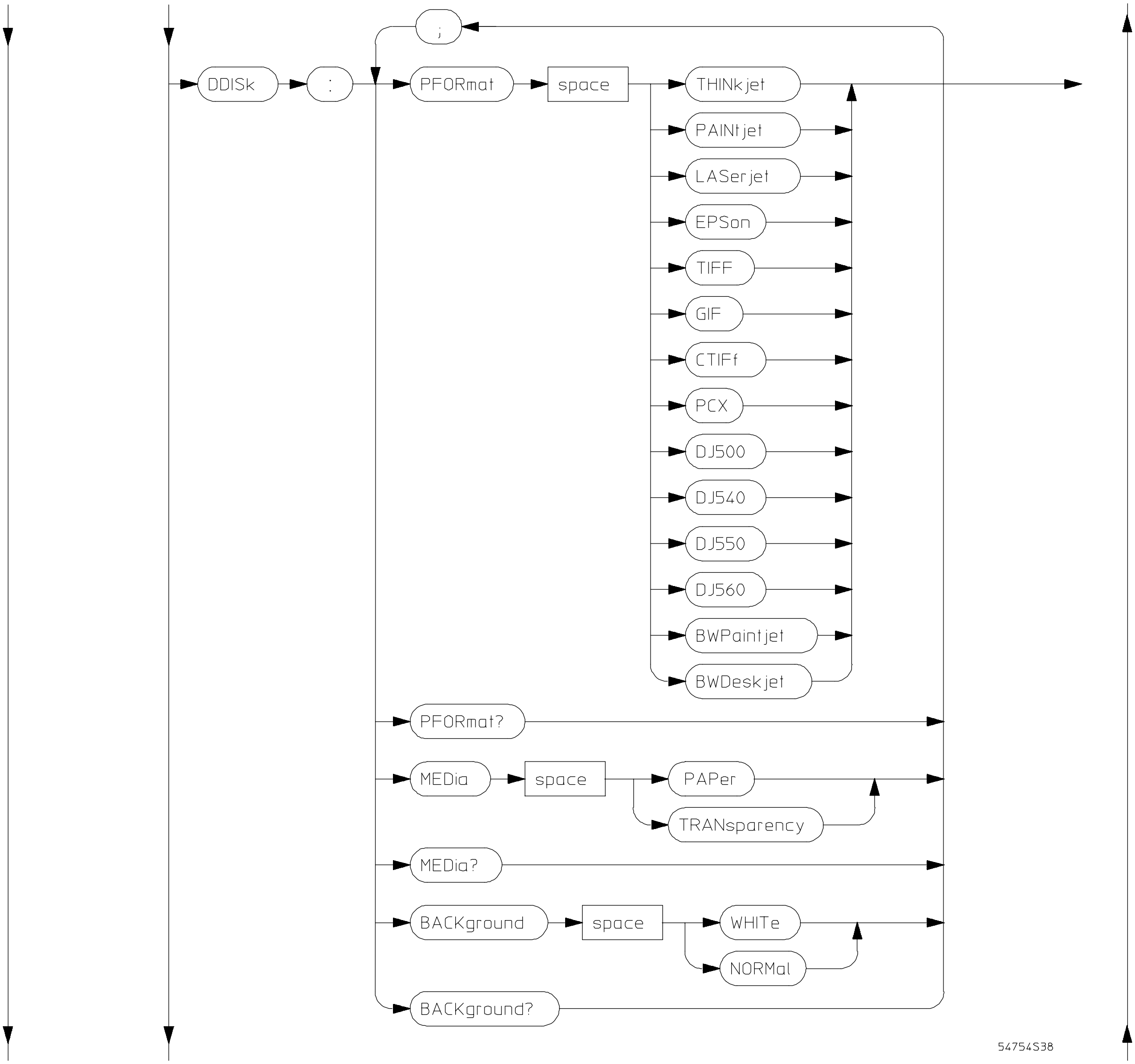
Mask Test Subsystem Command Syntax Diagram



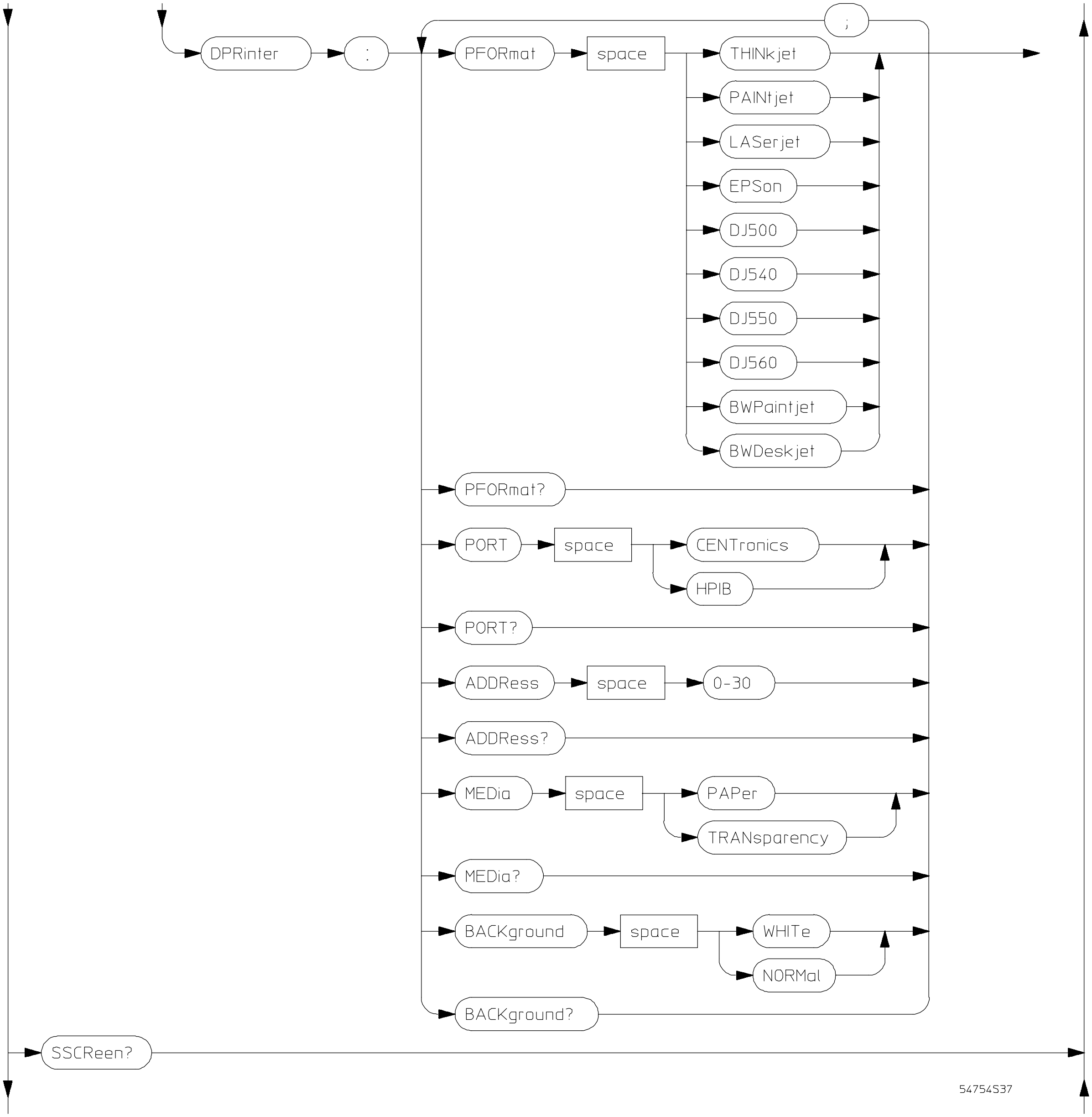
Mask Test Subsystem Command Syntax Diagram (continued)



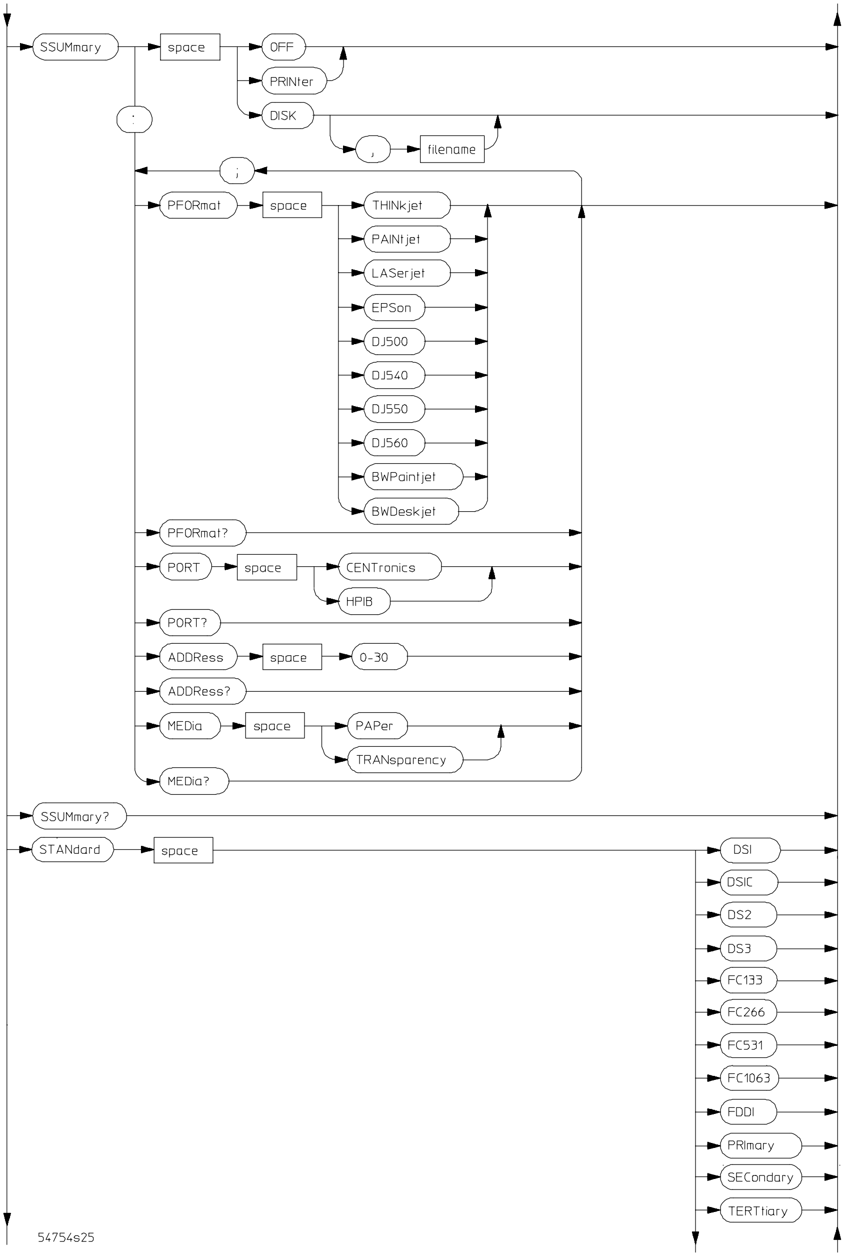
Mask Test Subsystem Command Syntax Diagram (continued)



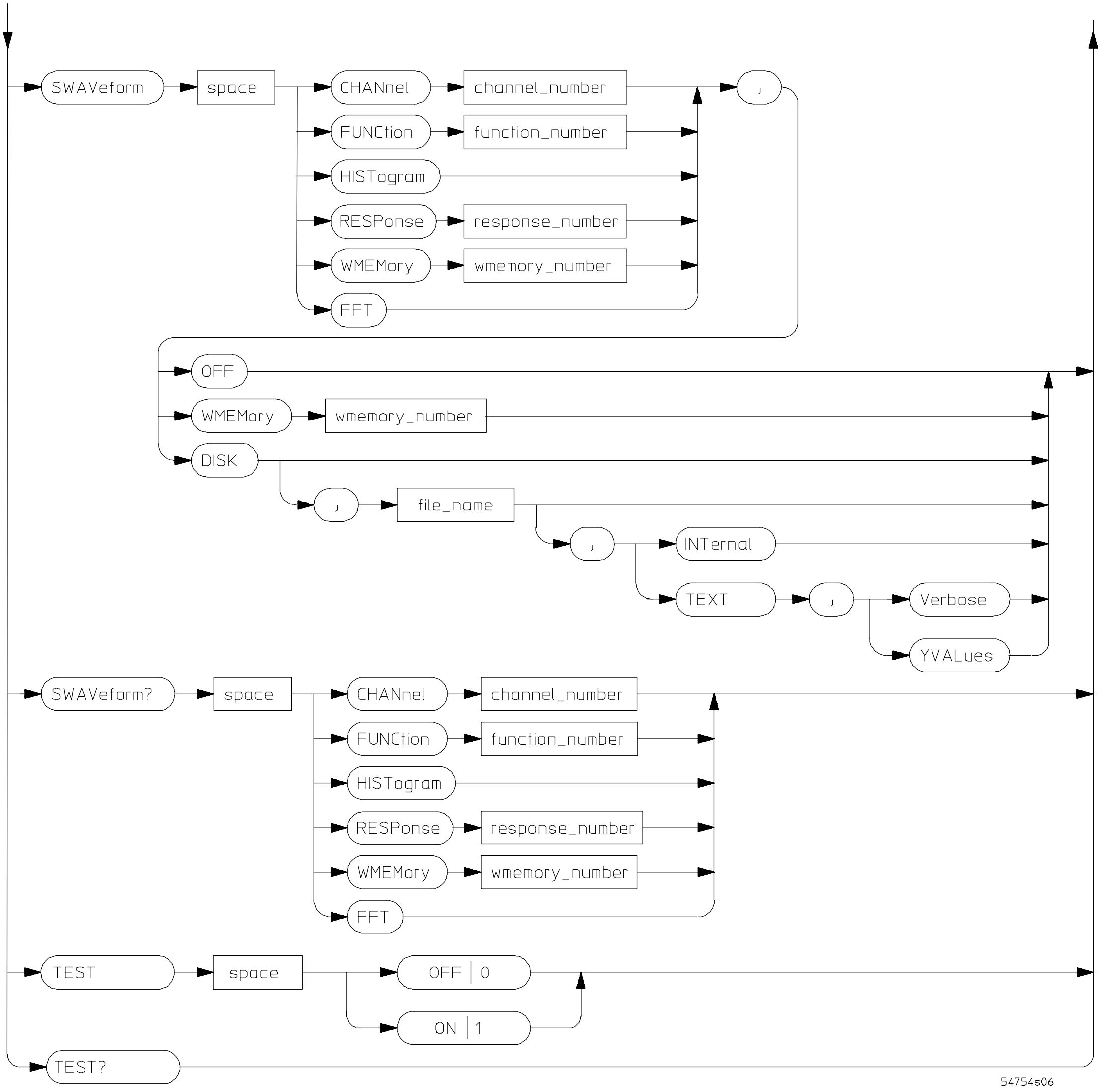
Mask Test Subsystem Command Syntax Diagram (continued)



Mask Test Subsystem Command Syntax Diagram (continued)



Mask Test Subsystem Command Syntax Diagram (continued)



Mask Test Subsystem Command Syntax Diagram (continued)

AMASk:SOURce

Command

```
:MTESt:AMASk:SOURce {CHANnel<number> |
FUNCtion<number> | RESPonse<number> |
WMEMory<number> | FFT}
```

Selects the source for the interpretation of the AMASk:XDELta and AMASk:YDELta parameters when AMASk:UNITs is set to CURRent. When UNITs are CURRent, the XDELta and YDELta parameters are defined in terms of the measurement system of the selected source. Suppose that UNITs are CURRent and that you set SOURce to CHANNEL1, which is using volts as a measurement system. Then you can define AMASk:XDELta in terms of volts and AMASk:YDELta in terms of seconds.

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For waveform memories: 1, 2, 3, or 4. For functions: 1 or 2. For TDR responses: 1, 2, 3, or 4.

Example

The following program sets the automask measurement system source to TDR response 1.

```
10 OUTPUT 707;":MTEST:AMASK:SOURCE RESPONSE1"
20 END
```

Query

```
:MTESt:AMASk:SOURce?
```

The query returns the currently set source.

Returned Format

```
[:MTESt:AMASk:SOURce] {CHANnel<number> | FUNCtion<number> |
RESPonse<number> | WMEMory<number> | FFT}<NL>
```

Example

The following program gets the source setting for automask and prints the result on the controller display.

```
10 DIM AMASK_SOURCE$[30]
20 OUTPUT 707;":MTEST:AMASK:SOURCE?"
30 ENTER 707;AMASK_SOURCE$
40 PRINT AMASK_SOURCE$
50 END
```

SWAVeform

Command

```
:MTESt:SWAVeform
<source>,<destination>[,<filename>[,<format>]]
```

Saves waveforms from a channel, function, TDR response, histogram, or waveform memory in the event of a failure detected by the Mask Test. Each waveform source can be individually specified, allowing multiple channels or functions to be saved to disk or waveform memories. Setting a particular source to OFF removes any waveform save action from that source.

<source> {CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number> | HISTogram | FFT}

<number> For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

<destination> {OFF | WMEMory<number> | DISK}

<filename> A descriptive file prefix consisting of up to four characters. If no filename is specified, the prefix defaults to CH1A..CH4B, FN1, FN2, FFT, HIST, MEM1..MEM4, RSP1..RSP4. /

<format> {TEXT [,YVALues | VERBose] | INTernal}

Example

The following example saves the waveform from TDR response 1 to waveform memory 1 when a failure occurs.

```
10 OUTPUT 707;":MTEST:SWAVEFORM RESPONSE1,WMEMORY1"
20 END
```

Query

```
:MTESt:SWAVeform? <source>
```

The query returns the current state of the :MTESt:SWAVeform command.

Returned Format

```
[:MTESt:SWAVeform]<source>,<destination>
[,<filename>[,<format>]]<NL>
```

Example

The following example gets the current save waveform configuration for channel 1 and prints it on the controller display.

```
10 DIM Save_wave$[200]         !dimension waveform
20 OUTPUT 707;":MTEST:SWAVEFORM? CHANNEL1"
30 ENTER 707;Save_wave$
40 PRINT Save_wave$
50 END
```

SCALe:SOURce

Command

```
:MTESt:SCALe:SOURce {CHANnel<number> |
FUNCtion<number> | RESPonse<number> |
WMEMory<number>}
```

Sets the source used by the mask subsystem for interpretation of the SCALe:Y1 and SCALe:Y2 parameters. SCALe:Y1 and SCALe:Y2 set the vertical boundaries of the coordinate system for mask testing, and are affected by the scaling of the selected source. For example, suppose that Y1 was set to 1 V and Y2 was set to +1 V. If Channel 1 was selected as the scaling source, and was set to a vertical scale factor of 100 mV per division, the Y1 and Y2 markers will be above and below the graticule, respectively. If Channel 2 was selected as the scaling source, and was set to a vertical scale factor of 500 mV per division, the Y1 and Y2 markers will be two divisions below and above the center of the graticule, respectively.

Interpretation of the X1 and ΔX settings is done using the time/div setting in the time base subsystem, if the source is a channel. The setting can be queried using the TIMebase:SCALe? command. Functions and waveform memories can have independent scale.

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

Example

The following example selects TDR response 1 as the source for interpretation of the Y1 and Y2 scaling values.

```
10 OUTPUT 707;":MTEST:SCALE:SOURCE RESPONSE1"
20 END
```

Query

```
:MTESt:SCALe:SOURce? <source>
```

The SCALe:SOURce? query returns the name of the source currently used to interpret the Y1 and Y2 scale factors.

Returned Format

```
[:MTESt:SCALe:SOURce]{CHANnel<number> | FUNCtion<number> |
RESPonse<number> | WMEMory<number>}<NL>
```

Example

The following example gets the current scale source setting from the oscilloscope and prints it on the controller display.

```
10 DIM Scale_Source$[30]         !dimension variable
20 OUTPUT 707;":MTEST:SCALE:SOURCE? CHANNEL1"
30 ENTER 707;Scale_Source$
40 PRINT Scale_Source$
50 END
```

13

Measure Commands

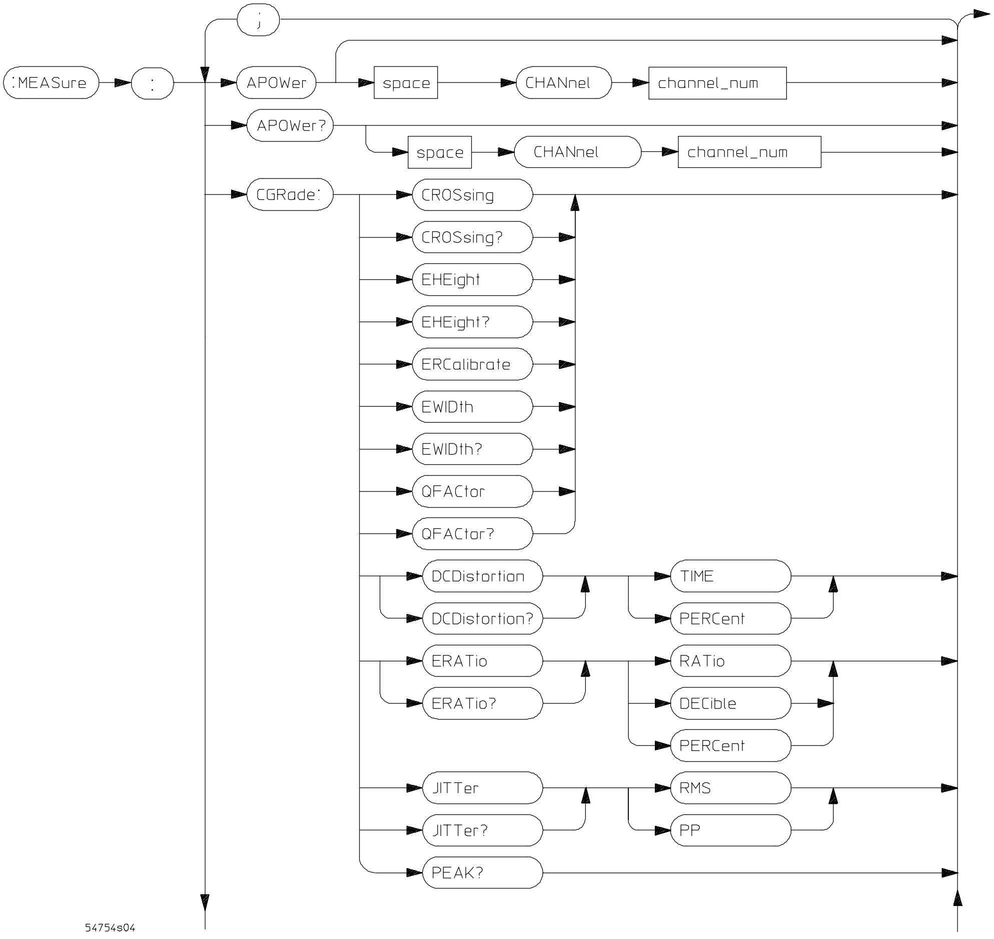
Measure Commands

The commands in the MEASure subsystem are used to make parametric measurements on displayed waveforms.

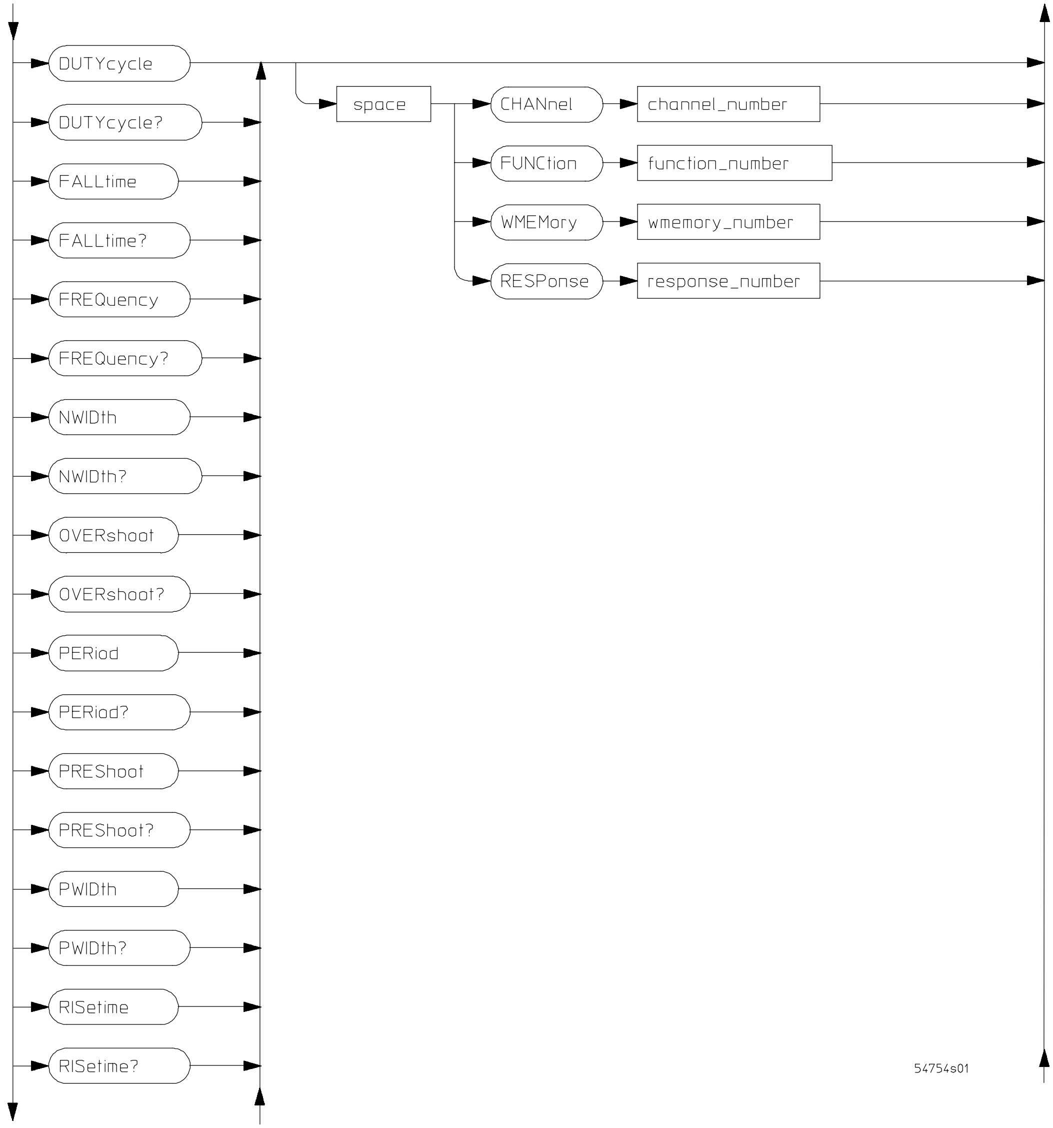
The TDR plug-in modifies operation of the following measure subsystem commands:

- DELTatime
- DUTycycle
- FALLtime
- FREQuency
- NWIDth
- OVERshoot
- PERiod
- PREShoot
- PWIDth
- RISetime
- SOURce
- TEDge
- TMAX
- TMIN
- TVOLt
- VAMPlitude
- VAVerage
- VBASe
- VLOWer
- VMAX
- VMIDdle
- VMIN

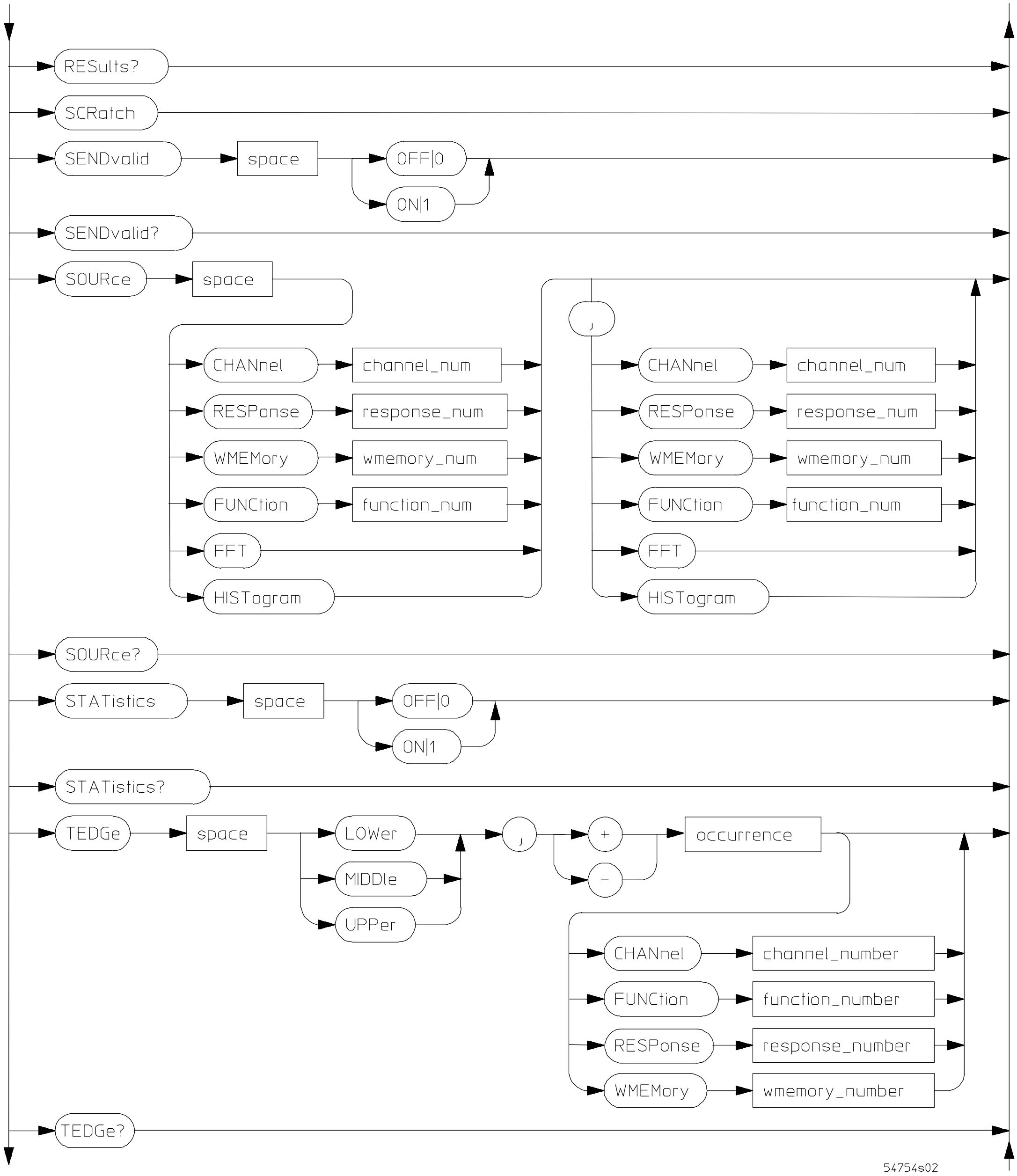
- VPP
- VRMS
- VTIMe
- VTOP
- VUPPer



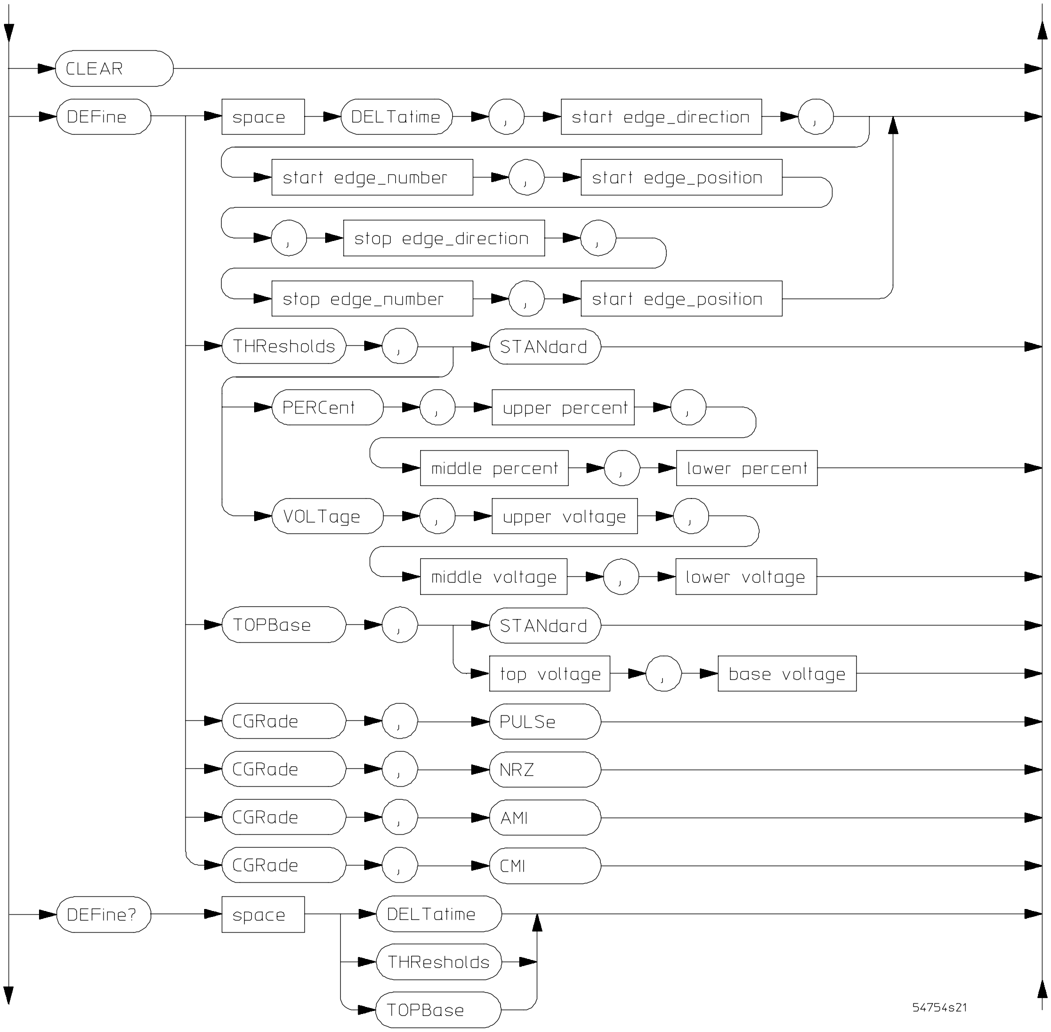
Measure Subsystem Command Syntax Diagram



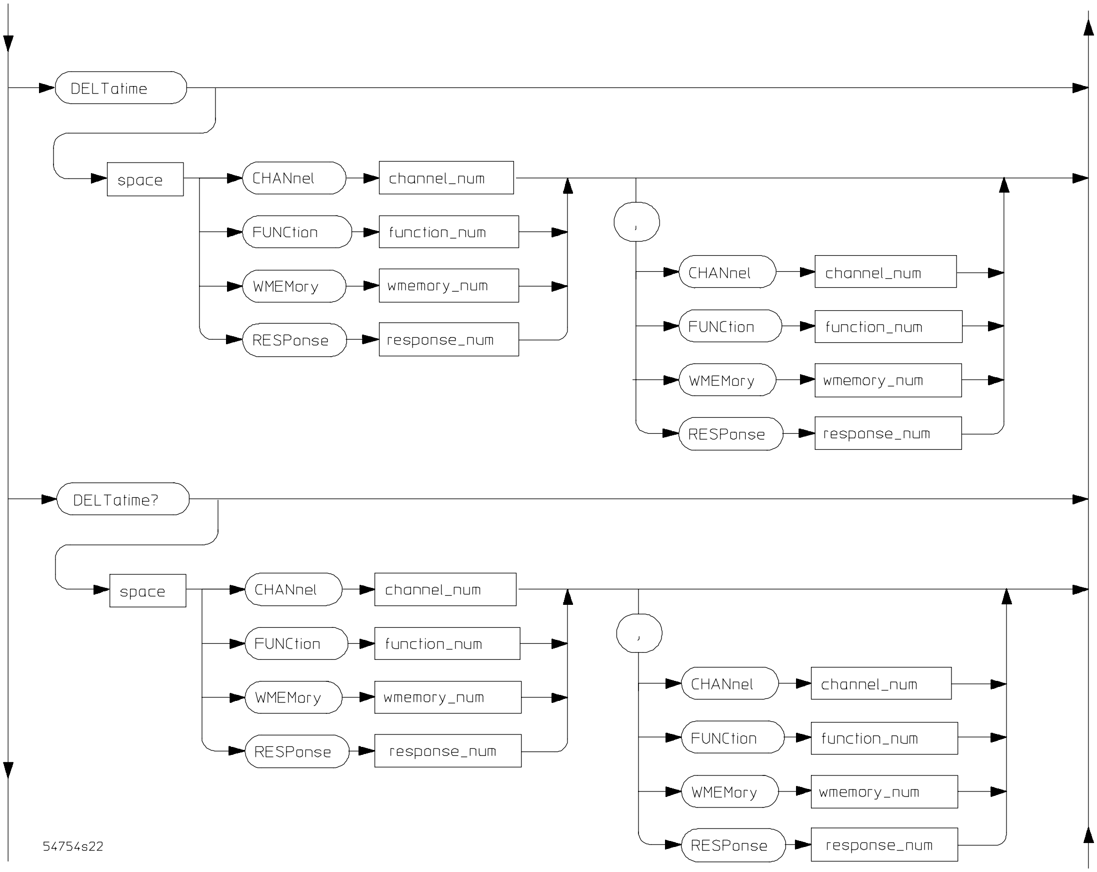
Measure Subsystem Command Syntax Diagram (continued)



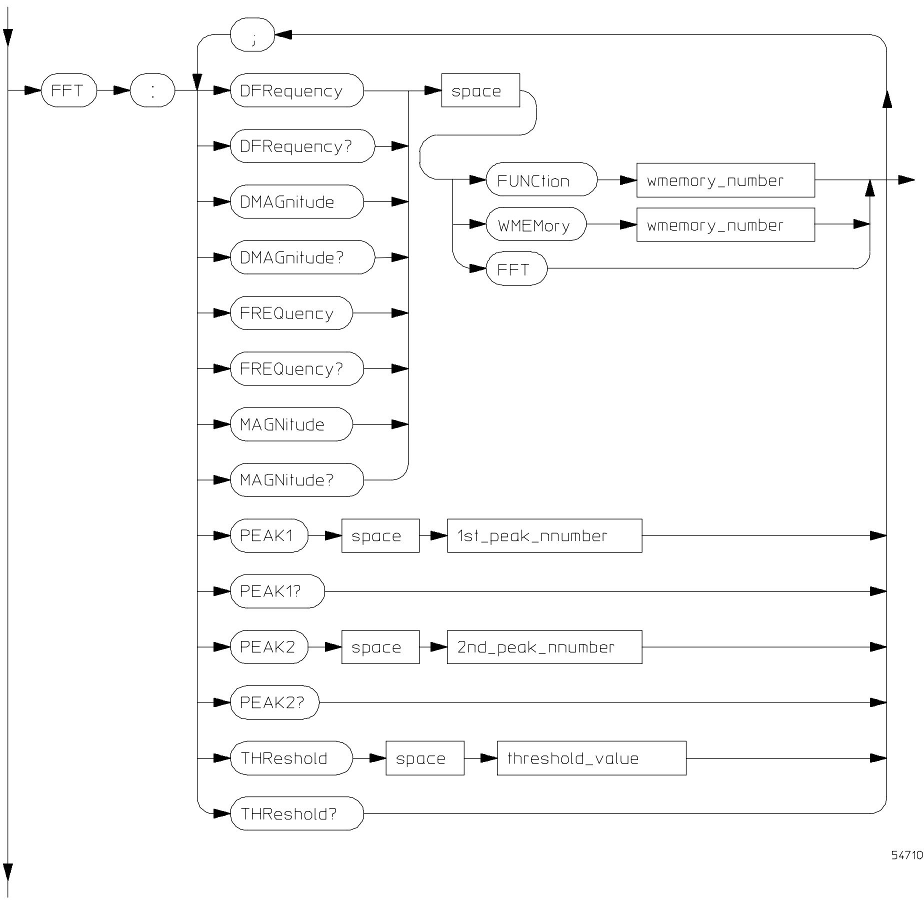
Measure Subsystem Command Syntax Diagram (continued)



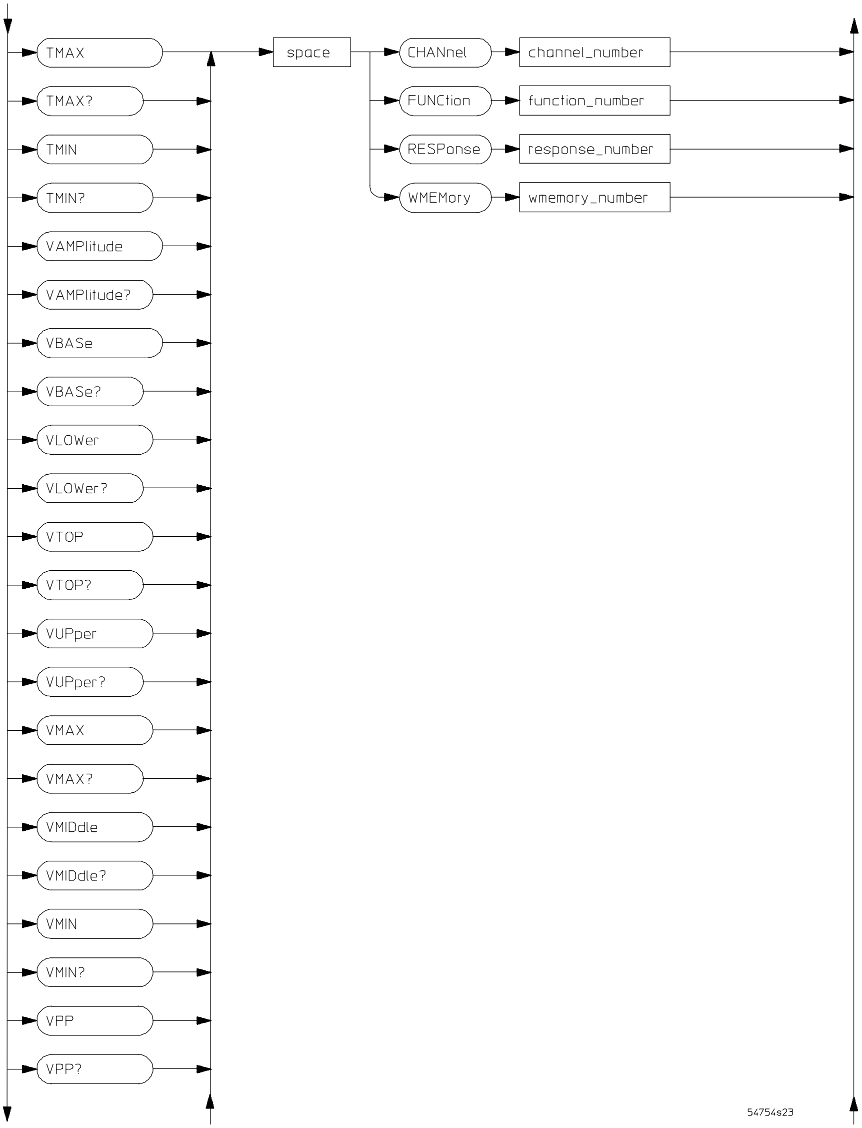
Measure Subsystem Command Syntax Diagram (continued)



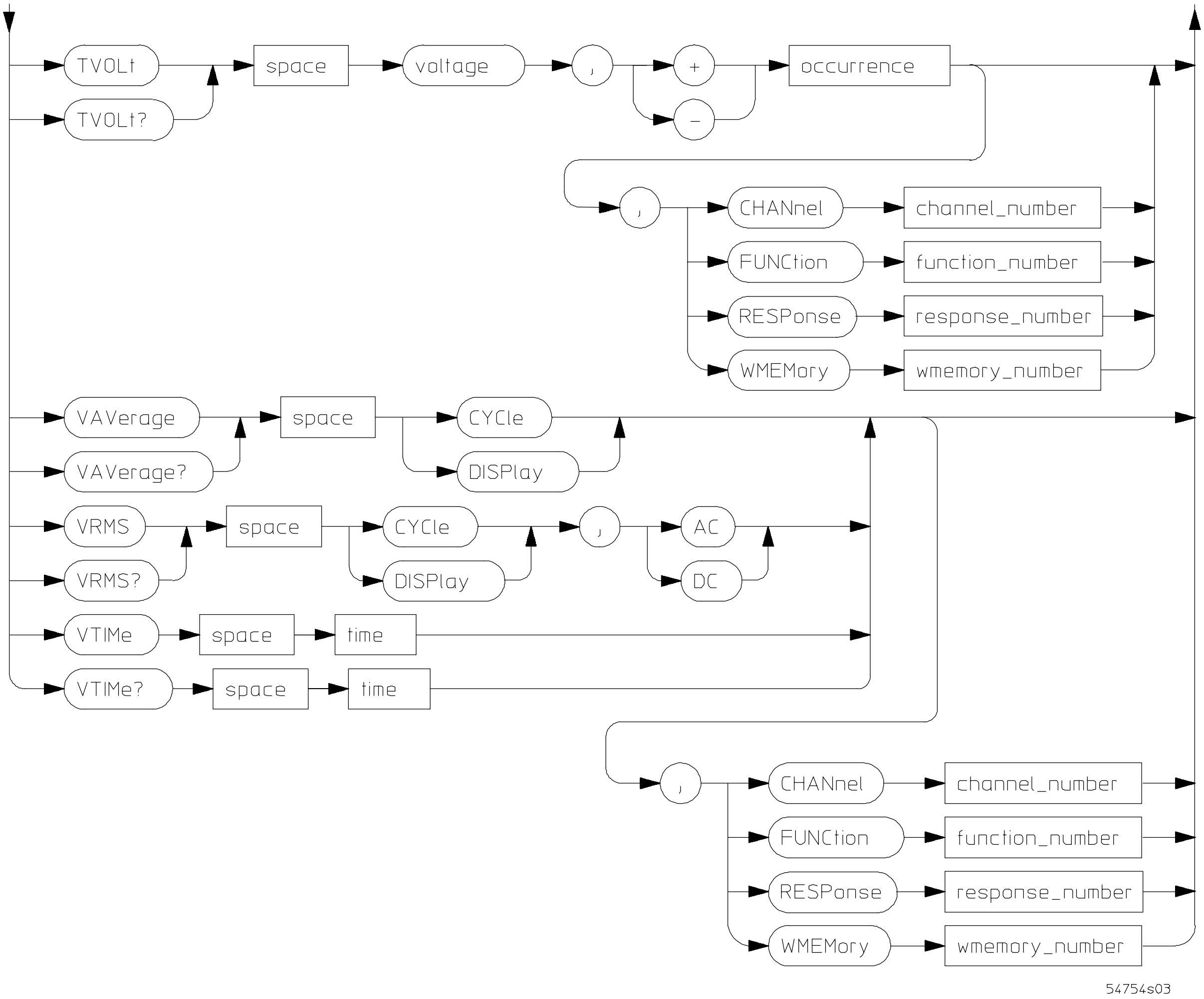
Measure Subsystem Command Syntax Diagram (continued)



Measure Subsystem Command Syntax Diagram (continued)



Measure Subsystem Command Syntax Diagram (continued)



Measure Subsystem Command Syntax Diagram (continued)

DELTatime

Command

```
:MEASure:DELTatime [<source>[,<source>]]
```

Measures the delta time between two edges. If one source is specified, the delta time from the leading edge of the specified source to the trailing edge of the specified source is measured. If two sources are specified, the delta time from the leading edge on the first source to the trailing edge on the second source is measured.

Sources are specified with the MEASure:SOURce command or with the optional parameters following the MEASure:DELTatime command. The rest of the parameters for this command are specified with the MEASure:DEFine command.

`<source>` {CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number>}

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

Example

The following example measures the delta time between channel 1 and TDR response 1.

```
10 OUTPUT 707;":MEASURE:DELTATIME CHANNEL1,RESPONSE1"
20 END
```

Query

```
:MEASure:DELTatime? [<source>[,<source>]]
```

The MEASure:DELTatime query returns the measured delta time value.

Returned Format `[:MEASure:DELTatime]<value>[,<result_state>]<NL>`

`<value>` Delta time from the first specified edge on one source to the next specified edge on another source.

`<result_state>` If SENDVALID is ON, the result state is returned with the measurement result. Refer to the MEASure:RESults command for a list of the result states.

Example

The following example places the current value of delta time in the numeric variable Value, then prints the contents of the variable to the controller's screen. This example assumes the source was set using MEASure:SOURce.

```
10 OUTPUT 707;":SYSTEM:HEADER OFF"     !Response headers off
20 OUTPUT 707;":MEASURE:DELTATIME?"
30 ENTER 707;Value
40 PRINT Value
50 END
```

NOTE

When receiving numeric data into numeric variables, turn off the headers. Otherwise, the headers may cause misinterpretation of the returned data.

DUTycycle

Command

```
:MEASure:DUTycycle [<source>]
```

Measures the ratio of the positive pulse width to the period. Sources are specified with the MEASure:SOURCe command or with the optional parameter following the DUTycycle command.

`<source>` {CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number>}

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

Example

The following example measures the duty cycle of the last specified signal.

```
10 OUTPUT 707;":MEASURE:DUTYCYCLE"
20 END
```

Query `:MEASure:DUTYCYCLE? [<source>]`

The MEASure:DUTycycle query returns the measured duty cycle of the specified source.

Returned Format `[:MEASure:DUTycycle]<value>[,<result_state>]<NL>`

`<value>` The ratio of the positive pulse width to the period.

`<result_state>` If SENDVALID is ON, the result state is returned with the measurement result. Refer to the MEASure:RESults command for a list of the result states.

Example

The following example places the current duty cycle of the specified signal in the numeric variable Value, then prints the contents of the variable to the controller's screen.

```
10 OUTPUT 707;":SYSTEM:HEADER OFF"     !Response headers off
20 OUTPUT 707;":MEASURE:DUTYCYCLE?"
30 ENTER 707;Value
40 PRINT Value
50 END
```

FALLtime

Command `:MEASure:FALLtime [<source>]`

Measures the time at the upper threshold of the falling edge, measures the time at the lower threshold of the falling edge, then calculates the fall time. The source is specified with the MEASure:SOURce command or with the optional parameter following the FALLtime command.

The first displayed falling edge is used for the fall time measurement. For best measurement accuracy, set the sweep speed as fast as possible while leaving the falling edge of the waveform on the display.

Fall time = time at lower threshold point - time at upper threshold point

`<source>` `{CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number>}`

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

Example

The following example measures the fall time of the last specified signal.

```
10 OUTPUT 707;":MEASURE:FALLTIME"
20 END
```

Query

```
:MEASure:FALLtime? [<source>]
```

The MEASure:FALLtime query returns the measured duty cycle of the specified source.

Returned Format `[:MEASure:FALLtime]<value>[,<result_state>]<NL>`

`<value>` Time at lower threshold - time at upper threshold.

`<result_state>` If SENDVALID is ON, the result state is returned with the measurement result. Refer to the MEASure:RESults command for a list of the result states.

Example

The following example places the current value for fall time in the numeric variable Value, then prints the contents of the variable to the controller's screen.

```
10 OUTPUT 707;":SYSTEM:HEADER OFF"     !Response headers off
20 OUTPUT 707;":MEASURE:FALLTIME?"
30 ENTER 707;Value
40 PRINT Value
50 END
```

FREQuency

Command

```
:MEASure:FREQuency [<source>]
```

Measures the frequency of the first complete cycle on the screen using the mid-threshold levels of the waveform (50% levels if standard measurements are selected). The source is specified with the MEASure:SOURce command or with the optional parameter following the FREQuency command.

The algorithm is:

If the 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).

`<source>` `{CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number>}`

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

Example

The following example measures the frequency of TDR response 3.

```
10 OUTPUT 707;":MEASURE:FREQUENCY RESPONSE3"
20 END
```

Query

```
:MEASure:FREQUENCY? [<source>]
```

The MEASure:FREQuency query returns the measured frequency.

Returned Format

```
[:MEASure:FREQuency]<value>[,<result_state>]<NL>
```

`<value>` The frequency value in Hertz of the first complete cycle on the screen using the mid-threshold levels of the waveform.

<result_state> If SENDVALID is ON, the result state is returned with the measurement result. Refer to the MEASure:RESults command for a list of the result states.

Example

The following example places the current frequency of the signal into the numeric variable Freq, then prints the contents of the variable to the controller's screen.

```
10 OUTPUT 707;":SYSTEM:HEADER OFF"     !Response headers off
20 OUTPUT 707;":MEASURE:FREQUENCY?"
30 ENTER 707;Freq
40 PRINT Freq
50 END
```

NWIDth

Command

```
:MEASure:NWIDTh [<source>]
```

Measures the width of the first negative pulse on the screen using the mid-threshold levels of the waveform (50% levels with standard measurements selected). The source is specified with the MEASure:SOURce command or with the optional parameter following the NWIDth command.

The algorithm is:

If the first edge on screen is rising then

nwidth = time at the second rising edge - time at the first falling edge

else

nwidth = time at the first rising edge - time at the first falling edge

<source> `{CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number>}`

<number> For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

Example

The following example measures the width of the first negative pulse on screen.

```
10 OUTPUT 707;":MEASURE:NWIDTH"
20 END
```

Query

```
:MEASure:NWIDth? [<source>]
```

The MEASure:NWIDth query returns the measured width of the first negative pulse of the specified source.

Returned Format

```
[:MEASure:NWIDth]<value>[,<result_state>]<NL>
```

`<value>` The width of the first negative pulse on the screen using the mid-threshold levels of the waveform.

`<result_state>` If SENDVALID is ON, the result state is returned with the measurement result. Refer to the MEASure:RESults command for a list of the result states.

Example

The following example places the current width of the first negative pulse on the screen into the numeric variable, Width, then prints the contents of the variable to the controller’s screen.

```
10 OUTPUT 707;":SYSTEM:HEADER OFF"     !Response headers off
20 OUTPUT 707;":MEASURE:NWIDTH?"
30 ENTER 707;Width
40 PRINT Width
50 END
```

OVERshoot

Command `:MEASure:OVERshoot [<source>]`

Measures the width of the overshoot of the first edge on the screen. The source is specified with the MEASure:SOURce command or with the optional parameter following the OVERshoot command.

The algorithm is:

If the first edge on screen is rising, then

overshoot = (Local V - V) / V

else

overshoot = (V - Local V) / V.

`<source>` `{CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number>}`

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

Example The following example measures the overshoot of the first edge on screen.

```
10 OUTPUT 707;":MEASURE:OVERSHOOT"
20 END
```

Query `:MEASure:OVERshoot? [<source>]`

The MEASure:OVERshoot query returns the measured overshoot of the specified source.

Returned Format `[:MEASure:OVERshoot]<value>[,<result_state>]<NL>`

`<value>` Ratio of overshoot to amplitude, in percent.

`<result_state>` If SENDVALID is ON, the result state is returned with the measurement result. Refer to the MEASure:RESults command for a list of the result states.

Example

The following example places the current value of overshoot into the numeric variable Value, then prints the contents of the variable to the controller's screen.

```
10 OUTPUT 707;":SYSTEM:HEADER OFF"     !Response headers off
20 OUTPUT 707;":MEASURE:OVERSHOOT?"
30 ENTER 707;Value
40 PRINT Value
50 END
```

PERiod

Command

```
:MEASure:PERiod [<source>]
```

Measures the period of the first complete cycle on the screen using the mid-threshold levels of the waveform (50% levels with standard measurements selected). The source is specified with the MEASure:SOURce command or with the optional parameter following the PERiod command.

The algorithm is:

If the first edge on screen is rising then

period = time at the second rising edge - time at the first rising edge

else

period = time at the second falling edge - time at the first falling edge.

`<source>` {CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number>}

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

Example

The following example measures the period of the waveform.

```
10 OUTPUT 707;":MEASURE:PERIOD"
20 END
```

Query

```
:MEASure:PERiod? [<source>]
```

The MEASure:PERiod query returns the measured period of the specified source.

Returned Format `[:MEASure:PERiod]<value>[,<result_state>]<NL>`

`<value>` Period of the first complete cycle on screen.

`<result_state>` If SENDVALID is ON, the result state is returned with the measurement result. Refer to the MEASure:RESults command for a list of the result states.

Example

The following example places the current period of the waveform into the numeric variable Value, then prints the contents of the variable to the controller's screen.

```
10 OUTPUT 707;":SYSTEM:HEADER OFF"     !Response headers off
20 OUTPUT 707;":MEASURE:PERIOD?"
30 ENTER 707;Value
40 PRINT Value
50 END
```

PREShoot

Command

```
:MEASure:PREShoot [<source>]
```

Measures the preshoot of the first edge on the screen. Sources are specified with the MEASure:SOURce command or with the optional parameter following the PREShoot command.

If the first edge on the screen is rising then

preshoot = (V - Local V) / V

else

preshoot = (Local V - V) / V.

`<source>` {CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number>}

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

Example

The following example measures the preshoot of the signal on screen.

```
10 OUTPUT 707;":MEASURE:PRESHOOT"
20 END
```

Query

```
:MEASure:PREShoot? [<source>]
```

The MEASure:PREShoot query returns the measured preshoot of the specified source.

Returned Format

```
[:MEASure:PREShoot]<value>[,<result_state>]<NL>
```

`<value>` Ratio of preshoot to amplitude, in percent.

`<result_state>` If SENDVALID is ON, the result state is returned with the measurement result. Refer to the MEASure:RESults command for a list of the result states.

Example

The following example places the current preshoot into the numeric variable Preshoot, then prints the contents of the variable to the controller’s screen.

```
10 OUTPUT 707;":SYSTEM:HEADER OFF"     !Response headers off
20 OUTPUT 707;":MEASURE:PRESHOOT?"
30 ENTER 707;Preshoot
40 PRINT Preshoot
50 END
```

PWIDth

Command

```
:MEASure:PWIDth [<source>]
```

Measures the width of the first positive pulse on the screen using the mid-threshold levels of the waveform (50% levels with standard measurements selected). The source is specified with the MEASure:SOURce command or with the optional parameter following the PWIDth command.

The algorithm is:

If the first edge on screen is rising then

pwidth = time at the first falling edge - time at the first rising edge

else

pwidth = time at the second falling edge - time at the first rising edge.

`<source>` {CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number>}

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

Example

The following example measures the width of the first positive pulse on the screen.

```
10 OUTPUT 707;":MEASURE:PWIDTH"
20 END
```

Query

```
:MEASure:PWIDTH? [<source>]
```

The MEASure:PWIDth query returns the measured width of the first positive pulse of the specified source.

Returned Format `[:MEASure:PWIDth]<value>[,<result_state>]<NL>`

`<value>` Width of the first positive pulse on the screen in seconds.

`<result_state>` If SENDVALID is ON, the result state is returned with the measurement result. Refer to the MEASure:RESults command for a list of the result states.

Example

The following example places the value of the width of the first positive pulse on the screen into the numeric variable Width, then prints the contents of the variable to the controller's screen.

```
10 OUTPUT 707;":SYSTEM:HEADER OFF"     !Response headers off
20 OUTPUT 707;":MEASURE:PWIDTH?"
30 ENTER 707;Width
40 PRINT Width
50 END
```

RISetime

Command `:MEASure:RISetime [<source>]`

Measures the rise time of the first displayed edge by measuring the time at the lower threshold of the rising edge, measuring the time at the upper threshold of the rising edge, then calculating the rise time with the following algorithm:

Rise time = time at upper threshold point - time at lower threshold point.

The source is measured with the MEASure:SOURce command or with the optional parameter following the RISetime command.

`<source>` `{CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number>}`

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

With standard measurements selected, the lower threshold is at the 10% point and the upper threshold is at the 90% point on the rising edge.

Example

The following example measures the rise time of the displayed signal.

```
10 OUTPUT 707;":MEASURE:RISETIME"
20 END
```

Query

```
:MEASure:RISetime? [<source>]
```

The MEASure:RISetime query returns the rise time of the specified source.

Returned Format `[:MEASure:RISetime]<value>[,<result_state>]<NL>`

`<value>` Rise time in seconds.

`<result_state>` If SENDVALID is ON, the result state is returned with the measurement result. Refer to the MEASure:RESults command for a list of the result states.

Example

The following example places the current value of rise time into the numeric variable Rise, then prints the contents of the variable to the controller's screen.

```
10 OUTPUT 707;":SYSTEM:HEADER OFF"     !Response headers off
20 OUTPUT 707;":MEASURE:RISETIME?"
30 ENTER 707;Rise
40 PRINT Rise
50 END
```

SOURce

Command

```
:MEASure:SOURce <source>[,<source>]
```

Selects the source for measurements. Two sources can be specified with this command. All measurements except MEASure:DELTatime are made on the first specified source. The delta time measurement uses two sources if two are specified. If only one source is specified, the delta time measurement uses that source for both of its parameters.

`<source>` {CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number> | HISTogram | FFT}

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

Example

The following example selects TDR response 1 as the source for measurements.

```
10 OUTPUT 707;":MEASURE:SOURCE RESPONSE1"
20 END
```

Query

```
:MEASure:SOURce?
```

The MEASure:SOURce query returns the current source selection.

Returned Format

```
[:MEASure:SOURce]<source>[,<source>]<NL>
```

Example

The following example places the currently specified sources in the string variable Source$, then prints the contents of the variable to the controller's screen.

```
10 DIM Source$[50]     !Dimension variable
20 OUTPUT 707;":MEASURE:SOURce?"
30 ENTER 707;Source$
40 PRINT Source$
50 END
```

TEDge

```
:MEASURE:TEDge
<meas_thres_txt>,<slope><occurrence>[,<source>]
```

Measures the time interval between the trigger event and the specified threshold level and transition. The source is specified with the MEASure:SOURce command or with the optional parameter following the TEDGe command.

`<meas_thres_txt>` `{UPPer | MIDDle | LOWer}`

`<slope>` {- (minus) for falling | + (plus) (optional) for rising}

`<occurrence>` Numeric value representing which edge of the occurrence should be used for the measurement.

`<source>` `{CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number>}`

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

Query

```
:MEASURE:TEDge?
<meas_thres_txt>,<slope><occurrence>[,<source>]
```

The query returns the time interval between the trigger event and the specified threshold level and transition.

Returned Format `[:MEASURE:TEDge]<time>[,<result_state>]<NL>`

`<time>` The time interval between the trigger event and the specified voltage level and transition.

`<result_state>` If SENDVALID is ON, the result state is returned with the measurement result. Refer to the MEASure:RESults command for a list of the result states.

Example

The following example returns the time interval between the trigger event and the 90% threshold on the second rising edge of the source waveform to the numeric variable Time. The contents of the variable are then printed to the controller's screen.

```
10 OUTPUT 707;":SYSTEM:HEADER OFF"      !Response headers off
20 OUTPUT 707;":MEASURE:TEDGE? UPPER,+2"
30 ENTER 707;Time
40 PRINT Time
50 END
```

NOTE

When receiving numeric data into numeric variables, turn off the headers. Otherwise, the headers may cause misinterpretation of returned data.

TMAX

Command `:MEASure:TMAX [<source>]`

Measures the first time at which the maximum voltage of the source waveform occurred. When FFT is the specified source, the frequency at which the first maximum value occurred is measured. The source is specified with the MEASure:SOURce command or with the optional parameter following the TMAX command.

`<source>` `{CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number> | FFT}`

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

Query

```
:MEASure:TMAX? [<source>]
```

The MEASure:TMAX query returns the time at which the first maximum voltage occurred.

Returned Format

```
[:MEASure:TMAX]<value>[,<result_state>]<NL>
```

`<value>` Time at which the first maximum voltage occurred.

`<result_state>` If SENDVALID is ON, the result state is returned with the measurement result. Refer to the MEASure:RESults command for a list of the result states.

Example

The following example returns the time at which the first maximum voltage occurred to the numeric variable Time, then prints the contents of the variable to the controller's screen.

```
10 OUTPUT 707;":SYSTEM:HEADER OFF"     !Response headers off
20 OUTPUT 707;":MEASURE:TMAX?"
30 ENTER 707;Time
40 PRINT Time
50 END
```

NOTE

When receiving numeric data into numeric variables, turn off headers. Otherwise, the headers may cause misinterpretation of the returned data.

TMIN

Command

```
:MEASure:TMIN [<source>]
```

Measures the time at which the first minimum voltage of the source waveform occurred. When FFT is the specified source, the frequency at which the first minimum value occurred is measured. The source is specified with the MEASure:SOURce command or with the optional parameter following the TMIN command.

<source> {CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number> | FFT}

<number> For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

Query

```
:MEASure:TMIN? [<source>]
```

The MEASure:TMIN query returns the time at which the first minimum voltage occurred.

Returned Format

```
[:MEASure:TMIN]<value>[,<result_state>]<NL>
```

<value> Time at which the first minimum voltage occurred.

<result_state> If SENDVALID is ON, the result state is returned with the measurement result. Refer to the MEASure:RESults command for a list of the result states.

Example

The following example returns the time at which the first minimum voltage occurred to the numeric variable Time, then prints the contents of the variable to the controller's screen.

```
10 OUTPUT 707;":SYSTEM:HEADER OFF"     !Response headers off
20 OUTPUT 707;":MEASURE:TMIN?"
30 ENTER 707;Time
40 PRINT Time
50 END
```

TVOLt

Command

```
:MEASure:TVOLt
<voltage>,<slope><occurrence>[,<source>]
```

Measures the time interval between the trigger event and the defined voltage level and transition. The source is specified with the MEASure:SOURce command or with the optional parameter following the TVOLt command.

`<voltage>` Voltage level at which time will be measured.

`<slope>` The direction of the waveform change when the specified voltage is crossed, either rising (+) or falling (–).

`<occurrence>` The number of the crossing to be reported (if one, the first crossing is reported; if two, the second crossing is reported, and so on).

`<source>`

```
{CHANnel<number> | FUNCtion<number> |
RESPonse<number> | WMEMory<number>}
```

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

Query

```
:MEASure:TVOLt?
<voltage>,<slope><occurrence>[,<source>]
```

The MEASure:TVOLt query returns the time interval between the trigger event and the specified voltage level and transition.

Returned Format

```
[:MEASure:TVOLt]<time>[,<result_state>]<NL>
```

`<time>` The time interval between the trigger event and the specified voltage level and transition.

`<result_state>` If SENDVALID is ON, the result state is returned with the measurement result. Refer to the MEASure:RESults command for a list of the result states.

Example

The following example returns the time interval between the trigger event and the transition through .250 volts on the third rising edge of the source waveform to the numeric variable, Time. The contents of the variable are then printed to the controller's screen.

```
10 OUTPUT 707;":SYSTEM:HEADER OFF"
20 OUTPUT 707;"MEASURE:TVOLT? -.250,+3"
30 ENTER 707;Time
40 PRINT Time
50 END
```

NOTE

When receiving numeric data into numeric variables, turn off headers. Otherwise, the headers may cause misinterpretation of the data.

VAMPlitude

Command

```
:MEASure:VAMPlitude [<source>]
```

Calculates the difference between the top and base voltage of the specified source. The source is specified with the MEASure:SOURce command or with the optional parameter following the VAMPlitude command.

<source> {CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number>}

<number> For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR respnses: 1, 2, 3, or 4.

Example

The following example calculates the difference between the top and base voltage of the specified source.

```
10 OUTPUT 707;":MEASURE:VAMPLITUDE"
20 END
```

Query

```
:MEASure:VAMPlitude? [<source>]
```

The MEASure:VAMPlitude query returns the calculated difference between the top and base voltage of the specified source.

Returned Format

```
[:MEASure:VAMPlitude]<value>[,<result_state>]<NL>
```

`<value>` Caclulated difference between the top and base voltage.

`<result_state>` If SENDVALID is ON, the result state is returned with the measurement result. Refer to the MEASure:RESults command for a list of the result states.

Example

The following example places the current amplitude value into the numeric variable Value, then prints the contents of the variable to the controller's screen.

```
10 OUTPUT 707;":SYSTEM:HEADER OFF"     !Response headers off
20 OUTPUT 707;":MEASURE:VAMPLITUDE?"
30 ENTER 707;Value
40 PRINT Value
50 END
```

VAVerage

Command

```
:MEASure:VAVerage {CYCLe | DISPlay},[<source>]
```

Calculates the average voltage over the displayed waveform. The source is specified with the MEASure:SOURce command or with the optional parameter following the VAVerage command.

`CYCLe` The CYCLe parameter instructs the average measurement to measure the average voltage across the first period on the display.

`DISPlay` The DISPlay parameter instructs the average measurement to measure all the data on the display.

`<source>` `{CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number>}`

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3 or 4.

Example The following example calculates the average voltage over the displayed waveform.

```
10 OUTPUT 707;":MEASURE:VAVERAGE DISPLAY"
20 END
```

Query `:MEASure:VAVerage? {CYCLe | DISPlay},[<source>]`

The MEASure:VAVerage query returns the calculated average voltage of the specified source.

Returned Format `[:MEASure:VAVerage]<value>[,<result_state>]<NL>`

`<value>` The calculated average voltage.

`<result_state>` If SENDVALID is ON, the result state is returned with the measurement result. Refer to the MEASure:RESults command for a list of the result states.

Example The following example places the current average voltage in the numeric variable Average, then prints the contents of the variable to the controller’s screen.

```
10 ":SYSTEM:HEADER OFF" !Response headers off
20 OUTPUT 707;":MEASURE:VAVERAGE? DISPLAY"
30 ENTER 707;Average
40 PRINT Average
50 END
```

VBASe

Command `:MEASure:VBASe [<source>]`

Measures the statistical base of the waveform. The source is specified with the MEASure:SOURce command or with the optional parameter following the VBASe command.

`<source>` `{CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number>}`

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

Example The following example measures the voltage at the base of the waveform.

```
10 OUTPUT 707;":MEASURE:VBASE"
20 END
```

Query `:MEASure:VBASe? [<source>]`

The MEASure:VBASe query returns the measured voltage value at the base of the specified source.

Returned Format `[:MEASure:VBASe]<value>[,<result_state>]<NL>`

`<value>` Voltage at the base of the waveform.

`<result_state>` If SENDVALID is ON, the result state is returned with the measurement result. Refer to the MEASure:RESults command for a list of the result states.

Example

The following example returns the current voltage at the base of the waveform to the numeric variable Voltage, then prints the contents of the variable to the controller's screen.

```
10 OUTPUT 707;":SYSTEM:HEADER OFF"     !Response headers off
20 OUTPUT 707;":MEASURE:VBASE?"
30 ENTER 707;Voltage
40 PRINT Voltage
50 END
```

VLOWer

Command

```
:MEASure:VLOWer [<source>]
```

Measures the voltage value at the lower threshold of the waveform. The source is specified with the MEASure:SOURce command or with the optional parameter following the VLOWer command.

`<source>` {CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number>}

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

Query

```
:MEASure:VLOWer? [<source>]
```

This query returns the measured lower threshold of the selected source.

Returned Format

```
[:MEASure:VLOWer]<value>[,<result_state>]<NL>
```

`<value>` Voltage value at the lower threshold.

`<result_state>` If SENDVALID is ON, the result state is returned with the measurement result. Refer to the MEASure:RESults command for a list of the result states.

Example

The following example returns the measured voltage at the lower threshold of the waveform to the numeric variable Vlower, then prints the contents of the variable to the controller's screen.

```
10 OUTPUT 707;":SYSTEM:HEADER OFF"     !Response headers off
20 OUTPUT 707;":MEASURE:VLOW?"
30 ENTER 707;Vlower
40 PRINT Vlower
50 END
```

VMAX

Command `:MEASure:VMAX [<source>]`

Measures the absolute maximum voltage present on the selected source waveform. When FFT is the specified source, the maximum value in spectrum is measured. The source is specified with the MEASure:SOURce command or with the optional parameter following the VMAX command.

`<source>` {CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number> | FFT}

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

Example

The following example measures the absolute maximum voltage on the waveform.

```
10 OUTPUT 707;":MEASURE:VMAX"
20 END
```

Query

```
:MEASure:VMAX? [<source>]
```

The MEASure:VMAX query returns the measured absolute maximum voltage present on the selected source waveform.

Returned Format `[:MEASure:VMAX]<value>[,<result_state>]<NL>`

`<value>` Absolute maximum voltage present on the waveform.

`<result_state>` If SENDVALID is ON, the result state is returned with the measurement result. Refer to the MEASure:RESults command for a list of the result states.

Example

The following example returns the measured absolute maximum voltage on the waveform to the numeric variable Maximum, then prints the contents of the variable to the controller's screen.

```
10 OUTPUT 707;":SYSTEM:HEADER OFF"     !Response headers off
20 OUTPUT 707;":MEASURE:VMAX?"
30 ENTER 707;Maximum
40 PRINT Maximum
50 END
```

VMIDdle

Command

```
:MEASure:VMIDdle [<source>]
```

Measures the voltage value at the middle threshold of the waveform. The source is specified with the MEASure:SOURce command or with the optional parameter following the VMIDdle command.

`<source>` `{CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number>}`

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

Query

```
:MEASure:VMIDdle? [<source>]
```

This query returns the voltage value at the middle threshold of the waveform.

Returned Format `[:MEASure:VMIDdle]<value>[,<result_state>]<NL>`

`<value>` The middle voltage present on the waveform.

`<result_state>` If SENDVALID is ON, the result state is returned with the measurement result. Refer to the MEASure:RESults command for a list of the result states.

Example

The following example returns the measured middle voltage on the waveform to the numeric variable Middle, then prints the contents of the variable to the controller's screen.

```
10 OUTPUT 707;":SYSTEM:HEADER OFF"     !Response headers off
20 OUTPUT 707;":MEASURE:VMID?"
30 ENTER 707;Middle
40 PRINT Middle
50 END
```

VMIN

Command

```
:MEASure:VMIN [<source>]
```

Measures the absolute minimum voltage present on the selected source waveform. When FFT is the specified source, the minimum value in the spectrum is measured. The source is specified with the MEASure:SOURce command or with the optional parameter following the VMIN command.

`<source>` `{CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number> | FFT}`

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

Example

The following example measures the absolute minimum voltage on the waveform.

```
10 OUTPUT 707;":MEASURE:VMIN"
20 END
```

Query

```
:MEASure:VMIN? [<source>]
```

The MEASure:VMIN query returns the measured absolute minimum voltage present on the selected source waveform.

Returned Format `[:MEASure:VMIN]<value>[,<result_state>]<NL>`

`<value>` Absolute minimum voltage present on the waveform.

`<result_state>` If SENDVALID is ON, the result state is returned with the measurement result. Refer to the MEASure:RESults command for a list of the result states.

Example

The following example returns the measured absolute minimum voltage on the waveform to the numeric variable Minimum, then prints the contents of the variable to the controller's screen.

```
10 OUTPUT 707;":SYSTEM:HEADER OFF"     !Response headers off
20 OUTPUT 707;":MEASURE:VMIN?"
30 ENTER 707;Minimum
40 PRINT Minimum
50 END
```

VPP

Command

```
:MEASure:VPP [<source>]
```

Measures the absolute maximum and minimum voltages on the selected source, then calculates the peak-to-peak voltage as the difference between the two voltages. When FFT is the specified source, the range of values in the spectrum is measured. The source is specified with the MEASure:SOURce command or with the optional parameter following the VPP command.

<source> {CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number> | FFT}

<number> For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

Example

The following example measures the peak-to-peak voltage of a previously selected source.

```
10 OUTPUT 707;":MEASURE:VPP"
20 END
```

Query

```
:MEASure:VPP? [<source>]
```

The MEASure:VPP query returns the peak-to-peak voltage of the specified source.

Returned Format `[:MEASure:VPP]<value>[,<result_state>]<NL>`

<value> Peak-to-peak voltage of the selected source.

<result_state> If SENDVALID is ON, the result state is returned with the measurement result. Refer to the MEASure:RESults command for a list of the result states.

Example

The following example places the current peak-to-peak voltage in the numeric variable Voltage, then prints the contents of the variable to the controller's screen.

```
10 OUTPUT 707;":SYSTEM:HEADER OFF"     !Response headers off
20 OUTPUT 707;":MEASURE:VPP?"
30 ENTER 707;Voltage
40 PRINT Voltage
50 END
```

VRMS

Command

```
:MEASure:VRMS {CYCLe | DISPlay}, {AC | DC}
[,<source>]
```

Measures the RMS voltage of the selected waveform by subtracting the average value of the waveform from each data point on the display. The source is specified with the MEASure:SOURce command or with the optional parameter following the VRMS command.

`CYCLe` The CYCLe parameter instructs the RMS measurement to measure the RMS voltage across the first period shown on the display.

`DISPlay` The DISPlay parameter instructs the RMS measurement to measure all the data on the display. Generally, RMS voltage is measured across one period; however, measuring multiple periods may be done using the DISPlay option. The DISPlay parameter is also useful when measuring noise.

`AC` The AC parameter is used to measure the RMS voltage, subtracting out the DC component.

`DC` The DC parameter is used to measure RMS voltage including the DC component.

The AC RMS, DC RMS, and VAVG parameters are related as in the following formula:

$DCVRMS = ACVRMS + VAVG$

`<source>` `{CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number>}`

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

Example The following example measures the RMS voltage of the previously selected waveform.

```
10 OUTPUT 707;":MEASURE:VRMS CYCLE,AC"
20 END
```

Query

```
:MEASure:VRMS? {CYCLe | DISPlay},{AC |
DC}[,<source>]
```

The MEASure:VRMS query returns the RMS voltage of the specified source.

Returned Format `[:MEASure:VRMS]<value>[,<result_state>]<NL>`

`<value>` RMS voltage of the selected waveform.

`<result_state>` If SENDVALID is ON, the result state is returned with the measurement result. Refer to the MEASure:RESults command for a list of the result states.

Example The following example places the current AC RMS voltage over one period of the waveform in the numeric variable Voltage, then prints the contents of the variable to the controller's screen.

```
10 OUTPUT 707;":SYSTEM:HEADER OFF"  !Response headers off
20 OUTPUT 707;":MEASURE:VRMS? CYCLE,AC"
30 ENTER 707;Voltage
40 PRINT Voltage
50 END
```

VTIMe

Command

```
:MEASure:VTIMe <time>[,<source>]
```

Measures the voltage at the specified time. The time is referenced to the trigger event and must be on the screen. When FFT is the specified source, the value at the specified frequency is measured. The source is specified with the MEASure:SOURce command or with the optional parameter following the VTIMe command.

`<source>` {CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number> | FFT}

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

`<time>` Displayed time from the trigger in seconds, or frequency in Hertz for FFT.

Query

```
:MEASure:VTIMe? <time>[,<seconds>]
```

The query returns the measured voltage.

Returned Format

```
[:MEASure:VTIMe]<value>[,<result_state>]<NL>
```

`<value>` Voltage at the specified time. When the source is FFT, the returned value is the vertical value at the horizontal setting passed in the VTIMe <time> parameter. Also, when the source is FFT, the time parameter is in Hertz.

`<result_state>` If SENDVALID is ON, the result state is returned with the measurement result. Refer to the MEASure:RESults command for a list of the result states.

Example

The following example places the voltage at 500 ms in the numeric variable Value, then prints the contents of that variable to the controller's screen.

```
10 OUTPUT 707;":SYSTEM:HEADER OFF"     !Response headers off
20 OUTPUT 707;":MEASURE:VTIME? 500E-3"
30 ENTER 707;Value
40 PRINT Value
50 END
```

VTOP

Command

```
:MEASure:VTOP [<source>]
```

Measures the statistical top of the selected source waveform. The source is specified with the MEASure:SOURce command or with the optional parameter following the VTOP command.

`<source>` `{CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number>}`

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3 or 4.

Example

The following example measures the voltage at the top of the waveform.

```
10 OUTPUT 707;":MEASURE:VTOP"
20 END
```

Query

```
:MEASure:VTOP? [<source>]
```

The MEASure:VTOP query returns the measured voltage at the top of the specified source.

Returned Format `[:MEASure:VTOP]<value>[,<result_state>]<NL>`

`<value>` Voltage at the top of the waveform.

`<result_state>` If SENDVALID is ON, the result state is returned with the measurement result. Refer to the MEASure:RESults command for a list of the result states.

Example

The following example places the value of the voltage at the top of the waveform in the numeric variable Value, then prints the contents of the variable to the controller's screen.

```
10 OUTPUT 707;":SYSTEM:HEADER OFF"     !Response headers off
20 OUTPUT 707;":MEASURE:VTOP?"
30 ENTER 707;Value
40 PRINT Value
50 END
```

VUPper

Command `:MEASure:VUPper [<source>]`

Measures the voltage value at the upper threshold of the waveform. The source is specified with the MEASure:SOURce command or with the optional parameter following the VUPper command.

`<source>` `{CHANnel<number> | FUNCtion<number> | RESPonse<number> | WMEMory<number>}`

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

Example

The following example measures the voltage at the upper threshold of the waveform.

```
10 OUTPUT 707;":MEASURE:VUPper"
20 END
```

Query

```
:MEASure:VUPper? [<source>]
```

The MEASure:VUPper query returns the measured upper threshold value of the selected source.

Returned Format `[:MEASure:VUPper]<value>[,<result_state>]<NL>`

`<value>` Voltage at the upper threshold.

`<result_state>` If SENDVALID is ON, the result state is returned with the measurement result. Refer to the MEASure:RESults command for a list of the result states.

Example

The following example places the value of the voltage at the upper threshold of the waveform in the numeric variable Value, then prints the contents of the variable to the controller's screen.

```
10 OUTPUT 707;":SYSTEM:HEADER OFF"     !Response headers off
20 OUTPUT 707;":MEASURE:VUPPER?"
30 ENTER 707;Value
40 PRINT Value
50 END
```

14

TDR Commands

TDR Commands

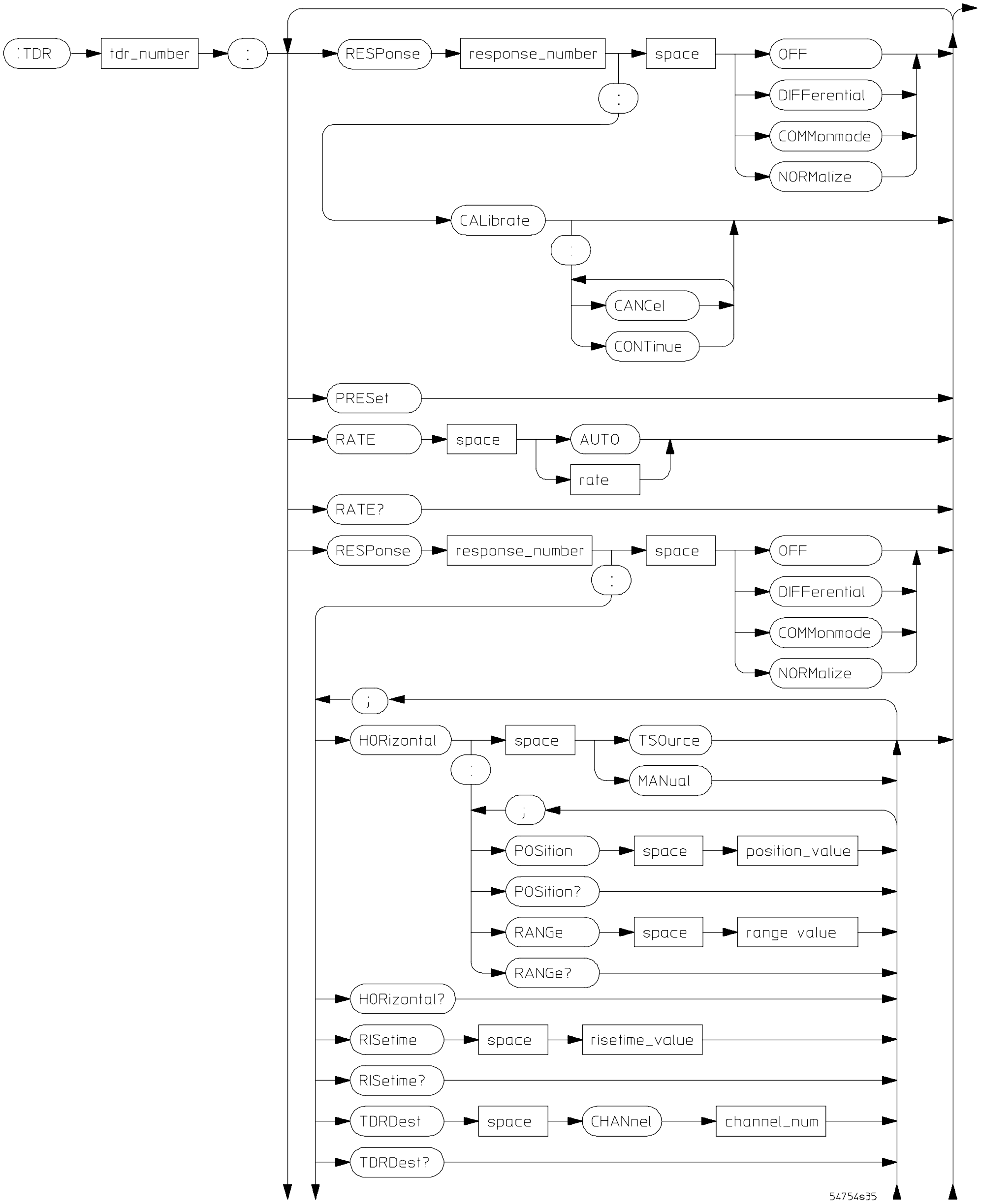
The TDR command subsystem includes all commands necessary to set up TDR measurements.

The TDR command subsystem includes the following commands:

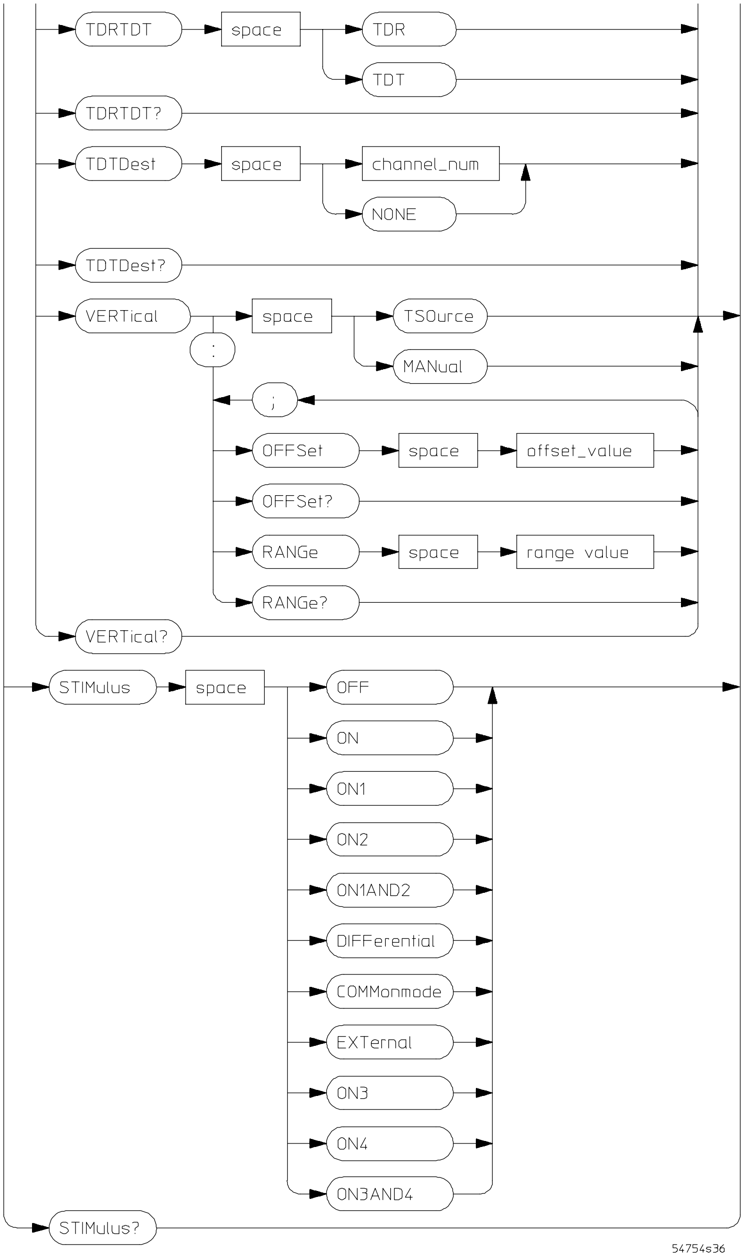
- RESPonse:CALibrate
- RESPonse:CALibrate:CANCel
- RESPonse:CALibrate:CONTinue
- PRESet
- RATE
- RESPonse
- RESPonse:HORizontal
- RESPonse:HORizontal:POSition
- RESPonse:HORizontal:RANGe
- RESPonse:RISetime
- RESPonse:TDRDest
- RESPonse:TDRTDT
- RESPonse:TDTDest
- RESPonse:VERTical
- RESPonse:VERTical:OFFSet
- RESPonse:VERTical:RANGe
- STIMulus

Slot Selection

All of the TDR subsystem commands are of the form :TDR{2 | 4}:<command>. The {2 | 4} option is used to identify the slot in which you have installed the TDR plug-in module. Because the 54753 and 54754 modules each occupy two slots, you select "2" if the module is in slots 1 and 2 and "4" if the module is in slots 3 and 4. For example, if the module is in slots 3 and 4, and you want to issue the TDR subsystem PRESet command, you use the command string :TDR4:PRESET.



TDR Subsystem Command Syntax Diagram



TDR Subsystem Command Syntax Diagram (continued)

RESPonse:CALibrate

Command

```
:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:CALibrate
```

Begins a TDR or TDT normalization and reference plane calibration. Which calibration is done (TDR or TDT) depends on the setting of the TDRTDT control. See the TDRTDT command in this chapter.

Example

The following example begins a TDR or TDT calibration.

```
10 OUTPUT 707;":TDR2:RESPONSE1:CALIBRATE"
20 END
```

RESPonse:CALibrate:CANCel

Command

```
:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:CALibrate:
CANCel
```

Activates the cancel softkey during a TDR or TDT normalization and reference plane calibration.

Example

The following example cancels the current calibration operation.

```
10 OUTPUT 707;":TDR2:RESPONSE1:CALIBRATE:CANCEL"
20 END
```

RESPonse:CALibrate:CONTinue

Command

```
:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:CALibrate:
CONTinue
```

Activates the continue softkey during a TDR or TDT normalization and reference plane calibration.

Example

The following example continues a paused calibration operation.

```
10 OUTPUT 707;":TDR2:RESPONSE1:CALIBRATE:CONTINUE"
20 END
```

PRESet

Command

```
:TDR{2 | 4}:PRESet
```

Sets up the instrument for TDR or TDT measurements, based on the stimulus. This command does the following:

- Turn on TDR channels.
- If the TDT destinations are not “none,” turn on the TDT destination channels. (See the TDTDest command in this chapter).
- Set the timebase to 500 ps/div and positions the incident edge on screen.
- Turn on averaging and set best flatness (see the ACQuire subsystem).
- For all channels that are on:
  - Set the attenuation units to ratio.
  - Set the attenuation to 1:1.
  - Set the bandwidth to low (12.4 GHz). (Set high for external stimulus.)
  - Set the units to volts.
  - Set the channel scale to 100 mV/div.
  - Set the channel offset to 200 mV or -200 mV for differential stimulus.

Example

The following example presets the instrument for TDR/TDT operations.

```
10 OUTPUT 707;":TDR2:PRESET"
20 END
```

RATE

Command

```
:TDR{2 | 4}:RATE {AUTO | <rate>}
```

Sets the period of the TDR pulse generator. You should usually leave this set to AUTO unless you need to define a specific rate. In AUTO, the instrument will attempt to keep subsequent periods off screen when the timebase is changed.

`<rate>` Period to which you want to set the generator, in Hertz. You can add a suffix to indicate that the rate is in Hertz (HZ, KHZ, MHZ, and so on).

Example

The following example sets the pulse generator to 120 MHz.

```
10 OUTPUT 707;":TDR2:RATE 120 MHZ"
20 END
```

Query

```
:TDR{2 | 4}:RATE?
```

Returns the current period of the pulse generator, even when the control is set to AUTO.

Returned Format

```
[:TDR{2 | 4}:RATE]{AUTO | <rate>}<NL>
```

Example

The following example gets the current rate setting and stores it in the variable Rate$, then prints the contents of the variable to the controller's screen.

```
10 DIM Rate$[30]
20 OUTPUT 707;":TDR2:RATE?"
30 ENTER 707;Rate$
40 PRINT Rate$
50 END
```

RESPonse

Command

```
:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4} {OFF |
DIFFerential | COMMonmode | NORMalize}
```

Turns on or off the response of a TDR or TDT stimulus.

- For the 54753A, the response may be 1 or 3 and the response setting may be OFF or NORMalize.
- For the 54754:

Stimulus:	OFF	On 1 On 2 On 1 and 2	On 3 On 4 On 3 and 4	DIFFerential	COMMon mode
Response:	OFF	OFF/NORMalize		OFF DIFFerential COMMon mode	OFF DIFFerential COMMon mode

`OFF` Turns off the specified response.

`NORMalize` Turns on the normalized response of the channel.

`DIFFerential` Turns on the differential response.

`COMMonmode` Turns on the common mode response.

Example

The following example turns on common mode response on response 1 for an 54754A.

```
10 OUTPUT 707;":TDR2:RESPONSE1 COMMONMODE"
20 END
```

Query

```
:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}?
```

Returns the current response setting for the specified response setting.

Returned Format

```
[:TDR{2 | 4}:RESPonse]{1 | 2 | 3 | 4} {OFF | DIFFerential |
COMMonmode | NORMalize}<NL>
```

Example

The following example gets the current response setting for response 2, stores it in the variable Control$, then prints the contents of the variable to the controller's screen.

```
10 DIM Control$[20]
20 OUTPUT 707;":TDR2:RESPONSE2?"
30 ENTER 707;Control
40 PRINT Control
50 END
```

RESPonse:HORizontal

Command

```
:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:HORizontal
{TSOurce, MANual}
```

Specifies whether the TDR/TDT response should track the source channel's horizontal scale (TSOurce), or a user-defined scale specified with the HORizontal:POSItion and HORizontal:RANGe commands (MANual). TSOurce is the usual setting.

Example

The following example sets TDR response 1 to track the source channel's horizontal scale:

```
10 OUTPUT 707;":TDR2:RESPONSE1:HORIZONTAL TSOURCE"
20 END
```

Query

`:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:HORizontal?`

Returns the current horizontal tracking mode for the specified response.

Returned Format

```
[:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:HORizontal]{TSOurce,
MANual}<NL>
```

Example

The following example gets the current horizontal tracking mode for response 1, puts it in the variable Track$, then prints the contents of the variable to the controller's screen:

```
10 DIM Track$[20]
20 OUTPUT 707;":TDR2:RESPONSE1:HORIZONTAL?"
30 ENTER 707;Track$
40 PRINT Track$
50 END
```

RESPonse:HORizontal:POSition

Command

```
:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4):
HORizontal:POSition <position>
```

Specifies the horizontal position of the TDR/TDT response when horizontal tracking is set to manual. The position is always referenced to center screen.

`<position>` Offset from the center of the screen, in seconds.

Example

The following example sets the horizontal position for response 1 to 20 ns. This assumes that manual tracking has already been selected.

```
10 OUTPUT 707;":TDR2:RESPONSE1:HORIZONTAL:POSITION 20E9"
20 END
```

Query

```
:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:
HORizontal:POSition?
```

Returns the current horizontal position setting for the specified response.

Returned Format

```
[:TDR{2 | 4}:RESPonse{1 | 2 | 3 |
4):HORizontal:POSition]<position><NL>
```

Example

The following example gets the current horizontal position setting for response 1, puts it into the variable Pos$, then prints the contents of the variable to the controller’s screen.

```
10 DIM Pos$[20]
20 OUTPUT 707;":TDR2:RESPONSE1:HORIZONTAL:POSITION?"
30 ENTER 707;Pos$
40 PRINT Pos$
50 END
```

RESPonse:HORizontal:RANGe

Command

```
:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:
HORizontal:RANGe <range>
```

Specifies the range of the TDR/TDT response when the horizontal tracking is set to manual.

`<range>` Horizontal range in seconds.

Example

The following example sets the horizontal range for TDR response 1 to 120 ms. This assumes that manual tracking has already been selected.

```
10 OUTPUT 707;":TDR2:RESPONSE1:HORIZONTAL:RANGE 120 MS"
20 END
```

Query

```
:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:
HORizontal:RANGe?
```

Returns the current horizontal range setting for the specified response.

Returned Format

```
[:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:HORizontal:RANGe]<range><NL>
```

Example

The following example gets the current horizontal range setting for response 2, stores it in the numeric variable Range, then prints the contents of the variable to the controller's screen.

```
10 OUTPUT 707;":TDR2:RESPONSE2:HORIZONTAL:RANGE?"
20 ENTER 707;Range
30 PRINT Range
40 END
```

RESPonse:RISetime

Command

```
:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:RISetime
<risetime>
```

Sets the risetime for the normalized response. The risetime setting is limited by the timebase settings and the record length. The normalize response function allows you to change the risetime of the normalized step.

`<risetime>` Risetime setting in seconds. The Risetime function allows you to change the normalized step's risetime from a minimum of:

10 ps

or

$$\text{min} = 8 \text{ points} \times \frac{\text{time per division } ({}^{s}\!/_{\text{div}}) \times 10 \text{ divisions}}{\text{record length}}$$

whichever is greater, to a maximum of

$$\text{max} = 5 \text{ x time per division (s/div)}$$

While the TDR step's risetime applied to the system under test is fixed, the measured response has a set of mathematical operations applied to it. These mathematical operations effectively change the displayed response to the system just as if a different TDR step risetime had actually been applied. This allows you to select a risetime for TDR/TDT measurements that is close to the actual risetime used in your system. This risetime value applies to both TDR and TDT normalized channels.

Example

The following example sets the risetime for response 1 to 100 ps.

```
10 OUTPUT 707;"TDR2:RESPONSE1:RISETIME 100 PS"
20 END
```

Query

```
:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:RISetime?
```

Returns the normalized response risetime setting.

Returned Format

```
[:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:RISetime]<risetime><NL>
```

Example

The following example gets the current risetime setting and stores it in the numeric variable Risetime, then prints the contents of the variable to the controller's screen.

```
10 OUTPUT 707;":TDR2:RESPONSE1:RISETIME?"
20 ENTER 707;Risetime
30 PRINT Risetime
40 END
```

RESPonse:TDRDest

Command

```
:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:TDRDest
CHANnel<number>
```

Selects a TDR destination channel for an external stimulus. When you use an external stimulus, you must use this command to specify where the TDR channel is coming into the instrument.

A channel is valid as a TDR destination if it meets the following criteria:

- Must be an electrical channel.
- Must not have an active TDR stimulus.
- Must not be the destination of a TDT measurement.

`<number>` An integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced.

Example

The following example sets channel 2 as the TDR destination channel for response 2:

```
10 OUTPUT 707;":TDR2:RESPONSE2:TDRDEST CHANNEL2"
20 END
```

Query

```
:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:TDRDest?
```

Returns the current TDR destination channel for the selected response.

Returned Format

```
[:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:TDRDest]<channel><NL>
```

Example

The following example gets the current TDR destination channel for response 3, stores it in the variable Dest$, then prints the contents of the variable to the controller's screen:

```
10 DIM Dest$[20]
20 OUTPUT 707;":TDR2:RESPONSE3:TDRDEST?"
30 ENTER 707;Dest$
40 PRINT Dest$
50 END
```

RESPonse:TDRTDT

Command

```
:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:TDRTDT {TDR |
TDT}
```

Sets the instrument to make either TDR or TDT measurements. When the setting is TDT, both TDR and TDT measurements can be made simultaneously. You must select a TDT destination channel before selecting TDT.

The normalization and reference plane calibration determines what type of calibration to make depending on whether TDR or TDT is chosen.

Example

The following example sets the instrument to make TDR measurements only on response 1.

```
10 OUTPUT 707;":TDR2:RESPONSE1:TDRTDT TDR"
20 END
```

Query

```
:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:TDRTDT?
```

Returns the current setting of the TDR/TDT control.

Returned Format

```
[:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:TDRTDT]{TDR | TDT}<NL>
```

Example

The following example gets the current setting of the TDR/TDT control for response 4, stores it in the variable Setup$, then prints the contents of the variable to the controller's screen.

```
10 DIM Setup$[20]
20 OUTPUT 707;":TDR2:RESPONSE4:TDRTDT?"
30 ENTER 707;Setup$
40 PRINT Setup$
50 END
```

RESPonse:TDTDest

Command

```
:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:TDTDest {NONE
| CHANnel<number>}
```

Selects a destination channel for a normalization measurement. For differential and common mode stimuli, the TDT destination is implied as follows:

- The TDT destination for channel 1 is channel 3.
- The TDT destination for channel 2 is channel 4.
- The TDT destination for channel 3 is channel 1.
- The TDT destination for channel 4 is channel 2.

A channel is valid as a TDT destination if it meets the following criteria:

- Must be an electrical channel.
- Must not have an active TDR stimulus.
- Must not be the destination of another TDT measurement.
- Must not be the destination of a TDR measurement (external stimulus only).

You must select a valid TDT destination before setting the TDRTDT control to TDT.

`NONE` Deselects a channel as a TDT destination. This frees the channel to be the TDT destination of another TDR source.

`<number>` An integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced.

Example

The following example selects channel 3 as the TDT destination channel for response 4.

```
10 OUTPUT 707;":TDR2:RESPONSE4:TDTDEST CHANNEL3"
20 END
```

Query

```
:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:TDTDest?
```

Returns the current TDT destination channel for the specified response.

Returned Format

```
[:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:TDTDest]<channel><NL>
```

Example

The following example gets the TDT destination channel for response 1, puts it in the variable Dest$, then prints the contents of the variable to the controller’s screen.

```
10 DIM Dest$[20]
20 OUTPUT 707;":TDR2:RESPONSE1:TDTDEST?"
30 ENTER 707;Dest$
40 PRINT Dest$
50 END
```

RESPonse:VERTical

Command

```
:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:VERTical
{TSOurce | MANual}
```

Specifies whether the TDR/TDT response should track the source channel's vertical scale (TSOurce), or use a user-defined scale specified with the VERTical:OFFSet and VERTical:RANGe commands (MANual). TSOurce is the usual setting.

Example

The following example sets response 1 to use a user-defined vertical scale.

```
10 OUTPUT 707;":TDR2:RESPONSE1:VERTICAL MANUAL"
20 END
```

Query

```
:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:VERTical?
```

Returns the current vertical tracking mode for the specified response.

Returned Format

```
[:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:VERTical]{TSOurce |
MANual}<NL>
```

Example

The following example gets the current vertical tracking mode for response 4, puts it in the variable VertMode$, then prints the contents of the variable to the controller's screen.

```
10 DIM VertMode$[20]
20 OUTPUT 707;":TDR2:RESPONSE4:VERTICAL?"
30 ENTER 707;VertMode$
40 PRINT VertMode
50 END
```

RESPonse:VERTical:OFFSet

Command

```
:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:
VERTical:OFFSet <offset_value>
```

Sets the vertical position of the specified response when vertical tracking is set to MANual. The position is always referenced to center screen.

`<offset_value>` Offset value in volts, watts, or decibels, depending on the current channel UNITs. Suffix UNITs are ignored; only the scalar part is used ("m" in "mw").

Example

The following example sets the vertical offset to 50 mV for response 1. This assumes that the vertical tracking mode has already been set to MANual.

```
10 OUTPUT 707;":TDR2:RESPONSE1:OFFSET 50 MV"
20 END
```

Query

```
:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:
VERTical:OFFSet?
```

Returns the vertical offset for the specified response. This information is valid only when the vertical tracking mode is set to manual for the response.

Returned Format

```
[:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:VERTical:OFFSet]<volts><NL>
```

Example

The following example gets the vertical offset for response 1, stores it in the numeric variable Offset, then prints the contents of the variable to the controller's screen.

```
10 OUTPUT 707;":TDR2:RESPONSE1:VERTICAL:OFFSET?"
20 ENTER 707;Offset
30 PRINT OFFSET
40 END
```

RESPonse:VERTical:RANGe

Command

```
:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:
VERTical:RANGe <range_value>
```

Specifies the vertical range of the TDR/TDT response when the vertical tracking mode is set to MANual.

`<range_value>` Vertical range in volts, watts, or decibels, depending on the current UNITs setting and suffix supplied. (The suffix does not set the UNITs; it is ignored.)

Example

The following example sets the vertical range to 5 volts for response 1. This assumes that the vertical tracking mode has already been set to manual.

```
10 OUTPUT 707;":TDR2:RESPONSE1:VERTICAL:RANGE 5 V"
20 END
```

Query

```
:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:
VERTical:RANGe?
```

Returns the current vertical range setting for the specified response. This information is valid only when the vertical tracking mode is set to manual.

Returned Format

```
[:TDR{2 | 4}:RESPonse{1 | 2 | 3 | 4}:VERTical:RANGe]<volts><NL>
```

Example

The following example gets the vertical range setting for response 1, stores it in the numeric variable Range, then prints the contents of the variable on the controller's screen.

```
10 OUTPUT 707;":TDR2:RESPONSE1:VERTICAL:RANGE?"
20 ENTER 707;Range
30 PRINT Range
40 END
```

STIMulus

Command

```
:TDR{2 | 4}:STIMulus {OFF | ON | ON1 | ON2 |
ON1AND2 | DIFFerential | COMMonmode |EXTernal |
ON3 | ON4 | ON3AND4}
```

Turns the TDR/TDT stimulus on or off.

- For the 54753A, the stimulus may be OFF, ON, or EXTernal.
- For the 54754A in slots 1 and 2, the stimulus may be OFF, ON1, ON2, ON1AND2, DIFFerential, or COMMonmode.
- For the 54754A in slots 3 and 4, the stimulus may be OFF, ON3, ON4, ON3AND4, DIFFerential, or COMMonmode.

`OFF` Turn off the pulse generator, using the channel as a regular oscilloscope channel.

`ON, ON1, ON3` Turn on the channel 1 or channel 3 pulse generator for single-ended TDR or TDT measurements.

`ON2, ON4` Turn on the channel 2 or channel 4 pulse generator for single-ended TDR or TDT measurements.

`ON1AND2, ON3AND4` Turn on the pulse generator for channels 1 and 2 or channels 3 and 4 for simultaneous single-ended TDR or TDT measurements.

`DIFFerential` Turn on the pulse generator for channels 1 and 2 or channels 3 and 4 for differential TDR or TDT measurements.

`COMMonmode` Turn on the pulse generator for channels 1 and 2 or channels 3 and 4 for common-mode TDR or TDT measurements.

Example

The following example turns on pulse generators for channels 3 and 4 in an 54754A for single-ended TDR measurements:

```
10 OUTPUT 707;":TDR4:STIMULUS ON3AND4"
20 END
```

Query

```
:TDR{2 | 4}:STIMulus?
```

The query returns the current settings for the TDR pulse generators.

Returned Format

```
[:TDR{2 | 4}:STIMulus]{OFF | ON | ON1 | ON2 | ON1AND2 |
DIFFerential | COMMonmode |EXTernal | ON3 | ON4 | ON3AND4}<NL>
```

Example

The following example gets the current settings of the pulse generators and stores it in the variable Stim$, then prints the contents of that variable to the controller's screen.

```
10 DIM Stim$[30]
20 OUTPUT 707;":TDR4:STIMULUS?"
30 ENTER 707;Stim$
40 PRINT Stim$
50 END
```

15

Waveform Commands

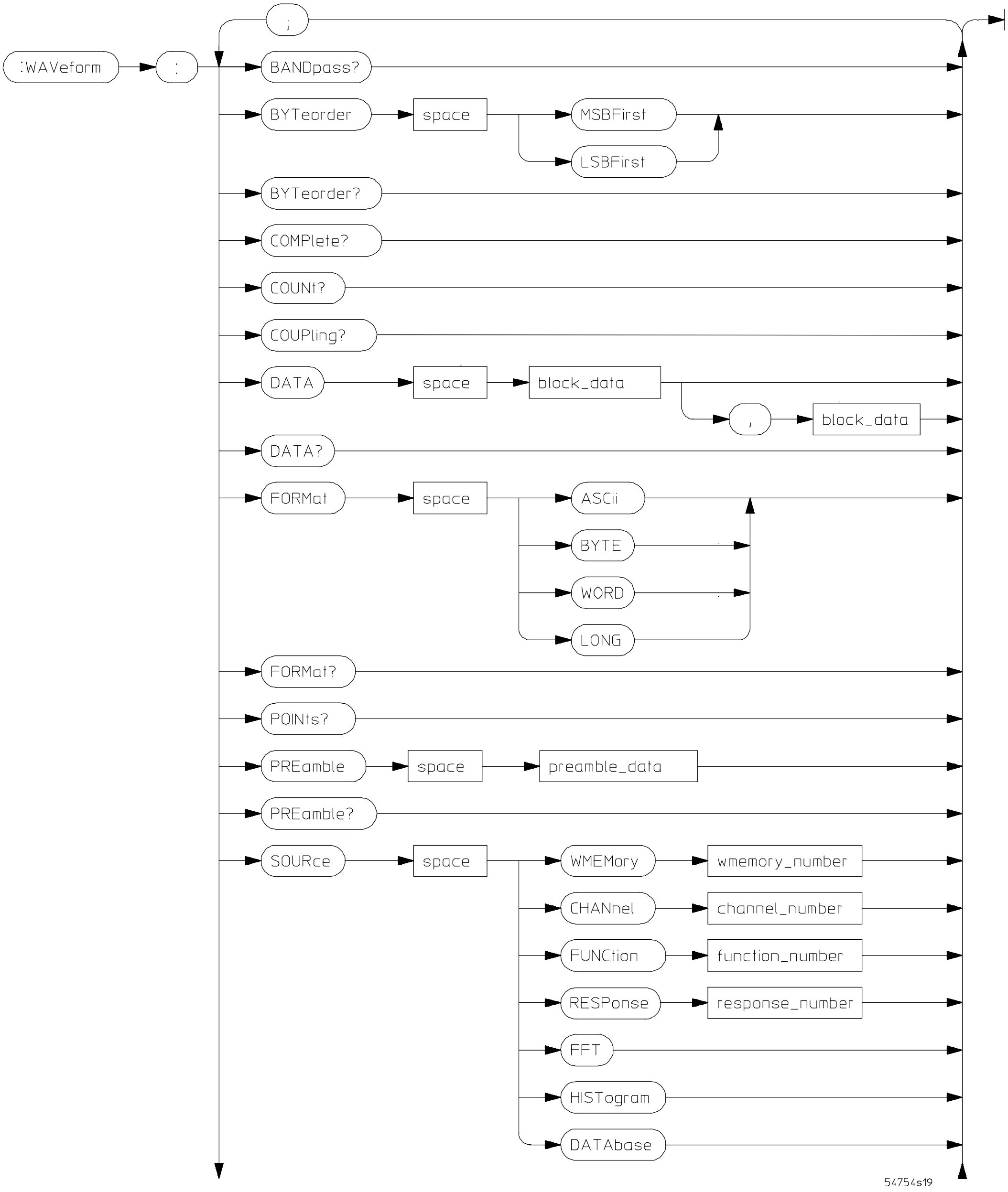
Waveform Commands

The WAVeform subsystem is used to transfer waveform data between an oscilloscope and a controller. It contains commands to set up the waveform data and to send or receive waveform records. The waveform record is contained in two portions: the preamble and the waveform data. The preamble contains the scaling and other values used to describe the data. The waveform data contains the actual data in the waveform. The preamble and waveform data must be read or sent with two separate commands: WAVeform:PREamble and WAVeform:DATA.

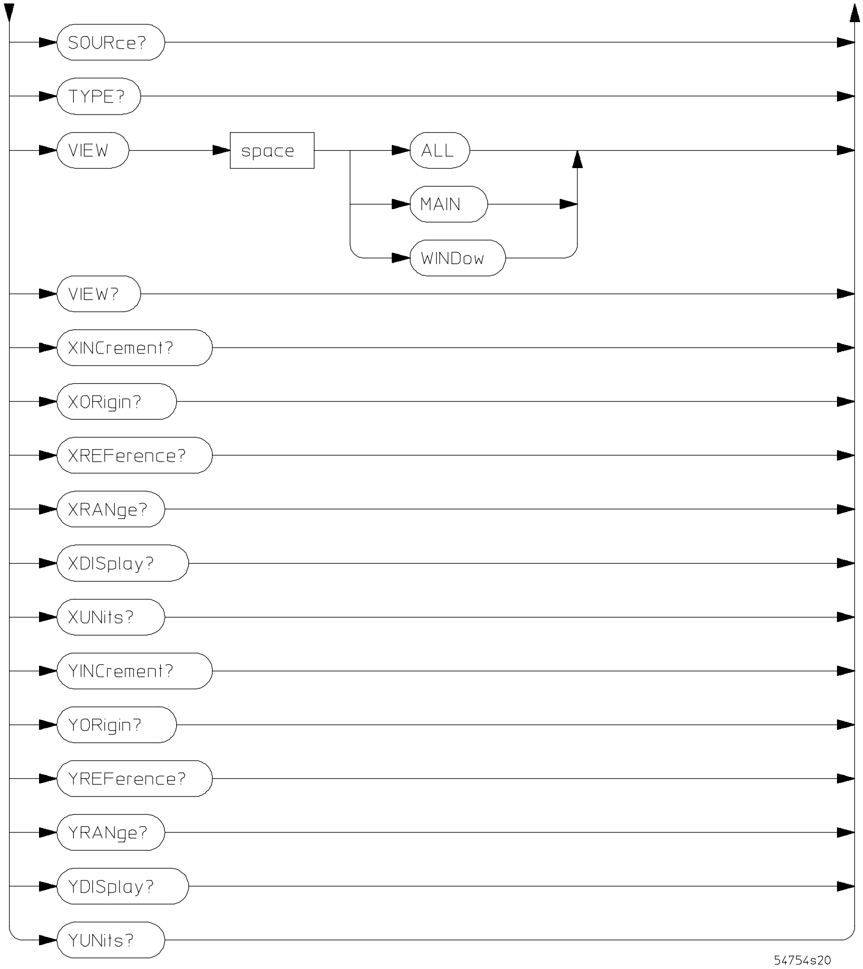
The TDR plug-in modifies the operation of the following waveform subsystem command:

- SOURce

:



Waveform Subsystem Command Syntax Diagram



Waveform Subsystem Command Syntax Diagram (continued)

SOURce

Command

```
:WAVeform:SOURce {CHANnel<number> |
FUNCtion<number> | RESPonse<number> |
WMEMory<number> | DATabase | HISTogram | FFT}
```

Selects a channel, function, TDR response, waveform memory, database, histogram, or FFT as the waveform source.

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. For TDR responses: 1, 2, 3, or 4.

Example

The following example selects TDR response 1 as the waveform source.

```
10 OUTPUT 707;":WAVEFORM:SOURCE RESPONSE1"
20 END
```

Query

```
:WAVeform:SOURce?
```

The WAVeform:SOURce query returns the currently selected waveform source.

Returned Format

```
[:WAVeform:SOURce]{CHANnel<number> | FUNCtion<number> |
RESPonse<number> | WMEMory<number> | DATabase | HISTogram |
FFT}<NL>
```

Example

The following example places the current selection for the waveform source in the string variable Selection$, then prints the contents of that variable to the controller screen.

```
10 DIM Selection$[50]      !Dimension variable
20 OUTPUT 707;":WAVEFORM:SOURCE?"
30 ENTER 707;Selection$
40 PRINT Selection$
50 END
```

16

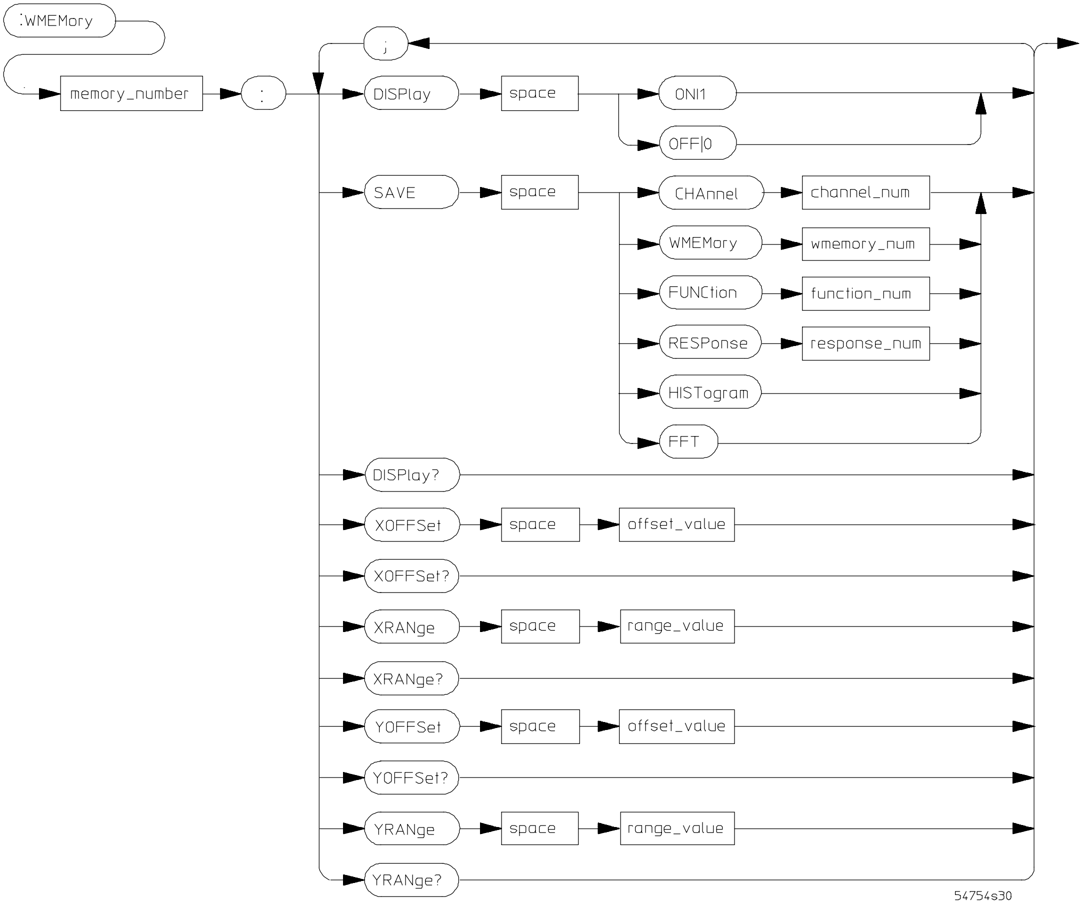
Waveform Memory Commands

Waveform Memory Commands

The Waveform Memory (:WMEMoryN) subsystem commands control the waveform save functions. They allow saving and displaying of waveforms, memories, and functions.

The TDR plug-in modifies the operation of the following waveform memory subsystem command:

- SAVE



Waveform Memory Subsystem Command Syntax Diagram

SAVe

Command

```
:WMEMory<number>:SAVE {CHANnel<number> |
FUNCtion<number> | RESPonse<number> |
WMEMory<number> | HISTogram | FFT}
```

Stores the specific channel, waveform memory, function, histogram, or FFT to the waveform memory.

`<number>` For channels: an integer, 1 through 4, indicating the slot in which the channel resides, followed by an optional A or B identifying which of two possible channels in the slot is being referenced. For functions: 1 or 2. For waveform memories (WMEMory): 1, 2, 3, or 4. Storing a waveform memory to itself does not change the memory. For TDR responses: 1, 2, 3, or 4.

Example The following example stores TDR response 2 to waveform memory 3.

```
10 OUTPUT 707;":WMEMORY3:SAVE RESPONSE2"
20 END
```

Index

A

absolute maximum voltage
  measuring, 13–36
absolute minimum voltage
  measuring, 13–38 to 13–39
accuracy
  and bandwidth filter, 8–6
acquire commands, 3–2
acquire subsystem command syntax
diagram, 3–2
ACQuire:BEST command, 3–3
acquiring data, 2–8 to 2–9
amperes, 5–9 to 5–10
amplitude measurement, 13–31
attenuation
  input, 5–5
automask
  source for, 12–10
average voltage measurement, 13–32 to 13–33

B

bandwidth limit filter, 8–6
BLANk command, 2–7

C

CALibrate:SAMPlers command, 4–4
calibration
  and marker measurements, 11–8
  canceling, 14–5
  continuing, 14–5
  TDR, 4–4, 14–5
calibration commands, 4–2
calibration subsystem command syntax
diagram, 4–2 to 4–3
calibration,and TDR/TDT setting, 14–15
canceling calibration, 14–5
captured data
  loading, 6–4
  storing, 2–11, 6–5
capturing data, 2–8 to 2–9
channel
  selecting TDR destination, 14–14
  selecting TDT destination, 14–16 to 14–17
channel subsystem command syntax
  diagram, 5–3
channel subsystem commands, 5–2
CHANnel:OFFSet command, 5–4
CHANnel:RANGe, 5–5
CHANnel:SCALe command, 5–6 to 5–7
CHANnel:TDRSkew command, 5–8
CHANnel:UNITs command, 5–9 to 5–10
command parser, 2–2
commands
  acquire subsystem, 3–2
  ACQuire:BEST, 3–3
  BLANk, 2–7
  CALibrate:SAMPlers, 4–4
  calibration subsystem, 4–2
  channel subsystem, 5–2
  CHANnel:OFFSet, 5–4
  CHANnel:RANGe, 5–5
  CHANnel:SCALe, 5–6 to 5–7
  CHANnel:TDRSkew, 5–8
  CHANnel:UNITs, 5–9 to 5–10
  COMMents, 2–8
  DIGitize, 2–8 to 2–9
  disk subsystem, 6–2
  DISK:LOAD, 6–4
  DISK:STORe, 6–5
  display subsystem, 7–2
  DISPlay:ASSign, 7–6 to 7–7
  function subystem, 8–2
  FUNCtion:ADD, 8–6
  FUNCtion:BWLimit, 8–6
  FUNCtion:DIFFerentiate, 8–7
  FUNCtion:DIVide, 8–8
  FUNCtion:FFTMagnitude, 8–8
  FUNCtion:INTegrate, 8–9
  FUNCtion:INVert, 8–10
  FUNCtion:MAGNify, 8–10
  FUNCtion:MAXimum, 8–11
  FUNCtion:MINimum, 8–12
  FUNCtion:MULTiply, 8–12
  FUNCtion:SUBTract, 8–13
  FUNCtion:VERSus, 8–14
  histogram subsystem, 9–2
  limit test subsystem, 10–2
  LTESt:SWAVeform, 10–8 to 10–9
  marker subsystem, 11–2
  MARKer:MODE, 11–6
  MARKer:PROPagation, 11–6 to 11–7
  MARKer:REFerence, 11–8
  MARKer:X1Y1source, 11–9
  MARKer:X2Y2source, 11–10
  MARKer:XUNITs, 11–11
  MARKer:YUNITs, 11–12
  mask test subsystem, 12–2
  measure subsystem, 13–2
  MEASure:DELTatime, 13–11
  MEASure:DUTycycle, 13–12
  MEASure:FALLtime, 13–13 to 13–14
  MEASure:FREQuency, 13–15
  MEASure:NWIDTh, 13–16 to 13–17
  MEASure:OVERshoot, 13–18
  MEASure:PERiod, 13–19
  MEASure:PREShoot, 13–20 to 13–21
  MEASure:PWIDth, 13–22
  MEASure:RISetime, 13–23
  MEASure:SOURce, 13–24 to 13–25
  MEASure:TEDge, 13–26
  MEASure:TMAX, 13–27 to 13–28
  MEASure:TMIN, 13–29
  MEASure:TVOLt, 13–30
  MEASure:VAMPlitude, 13–31
  MEASure:VAVerage, 13–32 to 13–33
  MEASure:VBASe, 13–34
  MEASure:VLOWer, 13–35
  MEASure:VMAX, 13–36
  MEASure:VMIDdle, 13–37
  MEASure:VMIN, 13–38 to 13–39
  MEASure:VPP, 13–40
  MEASure:VRMS, 13–41 to 13–42
  MEASure:VTIMe, 13–43
  MEASure:VTOP, 13–44
  MEASure:VUPper, 13–45 to 13–46
  MENU, 2–10
  MTESt:AMASk:SOURce, 12–10
  MTESt:SCALe:SOURce, 12–12 to 12–13
  MTESt:SWAVeform, 12–11
  root level, 2–2
  STORe:WAVEform, 2–11
  TDR subsystem, 14–2
  TDR:CALibrate, 14–5
  TDR:CALibrate:CANCel, 14–5
  TDR:CALibrate:CONTinue, 14–5
  TDR:PRESet, 14–6
  TDR:RATE, 14–7
  TDR:RESPonse, 14–8
  TDR:RESPonse:HORizontal, 14–9
  TDR:RESPonse:HORizontal:POSition, 14–10
  TDR:RESPonse:HORizontal:RANGe, 14–11
  TDR:RESPonse:RISetime, 14–12 to 14–13
  TDR:RESPonse:TDRDest, 14–14
  TDR:RESPonse:TDRTDT, 14–15

TDR:RESPonse:TDTDest, 14–16 to 14–17
TDR:RESPonse:VERTical, 14–18
TDR:RESPonse:VERTical:OFFSet, 14–19
TDR:RESPonse:VERTical:RANGe, 14–20
TDR:STIMulus, 14–21 to 14–22
VIEW, 2–12
waveform memory subsystem, 16–2
waveform subsystem, 15–2
WAVeform:SOURce, 15–5
WINDow:SOURce, 9–5
WMEMory:SAVE, 16–4
COMMents command, 2–8
common mode response, 14–8
common mode stimulus, 14–21 to 14–22
continuing calibration, 14–5
coordinates
scaling for mask test, 12–12 to 12–13

D
data
acquiring, 2–8 to 2–9
loading, 6–4
storing, 6–5
deflection
vertical, 5–5
delta time measurement, 13–11
sources for, 13–24 to 13–25
destination channel
selecting TDR, 14–14
selecting TDT, 14–16 to 14–17
dielectric constant, 11–6 to 11–7
differential response, 14–8
differential stimulus, 14–21 to 14–22
DIGitize command, 2–8 to 2–9
disk
loading waveforms from, 6–4
saving waveforms to, 6–5, 10–8 to 10–9
saving waveforms to file, 12–11
disk subsystem command syntax diagram, 6–3
disk subsystem commands, 6–2
DISK:LOAD command, 6–4
DISK:STORe command, 6–5
display
mapping waveforms to, 7–6 to 7–7
turning off, 2–7 to 2–8
turning on, 2–12
display subsystem command syntax
diagram, 7–3 to 7–5
display subystem commands, 7–2
DISPlay:ASSign command, 7–6 to 7–7
distance
and horizontal units, 11–11
and vertical units, 11–12
distance measurement, 11–6 to 11–7
duty cycle measurement, 13–12

E
external stimulus, 14–21 to 14–22

F
failure
and saving waveforms, 10–8 to 10–9
saving waveforms on, 12–11
fall time measurement, 13–13 to 13–14
files
loading waveforms, 6–4
saving limit test failure to, 10–8 to 10–9
saving waveforms to, 12–11
storing waveforms in, 6–5
filter
bandwidth limit, 8–6
flatness
step, 3–3
format
waveform data, 6–5
frequency
limiting maximum, 8–6
frequency measurements, 13–15
function subsystem command syntax diagram, 8–3 to 8–5
function subsystem commands, 8–2
FUNCtion:ADD command, 8–6
FUNCtion:BWLimit command, 8–6
FUNCtion:DIFFerentiate command, 8–7
FUNCtion:DIVide command, 8–8
FUNCtion:FFTMagnitude command, 8–8
FUNCtion:INTegrate command, 8–9
FUNCtion:INVert command, 8–10
FUNCtion:MAGNify command, 8–10
FUNCtion:MAXimum command, 8–11
FUNCtion:MINimum command, 8–12
FUNCtion:MULTiply command, 8–12
FUNCtion:SUBTract command, 8–13
FUNCtion:VERSus command, 8–14

G
gain, 5–9 to 5–10
graticule
single, 7–6 to 7–7

H
histogram markers, 11–6
histogram subsystem command syntax diagram, 9–3 to 9–4
histogram subsystem commands, 9–2
histogram window
source for, 9–5
horizontal
mask test scaling, 12–12 to 12–13
horizontal display units, 11–11
horizontal position of TDR response, 14–10
horizontal range for TDR response, 14–11
horizontal scale
histogram window, 9–5
horizontal tracking
and TDR response, 14–9

I
initializing TDR, 14–6
input scaling, 5–5 to 5–7
input voltage
measuring maximum, 13–27 to 13–28
measuring minimum, 13–29
instrument calibration, 4–4
intervals
measuring time, 13–11

L
limit test failure
saving waveforms on, 10–8 to 10–9
limit test subsystem command syntax diagram, 10–3 to 10–7
limit test subsystem commands, 10–2
limiting band width, 8–6
loading waveforms, 6–4
LTESt:SWAVeform command, 10–8 to 10–9

M
magnify
offset, 5–4
range, 5–5
scale, 5–6 to 5–7
manual markers, 11–6
mapping waveforms to display, 7–6 to 7–7
marker mode, 11–11 to 11–12
marker subsystem command syntax
diagram, 11–3 to 11–5
marker subsystem commands, 11–2
MARKer:MODE command, 11–6
MARKer:PROPagation command, 11–6 to 11–7
MARKer:REFerence command, 11–8
MARKer:X1Y1source command, 11–9
MARKer:X2Y2source command, 11–10
MARKer:XUNITs command, 11–11
MARKer:YUNITs command, 11–12
markers, 11–6, 11–8
X1 and Y1 source, 11–9
X2 and Y2 source, 11–10
mask test
saving failed waveforms, 12–11
source for automask, 12–10
mask test subsystem command syntax diagram, 12–3 to 12–9
mask test subsystem commands, 12–2
maximum input voltage measurement, 13–27 to 13–28
measure subsystem command syntax diagram, 13–3 to 13–10
measure subsystem commands, 13–2
MEASure:DELTatime command, 13–11
MEASure:DUTycycle command, 13–12
MEASure:FALLtime command, 13–13 to 13–14
MEASure:FREQuency command, 13–15
MEASure:NWIDTh command, 13–16 to 13–17
MEASure:OVERshoot command, 13–18
MEASure:PERiod command, 13–19
MEASure:PREShoot command, 13–20 to 13–21
MEASure:PWIDth command, 13–22
MEASure:RISetime command, 13–23
MEASure:SOURce command, 13–24 to 13–25
MEASure:TEDge command, 13–26
MEASure:TMAX command, 13–27 to 13–28
MEASure:TMIN command, 13–29
MEASure:TVOLt command, 13–30
MEASure:VAMPlitude command, 13–31
MEASure:VAVerage command, 13–32 to 13–33
MEASure:VBASe command, 13–34
MEASure:VLOWer command, 13–35
MEASure:VMAX command, 13–36
MEASure:VMIDdle command, 13–37
MEASure:VMIN command, 13–38 to 13–39
MEASure:VPP command, 13–40
MEASure:VRMS command, 13–41 to 13–42
MEASure:VTIMe command, 13–43
MEASure:VTOP command, 13–44
MEASure:VUPper command, 13–45 to 13–46
measurement markers, 11–6
measurements
reference position, 11–8
source selection for, 13–24 to 13–25
measuring
absolute maximum voltage, 13–36
absolute minimum voltage, 13–38 to 13–39
average voltage, 13–32 to 13–33
duty cycle, 13–12
fall time, 13–13 to 13–14
frequency, 13–15
lower threshold, 13–35
maximum input voltage, 13–27 to 13–28
minimum input voltage, 13–29
negative pulse width, 13–16 to 13–17
overshoot, 13–18
peak-to-peak voltage, 13–40
period, 13–19
positive pulse width, 13–22
preshoot, 13–20 to 13–21
rise time, 13–23
RMS voltage, 13–41 to 13–42
signal amplitude, 13–31
statistical base voltage, 13–34
statistical top voltage, 13–44
trigger-to-edge time, 13–26
trigger-to-voltage time, 13–30
upper threshold voltage, 13–45 to 13–46
voltage at time, 13–43
waveforms, 2–9
memory
storing in, 16–4
MENU command, 2–10
middle threshold
measuring, 13–37
minimum input voltage measurement, 13–29
mode
marker, 11–11 to 11–12
MS-DOS file system, 6–5
MTESt:AMASk:SOURce command, 12–10
MTESt:SCALe:SOURce command, 12–12 to 12–13
MTESt:SWAVeform command, 12–11

N
negative pulse width, 13–16 to 13–17
normalization accuracy
and bandwidth filter, 8–6
normalization and reference plane
calibration, 14–5
normalization calibration, 11–8
normalized response, 14–8
normalized TDR response risetime, 14–12 to 14–13

O
offset
magnify, 5–4
voltage setting, 5–4
ohms, 5–9 to 5–10
overshoot measurement, 13–18

P
parser, 2–2
peak-to-peak voltage
measuring, 13–40
period
and duty cycle, 13–12
measurement, 13–19
position
vertical, 5–4
vertical for TDR/TDT, 14–19

position of TDR/TDT response, 14–10
positive pulse width, 13–12
  measuring, 13–22
presetting TDR setup, 14–6
preshoot measurement, 13–20 to 13–21
propagation velocity, 11–6 to 11–7
pulse generator, 14–21 to 14–22
  rate for TDR, 14–7
pulse width
  negative, 13–16 to 13–17
  positive, 13–12

R
range
  for TDR response, 14–11
  magnifying, 5–5
  vertical of TDR/TDT, 14–20
reference plane calibration, 11–8
reference position for marker measurements, 11–8
reflect, 5–9 to 5–10
response
  horizontal tracking, 14–9
rise time measurement, 13–23
risetime
  of normalized TDR response, 14–12 to 14–13
RMS voltage measurement, 13–41 to 13–42
root level command syntax diagram, 2–3 to 2–6
root level commands, 2–2

S
saving waveforms, 12–11
  on limit test failure, 10–8 to 10–9
scaling
  for histogram window, 9–5
  input, 5–5 to 5–7
  mask test coordinates, 12–12 to 12–13
selecting
  measurement sources, 13–24 to 13–25
  slots for TDR, 14–2
  TDR or TDT, 14–15
  TDT destination channel, 14–16 to 14–17
  waveform source, 15–5
signal amplitude measurement, 13–31
simulating a band-limited signal, 8–6
slot selection, 14–2
source
  X1 and Y1 markers, 11–9
  X2 and Y2 markers, 11–10
source for histogram window, 9–5
source selection for measurements, 13–24 to 13–25
sources
  selecting, 15–5
  used for mask test scaling, 12–12 to 12–13
statistical base voltage measurement, 13–34
statistical top voltage measurement, 13–44
step flatness, 3–3
stimulus
  and destination channel, 14–14
  TDR/TDT, 14–21 to 14–22
STORe:WAVEform command, 2–11
storing captured data, 2–11
storing waveforms, 6–5
storing waveforms in memory, 16–4
syntax diagram
  acquire subsystem, 3–2
  calibration subsystem, 4–2 to 4–3
  channel subsystem, 5–3
  disk subsystem, 6–3
  display subsystem, 7–3 to 7–5
  function subsystem, 8–3 to 8–5
  histogram subsystem, 9–3 to 9–4
  limit test subsystem, 10–3 to 10–7
  marker subsystem, 11–3 to 11–5
  mask test subsystem, 12–3 to 12–9
  measure subsystem, 13–3 to 13–10
  root level command, 2–3 to 2–6
  TDR subsystem, 14–3 to 14–4
  waveform memory subsystem, 16–3
  waveform subsystem, 15–3 to 15–4

T
TDR
  and bandwidth filter, 8–6
  best flatness, 3–3
  calibration, 4–4
  destination channel, 14–14
  distance measurement markers, 11–6 to 11–7
  horizontal position, 14–10
  initializing, 14–6
  magnify offset, 5–4
  magnify range, 5–5
  magnify scale, 5–6 to 5–7
  markers, 11–6
  pulse generator rate, 14–7
  range of horizontal response, 14–11
  response settings, 14–8
  risetime of normalized response, 14–12 to 14–13
  selecting, 14–15
  slot selection, 14–2
  stimulus, 14–21 to 14–22
  units for, 11–11 to 11–12
  vertical position, 14–19
  vertical range, 14–20
  vertical tracking, 14–18
TDR subsystem command syntax diagram, 14–3 to 14–4
TDR subsystem commands, 14–2
TDR:CALibrate command, 14–5
TDR:CALibrate:CANCel command, 14–5
TDR:CALibrate:CONTinue command, 14–5
TDR:PRESet command, 14–6
TDR:RATE command, 14–7
TDR:RESPonse command, 14–8
TDR:RESPonse:HORizontal command, 14–9
TDR:RESPonse:HORizontal:POSition command, 14–10
TDR:RESPonse:HORizontal:RANGe command, 14–11
TDR:RESPonse:RISetime command, 14–12 to 14–13
TDR:RESPonse:TDRDest command, 14–14
TDR:RESPonse:TDRTDT command, 14–15
TDR:RESPonse:TDTDest command, 14–16 to 14–17
TDR:RESPonse:VERTical command, 14–18
TDR:RESPonse:VERTical:OFFSet command, 14–19
TDR:RESPonse:VERTical:RANGe command, 14–20
TDR:STIMulus command, 14–21 to 14–22

TDT
  selecting, 14–15
  selecting destination channel, 14–16 to 14–17
  stimulus, 14–21 to 14–22
thresholds
  and fall time, 13–13 to 13–14
  and frequency measurements, 13–15
  and period measurement, 13–19
  and rise time measurement, 13–23
  measuring lower, 13–35
  measuring middle, 13–37
  measuring upper, 13–45 to 13–46
throughput, 3–3
time
  and voltage measurement, 13–43
time interval
  measurements, 13–11, 13–23
  measuring trigger-to-voltage, 13–30
  trigger-to-edge, 13–26
trigger reference, 11–8
trigger-to-edge measurement, 13–26
trigger-to-voltage time measurement, 13–30
turning off display, 2–7 to 2–8
turning on display, 2–12
types of markers, 11–6

U
units
  and automask source, 12–10
  horizontal display, 11–11
  vertical, 5–9 to 5–10
  vertical display, 11–12
user-defined scale
  for TDR horizontal tracking, 14–9
  for vertical tracking, 14–18

V
velocity
  of propagation, 11–6 to 11–7
vertical
  mask test scaling, 12–12 to 12–13
vertical deflection, 5–5
vertical display units, 11–12
vertical position
  for TDR tracking, 14–19
vertical position setting, 5–4
vertical range
  of TDR/TDT, 14–20
vertical scale
  histogram window, 9–5
vertical tracking
  of TDR/TDT, 14–18
vertical units, 5–9 to 5–10
VIEW command, 2–12
voltage
  measuring absolute maximum, 13–36
  measuring absolute minimum, 13–38 to 13–39
  measuring at time, 13–43
  measuring average, 13–32 to 13–33
  measuring lower threshold, 13–35
  measuring maximum, 13–27 to 13–28
  measuring middle threshold, 13–37
  measuring minimum, 13–29
  measuring peak-to-peak, 13–40
  measuring RMS, 13–41 to 13–42
  measuring statistical base, 13–34
  measuring statistical top, 13–44
  measuring upper threshold, 13–45 to 13–46
  offset, 5–4
voltage measurement, 13–31
volts per division setting, 5–5

W
watts, 5–9 to 5–10
waveform markers, 11–6
waveform memory
  storing in, 16–4
waveform memory subsystem command syntax diagram, 16–3
waveform memory subsystem commands, 16–2
waveform subsystem command, 15–2
waveform subsystem command syntax diagram, 15–3 to 15–4
WAVeform:SOURce command, 15–5
waveforms
  capturing, 2–8 to 2–9
  displaying, 2–12
  hiding, 2–7 to 2–8
  loading, 6–4
  mapping to display, 7–6 to 7–7
  measuring, 2–9
  saving limit test failure, 10–8 to 10–9
  saving on failure, 12–11
  selecting source, 15–5
  storing, 6–5
width
  of overshoot, 13–18
  of positive pulse, 13–22
width measurement
  of negative pulse, 13–16 to 13–17
WINDow:SOURce commands, 9–5
WMEMory:SAVE command, 16–4