Programmer Manual

Tektronix

2790 Series
Spectrum Analyzers

070-8643-00

Please check for change information at the rear of this manual.

First Printing AUG 1992

Instrument Serial Numbers

Each instrument manufactured by Tektronix has a serial number on a panel insert or tag, or stamped on the chassis. The first letter in the serial number designates the country of manufacture. The last five digits of the serial number are assigned sequentially and are unique to each instrument. Those manufactured in the United States have six unique digits. The country of manufacture is identified as follows:

B010000	Tektronix, Inc., Beaverton, Oregon, USA
E200000	Tektronix United Kingdom, Ltd., London
J300000	Sony/Tektronix, Japan
H700000	Tektronix Holland, NV, Heerenveen, The Netherlands

Instruments manufactured for Tektronix by external vendors outside the United States are assigned a two digit alpha code to identify the country of manufacture (e.g., JP for Japan, HK for Hong Kong, IL for Israel, etc.).

Tektronix, Inc., P.O. Box 500, Beaverton, OR 97077

Printed in U.S.A.

WARRANTY

Tektronix warrants that this product will be free from defects in materials and workmanship for a period of one (1) year from the date of shipment. If any such product proves defective during this warranty period, Tektronix, at its option, either will repair the defective product without charge for parts and labor, or will provide a replacement in exchange for the defective product.

In order to obtain service under this warranty, Customer must notify Tektronix of the defect before the expiration of the warranty period and make suitable arrangements for the performance of service. Customer shall be responsible for packaging and shipping the defective product to the service center designated by Tektronix, with shipping charges prepaid. Tektronix shall pay for the return of the product to Customer if the shipment is to a location within the country in which the Tektronix service center is located. Customer shall be responsible for paying all shipping charges, duties, taxes, and any other charges for products returned to any other locations.

This warranty shall not apply to any defect, failure or damage caused by improper use or improper or inadequate maintenance and care. Tektronix shall not be obligated to furnish service under this warranty a) to repair damage resulting from attempts by personnel other than Tektronix representatives to install, repair or service the product; b) to repair damage resulting from improper use or connection to incompatible equipment; or c) to service a product that has been modified or integrated with other products when the effect of such modification or integration increases the time or difficulty of servicing the product.

THIS WARRANTY IS GIVEN BY TEKTRONIX WITH RESPECT TO THIS PRODUCT IN LIEU OF ANY OTHER WARRANTIES, EXPRESSED OR IMPLIED. TEKTRONIX AND ITS VENDORS DISCLAIM ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. TEKTRONIX' RESPONSIBILITY TO REPAIR OR REPLACE DEFECTIVE PRODUCTS IS THE SOLE AND EXCLUSIVE REMEDY PROVIDED TO THE CUSTOMER FOR BREACH OF THIS WARRANTY. TEKTRONIX AND ITS VENDORS WILL NOT BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IRRESPECTIVE OF WHETHER TEKTRONIX OR THE VENDOR HAS ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

Welcome

This is the Programmer Manual for the 2790 series of spectrum analyzers. This manual provides information on operating your spectrum analyzer using the General Purpose Interface Bus (GPIB) interface.

Related Manuals

Other documentation for the spectrum analyzer includes the following:

- The *User Manual* describes the operation of the spectrum analyzer.

2792	070−8631−00
2794	070−8632−00
2795	070−8633−00
2797	070−8634−00

- The *Service Manual* provides information to maintain and service your Spectrum Analyzer. Each service manual set comprises two volumes.

2792	070−8636−00	070−8639−00
2794	070−8637−00	070−8640−00
2795	070−8638−00	070−8641−00
2797	070−8639−00	070−8642−00

Contents

Getting Started

Syntax and Commands

Status & Events

Programming Examples

Appendices

Index

List of Figures

List of Tables

Getting Started

This section provides information you will need to get your spectrum analyzer set up for use with a GPIB controller.

Tektronix 2790 series Spectrum Analyzers provide remote control and automated spectrum data acquisition and analysis through the rear panel GPIB interface connector. With this interface, most functions can be controlled remotely. Waveform processing functions are added to do some spectrum analysis locally.

The IEEE STD 488 General Purpose Interface Bus (GPIB) port in the spectrum analyzer rear panel allows it to be used with a wide variety of systems and controllers; the instrument follows the Tektronix Interface Standard for GPIB Codes, Formats, Conventions, and Features Version 81.1. This standard promotes ease of operation and makes this spectrum analyzer compatible with other Tektronix instruments and, as much as possible, with GPIB instruments from other manufacturers.

NOTE

If you plan to use programs written for a Tektronix 49X series spectrum analyzer with your 2790 series instrument, be sure to read Appendix A: 49X Compatibility *which lists important differences between the two instrument types.*

Setting Up Remote Communications

This section tells you how to set up your spectrum analyzer for remote communications. These instructions assume that you have reviewed the User manual for your spectrum analyzer and are familiar with its use and operation. You should also refer to the documentation supplied with your controller for information that is controller-specific.

GPIB Requirements

Observe these rules when you use your spectrum analyzer with a GPIB network:

- Assign a unique device address to each device on the bus. No two devices can share the same device address.
- Do not connect more than 15 devices to any one bus.
- Connect one device for every 2 meters (6 feet) of cable used.
- Do not use more than 20 meters (65 feet) of cable to connect devices to a bus.

- Turn on at least two-thirds of the devices on the network while using the network.
- Do not connect GPIB devices in parallel or loop configurations.

Connecting to a System

Connect your spectrum analyzer directly to a GPIB system using the cable available as an optional accessory (contact your local Tektronix Field Office or representative for ordering information).

The IEEE STD 488 PORT is shown in Figure 1-1. Printed under the IEEE STD 488 PORT are the Interface Function abbreviations and the codes indicating their use in the instrument (refer to "IEEE 488 Functions" later in this section for an explanation of each function). The "E2" notation following the function label indicates that three-state drivers are used — rather than open-collector drivers — because of the high-speed operation of the instrument.

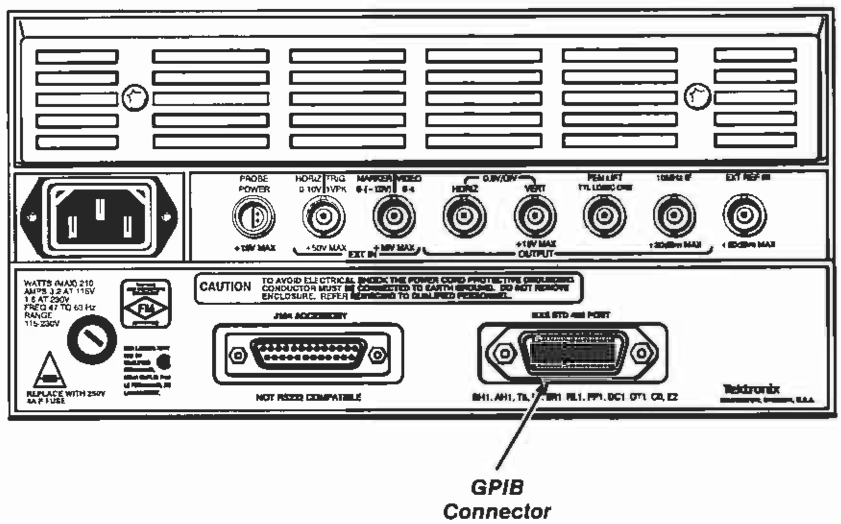


Figure 1-1: The Rear-Panel IEEE STD 488 PORT (GPIB)

The instrument start-up procedure is provided in the User Manual. Refer to the user manual for instructions on how to begin operating the instrument. If you do not have the manual, refer to your local Tektronix Field Office or representative for manual ordering information.

NOTE

An internal switch setting causes the spectrum analyzer to assert SRQ when power is first applied. This SRQ requires immediate action by some controllers, so this default setting is not recommended for those controllers. Because changing the switch requires that the instrument cover be removed, refer this task to qualified service personnel.

The initial power-on setting of all programmable functions is restored by the INIt command. Refer to the "Command Description" part of the tabbed *Syntax and Commands* section of this manual for information about the power up setting of individual commands or functions.

Setting the GPIB Address

You must set the GPIB address of the instrument set before communication can take place. If you are not already familiar with the spectrum analyzer front panel, the external interface keys and indicators are briefly discussed following this section.

The GPIB address is set by pressing the EXT INTFC CONFIG GPIB menu key sequence. Then, press the Set GPIB Address softkey and use the keypad to enter the value of the GPIB address (refer to Table 1-2). The instrument's primary address can be from 0 to 31.

The address transmitted by the controller is seven bits wide. The first five bits are the primary address and the last two bits determine whether it is a listen address (32 + primary address) or talk address (64 + primary address). For example; 0100010 is primary address 2, a listener, and 1000010 is primary address 2, a talker. Secondary addresses (when both bits 6 and 7 are set) are not used by the spectrum analyzer, so are ignored.

NOTE

Do not use address 0 with Tektronix 4050-Series controllers; they reserve this address for themselves.

Selecting a primary address of 31 logically removes the spectrum analyzer from the bus; it does not respond to any GPIB address, but remains both unlistened and untalked.

The primary address is stored in nonvolatile RAM and is not affected by the front panel PRESET button, the INIT command, or the settings recall function.

Default Settings

If you do not make any selection, or if a nonvolatile RAM checksum error is detected, the settings listed in Table 1-1 apply:

Table 1-1: Default Instrument Settings

Setting	Value
GPIB Settings	
GPIB Primary Address	1
Message Terminator	EOI
Talk only	off
Listen only	off
49X Compatibility Mode	off
Plotter Settings	
Plotter	HC100
Offset	0

Table 1-2: BUS Addresses

	Primary Address	Listen Address	Talk Address
16 8 4 2 1			
0 0 0 0 0	0	32	64
0 0 0 0 1	1	33	65
0 0 0 1 0	2	34	66
0 0 0 1 1	3	35	67
0 0 1 0 0	4	36	68
0 0 1 0 1	5	37	69
0 0 1 1 0	6	38	70
0 0 1 1 1	7	39	71
0 1 0 0 0	8	40	72
0 1 0 0 1	9	41	73
0 1 0 1 0	10	42	74
0 1 0 1 1	11	43	75
0 1 1 0 0	12	44	76
0 1 1 0 1	13	45	77
0 1 1 1 0	14	46	78
0 1 1 1 1	15	47	79
1 0 0 0 0	16	48	80
1 0 0 0 1	17	49	81
1 0 0 1 0	18	50	82
1 0 0 1 1	19	51	83
1 0 1 0 0	20	52	84
1 0 1 0 1	21	53	85
1 0 1 1 0	22	54	86
1 0 1 1 1	23	55	87
1 1 0 0 0	24	56	88
1 1 0 0 1	25	57	89
1 1 0 1 0	26	58	90
1 1 0 1 1	27	59	91
1 1 1 0 0	28	60	92
1 1 1 0 1	29	61	93
1 1 1 1 0	30	62	94
1 1 1 1 1	31	UNL	UNT

Selecting the Bus Terminator (LF or EOI)

The terminator for messages on the bus is set by pressing the EXT INTFC CONFIG GPIB Operation Mode menu key sequence. Use this menu to select Terminator lf/**EOI**.

If **LF** is selected, the spectrum analyzer interprets either the data byte LF or the end message (EOI asserted concurrently with a data byte) as the end of a message. If **EOI** is selected, the spectrum analyzer interprets the end message (EOI asserted as well as a data byte) as the end of a message.

This function also selects the output terminator. Set to **LF** or EOI, the instrument adds CR and LF (with EOI asserted as well as LF) after the last byte of the message. Set to **EOI**, the instrument asserts EOI concurrently with the last byte of the message. Select EOI for Tektronix controllers. A change takes immediate effect.

Figure 1-2 shows the effect of the bus terminator for both input and output.

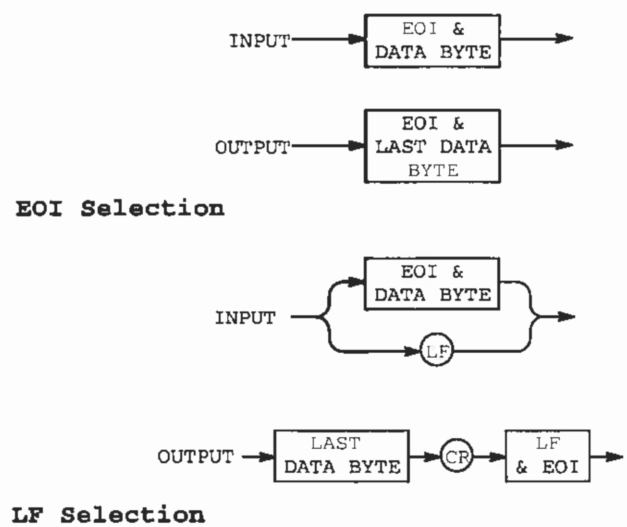


Figure 1-2: Effect of Message Terminator for Input And Output

Selecting the TALK ONLY and LISTEN ONLY Modes

To select the talk-only or listen-only mode, press the EXT INTFC CONFIG GPIB Operation Mode menu key sequence. Then, use the soft keys to select the Talk Only on/**OFF** and Listen Only on/**OFF**.

Both talk-only and listen-only modes must be **OFF** when the spectrum analyzer is used with any controller. In contrast, both modes must be **ON** to allow spectrum analyzer output to be exchanged with a storage device without the need of a controller. The talk-only mode must be **ON** to allow spectrum analyzer output to be sent to a plotter. With the listen-only mode **ON**, information sent to a storage device can be fed back to the spectrum analyzer.

In 49X compatibility mode only: when RETURN TO LOCAL is pressed while in "talk only" mode the settings, or settings and waveform (depending upon the setting of an internal memory board switch), are sent over the GPIB.

The External Interface Keys and Indicators

Keys and indicators related to GPIB operation of your spectrum analyzer are shown in Figure 1-3. Note that the front panel shown may differ from your spectrum analyzer in some details related to instrument models and options.

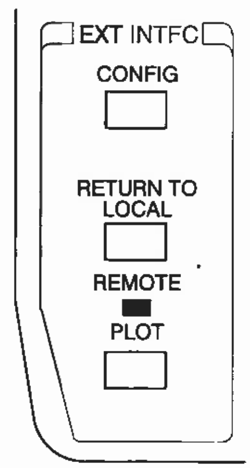


Figure 1-3: External Interface (EXT INTFC) Keys and Indicators

RETURN TO LOCAL and REMOTE

The remote indicator is lit when the spectrum analyzer is under control of the GPIB controller. While under remote control, most other front-panel controls, keys, and menu items are not active; indicators will still reflect the current state of all front-panel functions.

The REMOTE indicator is not lit when the instrument is under local, operator control. While under local control, the instrument does not execute GPIB messages that would conflict with front-panel controls, and it does not accept the CURVE input command.

When the instrument is under remote control and RETURN TO LOCAL is pressed, local control is restored to the operator unless the controller prevents this with the local lockout message. Programmable functions do not change when switching from remote to local control.

NOTE

In 49X compatibility mode only: when RETURN TO LOCAL is pressed while in "talk only" mode, the settings or settings and waveform (depending upon the setting of an internal memory board switch) are sent over the GPIB.

PLOT

Press this key when the spectrum analyzer is in the talk-only mode for the instrument to send the appropriate commands over the GPIB to a plotter. The plotter must be in the listen-only mode and connected to the bus (see the Talker and Listener descriptions later in this section).

The spectrum analyzer display — waveform, marker(s), graticule, and CRT readout — can be recreated on a Tektronix HC100, 4662 or 4662 Option 31 Plotter (or a 4663 in the 4662 emulation mode) as well as a Hewlett-Packard HP7470A or HP7475A plotter.

Select the plotter type with the EXT INTFC CONFIG menu. A bus controller is not required.

Send SRQ

The Send SRQ (send service request) menu item notifies the controller to listen and/or respond.

For example, if the controller put instructions on the screen, in the TEXt LONg mode, to set up test equipment, etc., the last line of the instructions might say "PRESS Send SRQ KEY WHEN READY". This would instruct the controller to go on to the next step. An SRQ will only be issued if RQS is on.

GPIB CRT Readout

A character appears at the end of the upper line of the lower CRT readout when the spectrum analyzer is talking (T), or listening (L); see Figure 1-4. Two characters will appear in this location if the instrument is talking or listening and also requesting service (S).

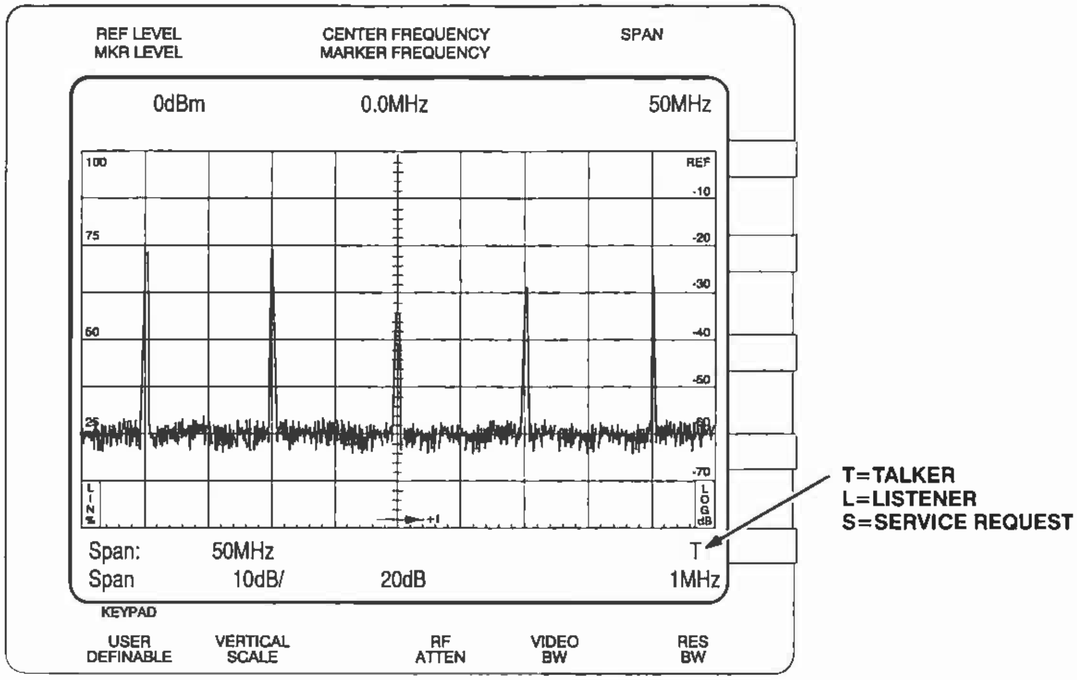


Figure 1-4: Status of Active GPIB Functions

IEEE 488 Functions

The spectrum analyzer is compatible with IEEE STD 488–1978. The connector and the signal levels at the connector follow the specifications in the IEEE 488 standard. Table 1-3 lists interface capabilities as defined in the standard.

Table 1-3: Programmable Spectrum Analyzer IEEE 488 Interface Functions

Function	Implemented As
Source Handshake	SH1
Acceptor Handshake	AH1
Talker	T5
Listener	L3
Service Request	SR1
Remote Local	RL1
Parallel Poll	PP1
Device Clear	DC1
Device Trigger	DT1

Table 1-3: Programmable Spectrum Analyzer IEEE 488 Interface Functions (Cont.)

Function	Implemented As
Controller	C0

Source Handshake (SH1)

The spectrum analyzer has complete capability to transfer messages to other devices on the bus. Although tri-state drivers are used on the data lines, T1 (DAV delay for data setting) is greater than 2 µs.

Acceptor Handshake (AH1)

The spectrum analyzer has complete capability to receive messages on the bus.

Talker (T5)

The spectrum analyzer has the complete talker function including serial poll; it unaddresses as a talker when addressed as a listener. The instrument operates in a simple system in a talk-only mode if the Talk Only on/**OFF** menu item is set to **ON**.

Listener (L3)

The spectrum analyzer has the complete listener function; unaddresses as a listener when addressed as a talker. The instrument operates in a simple system in a listen-only mode if the Listen Only on/**OFF** menu item is set to **ON**.

Service Request (SR1)

The spectrum analyzer has the complete service request function; asserts SRQ (service request) for the conditions indicated under "Status Byte" in the tabbed *Status and Events* section of this manual. The spectrum analyzer reports the corresponding status when polled.

Remote/Local (RL1)

The spectrum analyzer has the complete remote/local function. The front-panel RETURN TO LOCAL key returns the instrument from remote to local control unless the LLO (local lockout) message was previously received. The GTL (go to local) message also returns the instrument from remote to local control. Refer to the discussion under "Status Byte" in the tabbed section *Status and Events* of this manual for the effect of busy status on remote/local transitions.

The current value of programmable functions is maintained when switching from local to remote control.

The spectrum analyzer must be under remote control to begin executing device-dependent messages that change the state of local controls or to load data into digital storage.

Once begun, execution continues even if REN (remote enable) goes false. The spectrum analyzer changes settings for which there is no local control and outputs data while under local control.

Parallel Poll (PP1)

The spectrum analyzer responds to a parallel poll to indicate if service is requested.

Device Clear (DC1)

The spectrum analyzer responds to the DCL (device clear) and SDC (selected device clear) interface messages by resetting its input and output buffers to restart bus communications.

When these messages are executed, they clear outstanding SRQ conditions and set the ERR query response to zero. Power-up status, if selected internally, is an exception; see "Status Byte" in the tabbed section *Status and Events* of this manual for more on power-up status and for the effect of busy status on the execution of DCL and SDC.

Device Trigger (DT1)

The spectrum analyzer DT (device trigger) function allows the GET (group execute trigger) message to cause the instrument to stop the current sweep and rearm for the new sweep.

The new sweep begins when the triggering conditions are met. The DT command must be on and the instrument must be in the REMOTE mode for GET to have any effect.

Controller (C0)

The spectrum analyzer does not act as a controller.

Command Syntax

You can control the spectrum analyzer through the GPIB interface using commands and queries. This section describes the syntax these commands and queries use. It also describes the conventions the spectrum analyzer uses to process them. The next two parts of this section, "Functional Command Groups" and "Command Descriptions" list and discuss the actual commands and queries used with the spectrum analyzer.

You transmit commands to the spectrum analyzer using the enhanced American Standard Code for Information Interchange (ASCII) character encoding. Appendix C contains a chart of the ASCII character set.

The goal of the programmable spectrum analyzer device-dependent message structure is to enhance compatibility with a variety of GPIB systems, yet be simple and obvious to use.

To make spectrum analyzer messages easy to understand and write, ordinary engineering terms are used. Message codes (mnemonics) are chosen to be short, yet remind you of their function.

For example, to set the instrument center frequency to 500.000 MHz, the message `FREQ 500.000 MHZ` could be sent over the bus after the instrument has been addressed as a listener.

Variations on this message are allowed. An examples would be to make it shorter or send the frequency in scientific notation. The example above shows the logical and simple format of spectrum analyzer messages.

Syntax Diagrams

Spectrum analyzer messages are presented in this manual in syntax diagrams that show the sequence of elements transferred over the bus. Each element is enclosed in a circle, oval, or box.

Circles or ovals contain the mnemonics for literal elements; that is, a character or string of characters that must be sent exactly as shown. Because most mnemonics may be shortened, the required characters in command and query literal elements (that is, the first three characters of the element) are shown in upper-case type. Although mnemonics may be shown all upper case, the spectrum analyzer accepts either upper-case or lower-case ASCII characters. Query responses are typically shown exactly as they will be returned.

Boxes are used for defined elements and contain a name that stands for the element defined elsewhere. NUM is such an name and is defined under Numbers later in this section.

Elements of the syntax diagram are connected by arrows that show the possible paths through the diagram (that is, the sequence in which elements must be transferred).

Parallel paths mean that one, and only one, of the paths must be followed; while a path around an element or group of elements indicates an optional skip.

Arrows indicate the direction that must be followed (usually the flow is to the right; but, if an element may be repeated, an arrow returns from the right to the left of the element).

Figure 2-1 shows some typical syntax paths.

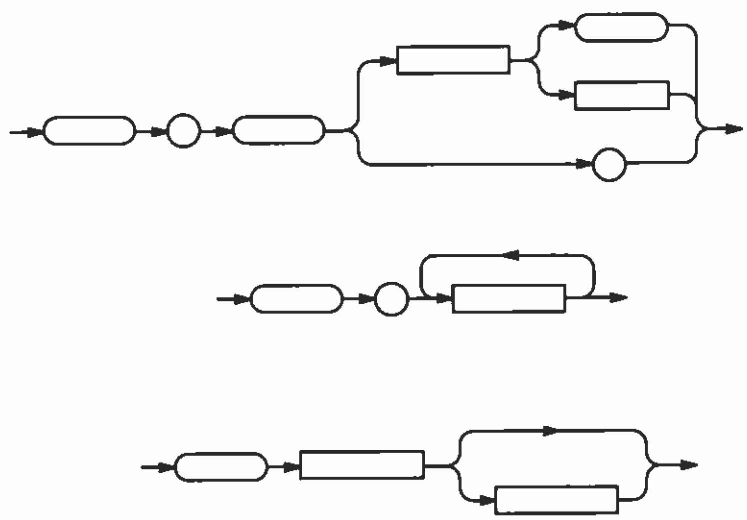


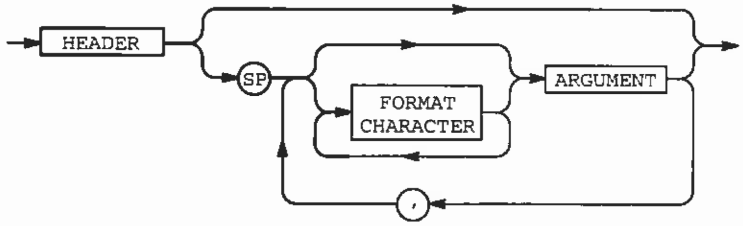
Figure 2-1: Typical Syntax Diagrams

Spectrum Analyzer Input Messages

Input Message Format

A remote control message to the spectrum analyzer comprises one or more message units of two types. The message units either consist of commands that the spectrum analyzer inputs as control or measurement data, or they consist of queries that request the spectrum analyzer to output data.

One or more message units can be transmitted as a message to the spectrum analyzer. Message units contain ASCII characters (binary may also be used for data). The spectrum analyzer accepts either upper-case or lower-case characters for the mnemonics shown in the syntax diagrams. Command and query input can be shortened to the initial three characters, unless noted otherwise in *Command Descriptions*.



Message Unit Delimiter (;)

Message units are separated by a semicolon (;). A semicolon is optional following the last message unit.

Message Terminator (TERM)

The end-of-message terminator may be either the END message (EOI asserted concurrently with the last data byte), or the ASCII code for line feed (LF) sent as the last data byte.

The active terminator is selected using the EXT INTFC CONFIG and Operation Mode menus.

Format Characters

Format characters may be inserted at many points to make the message more intelligible, but are required only if they are included as a literal element (that is, in circles or ovals) with no bypass.

Allowable format characters are SP (space), CR (carriage return), and LF (line feed — unless the end of message terminator is set for LF), as well as all other ASCII control characters and comma (,). At some points in a message, the spectrum analyzer may accept other non-alphanumeric characters, such as quotation marks (").

Input Buffering and Execution

The spectrum analyzer buffers each message it receives with a capacity that exceeds that required for the SET? response. The spectrum analyzer waits until the end of the message to decode and execute it.

A command error in any part of a message prevents its execution. When the instrument is under local control, commands that would conflict with local control are ignored (see the Remote/Local discussion under "IEEE 488 Functions" in the *Getting Started* section of this manual).

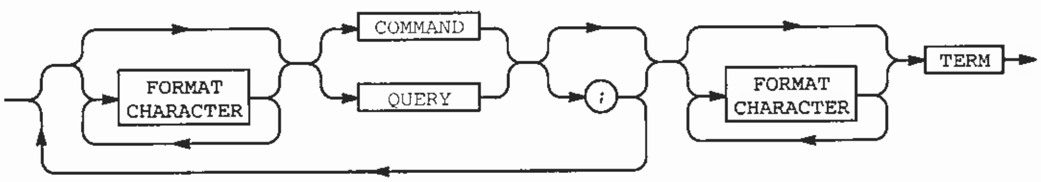
If the message contains multiple message units, none are acted on until the instrument sees the end-of-message terminator.

When the spectrum analyzer sees the terminator, it executes the commands in the message in the order they were received. The instrument remains busy until it is done executing the commands in the buffer, unless the process is stopped by the DCL (Device Clear) or SDC (Selected Device Clear) interface messages. While busy, further input is not accepted (see "Status Byte" in the *Status and Events* section of this manual for more on busy status). Output, if requested, is begun only after the entire input message is executed.

Because display (measurement) data input and output and waveform processing share the same buffer, conflicts can arise. This is noted as an interaction under the "CURve" command in the *Command Descriptions* section of this tabbed section. This interaction is further discussed under "Multiple Use of Display Buffer" in the tabbed section *Programming Examples*.

Command Format

A command message unit either sets an operating mode or parameter, or it transfers data to the instrument. The command format to set a mode or parameter includes the following possible paths.



The general command format for display data transfers is somewhat more complicated; *Command Descriptions* later in this section gives the necessary syntax for each command.

Header

Header elements are mnemonics that represent a function; for example, FREQ for center frequency and CURVE for the display trace.

Header Delimiter (SP)

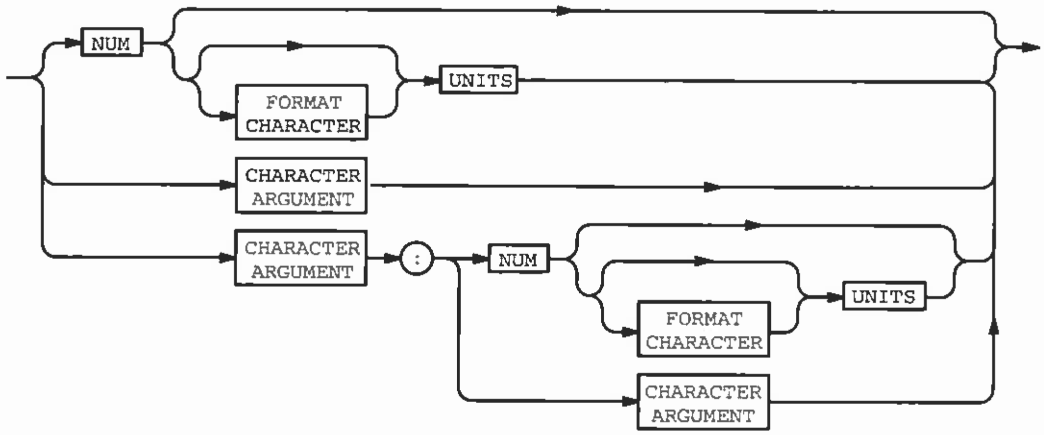
A space (SP) must separate the header from any argument(s).

Argument Delimiter (,)

A comma (,) must separate individual arguments, and a colon (:) must separate link arguments.

Argument Format

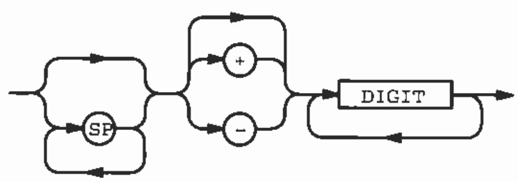
The following diagram illustrates that arguments following the header may be a number, a group of characters, or either a number or a group of characters linked to another argument.



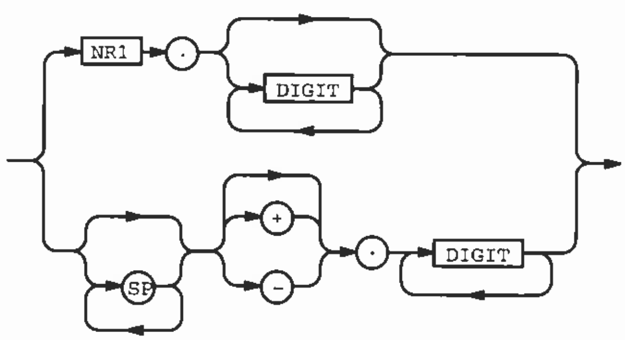
Numbers

The defined element NUM is a decimal number in any of three formats; NR1, NR2, or NR3.

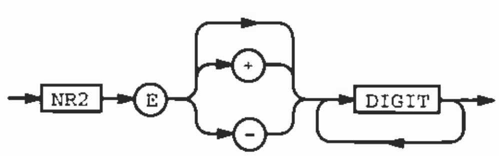
NR1 is an integer (no decimal point).



NR2 is a floating point number (decimal point required).



NR3 is a floating point number in scientific notation.



NUM arguments may serve two functions. The first is to select the value of a continuous function (for example, the center frequency with FREQ). In this case, if NUM exceeds the range of the function, the spectrum analyzer does not execute the command, but issues an error message (see the command description for "POInt," later in this tabbed section, for an exception). Numbers within the range are rounded.

The second function of a NUM argument is to substitute for character arguments in ON/OFF or mode selection. In this case, if NUM exceeds the selection range, it is rounded to the nearest end of the range. No error message is issued. Numbers within the range are rounded.

Unless otherwise stated, the values for the NUM argument are

- 0 = OFF
- $<+0.5$ are rounded to 0
- $\geq +0.5$ are rounded to 1
- 1 = ON

Units

The spectrum analyzer accepts arguments in engineering notation; that is, engineering units may be appended to a number argument. The instrument treats the combined number and units as scientific notation where the first letter of the units element represents a power of 10. K=1E+3, G=1E+9, and M=1E−3 or M=1E+6 (the value of M depends on the function, where MSEC stands for 1E−3 (milliseconds) in the TIME (time/div) command, and MHZ stands for 1E+6 (megahertz) in the SPAN (span/div) command).

Only the first letter of the units element is of importance; the rest of the units element (that is, SEC or HZ) does not contribute to the value of the command argument and can be omitted.

NOTE

This does not apply to the dBm and dBmV units used with the RLUNITS, REFLVL, and MAXPWR commands, where all letters must be used to avoid an ambiguity error.

Although more than one format character may precede the units, only a space (SP) is shown in the command syntax diagrams in this manual.

In most cases (other than RLUNITS, REFLVL, and MAXPWR), if no units element (terminator) is sent with the number, one of the following will be implied as the default condition depending on usage:

- volts
- dB
- seconds
- Hz

Character Argument

Arguments may be either words or mnemonics. ON and OFF, for instance, are arguments for the commands that correspond to spectrum analyzer front-panel keys like SAVE A.

Link Argument

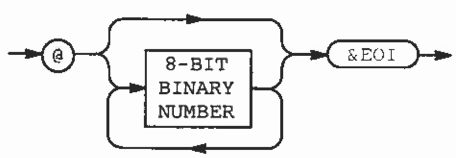
The bottom path in the argument diagram combines both character and number arguments in a link argument. The link is the colon (:), which delimits the first and second arguments. For example, the VRTDSP (vertical display) command employs link arguments to make scale factors available.

String Argument

A string argument is used when a message is to be displayed on a plotter or display unit for human interpretation, as with the RDOUT command. The characters are enclosed in quotes to delimit them as a string argument.

End Block

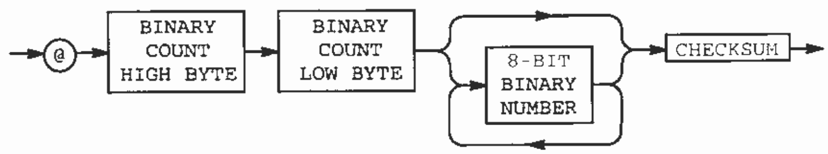
End block binary is a sequence of binary numbers that is preceded by the ASCII code for at (@); EOI must be asserted concurrently with the last data byte. The end block can only be the last data type in a message.



Binary Block

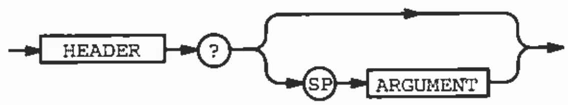
Binary block is a sequence of binary numbers that is preceded by the ASCII code for percent (%) and a two-byte binary integer representing the number of binary numbers plus one (the extra byte is the checksum) and followed by the checksum.

The checksum is the 2's-complement of the modulo-256 sum of all preceding bytes except the %. Thus, the modulo-256 sum of all bytes except the % should equal zero to provide an error-check of the binary block transfer.



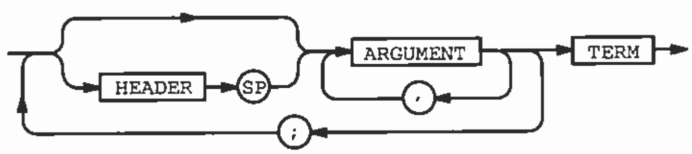
Query Format

A query message unit requests either function or display data from the instrument. The query message unit format is shown below.



Query Response Format

A query readies the spectrum analyzer for output. In query responses, the response header can either be returned with the response, or not returned (this depends on whether the header command (HDR) is turned ON or OFF). The output message format in response to a mode or parameter query is as follows.



Spectrum Analyzer Output Messages

When the spectrum analyzer executes a query, it buffers an output message unit that is a response to the query. Output message units contain ASCII characters (except when binary data is requested).

Output Message Format

The output message unit combines the header and appropriate argument(s).

Message units are combined if the output includes a response to the SET? query or to more than one query.

The header for query responses can either be turned on or off.

Output Message Execution

The spectrum analyzer begins output when talked, and it continues until it reaches the end of the information in its buffer or is interrupted by a DCL (Device Clear), UNT (Untalk), or IFC (Interface Clear) message.

If the spectrum analyzer is interrupted and the buffer is not cleared, the spectrum analyzer will resume output if it is retalked. The buffer may be cleared by the DCL messages, or if it is listened, by the SDC message or any device-dependent message.

If not interrupted, the spectrum analyzer terminates the output according to the selection made when the EXT INTFC CONFIG GPIB Menu Operating Mode Menu and Termination lf/**EOI** key sequence is executed. Default is **EOI**.

Functional Command Groups

This section lists spectrum analyzer commands in functional groups. If you are looking for all the commands related to a specific functional area of your spectrum analyzer, use these lists to locate associated command and query names.

Specific information about each command, if you already know the name of the command, is provided under the heading "Command Descriptions" following this section. To make finding specific command names easier, "Command Description" lists commands in alphabetical order.

The commands in this section are arranged under the following functional areas:

- Digital Storage
- Display Control
- Display Data and CRT I/O
- Frequency
- General Purpose
- Instrument Parameters
- Macros
- Marker Control
- Span and Resolution
- Status Byte
- Sweep Control
- Vertical Display and Reference Level
- Waveform Processing

Each functional block lists the command mnemonic and a brief functional description. The alphabetical list provides more detail on each command.

Digital Storage Commands

Table 2-1: Digital Storage Commands

Mnemonic	Function
AVIew	View A
AVGpos	Average peak cursor position
BVIew	View B
SAVea	Save A
BMIna	B minus Save A
DSTore	Store display
DREcal	Recall display
MXHld	Hold maximum
CRSor	Cursor mode

Display Control Commands

Table 2-2: Display Control Commands

Mnemonic	Function
CLIp	Blank baseline
DINt	Display intensity
GRAt	Graticule
HPOs	Horizontal position
VPOs	Vertical position
REDout	Readout on/off

Display Data and CRT Readout I/O

Table 2-3: Display Data and CRT Readout I/O

Mnemonic	Function
WFMpre	Waveform preamble
CURve	Display curve
WAVfrm?	Waveform query
DPRe?	Display preamble query
DCOpy?	Copy display query
RDOut	Readout message

Table 2-3: Display Data and CRT Readout I/O (Cont.)

Mnemonic	Function
TEXt	Text mode
ENRdo?	Entry readout query
UPRdo?	Upper readout query
MDRdo?	Middle readout query
LORdo?	Lower readout query

Frequency Commands

Table 2-4: Frequency Commands

Mnemonic	Function
FREq	Center frequency
TUNe	Incremental frequency change
TMOde	Set tune mode
FIRst	First LO frequency
SECond	Second LO frequency
TGMode	Tracking generator mode
SAMode	Sideband analyzer mode
DIScor	Disable tuning corrections
FRQrng	Frequency range
STEp	Step size
MSTep	Minus step
PSTep	Plus step
COUnt	Counter
CREs	Counter resolution
CNTcf	Count to center frequency
STStop	Sweep start and stop
DELfr	Delta frequency
DEGaus	Degauss tuning coils
EXMxr	External mixer input
CLVal	External mixer conversion loss value
CONlos	External mixer conversion loss mode

Table 2-4: Frequency Commands (Cont.)

Mnemonic	Function
IDEnt	Identify
IMPed	Impedance (Option 07 only)

General Purpose

Table 2-5: General Purpose Commands

Mnemonic	Function
HELP?	Help
KPAsgn	Assign keypad
UDAsgn	Assign user definable knob
STOre	Store settings
RECall	Recall settings
RDAta	Register data
PLOt?	Plot
PTYpe	Plotter type
POFset	Plot (B minus A) offset
ECR	End of sweep corrections
RVAlid?	Register valid
SSR	Send service request
TESt?	Internal test query

Instrument Parameters

Table 2-6: System Commands and Queries

Mnemonic	Function
SET?	Instrument settings
INIt	Initialize settings
ID?	Identify query
HDR	Header
MODe	Mode (native or 49X compatibility)

Front Panel Only

The following controls and adjustments are operated only from the instrument front panel (no remote control).

- Manual Scan
- Power
- Log Cal
- Ampl Cal

Macro Commands

There is a total of 8 Kbytes of memory dedicated for macro use. You may find it helpful to know the number of bytes used for each command and keep this in mind while preparing macros. The maximum number of bytes used by each command (for commands that can be used in a macro) is included in the "Command Description" section of this manual. Macro memory use is also summarized in Appendix B: *Macro Memory*.

Table 2-7: Macro Commands

Mnemonic	Function
Math Commands	
`PLUs`	Add X and Y registers
`SUBt`	Subtract X from Y Register
`MULt`	Multiply X and Y registers
`DIVide`	Divide Y by X register
Register Commands	
`PUTreg`	Put X register
`EXChg`	Exchange X and Y registers
`INTegr`	Convert X register to an integer
`POP`	Put Y register into X register
`ENPar`	Enter parameter into X register
`ENTer`	Enter value in X register
Branching and Looping Commands	
`GOTo`	Go to line (LABel)
`LABel`	Label point
`FOR`	Variable = X to Y Step Z
`NEXt`	Next variable
`RETurn`	Return from subroutine

Table 2-7: **Macro Commands (Cont.)**

Mnemonic	Function
Branching and Looping Commands	
GOSub	Go to subroutine (LABel)
IF	If statement is true
Print Commands	
CLEar	Clear macro readout buffer
DSLine	Display line
MRDo	Macro readout
PRInt	Display number or string
Data Commands	
MDAta	Store numeric data
REAd	Read and store in X register
MREsto	Restore data pointer
General Purpose Commands	
PAUse	Pause macro
RUN	Run macro
DONe	Macro finished
EMAc	End macro definition
KILl	Delete macro
MACro?	Macro status
MCStop	Stop macro
MEMory?	Macro memory
STNum	Store number
MENu	Display menu
SWEep	Take a sweep
VAR	Variable
STMac	Store macro
GETwfm	Get waveform
INPnum[1]	Input number

[1] Refer to Appendix A: *49X Compatibility Mode*

Marker Control

Table 2-8: Marker Control Commands

Mnemonic	Function
Marker System Control	
MARker	Marker mode
MTRace	Marker trace position
M1Asgn[1]	Assign marker 1
M2Asgn[1]	Assign marker 2
NSElvl	noise level normalization
SGTrak	Signal track
Marker Positioning	
DPMk	Display pointer to marker
MAMpl?	Marker amplitude query
MCEn	Marker to center
MEXchg	Exchange markers
MFReq	Marker frequency
MKTime	Marker time
MKDp	Marker to display pointer
MLOcat?	Marker location query
MTOp	Marker to reference level
MTUne	Tune marker
Marker Finding	
HRAmpl	Next higher amplitude
LRAmpl	Next lower amplitude
BWNum	Marker bandwidth number
BWMode	Marker bandwidth mode
MFBig	Marker peak find
MLFtnx	Marker left to next
PKFind	Peak find
PKCen	Marker to maximum and center
MMAx	Move marker to maximum
MMIn	Move marker to minimum

Table 2-8: Marker Control Commands (Cont.)

Mnemonic	Function
Marker Finding (Cont.)	
MRGtnx	Marker right to next
THRhld	Marker threshold
MVLFdb	Move marker left (dB)
MVRtdb	Move marker right (dB)
STYpe	Signal type
SGErr	Signal find error
Miscellaneous	
ZOOm	Zoom
ZETime	Zero time

[1] Refer to Appendix A: *49X Compatibility Mode*

Span and Resolution Commands

Table 2-9: Frequency Span and Resolution Commands

Mnemonic	Function
SPAn	Frequency span
ZERosp	Zero span mode
MXSpn	Maximum span mode
RESbw	Resolution bandwidth
AREs	Automatic resolution bandwidth

Sweep Control Commands

Table 2-10: Sweep Control Commands

Mnemonic	Function
TRIg	Triggering mode
SIGswp	Single sweep
TIMe	Time per screen

Status Byte

Table 2-11: Status Byte (response to serial poll)

Mnemonic	Function
DT	Define triggered events
EVEnt?	Event information query
ALLev?	All events query
ERR?	Error query
ERCnt?	Error count query
NUMev?	Number of events
EVQty?	Event quantity query

Vertical Display and Reference Level Commands

Table 2-12: Vertical Display and Reference Level Commands

Mnemonic	Function
VRTdsp	Vertical Display
REFlvl	Reference Level
RLUnit	Reference level units
ROFset	Reference level offset
RSTep	Reference level step size
CAL	Calibrate parameters
ENCal	Enable calibration factors
FINe[1]	Fine reference level steps
RLMode	Reference level mode
RGMode	Reduced gain mode
PEAk	Peaking
MINatt	Minimum RF attenuation
MAXpwr	Maximum allowable input power
RFAtt?	RF attenuation query
PLStr	Pulse stretcher
VIDflt	Video filter mode

[1] Refer to Appendix A: *49X Compatibility Mode*

Waveform Processing

Table 2-13: Waveform Processing

Mnemonic	Function
`POInt`	Display data pointer
`FIBig`	Find big
`LFTnxt`	Left next
`RGTnxt`	Right next
`FMAx`	Find maximum value
`FMIn`	Find minimum value
`CENsig`	Center signal
`TOPsig`	Move to top of screen

Waveform Data and Command Interactions

- Several of the waveform processing commands listed in this section operate only on the waveform specified by the last WFMpre or CURve command: either A or B, or both A and B).

  The waveform involved is first copied into a buffer. If the waveform is only half-resolution (either A or B), it is duplicated in the buffer to make a full 1000-point waveform before processing.

  Thus, whether the command operates on A or B or both, the range of X values for the display data point is always 1 to 1000.

- Waveform processing commands in this section that update the display data point use the same buffer memory as display data I/O; therefore, commands for these two functions can interact if executed as part of the same message. This command interaction can cause invalid data output with either CURve? or CURve.

  When the two conditions described below exist together, it can cause CURve? data output commands to be invalid:

  a. CURve? is followed by a command to update the display data point, and

  b. digital storage is updated during the execution of the message (either by repetitive sweeps or by the SIGswp command).

  When both of these conditions exist, the curve data output that follows completion of the entire message will not be the data that was loaded in the buffer at the time CURve? was executed.

  Instead, the curve data that is output will be the data that was loaded by the later command to update the display data point, because this later data overwrites the data already loaded in the buffer at the time CURVE? was executed.

The curve data is output as expected if CURve? follows the command to update the display data point instead of preceding it, because no conflict occurs in the way the commands use the buffer.

- VRTDSP LIN interacts with FIBIG, RGTNXT, and LFTNXT because they transform linear data into logarithmic data before execution.

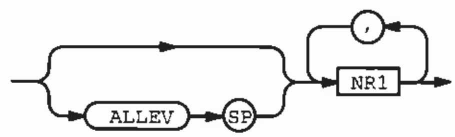
  This interaction is not apparent unless the transformed data is output over the GPIB or loaded into digital storage because of either of the conditions (a or b) noted in the interaction described above.

For additional information, refer to "Multiple Use of Display Buffer" in the *Programming Examples* section of this manual.

ALLev? (all events)

Query:

Response:

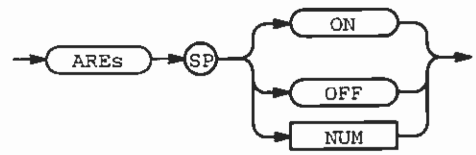


NR1 — is an event code as defined in *Command Syntax, Numbers*.

Events are cleared when their event codes have been reported.

AREs (automatic resolution bandwidth)

Syntax:



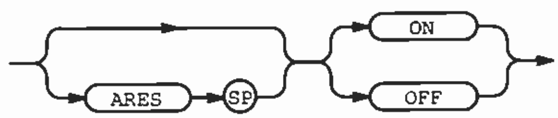
Arguments: **ON** — The current span is matched with an appropriate resolution bandwidth that maintains calibrated performance for the current sweep speed, if possible. When both auto resolution and auto time are selected, resolution is selected based on span, and time is selected to maintain calibration.

OFF — ARES ON is cancelled, leaving the resolution bandwidth at the current value.

Interactions: ARES OFF also cancels RESBW AUTO. ARES is turned off by any RESBW command except RESBW AUTO.

Query:

Response:



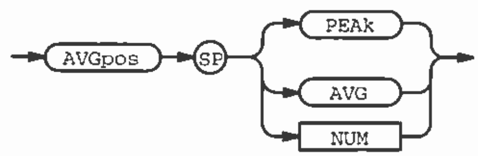
ARES is not included in the response to the SET query because AUTO is included in the RESBW? response.

Macro Memory Used: 2 bytes.

Power-up Value: ON.

AVGpos (average position)

Syntax:



Arguments:

AVG — Puts digital storage into average mode for all signal levels.

PEAK — Puts digital storage into peak mode for all signal levels.

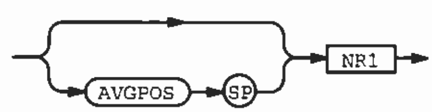
NUM — Sets the peak/average cursor to NUM screen units.
0 equates to AVGPOS PEAK.
255 equates to AVGPOS AVG.

Range: Bottom of displayed screen = 25.
Top of displayed screen = 225.

Interactions: NUM values >0 and <25, and >225 and <250 position the cursor below or above the displayed screen and do not affect the display; however, they do affect waveform output in response to the CURVE? query because CURVE includes one division above and below the displayed screen.

Query:

Response:



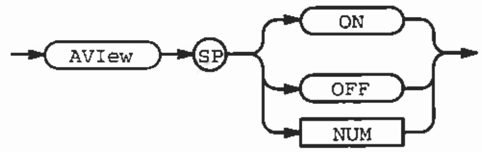
AVGPOS 0 is returned if digital storage is in peak mode. AVGPOS 255 is returned if digital storage is in average mode.

Macro Memory Used: 2 bytes.

Power-up Value: Last front panel setting.

AView (A waveform display)

Syntax:



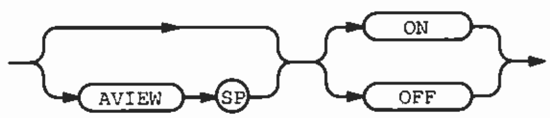
Arguments: **ON** — The A waveform is displayed on the CRT. The A and B waveforms are independent and may be displayed together or separately; however, both waveforms will be displayed if either AVIEW or BVIEW is on and SAVEA is off.

OFF — The display of the A waveform is turned off. (See the ON description for operation with SAVEA off.) If both AVIEW and BVIEW are turned off, the input signal is displayed in real time rather than a digitized trace being displayed.

Interactions: While SAVEA is ON, any updating of the trace display in A is halted.

Query:

Response:

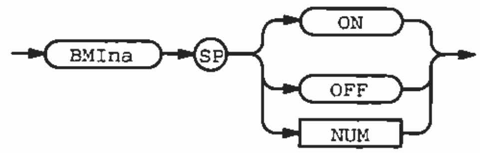


Macro Memory Used: 2 bytes.

Power-up Value: ON.

BMIna (B minus Save A waveform display)

Syntax:



Arguments: **ON** — The instrument turns on SAVEA if it is off and then turns on a display of the difference between the A waveform and the B waveform, which is continuously updated. The difference trace baseline is set at screen center.

OFF — The difference display is turned off.

NOTE

The B minus Save A trace cannot be acquired via the GPIB. It is only displayed on the CRT.

Interactions: BMINA ON turns SAVEA ON. SAVEA OFF turns BMINA OFF.

Query:

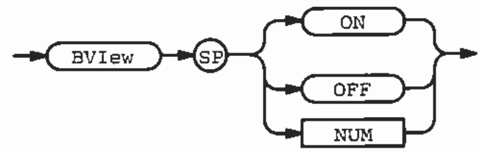
Response:

Macro Memory Used: 2 bytes.

Power-up Value: OFF.

BView (B waveform display)

Syntax:

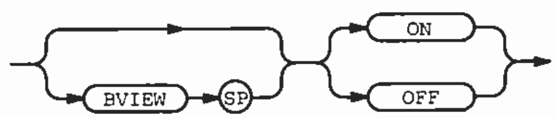


Arguments: **ON** — The B waveform is displayed on the CRT. The A and B waveforms are independent and may be displayed together or separately; however, both waveforms will be displayed if either AVIEW or BVIEW is on and SAVEA is off.

OFF — The display of the B waveform is turned off. (See the ON description for operation with SAVEA off.) If both AVIEW and BVIEW are turned off, the input signal is displayed in real time.

Query:

Response:

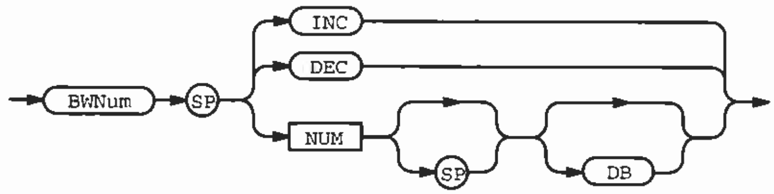


Macro Memory Used: 2 bytes.

Power-up Value: ON.

BWNum (marker bandwidth number)

Syntax:



Arguments: **INC** — Increments the marker bandwidth level 1 dB.

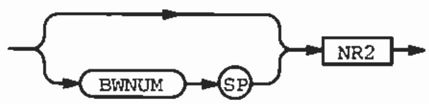
DEC — Decrements the marker bandwidth level 1 dB.

NUM — Sets the marker bandwidth to the value specified. Decibel units may be either expressed or omitted.

Interactions: BWNUM sets the level below the signal peak used in the bandwidth mode (BWMODE) at which the bandwidth is found. This number is stored in battery-powered memory.

Query:

Response:

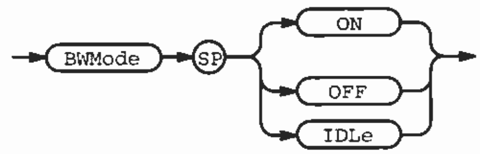


Macro Memory Used: 3 bytes.

Power-up Value: The value set is stored in memory. If a number has never been set or if the memory fails, the value will be 6 dB.

BWMode (marker bandwidth mode)

Syntax:



Arguments: **ON** — The BWMODE command moves the delta markers down from the peak of the signal that the primary marker is on by the value set in BWNUM. (This value could also have been set from the front panel. If no value has ever been set, the value used will be 6 dB.)

BWMODE moves in 1⁄10 dB steps. The primary marker is placed on the right (higher frequency) side of the signal and the secondary marker is placed on the left (lower frequency) side of the signal.

OFF — The BWMODE is disabled.

IDLE — If the primary marker is not on a signal or if a point NUM dB (set by the BWNUM command) down cannot be found on each side of the signal, the secondary marker moves to the location of the primary marker, and BWMODE goes to IDLE. Since IDLE can be returned as a query response, it is accepted as a command that has the same effect as ON.

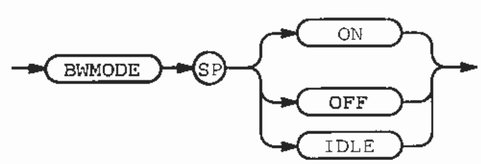
Interactions: When BWMODE goes to IDLE, marker execution warning message 130 is issued if SIGERR is on.

BWMode sets MARKER to DELTA. The markers are reset after the marker position or BWNum is changed (or at every sweep if on an active trace). The definition of the criteria for a signal is set by the THRHLD command.

The Lin Mode, Multiband Sweep Mode, and Zero Span Mode are not available in BWMODE.

Query:

Response:

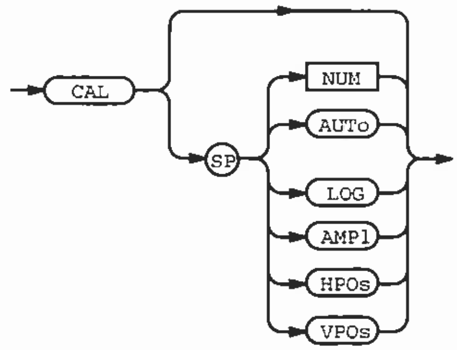


Macro Memory Used: 2 bytes.

Power-up Value: OFF.

CAL (calibration)

Syntax:



NOTE

The instrument assumes a 100 MHz calibrator is connected during CAL AUTO, CAL AMPL, and CAL LOG operation.

Arguments: **NUM**

0 = AUTO
1 = LOG
2 = AMPL
3 = HPOS
4 = VPOS

AUTO — (or, `CAL` , with no arguments) The resolution bandwidth filter frequencies are calibrated with respect to 10 MHz and levels relative to the 3 MHz filter level (within a range of +2, −4 dB) and bandwidths used in dB/Hz normalization are measured. During operation, the word MEASURING appears on the screen. All calibration results are stored in battery-powered memory.

LOG — The instrument is set up so you can set the front-panel LOG CAL adjustment. CAL LOG has an indefinite execution time and will operate until either a device clear (DCL) is received from the GPIB port or the spectrum analyzer is returned to local control from the instrument front panel. An instruction message appears on the screen.

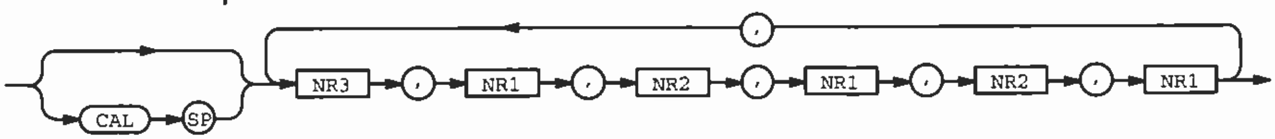
AMPL — The instrument is set up so you can set the front-panel AMPL CAL adjustment. CAL AMPl has an indefinite execution time and will operate until either a device clear (DCL) is received from the GPIB port or the spectrum analyzer is returned to local control from the instrument front panel. An instruction message appears on the screen.

HPOS — The instrument is set up so you can set the front-panel horizontal POSITION control. CAL HPOs has an indefinite execution time and will operate until either a device clear (DCL) is received from the GPIB port or the spectrum analyzer is returned to local control from the instrument front panel. An instruction message appears on the screen.

VPOS — The instrument is set up so you can set the front-panel vertical POSITION control. CAL VPOs has an indefinite execution time and will operate until either a device clear (DCL) is received from the GPIB port or the spectrum analyzer is returned to local control from the instrument front panel. An instruction message appears on the screen.

Query:

Response:



In the CAL? response, the same data is given in succession from the widest to the narrowest filter. The data given for each filter is the following.

- frequency error
- frequency calibration code
- level error
- level calibration code
- noise bandwidth factor
- bandwidth calibration code

The frequency error is the difference between the measured filter frequency and 10 MHz, expressed in Hz.

The level error is the difference between the measured filter level and the measured level of the 3 MHz filter, expressed in dB.

The noise bandwidth is expressed as the dB correction used to normalize the filter's output to 1 Hz. Use Table 2-14 to decode the calibration code numbers.

Table 2-14: Calibration Codes

Code Number	Description
0	A calibration value for this item has not been found (that is, this filter has never been calibrated before). This will always be returned for the 10 Hz filter noise bandwidth factor.
1	A calibration value for this item has been found, but the most recent calibration attempt failed (the last previously-good value is used).

Table 2-14: Calibration Codes (Cont.)

Code Number	Description
2	The value recorded for this item is the limit value (that is, the best it could do). The actual required correction would exceed the limit (+2, −4 dB), so this item is not calibrated. (This applies to amplitude calibration only.)
3	A calibration value for this item has been found, but the most recent calibration attempt failed (the last previously-good value is used). The value recorded for this item is the limit value (that is, the best it could do). The actual required correction would exceed the limit (+2, −4 dB), so this item is not calibrated. (This applies to amplitude calibration only.)
4	The last calibration attempt for this item succeeded.
5	This filter is the reference for level calibration. (This applies to amplitude calibration only.)

CENsig (center signal)

Syntax: → (CENsig) →

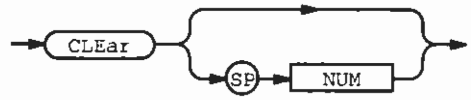
This command tunes the frequency to center the signal represented by the display data point (or as close as possible, given the specified span accuracies).

This command does not get a new display data point or digital storage waveform. Therefore, if a new waveform is acquired after CENSIG is sent, the display data point may no longer match the signal of interest.

Macro Memory Used: 1 byte.

CLEar (clear macro readout buffer)

Syntax:



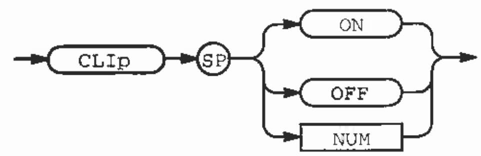
Arguments: NUM — Only the numbered line (1 to 16) in the macro readout buffer will be cleared.

CLEAR without an argument or NUM greater than 16 will clear the entire macro readout buffer.

Macro Memory Used: 2 bytes.

CLIp (blank baseline)

Syntax:

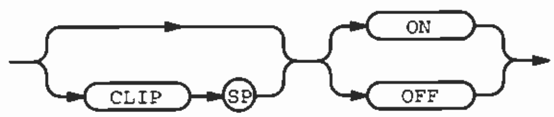


Arguments: **ON** — Approximately one graticule division of the screen trace is turned off at the baseline of the CRT. This eliminates the bright baseline when photographing the display.

OFF — The full trace is displayed on the CRT.

Query:

Response:



Macro Memory Used: 2 bytes.

Power-up Value: OFF.

CLVal (conversion loss value)

2792 Option 04 and 2794 Only

Syntax:

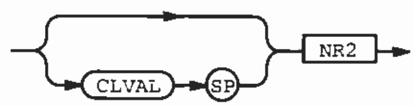
Arguments: NUM — Sets the conversion loss of an external mixer to a value, in dB, specified by NUM.

Positive values indicate loss; negative values indicate gain.

Range: ±100 dB

Query:

Response:



Macro Memory Used: 3 bytes.

Power-up Value: Factory default depends on the band.

CNTcf (count to center frequency)

Syntax: → CNTcf →

A count of the signal is taken; this signal count result is then transferred to the center frequency. This tunes the spectrum analyzer to the signal counted. Accuracy is limited by the count resolution in use when the signal count is done.

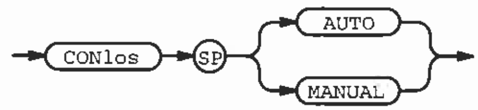
Interactions: CNTCF is not available if either MXSPN or ZEROSP is ON.

Macro Memory Used: 1 byte.

CONlos (conversion loss mode)

2792 Option 04 and 2794 Only

Syntax:



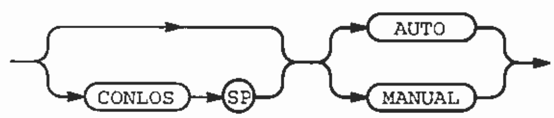
This command sets the conversion loss mode for an external mixer.

Arguments: **AUTO** — Sets the external mixer conversion loss using factory default values.

MANUAL — Sets the external mixer conversion loss to the value specified by either CLVAL or the front panel.

Query:

Response:

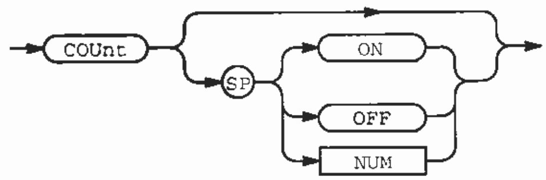


Macro Memory Used: 2 bytes.

Power-up Value: AUTO.

COUnt (counter)

Syntax:



Arguments: When no argument is included, an immediate signal count occurs whether or not the counter mode is on; no change occurs in the ON/OFF status of the counter mode.

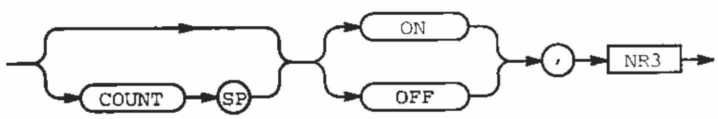
ON — The counter mode is turned on.

OFF — The counter mode is turned off.

Interactions: If a marker(s) is on an unsaved trace, COUNT will count at the marker. COUNT will count at center screen if MKR is OFF. If MXSPN is ON, counting will occur at the maximum span dot position rather than center screen. Signals cannot be counted on a trace saved with SAVEA or with STORE and RECALL.

Query: COUNT (counter) query

Response:



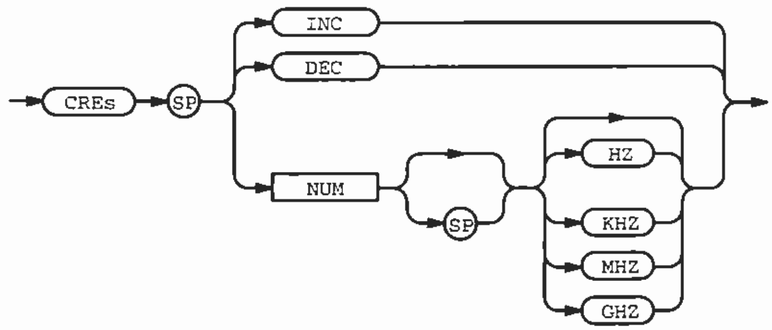
The number returned in this response is the result of the last count, regardless of whether COUNT is ON or OFF. If a signal count has not been made, 0 will be returned. The number will not be returned with the SET query response.

Macro Memory Used: 2 bytes.

Power-up Value: OFF.

CREs (counter resolution)

Syntax:



Arguments: **INC** — Increments the counter resolution in decade steps. If the initial value is not an even decade, the new value will be the nearest decade up.

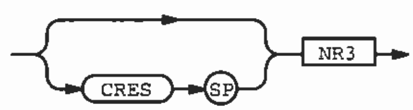
DEC — Decrements the counter resolution in decade steps. If the initial value is not an even decade, the new value will be the nearest decade down.

NUM — The proper decade of counter resolution is selected for use. Numbers that are not powers of ten will be set to the next lower power of ten, up to a maximum of 1 GHz resolution. If the number is out of range, execution error message 30 will be issued.

Range: The range is 0 to 1 GHz.

Query:

Response:

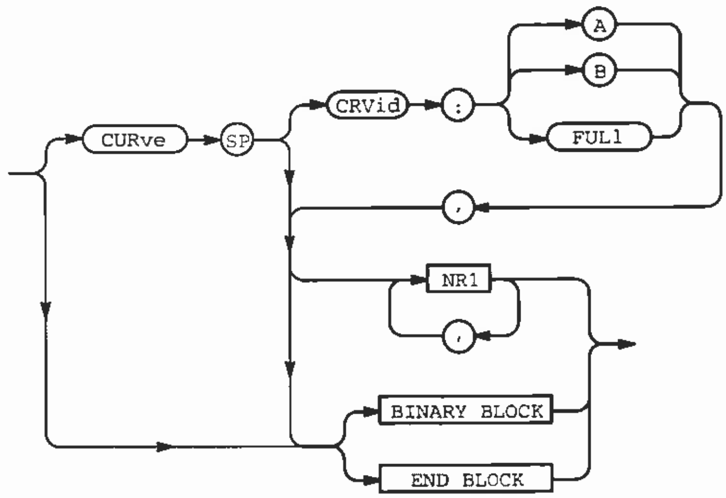


Macro Memory Used: 4 bytes.

Power-up Value: 1 Hz.

CURve (display curve)

Syntax:



NOTE

The instrument should be in the Single Sweep mode and not be sweeping during the CURVE command or query. If it is sweeping during the CURVE query, it could give erroneous information, unless transferring a SAVEA display.

Arguments: **CRVID** — The destination (A, B, or FULl) is selected for the waveform being sent. If this argument is omitted, the last CRVid in a CURVE command or WFID in a WFMPRE command takes precedence. A or B indicates a 500-point transfer; FULL indicates 1000 points.

NUM — This is a sequence of ASCII-coded digits, delimited by commas between successive numbers.

BINARY BLOCK — Binary block is a sequence of binary numbers that is preceded by the ASCII code for percent (%) and a two-byte binary integer representing the number of binary numbers plus one (the extra byte is the checksum) and followed by the checksum.

The checksum is the 2's-complement of the modulo-256 sum of all preceding bytes except the first (%). Thus, the modulo-256 sum of all bytes. except the first (%) should equal zero to provide an error check of the binary block transfer.

END BLOCK — End block is a sequence of binary numbers that is preceded by the ASCII code for at (@); EOI must be asserted concurrently with the last data byte. The end block can only be the last data type in the message.

Interactions: A waveform sent in a CURVE command is overwritten in the display I/O buffer if preceded by a CURVE query in the same message. This causes the queried display data to be put back into digital storage.

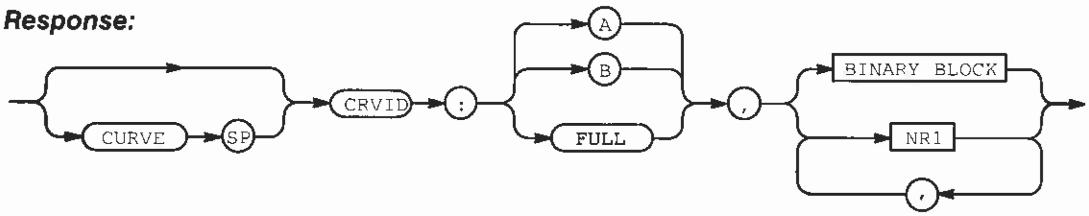
Examples:

`CURve CRVID:FULL,100,100,101,99,`
<996 more numbers>
`CURve` <500 or 1000 numbers>
`CUR` <BINARY BLOCK>

Macro Memory Used: The CURVE command cannot be used within a macro.

Query:

Response:



The data type (BINARY BLOCK or NR1) is set by the last WFMPRE command.

Waveform data is related to the display as shown in Figure 2-2.

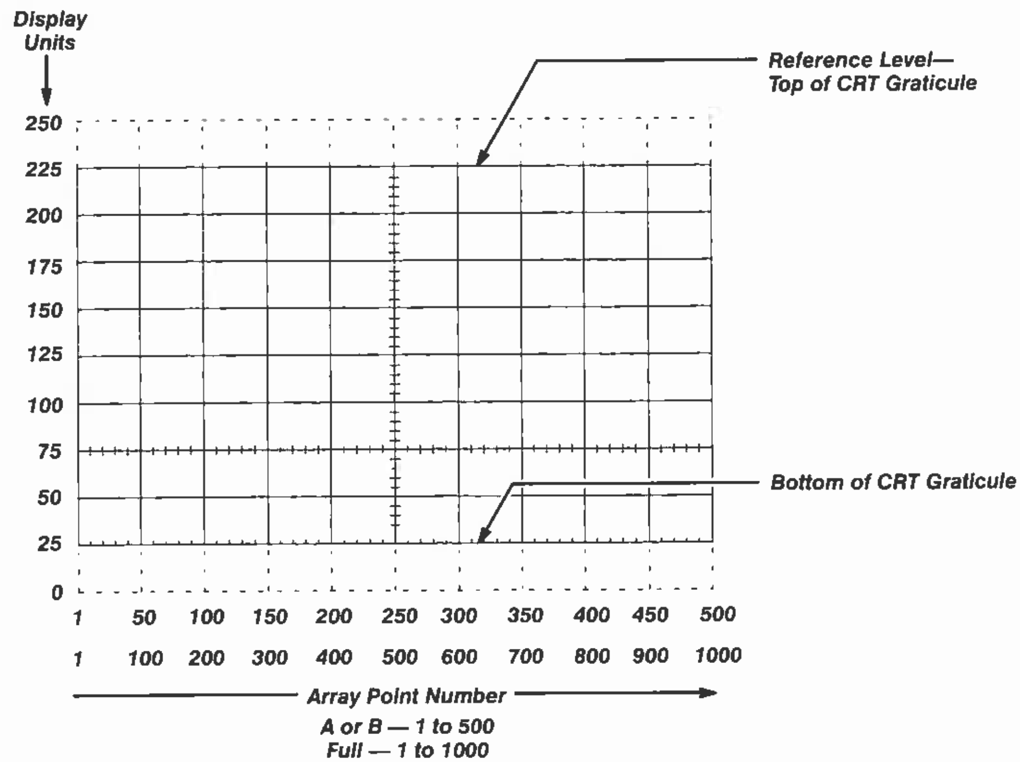


Figure 2-2: Waveform Data Related to the Display

DCOpy? (copy display)

Query:

Response: The DCOPY query response is the same as the response to ID?, WFMPRE?, DPRE?, and CURVE?.

DCOPY? allows transmission of information from one device to another in "display" units so that a hard copy can be made of the display.

DEGaus (degauss tuning coils)

Syntax: → DEGaus →

A current is momentarily turned off to remove residual magnetism in the 1ST LO and preselector.

Macro Memory Used: 1 byte.

DINt (display intensity)

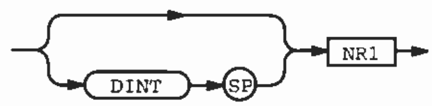
Syntax:

This command sets the display intensity to a level corresponding to a DAC value specified by NUM.

Range: 0 to 255, where 255 corresponds to maximum intensity.

Query:

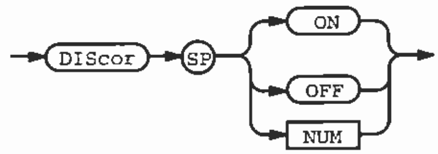
Response:



Power-up Value: Last front panel setting.

DIScor (disable tuning corrections)

Syntax:



This command is included to allow disabling of the frequency control loop in the instrument for speed or diagnostics purposes.

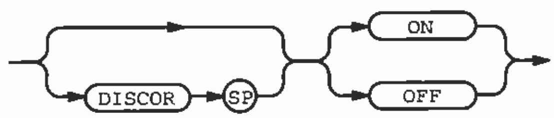
Arguments:

ON — Center frequency corrections are disabled.

OFF — Center frequency corrections are enabled.

Query:

Response:



Macro Memory Used: 2 bytes.

Power-up Value: OFF.

DIVide (divide Y by X register)

Syntax: DIVide

The DIVIDE command divides the contents of the Y register by the contents of the X register and puts the result in the X register (XREG = YREG ÷ XREG).

Range: ±9.999 999 999 999 E+99 to ±9.999 999 999 999 E−99.

Interactions: The Y register is unchanged after the DIVIDE command.

Examples: The following example illustrates the use of the DIVIDE command (10÷5).

```
 90   Z=1 ! ADDRESS OF SPECTRUM ANALYZER
100   Print#z:"ENTER 10"
110   Print#z:"ENTER 5"
120   Print#z:"DIVIDE"
130   Print#z:"STNUM 1"
140   Print#z:"VAR? 1"
150   Input#z:r
160   Print r
```

Line 100	Enters 10 into XREG.
Line 110	Moves the contents of XREG to YREG, then enters 5 in XREG.
Line 120	Divides the contents of YREG (10) by the contents of XREG (5). YREg still contains 10.
Line 130	Stores the contents of XREG in variable number 1.
Line 150	Puts the contents of variable number 1 into r.
Line 160	Prints the contents of r (2).

Macro Memory Used: 1 byte.

DONe (macro execution finished)

Syntax: → DONe →

The DONE command tells the spectrum analyzer that macro execution is finished. Every macro must include the DONE command unless the macro is in a continuous loop (e.g., a routine that measures bandwidth at the end of each sweep is in a continuous loop).

NOTE

The last command in a macro before the EMAC command must be GOTO or RETURN or DONE. If it is not, macro execution error message 178 is issued.

Macro Memory Used: 1 byte.

DPMk (display pointer to marker)

Syntax:

The DPMK command moves the display pointer to the primary marker position.

Interactions: DPMK cancels the WFID portion of any previous WFMPRE command or the CRVID portion of any previous CURVE command, and selects the FULL waveform for data transfers and waveform processing.

If MARKER is OFF, DPMK sets MARKER to SINGLE.

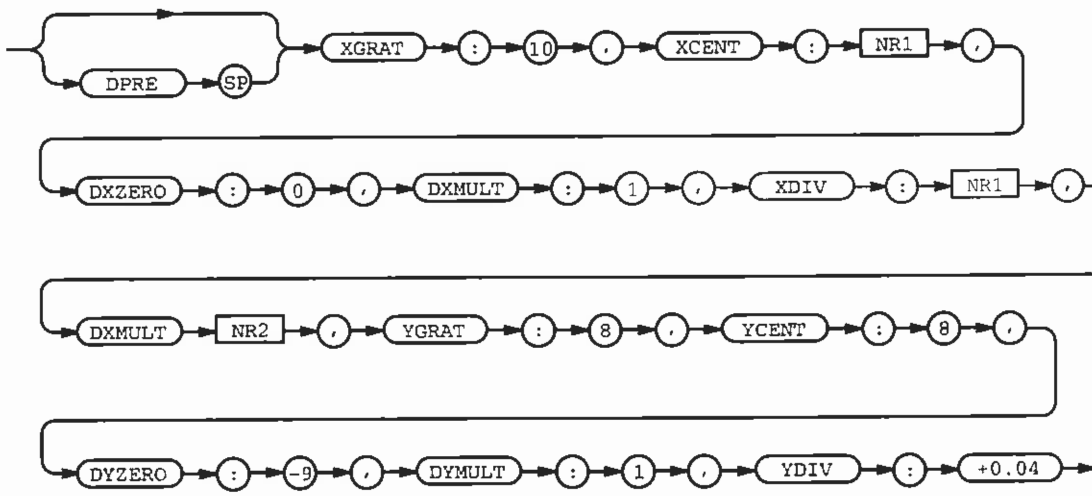
Macro Memory Used: 1 byte.

DPRe? (display preamble)

Query:

DPRE? calls for transmission of the display preamble. The display preamble contains numeric data items to be used with corresponding curves to reproduce a display.

Response:



GRAT:10 — Specifies the X (horizontal) graticule size.

XCENT — Is the X center of the display data in number of divisions relative to the left-hand side of the graticule.

DXZERO:0 — Displays the X offset in divisions relative to XCENT.

DXMULT:1 — Displays the X multiplier.

XDIV — Displays X divisions/unit.

YGRAT:8 — Specifies the Y (vertical) graticule size.

YCENT:8 — Is the Y center of the display in number of divisions relative to the bottom of the graticule.

DYZERO:-9 — Displays the Y offset in divisions relative to YCENT.

DYMULT:1 — Displays the Y multiplier.

YDIV:+0.04 — Displays Y divisions/unit.

X-Axis Scaling

X-axis specifications XGRAT, XCENT, DXZERO, DXMULT, and XDIV are used to interpret the position of the ordered points in absolute X values.

*DXN = DXMULT * (N−PT.OFF) * XDIV + XCENT + DXZERO* where:

DXN is the X value in graticule divisions

Y-Axis Scaling

Y-axis specifications YGRAT, YCENT, DYZERO, DYMULT, and YDIV are used to interpret the position of the ordered points in absolute Y values.

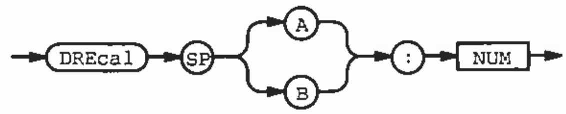
*DYN = DYMULT * VALN * YDIV + YCENT + DYZERO*

where:

DYN is the Y value in graticule divisions
VALN is the current data value

DREcal (recall display)

Syntax:



Arguments: **A** — A waveform is recalled from the memory specified by NUM (0–8) and put in the A waveform display. If AVIew is ON and BVIew and BMINA are OFF, the readout associated with a recalled A waveform is displayed.

B — A waveform is recalled from the memory specified by NUM (0–8) and put in the B waveform display. If BVIEW or BMINA is ON, the readout associated with a recalled B waveform is displayed if in single sweep.

NOTE

The contents of B will be overwritten on the next sweep unless SINGLE SWEEP is ON.

Interactions: DRECAL turns SAVEA ON. The B waveform display will be overwritten if the instrument is not in the single-sweep mode.

If you try to recall a waveform from an empty memory location, execution error message 62 will be issued.

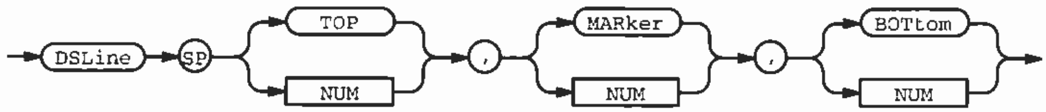
Examples:

```
DRE A:4
DRE B:2
```

Macro Memory Used: 3 bytes.

DSLine (display line)

Syntax:



The DSLINE command affects the normal 3-line readout and displays the normal top line of CRT readout (TOP), the normal marker (2nd) line of CRT readout (MARKER), and the normal bottom line of CRT readout (BOTTOM).

TOP, MARKER, and BOTTOM must appear in the command line in the same position as they appear in the syntax diagram above. If a line number is used instead of TOP, MARKER, or BOTTOM, that line from the macro readout buffer is displayed.

The location of the line number in the command line does not matter. If 0 is used as a line number, that line will not be displayed (the line will be blank). If NUM is greater than 16, the normal readout is displayed.

Range: NUM is 1 to 16.

Interactions: The TEXT mode must be in SHORT (normal 3-line) mode for DSLINE to have any effect on the readout.

Examples: See Example Descriptions Below

`DSLINE 1,2,0`	See note 1
`DSLINE 0,1,2`	See note 2
`DSLINE TOP,4,8`	See note 3
`DSLINE 1,MARKER,4`	See note 4

The following examples illustrate incorrect use of DSLINE.

`DSLINE 1,2,1`	See note 5
`DSLINE TOP,1,2`	See note 6

On the screen the first eight lines are displayed; then there is a blank area (about the width of two lines); then the last eight lines are displayed.

If a line number is used more than once, macro execution error message 169 will be issued.

Example Descriptions:

1. Two lines are displayed (line 1 from the macro readout buffer in place of the normal top line and line 2 from the macro readout buffer in place of normal marker line).
2. Two lines are displayed (the result is the same as description 1 above).
3. Three lines are displayed (normal top line, line 4 from the macro readout buffer at the fourth line on the screen, and line 8 from the macro readout buffer at the eighth line on the screen.

4. Three lines are displayed (line 1 from the macro readout buffer in place of the top line, normal marker line, and line 4 from the macro readout buffer at the fourth line on the screen.
5. Line 1 is used twice.
6. Line 1 is used twice (TOP is displayed on line 1).

The following example illustrates the use of the DSLINE command.

```
 80    Z=1 ! ADDRESS OF SPECTRUM ANALYZER
100    Print#z:"CLEAR"
110    Print#z:"PRINT 3,0,'LINE 3'"
120    Print#z:"PRINT 7,0,'LINE 7'"
130    Print#z:"PRINT 8,0,'LINE 8'"
140    Print#z:"DSLINE TOP,3,7"
150    Wait 5
160    Print#z:"DSLINE 7,MARKER,8"
```

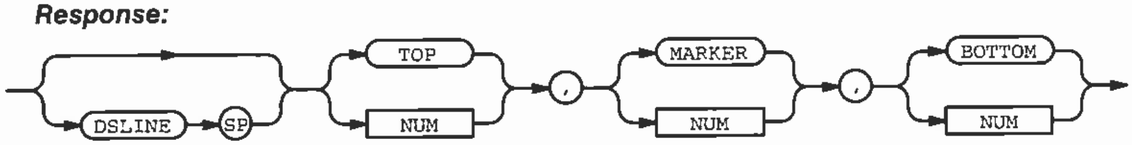
Line 100	Clears the macro readout buffer.
Line 110	Puts LINE 3 in line 3 of the macro readout buffer.
Line 120	Puts LINE 7 in line 7 of the macro readout buffer.
Line 130	Puts LINE 8 in line 8 of the macro readout buffer.
Line 140	Displays three lines: normal top line, line 3 of the macro readout buffer at the third line on the screen, and line 7 of the macro readout buffer at the seventh line on the screen.
Line 160	Displays three lines: normal marker line at the second line on the screen, line 7 of the macro readout buffer at the seventh line on the screen, and line 8 of the macro readout buffer at the eighth line on the screen.

Macro Memory Used: 4 bytes.

Power-up Value: TOP, MARKER, and BOTTOM.

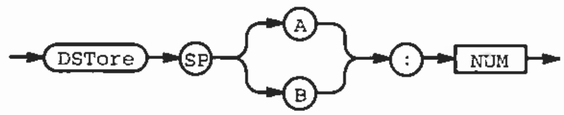
Query:

Response:



DSTore (store display)

Syntax:



Arguments: **A** — The A waveform is stored in the memory location indicated by NUM. If the number requested is out of the range limit, execution error message 47 is issued.

B — The B waveform is stored in the memory location indicated by NUM. If the number requested is out of the range limit, execution error message 47 is issued.

The readout and markers associated with the display are stored with the display.

Range: 0 to 8

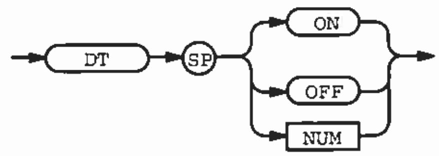
Examples:

```
DSTORE A:4
DST B:2
```

Macro Memory Used: 3 bytes.

DT (define triggered events)

Syntax:



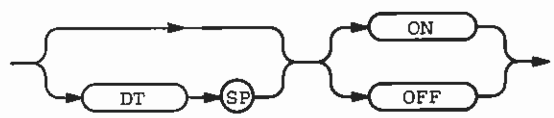
Arguments: **ON** — GET is enabled to trigger a new sweep.

OFF — The response to GET is disabled.

Interactions: If DT is OFF and GET is received, execution error message 45 will be issued.

Query:

Response:

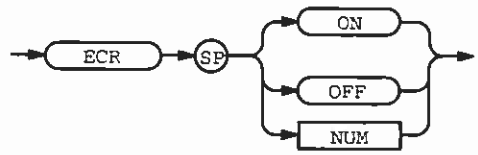


Macro Memory Used: 2 bytes.

Power-up Value: ON.

ECR (end of sweep corrections)

Syntax:



This command causes oscillator corrections to occur either at the end of every sweep or as needed, based on the drift rate of the oscillators.

Arguments:

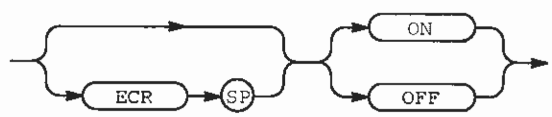
ON — Oscillator corrections occur at the end of every sweep.

OFF — The time between oscillator corrections is determined by the drift rate of the oscillators.

Interactions: When ECR is ON, corrections will generally occur more frequently than when ECR is OFF. The extra time spent correcting the oscillators may lengthen the response time to other commands and queries.

Query:

Response:



Macro Memory Used: 2 bytes.

Power-up Value: OFF.

EMAc (end macro)

Syntax:

The EMAC command tells the spectrum analyzer that entering input for a new macro is done. The spectrum analyzer then stops storing, compiles the macro, and checks for errors.

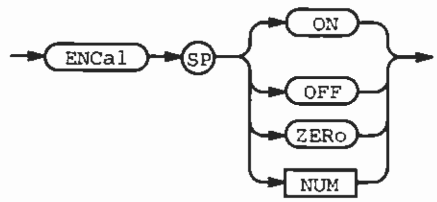
Table 2-15 lists the errors that may be sent when the macro is being compiled.

Table 2-15: Error Messages

Error Number	Description
106	Argument out of range (LABEL)
169	Line number used more than once (DSLINE command)
170	Missing LABEL
171	Missing ENDI
172	Missing NEXT
173	ELSE command not expected
174	ENDI command not expected
175	NEXT command not expected
177	Index out of range (DISBUF:INDEX or VAR:INDEX)
178	FOR index is already used

ENCal (enable calibration)

Syntax:



Arguments: **On** — The calibration factors are used internally to correct frequency and level errors and noise bandwidth in the filters.

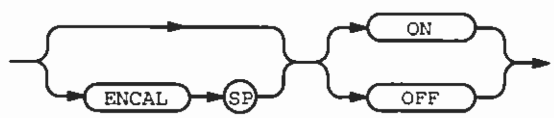
OFF — Filter amplitude and frequency are not corrected, and the nominal noise bandwidth is used.

ZERO — Set calibration factors to 0; this does not affect OFF/ON status.

NUM 0 = OFF
1 = ON
2 = ZERO

Query:

Response:

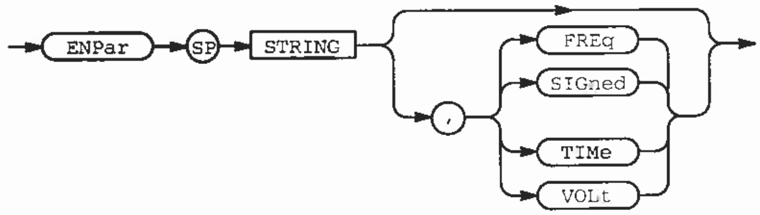


Macro Memory Used: 2 bytes.

Power-up Value: ON.

ENPar

Syntax:



A number entered using the keypad is placed into the X register. An optional argument sets the offset of the terminated keys. Omitting a terminator key multiplies the entered value by 1.

STRING — The string is entered using the keypad, delimited by a set of single (') or double (") quotes that are displayed as an entry prompt. Maximum string length is 25 characters; longer strings are truncated.

FREQ — Keys used as terminators and their effects are listed below:

- GHz multiplies entry by 1E9
- MHz multiplies entry by 1E6
- kHz multiplies entry by 1E3
- Hz multiplies entry by 1

SIGNED — Keys used as terminators and their effects are listed below:

- +dBx multiplies entry by 1
- −dBx multiplies entry by −1

TIME — Keys used as terminators and their effects are listed below:

- µSEC multiplies entry by 1E−6
- mSEC multiplies entry by 1E−3
- SEC multiplies entry by 1

VOLT — Keys used as terminators and their effects are listed below:

- nV multiplies entry by 1E−9
- µV multiplies entry by 1E−6
- mV multiplies entry by 1E−3
- V multiplies entry by 1

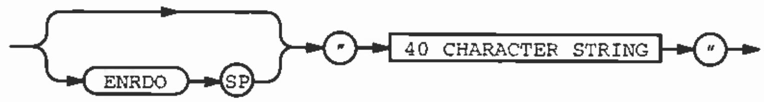
Interactions: GPIB and macro execution is paused until a terminator key is pressed or the command is aborted using the [ESC] key. Pressing [ESC] or a terminator key without entering a string leaves the X register unchanged.

Macro Memory Used: 30 bytes.

ENRdo? (enter readout)

Query:

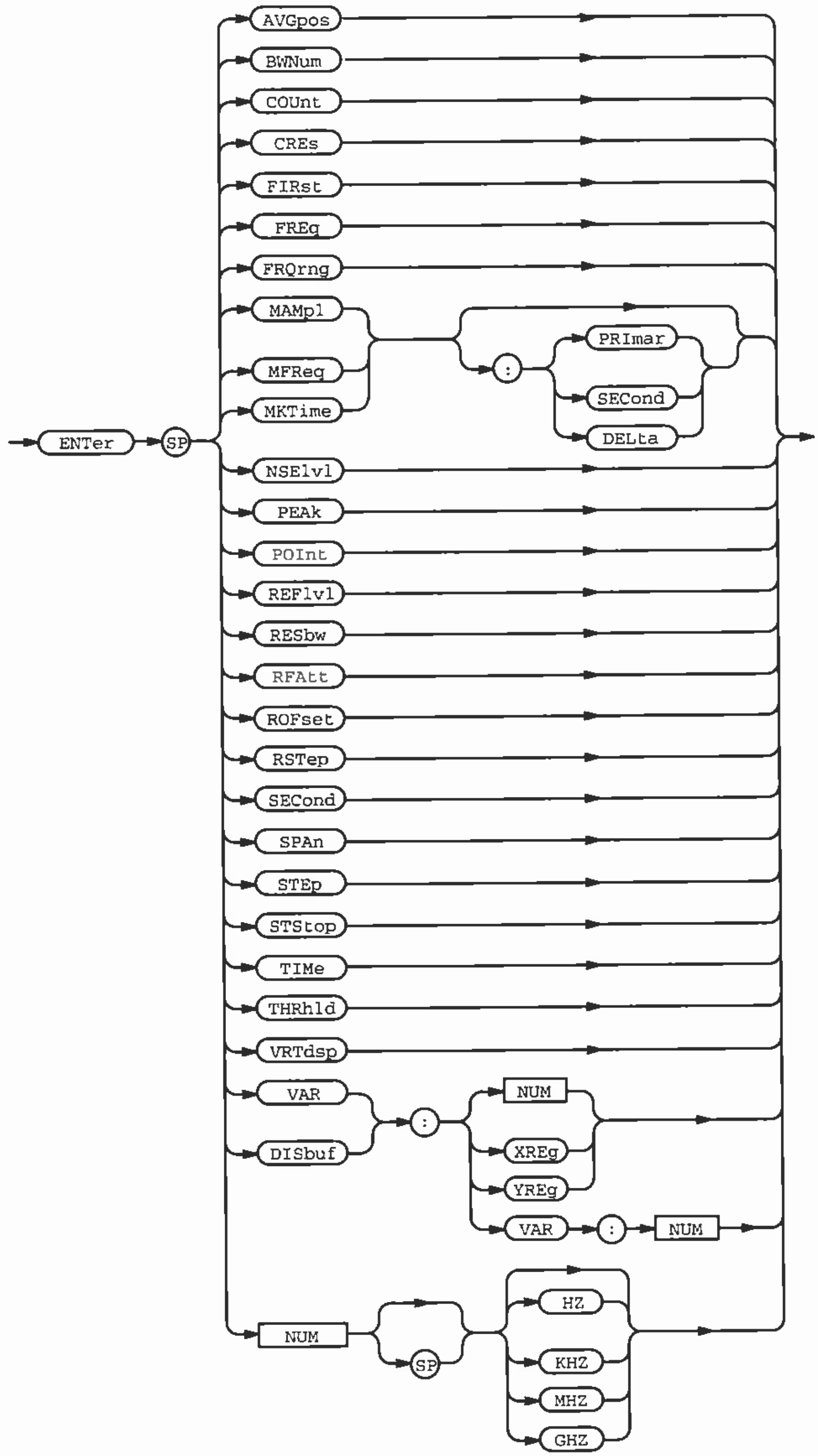
Response:



This query returns the current entry line as a 40-character string; blanks are transmitted as spaces.

ENTer (enter value in X register)

Syntax:



The ENTER command in a macro works much like a query in a controller program to get the current value of the given argument (setting or variable). The value is placed in the X register.

For a description of the command arguments, see the description of the command whose mnemonic is the same as the argument. When ENTER does not have an argument, the contents of XREG are copied into YREG (both XREG and YREG will hold the same value).

Arguments: **POINT** — The X POINT value will be put into XREG and the Y POINT value will be put into YREG.

STSTOP — The start frequency will be put into YREG and the stop frequency will be put into XREG.

VAR — The variable is entered into XREG — for example, VAR [NUM], VAR [XREG], VAR [YREG], VAR [VAR(NUM)].

If the index is out of range, macro execution error message 176 is issued and the macro is aborted.

DISBUF — The display buffer point indexed by NUM, XREG, YREG, or VAR [NUM] is entered into XREG — for example, DISBUF [NUM], DISBUF [XREG], DISBUF [YREG], DISBUF [VAR (INDEX)].

Execution error message 176 is issued for the following conditions:

- If the overall index for DISBUF (the part following the first colon (:) is outside the range of 1–1000.
- If the NUM indicating which variable to use is outside of the range of 1–30.

NOTE

The GETWFM command must be used before ENTER DISBUF can be used. See the GETWFM command description for more information.

RSTEP — Sets the X register to the reference level step size. If the vertical display is linear and the step size is set to AUTO, the step size varies; therefore, there is no fixed value to enter. In this case, 1 dB is entered into the register.

NUM — A number, with or without a units designator, is entered into XREG.

SPAN — The value entered into the X register differs if the instrument is in 49X compatibility mode. In normal operation, full-screen values are entered; in 49X compatibility mode, per-division values are entered.

TIME — The value entered into the X register differs if the instrument is in 49X compatibility mode. In normal operation, full-screen values are entered; in 49X compatibility mode, per-division values are entered.

Range: **VAR** — 1 to 30.

DISBUF — 1 to 1000.

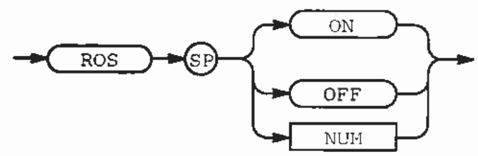
Interactions: Before the value of XREg is changed, the value in XREG is copied into YREG.

Examples:

```
ENTer MFREQ
ENTer MAMPL:SEC
ENTer VAR:YREG
ENTer DISBUF:VAR:NUM
ENTer 100
ENTer 100MHZ
```

Macro Memory Used: 10 bytes.

EOS (end-of-sweep)

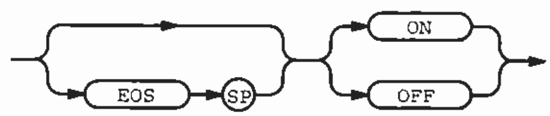


Arguments: **ON** — The spectrum analyzer asserts SRQ (if RQS is ON) when a sweep completes.

OFF — The spectrum analyzer does not assert SRQ for the EOS condition.

Query:

Response:



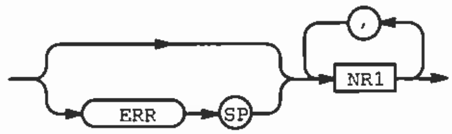
Macro Memory Used: 2 bytes.

Power-up Value: OFF.

ERR? (error)

Query:

Response:



ERR? returns any current error codes, in numerical order.

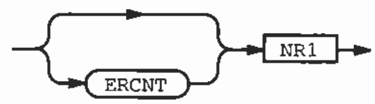
Reading the current code(s) clears the error response.

For more information, refer to the tabbed section *Status and Events*. All error codes are listed in Table 3-3.

ERCnt? (error count)

Query:

Response:



ERCNT? returns the number of error codes to be returned for an ERR query.

EVEnt? (event information)

Query:

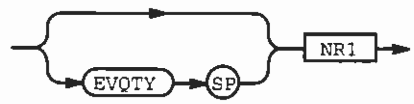
The EVEnt query returns more detailed information about the event reported in the last serial poll status byte. It also allows a controller to get information about events when the device's ability to assert RQS has been turned off (RQS OFF).

Response: NR1 is an event code defined in the tabbed section *Status and Events*, Table 3-3. The EVENT is cleared when the event code is reported.

EVQTY? (event quantity)

Query:

Response:



NR1 — This specifies the number of events that will be returned in the next ALLEV?. If the NUMEV setting is 0 and EVQTY is not executed, ALLEV? returns an unspecified number of events.

EXChg (exchange X and Y registers)

Syntax:

→(EXChg)→

The EXCHG command exchanges the current contents in X register and Y register.

Examples: The following example illustrates the use of the EXCHG command.

```
 80  Z=1 ! ADDRESS OF SPECTRUM ANALYZER
100  Print #z:"ENTER 5"
110  Print #z:"ENTER 10"
120  Print #z:"EXCHG"
130  Print#z:"STNUM 1"
140  Print#z:"VAR? 1"
150  Input#z:r
160  Print r
```

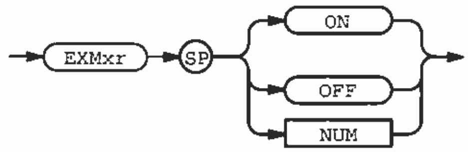
Line 100	Enters 5 into XREG.
Line 110	Moves the contents of XREG to YREG, and enters 10 in XREG.
Line 120	Enters 5 into XREG, and moves 10 into YREG.
Line 130	Stores the contents of XREG in variable number 1.
Line 150	Puts the contents of variable number 1 into r.
Line 160	Prints the contents of r (5).

Macro Memory Used: 1 byte.

EXMxr (external mixer input)

2792 Option 04 and 2794 Only

Syntax:



Arguments: **ON** — The front-panel EXTERNAL MIXER input is selected, which requires an external mixer.

OFF — The coaxial RF INPUT is selected.

Interactions: The EXTERNAL MIXER input is automatically selected for the waveguide bands (FRQRNG 6 and above) and cannot be defeated by EXMXR OFF. When active, this input bypasses the input attenuator, so it is up to the operator to prevent input overload.

Query:

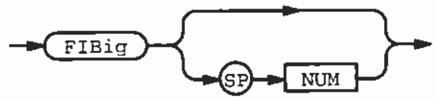
Response:

Macro Memory Used: 2 bytes.

Power-up Value: OFF.

FIBig (find big)

Syntax:



This command seeks to acquire the largest signal peak with a point of greater value than NUM and sets the display data point to the peaks. A pattern recognition routine is used to recognize signals.

If a signal peak greater than NUM is not found, the display data point is set to 500,0.

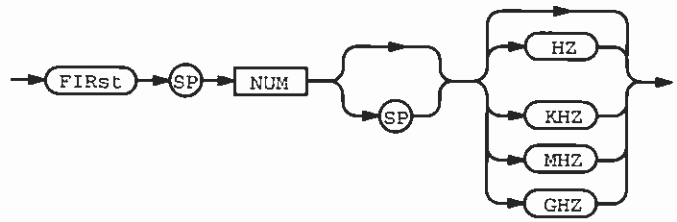
If NUM is omitted from the command, a default value of 0 is used.

Interactions: Refer to "Waveform Data and Command Interactions" on page 2-20.

Macro Memory Used: 2 bytes.

FIRst (1ST LO frequency)

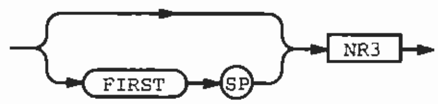
Syntax:



Arguments: NUM — The instrument 1ST LO is set to the specified frequency. The resulting center frequency will be displayed on the CRT.

Query:

Response:



Range: Refer to Table 2-21, under "TUNE".

Interactions:

Examples:

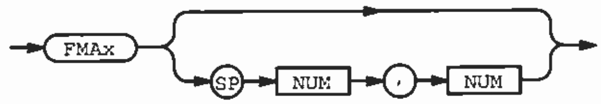
```
FIR 2.8 GHZ
FIRst 2.8 GHZ
```

Macro Memory Used: 6 bytes.

Power-up Value: 2072 MHz.

FMAx (find maximum value)

Syntax:



This routine sets the display data point to the point in digital storage with the largest Y value. If the largest Y value is located at more than one point, the first (left-most) point is acquired.

Arguments: NUM — The optional arguments are two display X values. The FMAX command will limit its search over this X range; otherwise, the full X range (1 to 1000) will be searched.

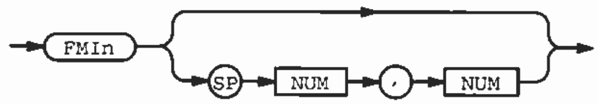
Range: 1 to 1000.

Interactions: Refer to "Waveform Data and Command Interactions" on page 2-20.

Macro Memory Used: 5 bytes.

FMIn (find minimum value)

Syntax:



This routine sets the display data point to the point in digital storage with the smallest Y value. If the smallest Y value is located at more than one point, the first (left-most) point is acquired.

Arguments: NUM — The optional arguments are two display X values. The FMIN command will limit its search over this X range; otherwise, the full X range (1 to 1000) will be searched.

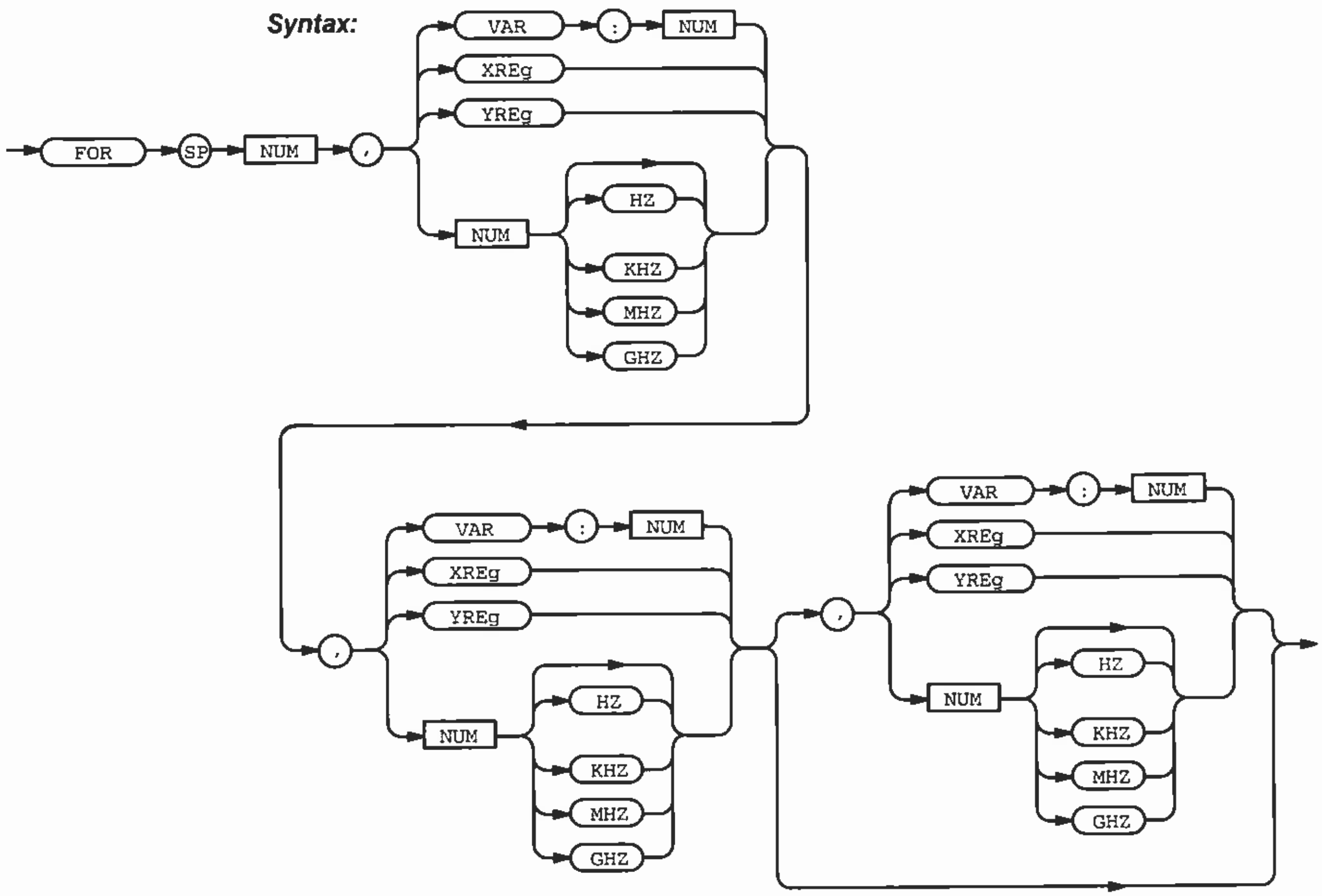
Range: 1 to 1000.

Interactions: Refer to "Waveform Data and Command Interactions" on page 2-20.

Macro Memory Used: 5 bytes.

FOR (variable = X to Y step Z), and NEXt

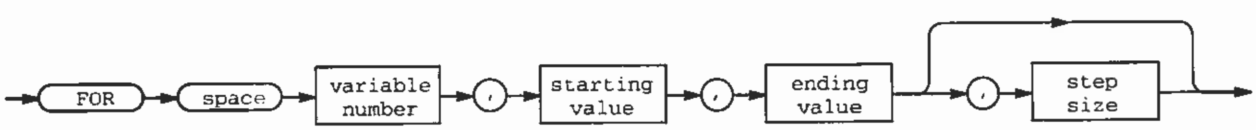
Syntax:



Arguments: **NUM** — Identifies the variable used as a loop index.

NUM — The second argument indicates the starting variable value, the third argument indicates the limiting value, and the (optional) fourth argument indicates the step size. The step size is 1 if the fourth argument is not set.

The following is a simplified representation of the FOR command syntax diagram.



If the ending value and/or the step size is taken from the X register, the Y register or a variable, and the value changes, during execution of the FOR statement, there is no effect. The value(s) at the start of the loop are used.

Range: NUM is 1 to 30.
Arguments 2, 3, and 4 are ±549,755,813,877 (maximum value for a 5-byte hexadecimal number).

Interactions: The FOR command requires a NEXT command. If the NEXT is missing, macro execution error message 172 is issued after the EMAC command has compiled the macro and checked for errors.

Examples:

`FOR 1,1,10`	(For variable(1)=1 to 10)
`FOR 1,1,10,2`	(For variable(1)=1 to 10 step 2)
`FOR 8,YREG,XREG`	(For variable(8)=YREG to XREG)
`FOR 2,VAR:1,10,5`	(For variable(2)=VAR(1) to 10 step 5)

The following example illustrates the use of the FOR command.

CAUTION

This program will delete any program stored in macro location 7.

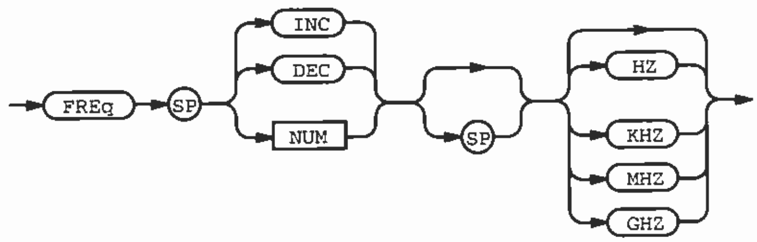
```
 70   Z=1 ! ADDRESS OF SPECTRUM ANALYZER
 80   Print #z:"KILL 7"
 90   Print #z:"STMAC 7,'FOR TESt'"
100   Print #z:"FOR 1,100MHZ,500MHZ,100MHZ"
110     Print #z:"ENTER VAR:1"
120     Print #z:"PUTREG FREQ;MFBIG;PAUSE 5"
130   Print #z:"NEXT"
140   Print #z:"DONE"
150   Print #z:"EMAC"
```

Line 100	For variable(1)=100 MHz to 500 MHz in 100 MHz steps.
Line 110	Enters the value of variable number into XREG.
Line 120	Sets the center frequency to the value in XREG, moves the marker to the biggest signal, and waits 5 seconds before the next macro command is executed.
Line 130	Repeats until variable(1) is greater than 500 MHz.

Macro Memory Used: FOR is 20 bytes, and NEXT is 5 bytes.

FREq (center frequency)

Syntax:



Arguments: **INC** — Increments the frequency as determined by the STEP command, if set.

DEC — Decrements the frequency as determined by the STEP command, if set.

NUM — The instrument centers its span about the value in the command argument. The range of values and resolution of the instrument response are the same as for front-panel operation.

Range: The frequency range varies by spectrum analyzer model:

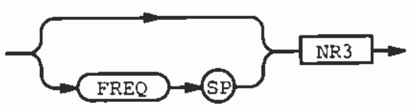
■ 2792	0 to 21 GHz
■ 2792 Option 04	0 to 27 GHz
■ 2794	0 to 325 GHz
■ 2795	0 to 1.8 GHz
■ 2797	0 to 7.1 GHz

Examples:

```
FREQ 200MHZ
FREQ 100000
FRE 200 MHZ
```

Query:

Response:

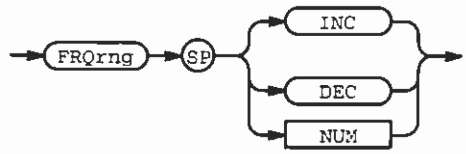


Macro Memory Used: 6 bytes.

Power-up Value: 0 MHz.

FRQrng (frequency range)

Syntax:



Arguments: **INC** — The instrument changes to the next higher frequency range, if possible.

DEC — The instrument changes to the next lower frequency range, if possible.

NUM — The instrument accepts number arguments (in a range determined by model) and changes the frequency range accordingly.

Non-integer values are rounded.

If the number is too large or too small, the spectrum analyzer maintains its current frequency range and reports execution error message 29.

If a value of 0 is used, no action is taken.

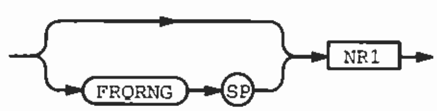
Range:

- 2792 — 1 to 5
- 2792 Option 04 — 1 to 6
- 2794 — 1 to 12
- 2795 — 1
- 2797 — 1 to 3

Interactions: The instrument automatically selects the frequency closest to that already in use that encompasses the frequency setting that responds to the FREQ command. In option 07 instruments when using 75 Ω input, if the requested frequency range is outside the allowable limits (if you send anything except `FRQRNG 1`), execution error message 102 is issued.

Query:

Response:



Macro Memory Used: 2 bytes.

Power-up Value: Frequency Range 1.

GETwfm (get waveform)

Syntax: GETwfm

The GETWFM command will get the current waveform (A&B) and store it in the array DISBUF. The purpose of the GETWFM command is to load the waveform so you can use the ENTer DISBUF command.

The Macro Example below finds the highest point in the display and works like the PKFIND command. The highest point is left in XREG.

Examples: The following example illustrates the use of the GETWFM command.

CAUTION

This program will delete any program stored in macro location 7.

```
 70   Z=1 ! ADDRESS OF SPECTRUM ANALYZER
 80   Print#z:"KILL 7"
 90   Print#z:"STMAC 7,'GETWFM TEST'"
100   Print#z:"GETWFM"
110   Print#z:"ENTER 0"
120   Print#z:"FOR 1,1,1000"
130       Print#z:"ENTER DISBUF:VAR:1"
140       Print#z:"IF YREG,GRT,XREG"
150       Print#z:"POP"
160       Print#z:"END I"
170   Print#z:"NEXT"
180   Print#z:"DONE"
190   Print#z:"EMAC"
```

Line 100	Gets the current waveform.
Line 110	Initializes the maximum value.
Line 120	For variable(1)=1 to 1000.
Line 130	Enters DISBUF (VAR(1)).
Line 140	If YREG (old maximum value) is greater than XREG.
Line 150	Puts higher value into XREG.

Macro Memory Used: 1 byte.

GOSub (go to LABeled subroutine)

Syntax: → GOSub → SP → NUM →

Arguments: NUM — Macro control is transferred to the LABELED subroutine specified by NUM. The macro will return to the next command following the GOSUB when the macro finds a RETURN statement.

If there are more than 20 nested GOSUB before a RETURN, macro execution error message 167 will be issued, and the macro will be aborted.

Range: 1 to 100.

Examples:

The following example illustrates the use of the GOSUB command.

CAUTION

This program will delete any program stored in macro location 7.

```
 70   Z=1 ! ADDRESS OF SPECTRUM ANALYZER
 80   Print #z:"KILL 7"
 90   Print #z:"STMAC 7,'GOSUB TEST'"
100   Print #z:"FREQ 100M"
110   Print #z:"GOSUB 1"
120   Print #z:"FREQ 200M"
130   Print #z:"GOSUB 1"
140   Print #z:"FREQ 300M"
150   Print #z:"GOSUB 1"
160   Print #z:"DONE"

170   Print #z:"LABEL 1"
180       Print #z:"SWEEP;MFBIG;PAUSE 5"
190       Print #z:"RETURN"
200   Print #z:"EMAC"
```

Lines 100 and 110	Sets the center frequency to 100 MHz, and sends macro control to LABEL 1.
Lines 120 and 130	Sets the center frequency to 200 MHz, and sends macro control again to LABEL 1.
Lines 140 and 150	Sets the center frequency to 300 MHz, and sends macro control again to LABEL 1.

Line 180	Takes a sweep, moves the marker to the maximum signal, and waits 5 seconds before the next macro command is executed.
Line 190	The macro returns to the line following the last GO-SUB. When the macro goes up to line 160, the macro is through running.

Macro Memory Used: 5 bytes.

GOTo (go to LABeled line)

Syntax:

If the macro is instructed to go to a label that does not exist, macro execution error message 170 is issued. Error message 170 will only be seen after the EMAC command when the macro is compiled and errors are located.

NOTE

The last command in a macro before the EMAC command must be GOTO or RETURN or DONE. If it is not, macro execution error message 178 is issued.

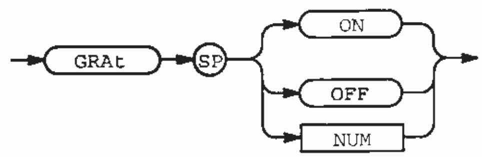
Arguments: NUM — Tells the macro which LABEL to go to and continue execution.

Range: 1 to 100.

Macro Memory Used: 5 bytes.

GRAt (graticule)

Syntax:

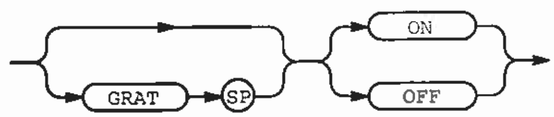


Arguments: **ON** — The CRT graticule light is turned on.

OFF — The CRT graticule light is turned off.

Query:

Response:

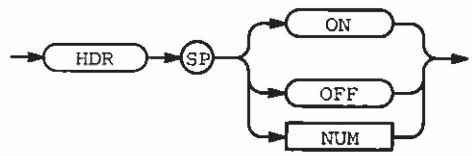


Macro Memory Used: 2 bytes.

Power-up Value: OFF.

HDR (header)

Syntax:



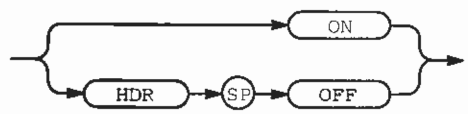
Arguments: **ON** — The header for query responses is turned on.

OFF — The header for query responses is turned off.

Interactions: The HDR command has no affect on the SET? response, since the headers are necessary to interpret the response.

Query:

Response:



Macro Memory Used: 2 bytes.

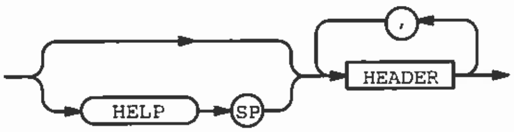
Power-up Value: ON.

HELp? (help)

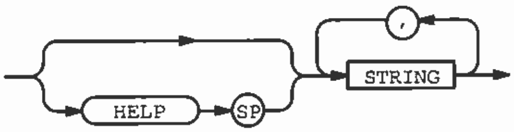
Query:

Response:

With No Argument



The response is a list of all command headers in the GPIB language.



The string will report that this instrument has no help text.

HPOs (horizontal position)

Syntax:

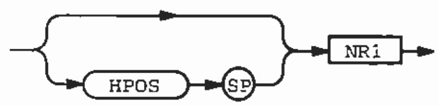
This command sets the CRT horizontal display adjustment to a position corresponding to a DAC value specified by NUM.

Arguments: NUM — is the value loaded into the horizontal position DAC.

Range: From 0 to 255, where 0 corresponds to the left-most display position and 255 the right-most.

Query:

Response:



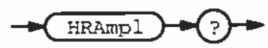
Power-up Value: Last front panel setting.

HRAmpl (next higher amplitude)

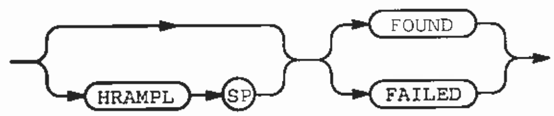
Syntax:

The HRAMPL command moves the primary marker to the next higher amplitude signal on the display. If the marker is on the highest signal on the display or if no signal is found, the marker does not move.

Query:



Response:



FOUND is returned if the last HRAMPL command found a signal.

FAILED is returned if the last HRAMPL command did not find a signal.

If the HRAMPL query is given before any HRAMPL command, FAILED is returned.

Interactions: If SGERR is ON, marker execution warning message 130 is issued if a signal is not found. If MARKER is OFF, HRAMPL sets MARKER to SINGLE.

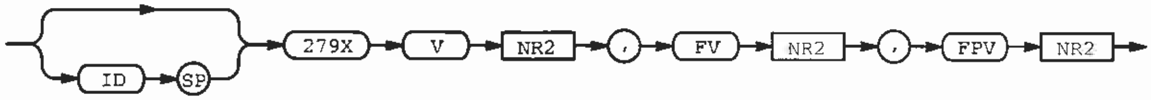
The criteria for a signal are set by the THRHLD and STYPE commands.

Macro Memory Used: 1 byte.

ID? (identify)

Query:

Response:



279x — The instrument type number. If the instrument is in 49X compatibility mode, the 490 series instrument most closely resembling the instrument is returned.

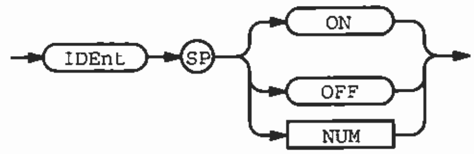
V<NR2> — Tektronix Interface Standard for GPIB Codes, Formats, Conventions, and Features version number.

FV<NR2> — Instrument firmware version number.

FPV<NR2> — Front-panel processor firmware version number.

IDEnt (identify)

Syntax:



Arguments: ON — The signal identify function is turned on. Spurious conversion products are shifted horizontally on alternate traces. The trace is also offset vertically on alternate sweeps so true signals stand out.

OFF — The signal identify function is turned off.

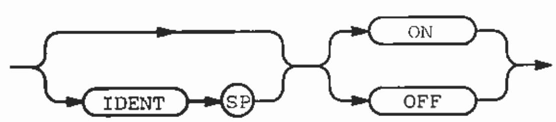
Interactions: The span must be 500 kHz or less in bands 1–5 (500 MHz or less in bands 6–12) for the identify function to operate.

The SIGSWP commands cause alternating normal and vertically offset sweeps; the first sweep is normal, the next offset, and so on.

If SGTRAK is on, execution error message 101 is issued if IDENT is ON and the command is not executed.

Query:

Response:

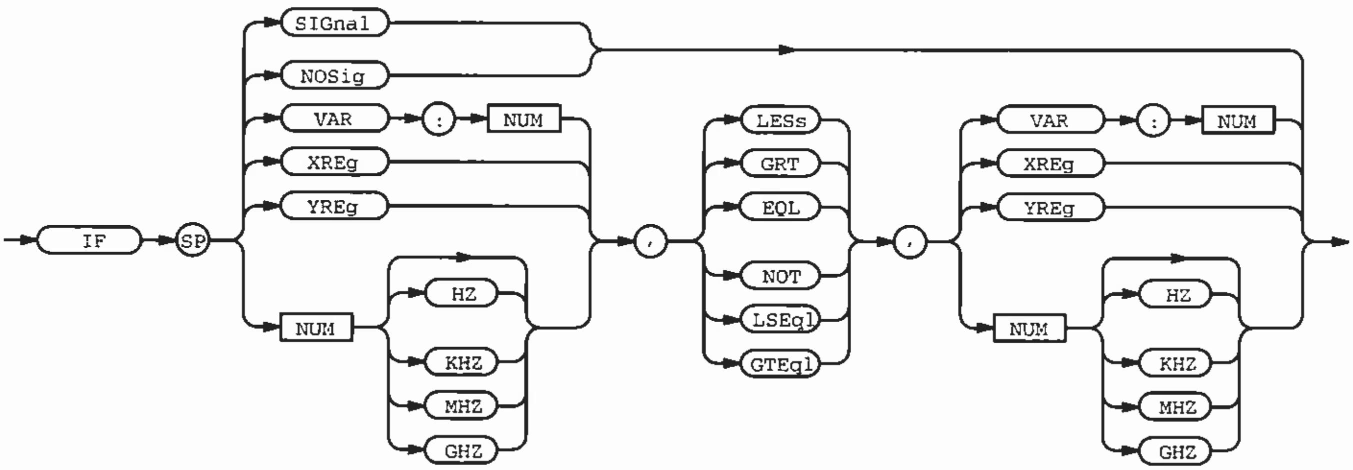


Macro Memory Used: 2 bytes.

Power-up Value: OFF.

IF (if statement is true), also ELSe and ENDi

Syntax:



Arguments: **SIGNAL** — True if the last marker signal find command found a signal. The marker signal find commands are MLFTNX, MFGTNX, MFBIG, HRAMPL, LRAMPL, PKFIND, MVRTDB, and MVLTDB.

NOSIG — True if the last marker signal find command did not find a signal. The marker signal find commands are MLFTNX, MRGTNX, MFBIG, HRAMPL, LRAMPL, PKFIND, MVRTDB, and MVLTDB.

VAR — Compares the contents of a variable to either another variable, the X register, the Y register, or a number.

XREG — Compares the contents of the X register to either a variable, the X register, the Y register, or a number.

YREG — Compares the contents of the Y register to either a variable, the X register, the Y register, or a number.

NUM — Compares a number to either a variable, the X register, the Y register, or another number.

Table 2-16: IF Comparators

Argument	Symbol	Description
LESS	<	IF less than
GRT	>	IF greater than
EQL	=	IF equal to
NOT	< >	IF not equal to
LSEQL	≤	IF less than or equal to
GTEQL	≥	IF greater than or equal to

Examples: The following example illustrates the use of the IF command.

CAUTION

This program will delete any program stored in macro location 7.

```
 70   Z=1 ! ADDRESS OF SPECTRUM ANALYZER
 80   Print#z:"KILL 7"
 90   Print#z:"STMAC 7,'SIGNAL TEST'"
100   Print#z:"SWEEP;MFBIG"
110   Print#Z:"CLEAR 2"
120   Print#z:"IF SIGNAL"
130       Print#z:"PRINT 2,10,'TEST PASSED'"
140   Print#z:"ELSE"
150       Print#z:"PRINT 2,10,'TEST FAILED'"
160   Print#z:"ENDI"
170   Print#z:"DSLINE TOP,2,BOTTOM"
180   Print#z:"DONE"
190   Print#z:"EMAC"
```

Line 90	Stores the following GPIB commands as macro number 7 (in the macro menu) and titles it SIGNAL TEST. The commands will not be executed, just stored in memory.
Line 100	Takes a sweep, and the marker finds the biggest signal.
Line 110	Clears line 2 of the macro readout buffer.
Line 120	If a signal is found, the test is passed.
Line 130	Prints the test passed message to the buffer. More than one statement can follow the IF or ELSE commands.
Line 140	If a signal is not found, the test failed.
Line 150	Prints the test failed message to the buffer. More than one statement can follow the IF or ELSE commands.

Line 160	This is the end of the IF command.
Line 170	Displays the normal top and bottom lines and prints the test results on line 2.

Interactions: The IF command requires an ENDI (end of the IF statement) command. If the ENDI is missing, macro execution error message 174 is issued and macro entry is aborted.

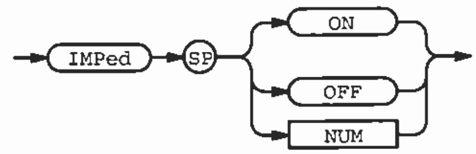
The ELSE command is optional with the IF command.

Macro Memory Used: IF is 23 bytes, ELSE is 4 bytes, and ENDI is 1 byte.

IMPed (impedance)

Option 07 Only

Syntax:

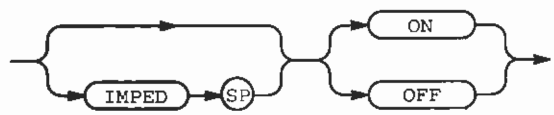


Arguments: **ON** — The 75 Ω input is used. If Option 07 is not installed, command error message 8 will be issued.

OFF — The 50 Ω input is used.

Query:

Response:



Macro Memory Used: 2 bytes.

Power-up Value: OFF.

INIt (initialize settings)

Syntax:

INIT resets the instrument as if the power was turned off, then turned back on. For the power-up status of any command, refer to the command description.

Interactions: IEEE 488 interface functions are not affected and the instrument remains under remote control.

RQS is set to OFF if either the LISTEN ONLY or TALK ONLY switch is set.

Macro Memory Used: 1 byte.

INTegr (convert X register to an integer)

Syntax: → (INTeger) →

The INTEGR command truncates the number currently stored in XREG into an integer.

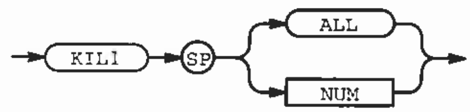
Examples: XREG Contents:

Before INTEGR	**After INTEGR**
123.45	123
1.96432 E+2	196
4.83723 E+20	4.83723 E+20 (no effect)

Macro Memory Used: 1 byte.

KILI (delete macros)

Syntax:



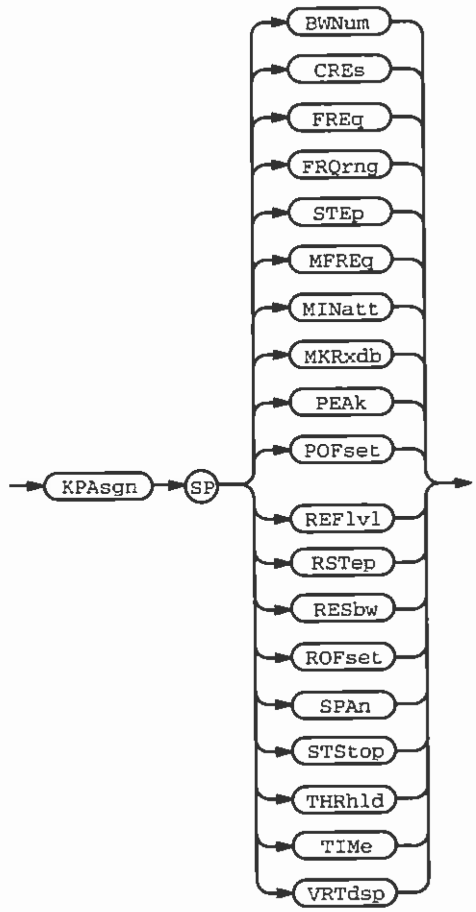
A macro cannot be changed once it is stored in memory; it must be deleted (KILL NUM) and completely re-entered.

Arguments: **ALL** — All macros will be deleted from memory.

NUM — Only macro NUM will be deleted from memory.

KPAsgn (keypad assign)

Syntax:



This command assigns the keypad to enter a value for any of the permitted arguments.

Arguments: The arguments are shown in the syntax diagram. Table 2-17 lists the arguments, a brief description of the entry, and the duration of the assignment. Assignment labeled "until changed" persist until the keypad is explicitly reset to some other assignment. Assignments labeled "temporary" revert to the last fixed assignment before KPASGN was issued when the entry is complete.

Table 2-17: Keypad Assignment

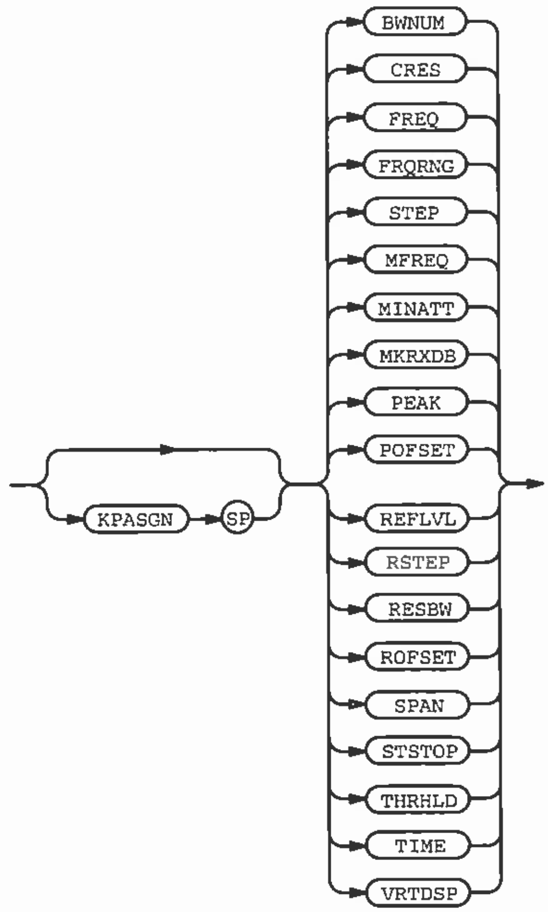
Argument	Keypad Entry	Assignment Duration
BWNUM	marker bandwidth level	temporary
CRES	counter resolution	temporary
FREQ	center frequency	until changed
FRQRNG	frequency range	temporary

Table 2-17: Keypad Assignment (Cont.)

Argument	Keypad Entry	Assignment Duration
MFREQ	marker frequency	until changed
MINATT	minimum RF attenuation	until changed
MKRXDB	move marker xdB	temporary
PEAK	preselector/external mixer peaking	until changed
POFSET	plotter B minus A offset	temporary
REFLVL	reference level	until changed
RESBW	resolution bandwidth	until changed
ROFSET	reference level offset	temporary
RSTEP	reference level step size	temporary
SPAN	frequency span	until changed
STSOP	start/stop frequency	temporary
STEP	frequency step size	temporary
THRHLD	marker threshold	temporary
TIME	sweep time	until changed
VRTDSP	vertical scale	until changed

Query:

Response:



Macro Memory Used: 2 bytes.

Power-up Value: FREQ.

LABel (label point in macro)

Syntax:

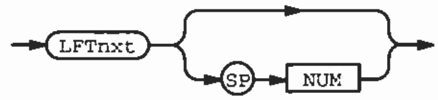
Arguments: NUM — Sets the reference point for the GOTO and GOSUB commands in a macro.

Range: 1 to 100.

Macro Memory Used: 2 bytes.

LFTnxt (left next)

Syntax:



This command searches to the left of the current point to acquire the peak of a signal whose value is greater than NUM. A pattern recognition routine is used to recognize signals. The display data point is set to the peak.

If no signal peak greater than NUM is found, the display data point is set to 0,0.

If NUM is omitted from the command, a default value of 0 is used.

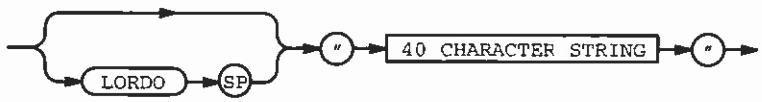
Interactions: Refer to "Waveform Data and Command Interactions" on page 2-20.

Macro Memory Used: 2 bytes.

LORdo? (lower readout)

Query:

Response:



This query returns the bottom line of the readout information that appears on the instrument CRT.

CHARACTER — Characters are from the CRT readout. Blanks are transmitted as spaces.

Regular readout that would be displayed if GPIB did not have control (whether visible on the screen or not) is the readout returned by the query, not a message sent to the instrument by RDOUT.

With both AVIEW and SAVEA ON and both BMINA and BVIEW OFF, the returned readout will be the saved readout.

Refer to the recall display (DRECAL) or the save A waveform (SAVEA) commands for more information.

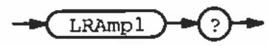
LRAmpl (next lower amplitude)

Syntax:

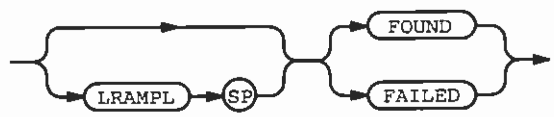
The LRAMPL command moves the primary marker to the next lower amplitude signal on the display.

If the marker is on the lowest signal on the display or a signal cannot be found, the marker does not move.

Query:



Response:



FOUND is returned if the last LRAMPL command found a signal.

FAILED is returned if the last LRAMPL command did not find a signal.

If the LRAMPL query is given before any LRAMPL command, FAILED is returned.

Interactions: If SGERR is ON, marker execution warning message 130 is issued if a signal is not found.

If MARKER is OFF, LRAMPL sets MARKER to SINGLE.

The criteria for a signal are set by the THRHLD and STYPE commands.

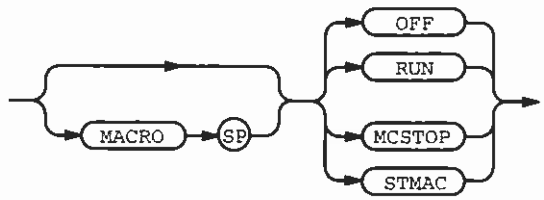
Macro Memory Used: 1 byte.

MACro? (macro status)

Query:

The MACRO query will return the current macro status.

Response:



OFF — There is no macro action.

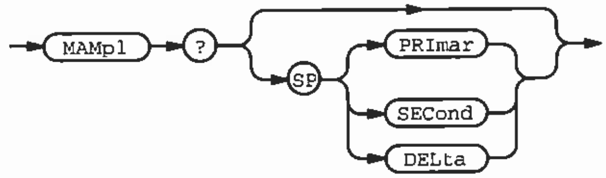
RUN — There is a macro running.

MCSTOP — There is a macro stopped waiting for the RUN/STOP key to be pressed or the RUN command to be sent.

STMAC — A macro is being stored in memory.

MAMpl? (marker amplitude)

Query:



MAMPL? or **MAMPL PRIMAR** — The amplitude of the primary marker is returned.

MAMPL PRIMAR — The amplitude of the secondary marker is returned.

MAMPL DELTA — The amplitude of the primary marker with respect to the secondary marker is returned.

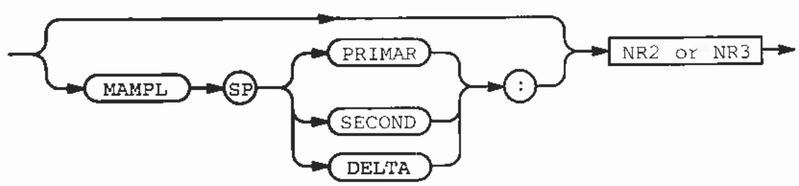
NOTE

If the marker whose amplitude is being requested (or, in the case of MAMPL? DELTA, the amplitude of either marker) is out of the range of digital storage, one of the following will be returned.

- *−200.0 if under-range*
- *+200.0 if over-range*
- *+999.9 if markers are off*

Interactions: The amplitude is returned in the current reference level units if in a log display mode, or in volts if in a linear display mode. If the frequency of the secondary marker is off-screen, MLOCAT? SECOND and MLOCAT? DELTA use the last known secondary marker amplitude.

Response:



NR2 is returned in the Log Mode, and NR3 is returned in the Lin Mode.

MAMPL is not included in the response to SET?.

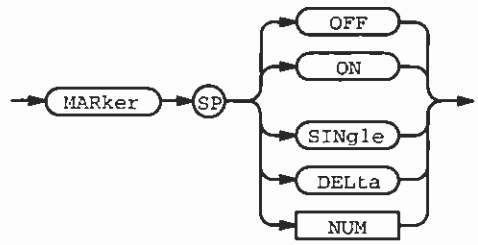
Interactions: If HDR is OFF, PRIMAR, SECOND, or DELTA and the delimiter : are eliminated along with the MAMPL header.

Examples:

```
MAMPL?
MAM? SEC
```

MARker (marker mode)

Syntax:



Arguments: **ON** or **SINGLE** — The single-marker is turned on.

OFF — The marker is turned off.

DELTA — The delta-marker is turned on.

NUM —
0 = OFF
1 = ON or SINGLE
2 = DELta

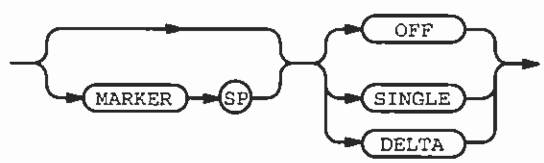
Interactions: MARKER SINGLE or ON or MARKER DELTA sets TMODE to MARKER.

MARKER OFF sets TMODE TO FREQ.

MARKER SINGLE, or MARKER DELTA are selected by most other marker commands.

Query:

Response:

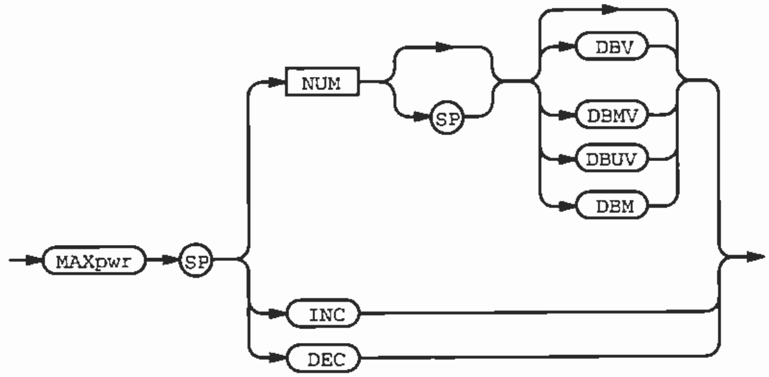


Macro Memory Used: 2 bytes.

Power-up Value: OFF.

MAXpwr (maximum input power)

Syntax:



Arguments: **NUM** — This argument corresponds to the maximum power expected at the spectrum analyzer RF input. Setting MAXPWR appropriately protects the RF input from overload at the expected maximum power level.

The instrument selects a minimum RF attenuation so that the NUM signal level is reduced to no more than −23 dBm at the 1st Mixer.

The maximum non-destructive power level that can be connected to the RF input is +30 dBm. If no units are specified, the instrument assumes the current reference level units. If the number selected is out of range, execution error message 33 is issued.

NOTE

To ensure the correct response, all of the letters in each of the units mnemonics for the MAXPWR command must be entered; not just the first three letters as for other mnemonics.

INC — The minimum RF attenuation is changed to the next higher step, if any.

DEC — The minimum RF attenuation is changed to the next lower step, if any.

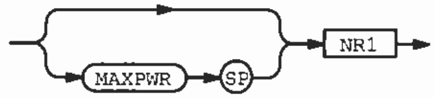
Interactions: The range of RF attenuation is limited in response to the MAXPWR command, which limits the range of the REFLVL command. MAXPWR cancels the previous limit set by either MINATT or MAXPWR.

Examples:

```
MAXPWR +20DBMV
MAX 18 DBUV
MAXPWR DEC
```

Query:

Response:



Only the number value will be returned; the units will not be indicated (the units will be the current reference level units).

Macro Memory Used: 4 bytes.

Power-up Value: −18 to +42; dependent on MINATT value (−18 + MINATT value).

MCEn (marker to center)

Syntax: MCEn

The marker is moved to center screen and the marker frequency becomes the new center frequency.

Marker execution error message 121 is issued if the marker is not on an active trace.

Interactions: If MARKER is OFF, MCEN sets MARKER to SINGLE. In this case, since the primary marker appears at the center of the screen, the center frequency does not change.

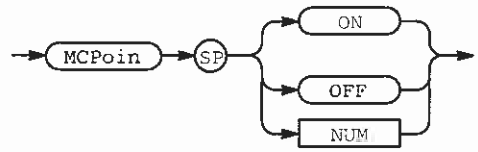
When counting at the marker, it is the counted frequency that is tuned to center.

MCEN is not available if ZEROSP or MXSPN is ON.

Macro Memory Used: 1 byte.

MCPoin (marker coupled to the display pointer)

Syntax:



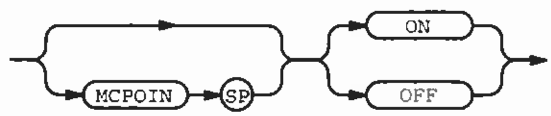
This command affects the movement of the internal (invisible) display pointer.

Arguments: **ON** — The display pointer tracks the primary marker.

OFF — The display pointer does not track the primary marker.

Query:

Response:



Interactions: The WFID portion of any previous WFMPRE command or the CRVID portion of any previous CURVE command is cancelled, and the FULL waveform for data transfers and waveform processing is selected.

A WFID or CRVID other than FULL sets MCPOIN to OFF.

If MARKER is OFF, MCPOIN sets MARKER to SINGLE.

Macro Memory Used: 1 byte.

Power-up Value: OFF.

MCStop (macro stop)

Syntax:

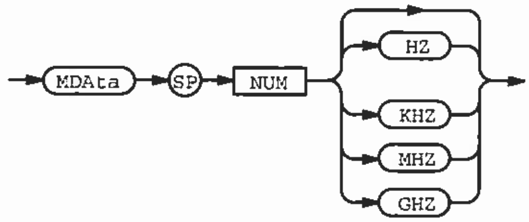
```
→( MCStop )→
```

The MCSTOP command stops macro execution. The macro can be restarted by pressing [RUN/STOP] on the front panel.

Macro Memory Used: 1 byte.

MDAta (store numeric data)

Syntax:



Arguments: NUM — The numeric data to be used with the READ command is stored.

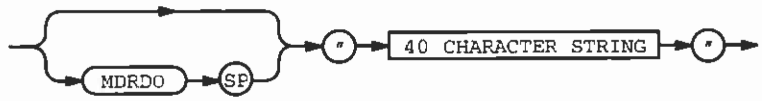
The program with the MRESTO command description illustrates the use of the MDATA command also.

Macro Memory Used: 9 bytes.

MDRdo? (middle readout)

Query:

Response:



This query returns the middle line of readout information that appears on the instrument CRT. This line of readout normally represents marker amplitude and frequency.

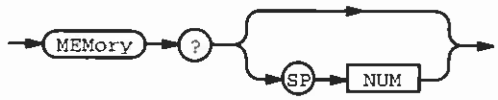
Arguments: **CHARACTER** — Characters are from the second row of regular CRT readout. Blanks are transmitted as spaces.

Regular readout that would be displayed if GPIB did not have control (whether visible on the screen or not) is the readout returned by the query, not a message sent to the instrument by RDOUT.

Refer to the recall display (DRECAL) or the Save A waveform (SAVEA) commands for more information.

MEMory? (memory used)

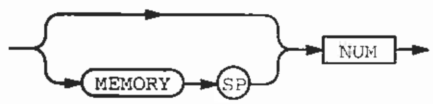
Query:



The MEMORY query serves two separate purposes.

- MEMORY query with no argument returns the amount of memory remaining for storing macros (from the original 8 kbytes reserved for macros).
- MEMORY query with a number argument returns the amount of memory that macro number NUM is using.

Response:



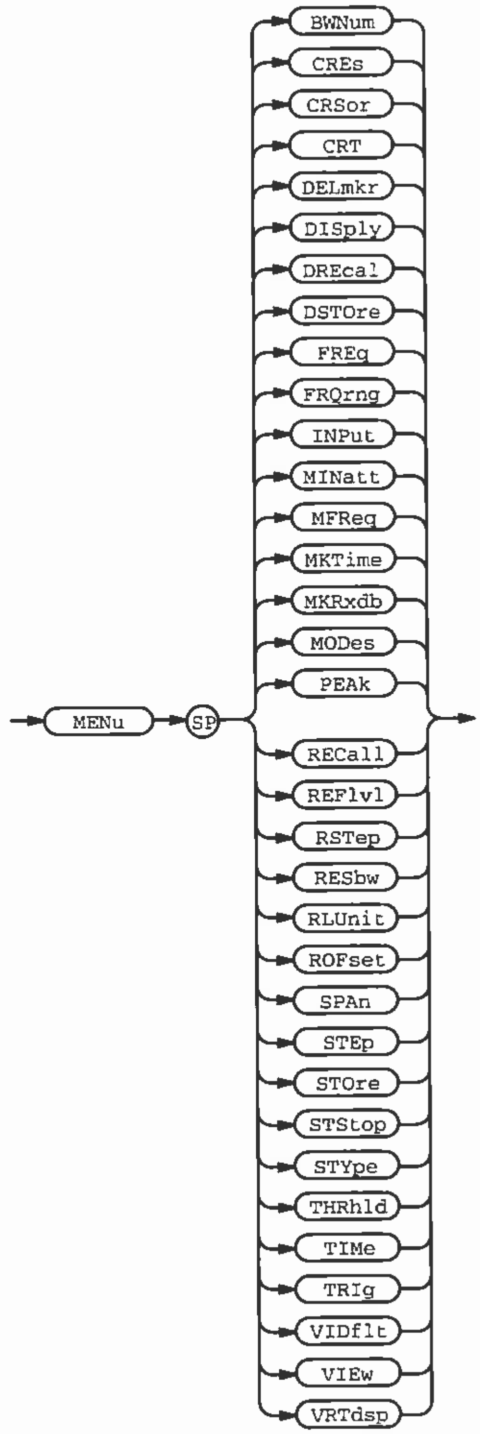
If a number argument that is out of range (0–7) is sent, execution error message 106 is issued and 0 is returned.

If a macro is not stored at the NUM location, macro execution error 165 is issued and 0 is returned.

Range: 0 to 7.

MENu (display menu)

Syntax:



The MENU command will display the requested menu.

During the MENU command, macros or GPIB execution suspends and the front panel is active.

Macro or GPIB execution continues with the front panel locked out when all menus are cleared from the screen by the user.

For a description of the command arguments, see the description of the command whose mnemonic is the same as the argument.

Arguments: Table 2-18 lists the arguments for the MENU command. The "Menu Called" column identifies which menu is called by the argument; the "Keypad Entry" column specifies the keypad entry assignment. If no entry assignment is given, the menu takes input from the soft keys only.

Table 2-18: MENU Command Arguments

Argument	Menu Called	Keypad Entry
`BWNUM`	MARKERS/ BW MARKER	Set BW Level
`CRES`	COUNTER	Set Resolution
`CRSOR`	WF VIEW/PK/AVG	
`CRT`	DISPLAY/CRT ADJUST	
`DELMKR`	DELTA MARKER	
`DISPLY`	DISPLAY	
`DRECAL`	WF RECALL	Register Number
`FREQ`	FREQUENCY	Center Frequency
`FQRRNG`	FREQUENCY	Frequency Range
`INPUT`	INPUT	
`MINATT`	MINATTEN	Minimum Attenuation
`MFREQ`	FREQUENCY	Center Frequency
`MKTIME`	MARKERS	Marker TIMe
`MKRXDB`	MARKERS/ MARKER XDB	Set Marker XdB value
`MODES`[1]	UTIL/ SPEC MODES	
`PEAK`	INPUT/ PEAK	Peak Value
`RECALL`	SET RECALL	
`REFLVL`	REF LEVEL	Reference Level
`RESBW`	RES BW	Resolution Bandwidth
`RLUNIT`	REF LEVEL/ UNITS	

Table 2-18: MENU Command Arguments (Cont.)

Argument	Menu Called	Keypad Entry
ROFSET	REF LEVEL	Ref Level Offset
RSTEP	REF LEVEL	Step Size
SPAN	SPAN	Span
STEP	FREQUENCY/ STEP SIZE	Set Step in Hz
STORE	SET STORE	
STSTOP	REQUENCY	Start/Stop Frequency
STYPE	MAKERS/ PEAK	
THRHLD	MARKERS/ PEAK/ THRESHOLD	Set Level
TIME	SWEEP	Sweep Speed
TRIG	TRIGGER	
VIDFLT	VIDEO BW	
VIEW	VIEW	
VRTDSP	REF LEVEL/ VERT SCALE	Vertical Scale

1 **Although the UTIL menu is not a valid MENU argument, the second-level menu MODES is accessible by this command. In addition to UTIL, two other menus cannot be called by the MENU command: CONFIG and PROGRAM.**

Macro Memory Used: 2 bytes.

MEXchg (marker exchange)

Syntax: →(MEXchg)→

The primary marker moves to the former location of the secondary marker, and the secondary marker moves to the former location of the primary marker.

If the secondary marker is off the screen before the marker exchange, the instrument center frequency will be set to the old secondary marker frequency, and the old primary marker (the new secondary marker) will be off the screen.

Interactions: MEXCHG sets MARKER to DELTA.

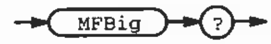
Macro Memory Used: 1 byte.

MFBig (marker peak find)

Syntax:

The MFBIG command moves the primary marker to the peak of the largest on-screen signal. If no signal peak is found, the marker does not move.

Query:



Response:

FOUND is returned if the last MFBIG command found a signal.

FAILED is returned if the last MFBIG command did not find a signal.

If the MFBIG query is given before any MFBIG command, FAILED is returned.

Interactions: If MARKER is OFF, MFBIG sets MARKER to SINGLE.

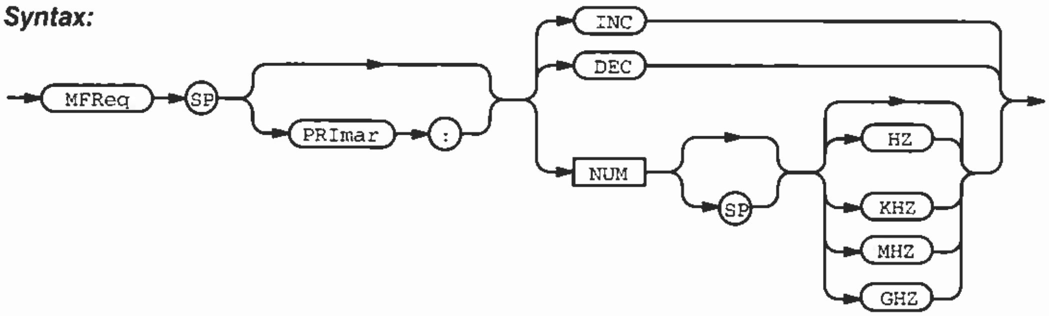
If SGERR is ON, marker execution warning message 130 is issued if a signal is not found.

The definition of the criteria for a signal is set by the THRHLD and STYPE commands.

Macro Memory Used: 1 byte.

MFReq (marker frequency)

Syntax:



The MFREQ command sets the frequency of the primary marker to the value given by NUM.

Arguments: **INC** — Increments the primary marker frequency sequence as determined by the STEP command.

DEC — Decrements the primary marker frequency sequence as determined by the STEP command.

NUM — Sets the marker frequency to the value in the command argument. The range of values and resolution of the instrument response are the same as for front-panel operation.

Range: The active trace range varies by spectrum analyzer model:

- 2792 — 0 to 21 GHz
- 2792 Option 04 — 0 to 27 GHz
- 2794 — 0 to 325 GHz
- 2795 — 0 to 1.8 GHz
- 2797 — 0 to 7.1 GHz

Inactive trace is limited to the edge of the screen

Interactions: If MARKER is OFF, MFREQ sets MARKER to SINGLE.

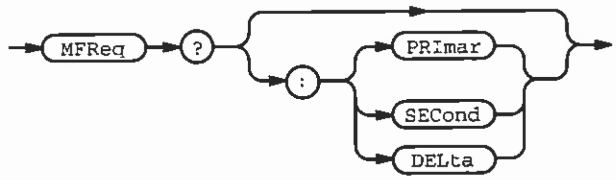
MFREQ causes marker execution error message 120 to be issued if the primary marker is on an inactive trace and the frequency is not on the screen.

MFREQ moves the primary marker to center screen and changes center frequency if the primary marker is on an active trace and the frequency is not on the screen.

Examples:

```
MFREQ 100000
MFR 1.8 GHZ
MFREQ PRIMAR:200 KHZ
MFR PRI:200MHZ
```

Query:

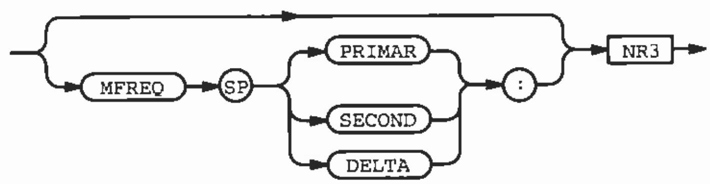


Arguments: **MFREQ?** or **MFREQ? PRIMAR** — The frequency is returned for the primary marker.

MFREQ? SECOND — The frequency is returned for the secondary marker.

MFREQ? DELTA — The frequency is returned for the primary marker with respect to the secondary marker.

Response:



Interactions: Any MFREQ? query returns 9.999999E+99 if the requested marker or delta is not on.

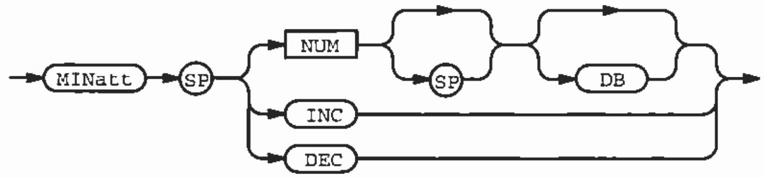
If HDR is OFF, PRIMAR, SECOND, or DELTA and the following delimiter (:) are eliminated along with the MFREQ header.

Macro Memory Used: 6 bytes.

Power-up Value: Markers are off when power is first turned on. When markers are turned on, MFREQ is set to the center frequency of the marker trace, unless the marker is on a recalled trace that had a stored marker frequency.

MINatt (minimum RF attenuation)

Syntax:



Arguments: **NUM** — The gain distribution set by the instrument is limited; RF attenuation may not be reduced below the attenuator step in the number argument.

If NUM is not an even decade from 0 to 60, the next higher step (0, 10, 20, ... 60) is selected.

If the number selected is out of range, execution error message 33 is issued.

INC — The minimum RF attenuation is changed to the next higher step, if possible.

DEC — The minimum RF attenuation is changed to the next lower step, if possible.

Range: 0 to 60.

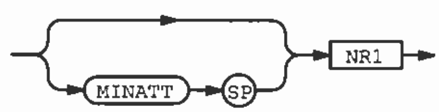
Interactions: The range of RF attenuation is limited in response to the REFLVL command, which limits the range of the REFLVL command. The previous limit set by either MINATT or MAXPWR is cancelled.

Examples:

```
MINATT 20
MIN 42 DB
MINATT INC
```

Query:

Response:



Macro Memory Used: 2 bytes.

Power-up Value: 0 dB.

MKDp (marker to display pointer)

Syntax: → MKDp →

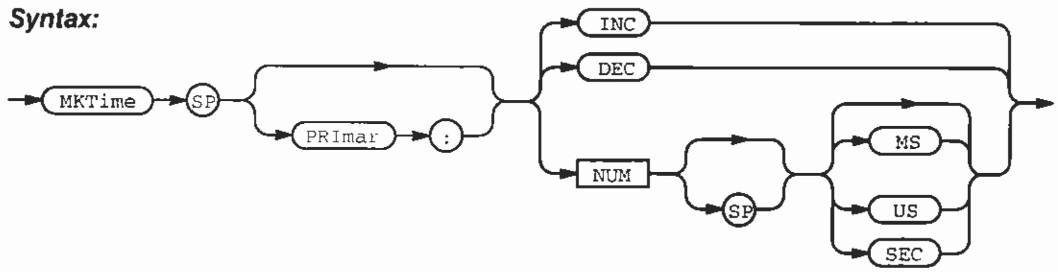
The primary marker is moved to the same horizontal location as the display pointer.

Interactions: If MARKER is OFF, MKDP sets MARKER to SINGLE.

Macro Memory Used: 1 byte.

MKTime (marker time)

Syntax:



The MKTIME command sets the time of the primary marker with respect to the trigger point (1/2 division to the left of the screen).

Arguments: **NUM** — Sets the time of the primary marker to the value specified.

INC — Increments the time of the primary marker by 1 division.

DEC — Decrements the time of the primary marker by 1 division.

Interactions: MKTIME is available only when the instrument is in the zero span mode.

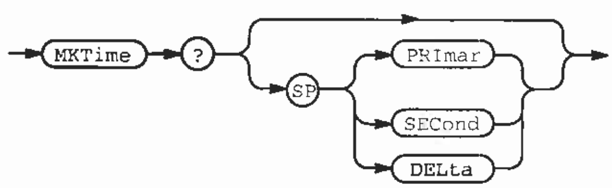
If MKTIME is used when the instrument is not in the zero span mode or ZETIME is OFF, marker execution error message 107 is issued.

An attempt to set a time that would be off either the left or right of the screen will cause marker execution error message 108 to be issued.

Examples:

```
MKTIME 1MS
MKT .1 S
```

Query:



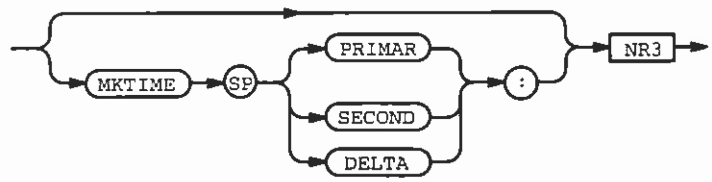
Arguments: **MKTIME?** or **MKTIME? PRIMAR** — The time is returned for the primary marker.

MKTIME? SECOND — The time is returned for the secondary marker.

MKTIME? DELTA — The time is returned for the primary marker with respect to the secondary marker.

Interactions: The MKTIME query will return −200 if the time value is unavailable, +200 if the instrument is not in the zero span mode or ZETIME is OFF, and +999.99 if the marker system is off (or MARKER is not set to DELTA when the secondary or delta time is requested).

Response:



Interactions: If HDR is OFF, PRIMAR, SECOND, or DELTA and the following delimiter (:) are eliminated along with the MKTIME header.

Macro Memory Used: 6 bytes.

Power-up Value:

- Markers are off when power is first turned on.
- When markers are turned on in zero span, or the instrument is set to zero span with the markers on.
- ZETIME is set to ON.
- The markers are placed at center screen.
- The time value is set accordingly.

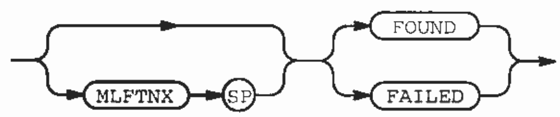
MLFtnx (marker left next)

Syntax:

The MLFTNX command moves the primary marker to the peak of the next signal to the left of the present marker position. If no signal peak is found, the marker does not move.

Query:

Response:



FOUND is returned if the last MLFTNX command found a signal.

FAILED is returned if the last MLFTNX command did not find a signal.

If the MLFTNX query is given before any MLFTNX command, FAILED is returned.

Interactions: If MARKER is OFF, MLFTNX sets MARKER to SINGLE.

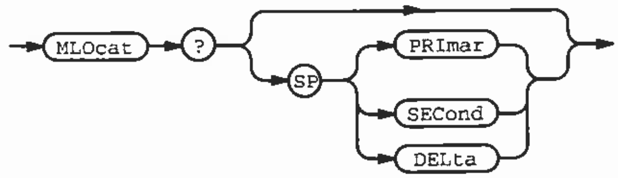
If SGERR is ON, marker execution warning message 130 is issued if a signal is not found.

The definition of the criteria for a signal is set by the THRHLD and STYPE commands.

Macro Memory Used: 1 byte.

MLOcat? (marker location)

Query:

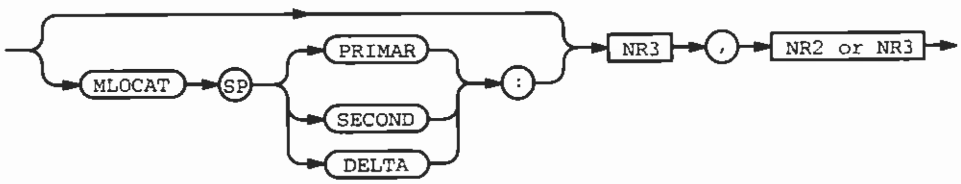


Arguments: **MLOCAT?** or **MLOCAT? PRIMAR** — The amplitude and frequency are returned for the primary marker.

MLOCAT? SECOND — The amplitude and frequency are returned for the secondary marker.

MLOCAT? DELTA — The amplitude and frequency are returned for the primary marker with respect to the secondary marker.

Response:



Interactions: In LOG, the amplitude is returned in reference level units (NR2).

In LIN, the amplitude is returned in volts (NR3).

If the frequency of the secondary marker is off-screen, MLOCAT? SECOND and MLOCAT? DELTA use the last known secondary marker amplitude.

NOTE

If the amplitude of the marker being requested is out of the range of digital storage, one of the following will be returned for amplitude:

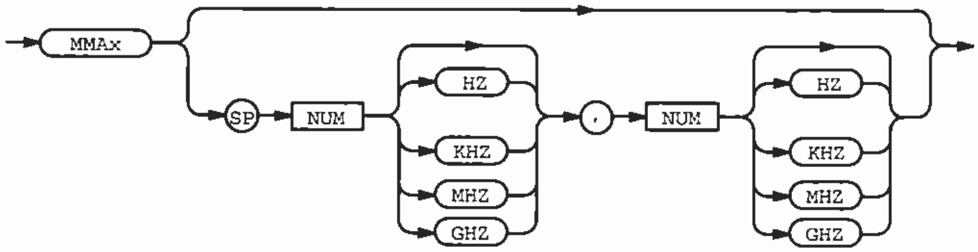
- *−200 if under-range*
- *+200 if over-range*
- *+999.9 if markers are off*

If the requested marker is off, 9.999999E+99 will be returned for frequency.

Interactions: If HDR is OFF, PRIMAR, SECOND, or DELTA and the following delimiter (:) are eliminated along with the MLOCAT header.

MMAx (move marker to maximum)

Syntax:



The MMAX command sets the primary marker to the largest vertical value in digital storage.

If the largest value is located at more than one point, the first (left-most) point is used.

Arguments: NUM — The optional arguments are two frequency values. If these are present, the search is limited to the intersection of the given frequency range and the range displayed on the screen.

If the given range is totally outside the range displayed on the screen, execution error message 28 is issued.

Interactions: If MARKER is OFF, MMAX sets MARKER to SINGLE.

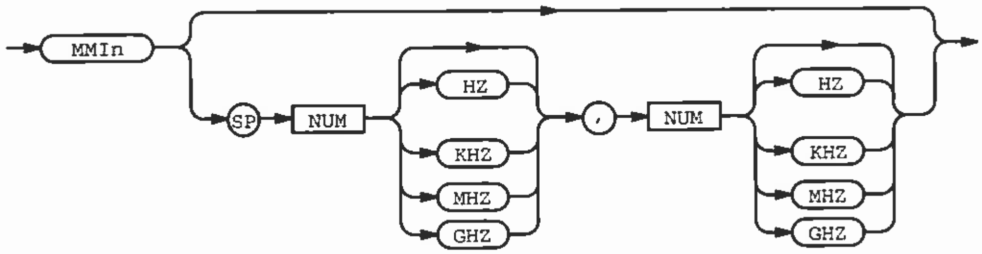
Examples:

```
MMAx
MMAx 15.0MHZ,19.0MHZ
MMA 15.0MHZ,19.0MHZ
```

Macro Memory Used: 12 bytes.

MMIn (move marker to minimum)

Syntax:



The MMIN command sets the primary marker to the smallest vertical value in digital storage.

If the smallest value is located at more than one point, the first (left-most) point is used.

Arguments: **NUM** — The optional arguments are two frequency values. If these are present, the search is limited to the intersection of the given frequency range and the range displayed on the screen.

If the given range is totally outside the range displayed on the screen, execution error message 28 is issued.

Interactions: If MARKER is OFF, MMIN sets MARKER to SINGLE.

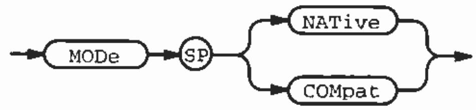
Examples:

```
MMIN
MMIN 15.0MHZ,19.0MHZ
MMI 15.0MHZ,19.0MHZ
```

Macro Memory Used: 12 bytes.

MODe (compatibility mode)

Syntax:



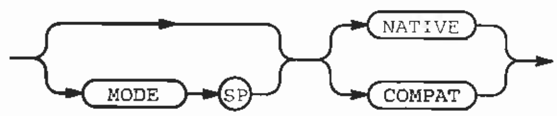
Arguments: **NATIVE** — The normal operating mode for a 2790 series spectrum analyzer.

COMPAT — Puts the spectrum analyzer into 49X compatibility mode, allowing programs written explicitly for 49X spectrum analyzers to run on 2790 series instruments.

Interactions: Refer to Appendix A: *49X Compatibility Mode* for a discussion of operating considerations before using this mode.

Query:

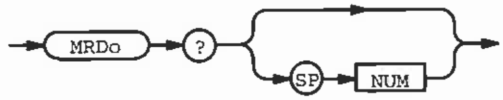
Response:



Power-up Value: Last value stored in memory.

MRDo? (macro readout)

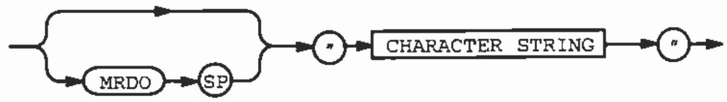
Query:



Arguments: NUM — The line selected by NUM will be returned from the readout buffer that the PRINT command uses.

If MRDO is sent without an argument or NUM is greater than 16, all 16 lines of the readout buffer will be returned.

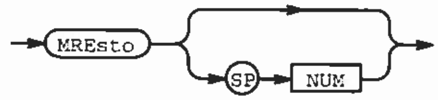
Response:



When all lines are returned, they are combined into one string, with no spaces between the lines.

MREsto (restore data pointer)

Syntax:



The MRESTO command sets the DATA pointer to the first command in the macro (if NUM is used, the DATA pointer is set to LABEL NUM).

Arguments: NUM — When a READ command is used, the macro will start looking for the first MDATA command after the LABEL number specified.

No arguments — When a READ command is used, the macro will start looking for the first MDATA command in the macro.

Examples: The following example illustrates the use of the MDATA, READ, and MRESTO commands.

CAUTION

This program will delete any program stored in macro location 7.

```
 70   Z=1 ! ADDRESS OF SPECTRUM ANALYZER
 80   Print#z:"KILL 7"
 90   Print#z:"STMAC 7,'MREsto TESt'"
100   Print#z:"MDATA 1GHz"
110   Print#z:"LABEL 1"
120   Print#z:"MDATA 2GHZ"
130   Print#z:"MDATA 3GHZ"
140   Print#z:"MDATA 4GHZ"
150   Print#z:"MRESto"
160   Print#z:"GOSUB 2"
170   Print#z:"GOSUB 2"
180   Print#z:"GOSUB 2"
190   Print#z:"MRESTO 1"
200   Print#z:"GOSUB 2"
210   Print#z:"GOSUB 2"
220   Print#z:"GOSUB 2"
230   Print#z:"DONE"
240   Print#z:"LABEL 2"
250   Print#z:"READ"
260   Print#z:"PUTREG FREQ"
270   Print#z:"SWEEP;MFBIG;PAUSE 5"
280   Print#z:"RETURN"
290   Print#z:"EMAC"
```

Line 100	Stores the value of 1 GHz.
Line 110	Sets label marker 1 at this position.
Line 120	Stores the value of 2 GHz.
Line 130	Stores the value of 3 GHz.
Line 140	Stores the value of 4 GHz.
Line 150	Sets the DATA pointer to the first MDATA value (1GHZ).
Line 160	GOSUBS to LABEL 2, which reads 1GHZ into XREG, then sets the center frequency to 1 GHz.
Line 170	GOSUBS to LABEL 2, which reads 2GHZ into XREG, then sets the center frequency to 2 GHz.
Line 180	GOSUBS to LABEL 2, which reads 3GHZ into XREG, then sets the center frequency to 3 GHz.
Line 190	Sets the DATA pointer to the first item after LABEL 1 (2GHZ).
Line 200	GOSUBS to LABEL 2, which reads 2GHZ into XREG, then sets the center frequency to 2 GHz.
Line 210	GOSUBS to LABEL 2, which reads 3GHZ into XREG, then sets the center frequency to 3 GHz.
Line 220	GOSUBS to LABEL 2, which reads 4GHZ into XREG, then sets the center frequency to 4 GHz.
Line 240	Sets label marker 2 at this position.
Line 250	Reads the value in the next MDAT command.
Line 260	Sets the center frequency to the value in XREG.
Line 270	Takes a sweep, moves the marker to the maximum signal, and waits 5 seconds before the next macro command is executed.
Line 280	The macro returns to the line following the last GOSUB.

Macro Memory Used: 5 bytes.

MRGtnx (marker right next)

Syntax:

The MRGTNX command moves the primary marker to the peak of the next signal to the right of the present marker position.

If no signal peak is found, the marker does not move.

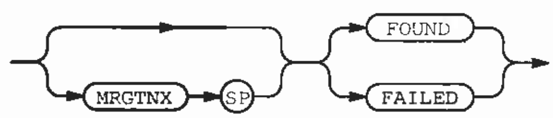
Interactions: If MARKER is OFF, MRGTNX sets MARKER to SINGLE.

If SGERR is ON, marker execution warning message 130 is issued if a signal is not found.

The criteria for a signal are set by the THRHLD and STYPE commands.

Query:

Response:



FOUND is returned if the last MRGTNX command found a signal.

FAILED is returned if the last MRGTNX command did not find a signal.

If the MRGTNX query is given before any MRGTNX command, FAILED is returned.

Macro Memory Used: 1 byte.

MSTep (minus step)

Syntax: → (MSTep) →

This command decreases the center frequency, if you are in the tune frequency mode, by the value set in the STEP command, if possible.

If you are in the tune marker mode, the primary marker frequency is decreased.

If the step marker is on a saved trace and you go outside the displayed trace, execution error message 120 will be issued.

Macro Memory Used: 1 byte.

Power-up Value: The power up step size will be 10% of the current span.

MTOp (marker to reference level)

Syntax:

MTOp

The MTOP command sets the reference level to the level of the marker (or as close as possible, given the specified vertical display and reference level accuracies).

Interactions: Marker execution error message 121 is issued if the primary marker is on an inactive trace.

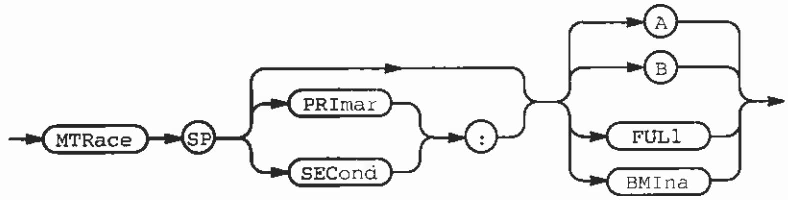
If MARKER is OFF, MTOP sets MARKER to SINGLE.

MTOP is not available in BWMODE.

Macro Memory Used: 1 byte.

MTRace (marker trace position)

Syntax:



MTRACE allows either the primary or secondary marker to be placed on a trace location other than default.

Arguments: **MTRACE** or **MTRACE PRIMAR** — The primary marker is set.

MTRACE SECOND — The secondary marker is set.

A — Place the selected marker on the designated trace.

B — Place the selected marker on the designated trace.

FULL — Place the selected marker on the designated trace.

BMINA — Place the selected marker on the designated trace.

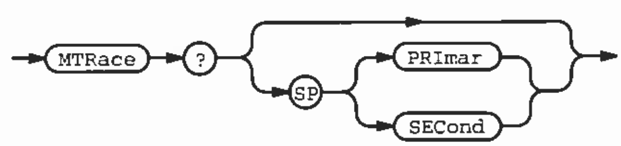
Examples:

```
MTRACE B
MTR PRI:A
MTRACE SECOND:BMINA
```

Interactions:

- MTRACE SECOND sets MARKER to DELTA. If MARKER is OFF, MTRACE or MTRACE PRIMAR sets MARKER to SINGLE.
- Arguments A, B, and BMINA set SAVEA ON.
- BMINA sets BMINA ON.
- Argument FULL sets SAVEA OFF. SAVEA OFF moves any marker(s) on A or B to FULL; SAVEA ON moves any marker(s) on FULL to A or B.
- If BWMODE is ON, one MTRACE command will move both markers to the same trace.
- If the marker is moved off the active trace, it will go back to the active trace if the instrument is in MAX SPAN when it is returned to local control.
- If either marker is placed on a zero-span trace, the other marker will move there also.

Query:



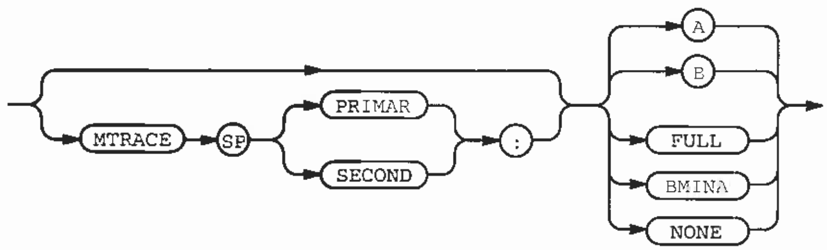
Arguments: **MTRACE?** or **MTRACE? PRIMAR** — The trace containing the primary marker is returned.

MTRACE? SECOND — The trace containing the secondary marker is returned.

Examples:

```
MTRACE?
MTR? SEC
```

Response:



FULL is returned when SAVEA is OFF; A, B, or BMINA (B–Saved A) is returned when SAVEA is ON.

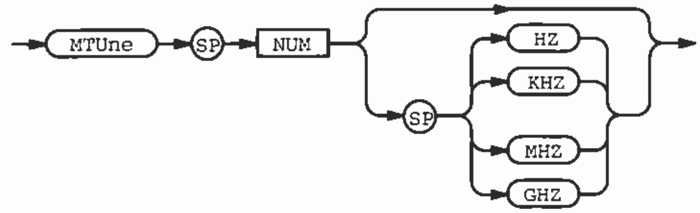
NONE is returned when MARKER is OFF or when MTRACE? SECOND is requested while MARKER is set to SINGLE.

Interactions: If HDR is OFF, PRIMAR or SECOND and the following delimiter (:) are eliminated along with the MTRACE header.

Macro Memory Used: 3 bytes.

MTUne (tune marker)

Syntax:



The primary marker frequency is changed by the value of the number argument. Marker execution error message 120 is issued if the marker is not on an active trace and the resulting marker frequency would not be on the screen.

MTUNE moves the primary marker to center screen and changes center frequency if the primary marker is on an active trace and the frequency is not on the screen.

Range: The same as TUNE; refer to Table 2-21.

Interactions: If MARKER is OFF, MTUNE sets MARKER to SINGLE.

Examples:

```
MTUNE 100
MTU 200 MHZ
```

Macro Memory Used: 6 bytes.

MULt (multiply X and Y registers)

Syntax:

The MULT command multiplies the contents of the X register with the contents of the Y register and puts the result in the X register (XREG = XREG × YREG).

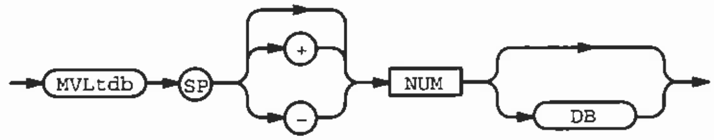
Range: ±9.999 999 999 999 E+99 to ±9.999 999 999 999 E−99.

Interactions: The Y register is unchanged after the MULT command.

Macro Memory Used: 1 byte.

MVLtdb (move marker left x dB)

Syntax:



The MVLTDB command moves the primary marker to the left and NUM DB down (negative NUM) or up (positive NUM or NUM without a sign) from the current position.

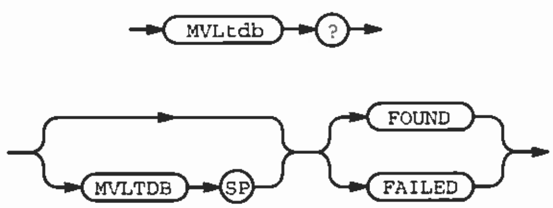
If the requested amplitude cannot be found, the marker does not move.

Range: ±120 dB.

Interactions: If MARKER is OFF, MVLTDB sets MARKER to SINGLE.

If SGERR is ON, marker execution warning message 130 is issued if the requested amplitude is not found.

Query:



FOUND is returned if the last MVLTBD command moved the marker to the requested position.

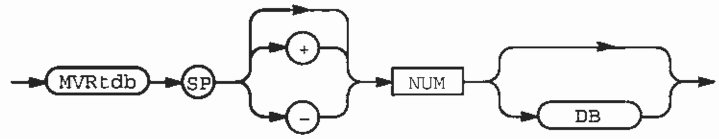
FAILED is returned if the last MVLTBD command could not move the marker to the requested position.

If the MVLTDB query is given before any MVLTBD command, FAILED is returned.

Macro Memory Used: 3 bytes.

MVRtdb (move marker right x dB)

Syntax:



The MVRTDB command moves the primary marker to the right and NUM DB down (negative NUM) or up (positive NUM or NUM without a sign) from the current position.

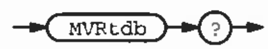
If the requested amplitude cannot be found, the marker does not move.

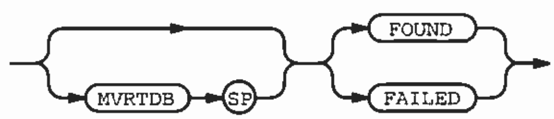
Range: ±120 dB.

Interactions: If MARKER is OFF, MVRTDB sets MARKER to SINGLE.

If SGERR is ON, marker execution warning message 130 is issued if the requested amplitude is not found.

Query:





FOUND is returned if the last MVRTDB command moved the marker to the requested position.

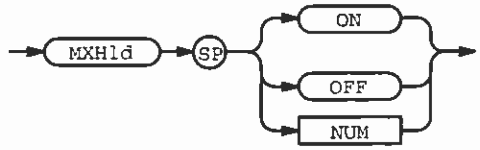
FAILED is returned if the last MVRTDB command could not move the marker to the requested position.

If the MVRTDB query is given before any MVRTDB command, FAILED is returned.

Macro Memory Used: 3 bytes.

MXHld (max hold)

Syntax:



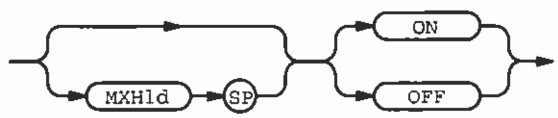
Arguments: **ON** — Digital storage holds the maximum value obtained for each point in both the A and B waveforms; a point is updated only if the new value is greater than the current value.

The A waveform is not affected if SAVEA is on.

OFF — The B waveform is continuously updated; the A waveform is updated only if SAVEA is OFF.

Query:

Response:

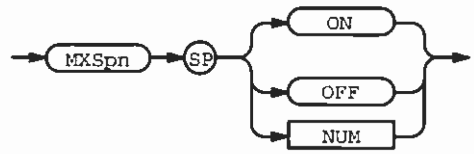


Macro Memory Used: 2 bytes.

Power-up Value: OFF.

MXSpn (max span mode)

Syntax:



Arguments: **ON** — The instrument sweeps the entire frequency range in use. FREQ no longer corresponds to center frequency; it now corresponds to the frequency at the tunable dot above the display or the marker on the display. The previous FREQ SPAN is saved, and it is restored when MXSPN is OFF.

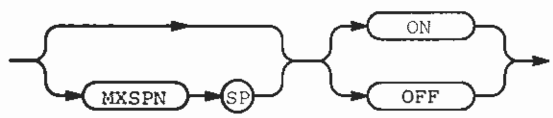
OFF — MXSPN is cancelled, leaving the FREQ SPAN/DIV at the value previously selected.

Interactions: Changing SPAN setting turns MXSPN OFF.

PKCEN, MCEN, and CNTCF are not available when MXSPN is ON.

Query:

Response:



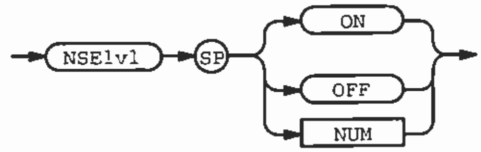
The response is the current MXSPN condition.

Macro Memory Used: 2 bytes.

Power-up Value: OFF.

NSElvl (noise level normalization)

Syntax:



The primary marker amplitude readout normalizes to the resolution bandwidth, and changes the units of the marker amplitude readout from units to units/Hz.

This command assumes the primary marker is on noise, not on a signal. If the marker is on a signal, the marker amplitude readout is incorrect.

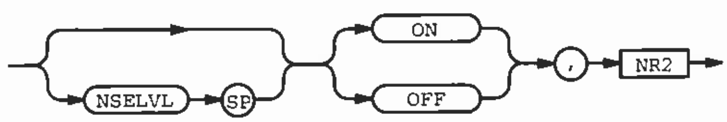
Arguments: **ON** — Normalization is turned on.

OFF — Normalization is turned off.

Interactions: The marker amplitude readout is in reference level units/Hz. If MARKER is OFF, NSELVL sets MARKER to SINGLE.

Query:

Response:



The noise level at the position of the primary marker is returned, regardless of whether NSELVL is ON or OFF. The number is not returned with the NSELVL portion of the SET? response.

If delta markers are on, the secondary marker is assigned to be on a signal, not noise. The readout gives the dBc/Hz of the primary marker with respect to the secondary marker.

NOTE

If the primary marker is out of the range of digital storage, one of the following will be returned

- *−200.0 if under-range*
- *+200.0 if over-range*
- *+999.9 if markers are off*

Macro Memory Used: 2 bytes.

Power-up Value: OFF.

NUMev (number of events)

Syntax:

Arguments: **NR1** — This specifies that a fixed number of event codes is to be returned in ALLEV?

If fewer events are pending when ALLEV? is executed, the response is filled with zeros to provide the specified number.

A value of zero sets the spectrum analyzer to return a variable number of event codes. If the number requested is out of range, execution error message 46 will be issued.

Range: 0 to 39.

Query:

NUMEV? returns the current value for NUMEV.

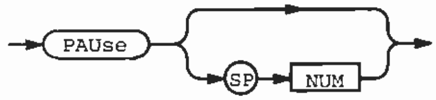
Response:

Macro Memory Used: 2 bytes.

Power-up Value: 0.

PAUse (macro pause)

Syntax:



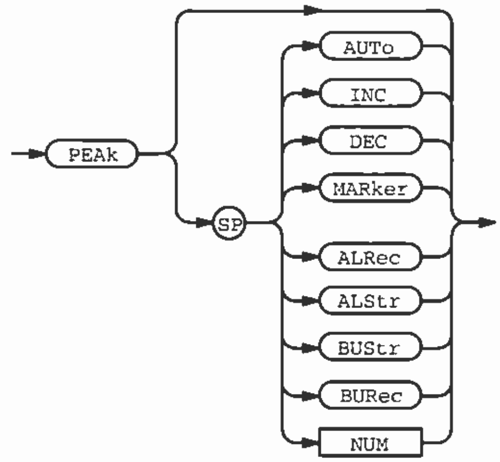
The PAUSE command pauses the current macro for 1 second, or NUM seconds if NUM is used.

Macro Memory Used: 2 bytes.

PEAk (peaking)

2792, 2794, 2797 only

Syntax:



Arguments: **AUTo** — During several sweeps, the spectrum analyzer automatically tunes the PEAK control to peak the largest signal in a window around the display data point (refer to "Waveform Processing" in "Functional Command Groups").

The peak code consists of numbers at 500 MHz intervals when using the preselector, and one number per band when using external mixers. These numbers are stored in memory. If a signal is not found within the window, the previously-acquired peaking code stored in memory is used.

End-of-sweep interrupts are not issued and the TRIGGERING, TIME, MAX HOLD, and REFLVL values may be changed by the spectrum analyzer while PEAK is active. The previous values are restored when PEAK AUTO is through.

Although this command uses digital storage, it does not overwrite the A portion if SAVEA is ON.

The PEAk command without an argument is the same as PEAK AUTO.

INC — The value of PEAK is changed +1 from its current value; the new value is stored in memory.

DEC — The value of PEAK is changed −1 from its current value; the new value is stored in memory.

NUM — The number is stored in memory. Nonintegers or numbers outside the range are rounded to the nearest integer in the range; no warning is issued. This affects the current peaking number only.

MARKER — PEAK MARKER acts the same as PEAK AUTO; except, PEAK MARKER will peak ±1 division from the marker and it will turn the marker on if it is off.

ALREC — Recalls the backup data set into the current operating data set.

ALSTR — Stores the current operating data set into the backup data set.

BUREC — Recalls the peaking value for the present interval from the backup set of peaking values, placing it in the current set.

BUSTR — Stores the peaking value for the present interval into the backup set of peaking values.

Range: 0 to 1023.

Interactions: AUTO may be used when the external mixer is being used or with the internal mixer in bands 2 through 5.

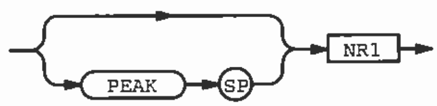
Under the conditions where AUTO may not be used, peaking is not used.

Examples:

```
PEAK
PEA AUTO
PEAK 512
```

Query:

Response:



Returns the number representing the present peaking value.

Macro Memory Used: 4 bytes.

Power-up Value: OFF

PKCen (marker to maximum and center)

Syntax:

The PKCEN command is a combination of the PKFIND and MCEN commands.

The primary marker is moved to the largest vertical value in digital storage (above the threshold). If several values of the same amplitude are found, the left most is selected. Then, the center frequency is tuned to the frequency of the primary marker.

For additional information, refer to the descriptions of PKFIND and MCEN earlier in this section.

Interactions: PKCEN is not available if ZEROSP or MXSPN is ON.

Macro Memory Used: 1 byte.

PKFind (marker to maximum above threshold)

Syntax:

The primary marker is moved to the largest left-most vertical value in digital storage if that value is above the threshold.

If no value is found above the threshold, the marker does not move. PKFIND locates the left-most peak (or the center peak of a cluster), but it is not a signal processing command with built-in intelligence.

Peak B would be selected from the cluster in Figure 2-3A; peak A would be selected in Figure 2-3B because the low point (B) would stop a search from continuing to the cluster (C).

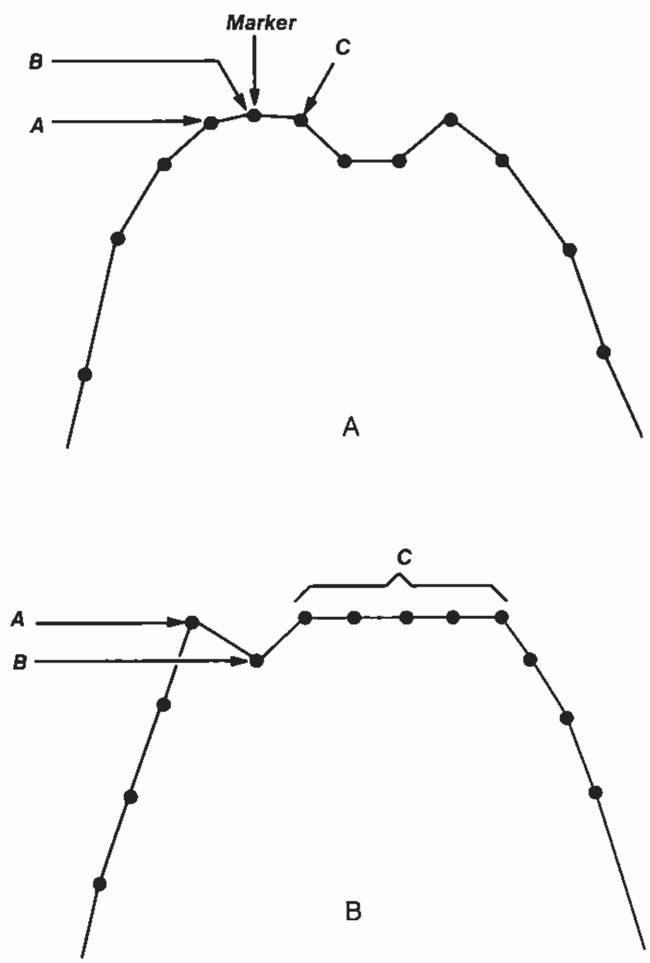
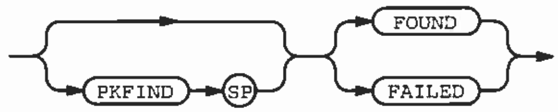


Figure 2-3: Using the PKFIND Command

Query:

Response:



FOUND is returned if the last PKFIND command found a signal.

FAILED is returned if the last PKFIND command did not find a signal. If the PKFIND query is given before any PKFIND command, FAILED is returned.

Interactions: If MARKER is OFF, PKFIND sets MARKER to SINGLE.

If SGERR is ON, marker execution warning message 130 is issued if a value is not found.

The criteria for a signal is set by the THRHLD command. PKFIND does not use STYPE.

Macro Memory Used: 1 byte.

PLOt? (plot display)

Query:

The PLOT query sends information to plot the display on a Tektronix HC100, 4662 Opt 01, or 4662 Opt 31 Interactive Digital Plotter, or a Hewlett-Packard HP7470A, or HP7475A, or plotters which emulate these.

- If REDOUT is ON, the corresponding settings will be plotted.

  If REDOUT is also ON, the bezel information will also be plotted (this assumes that the normal instrument readout is being displayed and not the macro readout buffer or text sent with the RDOUT command).
- If GRAT is ON, the scale along the right-hand side of the screen will be plotted, as well as the graticule information.
- Markers and digital storage must be on for the marker(s) to be plotted.
- The position of the marker(s) will be plotted out as an X.
- VIEWA must be ON to plot the A waveform, VIEWB must be ON to plot the B waveform, and BMINA must be ON to plot the difference between the A and B waveform.
- The readout settings currently displayed on the instrument are the only readout settings plotted.
- The plot can be in more than one color when using a Tektronix HC100, or 4662 Opt 31. The graticule, marker(s), and bezel information will plot in one color, and the waveform in another color.
- If the macro readout buffer is displayed, it alone will plot.

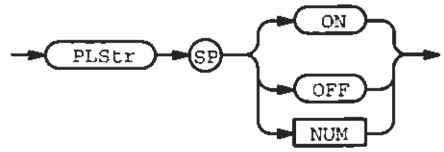
Response: The response to PLOT? depends on the plotter in use (refer to the select plotter command (PTYPE) for a description of the plotter selections).

NOTE

Since the GPIB languages of the plotters does not conform to the Tektronix Interface Standard for GPIB Codes, Formats, Conventions, and Features, this response does not follow the standard.

PLStr (pulse stretcher)

Syntax:



Arguments: **ON** — The fall time of detected signals is increased so very narrow pulses in a line spectrum display can be seen.

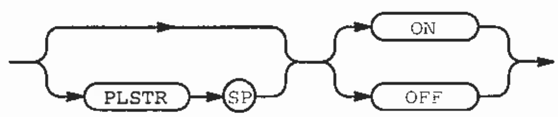
The effect is apparent for signals analyzed at resolution bandwidths that are narrow compared to the span. It may be necessary to turn on the pulse stretcher for digital storage of such signals, especially if the cursor is set high enough to average them.

Pulse stretcher may be required to view and store fast pulsed signals. For short pulses, the signal may exist for less time than is required for a point to be digitized, causing either no value or too low a value to be stored.

OFF — The pulse stretcher is turned off.

Query:

Response:



Macro Memory Used: 2 bytes.

Power-up Value: OFF.

PLUs (add X and Y registers)

Syntax:

The PLUS command adds the contents of the X register to the contents of the Y register and puts the result in the X register (XREG = XREG + YREG).

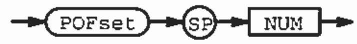
Range: ±9.999 999 999 999 E+99 to ±9.999 999 999 999 E−99.

Interactions: The Y register is unchanged after the PLUS command.

Macro Memory Used: 1 byte.

POFset (set K)

Syntax:



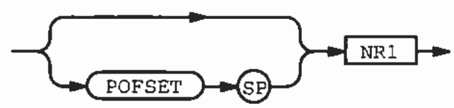
Arguments: NUM — Sets the reference position for plotting B–Save A waveforms.

Range: 0 to 255 (see Interactions).

Interactions: Although the range is 0 to 255 the range visible on the screen is 25 at the bottom to 225 at the top. The nearest limit is used if the selected number is out of range (no error is reported).

Query:

Response:



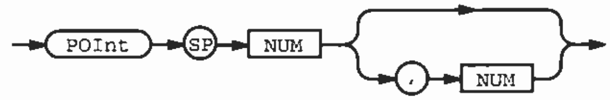
Range: 0 to 255.

Macro Memory Used: 2 bytes.

Power-up Value: The last value stored in memory.

POInt (display data point)

Syntax:



Arguments: NUM — The first number is the X value of a display data point. The horizontal scale is always the same as a full, 1000-point waveform. Refer to "Waveform Data and Command Interactions" on page 2-20.

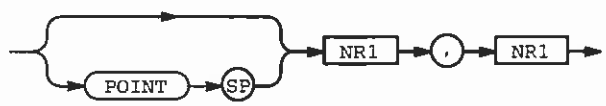
The second number is the Y value of a display data point. The vertical scale is the same as illustrated for the CURVE query.

If the second number is not entered, digital storage is asked for the value of the waveform at X (the first number). This makes the display data point correspond to a point in digital storage.

If the second number is supplied in the POINT command, the display data point may not correspond to any point in digital storage.

Query:

Response:



The first number is the X value of the display data point; the second number is the Y value of the display data point.

Note that the query response may not match any point in digital storage if the Y value was set by a POINT command or if digital storage was updated after the display data point was acquired.

Interactions: The SET? response sent back to the instrument sets both the X and Y values of the display data point, which may not correspond to any point in digital storage. Refer to "Waveform Data and Command Interactions" on page 2-20.

Examples: `POINT 500,125` (center screen)
`POI 1,25` (screen bottom left)
`POI 1000,225` (screen top right)

Macro Memory Used: 5 bytes.

Power-up Value: 500,225.

POP (put Y register into X register)

Syntax: → POP →

The POP command duplicates the current contents of Y register and puts this into the X register.

Interactions: The Y register will remain unchanged after the POP command.

Examples: The following example illustrates the use of the POP command.

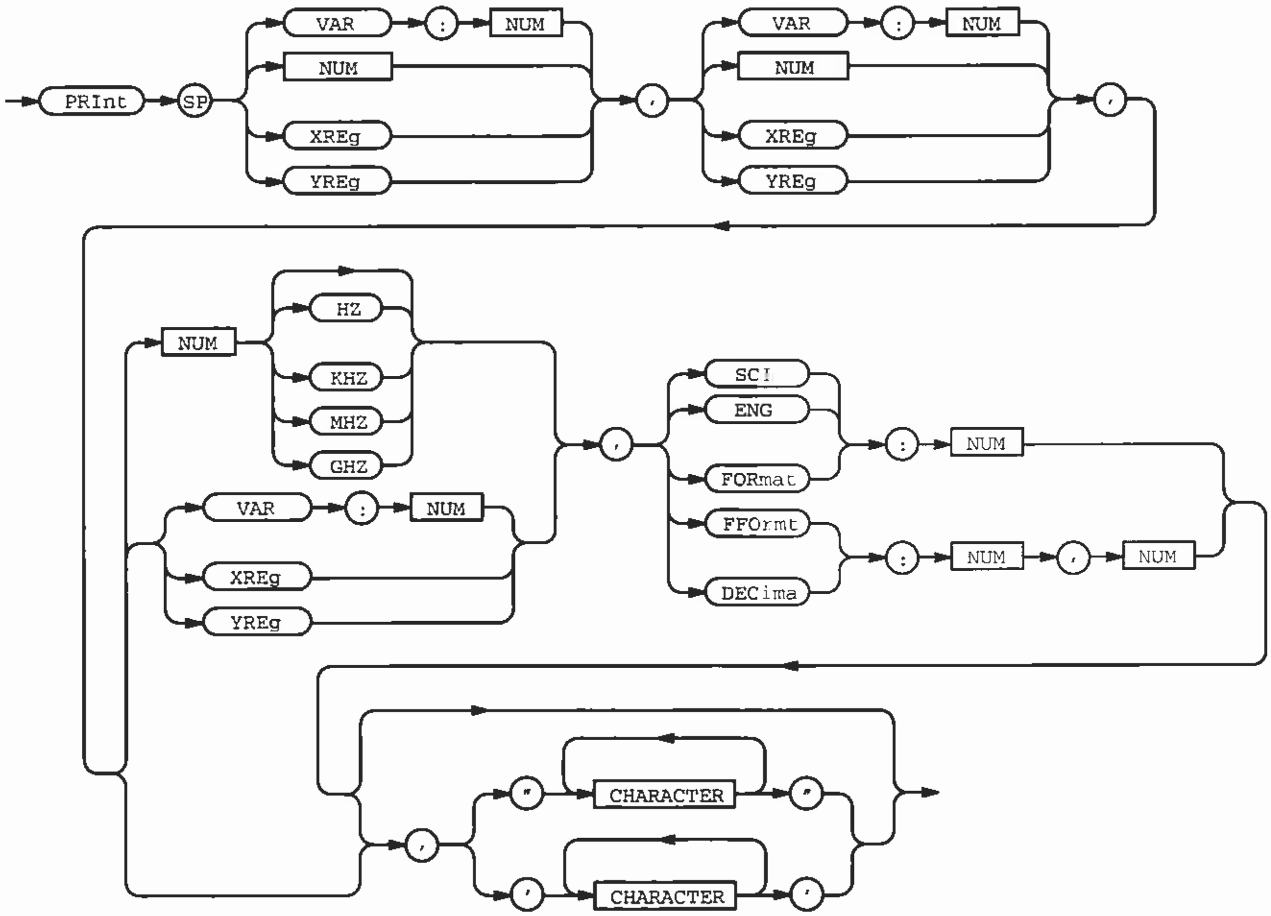
```
 90   Z=1 ! ADDRESS OF SPECTRUM ANALYZER
100   Print #z:"ENTER 5"
110   Print #z:"ENTER 10"
120   Print #z:"POP"
130   Print#z:"STNUM 1"
140   Print#z:"VAR? 1"
150   Input#z:r
160   Print r
```

Line 100	Enters 5 into XREG.
Line 110	Moves the contents of XREG to YREG, and enters 10 in XREG.
Line 120	Duplicates the contents of YREG (5) and puts the result in XREG (now also 5).
Line 130	Stores the contents of XREG in variable number 1.
Line 150	Puts the contents of variable number 1 into r.
Line 160	Prints the contents of r (5).

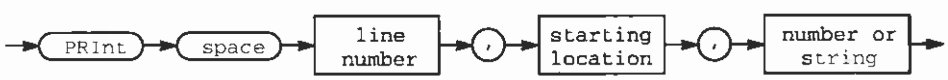
Macro Memory Used: 1 byte.

PRInt (print number or string)

Syntax:



The following is a simplified representation of the PRINT command syntax diagram.



The PRINT command prints a number and a string of up to 29 characters or a string of up to 40 characters.

The first argument sets the line number on which the string will be printed.

The second argument sets the starting position for the string.

The third argument determines whether it is a number or a string that is to be printed. Numbers can be printed in scientific notation (SCI), engineering notation (ENG), formatted (FORMAT), frequency format (FFORMT), or decimal (DEC).

The following briefly describes the five ways in which a number can be returned. The first, or only, link argument after the format specifier gives the total number of character positions occupied by the output. Numbers are right justified and preceded by leading spaces. If the number will not fit in the width specified, a string of asterisks (*) of the width specified will be output.

SCI — prints a number in NR3 format. The number is justified as described above and there are no trailing zeros.

ENG — prints a number in the same manner as SCI, except that the decimal point position is adjusted to make the exponent a power of 3. This corresponds to common engineering units such as GHz and ms.

FORMAT — prints a number in NR1 format. A number to be printed with FORMAT cannot have a fractional part.

FFORMAT — prints a number as a frequency in NR2 format with an identifier (HZ, KHZ, MHZ, or GHZ) added. There is no space between the number and the identifier. The identifier is chosen so that the number to the left of the decimal point has a range of 1–999. The second link argument gives the resolution, as a power of 10 (in Hz), to which the number is displayed. A number to be printed with FFORMAT cannot have a fractional (less than 1 Hz) part.

DECIMAL — prints a number in NR2 format. The second link argument gives the number of digits to the right of the decimal point.

Range: The first argument (line number) is 1 to 16; the second argument (starting position) is 1 to 40.

If FORMAT or FFORMT is used, the range of the third argument is ±549,755,813,887 (maximum value for a 5-byte hexadecimal number).

If no number is to be printed (just a string), the string, plus the starting position, may be up to 40 characters.

If the string extends beyond the 40-character maximum, it will be truncated.

If a number is to be printed, the string may be up to 29 characters.

Examples: The following examples assume that the number indicated by the result shown is in the X register.

```
PRINT 1,5,XREG,SCI:14                "      3.122E+10"
PRINT 1,5,XREG,SCI:14," WATTS"       "       3.122E-3 WATTS"
PRINT 1,5,XREG,ENG:14                "      -31.12E+0"
PRINT 1,5,XREG,ENG:14," WATTS"       "      71.22E-E WATTS"
PRINT 1,5,VAR:3,DEC:14,2             "        -101.34"
PRINT 1,5,VAR:3,DEC:14,0             "          73652"
PRINT 1,5,VAR:3,DEC:14,0," HZ"       "          73652 HZ"
PRINT 1,5,YREG,FORMAT:14             "      2 738 274"
PRINT 1,5,YREG,FORMAT:14,"HZ"        "      2 738 274HZ"
PRINT 1,5,XREG,FFORMT:14,3           "         12.738MHZ"
PRINT 1,5,XREG,FFORMT:14,0           "      12,738 749MHZ"
PRINT 1,5,XREG,FFORMT:14,6           "             12MHZ"
PRINT 1,5,XREG,FFORMT:14,2           "        12.738 3MHZ"
```

Macro Memory Used: 47 bytes.

PSTep (plus step)

Syntax:

→(PSTep)→

This command increases the center frequency, if you are in the tune frequency mode, by the value set in the STEP command, if possible.

If you are in the tune marker mode, the primary marker frequency is increased.

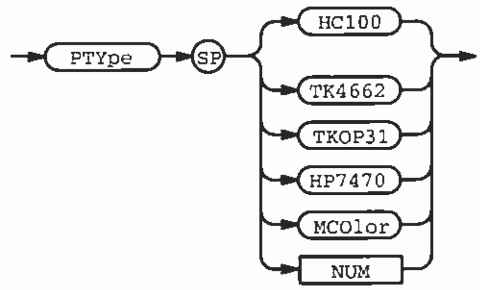
If the step marker is on a saved trace and you go outside the displayed trace, execution error message 120 will be issued.

Macro Memory Used: 1 byte.

Power-up Value: The power up step size will be 10% of the current span.

PTYpe (plotter type)

Syntax:



Arguments:

HC100 — Selects the Tektronix HC100 as the plotter driven by the data generated by PLOT?.

TK4662 — Selects the Tektronix 4662 Opt 01 (or the 4663 in a one-pen configuration) as the plotter driven by the data generated by PLOT?.

TKOP31 — Selects the Tektronix 4662 Opt 31 (or the 4663 in a two-pen configuration) as the plotter driven by the data generated by PLOT?.

HP7470 — Selects the Hewlett-Packard HP7470A as the plotter driven by the data generated by PLOT?.

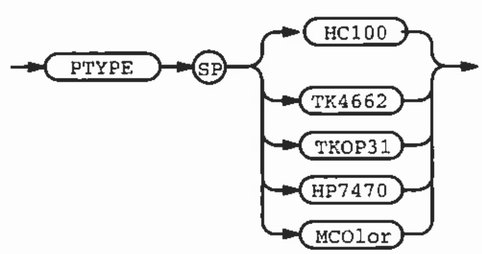
MCOLOR — Selects the Hewlett-Packard HP7475A as the plotter driven by the data generated by PLOT?.

NUM —

0 = TK4662
1 = TKOP31
2 = HP7470
3 = MCOLOR
4 = HC100

Query:

Response:

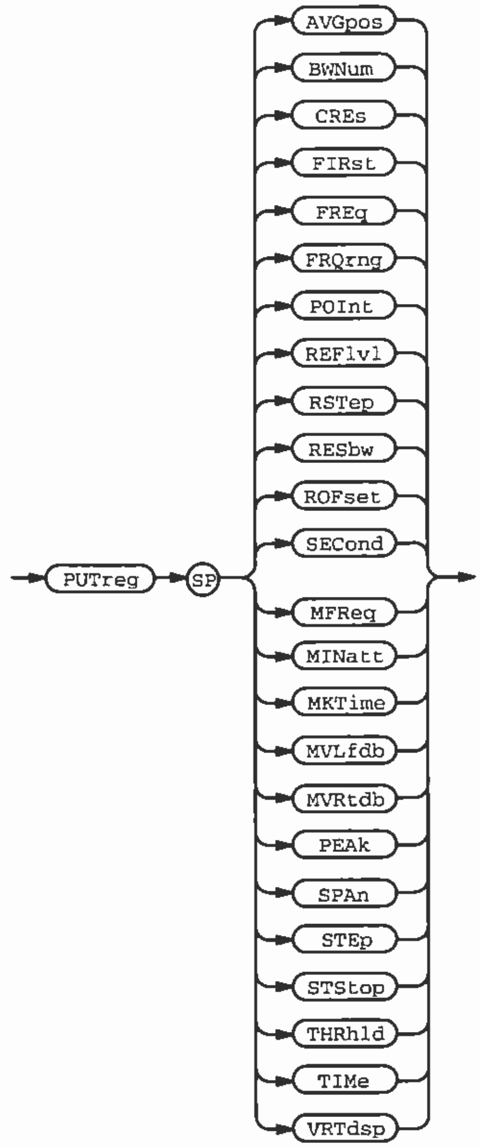


Macro Memory Used: 2 bytes.

Power-up Value: Last value stored in memory.

PUTreg (put X register into a setting)PUT

Syntax:



The PUTREG command sets the argument to the value currently in XREG; except STSTOP, which requires two numbers and uses both the X and Y registers (the START FREQ goes into YREG and the STOP FREQ goes into XREG).

For a description of the command arguments, see the description of the command whose mnemonic is the same as the argument.

Examples:

```
PUTREG STEP        See 1
PUTREG RESBW       See 2
PUTREG STSTOP      See 3
PUTREG MVRTDB      See 4
```

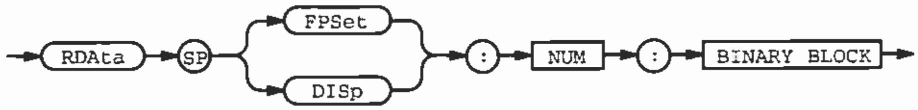
1. Puts the value of XREG into the step size.

2. Sets the resolution bandwidth to the value in XREG.
3. Puts the contents of YREG into START FREQ and the contents of XREG into STOP FREQ.
4. Puts the value of XREG into X and calls the MVRTDB routine.

Macro Memory Used: 2 bytes.

RDAta (register data)

Syntax:



This command transfers, directly to a numbered storage register, either a front-panel setting or a waveform and associated data.

The data is transferred in a coded binary format.

The data is obtained from a previous RDATA query.

Arguments:

FPSET — The front-panel settings contained in the binary block are transferred to the indicated register.

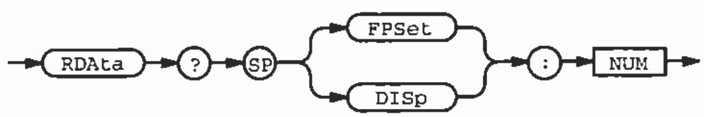
DISP — The waveform and the associated readout and scaling data contained in the binary block are transferred to the associated register.

NUM — The number of the storage register to which data will be transferred. If the number is outside of the range of 0–9 when FPSET is used or the range 0–8 when DISP is used, execution error message 47 will be issued.

The binary data sent is internally checksummed. (This checksum is different than the checksum added in the binary block format.)

If the internal checksum does not match the data, execution error 190 will be issued.

Query:



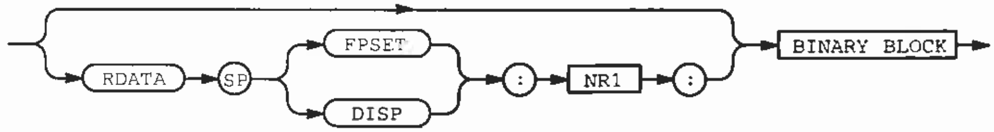
The RDATA query transfers, directly from a numbered storage register, either a front-panel setting or a waveform and associated data. The data is transferred in a coded binary format. It is intended that this information be used only as data for a subsequent RDATA command.

FPSET — The front-panel settings contained in the indicated register are transferred.

DISP — The waveform and the associated readout and scaling data contained in the associated register are transferred.

NUM — The number of the storage register to which data will be transferred. The number must be in the range of 0–9 when FPSET is used or the range 0–8 when DISP is used.

Response:



The length of the binary data, exclusive of the byte count and checksum (see Binary Block in Section 2 of this manual) is 124 bytes when a setting is being returned and 644 bytes when a display is being returned.

If the register number sent with the query is out of range, a register number of −1 will be returned.

If the requested register did not contain valid data, a register number of −2 will be returned.

In either of the above cases all binary data bytes will be 0.

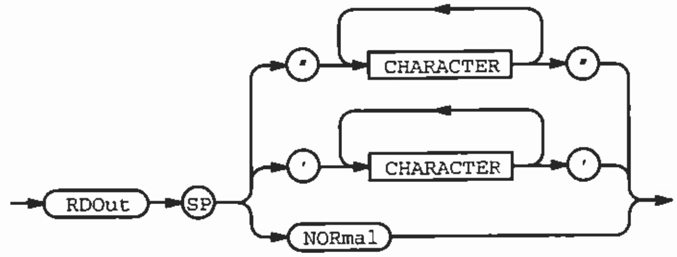
Interactions: The RDATA? FPSET response is not the same length as the response from 49X instruments.

CAUTION

Any program that is running on a 2790 series instrument will crash if sent a previously-learned RDATA? FPSET response from a 49X instrument.

RDOut (readout message)

Syntax:



Arguments: CHARACTER — In the TEXT SHORT mode, the spectrum display remains on the CRT, the readout is cleared, and up to the first 40 characters, are displayed across the bottom of the spectrum analyzer CRT.

When the RDOUT command sends a new line of characters, it is entered at the bottom of the CRT and the previous bottom line of characters is moved to the top of the CRT. Each succeeding line of characters is displayed at the bottom of the CRT, and the previous bottom line moves to the top, discarding the previous top line. Thus, each new RDOUT command causes the spectrum analyzer readout to scroll.

In the TEXT LONG mode, the screen is completely blanked and up to the first 40 remotely-entered characters are displayed in the 1st line at the top of the CRT screen. Successive lines of characters are entered on the following lines until the 16th (bottom) line is reached. Then, as each successive line of characters is entered, the entire screen scrolls up one line, the first line is discarded, and the new RDOUT command characters become the 16th line.

NORMAL — Normal spectrum analyzer readout is restored.

Interactions: If a CRT message sent with RDOUT remains on the screen after the spectrum analyzer is returned to local control, normal readout can be restored by changing any control that causes the normal readout to be updated.

TEXT LONG will blank the screen.

Macro Memory Used: 41 bytes.

Power-up Value: NORMAL.

REAd (read and store in XREG)

Syntax:

The READ command reads the number at the current position of the DATA pointer and puts that value in XREG.

The macro starts looking from where the pointer is and looks until it finds the first MDATA command. The value of the number is put in XREG, and the pointer now points to the next command in the macro.

The program with the MRESTO command description illustrates the use of the READ command.

Interactions: The data pointer is set to the first command in the macro when the macro is started.

If there are no more MDATA commands for READ to read, macro execution error message 166 will be issued.

Macro Memory Used: 1 byte.

RECall (recall settings)

Syntax:

Arguments: NUM — The instrument control settings are recalled from the selected memory location.

Range: 0 to 9.

Interactions: If you try to recall settings from an empty memory location, execution error message 62 is issued.

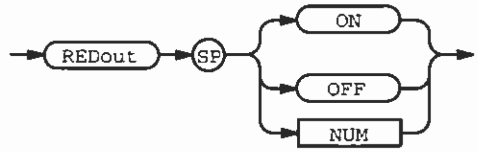
The instrument STORES its current settings in memory 0 automatically when the power is turned off, overwriting any previously-stored settings.

Macro Memory Used: 2 bytes.

Power-up Value: RECALL 0.

REDout (readout)

Syntax:

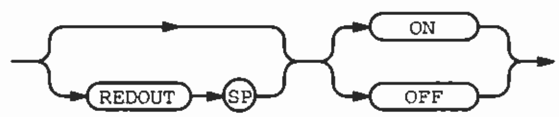


Arguments: **ON** — The instrument readout lines are displayed on the CRT.

OFF — The instrument readout lines are not displayed on the CRT; the readout is blanked. Menus will still be displayed. The entry line will be displayed while an entry is in process.

Query:

Response:

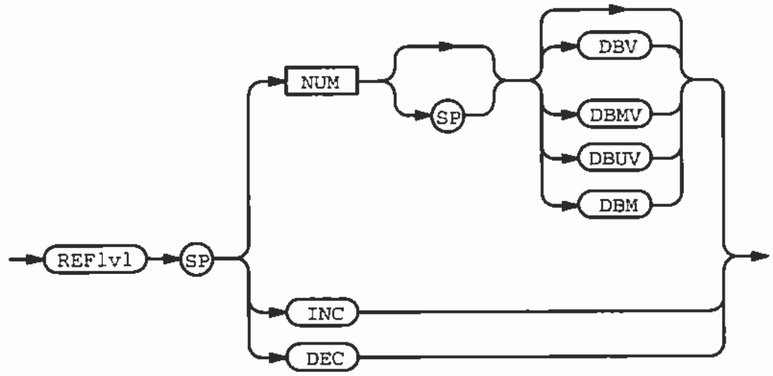


Macro Memory Used: 2 bytes.

Power-up Value: ON.

REFlvl (reference level)

Syntax:



NOTE

To ensure the correct response, all of the letters in each of the unit mnemonics for the REFLVL command must be entered; not just the first three letters as for other mnemonics.

Arguments:

INC — The reference level is stepped up by 1 dB if possible.

DEC — The reference level is stepped down by 1 dB if possible.

NUM — The instrument sets the reference level to the nearest 0.25 dBm.

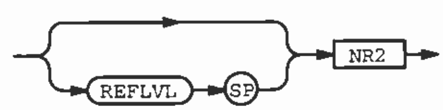
If the number selected is out of range, execution error message 34 is issued. If no units are specified, the instrument assumes the current reference level units.

Examples:

```
REF -200DBV
REFLVL -10 DBMV
REF -30DBUV
REFLVL -25 DBM
REF INC
```

Query:

Response:



Only the number value will be returned; the units will not be indicated (the number will be returned in the current reference level units). The value returned is the absolute reference level.

Macro Memory Used: 4 bytes.

Power-up Value: 0 dBm.

REPeat (repeat execution)

Syntax:

Arguments: NUM — This determines the number of times, in addition to the first time, the spectrum analyzer is to repeat the commands or queries that come before REPEAT.

Range: 0 to 16,777,215 (2^{24} −1).

Since REPEAT may itself be one of the commands that comes before a REPEAT, the nested (first) REPEAT will only be performed on the first pass through the commands that come before the second REPeat. Refer to "Examples."

Interactions: A REPEAT loop can only be stopped by DCL.

Pressing [RESET TO LOCAL] does not stop the loop; it only causes execution error messages to be reported if the loop contains front-panel commands.

If [RESET TO LOCAL] is pressed while a message that includes REPEAT is being acted on, the message will only be repeated 256 times. (Since most commands are ignored after the [RESET TO LOCAL] button is pressed, the REPEAT loop completes quickly).

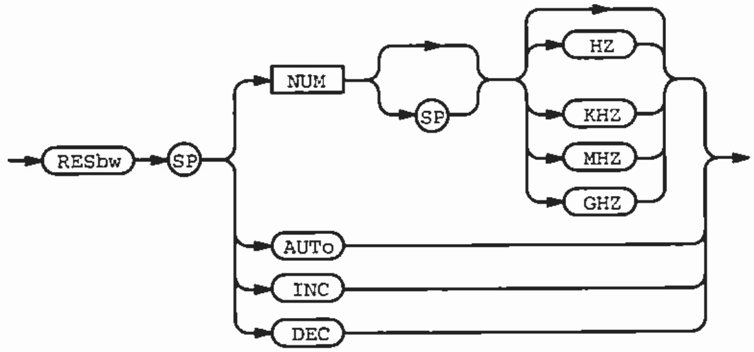
Examples:

```
RGTNXT;FREQ?;REPEAT 10;FREQ 1.4 GHZ;REPEAT 1
```

This causes the spectrum analyzer to output 12 frequency values, because it only performs the frequency query once on its second pass through the entire message.

RESbw (resolution bandwidth)

Syntax:



Arguments: **NUM** — The nearest available resolution bandwidth filter is selected; numbers between bandwidths that can be selected from the front panel are rounded.

Positive numbers above or below the range of bandwidth steps are rounded to the nearest step; refer to Table 2-19 (execution error message 32 is issued if the argument is beyond the normal range).

AUTO — Auto resolution is selected (equivalent to ARES ON).

INC — The next larger step is selected (if possible).

DEC — The next smaller step is selected (if possible).

Range:

Table 2-19: Resolution Bandwidth Selection

Value[1]	Selects	
3.17 Hz–31.6 Hz	10 Hz	(except 2792)
31.7 Hz–316 Hz	100 Hz	(except 2792)
317 Hz–3.16 kHz	1 kHz	
3.17 kHz–31.6 kHz	10 kHz	
31.7 kHz–316 kHz	100 kHz	(except Option 07))
31.7 kHz–316 kHz	300 kHz	(Option 07 Only)
317 kHz–1.72 MHz	1 MHz	
1.73–5.44 MHz	3 MHz	

[1] **Values outside the ranges listed cause execution error message 32 to be issued.**

Interactions: Any argument except AUTO cancels ARES ON.

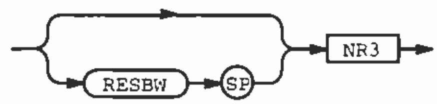
Reducing resolution bandwidth may require a slower sweep speed (TIME) to maintain a calibrated display unless TIME is set to AUTO.

Examples:

```
RES 100
RESBW 1KHZ
RES 1.5 MHZ
RESBW INC
```

Query:

Response:



The response to the SET query includes the AUTO argument (see SET?) because, in AUTO, the instrument can determine the bandwidth.

Macro Memory Used: 5 bytes.

Power-up Value: 3 MHz.

RETurn (return from a subroutine)

Syntax: → RETurn →

The RETURN command returns macro control to the line following the last GOSUB command.

Refer back to the example for GOSUB to see the use of the RETURN command.

If there is a RETURN without a GOSUB, macro execution error message 168 (RETURN NOT EXPECTED) will be issued, and the macro will be aborted.

NOTE

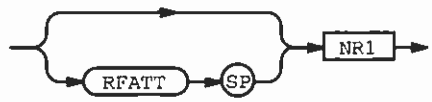
The last command in a macro before the EMAC command must be GOTO or RETURN or DONE. If it is not, macro execution error message 178 is issued.

RFAtt? (RF attenuation)

Query:

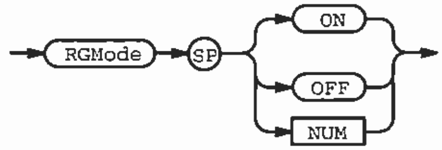
Requests the current value of RF attenuation.

Response:



RGMode (reduced gain mode)

Syntax:



This command enables or disables 10 dB of IF gain and RF attenuation reduction when in 10 dB/division.

Arguments:

ON — The reduced gain mode is turned on.

OFF — The reduced gain mode is turned off.

Interactions:

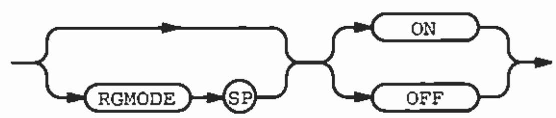
When IDENT and RGMODE are ON, the identify trace moves up instead of down.

RGMODE affects the maximum reference level you can get with the REFLVL command.

When not in 10 dB/div vertical display, RGMODE does not affect the gain distribution.

Query:

Response:

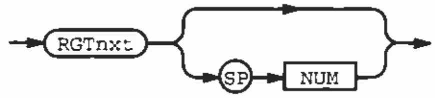


Macro Memory Used: 2 bytes.

Power-up Value: OFF.

RGTnxt (right next)

Syntax:



This command searches to the right of the current point to acquire the peak of a signal whose value is greater than NUM and sets the display data point to the peak.

If a signal peak greater than NUM is not found, the display data point is set to 1001,0.

If NUM is omitted from the command, a default value of 0 is used.

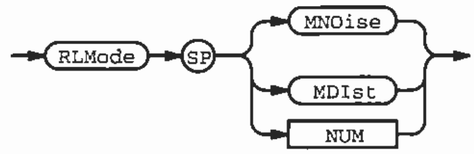
A pattern recognition routine is used to recognize signals.

Interactions: Refer to "Waveform Data and Command Interactions" on page 2-20.

Macro Memory Used: 2 bytes.

RLMode (reference level mode)

Syntax:



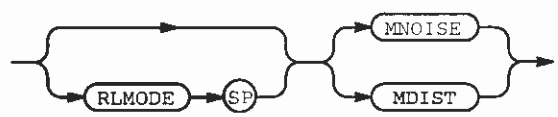
Arguments: `MNOISE` — The instrument is requested to assign gain distribution with minimum RF attenuation for a given reference level. Generally, this yields 10 dB less RF attenuation than the MDIST argument and results in less displayed noise (but may increase distortion).

`MDIST` — Generally, this yields 10 dB more RF attenuation than the MNOISE argument and results in lower signal levels in the spectrum analyzer, hence less distortion.

`NUM` —
0 = MDIST
1 = MNOISE

Query:

Response:



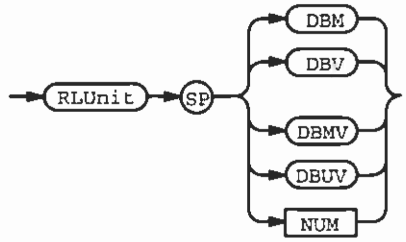
Interactions: This command affects the gain distribution obtained with the REFLVL command (see also MINATT and MAXPWR).

Macro Memory Used: 2 bytes.

Power-up Value: OFF.

RLUnit (reference level units)

Syntax:



NOTE

To ensure the correct response, all of the letters in each of the unit mnemonics for the RLUNIT command must be entered; not just the first three letters as for most other mnemonics.

Arguments:

DBM — The reference level (REFLVL) units are set to dBm.

DBV — The reference level (REFLVL) units are set to dBV.

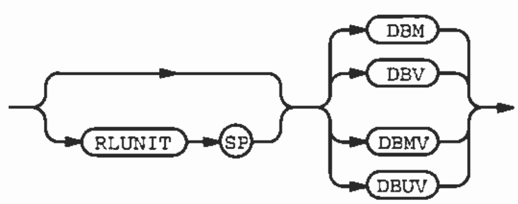
DBMV — The reference level (REFLVL) units are set to dBmV.

DBUV — The reference level (REFLVL) units are set to dBμV.

NUM —
0 = dBm
1 = dBV
2 = dBmV
3 = dBuV

Query:

Response:



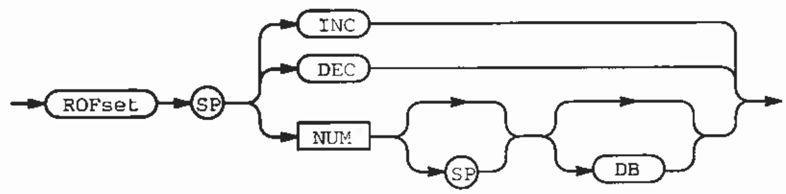
Interactions: In instruments with Option 07 installed, dBmV is automatically selected when the 75 Ω input is chosen, and dBm is automatically selected when the 50 Ω input is chosen. The units designator can be overridden once the input selection has been made.

Macro Memory Used: 2 bytes.

Power-up Value: dBm.

ROFset (reference level offset)

Syntax:



Arguments: **NUM** — This sets the offset that will be applied to the reference and marker levels.

INC — Increments the offset by 1 dB.

DEC — Decrements the offset by 1 dB.

Range: −30.0 dB to +30.0 dB.

Interactions: The offset value will affect the responses to the queries of REFLVL, MAMPL, MLOCAT, MAXPWR, and THRHLD.

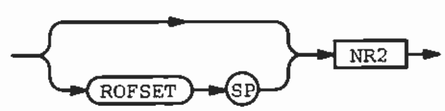
ROFSET must be accounted for in the values sent with the REFLVL, THRHLD, and MAXPWR commands.

Examples:

```
ROF 26
ROFSET 30 DB
ROF 7.5DB
```

Query:

Response:

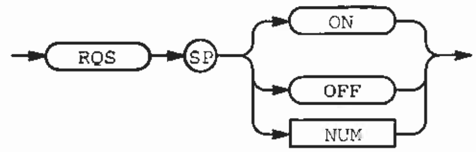


Macro Memory Used: 3 bytes.

Power-up Value: 0 dB.

RQS (request service)

Syntax:



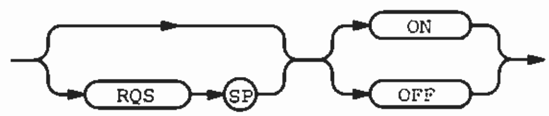
Arguments: **ON** — SRQ is asserted when abnormal status conditions occur.

OFF — SRQ is not asserted (is masked) when abnormal status occurs.

Interactions: RQS is always OFF in the talk-only and listen-only modes.

Query:

Response:

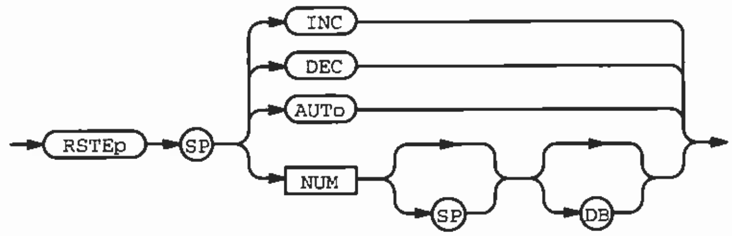


Macro Memory Used: 2 bytes.

Power-up Value: ON.

RSTep (reference level step)

Syntax:



This command sets the reference step level size for the dedicated reference level increment and decrement keys.

Arguments:

NUM — Sets the reference step level size to the number of dB specified.

AUTO — in log mode, sets a step size equal to the vertical scale; in linear mode the step size becomes variable to select a 1–2–5 vertical scale sequence.

INC — Increments the step size 1 dB, if possible.

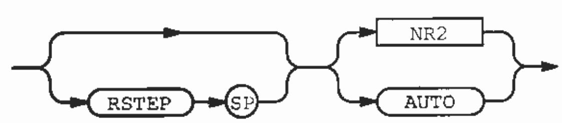
DEC — Decrements the step size 1 dB, if possible.

Range: NUM can be from 1 to 30.

Interactions: When leaving AUTO the step size is first set to the vertical scale (if in log mode) or 1 dB (if in linear mode). Therefore, if the instrument is in linear mode, with AUTO step size, sending the command `RSTEP DEC` will result in an error as the instrument attempts to set the step size to 0 dB.

Query:

Response:

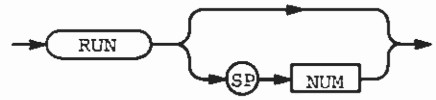


Macro Memory Used: 3 bytes.

Power-up Value: AUTO.

RUN (run macro)

Syntax:



Arguments: NUM — The macro number NUM will be run. Any macro currently running will be aborted and macro NUM will be started.

Interactions: If any GPIB command comes while a macro is executing, the macro is aborted.

RUN without an argument will run the last macro run.

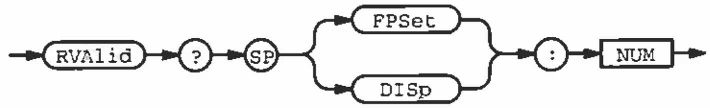
If RUN is sent with a NUM (0–7) where no macro is located, macro execution error message 165 is issued.

Examples: Here are two short examples of illegal RUN arguments.

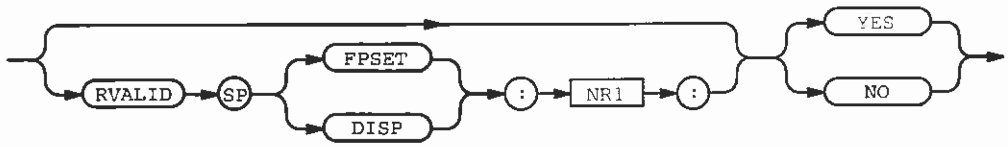
- `RUN 10` is sent. Since macro locations are 0 through 7, 10 is an illegal macro number, and the last macro that was run will be run again.
- `RUN -1` is sent. Any negative number is illegal, and the macro in the 0 location will be run.

RVAlid? (register valid)

Query:



Response:



YES — Is returned if the indicated register contains a valid front-panel setting or display.

NO — Is returned if the indicated register does not contain a valid front-panel setting or display.

Range: 0–9 if FPSET is used, and 0–8 if DISP is used.

Interactions: If HDR is OFF, the register type, register number, and separators are eliminated along with the header.

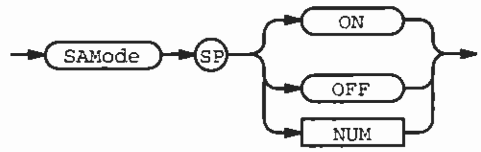
If the register number is out of range, the value −1 is substituted for the register number in the response (assuming HDR is ON) and NO is returned.

Examples:

```
RVALID? FPSET:2
RVA? FPS:7
RVA? DISP:7
```

SAMode (sideband analyzer mode)

Syntax:



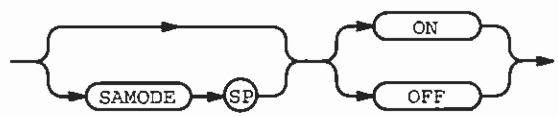
SAMODE is active in Band 1 only. When the SAMODE command is ON, the spectrum analyzer phase locks in spans equal to or less than 500 kHz (50 kHz/Div). When SAMODE is off, phase locking occurs at spans equal to or less than 2 MHz. This extends the usefulness of the 1405 TV Sideband Analyzer, which uses only the first local oscillator of the spectrum analyzer.

Arguments: **ON** — The sideband analyzer mode is turned on.

OFF — The sideband analyzer mode is turned off.

Query:

Response:

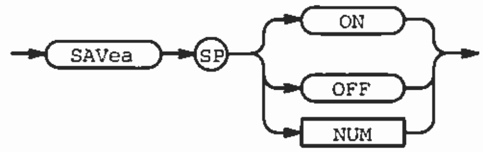


Macro Memory Used: 2 bytes.

Power-up Value: OFF.

SAVea (save A waveform)

Syntax:



Arguments: **ON** — The A waveform updating is stopped and the current contents are saved. This allows comparison with the B waveform, which is continuously updated.

The information in the CRT readout is saved, and will be displayed instead of the current instrument settings if only AVIEW is ON (that is, both BVIEW and BMINA OFF).

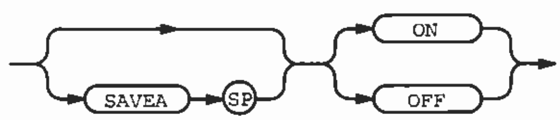
OFF — The A waveform updating is resumed.

Interactions: BMINA ON turns SAVEA ON.

SAVEA OFF turns BMINA OFF.

Query:

Response:

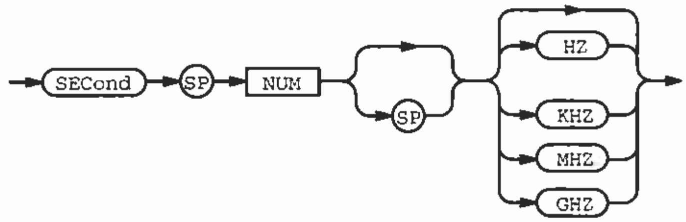


Macro Memory Used: 2 bytes.

Power-up Value: OFF.

SECond (2ND LO frequency)

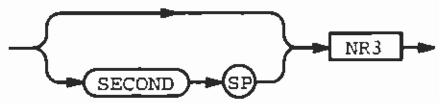
Syntax:



Arguments: NUM — The instrument 2nd LO is set to the specified frequency. The resulting center frequency will be displayed.

Query:

Response:



Range: Bands 1 and 5–12 is 2181–2183 MHz.

Bands 2–4 is 718–720 MHz.

Refer to Table 2-21 on page 2-227.

Examples: `SECOND 2182 MHZ`

Macro Memory Used: 6 bytes.

Power-up Value: 2182 MHz.

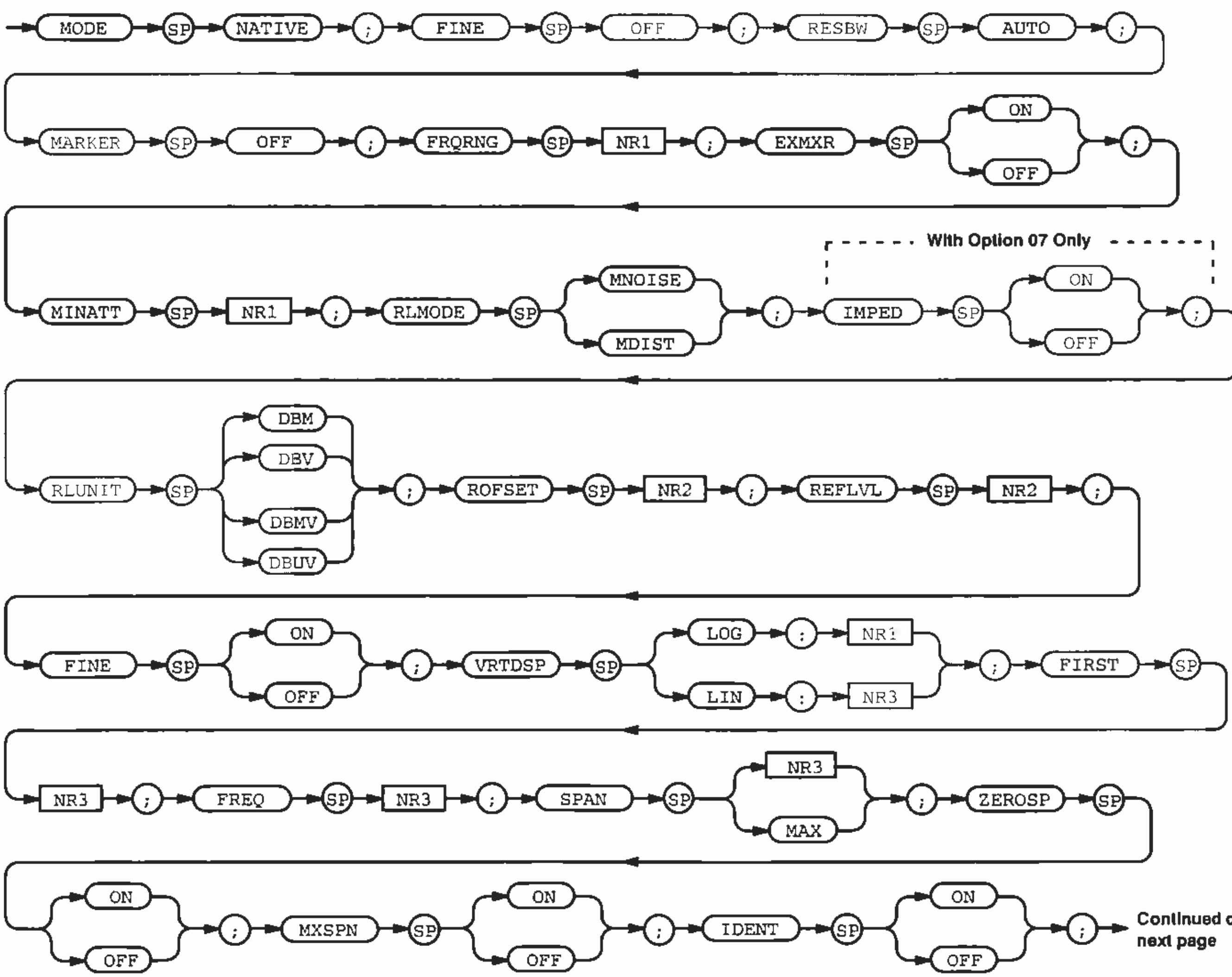
SET? (instrument settings)

Query:

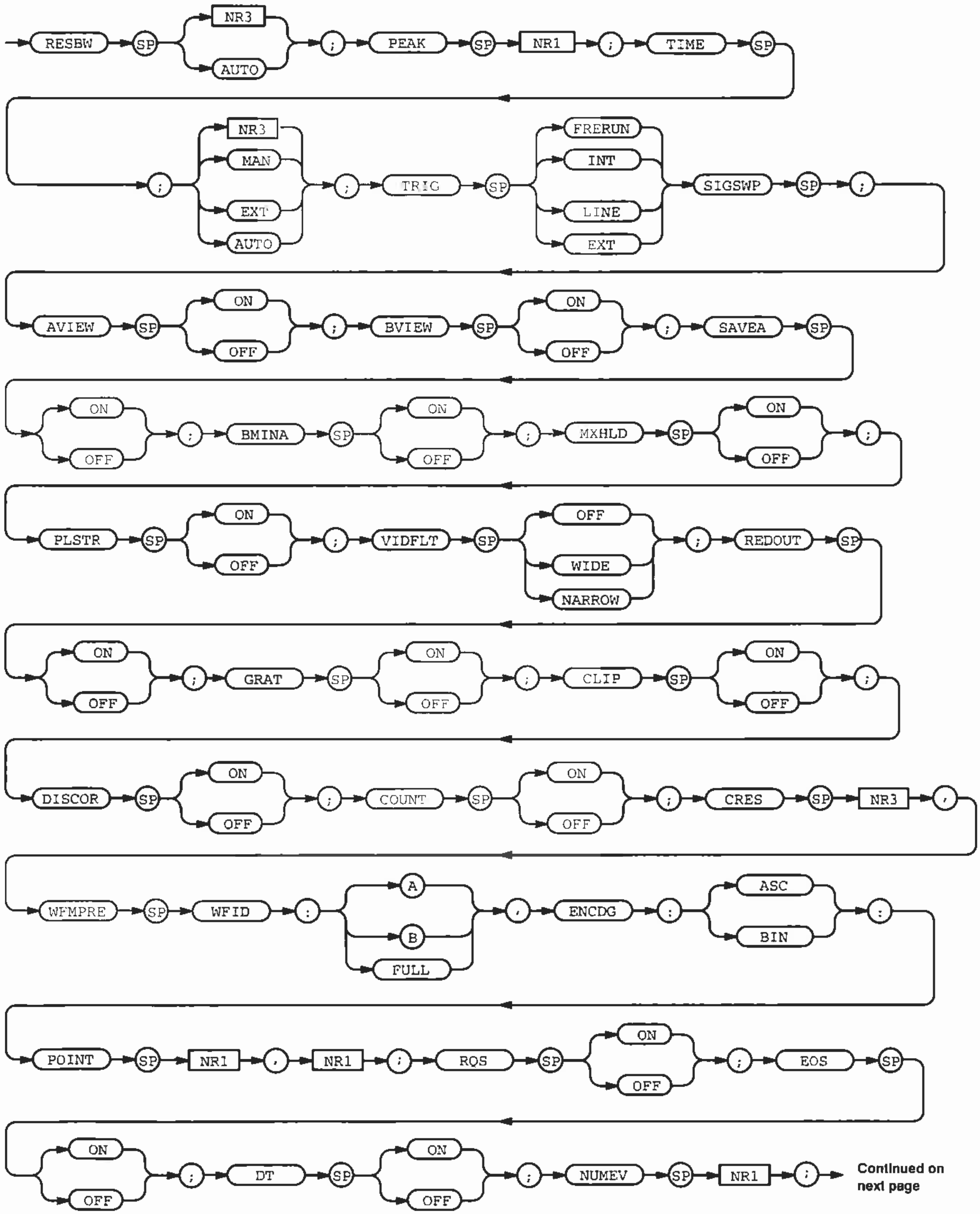
The instrument returns a string of commands that can be "learned" for later transfer to the spectrum analyzer when the same setup is desired. The response includes only those functions necessary for such a setup. To ensure that no interaction occurs that might alter the setup, some commands are turned off before the setup begins.

For identification purposes, the command headers are always returned in the response to SET? query even if HDR is turned off.

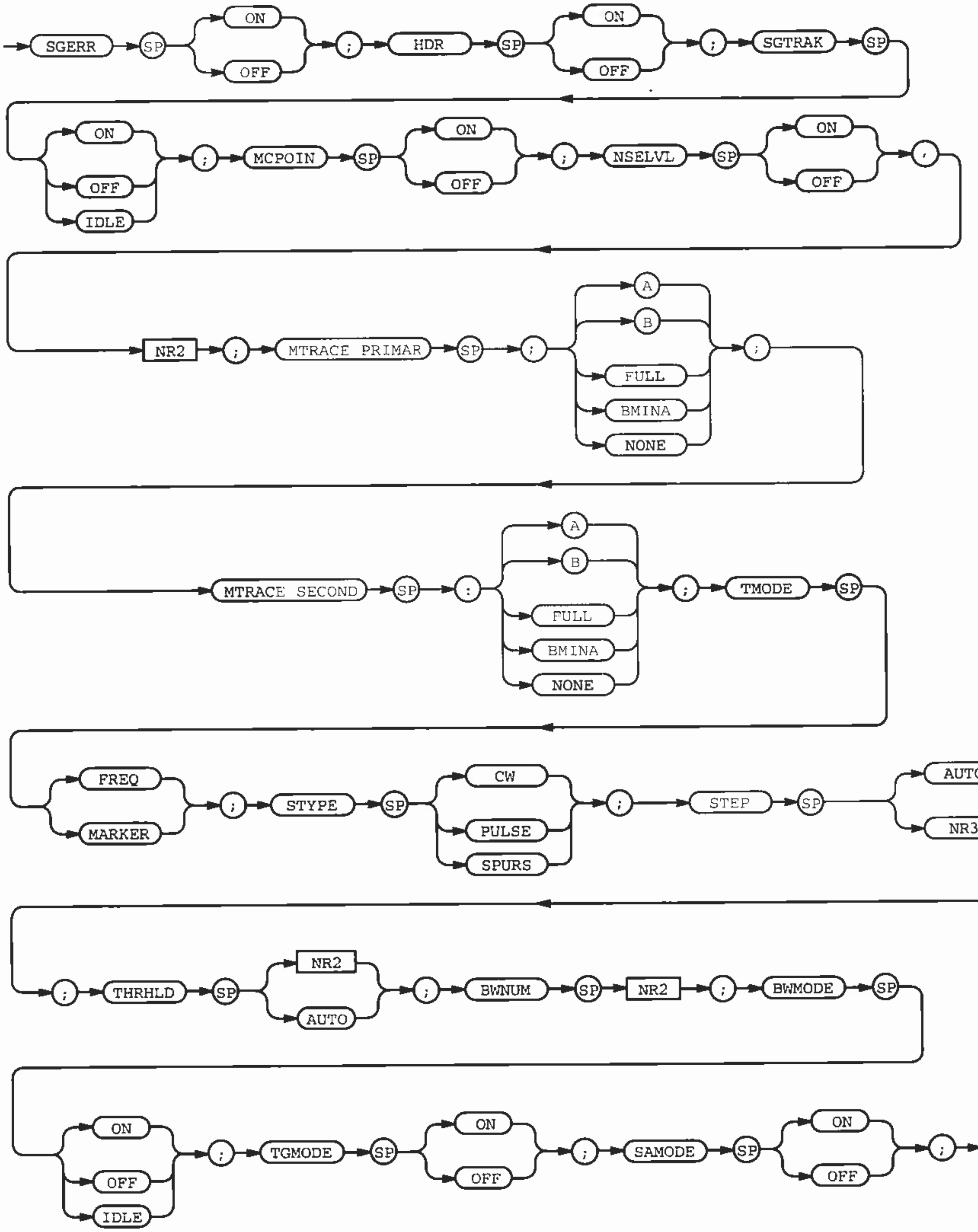
Response: Variable according to instrument type – typical only.



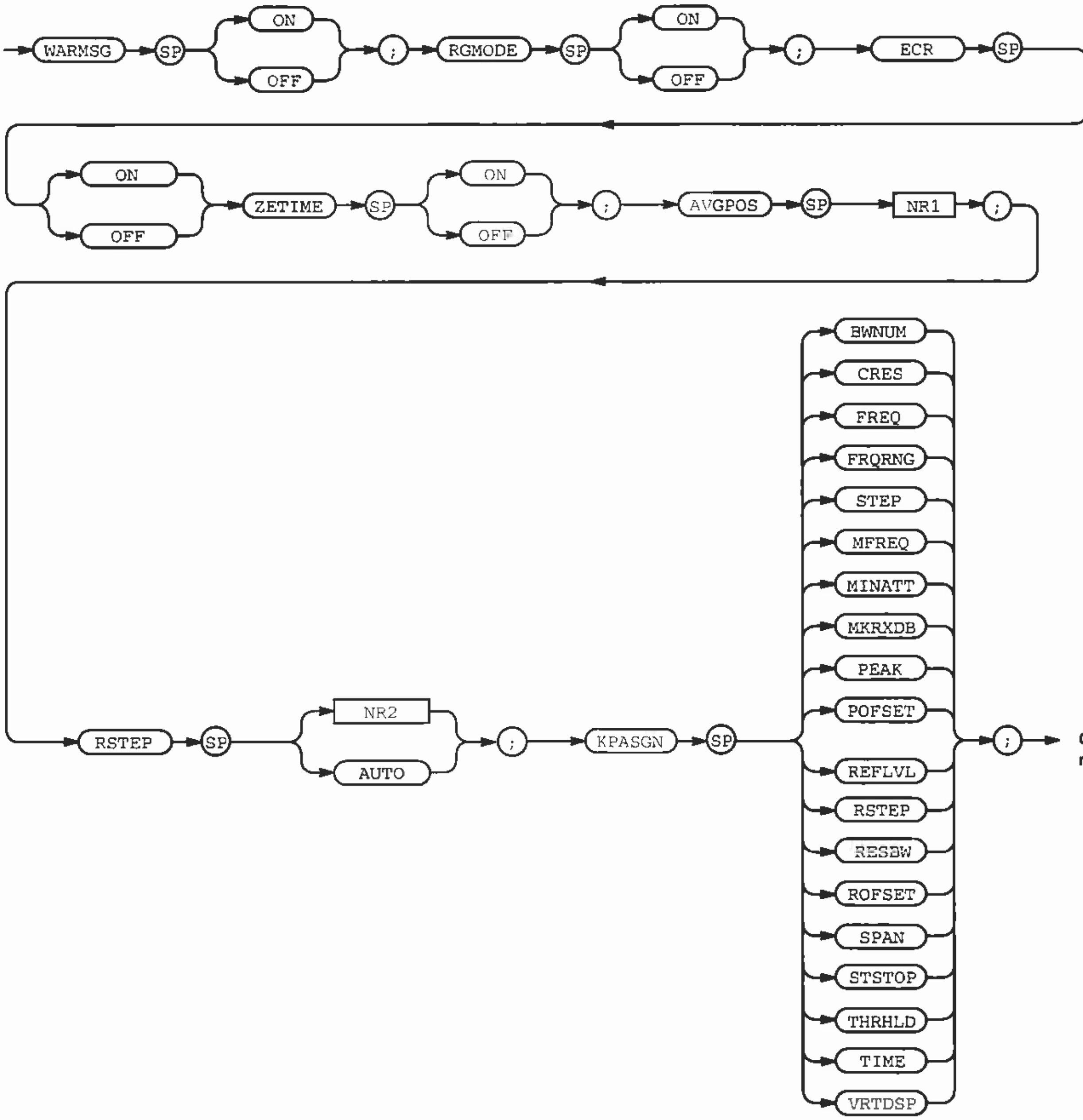
Continued on next page



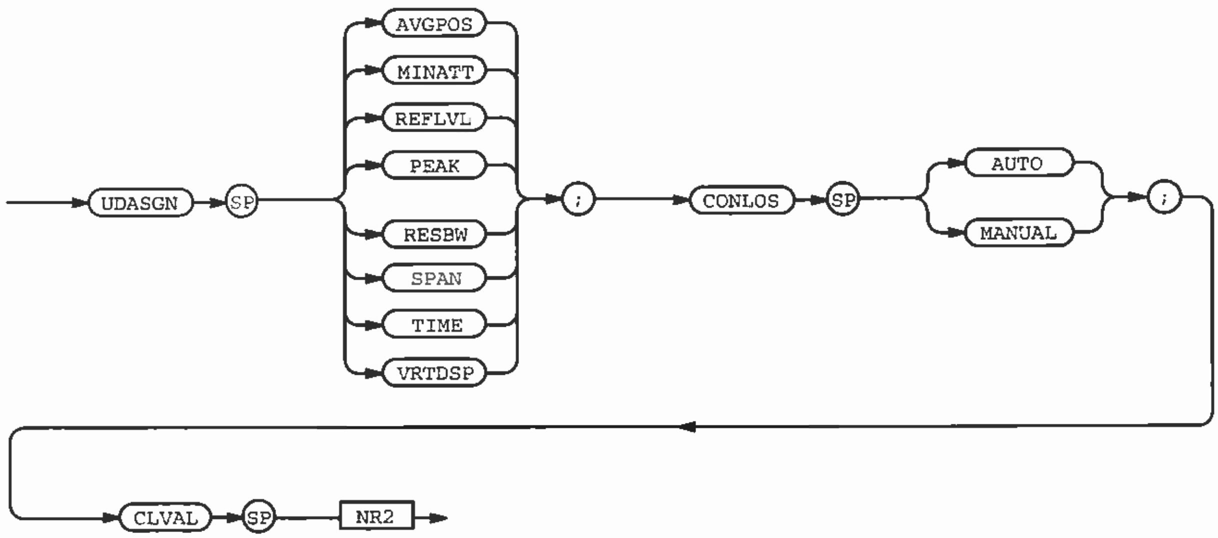
Continued on next page



Continued on next page

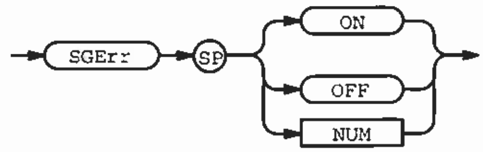


Continued on next page


UDASGN
SP
AVGPOS
MINATT
REFLVL
PEAK
RESBW
SPAN
TIME
VRTDSP
;
CONLOS
SP
AUTO
MANUAL
;
CLVAL
SP
NR2

SGErr (signal find error)

Syntax:



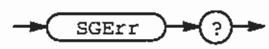
Arguments: **ON** — The spectrum analyzer asserts SRQ, if RQS is ON, when any of the following conditions exist:

- The internal signal identification routine cannot find the signal requested by the MFBIG, MRGTNX, MLFTNX, HRAMPL, or LRAMPL commands.
- The internal signal identification routine cannot find the amplitude requested by the PKFIND, PKCEN, MVRTDB, or MVLTBD commands.
- The requested bandwidth cannot be found by the BWMODE command.

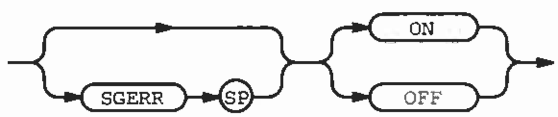
OFF — The spectrum analyzer does not assert SRQ when any of the above commands fail.

Interactions: RQS must be on for marker execution warning message 130 to be issued.

Query:



Response:

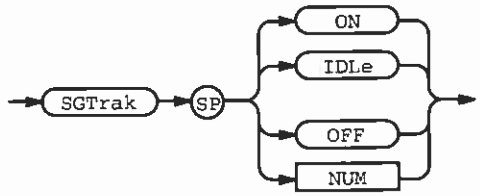


Macro Memory Used: 2 bytes.

Power-up Value: OFF.

SGTrak (signal track)

Syntax:



SGTRAK attempts to keep the signal at center screen as long as the signal does not drift off screen between sweeps.

Marker execution error message 120 is issued if the marker is on an inactive trace.

The signal track function takes effect at the end of the sweep after the SGTRAK command is given. If there is no signal at the marker location or the signal disappears, SGTRAK goes to IDLE.

SGTRAK is on during IDLE, but it is not tracking because there is no signal at the marker location.

SGTRAK is not available in Max Span (if MAX is entered, SGTRAK will be turned off.

Arguments: **ON** or **IDLE** — Signal track is turned on.

OFF — Signal track is turned off.

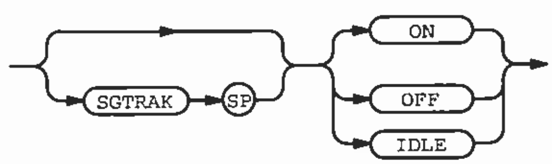
Interactions: If MARKER is OFF, SGTRAK sets MARKER to SINGLE.

Neither IDENT nor BWMODE are available while SGTRAK is ON; execution error message 101 will be issued if either is used.

The definition of the criteria for a signal is set by the THRHLD command.

Query:

Response:



Macro Memory Used: 2 bytes.

Power-up Value: OFF.

SIGswp (single-sweep)

Syntax:

If the instrument is not in single-sweep mode, the SIGSWP command activates single-sweep mode, which stops the current sweep.

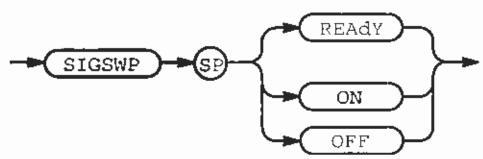
If the instrument is already in single-sweep mode, this command arms the sweep and lights the front-panel READY light, which remains lighted for the duration of the sweep.

The spectrum analyzer makes a single sweep of the selected spectrum when the conditions determined by the TRIG command are met. For most programming applications single sweep mode is the preferred mode of operation.

For some examples using SIGSWP, refer to the description of the WAIT command and the tabbed section *Programming Examples*.

Query:

Response:



The SIG? query response will not appear in a SET query response if single-sweep is not active (see SET?).

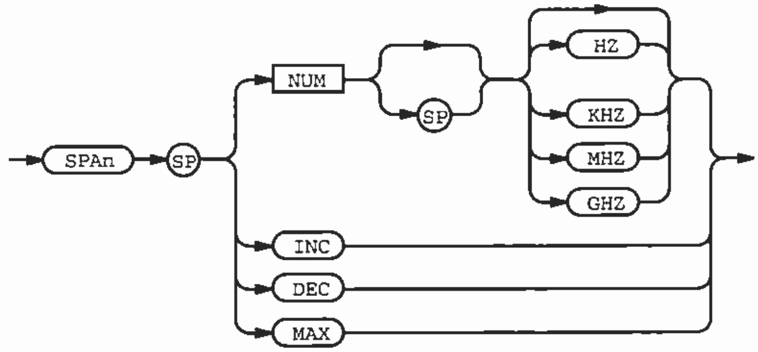
Interactions: Any TRIG command cancels the single-sweep mode.

Macro Memory Used: 1 byte.

Power-up Value: OFF.

SPAn (frequency span)

Syntax:



Arguments: **NUM** — The span/screen is selected. The value of the argument is rounded to two significant digits.

NOTE

In 49X compatibility mode, span/division is used instead of span/screen.

Zero converts the instrument to the time domain; in zero span mode, the instrument displays signals within its bandpass (RESBW) about its center frequency (FREQ).

If the number is too large, execution warning message 50 is issued, and the instrument defaults to MAX.

If the number is too small, execution warning message 111 is issued, and the spectrum analyzer defaults to the minimum span.

INC — The next larger span is selected in the front-panel 1–2–5 sequence, if possible.

DEC — The next smaller span is selected in the front-panel 1–2–5 sequence, if possible.

MAX — The entire frequency range in use is swept.

Range: Minimum value 100 Hz (2794, 2795, 2797) or 2 kHz (2792); maximum values for direct entry are given in Table 2-21.

The maximum value for INC or DEC is the 1–2–5 step that is less than or equal to the value in Table 2-21.

INC and DEC will default to maximum or zero, respectively, when they reach the end of the range.

If the entry is out of range, execution error message 31 is issued.

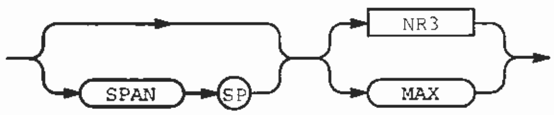
Interactions: Changing the SPAN setting turns ZEROSP OFF and MXSPN OFF.

Examples:

```
SPA 200
SPAN 50KHZ
SPA 100 MHZ
SPAN DEC
```

Query:

Response:



The numerical response is in span/screen.

If the instrument is in 49X compatibility mode, span/division is used instead of span/screen.

Macro Memory Used: 5 bytes.

Power-up Value: Maximum.

SSR (send service request)

Syntax:

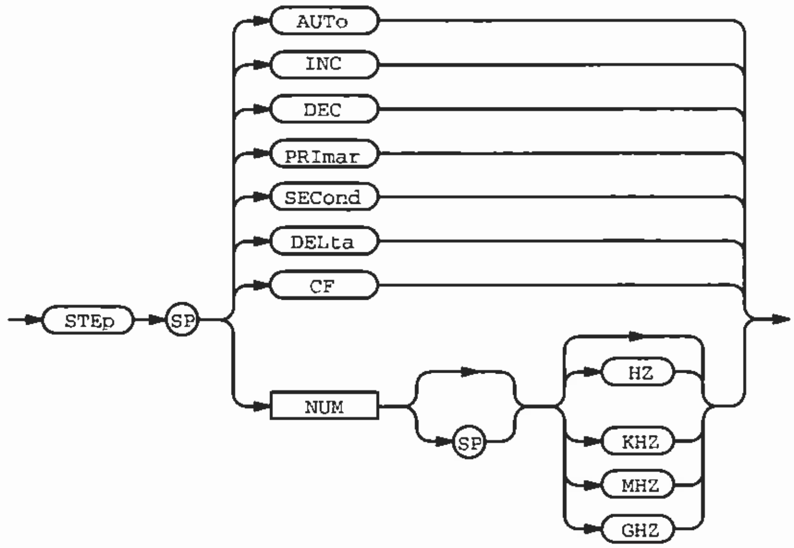
The SSR command asserts SRQ. The status byte will report this as an internal error warning, and system error 100, event 403 will be returned.

Interactions: The service request will be asserted only if RQS is ON.

Macro Memory Used: 2 bytes.

STEp (step size)

Syntax:



The STEP command sets the frequency step size used by the MSTEP and PSTEP commands.

Arguments:

AUTO — Sets the step size to 10% of the resolution bandwidth if the span is zero. If the span is not zero, the step size is set to one division of span.

INC — Increments the step in 1–2–5 sequence to the limits shown in Table 2-20. If the step size before the increment is not in the 1–2–5 sequence, the step size after the increment will be the next value that is in sequence.

DEC — Decrements the step in 1–2–5 sequence to the limits shown in Table 2-20. If the step size before the decrement is not in the 1–2–5 sequence, the step size after the decrement will be the next value that is in sequence.

Table 2-20: Step Size Limits

Instrument	Lower Limit	Upper Limit
2795	1 Hz	1.8 GHz
2797	1 Hz	7.1 GHz
2792	1 kHz	21 GHz
2794	1 Hz	155 GHz

PRIMAR — Sets the step size to the primary marker frequency.

SECOND — Sets the step size of the secondary marker frequency.

DELTA — Sets the step size to the absolute value of the difference in frequency between the primary and secondary markers.

CF — Sets the step size to the absolute value of the center frequency.

NUM — Sets the step size to the frequency specified.

Range: In the waveguide mixer bands, STEP can be used within a single band only.

Interactions: STEP DELTA and STEP SECOND cause marker execution error message 123 to be issued if MARKER is not set to DELTA.

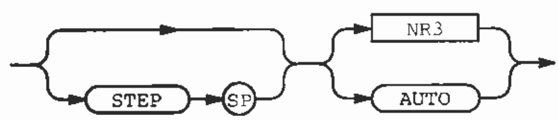
If the frequency or delta to which the step size is to be set is larger than 155 GHz, the step size will be set to 155 GHz.

Examples:

```
STE DEL
STE PRIMAR
STEP SEC
STE 20 KHZ
STEP 100 MHZ
```

Query:

Response:

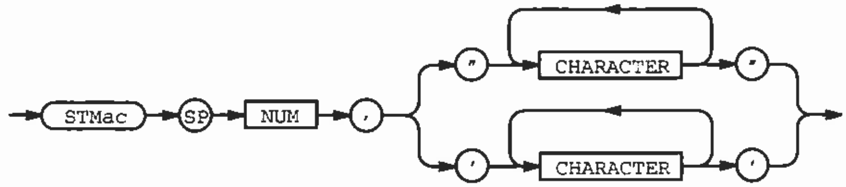


Macro Memory Used: 7 bytes.

Power-up Value: AUTO.

STMac (store macro)

Syntax:



The STMAC command tells the spectrum analyzer that the following GPIB commands will be stored as macro number NUM with a title string.

The spectrum analyzer will continue to store all commands until it receives the EMAC (end macro) command, which tells it that entry for macro NUM is done.

Any query lines given between STMAC and EMAC will be executed as they are read, and will not be saved as part of the macro.

Range: NUM can be from 0 to 7.

CHARACTER can be up to 22 characters.

Interactions: If the STMAC command is given and a macro is running or stopped, then that macro is aborted. If a macro is being entered and an illegal command is entered, macro entry is aborted and the instrument is returned to the regular operation mode.

Examples: STMAC 1,"HARMONIC TEST"

STNum (store number)

Syntax:

The STNUM command stores the X register (XREG) value in the variable NUM.

If NUM is less than 1, NUM is set to 1.

If NUM is out of range, macro execution error message 176 is issued.

Range: NUM can be from 1 to 30.

Examples:

`STNUM 1`	Copies the number in XREG to variable number 1.
`STNUM 9`	Copies the number in XREG to variable number 9.

Macro Memory Used: 2 bytes.

STOre (store settings)

Syntax:

Arguments: NUM — The instrument control settings are stored into the memory location specified by NUM.

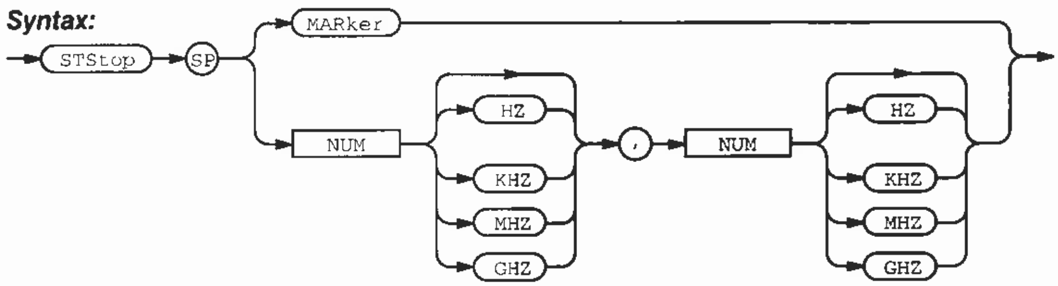
Range: NUM can be from 0 to 9.

Macro Memory Used: 2 bytes.

Power-up Value: The instrument STORES its current settings in memory 0 automatically when the power is turned off, overwriting any previously-stored settings.

STStop (start-stop sweep)

Syntax:



Arguments: **MARKER** — The frequency and span are set so that the instrument sweeps over the frequency range delimited by the markers.

The lowest frequency marker sets the start frequency, and the highest frequency marker sets the stop frequency.

NUM — The starting frequency of the display is set to the first NUM.

The ending frequency is set to the second NUM.

Execution error message 28 is issued if the second NUM is less than the first NUM.

Range: Bands 1 and 6–12: Band limits of the band the start and stop frequencies were set in, not the band the instrument is currently in.

Bands 2 – 5: Any value within the bands.

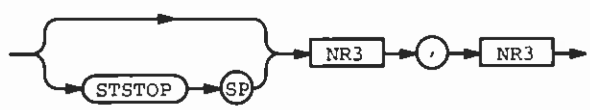
Interactions: Marker execution error message 123 is issued if the STSTOP MARKER command is given when MARKER is not set to DELTA.

Examples:

```
STS MARKER
STSTOP 10MHZ,130MHZ
STS 100000HZ,66MHZ
```

Query:

Response:



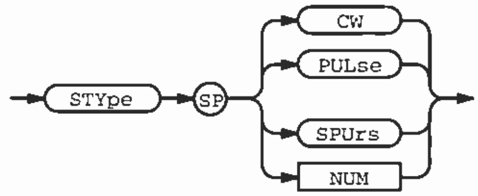
The response is the present start and stop frequency (in that order), whether the values were entered as start-stop frequencies or result from the combination of a center frequency and span.

Macro Memory Used: 12 bytes.

Power-up Value: Start frequency, 0 Hz; stop frequency, 1.8 GHz.

STYpe (signal type)

Syntax:



Arguments:

CW — Continuous wave signals are identified.

PULSE — Pulsed signal groups are identified.

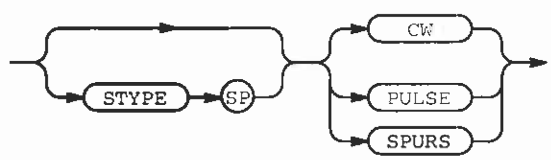
SPURS — All signals above the threshold are identified.

NUM —
0 = CW
1 = PULSE
2 = SPURS

Interactions: The STYPE command affects the marker finding commands MFBIG, MLFTNX, MRGTNX, HRAMPL, and LRAMPL.

Query:

Response:



Macro Memory Used: 2 bytes.

Power-up Value: CW.

Examples: Figure 2-4 is a signal enlarged to show how the spectrum analyzer locates the signal peak with one of the signal processing commands.

The signal processing commands are MLFTNX, MRGTNX, MFBIG, HRAMPL, and LRAMPL.

The spectrum analyzer measures both the individual left-most and right-most peaks of a signal. From this reading, the spectrum analyzer calculates the exact center of the signal. If this location is one of the maximum digital storage points, the marker is positioned here. If, as in Figure 2-4, the calculated center of the signal is not equal to the maximum digital storage point, the marker is positioned on the closest point to the center.

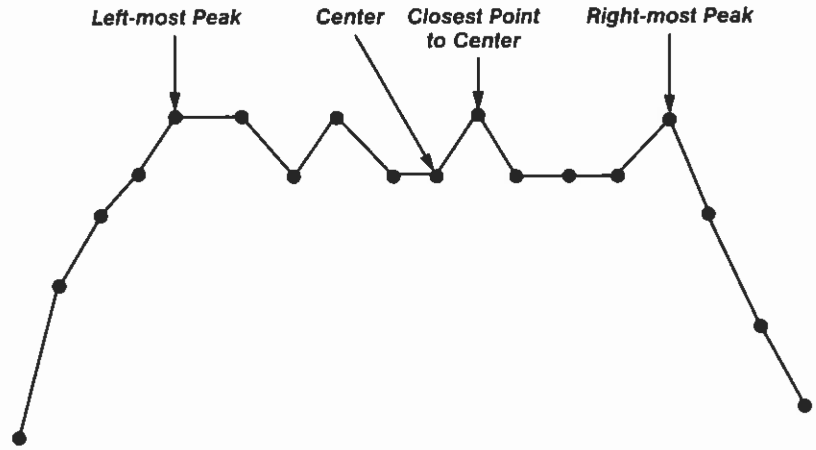


Figure 2-4: Locating the Signal Peak

To the finding routine, a "candidate" signal consists of a peak above threshold and two points (one on each side of the peak) that are 3 dB below the peak. The location of the candidate signal is the highest amplitude point on the signal. Whether or not the candidate is recognized as a signal depends upon the processing mode chosen.

When SPURS is chosen, all candidates are taken to be signals.

When CW is chosen, a signal (to be a signal) must be at least half as wide as would be predicted from the resolution filter in use. (Note that this is not the same algorithm as the one used by the data-point-related commands. In particular, the data-point algorithm looks for a specific width, while the marker-related algorithm looks only for a minimum width. Note also that if the span is wide in comparison with the resolution bandwidth, there may be no difference between SPURS and CW.)

When PULSE is chosen, if two candidate signals are within two minor divisions (0.4 of a major division), they are assumed to be either time-related lines or spectral lines belonging to the same pulse. This extends to multiple lines; in a group of such lines, the highest-amplitude line will be identified as the center of the signal.

Figures 2-5 through 2-9 illustrate the use of STYPE. All of the figures use the signal finding command of MRGTNX. Any of the other signal finding commands (MLFTNX, MFBIG, HRAMPL, and LRAMPL) work similarly, according to their specific function.

Figures 2-5, 2-6, and 2-7 — If CW was selected, the spectrum analyzer would not identify any signal because none of the signals displayed meets the minimum bandwidth criteria.

If PULSE was selected, the signals labeled D, E, and F would be identified because the other signals in the display are less than 2 minor divisions apart. If the signals were greater than 2 minor divisions apart, PULSE would have identified all labeled signals (A, B, C, etc.).

If SPURS was selected, all signals would be identified (A, B, C, etc.).

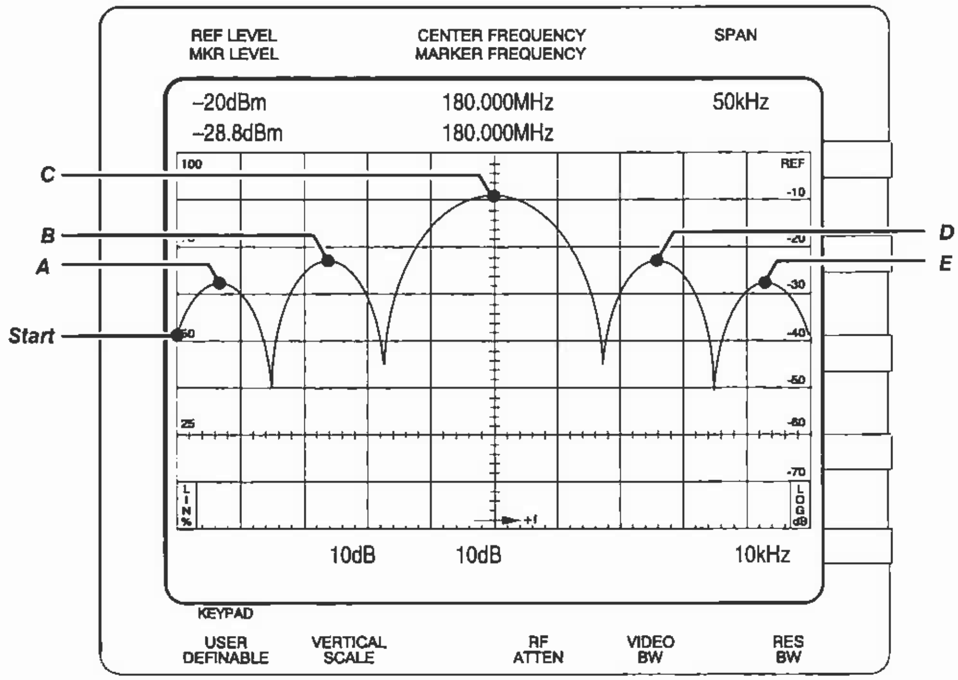


Figure 2-5: Signal Finding Example 1

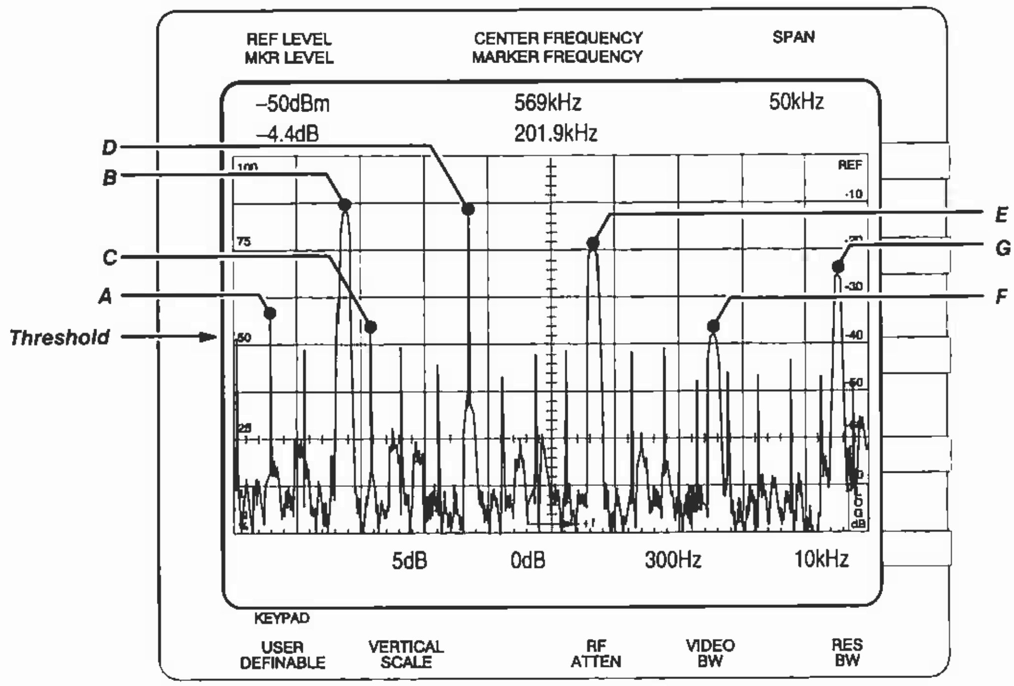


Figure 2-6: Signal Finding Example 2

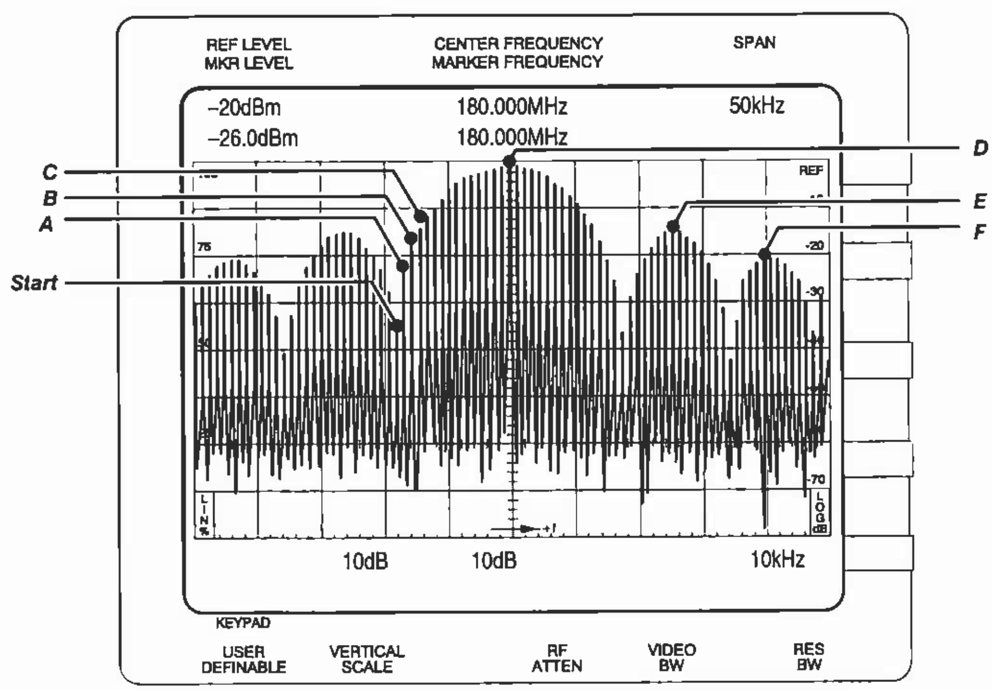


Figure 2-7: Signal Finding Example 3

Figure 2-8 — The MRGTNX selection begins at the left screen margin. With this display, CW, PULSE, and SPURS will each identify all of the signals because they all meet the minimum bandwidth criteria (that is, the selections would be A, B, C, D, and E).

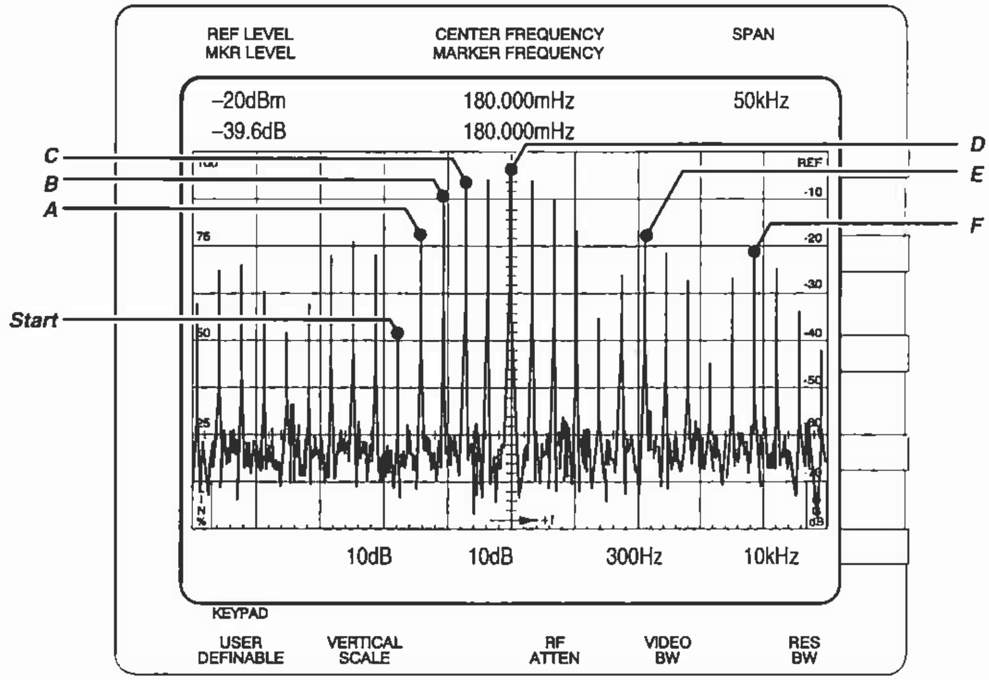


Figure 2-8: Signal Finding Example 4

Figure 2-9 — In Figure 2-9 the threshold is assumed to be −70 dBm. If CW was selected, signals B, E, F, and G would be identified. The other signals would not be identified because they do not meet the minimum bandwidth criteria.

If PULSE was selected, signals A, B, D, E, F, and G would be identified. Signal C would be skipped, because it is within 2 minor divisions from signal B. The PULSE algorithm will think signal C is a part of signal B.

If SPURS was selected, all signals would be identified.

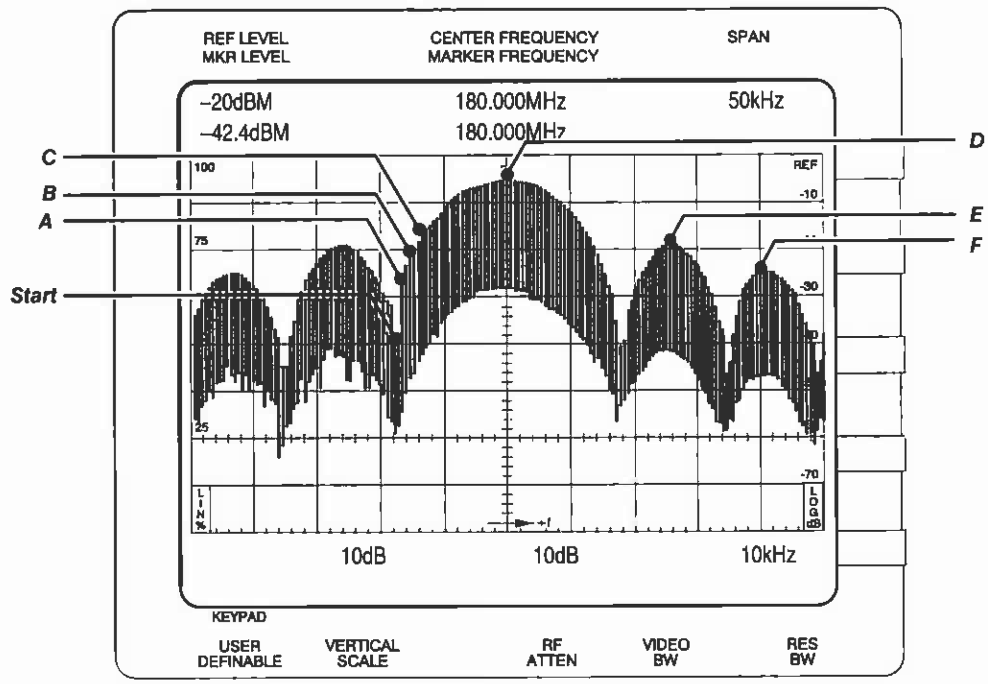


Figure 2-9: Signal Finding Example 5

SUBt (subtract X from Y register)

Syntax: SUBt

The SUBT command subtracts the contents of the X register from the contents of the Y register and puts the result in the X register (XREG = YREG − XREG).

Range: ±9.999 999 999 999 E+99 to ±9.999 999 999 999 E−99.

Interactions: The Y register is unchanged after the YREG command.

Macro Memory Used: 1 byte.

SWEep (take a sweep)

Syntax:

→ SWEep →

The SWEEP command will start a new sweep and wait until the sweep has finished before executing the next macro command. The SWEEP Command is equivalent to the SIG; WAIT command sequence.

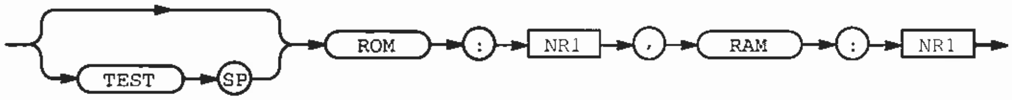
Macro Memory Used: 1 byte.

TESt? (internal test)

Query:

This query checks the system ROM and RAM.

Response:

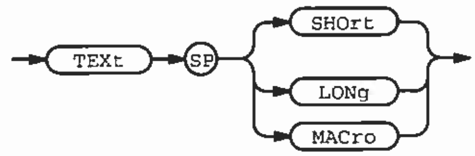


The TEST query response consists of two decimal numbers that indicate if a ROM or RAM IC was found to be defective.

These numbers must be translated to their binary equivalents to determine the ROM and RAM locations. (If all ROM and RAM are good, the TEST query response will be ROM:0,RAM:0.)

TEXt (text mode)

Syntax:



Arguments: **SHORT** — The readout is switched to the normal mode with a spectrum display.

When the RDOUT commands are used, the normal top and bottom readout is not displayed, and customer-specified characters are sent to the top and bottom lines.

LONG — The readout is switched to the 16-line mode without a spectrum display.

RDOUT commands will fill the top line first, then fill successive lines until all lines have characters. When all 16 lines have characters, the entire screen scrolls up. Send TEXT LONG again to clear the page of the readout and begin sending characters to the top line again.

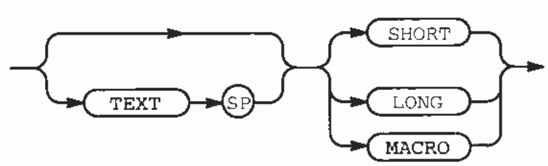
MACRO — The readout is switched to the 16-line mode without a spectrum display, and the macro readout buffer is displayed.

The PRINT command is used to display data into the macro readout buffer. The CLEAR command can be used to clear the macro readout buffer.

Interactions: If the CRT readout is not in the NORMAL mode when TEXT is executed, the readout affected by the command will be cleared (this could be used as a "page" command to clear the screen for new text). RDOUT NORMAL restores normal spectrum analyzer readout.

Query:

Response:

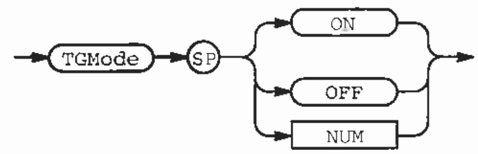


Macro Memory Used: 2 bytes.

Power-up Value: SHORT.

TGMode (tracking generator mode)

Syntax:



The TGMODE command allows higher frequency accuracy when using a tracking generator.

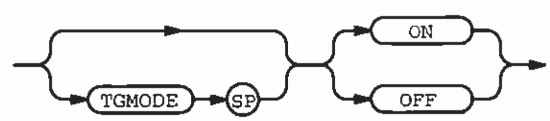
When TGMODE is ON the frequency correction factors for all resolution bandwidth filters wider than 10 kHz are disabled. These wide filters may be centered too far from 10 MHz for the difference to be corrected with the Tracking Adjust control on the tracking generator.

Arguments: **ON** — The tracking generator mode is turned on.

OFF — The tracking generator mode is turned off.

Query:

Response:

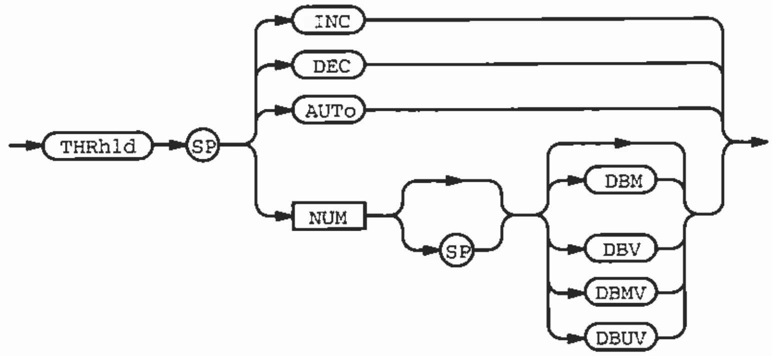


Macro Memory Used: 2 bytes.

Power-up Value: OFF.

THRhld (marker threshold)

Syntax:



NOTE

To ensure the correct response, all of the letters in each of the unit mnemonics for the THRHLD command must be entered, not just the first three letters as for other mnemonics.

The THRHLD command sets the threshold for the marker signal find commands MLFTNX, MRGTNX, MFBIG, HRAMPL, LRAMPL, SGTRAK, PKFIND, PKCEN, and BWMODE. THRHLD moves in 1 dB steps.

Arguments: **AUTO** — The threshold is set to approximately the sensitivity specification plus RF attenuation plus the video filter offset. The video filter offset is 10 dB if there is no filter, 4 dB if WIDE is ON, and 2 dB if NARROW is ON.

INC — Increments the threshold by 1 dB.

DEC — Decrements the threshold by 1 dB.

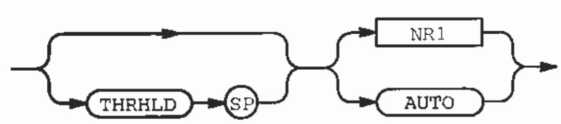
NUM — The threshold is set to the level input. If no units are specified, the spectrum analyzer assumes the current reference level units.

Examples:

```
THRHLD AUTO
THRHLD  -40DBMV
```

Query:

Response:

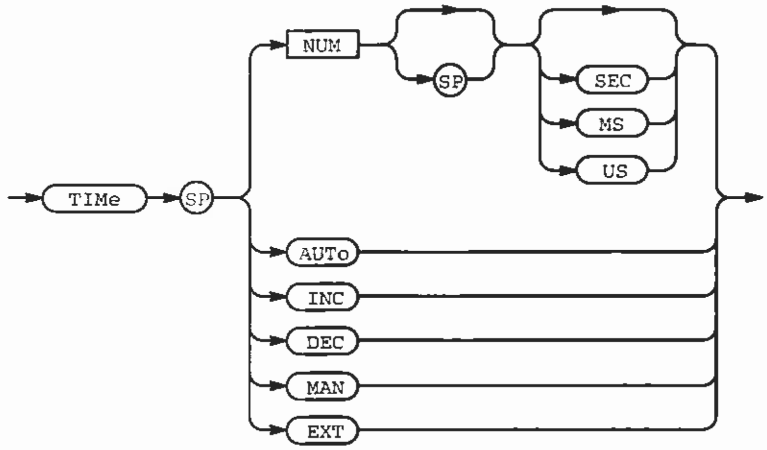


Macro Memory Used: 4 bytes.

Power-up Value: AUTO.

TIMe (time/screen)

Syntax:



Arguments:

NOTE

In 49X compatibility mode, time/division is used instead of time/screen.

NUM — 1–2–5 sequence in the range 20.0 E^{-6} to 10. Numbers not in this sequence are rounded to the nearest step.

If the number selected is out of range, execution error message 37 is issued.

AUTO — The instrument selects the fastest sweep allowed for calibrated response.

INC — The sweep rate is changed +1 in the sequence, if possible.

DEC — The sweep rate is changed −1 in the sequence, if possible.

MAN — The sweep is coupled to the MANUAL SCAN control so you can manually scan the spectrum. As the control is turned, the horizontal position of the CRT beam and the instrument front-end tuning are varied.

EXT — The sweep is coupled to HORIZ|TRIG (EXT IN) on the rear panel. The horizontal position of the CRT beam and the instrument front-end tuning are varied by an external signal. A signal in the range 0 to +10 V scans the spectrum.

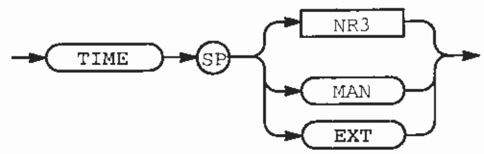
Interactions: Too fast a sweep speed for a given resolution bandwidth will uncalibrate the display. For digital storage to properly acquire spectrum data, 100 ms is the maximum usable sweep rate.

Examples:

```
TIME 1
TIM 10 MS
TIME MAN
```

Query:

Response:



The SET? response includes AUTO as a possible argument (see SET? under Instrument Parameters in the System Commands and Queries section of this manual).

The numerical response is in time/screen.

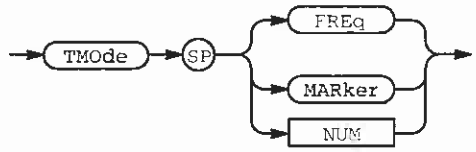
If the instrument is in 49X compatibility mode, time/division is used instead of time/screen.

Macro Memory Used: 2 bytes.

Power-up Value: AUTO (100 ms).

TMOde (set tune mode)

Syntax:



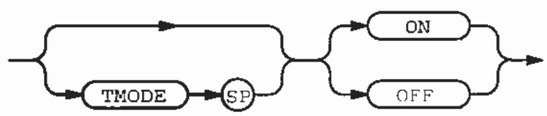
Arguments: **MARKER** — The PSTEP and MSTEP commands will change the marker frequency.

FREQ — The PSTEP and MSTEP commands will change the center frequency.

NUM —
0 = FREQ
1 = MARKER

Query:

Response:



Interactions: If MARKER is OFF, TMODE MARKER sets MARKER to SINGLE.

TMODE sets the tuning mode that the front-panel FREQUENCY/MARKERS knob will have when the spectrum analyzer is returned to local control.

Macro Memory Used: 2 bytes.

Power-up Value: FREQ.

TOPsig (move to top of graticule)

Syntax:

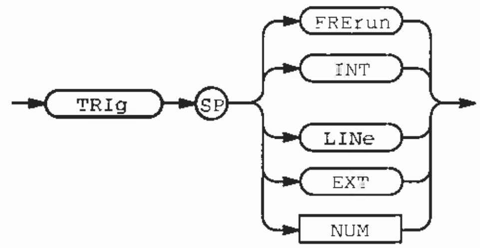
This command sets the reference level to the level of the display data point (or as close as possible, given the specified vertical display and reference level accuracies).

This command does not acquire a new display data point or digital storage waveform. Therefore, if a new waveform is acquired after TOPSIG is run, the display data point may no longer match the signal of interest.

Macro Memory Used: 1 byte.

TRIg (triggering)

Syntax:



This command sets the instrument triggering mode to one of the following states:

Arguments: **FRERUN** — The instrument sweep is allowed to run repetitively.

A trigger is not required (or is ignored), so the instrument generates a sweep immediately after the hold-off period that follows the previous sweep.

This is a simple and common setup used to acquire a spectrum for manual operation.

INT — The spectrum analyzer generates a sweep only when it is triggered by an input signal.

A signal amplitude of at least 2 divisions is required and must occur after the hold-off period that follows the previous sweep.

This sweep mode is often used to examine time-domain signals in the zero span mode (ZEROSP).

LINE — The power line input is selected as the trigger signal (useful in both the frequency domain and time domain modes for signals with components related to the power line frequency).

EXT — The sweep is triggered by a signal with an amplitude of at least +1.0 V peak connected to HORIZ|TRIG (EXT IN) on the rear panel.

NUM —
0 = FRERUN
1 = INT
2 = LINE
3 = EXT

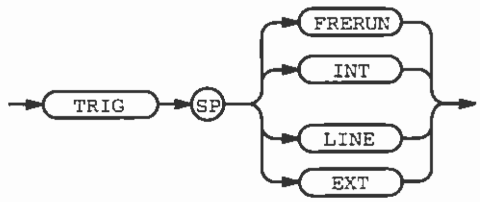
Interactions: The signal frequency required for internal trigger is related to the center frequency.

In the frequency domain mode, the required frequency corresponds to 1/2 division to the left of the left graticule edge. The required frequency must be within the selected frequency range.

In the time domain mode, the required frequency is the center frequency.

Query:

Response:

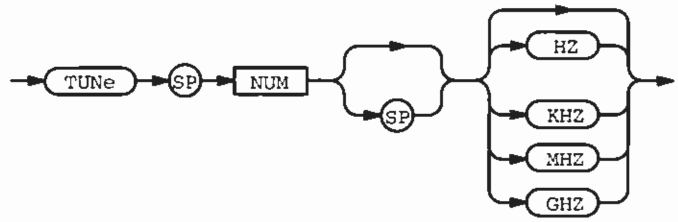


Macro Memory Used: 2 bytes.

Power-up Value: FRERUN.

TUNe (incremental frequency change)

Syntax:



Arguments: NUM — The instrument changes its center frequency by using the value of the command argument as an offset to its previous center frequency.

Range: The frequency resulting from using the TUNE command must be within the frequency range in use except in Max span.

The maximum value for each frequency range (that is, the value needed to tune from one end of the range to the other) is given in Table 2-21.

Table 2-21: Ranges for the TUNE, FIRST, SECOND, FRQRNG, STEP, and SPAN Commands

Band	Freq Range (GHz)	1st IF (MHz)	1st LO Range (MHz)	N	2nd LO Range (MHz)	Tune Range (GHz)	Maximum Span
1	0–1.8	2072	2072–3872	1–	2181–2183	1.8	170 MHz
2	1.7–5.5	829	2529–6329	1–	718.83–719.17	3.8	370 MHz
3	3.0–7.1	829	2171–6271	1+	718.83–719.17	3.8	400 MHz
4	5.4–18	2072	2076–6276	3–	718.83–719.17	12.6	1.2 GHz
5	15–21	2072	4309–6309	3+	2181–2183	6	590 MHz
6	18–27	2072	2655–4071	6+	2181–2183	9	790 MHz
7	26–40	2072	2443–3793	10+	2181–2183	14	1.3 GHz
8	33–60	2072	3092–5790	10+	2181–2183	27	2.6 GHz
9	50–90	2072	3195–5862	15+	2181–2183	40	3.9 GHz
10	75–140	2072	3170–6000	23+	2181–2183	65	6.4 GHz
11	110–220	2072	2917–5890	37+	2181–2183	110	10 GHz
12	170–325	2072	2998–5841	56+	2181–2183	155	15 GHz

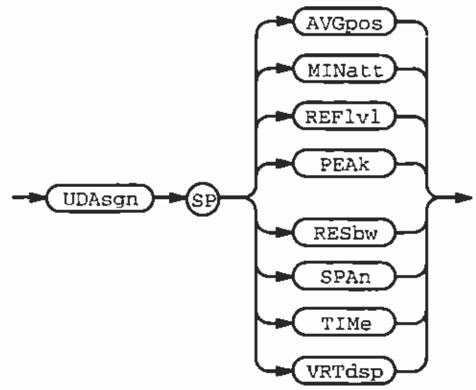
Examples:

```
TUN 10 MHZ
TUNE 1.0E6
TUNE 100 KHZ
```

Macro Memory Used: 6 bytes.

UDAsgn (user definable knob assign)

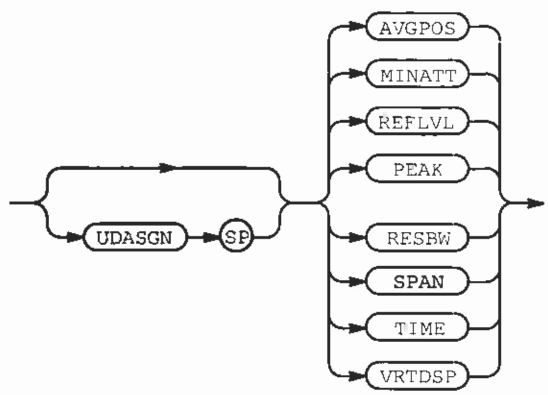
Syntax:



This command assigns the user definable knob to the function corresponding to the argument specified. For more information, refer to the command description for the argument.

Query:

Response:



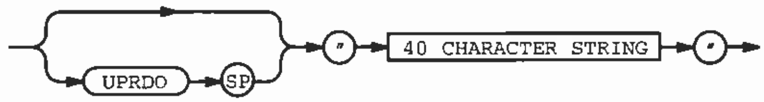
Macro Memory Used: 2 bytes.

Power-up Value: SPAN.

UPRdo? (upper readout)

Query:

Response:



This query returns the upper line of the readout information that appears on the instrument CRT.

Response: **CHARACTER** — Characters are from the CRT readout. Blanks are transmitted as spaces.

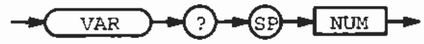
Regular readout that would be displayed if GPIB did not have control (whether visible on the screen or not) is the readout returned by the query, not a message sent to the instrument by RDOUT.

With AVIEW and SAVEA both ON and BMINA and BVIEW both OFF, the returned readout will be the saved readout.

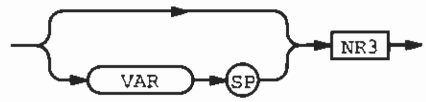
Refer also to the recall display (DRECAL) or the Save A waveform (SAVEA) command descriptions.

VAR? (variable)

Query:



Response:

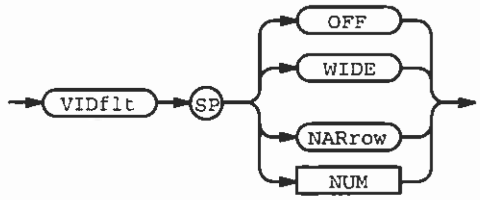


The VAR query response will return the value of variable NUM. If NUM is out of range, 0 is returned (no error message is issued).

Range: From 1 to 30.

VIDflt (video filter)

Syntax:



This command controls the video bandwidth of the instrument.

Arguments:

OFF — Both the wide and the narrow video filters are turned off.

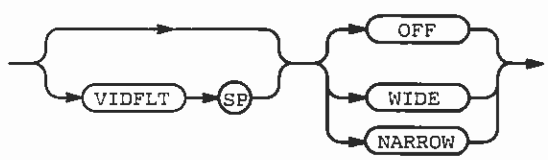
WIDE — A filter is turned on in the video amplifier (after the detector) to average noise in the display. The wide filter reduces video bandwidth to about $1/30$ of the selected resolution bandwidth.

NARROW — The narrow video filter reduces video bandwidth to about $1/300$ of the selected resolution bandwidth.

NUM —
0 = OFF
1 = WIDE
2 = NARROW

Query:

Response:



Interactions: It may be necessary to reduce sweep speed (TIME) to maintain a calibrated display unless TIME is in AUTO, because the instrument's overall bandwidth is reduced by video filtering.

Macro Memory Used: 2 bytes.

Power-up Value: OFF.

VPOs (vertical position)

Syntax:

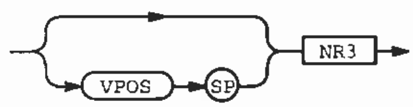
This command sets the CRT vertical display adjustment to a position corresponding to a DAC value specified by NUM.

Arguments: NUM – is a value loaded into the vertical position DAC.

Range: From 0 to 255, where 0 corresponds to the bottom-most display position and 255 the top-most.

Query:

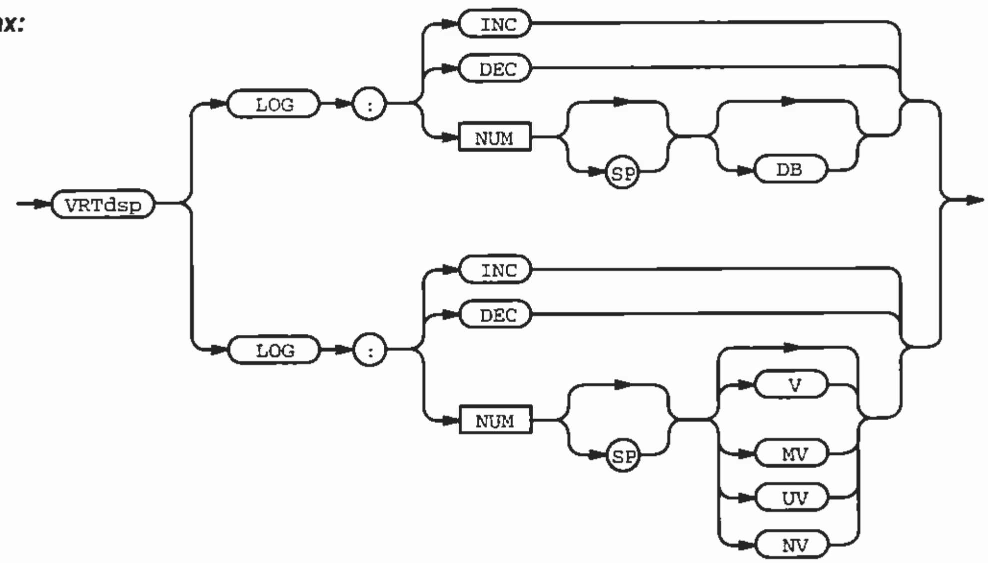
Response:



Power-up Value: Last front panel setting.

VRTdsp (vertical display)

Syntax:



Arguments: **LOG** — The display is scaled to the dB/division specified by integers in the range 1 to 15; non-integers are rounded.

INC or DEC changes the scale factor in 1 dB increments, if possible.

VRTDSP LOG values outside this range cause execution error message 36 to be issued.

LIN — The display is scaled in volts/division.

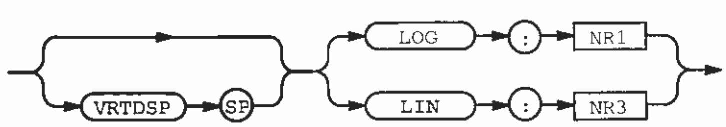
NUM is adjusted to set the reference level 8X volts/division to the nearest 1 dB. If NUM is omitted, the display is scaled to leave the reference level at its current value; Volts/division = 1/8 of the volts equivalent of REFLVL.

INC or DEC changes the scale factor to correspond to a 1 dB change in REFLVL if RSTEP is not in AUTO or in a 1–2–5 sequence it is.

Out-of-range values cause the instrument to report execution error message 34.

Query:

Response:



Range: Log is 1–15 dBV.

Lin is 39.6 nV/div to 2.8 V/div, if the reference level offset is zero.

Examples:

```
VRT LOG:3
VRTDSP LOG:2DB
VRT LOG:1 DB
VRTDSP LIN
VRT LIN:2
VRTDSP LIN:1.5MV
VRT LIN:75 NV
VRTDSP LIN:INC
```

Macro Memory Used: 3 bytes.

Power-up Value: LOG:10 dB/division.

WAIt (wait for end of sweep)

Syntax: → WAIt →

The spectrum analyzer delays action on commands in its input buffer that follow the WAIT command sets the status byte to busy, and does not input device-dependent messages.

The wait condition is ended in either of two ways:

- WAIT ends if an end-of-sweep is present. If this occurs, the controller can request spectrum data and be guaranteed that the data has been updated. The request message would be similar to the following:

  ```
  SIGSWP;SIGSWP;WAIT;WFMPRE?;CURVE?
  ```

  The first SIGSWP command sets the spectrum analyzer to the single-sweep mode if it was previously in a repetitive-sweep mode. The next SIGSWP arms the sweep, and WAIt delays further action until the sweep completes. The message ends by the request of a waveform preamble and data.

  If the sweep is in the single-sweep mode and is not armed (the READY light is off) when the WAIT command comes up, the spectrum analyzer continues to execute the message in the buffer and does not wait.

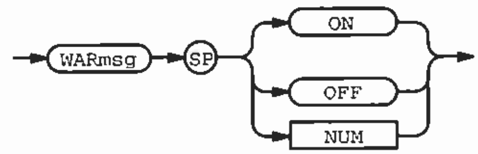
- WAIT is ended if DCL or SDC (while listener-addressed) is received. This empties the input and output buffer so any commands that follow WAIt are discarded. (See STATUS BYTE in Section 3.)

Interactions: WAIT delays completion of any portion of a message that follows or input of new messages, until one of the ending conditions just outlined occurs.

Macro Memory Used: 1 byte.

WARmsg (warning message)

Syntax:

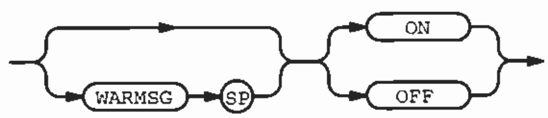


Arguments: **ON** — All warning messages will be issued.

OFF — No warning messages will be issued.

Query:

Response:



Macro Memory Used: 2 bytes.

Power-up Value: ON.

WAVfrm? (waveform)

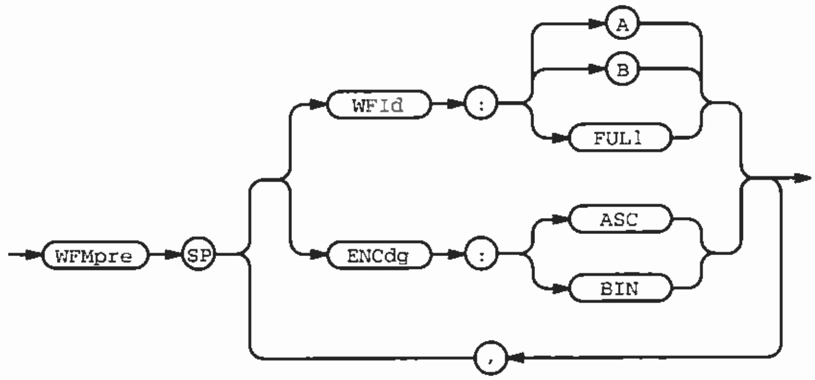
Query:

The WAVFRM query response is the same as the response to WFMPRE?;CURVE?.

The most recent WFID and CRVID arguments select whether A, B, or both memories are selected for data transfers and waveform processing in ASCII or binary numbers (refer to both the WFMPRE and CURVE queries).

WFMpre (waveform preamble)

Syntax:



NOTE

The WFMPRE command has no effect on the Marker Finding commands.

The WFID path of the waveform preamble command allows the choice of either the A or B waveform or both (FULL).

Following the ENCDG path, the waveform preamble command allows selection of either ASCII-coded decimal or binary waveform data.

The contents of digital storage determine if a half-resolution or full-resolution waveform is obtained, or two different waveforms. This is because of the way digital storage is handled in the spectrum analyzer.

The B waveform is updated with each sweep; the A waveform is updated only if SAVEA is OFF. The values stored for each waveform are alternate points on the current display (i.e., B, A, B, A, B, A,.. beginning at the left edge of the screen and moving to the right).

With SAVEA OFF, each waveform is a half-resolution replica of data from the last sweep (A data points offset by 1 from corresponding B data points).

Full-resolution (FULL) transfers merge the two waveforms for 1000 data points (100 points/div), and half-resolution transfers (A or B) separate the waveforms for 500 data points (50 points/division).

If the waveforms are separated and SAVEA is OFF, signals resolved to a single point (with very narrow resolution bandwidths compared to span) appear in either A memory or B memory, but not both.

With SAVEA ON, only the B waveform is filled with data from the current sweep, so transfers can involve two unrelated waveforms.

Arguments:

WFID — Either the A or B waveform or both A and B (FULL) waveforms are selected for data transfers and waveform processing.

ENCDG — Either ASCII-coded decimal numbers or binary numbers are selected for data transfer.

The two arguments may be selected independently or strung together in the same command.

The WFMPRE command cannot be used within a macro.

Interactions:

The WFID portion of any previous WFMPRE command or the CRVID portion of any previous CURVE command is cancelled. A WFID or CRVID other than FULL sets MCPOIN to OFF. MCPOIN sets WFID to FULL.

Power-up Value:

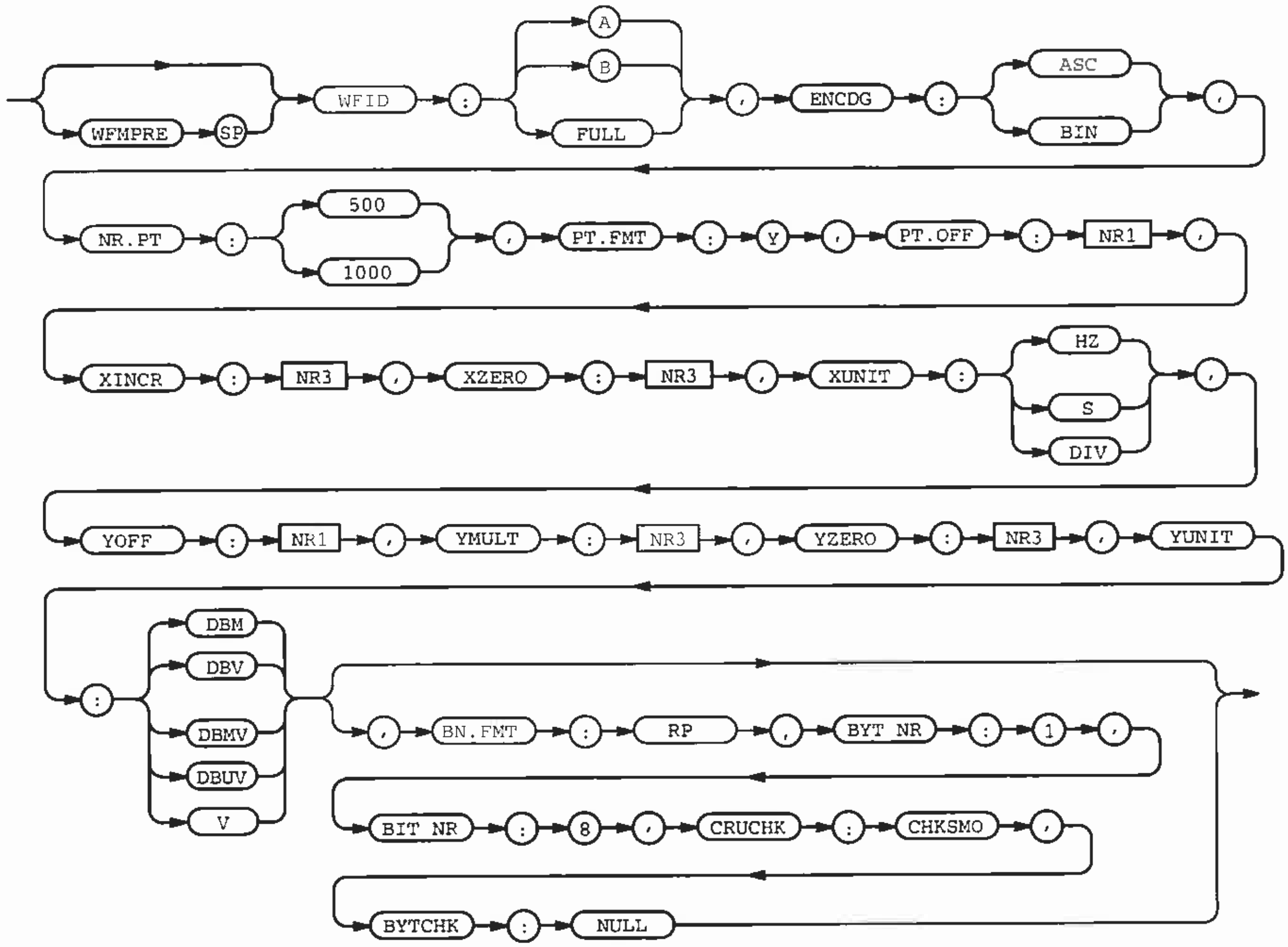
Full (1000 point), ASCII-coded digits.

Examples:

```
WFMPRE WFID:FULL
WFMPRE ENCDG:ASC
WFM WFID:A,ENC:BIN
```

Query:

Response:



Items that follow the waveform identification and coding specify other data packet parameters that refer to number of points, scaling, and error checking.

NR.PT — Specifies either 500 or 1000 points in the curve to follow.

PT.FMT:Y — Indicates all curve data is Y (display vertical) values. The data is ordered; each point's X (display horizontal) value is determined by its point number and parameters in the waveform preamble.

PT.OFF — Relates the first point to the X origin by the point offset.

XINCR — Is the difference between adjacent data points.

XZERO — Points to the X origin.

XUNIT — Identifies the horizontal display units in hertz, seconds, or divisions.

YOFF — Relates Y data to the Y origin by the Y offset.

YMULT — Scales the Y values.

`YZERO` — Points to the Y origin.

`YUNIT` — Identifies the units that apply to the Y values dBm, dBV, dBmV, dBμV, mV, μV, nV or volts.

`BN.FMT:RP` — Means each binary number (single byte) stands for a binary positive integer.

`BYT/NR:1` — Means that binary numbers or ASCII-coded digits are transferred as single bytes.

`BIT/NR:8` — Indicates the precision (max number of significant bits) of the binary numbers.

`CRV CHK:CHKSM0` — Specifies that the last byte of a binary transfer is a 2's complement, modulo-256 checksum for the preceding bytes (except for the first byte, which is a percent sign).

`BYTCHK:NULL` — Indicates no byte check is appended to binary data transfers.

X-Axis Scaling

X-axis specifications XINCR, PT.OFF, and XZERO are used to interpret the position of the ordered points as absolute X values.

*XN = XZERO + XINCR * (N−PT.OFF)*

where:

XN is the value in XUNITS on the X axis

XZERO is the center frequency, except in the following cases:

XZERO = 0 for time-domain data (ZEROSP)

XINCR is the absolute point-to-point distance on the X axis:

XINCR = (span/div)/100 for FULL in frequency domain
XINCR = (span/div)/50 for A or B in frequency domain
XINCR = TIME/100 for FULL in time domain
XINCR = TIME/50 for A or B in time domain

N is the point number (0, 1, 2, 3,..)

PT.OFF is graticule center for frequency-domain transfers and left graticule edge for time-domain transfers

PT.OFF = 250 for A or B in frequency domain
PT.OFF = 500 for FULL in frequency domain
PT.OFF = 0 in time-domain

Examples: For example, point 100 could have the following absolute values:

XN = 997 MHz for A or B with FREQ 1 GHz and SPAN 1 MHz
XN = 996 MHz for FULL with FREQ 1 GHz and SPAN 1 MHz
XN = 2 ms for FULL with SPAN 0 and TIME 2 ms

Y-Axis Scaling

Y-axis specifications YMULT, YZERO, and YOFF are used to interpret the data as the absolute value of the ordered data points.

YN = YZERO + YMULT (VALN−YOFF)*

where:

YN is the value in YUNITS of point number N

YZERO is the reference level in log vertical display mode and 0 in linear vertical display mode

YMULT is the scale factor divided by 25

VALN is the unscaled integer data at point N

YOFF is 225 (top edge of graticule) in log vertical display mode and 25 (bottom edge of graticule) in linear vertical display mode

Examples: For example, data value 125 (graticule center) could have the following absolute values:

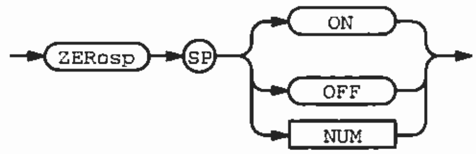
YN = −40 dBm at 10 dB/div with a reference level of 0 dBm

YN = 0.112 V in linear mode with a reference level of 0 dBm

The WFMPRE portion of the SET? response includes only the WFID and ENCDG arguments.

ZERosp (zero span mode)

Syntax:



Arguments: **ON** — The instrument is converted to a time domain mode with the frequency sweep defeated.

CRT readout shifts to the TIME mode on the horizontal axis instead of FREQ SPAN.

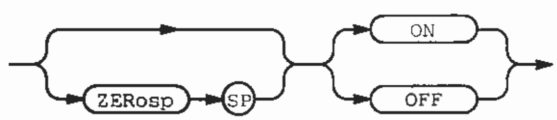
The previous FREQ SPAN is saved, and it is restored when ZEROSP is turned OFF.

OFF — ZEROSP is cancelled, leaving the FREQ SPAN at the value previously selected.

Interactions: Changing the SPAN setting turns ZEROSP OFF. PKCEN, MCEN, and CNTCF are not available in ZEROSP.

Query:

Response:



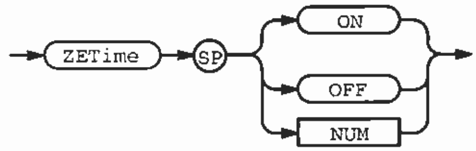
The response is the current zero span condition.

Macro Memory Used: 2 bytes.

Power-up Value: OFF.

ZETime (zero span)

Syntax:

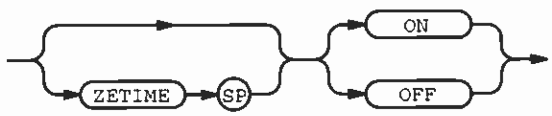


Arguments: **ON** — In zero span, the marker readout is amplitude and time.

OFF — In zero span, the marker readout is amplitude and frequency.

Query:

Response:



Interactions: The ZETIME command has no effect when the instrument is in a span setting other than zero span.

If the markers are on different traces, the secondary marker will move to the primary marker trace under either of the following conditions.

Under the same conditions, if the secondary marker is off-screen, it will move on-screen.

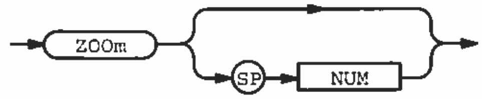
- If the ZETIME ON command is given with the primary marker on a zero span trace
- If the primary marker is moved to a zero-span trace while ZETIME is ON

Macro Memory Used: 2 bytes.

Power-up Value: OFF.

ZOOm

Syntax:



The ZOOM command moves the primary marker frequency to the center frequency and sets the SPAN to the next smaller span setting, if possible, in the front-panel 1–2–5 sequence.

Arguments: NUM — If the optional number argument is given, the span setting is reduced NUM times.

Numbers less than 1 are rounded to 1.

Execution warning message 111 is issued if the spectrum analyzer defaults to the lowest span/division because the span setting could not be reduced the requested number of times.

Interactions: Marker execution error message 121 is issued if the primary marker is not on an active trace.

If MARKER is OFF, ZOOM sets MARKER to SINGLE. In this case (since the marker initially appears at the center screen) the effect is to decrement the span only.

Macro Memory Used: 2 bytes.

Status and Events

The spectrum analyzer provides a status and event reporting system for the GPIB. This system informs you of certain significant events that occur within the spectrum analyzer. The primary sources of information are the status byte, interface messages, and error messages.

Status Byte

8	7	6	5	4	3	2	1	Decimal	Condition
0	1	0	X	0	0	0	1	65,81	Power on
0	X	0	X	0	0	1	0	2,18,66,82	End of Sweep
0	0	0	X	0	0	0	0	0,16	Ordinary operation
0	X	1	X	0	0	0	1	33,49,97,113	Command error
0	X	1	X	0	0	1	0	34,50,98,114	Execution error
0	X	1	X	0	0	1	1	35,51,99,115	Internal error
0	X	1	X	0	1	0	1	37,53,101,117	Execution error warning
0	X	1	X	0	1	1	0	38,54,102,118	Internal error warning

Four-bit status code

Spectrum analyzer busy condition

Abnormal (1)/normal (0) condition

SRQ is asserted (depends on RQS and EOS commands)

Power-on status	This is set when the instrument is turned on only if an internal switch is set; otherwise, SRQ is not asserted at power-up and power-on status does not exist. If selected by the switch, this status cannot be masked by the RQS command. The instrument is shipped with this switch off. Refer switch selection to qualified service personnel.
End-of-sweep status	This is set when the spectrum analyzer completes a sweep of the selected spectrum; it indicates that digital storage has been updated.

Ordinary operation status	This exists whenever there is no other status condition (nothing out of the ordinary) to report.
Command error	This occurs when a message cannot be analyzed or recognized.
Execution error	This results when a message is analyzed and is recognized, but cannot be executed; such as FREQ 999GHZ.
Internal error	This indicates that the spectrum analyzer has discovered a malfunction that could cause the instrument to operate incorrectly.
Execution warning	This results from a command that the spectrum analyzer has performed, but has a potential for error. An example is RESBW 10 KHZ in the maximum span mode.
Internal warning	This reports that a non-fatal operating condition has been detected.
Busy	This is reported whenever the spectrum analyzer acts on a message in its input buffer. This includes the WAIT command; while waiting, the spectrum analyzer reports busy status.

Effect of Busy on Device-Dependent Messages

The spectrum analyzer will not accept further device-dependent messages while the busy condition exists; if made a listener, it asserts NRFD. Commands that require interaction with the hardware can keep the spectrum analyzer busy for a second or more (significant to some bus controllers); for instance, commands such as DEGAUS and INIT in the COUNT mode.

The waveform processing commands can also require significant processor time. Of course, long messages such as the SET? response take a while to execute (see Execution Times, Table 4-1 in *Programming Examples*).

Although output operations such as the CURVE? response may take a long time to complete, the spectrum analyzer is busy only for the time it takes to load the output buffer.

Effect of Busy on Interface Messages

Interface messages and the GTL (Go To Local) message from the RESET TO LOCAL button are processed despite busy status. If RESET TO LOCAL interrupts a message, the spectrum analyzer tries to finish the rest of the message after local control is restored. At that time, commands that try to change a front-panel setting will result in error SRQs, because they conflict with local control.

The response of the spectrum analyzer to interface messages depends on how they are handled. Some interface messages are handled by the GPIB interface, while others require action by the spectrum analyzer. The latter generally involve the spectrum analyzer GPIB address and are part of the firmware rather than on the interface. The speed with which these commands can be handshaked depends on how fast the interrupt can be serviced; in most cases, a few hundred μs.

The following apply to IEEE 488 interface messages received by the spectrum analyzer.

1. Universal commands LLO, SPE, and SPD are handshaked and acted on by the interface, so they are unaffected by spectrum analyzer activity. The serial poll proceeds without delay if the talk address follows, since this function is handled by the interface.

2. UNL and UNT are handshaked by the interface, which immediately resets the talk or listen function, if active. Addresses that do not match are handshaked and discarded by the interface. When the current talk or listen address (MTA or MLA) is decoded by the interface, it holds up the handshake until the spectrum analyzer can get involved. The instrument will get involved as soon as it can service the interrupt. The CRT readout will be modified as soon as the programs are completed that update the addressed status.

   Because the spectrum analyzer gets involved when a current address is received, addressed commands are affected by the speed at which the service interrupts can be handled. Serial poll is similarly affected if MTA preceded SPE.

3. GTL is handshaked immediately by the interface. If the spectrum analyzer is already listen-addressed, the spectrum analyzer returns to local control (executes GTL) after executing any message in its buffer (except WAIT or message units following WAIT). REN unasserted is handled in the same manner as the GTL command.

4. DCL requires spectrum analyzer action that will hold up the handshake if the spectrum analyzer is busy. If the spectrum analyzer is listen-addressed, SDC is treated in the same manner. These two device-clear messages are executed as soon as they are received.

5. GET also requires spectrum analyzer action, so its handshake occurs only when the interrupt can be handled. GET is executed immediately, stopping the current sweep and rearming the sweep. If the spectrum analyzer is busy when GET is received, it will wait until the instrument is no longer busy to execute GET.

6. Parallel polls are handled by the spectrum analyzer, so PPC, PPE, PPD, and PPU must wait for the spectrum analyzer to service the interrupts before they can be executed. This assumes that the spectrum analyzer was addressed for the parallel poll sequence.

Busy and end-of-sweep are independent. Busy exists only while the spectrum analyzer is acting on a command, and end-of-sweep indicates that sweep and data-updating are complete. If a single-sweep command is sent, the spectrum analyzer remains busy only until it can start the sweep, while end-of-sweep does not occur until the operation is complete.

When polled, the spectrum analyzer reports a status code related to its SRQ, if any. Bit 5 always indicates the current condition. A serial poll clears the status byte that is reported. Since status is stacked, a new SRQ may be sent immediately.

Interface Messages

Table 3-1: Interface Messages

Title	Title
IF	Enabled
CPT	Command
GET (8)	Aborts and Then Rearms Sweep
GTL (1)	Go To Local control
IFC (IFC line)	Initializes Talker and Listener Functions
LLO (17)	Local Lockout
PPC (5)	Parallel Poll Configure
PPU (21)	Parallel Poll Unconfigure
SDC (4)	Same as DCL if Listener Addressed
SPD (25)	Serial Poll Disable
SPE (24)	Serial Poll Enable
TCT (9)	Take Control

Error and Event Codes

This section lists error and event codes. Table 3-2 lists event and error codes by type or group. Table 3-3 list ERR? responses numerically.

Table 3-2: ERR?/EVENT? Responses

Error Code	Event Code	Meaning
0	0	No error
Command Errors		
1	103	Illegal numeric format
4	109	END received in block binary
5	108	Block binary checksum error
6	103	Illegal placement of question mark
7	101	Query not recognized
8	101	Header not recognized
9	106	End of message unit not expected; arguments missing
10	103	Character argument not allowed
11	103	Numeric argument not allowed
12	103	String argument not allowed
13	103	Binary argument not allowed
14	103	Link not allowed for this argument
20	103	Special argument type not recognized
21	103	Special argument not allowed
22	103	Character argument not recognized
24	150	Input buffer overflow

Table 3-2: ERR?/EVENT? Responses (Cont.)

Error Code	Event Code	Meaning
Execution Errors		
26	250	Output buffer overflow; remaining output lost
27	201	Attempt to execute command in Local mode
28	205	Frequency out of range (FREQ, TUNE, FIRST, SECOND, MMAX, MMIN, MTUNE, MFREQ, STSTOP, STEP)
29	205	FRQRNG out of range
30	205	CRES out of range
31[1]	205	SPAN out of range
32[1]	205	RESBW out of range
33[1]	205	MAXPWR or MINATT out of range
34[1]	205	Level out of range (REFLVL, THRHLD, BWNUM, MVRTDB, MVLFDB)
35[1]	205	VRTDSP LIN out of range
36[1]	205	VRTDSP LOG out of range
37[1]	205	TIME out of range
39[1]	204	IDENTify not allowed in this span/div
40[1]	204	Signal finds not allowed in zero span
41	204	Invalid DATA or ADDR argument contents
42	204	DATA direction not compatible with ADDR direction
45	206	GET (Group Execute Trigger) ignored (not executed)
46[1]	205	NUMEV out of range
47[1]	205	STORE, RECALL, DSTORE, DRECAL, or RDATA out of range
48	204	PHSLK cannot be turned OFF/ON directly with PHSLK command
101[1]	204	Function not available when SGTRAK is on
102[1]	205	Frequency range limited in 75 Ω input — Option 07 only
103[1]	205	Frequency out of range after step
104[1]	204	Bandwidth mode is not available when in linear
105[1]	205	Illegal sweep range
106[1]	205	Argument out of range
109[1]	205	ROFSET out of range
110	560	STEP size out of range — set to maximum
111	561	SPAN defaulted to minimum span
210	204	Peaking not available in this band
Execution Warnings		
49	550	FREQ change caused EXMXR change
50	551	SPAN defaulted to MAX
52	553	UNCAL light turned on
53	555	Multiple use of display buffer

Table 3-2: ERR?/EVENT? Responses (Cont.)

Error Code	Event Code	Meaning
Internal Errors		
54	554	UNCAL light turned off
57	350	Tuning DAC carry operation failed
58	351	Failed to lock 1st LO
59	352	Lost 1st LO lock
60	353	Recentering failure on unlocking of 1st LO
61	354	Calibration failure
62	355	Battery–operated RAM checksum error
72	382	1st LO tuning system failed
74	386	2nd LO tuning system failed
75	388	Phase lock system failed
78	394	IF count failed
79	396	Power supply out of regulation
80	398	Frequency reference unlocked
82	383	1st LO tuning system recovered from a failure
84	387	2nd LO tuning system recovered from a failure
85	389	Phase lock system recovered from a failure
88	395	IF count recovered from a failure
89	397	Power supply regained regulation
90	399	Frequency reference re–locked
99	302	Unrecognized event occurred
System Events		
97	401	Power just came on
98	402	Operation complete (end of sweep)
100	403	SRQ was requested
Marker Execution Errors		
107[1]	204	MKTIME not available unless in zero span mode
108[1]	205	MKTIME out of range
120[1]	205	Frequency out of range because marker is on inactive trace
121[1]	204	Function not available when marker is on an inactive trace
122[1]	204	Function not available when marker is off
123[1]	204	Function not available when delta marker is off
124[1]	204	Function not available in BWMODE
125[1]	204	Function not available when marker is on a B–SAVEA trace
126[1]	204	Function not available when in dB/Hz
Marker Execution Warning		
130	558	Signal find commands could not find signal (MLFTNX, MRGTNX, MFBIG, HRAMPL, LRAMPL, PKFIND, MVRTDB, MVLFDB)

Table 3-2: ERR?/EVENT? Responses (Cont.)

Error Code	Event Code	Meaning
Macro Execution Errors		
160	204	Command not available when entering a macro
161	204	Command only available when entering a macro
163	204	Memory full — macro entry refused
164	204	Number is already used
165	204	No macro is stored at NUM
166[1]	204	No more MDATA commands to read
167[1]	204	Too many GOSUBs with no RETURN
168[1]	204	RETURN command not expected
169	204	Line number used more than once (DSLINE command)
170	204	Missing LABEL
171	204	Missing ENDI
172	204	Missing NEXT
173	204	ELSE command not expected
174	204	ENDI command not expected
175	204	NEXT command not expected
176[1]	204	Index out of range (DISBUF:INDEX or VAR:INDEX)
177	204	FOR index is already used
178	204	Missing DONE or RETURN or GOTO
Multiband Execution Errors		
140	204	Function either not available or cannot be changed when in multiband sweep
141	205	Frequency out of range because instrument is in multiband sweep
143	204	Multiband sweep cannot be started in external or manual sweep
Multiband Execution Warnings		
150	556	Multiband sweep started
151	557	Multiband sweep stopped
152	559	Start frequency changed to 1.7 GHz because of using internal mixer

1 If this error message is issued, the macro in progress will be aborted.

Table 3-3: ERR? Responses In Numerical Order

Error Code	Meaning
0	No error
1	Illegal numeric format
4	END received in block binary
5	Block binary checksum error
6	Illegal placement of question mark
7	Query not recognized
8	Header not recognized
9	End of message unit not expected; arguments missing
10	Character argument not allowed
11	Numeric argument not allowed
12	String argument not allowed
13	Binary argument not allowed
14	Link not allowed for this argument
20	Special argument type not recognized
21	Special argument not allowed
22	Character argument not recognized
24	Input buffer overflow
26	Output buffer overflow; remaining output lost
27	Attempt to execute command in Local mode
28[1]	Frequency out of range (FREQ, TUNE, FIRST, SECOND, MMAX, MMIN, MTUNE, MFREQ, STSTOP, STEP)
29	FRQRNG out of range
30	CRES out of range
31[1]	SPAN out of range
32[1]	RESBW out of range
33[1]	MAXPWR or MINATT out of range
34[1]	Level out of range (REFLVL, THRHLD, BWNUM, MVRTDB, MVLFDB)
35[1]	VRTDSP LIN out of range
36[1]	VRTDSP LOG out of range
37[1]	TIME out of range
39[1]	IDENTify not allowed in this span/div
40[1]	Signal finds not allowed in zero span
41	Invalid DATA or ADDR argument contents
42	DATA direction not compatible with ADDR direction
45	GET (Group Execute Trigger) ignored (not executed)
46[1]	NUMEV out of range
47[1]	STORE, RECALL, DSTORE, DRECAL, or RDATA out of range
49	FREQ change caused EXMXR change
50	SPAN defaulted to MAX
52	UNCAL light turned on
53	Multiple use of display buffer
54	UNCAL light turned off
57	Tuning DAC carry operation failed
58	Failed to lock 1st LO
59	Lost 1st LO lock

Table 3-3: ERR? Responses In Numerical Order (Cont.)

Error Code	Meaning
60	Recentering failure on unlocking of 1st LO
61	Calibration failure
62	Battery–operated RAM checksum error
72	1st LO tuning system failed
74	2nd LO tuning system failed
75	Phase lock system failed
78	IF count failed
79	Power supply out of regulation
80	Frequency reference unlocked
82	1st LO tuning system recovered from a failure
84	2nd LO tuning system recovered from a failure
85	Phase lock system recovered from a failure
88	IF count recovered from a failure
89	Power supply regained regulation
90	Frequency reference re–locked
97	Power just came on
98	Operation complete (end of sweep)
99	Unrecognized event occurred
100	SRQ was requested
101[1]	Function not available when SGTRAK is on
102[1]	Frequency range limited in 75 Ω input — Option 07 only
103[1]	Frequency out of range after step
104[1]	Bandwidth mode is not available when in linear
105[1]	Illegal sweep range
106[1]	Argument out of range
107[1]	MKTIME not available unless in zero span mode
108[1]	MKTIME out of range
109[1]	ROFSET out of range
110	STEP size out of range — set to maximum
111	SPAN defaulted to minimum span
120[1]	Frequency out of range because marker is on inactive trace
121[1]	Function not available when marker is on an inactive trace
122[1]	Function not available when marker is off
123[1]	Function not available when delta marker is off
124[1]	Function not available in BWMODE
125[1]	Function not available when marker is on a B–SAVEA trace
126[1]	Function not available when in dB/Hz
130	Signal find commands could not find signal (MLFTNX, MRGTNX, MFBIG, HRAMPL, LRAMPL, PKFIND, MVRTDB, MVLFDB)
140	Function either not available or cannot be changed when in multiband sweep
141	Frequency out of range because instrument is in multiband sweep
143	Multiband sweep cannot be started in external or manual sweep

Table 3-3: ERR? Responses In Numerical Order (Cont.)

Error Code	Meaning
150	Multiband sweep started
151	Multiband sweep stopped
152	Start frequency changed to 1.7 GHz because of using internal mixer
160	Command not available when entering a macro
161	Command not available when NOT entering a macro
162	Command only available when a macro is stopped
163	Memory full — macro entry refused
164	Number is already used
165	No macro is stored at NUM
166[1]	No more MDATA commands to read
167[1]	Too many GOSUBs with no RETURN
168[1]	RETURN command not expected
169	Line number used more than once (DSLINE command)
170	Missing LABEL
171	Missing ENDI
172	Missing NEXT
173	ELSE command not expected
174	ENDI command not expected
175	NEXT command not expected
176[1]	Index out of range (DISBUF:INDEX VAR:INDEX, or STNUM INDEX)
177	FOR index is already used
178	Missing DONE or RETURN or GOTO
210	Peaking not available in this band

1 If this error message is issued, the macro in progress will be aborted.

Programming Examples

This section covers some techniques for programming the spectrum analyzer. Spectrum analyzer programming can include developing macros as well as controller-based programs. Familiarity with the information in this section will speed your progress in putting the spectrum analyzer to work solving your measurement problems.

Notes on the Program Examples

Most examples in this appendix are based on IBM PCs or compatibles running MS-DOS, MS Quick Basic, and the National Instruments GPIB driver. Refer as necessary to the manual for your specific controller, and supplementary programs, as necessary.

Most of the programming examples show the full command names, all in upper case. Note that for all command names and most command arguments you can shorten the command or argument to three characters. Regardless of whether you use the long or short form of the commands, the commands may be entered using either upper or lower case characters.

Application Tips

In addition to programming examples, this section also provides some tips and advice regarding successful use of the spectrum analyzer programming commands. These programming considerations address both spectrum analyzer functions and controller-spectrum analyzer interaction over the GPIB.

Signal Processing

The instrument should be in the single sweep mode and not be sweeping during any of the signal processing commands or queries. Signal processing commands are the marker and waveform commands.

Right Way — In order to satisfactorily center a given signal, be sure to incorporate the SIGSWP command in your program, as in the following example.

```
INIT;FREQ 100MHZ;SPAN 1MHZ;SIGSWP;SIGSWP;WAIT;MFBIG;MLOCAT?
```

In this example, the first SIGSWP puts the spectrum analyzer into the single sweep mode. The second SIGSWP starts a sweep, and the WAIT tells the spectrum analyzer to wait for the sweep to end before acting on the MFBIG;MLOCAT? commands.

We recommend that the spectrum analyzer be in single sweep mode (SIGSWP) and that either SIGSWP;WAIT or SWEEP be used to take another sweep to make sure that the waveform reflects the desired setting before making measurements.

Wrong Ways — The following is an unsuccessful command line to try to center a signal:

```
INIT;FREQ 100MHZ;SPAN 1MHZ;MFBIG;MLOCAT?
```

In this example, the spectrum analyzer will not have made a full sweep at 100 MHz frequency and a span of 1 MHz before the MFBIG;MLOCAT? is started because there is no WAIT included in the command. Therefore, you do not know where the sweep was when the analyzer went to 100 MHz. The following command will also produce an error:

```
INIT;FREQ 100MHZ;SPAN 1MHZ;WAIT;MFBIG;MLOCAT?
```

In this example, the spectrum analyzer will most likely be part way through the sweep when the FREQ and SPAN commands are received. Although the WAIT will cause the spectrum analyzer to wait for the end of the sweep, only the last part of the waveform will correspond to the specified FREQ and SPAN settings, and the subsequent MFBIG may find an incorrect maximum point in the (old, un-updated) left part of the screen.

Running Programs Without a Controller (Macros)

Most of the programming examples in this section can be incorporated into macros designed to meet your particular needs. The examples in this section that contain commands that cannot be used within macros are noted.

Prepare your macros using a language such as BASIC and a controller. Once the macro is stored in the spectrum analyzer memory, it can be run at any time without the further use of a controller. We recommend that a copy of all macros be kept for ease in reconstructing a macro if it is lost. Any stored macro will be lost if the battery powered memory is interrupted (as when the battery is removed for long term storage).

Data Acquisition

When the spectrum analyzer is executing commands under program control, there are two programs running, not just one. One program is running in the controller and a second in the spectrum analyzer. The key to success is synchronizing the execution of the two programs.

In addition, the two programs must be synchronized with the data acquisition event; in this case, the sweep.

Synchronizing Controller and Spectrum Analyzer

Programs must run in the controller at the same time that the spectrum analyzer acts on messages that come over the GPIB. Summarization is done within the spectrum analyzer by the way it buffers and executes messages.

When the spectrum analyzer receives a message, it waits until the end of the message to begin acting on it. While busy acting on the message, the spectrum analyzer does not accept any other device-dependent messages. When it is finished with the message, the spectrum analyzer is ready to handshake another message, which it then acts on, and so forth. You can depend on the spectrum analyzer to assert the NRFD handshake line on the GPIB while it is busy; this prevents a controller GPIB output statement that would send further instructions to the spectrum analyzer.

For example, enter

```
100 FOR I=1 TO 10
105 CMD$="FREQ "+STR$(I)+"GHZ"
110 CALL IBWRT (PORT1%, CMD$)
120 NEXT I
```

Watch the spectrum analyzer FREQUENCY readout change while this loop is executing. You can see that the controller executes the loop more slowly than it would if line 110 only printed what is in quotes on the controller CRT. What is making the controller step through the loop at a more deliberate pace? It must wait at line 110 for the spectrum analyzer to execute the previous FREQ command.

A controller GPIB input statement can also be used to synchronize the controller and the spectrum analyzer. The controller could make a table of frequency ranges for the frequencies covered by the previous loop, filling the table only after the FREQ command is executed.

```
100 DIM F(10,2)
110 FOR I=1 TO 10
115 CMD$="FREQ "+STR$(I)+"GHZ;FRQRNG?
120 CALL IBWRT (PORT1%,CMD$)
125 RSP$=SPACE$(80)
130 CALL IBRD (PORT1%,RSP$)
135 F(I,1)=VAL(RSP$)
140 F(I,2)=I*1E+9
150 NEXT I
```

Line 130	Addresses the spectrum analyzer to talk; however, the spectrum analyzer does not begin talking until it finishes executing the message in line 120. This ensures that the spectrum analyzer updates the FRQRNG query response before handshaking out the response in line 130.

Synchronizing with the Sweep

Spectrum data can be acquired synchronously with the sweep that updates digital storage if a WAIT command is inserted in the message to the spectrum analyzer. Generally, WAIT is placed immediately after a SIGSWP command that arms a sweep so that data is acquired from a full sweep. WAIT delays the execution of commands or queries that follow in the same message until the full sweep is completed. This means you can direct the spectrum analyzer to acquire, process, and output data, all in the same message. If the commands or queries you add to process or output data follow WAIT, the results will be based on data acquired by the SIGSWP command.

For example, enter

```
100 DIM P(5,2)
105 CMD$="SIGSWP"
110 CALL IBWRT (PORT1%,CMD$)
120 FOR I=1 TO 5
125 CMD$="FREQ "+STR$(I)+"GHZ;SIGSWP;WAIT;FMAX;POINT?"
130 CALL IBWRT (PORT1%,CMD$)
135 RSP$=SPACE$(80)
140 CALL IBRD (PORT1%,RSP$)
142 P(I,1)=VAL(RSP$)
144 P(I,2)=VAL(MID$(RSP$,1+INSTR(RSP$,",")))
150 NEXT I
```

Lines 105, 110	Set the spectrum analyzer to the single sweep mode (if the spectrum analyzer is not already in the single-sweep mode). Succeeding SIGSWP commands arm the sweep.
Lines 123, 130	Illustrate how to use WAIT. WAIT follows SIGSWP and precedes the command and query that ready the spectrum analyzer to output the updated data. The spectrum analyzer does not handshake out the data in line 140 until it finishes executing the message in line 130.

Figure 4-1 further illustrates the two-program concept (one in the controller and one in the spectrum analyzer) and how they are synchronized with the sweep for data acquisition. This figure charts the execution of the two programs; arrows between the programs relate how one waits for the other. The WAIT command is executed in the loop that tests for the end of sweep; this synchronizes data acquisition with the sweep.

Using the End-of-Sweep SRQ

Although the previous method for synchronizing controller/spectrum analyzer operation with the sweep is recommended, there is another method. This alternative may be necessary with some operating systems or application programs that allow a short response time when the spectrum analyzer is made a talker or that must take care of other programming tasks while the spectrum analyzer is acquiring data.

In such cases, the spectrum analyzer can be set up to generate an SRQ at the end of each sweep by sending EOS ON to it. An SRQ and event message 402 will be generated at the end of each sweep. See Section 2 or your controller manual for details on how to set up an SRQ handler.

INPUT: An SRQ Alternative

An INPUT or READ statement in the right place is an alternative to waiting for an end-of-sweep SRQ. This tactic takes advantage of a spectrum analyzer output feature; if the spectrum analyzer has no output when it receives its talk address, it outputs a byte with all bits set to one (as soon as it is not busy).

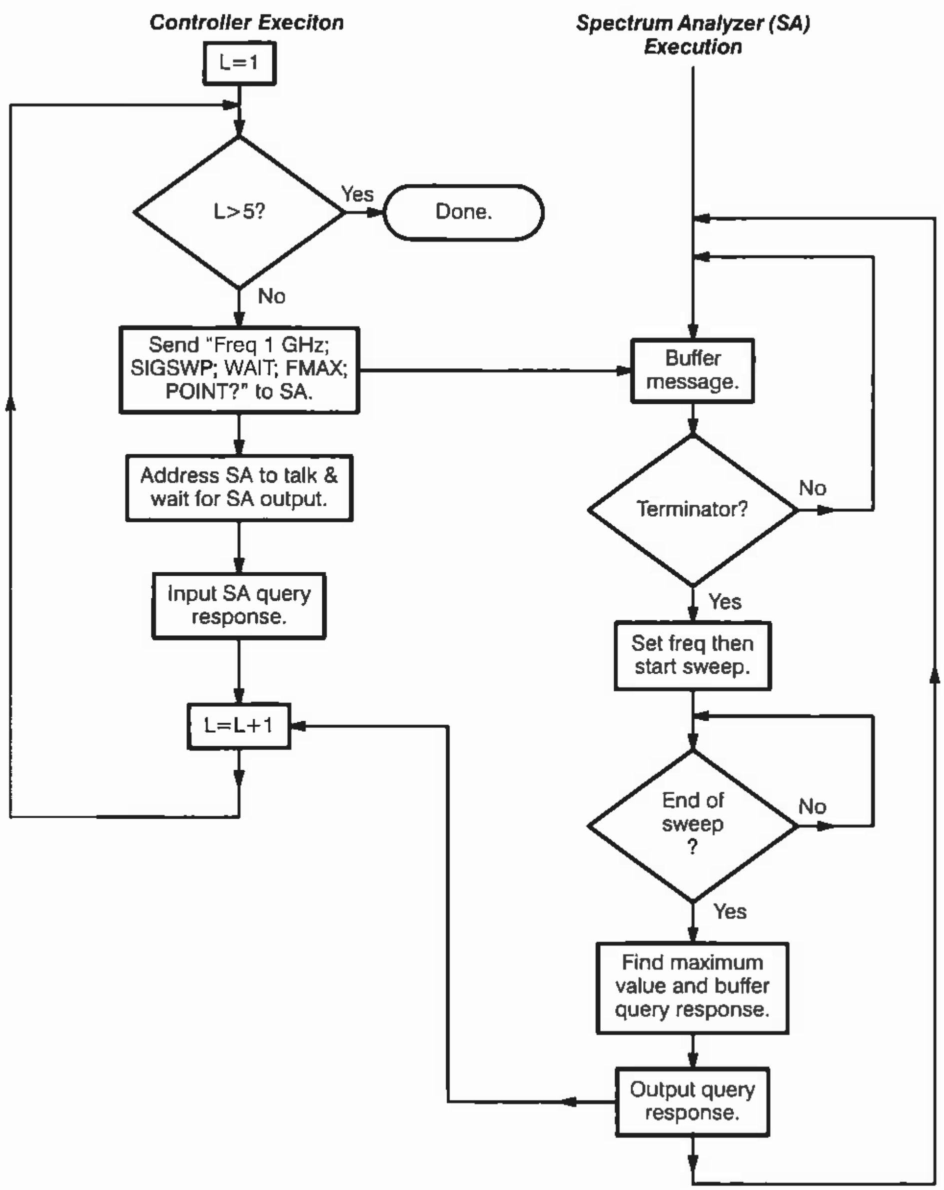


Figure 4-1: Synchronizing Controller and spectrum Analyzer for Data Acquisition

```
100 DIM P(5,2)
105 CMD$="SIGSWP"
110 CALL IBWRT(PORT1%, CMD$)
120 FOR I=1 TO 5
125 CMD$="FREQ "+STR$(I)+"GHZ;SIGSWP;WAIT"
130 CALL IBWRT(PORT1%,CMD$)
135 RSP$=SPACE$(80)
140 CALL IBRD(PORT1%,RSP$)
145 CMD$="FMAX;POI?"
150 CALL IBWRT(PORT1%,CMD$)
155 RSP$=SPACE$(80)
160 CALL IBRD(PORT1%,RSP$)
162 P(I,1)=VAL(RSP$)
164 P(I,2)=VAL(MID$(RSP$,1+INSTR(RSP$,",")))
170 NEXT I
180 END
```

Here the WAIT is put back into the spectrum analyzer message and the IBRD statement in line 140 stalls the controller while the spectrum analyzer makes a sweep.

Binary Waveform Transfer

Selecting binary, rather than ASCII coded decimal, speeds up waveform transfers. Neither the controller nor the spectrum analyzer has to perform a conversion between binary and ASCII. The difference is evident in the times for both kinds of transfer, listed in this section under Execution Times, Table 4-1. The following examples illustrates this effect.

Getting Spectrum Analyzer Binary CURVE Output

The spectrum analyzer encloses binary waveform data values in the binary block format. For details, see the syntax diagrams in Sections 4 and 7.

For a controller routine that handles block binary, enter the following.

```
500 REM GET SPECTRUM ANALYZER CURVE OUTPUT
510 DIM W%(1000),RDD%(1000)(declare integer array)
515 CMD$="WFMPRE ENC:BIN"
520 CALL IBWRT(PORT1%,CMD$)
525 CMD$="CURVE?"
530 CALL IBWRT(PORT1%,CMD$)
540 EOSV%=ASC("%")+&H400(set EOS to %[begin binary data])
550 CALL IBEOS(PORT1%,EOSV%)
560 RSP$=SPACE$(255)
570 CALL IBRD(PORT1%,RSP$)
580 CALL IBFIND (DEVICE$,PORT1%)(reset EOS)
600 RSP$=SPACE$(2)
610 CALL IBRD(PORT1%,RSP$)(read, byte count)
620 COUNT%=ASC(RSP$)*256+ASC(RIGHT$(RSP$,1))1
(convert to integer)
630 CNT%=COUNT%+5
```

```
640 CALL IBRDI(PORT1%,RDD%(1),CNT%)
(read waveform)
650 LGTH%=0:MODE%=8
660 CALL DEBLK(RDD%(1),W%(1),COUNT%,MODE%,LGTH%)
(convert array from binary block to integer)
```

Line 510	Declare an integer array large enough for a full 1000 points.
Lines 515, 520	Request the spectrum analyzer to format data in a binary format.
Lines 525, 530	Send the CURVE? query command, which gets the waveform data.
Lines 540–570	Read header up through % sign, then ignore it.
Lines 600-640	Read the byte count and convert it to integer format.
Lines 630, 640	Read the actual binary waveform data into array RDD%.
Lines 650, 660	Convert the binary waveform data in array RDD% into integer waveform data, and write the result into array W%.

NOTE

For more information on the DEBLK and ENBLK commands, refer to the applicable documentation from National Instruments or Tektronix.

Sending a Binary CURVE to the Spectrum Analyzer

The following routine employs end block format to transfer a waveform to the spectrum analyzer. Array W is transferred; if not already created by the preceding routine, W should be dimensioned to 1000 and filled with data in the range 0 to 255.

```
600 REM TRANSMIT AS BINARY CURVE TO THE SPECTRUM ANALYZER
605 V%=0
610 CALL IBEOT(PORT1%,V%)(disable EOI)
615 CMD$=CHR$(64)
620 CALL IBWRT(PORT1%,CMD$)
(send @ character)
625 CNT%=1000:MODE%=8:LGTH%=1000
630 CALL ENBLK(W%(1),RDD%(1),LGTH%,MODE%,(CNT%) (convert array from
integer to binary block)
640 V%=1
```

```
645 CALL IBEOT(PORT1%,V%) (enable EOI)
650 CALL IBWRTI(PORT1%,RDD%(1),(CNT%)
(send binary data)
```

Lines 615, 620	Send the block specifier @.
Lines 625, 630	Convert the integer waveform data in array W% into binary data and write the result into array RDD%.
Lines 645, 650	Write the binary data to the screen of the spectrum analyzer.

NOTE

The CURVE header has been omitted in this example, but would be accepted if sent.

Getting and Sending Binary Data with RDATA? and RDATA

The following routine illustrates the transfer of settings directly to and from the battery-backed-up memory. (A similar program may be used with displays.) The routine reads the settings from memory 1 and sends them to memory 2.

CAUTION

This program will overwrite any settings stored in memory locations 1 or 2.

```
110 DIM W%(130),RDD%(130)
115 CMD$="STORE 1"
120 CALL IBWRT(PORT1%,CMD$)
125 CMD$="RDATA? FPSET:1"
130 CALL IBWRT(PORT1%,CMD$)
135 EOSV%=ASC("%")+&H400
(set EOS to %[begin binary data])
140 CALL IBEOS(PORT1%,EOSV%)
145 RSP$=SPACE$(255)
150 CALL IBRD(PORT1%,RSP$)
160 CALL IBFIND(DEVICE$,PORT1%) (reset EOI)
175 RSP$=SPACE$(2)
180 CALL IBRD(PORT1%,RSP$)
190 COUNT%=ASC(RSP$)*256+ASC(RIGHT$(RSP$,1))-1
200 CNT%=COUNT%+5
210 CALL IBRDI(PORT1%,RDD%(1),CNT%)
215 LGTH%=0:MODE%=8
```

```
220 CALL DEBLK(RDD%(1),W%(1),COUNT%,MODE%,LGTH%) (unpack data)
225 V%=0
230 CALL IBEOT(PORT1%,V%) (disable END)
235 CMD$="RDATA FPSET:2:%"
240 CALL IBWRT(PORT1%,CMD$)
245 V%=1
250 CALL IBEOT(PORT1%,V%)
260 CALL ENBLK(W%(1),RDD%(1),LGTH%,MODE%,COUNT%)
270 CALL IBWRTI(PORT1%,RDD%(1),CNT%)
```

Line 110	Dimensions an integer array to hold the settings data. This must be an integer array.
Lines 115, 120	Places present instrument settings in memory 1 to insure there is something to transfer.
Lines 235, 240	Sends the command header to the instrument. The preceding IBEOT stops the controller from asserting EOI, so the header and the data that follows will be received by the spectrum analyzer as a single message.
Line 270	Sends the settings data to the instrument. The percent sign (%) and byte count preceding the data and the checksum following the data are sent by the controller.

Scaling, Saving and Graphing Waveform Data

The spectrum analyzer waveform data consists of numbers from 0 to 255 (called screen units). These numbers can be scaled to electrical units using data contained in the WFMPRE? response.

Here is an expanded version of the previous spectrum analyzer binary output program. This version transfers whatever portion of memory you have previously specified with a WFMPRE command: A, B, or FULL (power-up default is FULL). The program scales both the X and Y values and stores them in a two-wide array. The program also saves the non-scaled binary array so you can transfer it back into spectrum analyzer digital storage if you wish.

```
100 REM GET AND SCALE SPECTRUM ANALYZER BINARY CURVE OUTPUT
110 ERASE W%,M
115 CMD$="WFMPRE?"
120 CALL IBWRT(PORT1%,CMD$)
125 RSP$=SPACE$(255)
130 CALL IBRD(PORT1%,RSP$)
131 N=VAL(MID$(RSP$,INSTR(RSP$,"NR.PT:")+6))
132 X3=VAL(MID$(RSP$,INSTR(RSP$,"PT.OFF:")+7))
133 X2=VAL(MID$(RSP$,INSTR(RSP$,"XINCR:")+6))
134 X1=VAL(MID$(RSP$,INSTR(RSP$,"XZERO:")+6))
135 Y3=VAL(MID$(RSP$,INSTR(RSP$,"YOFF:")+5))
```

```
136 Y2=VAL(MID$(RSP$,INSTR(RSP$,"YMULT:")+6))
137 Y1=VAL(MID$(RSP$,INSTR(RSP$,"YZERO:")+6))
150 DIM M(N,2),W%(N),RDD%(N)
170 GOSUB 230
180 FOR I=1 TO N
190 M(I,1)=X1+X2*(I-X3)
200 M(I,2)=Y1+Y2*(W%(I)-Y3)
210 NEXT I
220 STOP
230 CMD$="WFM ENC:BIN;CURVE?":CALL IBWRT(PORT1%,CMD$)
240 EOSV%=ASC("%")+&H400
250 CALL IBEOS(PORT1%,EOSV%) (set EOS to %[begin binary data])
260 RSP$=SPACE$(255)
270 CALL IBRD(PORT1%,RSP$)
280 CALL IBFIND(DEVICE$,PORT1%)(reset EOS)
300 RSP$=SPACE$(2)
310 CALL IBRD(PORT1%,RSP$)
320 COUNT%=ASC(RSP$)*256+ASC(RIGHT$(RSP$,1))-1
330 CNT%=COUNT%+5
340 CALL IBRDI(PORT1%,RDD%(1),CNT%)
350 LGTH%=0:MODE%=8
360 CALL DEBLK(RDD%(1),W%(1),COUNT%,MODE%,LGTH%) (unpack waveform)
370 RETURN
```

Line 110	Clears the waveform arrays.
Lines 115, 120	Requests the waveform preamble and a binary curve.
Lines 125–137	Inputs the spectrum analyzer WFMPRE response, storing the first seven numbers it finds as variables N, X3, X2, etc.
Lines 150 and 170	Calls subroutine to put the waveform in W.
Lines 180–220	Scales the waveform integers and fills array M with the result. The first number in each element of the array is a frequency, and the second number is the power detected at that frequency. The elements can be printed on the screen with the statement PRINT M, or any element I can be printed with the statement PRINT M(I).
Lines 230–370	Requests a curve. The byte count words and checksum are input and checked automatically. Checksum failure results in an error interrupt to the controller that can be trapped.

Saving the Scaled Array

These statements write data to disk.

```
100 OPEN "WAVEFORM" FOR OUTPUT AS #2
110 WRITE #2, M
120 CLOSE #2
```

These statements return the data from the disk.

```
100 OPEN "WAVEFORM" FOR INPUT AS #2
110 INPUT #2,M
120 CLOSE #2
```

Storing Settings

Instrument settings can be saved in the instrument's non-volatile memory using the STORE command and restored using the RECALL command. Waveforms can also be stored and recalled using DSTORE and DRECAL, respectively.

Comparing FREQ and TUNE

The FREQ command argument is an absolute frequency, while the TUNE command argument is a relative frequency. Use FREQ to change the center frequency to some value you supply as a command argument. Use TUNE to change the center frequency by some offset you supply as a command argument.

The ranges of the FREQ and TUNE commands differ in several respects. The range of the FREQ command covers the entire range of the instrument. The TUNE command is affected by the current frequency setting except in Max Span.

Using PLOT

The spectrum analyzer can generate a plot of the display directly to one of the following plotters.

- Tektronix HC100
- Hewlett-Packard HP7475A

Connect the plotter to the spectrum analyzer GPIB port. The PLOT? query is used to get the plot commands from the spectrum analyzer. These can then be sent to the plotter.

Multiple Use of Display Buffer

An error message alerts you to possibly invalid data caused by multiple use of the display buffer; that is, using the buffer for more than one purpose during execution of a message. Also, you are informed in this manual of possible interaction involving waveform processing and waveform data I/O executed in the same spectrum analyzer message. (Refer to "Waveform Data and Command Interactions" on page 2-20).

There is no conflict in many cases because the spectrum analyzer buffers the message you send and then executes it in the order you sent it.

For example, you can use the spectrum analyzer as a waveform processor for spectrum data you previously acquired in array A by entering the following program.

```
100 REM BUFFER DEMO
105 CMD$="SIGSWP"
110 CALL IBWRT(PORT1%,CMD$)
115 V%=0
120 CALL IBEOT(PORT1%,V%)
125 CMD$="CURVE"
130 CALL IBWRT(PORT1%,CMD$)
140 FOR I=1 TO 1000
145 CMD$=" "+STR$(W%(I))
150 CALL IBWRT(PORT1%,CMD$)
160 NEXT I
165 V%=1
170 CALL IBEOT (PORT1%,V%)
175 CMD$=";FIBIG;POINT?"
180 CALL IBWRT(PORT1%,CMD$)
185 RSP$=SPACE$(80)
190 CALL IBRD(PORT1%,RSP$)
200 B1=VAL(RSP$)
210 B2=VAL(MID$(RSP$,1+INSTR(RSP$,",")))
```

In this case, the spectrum analyzer does what you ask; it loads a waveform into digital storage and returns the point at the peak of the largest signal. SIGSWP is included to keep the spectrum analyzer active trace from overwriting the stored trace written by the CURVE command.

Interaction is possible in other cases, however, because there is only one display data buffer used for both display input and output and as workspace for waveform processing. For instance, conflicts can arise if the CURVE command or the CURVE query are used in the same command line with another CURVE command or CURVE query.

Whether interaction results in invalid data depends on the relative position of these message units in the message. This follows from how these message units use the buffer.

Buffer Data Flow

Data flow through the buffer is diagrammed in Figure 4-2. This figure identifies the kinds of data operations as data paths or destinations branching from the right of the buffer. The partitions in digital storage memory are shown as data sources or paths branching from the left of the buffer.

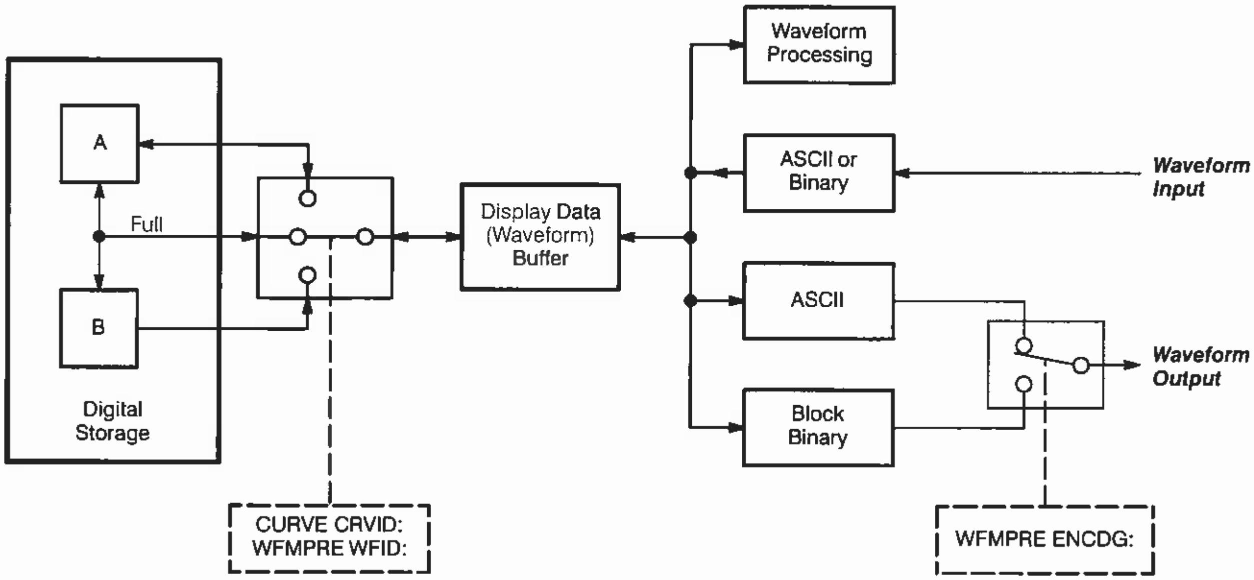


Figure 4-2: Buffer Flow

The WFMPRE and CURVE commands contain arguments that set switches to control data flow through the buffer. Either the CRVID argument or the WFID argument sets the switch to select A, B, or FULL (A and B) memory. The ENCDG argument sets the switch that selects either ASCII or block binary waveform output. Both switches are shown in their power-up default positions. They remain wherever they are set until changed by an appropriate command.

Order-Dependent Conflicts

Conflicts in the use of the buffer occur depending on the order in which waveform processing and I/O occurs. The CURVE command transfers the data to digital storage while executing, and the display data point commands act on the data while executing. The CURVE query, by contrast, does not transfer the data until after the entire message is executed (and the spectrum analyzer receives its talk address). Thus, if these message units are mixed in a message, the contents of the buffer may be changed between when it is loaded and when it is acted on or transferred.

```
100 REM WAVEFORM PROCESSING AND I/O
110 DIM A%(1000),B%(1000)
115 CMD$="CURVE?"
```

```
120 CALL IBWRT(PORT1%,CMD$)
125 EOSV%=ASC(",")+&H400(set EOS to comma)
130 CALL IBEOS(PORT1%,EOSV%)
135 RSP$=SPACE$(255)
140 CALL IBRD(PORT1%,RSP$) (read header and CRVID)
150 FOR I=1 TO 999
155 RSP$=SPACE$(8)
160 CALL IBRD(PORT1%,RSP$) (read one point)
170 W%(I)=VAL(RSP$)
180 NEXT I
190 CALL IBFIND(DEVICE$,PORT1%) (next EOS)
205 RSP$=SPACE$(8)
210 CALL IBRD(PORT1%,RSP$) (read last point)
220 W%(1000)=VAL(RSP$)
225 V%=0
230 CALL IBEOT(PORT1%,V%) (disable EOI)
235 CMD$="CURVE "
240 CALL IBWRT(PORT1%,CMD$)(start transfer to spectrum analyzer)
250 FOR I=1 TO 999
255 CMD$=STR$(W%(I))+" "
260 CALL IBWRT(PORT1%,CMD$) (send one point)
270 NEXT I
275 V%=1
280 CALL IBEOT(PORT1%,V%) (enable EOI)
285 CMD$=STR$(W%(1000))
290 CALL IBWRT(PORT1%,CMD$)(send last point)
```

Line 115, 120	Requests a curve, which the spectrum analyzer buffers.
Lines 125-220	Inputs the curve before it is overwritten by lines 225–290.

Finding Signals Using Markers

The marker system in the spectrum analyzer allows many waveform parameters to be found without transferring a waveform to the controller. You will get better results if you understand how the routines work and what their limitations are. This portion of the manual will help you gain that understanding and use the power of the spectrum analyzer in your application with more accurate and predictable results.

Understanding How The Marker System Works

The marker commands allow, through the STYPE command, a choice of peak search algorithms. These commands identify a candidate signal by means of three points; a peak and a point 3 dB down on each side of the peak. Whether or not the candidate is recognized as a signal depends upon the processing mode chosen. When SPURS is chosen, all candidates are taken to be signals. When CW is chosen, a signal (to be a signal) must be at least half as wide as would be predicted from the resolution filter in use. (Note that if the span is wide in comparison with the resolution bandwidth, there may be no difference between SPURS and CW.) When PULSE is chosen, if two candidate signals are within 0.4 of a division, they are assumed to be either time-related lines or spectral lines belonging to the same pulse. This extends to multiple lines; in a group of such lines, the highest-amplitude line will be identified as the center of the signal.

Two other factors affect the results. The first of these factors is separation. With any of the signal-processing commands, there must be at least a 3 dB notch between two signals in order for them to be individually recognized. The second factor is noise. Because noise is random, noise peaks may appear to be small signals. Whether or not these peaks are detected may be controlled by using a threshold with the signal-processing commands. This is discussed in the following description.

Setting the Threshold

We recommend that you set the value of THRHLD to AUTO. The threshold used for marker-related signal-processing commands will then be set slightly above the expected noise level, as determined by the spectrum analyzer sensitivity specification, attenuator setting, resolution bandwidth, and video filter bandwidth. The actual threshold value chosen will be returned in the response to the THRHLD? query.

The signal-finding commands use a threshold that may be set above the noise and below the signal level so that small noise peaks will not be falsely identified as signals. The threshold for the display data point command is sent in screen units as an optional argument to the commands. The threshold for the marker commands is set in dBm (or dBv, dBmv, or dBµv) by the THRHLD command.

No signal threshold setting will work in every case. An estimate of the noise level may be made by finding the amplitude of the negative noise peaks with FMIN (or MMIN) and adding a constant to get an estimate of the positive noise peaks. Because the peak-to-peak noise variation is a function of resolution bandwidth, the constant should be larger for wide bandwidths than for narrow bandwidths. Alternatively, the positive peak noise level can be found directly with FMAX (or MMAX) if all signals are tuned off the screen.

Acquiring Waveforms

Both the signal separation and the noise considerations mentioned previously need to be addressed when the waveform is acquired, as well as when the signal-processing commands are actually used.

Signal separation can be improved by selecting a narrower resolution with the RESBW command. This may require the sweep to be slowed to maintain amplitude and frequency calibration. (TIME AUTO will automatically maintain a calibrated display, if possible.)

Noise peaks can be reduced by smoothing the data as it is acquired by means of video filters and/or digital averaging. The bandwidths of the wide (VIDFLT WIDE) and narrow (VIDFLT NARROW) video filters are automatically changed in step with the resolution bandwidth.

Spectrum Search

To search a given frequency range for signals, set the spectrum analyzer to sweep the given range with FREQ and SPAN or STSTOP. If the frequency range is wide and/or the number of signals expected is large, search the range in sections.

With the marker or display data point at the left edge of the screen, find the lowest frequency signal with the MRGTNX command. Find successive signals by repeating the command. All signals in the display have been found when the MRGTNX command generates an SRQ message of NO SIGNAL FOUND. If the service request is disabled, either through SGERR OFF or RQS OFF, MRGTNX? can be used to determine whether the MRGTNX command found (FOUND will be returned) or did not find a signal (FAILED will be returned) .

Signal searches should be done in the single-sweep mode; take a new sweep only when all signals in one waveform have been found. If the total range is being searched in sections, move to the next section with a TUNE, FREQ, STSTOP, PSTEP, or MSTEP command and repeat the search.

By positioning the marker or display data point properly at the start of the search, the LFTNXT, MLFTNX, HRAMPL, and LRAMPL commands can also be used to find all signals in a given waveform.

Measuring Signal Frequency with COUNT

To measure the frequency of a signal, center the signal on screen using a FMAX;CENSIG combination or move the marker to the signal with PKFIND. The signal level at center screen (point 500) or at the marker must be at least 20 dB above the noise level. A 10 MHz/div span/div will provide a good compromise between the time required to span down and the need to be 20 dB above the noise. Send COUNT;COUNT? and input the signal frequency. The accuracy of the result is determined by the count resolution (CRES command).

```
SIGSWP (spectrum analyzer in single sweep mode)
SWEEP  (obtain a good waveform)
FMAX   (place marker on signal)
COUNT  (do count)
COUNT? (query count result)
```

Note that the COUNT performs a sweep as a side effect.

Noise

Noise can be overcome in several ways.

- Noise peaks can also be reduced (smoothed) by the video filters.
- Include a parameter in the signal search commands that sets a threshold for the signal search routine. If this parameter is set above the noise, but below any desired signal, the routine ignores the noise and finds the signal. A level may be estimated by using FMIN to locate the negative noise peaks and adding a constant to approximate the positive noise peaks. Adjust the constant if resolution bandwidth is changed. Another method is to force signals off-screen with the FREQ command and use FMAX to acquire the most positive noise peak.

Using REPEAT for Signal Tracking and Searches

The REPEAT command is used to repeat a set of commands as specified.

Spectrum Search Using REPEAT

The spectrum analyzer can perform a signal search by executing a loop in a single message and buffering the results without controller interaction. The controller can later turn its attention again to the spectrum analyzer and input the results. The following routine works on a waveform in digital storage that is not updated during processing.

```
STSTOP OMHZ,200MHZ        (select search range)
SIGSWP;SWEEP              (acquire waveform)
MFREQ OMHZ                (marker to left edge of
                          screen)
MRGTNX;MFREQ?;REPEAT 20(do 20 signal searches)
```

This example finds the first 20 signals between 0 MHz and 200 MHz.

Using the Time Measure Feature

This instrument employs a special time measurement feature that is available whenever the instrument is in the zero span mode of operation and ZETIME is on. The marker frequency readout or delta-marker frequency readout is replaced by a time or delta time readout, respectively. The time readout in the single-marker mode is the time to the marker position from the trigger point; this point is 1/2 division to the left of the screen. In the delta marker mode, the delta time readout gives the time difference between the two markers. In both cases, the time value is scaled from the marker position(s) and the time. No actual time measurement is done.

The time measurement feature is available only during certain timing conditions. If the TIME command is set to MAN, EXT, or a setting faster than 10 ms, a value of 200 will be returned by the MKTIME? query.

When the MARKER command is DELTA, both markers must be on the same trace for time measurement. If the markers are on different traces when ZEROSP is turned on, the secondary marker will move to the trace of the primary marker. (This marker will not move back when ZEROSP is turned off.) When either marker is assigned to a new trace using the MTRACE command, both markers (assuming delta markers are on) will move together.

In frequency-mode marker operation, the secondary marker remains at a constant frequency, while the primary marker remains at a constant horizontal location. However, in the time mode, both markers remain at constant horizontal positions as the sweep speed is changed.

In the frequency mode, the secondary marker can be tuned off screen, while in the time mode, the secondary marker cannot be tuned off screen.

The MFREQ command and the MFREQ? and MLOCAT? queries refer to frequency.

The MKTIME command and query that set and read the marker time, are only valid in Zero Span and ZETIME ON; any other use will result in a marker execution error message being issued. An attempt to set a time that would be off either the left or right of the screen will cause a marker execution error message to be issued. The MKTIME query will return Ð200 if the time value is unavailable, +200 if not in Zero Span or ZETIME is OFF, and +999.99 if the marker system is off (or MARKER is not set to DELTA when the secondary or delta time is requested).

Most of the frequency-related marker functions remain frequency related while ZEROSP is on. For instance, frequency entry with the MTUNE command still enters frequency. The STEP command still refers to frequency, and the PSTEP and MSTEP commands still increase and decrease the marker frequency. The MMAX and MMIN command arguments will still be frequency values. Thus, there will be no way to limit a marker maximum or minimum search to a specific horizontal range in zero span.

The MCEN and PKCEN commands are not available zero span.

Both BWMODE and SGTRAK will be IDLE while ZEROSP is on.

Using Multiband Sweep

It is possible to sweep a frequency range that covers more than one band as long as the entire frequency range is within the range of the preselector (1.7 GHz to 21 GHz) or 0 Hz to 21 GHz when using external mixers. The low-pass filter/preselector boundary may not be crossed when using the internal mixer, as this would lead to excessive wear of the preselector switch.

Entering the Multiband Sweep Mode — Enter the Multiband Sweep mode with the STSTOP command over the GPIB, or by recalling a setting using multiband sweep with the RECAL command. Multiband sweep is started automatically when STSTOP is used to enter a sweep that covers more than one band within the allowed multiband range. Send the STSTOP command with the desired start frequency followed by the stop frequency.

Multiband Sweep Operation — To sweep a range that covers more than one band, the spectrum analyzer first determines the bands involved and calculates the center frequency and span needed in each band to cover the desired range. Then, the microcomputer successively sets the instrument and performs one sweep in each band. The digital data is collected in the B digital display. This data is then compressed to cover the appropriate portion of the screen and is displayed in the A storage display.

If the start frequency is less than 3 GHz and the stop frequency is greater than 7.1 GHz, the 3 GHz to 7.1 GHz band is not used.

Instrument Operating Differences When in the Multiband Sweep Mode — Because of the method used to obtain a multiband sweep, certain instrument functions must be locked out and others will operate differently. To remind you of these differences, multiband execution error message 150 is issued when you enter the mode, and message 151 is issued when you exit the mode.

Following are the functional operating differences that are present when in the Multiband Sweep mode.

- To allow data collection in the B display of digital storage while AVIEW is ON, the storage must be set with AVIEW ON and BVIEW and BMINA OFF. These settings are changed automatically when the Multiband Sweep mode is entered and cannot be changed while in the mode. The existing settings are restored when multiband sweep is exited. If there is a waveform in SAVEA, it will be automatically overwritten. SAVEA and MXHLD operate normally. Since only the A waveform is displayed, SAVEA ON stops display updating.
- Displays may only be stored and recalled into the A register. If the B display is requested with the DRECAL or DSTORE command, execution error message 140 will be issued and no display will be recalled.
- If the multiband mode is exited using either the MXSPN or ZEROSP commands, the span saved will be the multiband span. The span will return to this value, or default to maximum, when either the Max or Zero Span Mode is cancelled (refer to the information under "Changing the Span in Exiting Multiband Sweep" for additional information). The instrument will not return to the multiband mode.
- The multiband frequency range displayed can only be changed by entering new start and stop frequencies with the STSTOP command. Changing the span with the SPAN command or directly entering a center frequency with the FREQ command exits the Multiband Sweep mode. Markers may be tuned over the displayed range only.
- The marker system treats the multiband display as if it were a saved or stored display; the tuning limits were mentioned previously. The MCEN and PKCEN commands cannot be used.
- With the ARES ON command, the resolution bandwidth used is the widest value required by the bands being swept. If any of the bands is uncalibrated, the UNCAL light will come on.

- With the TIME AUTO command, the sweep speed may vary as each band is swept. Any time value refers to a division of the sweep that gathered the data, not to the compressed display.
- If the sweep is not in TRIG FRERUN, the triggering conditions selected will be used for only the first (lowest-frequency) sweep of the sweep needed to do one complete multiband sweep. After this sweep is triggered, the remaining sweeps will be done in the Free Run mode. Similarly, once a single sweep is started with the SIGSWP command, the number of sweeps needed to form a complete display will occur. If a multiband sweep is interrupted by the SIGSWP command or GET command, the next sweep will be the lowest-frequency sweep.
- TIME MAN or EXT cannot be used when sweeping a multiband range.
- The COUNT command cannot be used in the Multiband Sweep mode.
- Auto peaking will be done as usual in a 2-division window centered on the center or marker frequency. If this range covers more than one band, peaking will be done in all bands covered. If there is at least one signal within a band, or portion of a band, the peak value of the frequency window that contains the largest signal will be updated.

Exiting Multiband Sweep — Multiband sweep may be exited in several ways.

- Use the FREQ command to enter a frequency.
- Recall a setting with RECAL with a sweep that falls within one band.
- Enter a sweep that falls within one band, using the STSTOP command.
- Enter FRQRNG.
- Change the span with the SPAN, ZEROSP, or MXSPN commands.

With MARKER ON, the center frequency is set to the Primary marker frequency. With MARKER OFF, the center frequency remains at the center frequency of the multiband sweep. The span is set to the span of the multiband sweep, or is defaulted to MXSPN if the multiband sweep value is larger than the maximum span of the band containing the center frequency. The command change that caused the exit from multiband sweep will then change either the center frequency or span, or both.

Comparing the Status Byte and the ERR? Response

The spectrum analyzer status byte and ERR? response described in *Status and Events* play complementary roles in GPIB system programming. The status byte is the spectrum analyzer response to a serial poll. The ERR? response is the spectrum analyzer answer to a device-dependent query message. The status byte provides information about instrument conditions by category; normal/abnormal, busy, command error, execution error, etc. The ERR? response details the cause of abnormal status; that is, what kind of error or warning prompted the spectrum analyzer to assert SRQ and report abnormal status.

Status bytes are not stacked. The code for the condition that caused the SRQ is not updated, although bit 5 reflects the present instrument state (1 for busy, 0 for not busy). Error codes, however, are accumulated until read and are reported in numerical order. While you can recover only one status byte, you may recover more than one code in the ERR? query response, indicating more than one abnormal condition occurred.

The status byte is cleared by a serial poll of that instrument. Error codes are cleared by reading them with the ERR? query. Reading the status byte does not clear the error codes, and vice versa. DCL and SDC (if addressed) clear both the status byte and error codes.

Execution and Transfer Times

The spectrum analyzer firmware typically takes 10 to 25 ms to execute commands received over the bus (refer to Table 4-1). This is the time the spectrum analyzer is busy following receipt of the end-of-message terminator (EOI or LF). Execution time for some commands stretches beyond 25 ms, however, because of interaction between the firmware and hardware or a wait to allow hardware response. If the spectrum analyzer is busy, any command will have to wait until the hardware is not busy (for example, if a signal count is being done, any command other than COUNT will have to wait).

Because of the way the spectrum analyzer handles output, it is free after it loads an output buffer. The additional time for the transfer (for CURVE, CURVE?, and SET?) is related to the listener for cases where the spectrum analyzer is faster.

Table 4-1: Execution and Transfer Times

Command	Time
FREQ, MFREQ	
0–1.8 GHz @ 10 MHz/div	800 ms
100 MHz step @ 10 MHz/div	150 ms
0–1.8 GHz @ 1 MHz/div	1.4 s
100 MHz step @ 1 MHz/div	350 ms
TUNE, MTUNE	
100 kHz step @ 100 kHz/div	80 ms
100 Hz step @ 100 kHz/div	60 ms
100 kHz step @ 100 Hz/div	1.4 s
100 Hz step @ 100 Hz/div	1.4 s
COUNT	
1 Hz resolution	2.6 s

Table 4-1: Execution and Transfer Times (Cont.)

Command	Time
1 kHz resolution	1.25 s
CNTCF	
1 Hz resolution	2.8 s
1 kHz resolution	1.75 s
EXMXR and FRQRNG if transfer switch or pre-selector/LPF switch is changed	Add 150 ms per switch
SPAN to phase lock span boundary (10 kHz/div to 1 MHz/div)	220 ms
IDENT ON	
@ 50 kHz	32.5 ms
@ 5 kHz	40.5 ms
@ 500 Hz	156 ms
REFLVL, RLMODE, MINATT, MAXPWR if RF attenuator is switched	Add 100 ms
CURVE	100 ms
CURVE Display Data Input	
Binary (from number array)	1.1 s
ASCII (as a string)	8.1 s
ASCII (as numbers)	20.0 s
CURVE?	60 ms
CURVE Display Data Output	
Binary (input as strings)	3.4 s
ASCII (input as a 10–19 string)	3.0 s
ASCII (input as number)	14.3 s
POINT (X argument only)	56 ms
FIBIG	760 ms
LFTNXT, RGTNXT	100 ms * (signal separation in div)
FMAX, FMIN	480 ms
SET? (command execution time)	0.5 s to 3 s
SET? response Display Data Output	624 ms

Table 4-1: Execution and Transfer Times (Cont.)

Command	Time
INIT	0.5 s to 3 s
CAL AUTO	35 s
PEAK AUTO	10,6 s @ 10 ms/div
DSTORE (display)	190 ms
DRECAL (display)	380 ms
STORE (settings)	80 ms
RECALL (settings)	500 ms
MFBIG	560 ms
MLFTNXT, MRGTNXT	400 ms * (signal separation in div)
TEST?	7.3 s
HRAMPL, LRAMPL	560 ms

Appendix A: 49X Compatibility Mode

This appendix discusses the special considerations that apply to using programs written for 49X series instruments with 2790 series spectrum analyzers.

Selecting 49X Compatibility Mode

To put your 2790 series spectrum analyzer into 49X compatibility mode, press the **CONFIG** key, select **GPIB Menu,** and **49X Mode ON**.

Command Differences

Command differences between native (2790 series) and compatibility (49X) modes are summarized in tables as follows:

- Table A-1 lists commands that are accepted by 2790 series spectrum analyzers but do not affect its operation.
- Table A-2 lists commands that are mode-dependent in their operation (the same command has a different effect in 49X compatibility mode than in native mode).
- Table A-3 lists commands that are similar to (and have the same name as) 49X commands, but differ in effect from the same command used with a 49X instrument.

Unique 49X Commands

The following commands are unique to the 49X compatibility mode. These commands are accepted by 2790 series spectrum analyzers but have no effect on their operation. The commands are accepted to allow programs written for 49X instruments to run on 2790 series instruments with little or no modification.

Table A-1 lists these unique commands. Because they are not documented in the *Syntax and Commands* section of this manual, they are documented here following the summary tables.

Table A-1: Unique 49X Commands and Queries

Mnemonic	Description
CRSOR	Peak/Average Cursor
DELFR	Delta Frequency
FINE	Fine Reference Level Steps
HELP?	Help
INPNUM	Input Number
M1ASGN	Assign Marker 1
M2ASGN	Assign Marker 2

Mode-Dependent Commands

The following commands are mode-dependent; their operation differs between native (2790 series) and compatibility (49X) mode. Because these commands are part of the 2790 series command set, they are documented in the *Command Descriptions* section of this manual. Refer to *Command Descriptions* for more information.

Table A-2: Mode-Dependent Commands

Mnemonic	Description	Summary of Difference
ENTE?	Enter Value in X Register	49X: SPAN/TIME values entered as units per division[1] 279X: SPAN/TIME values entered as units per screen
ID?	Identify	49X: Returns the 49X instrument most closely resembling the instrument in use. 279X: Returns instrument type
LORDO?	Lower Readout	49X: Same as 49X instruments 279X: Returns actual readouts from screen
MDRDO?	Macro Readout	49X: Same as 49X instruments 279X: Returns actual readouts from screen
PUTREG	Put X Register	49X: SPAn/TIMe values entered as units per division[1] 279X: SPAn/TIMe values entered as units per screen

Table A-2: Mode-Dependent Commands (Cont.)

Mnemonic	Description	Summary of Difference
SPAN	SPAN Frequency per Unit	49X: SPAN values entered as Span per division[1] 279X: SPAN values entered as Span per screen
TIME	TIME per Unit	49X: TIME values entered as Time per division[1] 279X: TIME values entered as Time per screen
UPRDO?	Upper Readout	49X: Same as 49X instruments 279X: Returns actual readouts from screen

[1] GPIB responses only; screen readout indicates span or time per screen.

Changed Commands

The following commands are similar to 49X commands, but operate differently when used with a 2790 series spectrum analyzer. Because most of these commands are part of the 2790 series command set, refer to *Command Descriptions* for more information; otherwise, see the command descriptions following this section.

Table A-3: Changed Commands

Mnemonic	Description	Summary of Difference
DSLINE	Display Line	Lines no longer automatically blanked if not displayed
FINE	Fine Reference Level Steps	No delta amplitude readout mode
MENU	Display Menu	279X menu structures are different; refer to *Command Descriptions*
MSTEP	Minus Step	If the command is sent before setting step size, the response is always as described in *Command Descriptions*
PEAK	Peaking	STORE can be used as an additional command argument; 49X peaking values KNOB and CAL have no effect; refer to *Command Descriptions*

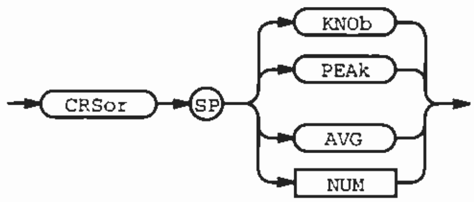
Table A-3: Changed Commands (Cont.)

Mnemonic	Description	Summary of Difference
PLOT?	Plot	Plots reflect 279X nomenclature and readouts
PSTEP	Plus Step	If the command is sent before setting step size, the response is always as described in *Command Descriptions*
RDATA	Register Data	
RDOUT	Readout Message	Displayed text is now mixed case instead of all upper case
SET?	Instrument Settings	The SET? response text differs; SPAN and TIME values are per division[1]
STEP	Step Size	Step values are as described in *Command Descriptions*
TEST?	Internal Test	TEST? query response differences related to 49X Option 45 do not apply
VRTDSP	Vertical Display	There is no delta amplitude readout mode

[1] GPIB responses only; screen readout indicates span or time per screen.

CRSor (peak/average cursor)

Syntax:



Arguments: KNOB — This argument has no effect.

CAUTION

PEAK — The peak value digitized at each point is used to update digital storage. This is the same as setting the cursor to its lowest (minimum) position. Use UDASGN AVGPOS to assign the peak/average cursor position to the user definable knob.

AVG — Average values are used to update the waveforms. PEAK AVG is the same as if the cursor is set to its highest (maximum) position.

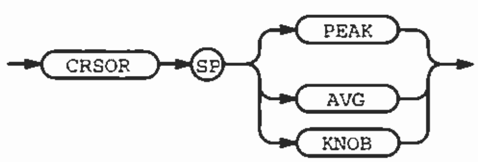
NUM —

0 = KNOB
1 = PEAK
2 = AVG

Interactions: Averaging can reduce the value in digital storage for signals with very narrow response or pulsed signals.

Query:

Response:



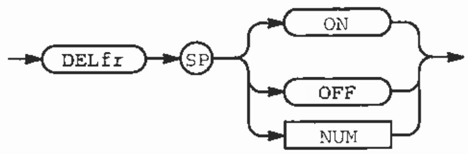
The query returns KNOB if the cursor is not set to peak or average, even if the user definable knob is not currently assigned to cursor control.

Macro Memory Used: 2 bytes.

Power-up Value: KNOB.

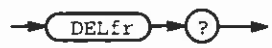
DELfr (delta frequency)

Syntax:

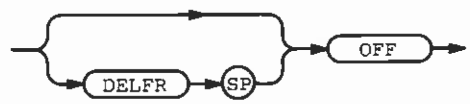


The DELFR command is accepted by the instrument, but has no effect on its operation.

Query:



Response:

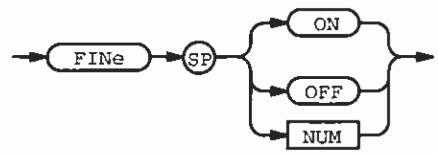


Macro Memory Used: 2 bytes.

Power-up Value: OFF.

FINe (fine reference level steps)

Syntax:



Arguments: **ON** — Small steps (1 dB, regardless of vertical scale setting) are selected for the INC or DEC arguments in the reference level command (see REFLVL and VERTDSP:LIN for details).

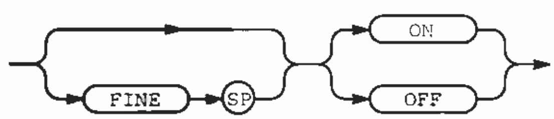
OFF — Sets the reference level step to AUTO.

Note

Using a log vertical scale of 2 dB/div with Fine off, the 49X reference step level would be 1 db. Under the same condition, the 279X reference step level will be 2 db.

Query:

Response:



Interactions: This command, along with VRTDSP, controls the spectrum analyzer response to REFLVL INC or DEC.

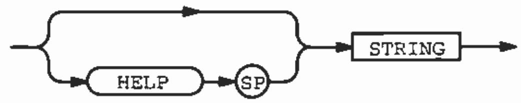
Macro Memory Used: 2 bytes.

Power-up Value: OFF.

HELp? (help)

Query:

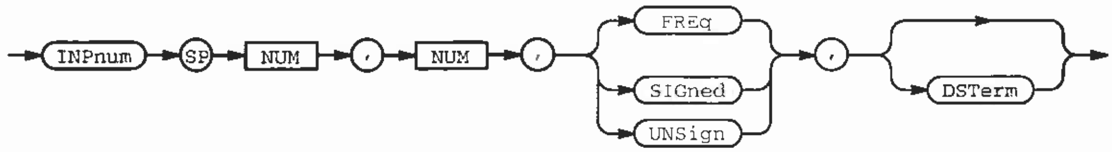
Response: The response is a list of all command headers in the GPIB language.



The string will report that this instrument has no help text.

INPnum (input number)

Syntax:



The INPNUM command allows you to input a number from the keypad and store the number in XREG.

With FREQ, the GHz, MHz, KHz, and Hz terminators are valid. GHz will multiply the input number by 1,000,000,000, MHz will multiply the input number by 1,000,000, KHz will multiply the input number by 1,000, and Hz will not change the input number.

With SIGNED, the +dBX and −dBX terminators are valid. Selecting +dBX will not change the input number but selecting −dBX will make the input number negative.

With UNSIGN, the dB terminator is valid.

If an undefined key is pressed, the macro will be aborted. If a terminator is pressed without a number or ESC is pressed, XREG stays the same. You can load a default value in XREG with the ENTER command before INPNUM; then, just press the terminator to get that value (default).

Range: The range of the first NUM is 1 to 12 if DSTERM is used or 1 to 15 if DSTERM is not used. The range of the second NUM is 1 to 20.

Examples: The following example illustrates the use of the INPNUM command.

```
 80   Z=1 ! ADDRESS OF SPECTRUM ANALYZER
100   Print#z:"CLEAR"
110   Print#z:"TEXT"
120   Print#z:"PRINT 1,1,'ENTER NUMBER OF
        HARMONICS TO LOOK FOR'"
130   Print#z:"PRINT 5,8,'ENTER NUMBER:'"
140   Print#z:"PRINT 14,1,'RANGE: 1 to 10'"
150   Print#z:"INPNUM 5,20,UNSIGN,DSTERM"
```

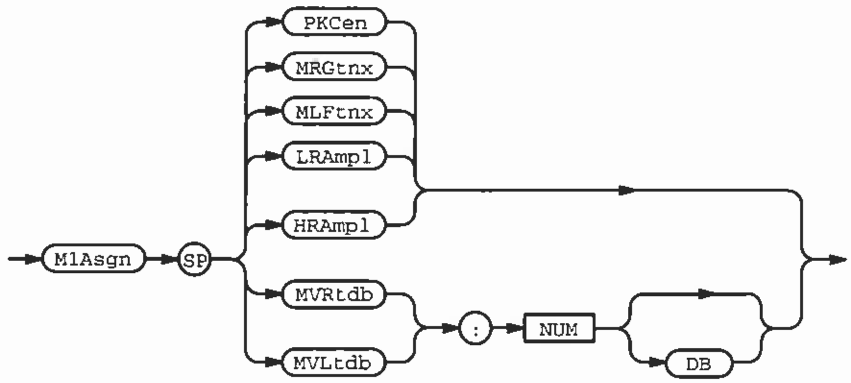
Line 100	Clears the macro readout buffer.
Line 120	Prints the statement on line 1 of the screen beginning at character position 1.
Line 130	Prints the statement on line 5 of the screen, indented 7 character spaces.
Line 140	Prints the statement on line 14 of the screen beginning at character position 1.

Line 150	Prepares to enter a number on line 5 of the screen, indented 19 character spaces.

Macro Memory Used: 5 bytes

M1Asgn (assign marker pushbutton 1)

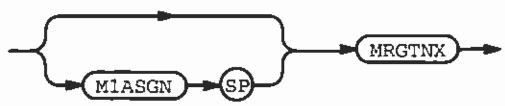
Syntax:



The M1ASGN command is accepted by the instrument, but has no effect on its operation.

Query:

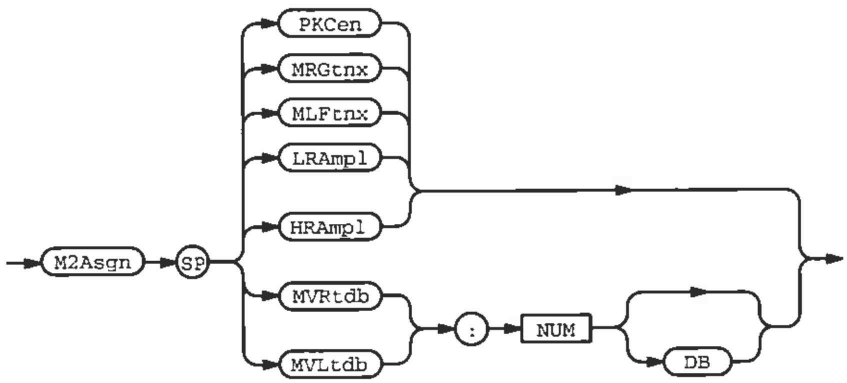
Response:



Responses to the M1ASGN query are not relevant to 2790 series instruments.

M2Asgn (assign marker pushbutton 2)

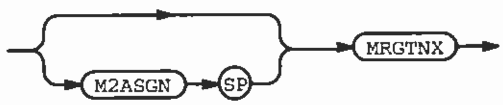
Syntax:



The M12ASGN command is accepted by the instrument, but has no effect on its operation.

Query:

Response:



Responses to the M1ASGN query are not relevant to 2790 series instruments.

Appendix B: Macro Memory

This appendix lists 2790 series commands that can be used in macros and the memory that they require. Commands omitted cannot be used in macros.

Table B-1: Macro Memory Requirements

Command	Macro Memory Used (bytes)	Description
ARES	2	automatic resolution bandwidth
AVGPOS	2	average position
AVIEW	2	A waveform display
BMINA	2	B minus A waverform display
BVIEW	2	B waveform display
BWMODE	2	bandwidth mode
BWNUM	3	marker bandwidth number
CENSIG	1	center signal
CLEAR	2	clear macro readout buffer
CLIP	2	blank baseline
CLVAL	3	conversion loss value
CNTCF	1	count to center frequency
CONLOS	2	conversion loss mode
COUNT	2	counter
CRES	4	counter resolution
DEGAUS	1	dugauss tuning coils
DINT	2	display intensity
DISCOR	2	disable tuning correction
DIVIDE	1	divide
DONE	1	macro execution finished
DPMK	1	display pointer to marker
DRECAL	3	recall display
DSLINE	4	display line

Table B-1: Macro Memory Requirements (Cont.)

Command	Macro Memory Used (bytes)	Description
DSTORE	3	store display
DT	2	define triggered events
ECR	2	end of sweep corrections
ELSE	4	else
EMAC	0	end macro
ENCAL	2	enable calibration
ENDI	1	end if
ENPAR	30	enter parameter
ENTER	10	enter value
EOS	2	end sweep
EXCHG	2	exchange registers
EXMXR	1	external mixer
FIBIG	2	find big
FIRST	6	first LO frequency
FMAX	5	find maximim value
FMIN	5	findminimum value
FOR	20	for
FREQ	6	center frequency
FRQRNG	2	frequency range
GETWFM	1	get waveform
GOSUB	5	go subroutine
GOTO	5	go to
GRAT	2	graticule
HDR	2	header
HRAMPL	1	next higher amplitude
IF	23	if
IDENT	2	idnetify
IMPED	2	impedance

Table B-1: Macro Memory Requirements (Cont.)

Command	Macro Memory Used (bytes)	Description
INIT	1	initialize settings
INTEGR	1	integer
KPASGN	2	keypad assignment
LABEL	2	macro label
LFTNXT	2	left next
LRAMPL	1	next lower amplitude
MARKER	2	marker mode
MAXPWR	4	maximum input power
MCEN	1	marker to center
MCPOIN	2	marker coupled to pointer
MCSTOP	1	macro stop
MDATA	9	store numeric data
MENU	2	display menu
MEXCHG	1	marker exchange
MFBIG	1	marker peak find
MFREQ	6	marker frequency
MINATT	2	minimum RF attenuation
MKDP	1	marker display pointer
MKTIME	6	marker time
MLFTNX	1	move left next
MMAX	12	marker move to maximimum
MMIN	12	marker move to minimum
MRESTO	5	restore data pointer
MRGTNX	1	marker right next
MSTEP	1	minus step
MTOP	1	marker to reference level
MTRACE	3	marker trace position
MTUNE	6	tune marker

Table B-1: Macro Memory Requirements (Cont.)

Command	Macro Memory Used (bytes)	Description
MULT	1	multiply
MVLTBD	3	move left x dB
MVRTDB	3	move right x dB
MXHLD	2	maximum hold
MXSPN	2	maximum span mode
NEXT	5	next variable
NSELVL	2	noise level normalization
NUMEV	2	number of events
PAUSE	1	pause macro
PEAK	4	peaking
PKCEN	1	marker to maximum and center
PKFIND	1	marker to maximum above threshold
PLSTR	2	pulse stretcher
PLUS	1	add
POFSET	2	set K
POINT	5	display data point
POP	1	put Y into X register
PRINT	47	print number or string
PSTEP	1	plus step
PTYPE	2	plotter type
PUTREG	2	put X register to setting
RDOUT	41	readout message
READ	1	read and store in X register
RECALL	2	recall settings
REDOUT	2	readout
REFLVL	4	reference level
RESBW	5	resolution bandwidth
RETURN	1	return from subroutine

Table B-1: Macro Memory Requirements (Cont.)

Command	Macro Memory Used (bytes)	Description
RGMODE	2	reduced gain mode
RGTNXT	2	right next
RLMODE	2	reference level mode
RLUNIT	2	reference level units
ROFSET	3	reference level offset
RQS	2	request for service
RSTEP	3	reference level step
SAMODE	2	sideband analyzer mode
SAVEA	2	save A waveform
SECOND	6	second low frequency
SGERR	2	signal find error
SGTRAK	2	signal track
SIGSWP	1	single sweep
SPAN	5	frequency span
SSR	2	send service request
STEP	7	step size
STNUM	2	store number
STORE	2	store settings
STSTOP	12	start-stop sweep
STYPE	2	signal type
SUBT	1	subtract
SWEEP	1	take a sweep
TEXT	2	text mode
TGMODE	2	tracking generator mode
THRHLD	4	marker threshold
TIME	5	time per screen
TMODE	2	set tune mode
TOPSIG	1	move to top of graticule

Table B-1: Macro Memory Requirements (Cont.)

Command	Macro Memory Used (bytes)	Description
TRIG	2	triggering
TUNE	6	incremental frequency change
UDASGN	2	assign user definable knob
VIDFLT	2	video filter
VRTDSP	3	vertical display
WAIT	1	wait for end of sweep
WARMSG	2	warning message
ZEROSP	2	zero span mode
ZETIME	2	zero span
ZOOM	22	zoom

Appendix C: ASCII Character Set

The chart on the following page describes the characters available for the spectrum analyzer.

Table C-1: The ASCII & GPIB Code Chart

<table>
<tr><th></th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th></tr>
<tr><td>0</td><td>0
NUL
0 0</td><td>20
DLE
10 16</td><td>40 0
SP
20 32</td><td>60 16
0
30 48</td><td>100 0
@
40 64</td><td>120 16
P
50 80</td><td>140 0
‘
60 96</td><td>160 16
p
70 112</td></tr>
<tr><td>1</td><td>1 GTL
SOH
1 1</td><td>21 LLO
DC1
11 17</td><td>41 1
!
21 33</td><td>61 17
1
31 49</td><td>101 1
A
41 65</td><td>121 17
Q
51 81</td><td>141 1
a
61 97</td><td>161 17
q
71 113</td></tr>
<tr><td>2</td><td>2
STX
2 2</td><td>22
DC2
12 18</td><td>42 2
”
22 34</td><td>62 18
2
32 50</td><td>102 2
B
42 66</td><td>122 18
R
52 82</td><td>142 2
b
62 98</td><td>162 18
r
72 114</td></tr>
<tr><td>3</td><td>3
ETX
3 3</td><td>23
DC3
13 19</td><td>43 3
#
23 35</td><td>63 19
3
33 51</td><td>103 3
C
43 67</td><td>123 19
S
53 83</td><td>143 3
c
63 99</td><td>163 19
s
73 115</td></tr>
<tr><td>4</td><td>4 SDC
EOT
4 4</td><td>24 DCL
DC4
14 20</td><td>44 4
$
24 36</td><td>64 20
4
34 52</td><td>104 4
D
44 68</td><td>124 20
T
54 84</td><td>144 4
d
64 100</td><td>164 20
t
74 116</td></tr>
<tr><td>5</td><td>5 PPC
ENQ
5 5</td><td>25 PPU
NAK
15 21</td><td>45 5
%
25 37</td><td>65 21
5
35 53</td><td>105 5
E
45 69</td><td>125 21
U
55 85</td><td>145 5
e
65 101</td><td>165 21
u
75 117</td></tr>
<tr><td>6</td><td>6
ACK
6 6</td><td>26
SYN
16 22</td><td>46 6
&
26 38</td><td>66 22
6
36 54</td><td>106 6
F
46 70</td><td>126 22
V
56 86</td><td>146 6
f
66 102</td><td>166 22
v
76 118</td></tr>
<tr><td>7</td><td>7
BEL
7 7</td><td>27
ETB
17 23</td><td>47 7
’
27 39</td><td>67 23
7
37 55</td><td>107 7
G
47 71</td><td>127 23
W
57 87</td><td>147 7
g
67 103</td><td>167 23
w
77 119</td></tr>
<tr><td>8</td><td>10 GET
BS
8 8</td><td>30 SPE
CAN
18 24</td><td>50 8
(
28 40</td><td>70 24
8
38 56</td><td>110 8
H
48 72</td><td>130 24
X
58 88</td><td>150 8
h
68 104</td><td>170 24
x
78 120</td></tr>
<tr><td>9</td><td>11 TCT
HT
9 9</td><td>31 SPD
EM
19 25</td><td>51 9
)
29 41</td><td>71 25
9
39 57</td><td>111 9
I
49 73</td><td>131 25
Y
59 89</td><td>151 9
i
69 105</td><td>171 25
y
79 121</td></tr>
<tr><td>A</td><td>12
LF
A 10</td><td>32
SUB
1A 26</td><td>52 10
*
2A 42</td><td>72 26
:
3A 58</td><td>112 10
J
4A 74</td><td>132 26
Z
5A 90</td><td>152 10
j
6A 106</td><td>172 26
z
7A 122</td></tr>
<tr><td>B</td><td>13
VT
B 11</td><td>33
ESC
1B 27</td><td>53 11
+
2B 43</td><td>73 27
;
3B 59</td><td>113 11
K
4B 75</td><td>133 27
[
5B 91</td><td>153 11
k
6B 107</td><td>173 27
{
7B 123</td></tr>
<tr><td>C</td><td>14
FF
C 12</td><td>34
FS
1C 28</td><td>54 12
,
2C 44</td><td>74 28
<
3C 60</td><td>114 12
L
4C 76</td><td>134 28
\
5C 92</td><td>154 12
l
6C 108</td><td>174 28
|
7C 124</td></tr>
<tr><td>D</td><td>15
CR
D 13</td><td>35
GS
1D 29</td><td>55 13
–
2D 45</td><td>75 29
=
3D 61</td><td>115 13
M
4D 77</td><td>135 29
]
5D 93</td><td>155 13
m
6D 109</td><td>175 29
}
7D 125</td></tr>
<tr><td>E</td><td>16
SO
E 14</td><td>36
RS
1E 30</td><td>56 14
.
2E 46</td><td>76 30
>
3E 62</td><td>116 14
N
4E 78</td><td>136 30
^
5E 94</td><td>156 14
n
6E 110</td><td>176 30
~
7E 126</td></tr>
<tr><td>F</td><td>17
SI
F 15</td><td>37
US
1F 31</td><td>57 15
/
2F 47</td><td>77 UNL
?
3F 63</td><td>117 15
O
4F 79</td><td>137 UNT
_
5F 95</td><td>157 15
o
6F 111</td><td>177
DEL (RUBOUT)
7F 127</td></tr>
<tr><td></td><td>ADDRESSED COMMANDS</td><td>UNIVERSAL COMMANDS</td><td colspan="2">LISTEN ADDRESSES</td><td colspan="2">TALK ADDRESSES</td><td colspan="2">SECONDARY ADDRESSES OR COMMANDS</td></tr>
</table>

<table>
<tr><td rowspan="3">KEY</td><td>octal</td><td>25 PPU</td><td>GPIB code</td></tr>
<tr><td></td><td>NAK</td><td>ASCII character</td></tr>
<tr><td>hex</td><td>15 21</td><td>decimal</td></tr>
</table>