TEK PROGRAMMERS MANUAL

Part No.: 070-5559-01
Product Group 26

494AP
SPECTRUM ANALYZER

Tektronix
COMMITTED TO EXCELLENCE

PREFACE

This manual is one of a set of product manuals for the TEKTRONIX 494AP Programmable Spectrum Analyzer. This manual describes the programmable functions of the spectrum analyzer and how to use them for remote operation. The manual organization is shown in the Table of Contents. The manuals that are available now in addition to this Programmers Manual are the

- 494A/494AP Operators Manual (standard accessory),
- 494A/494AP Service Manuals, Volume 1 and 2 (optional accessories),
- 494A/494AP Operators Handbook (optional accessory), and
- 494AP/495 Option 05 Programmers Reference Guide

For manual ordering information, contact your local Tektronix Field Office or representative or refer to the Accessories portion of the Replaceable Mechanical Parts list in the Service Manual, Volume 2.

Standards and Conventions Used

Most terminology is consistent with standards adapted by IEEE and IEC. A glossary of terms is provided in Appendix A. Abbreviations in the documentation are consistent with ANSI Y1.1-1972. GPIB functions conform to the IEEE 488-1978 Standard and the Tektronix Interface Standard for GPIB Codes, Formats, Conventions, and Features. Copies of ANSI and IEEE standards can be ordered from the Institute of Electrical and Electronic Engineers Inc. Contact your local Tektronix Field Office or representative if you have questions regarding the Tektronix reference document.

Change/History Information

Any change information that involves manual corrections or additional information is located behind the tabbed Change Information page at the back of this manual.

History information, as well as the updated data, is combined within this manual when the page(s) is revised. A revised page is identified by a revision date located in the lower inside corner of the page.

Unpacking and Initial Inspection

Instructions for unpacking and preparing the instrument for use are described in Section 3 of the Operators Manual.

Storage and Repackaging

Instructions for short- and long-term storage and instrument repackaging for shipment are described in Section 3 of the Operators Manual.

TABLE OF CONTENTS

Page

Section 4 (continued)

Page

Section 4 (continued)

Page

Section 6 (continued)

Section 7 DISPLAY DATA AND CRT READOUT I/O

Section 8 WAVEFORM PROCESSING

Page

Section 8 (continued)

Section 9 SYSTEM COMMANDS AND QUERIES

Section 10 HELPS AND HINTS

Page

Section 10 (continued)

Page

Appendix A IEEE STD 488 (GPIB) SYSTEM CONCEPTS

Appendix B PROGRAMMING FUNCTIONS

LIST OF ILLUSTRATIONS

LIST OF TABLES

SAFETY SUMMARY

(Refer all servicing to qualified servicing personnel)

The safety information in this summary is for both operating and servicing personnel. Specific warnings and cautions will be found throughout the manual where they apply, but may not appear in this summary.

CONFORMANCE TO INDUSTRY STANDARDS

This instrument complies with the following Industry Safety Standards and Regulatory Requirements.

Safety

CSA: Electrical Bulletin

FM: Electrical Utilization Standard Class 3820

ANSI C39.5 — Safety Requirements for Electrical and Electronic Measuring and Controlling Instrumentation.

IEC 348 (2nd edition) — Safety Requirements for Electronic Measuring Apparatus.

Regulatory Requirements

VDE 0871 Class B — Regulations for RFI Suppression of High Frequency Apparatus and Installations.

TERMS

In This Manual

CAUTION statements identify conditions or practices that could result in damage to the equipment or other property.

WARNING statements identify conditions or practices that could result in personal injury or loss of life.

As Marked on Equipment

CAUTION indicates a personal injury hazard not immediately accessible as one reads the marking, or a hazard to property, including the equipment itself.

DANGER indicates a personal injury hazard immediately accessible as one reads the marking.

SYMBOLS

In This Manual

⚠ This symbol indicates where applicable cautionary or other information is to be found.

As Marked on Equipment

⚡ DANGER — High voltage.

⏚ Protective ground (earth) terminal.

⚠ ATTENTION — refer to manual.

ⓘ Refer to manual

POWER

Power Source

This product is intended to operate from a power source that will not apply more than 250 V rms between the supply conductors or between either supply conductor and ground. A protective ground connection by way of the grounding conductor in the power cord is essential for safe operation.

Grounding the Product

This product is grounded through the grounding conductor of the power cord. To avoid electrical shock, plug the power cord into a properly wired receptable before connecting it to the power terminal. A protective ground connection by way of the grounding conductor in the power cord is essential for safe operation.

Danger From Loss of Ground

Upon loss of the protective-ground connection, all accessible conductive parts (including knobs and controls that may appear to be insulating) can render an electric shock.

Use the Proper Power Cord

Use only the power cord and connector specified for your product.

Use only a power cord that is in good condition.

CSA certification applies to the spectrum analyzer with CSA-certified power cords only (the power cord shipped with your instrument and Tektronix Option A4). International power cords (Tektronix Options A1, A2, A3, and A5) are approved only for the country of use, and are not included in the CSA certification.

Refer cord and connector changes to qualified service personnel.

For detailed information on power cords and connectors, see the Maintenance section in the Service Manual, Volume 1.

Use the Proper Fuse

To avoid fire hazard or equipment damage, use only the fuse of correct type, voltage rating, and current rating for your product (as specified in the Replaceable Electrical Parts list in Volume 2 of the Service Manual). Refer fuse replacement to qualified service personnel.

OPERATIONAL PRECAUTIONS

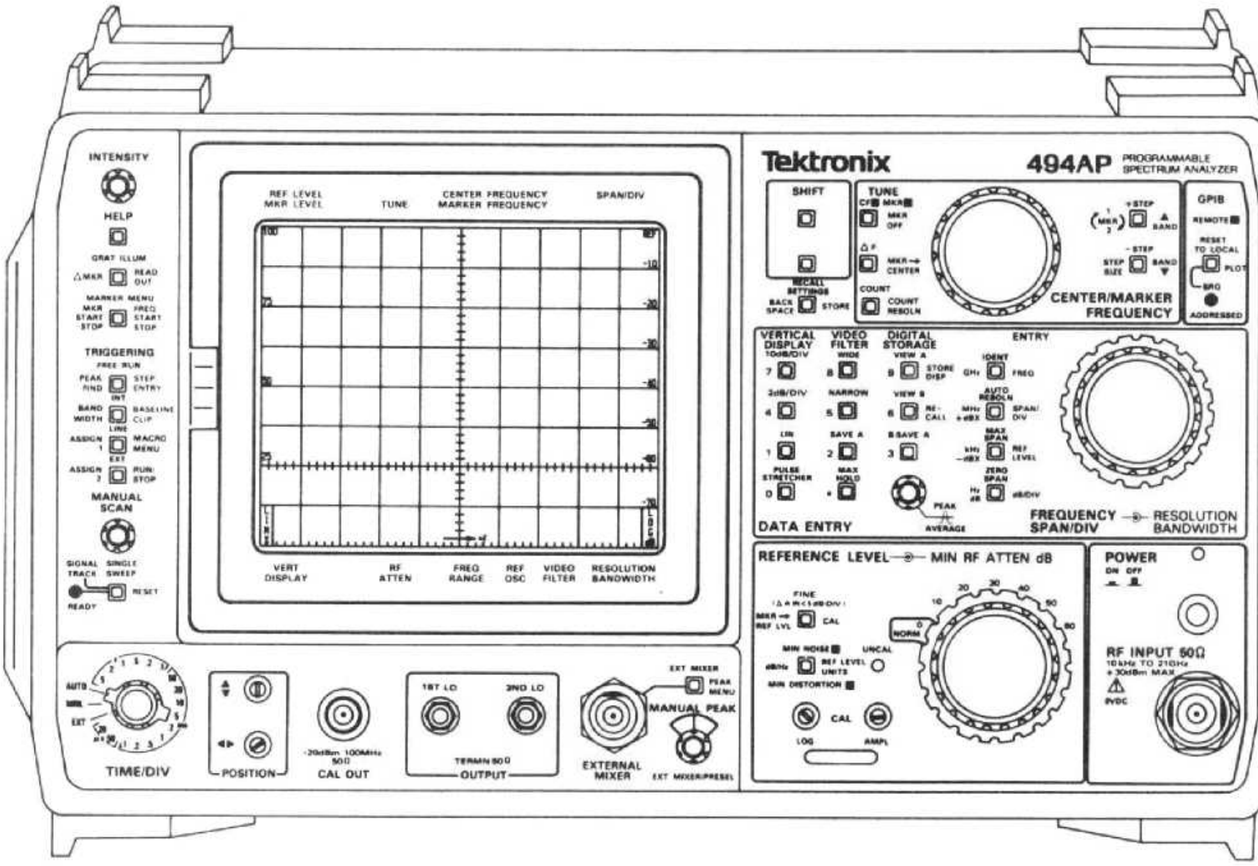
Do Not Operate in Explosive Atmospheres

To avoid explosion, do not operate this product in an explosive atmosphere unless it has been specifically certified for such operation.

Do Not Remove Covers or Panels

To avoid personal injury, do not remove the product covers or panels unless you are qualified to do so. Do not operate the product without the covers and panels properly installed.

Dangerous voltages exist at several points in this product. To avoid personal injury, do not touch exposed connections and components while power is on. REFER ALL SERVICING TO QUALIFIED SERVICE PERSONNEL.



5557-01

TEKTRONIX 494AP Programmable Spectrum Analyzer

INTRODUCTION TO GPIB OPERATION

The TEKTRONIX 494AP Programmable Spectrum Analyzer adds remote control and automated spectrum data acquisition and analysis to the performance and portability features of the TEKTRONIX 494A non-programmable Spectrum Analyzer. The spectrum analyzer front panel can be controlled remotely (except for those controls intended for local use only, such as INTENSITY). 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. 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.

GPIB PUSHBUTTON AND INDICATORS

(see Figure 1-1)

RESET TO LOCAL/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 and pushbuttons are not active; indicators will still reflect the current state of all front-panel functions except TIME/DIV, MIN RF ATTEN dB, and PEAK/AVERAGE.

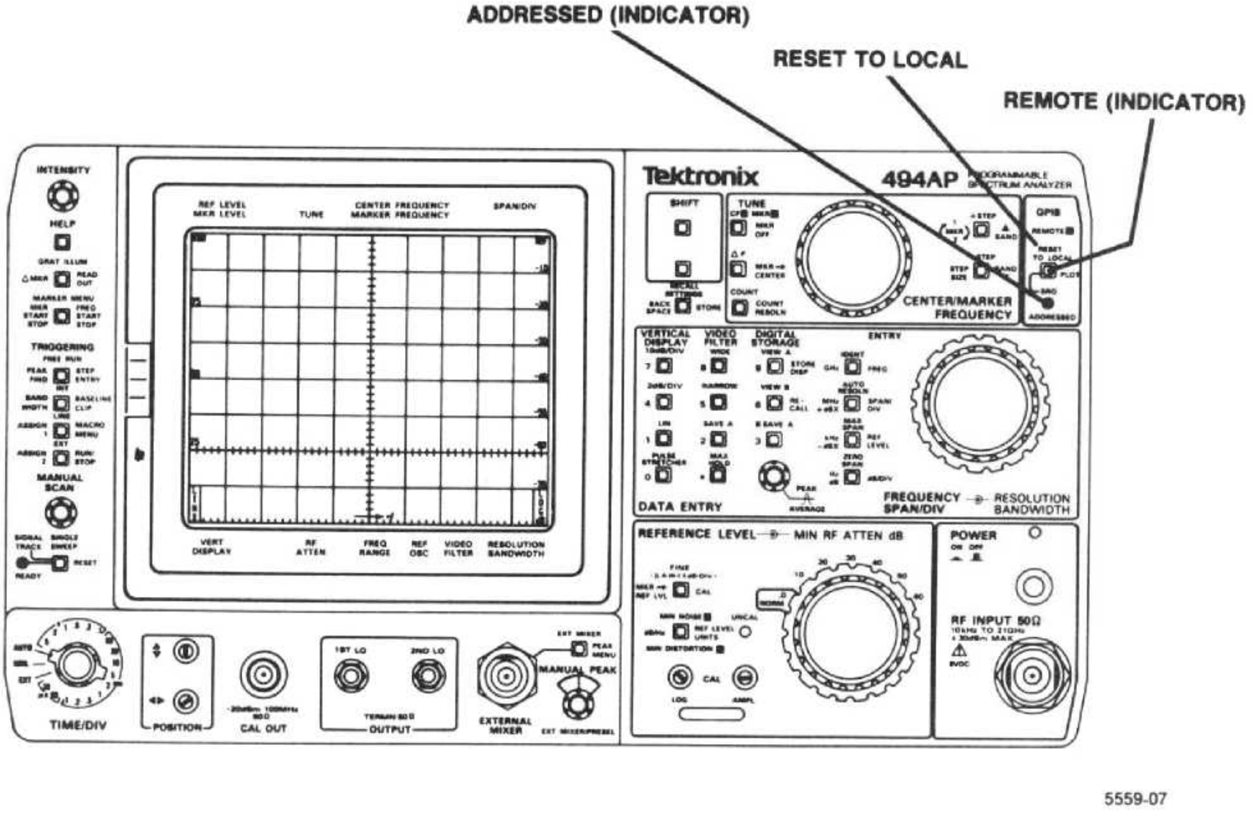


Figure 1-1. GPIB pushbutton and indicators.

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 RESET 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 except as necessary to match the settings of front-panel controls for TIME/DIV, MIN RF ATTEN dB, and PEAK/AVERAGE.

The spectrum analyzer flashes the instrument name, firmware version number, the GPIB address, and the macro status message on the crt when RESET TO LOCAL is pressed. If there is no macro running, the message will be MACRO–OFF(#) where # indicates the menu location of the last macro that was run. If there is a macro running, the message will be MACRO–RUN(#) where # indicates the menu location of the macro that is running. If there is a macro in process but stopped, the message will be MACRO–STOP(#) where # indicates the menu location of the macro that is stopped. If a macro is being stored in memory, the message will be MACRO–STORE(#) where # indicates the menu location where the macro is being stored.

<Blue-SHIFT> PLOT

Press this pushbutton when the spectrum analyzer is in the talk-only mode for the instrument to send the appropriate commands over the GPIB to a plotter, which must be in the listen-only mode, connected to the bus (see the TALK ONLY, LISTEN ONLY switch descriptions later in this section). The spectrum analyzer display (waveform, marker(s), graticule, and crt readout) can be recreated on a TEKTRONIX 4662 Option 01 or 4662 Option 31 Interactive Digital Plotter (or a 4663 in the 4662 emulation mode); or a Hewlett-Packard HP7470A or HP7475A, HP7580B, HP7585B, or HP7586B; or a Gould 6310 or 6320 plotter. Select plotter type with <blue-SHIFT> SAVE A, described in Section 4 of this manual. A bus controller is not required.

<Green-SHIFT> SRQ

This SRQ (service request) pushbutton sequence gets the controller's attention so it will listen/respond to the spectrum analyzer. 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 <GREEN-SHIFT> SRQ WHEN READY". This would instruct the controller to go on to the next step. An SRQ will only be issued if RQS is on.

ADDRESSED

This indicator is lit when the spectrum analyzer is addressed to listen or talk.

GPIB Function Readout

A single character appears in the lower crt readout when the spectrum analyzer is talking (T), or listening (L); see Figure 1-2. Two characters will appear in this location if the instrument is talking or listening and also requesting service (S), or if the instrument is in both the talk-only and listen-only modes.

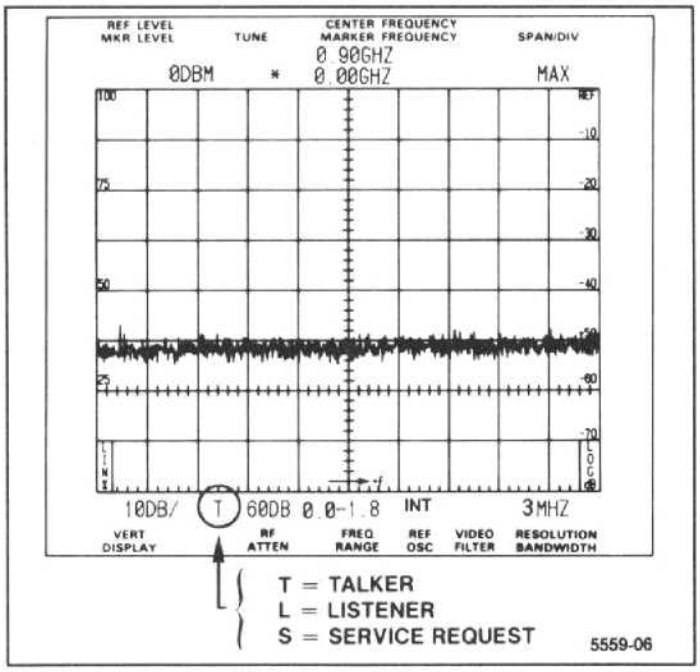


Figure 1-2. Status of active GPIB functions.

Setting the GPIB ADDRESS Switches

The rear-panel GPIB ADDRESS switches shown in Figure 1-3 set the value of the GPIB address (refer to Table 1-1). The instrument's primary address (0 through 31) is the value of the lower five bits, which are labeled 4 through 8 in Figure 1-3. These switches are read each time power is turned on to the instrument and again each time the RESET TO LOCAL or <blue-SHIFT> PLOT pushbutton is pressed.

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.

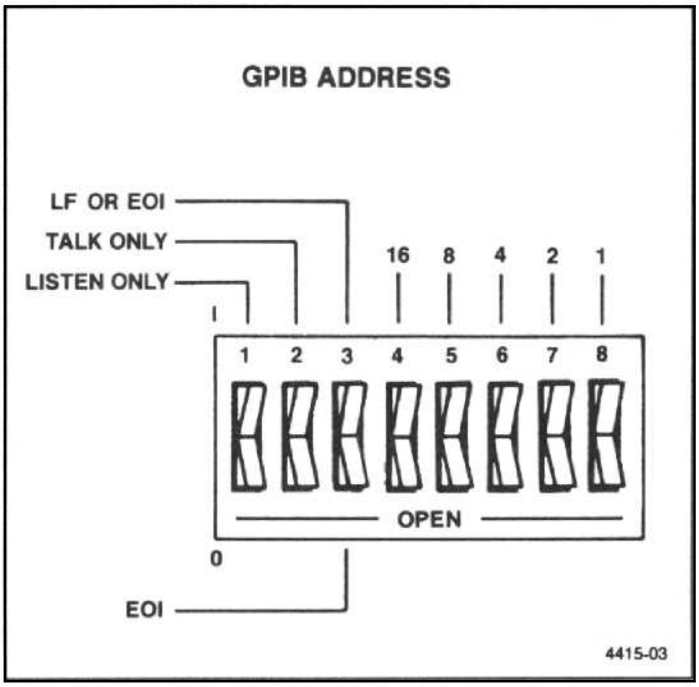


Figure 1-3. Rear-panel GPIB ADDRESS switches. The LF OR EOI switch (message terminator) and the TALK ONLY and LISTEN ONLY switches are part of the same switch bank.

Set the switches as desired, but 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. Remember, if you change these switches when the instrument is running, you must press RESET TO LOCAL or <blue-SHIFT> PLOT to cause the primary address to be updated.

Setting the LF OR EOI Switch

Switch 3 of the rear-panel GPIB ADDRESS switch bank (see Figure 1-3OQ) selects the terminator for messages on the bus. If LF OR EOI is selected (switch up, 1), 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 (switch down, 0), the spectrum analyzer interprets the end message (EOI asserted as well as a data byte) as the end of a message.

Table 1-1

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

Switch 3 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.

Figure 1-4 shows the effect of this switch for both input and output.

Select EOI (switch down) for Tektronix controllers. The other position of this switch is provided to accommodate most other controllers. A change in this switch takes immediate effect.

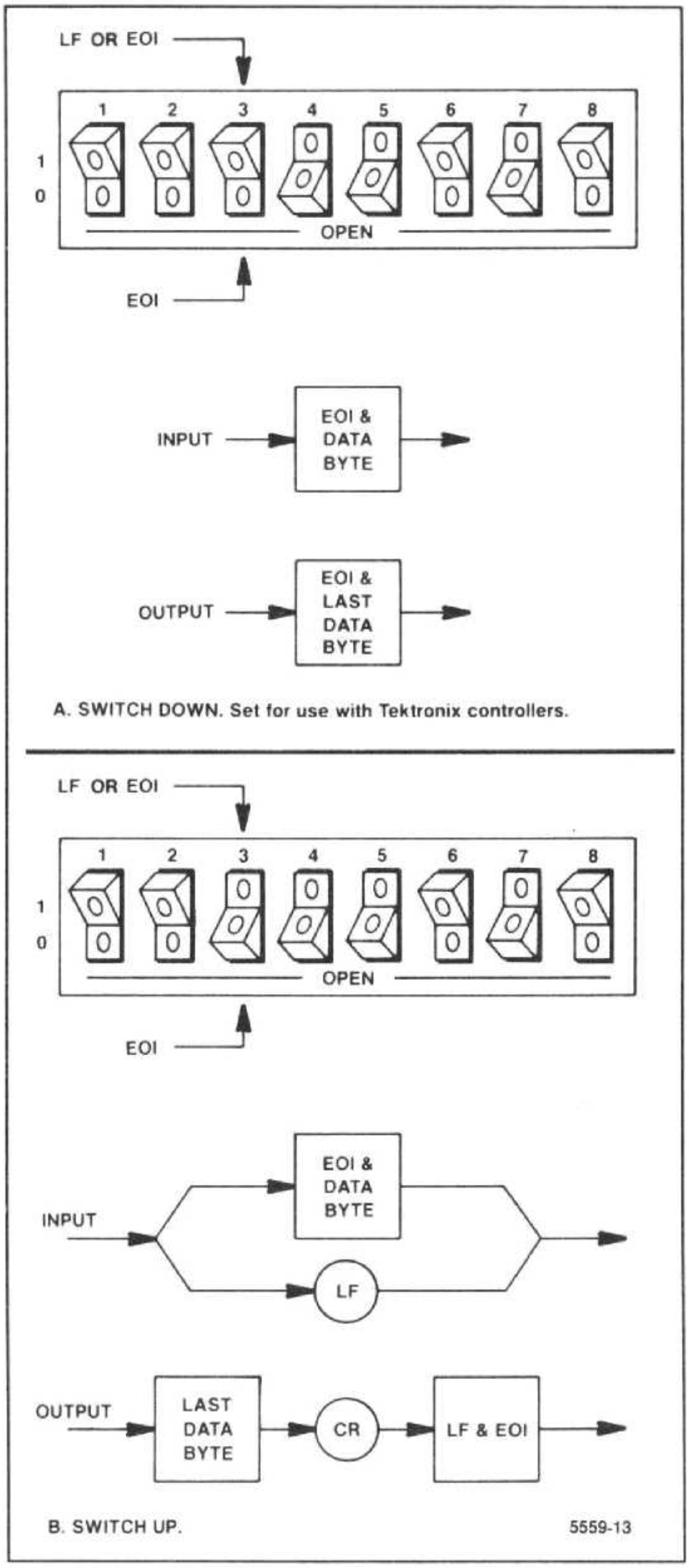


Figure 1-4. Effect of message terminator switch for input and output.

Setting the TALK ONLY and LISTEN ONLY Switches

The spectrum analyzer switches for talk-only and listen-only operation are part of the GPIB ADDRESS switch bank shown in Figure 1-3. Either, or both, switch is on when it is in the up, or 1, position, and either is off in the down, or 0, position. If instrument power is on, press RESET TO LOCAL or <blue-SHIFT> PLOT for a change in these switches to take effect.

Both the TALK ONLY and LISTEN ONLY switches must be off when the spectrum analyzer is used with any controller. As contrast, both switches must be on to allow spectrum analyzer output to be exchanged with a storage device without the need of a controller. The TALK ONLY switch must be on to allow spectrum analyzer output to be sent to a plotter. With the LISTEN ONLY switch on, information sent to a storage device can be fed back to the spectrum analyzer.

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 (refer to Appendix A in this manual for the basic concepts of the IEEE 488 standard). Table 1-2 lists interface capabilities as defined in the standard.

Table 1-2

PROGRAMABLE 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
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 switch is set to 1, up.

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 switch is set to 1, up.

Service Request (SR1)

The spectrum analyzer has the complete service request function; asserts SRQ (service request) for the conditions indicated under STATUS BYTE in Section 9 in this manual and reports the corresponding status when polled.

Remote/Local (RL1)

The spectrum analyzer has the complete remote/local function. The front-panel RESET TO LOCAL pushbutton 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 Section 9 of this manual for the effect of busy status on remote/local transitions.

The current value of most programmable functions is maintained when switching from local to remote control. Only the value of TIME/DIV, MIN RF ATTEN dB, and PEAK/AVERAGE may change to match the front-panel control settings when switching from remote to local control, so they won't conflict with local 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 Section 9 of this manual for more on power-up status and for the affect 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.

CONNECTING TO A SYSTEM

The spectrum analyzer can be connected directly to a GPIB system with 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 1-5. 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 earlier in this section for an explanation of each function). The E2 following the functions indicates that three-state drivers are used, rather than open-collector drivers, because of the high-speed operation of the instrument.

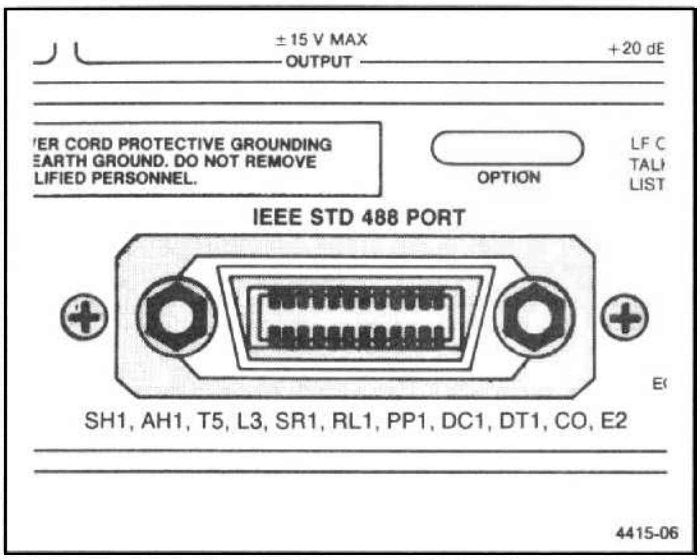

±15 V MAX
OUTPUT
+20 dE
'ER CORD PROTECTIVE GROUNDING
EARTH GROUND. DO NOT REMOVE
LIFIED PERSONNEL.
OPTION
LF C
TALI
LIST
IEEE STD 488 PORT
SH1, AH1, T5, L3, SR1, RL1, PP1, DC1, DT1, CO, E2
4415-06

Figure 1-5. The rear-panel IEEE STD 488 PORT (GPIB).

The GPIB is a flexible system that works either in a star or linear pattern shown in Figure 1-6. Up to 15 devices can be connected at one time. To maintain bus electrical characteristics, no more than one 2-meter cable should be connected for each device (one for the controller, one for the spectrum analyzer, and so on), and at least two-thirds of the devices connected must be on. (Appendix A details the IEEE STD 488 GPIB System Concepts.)

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

The instrument start-up procedure is provided in both the Operators Manual and the Operators Handbook. Refer to those books for instructions on how to begin operating the instrument. Refer to your local Tektronix Field Office or representative for Manual ordering information.

The initial power-on setting of all programmable functions is restored by the INIT command. Refer to Section 9 of this manual for more on this command and a list of the initial power-on settings.

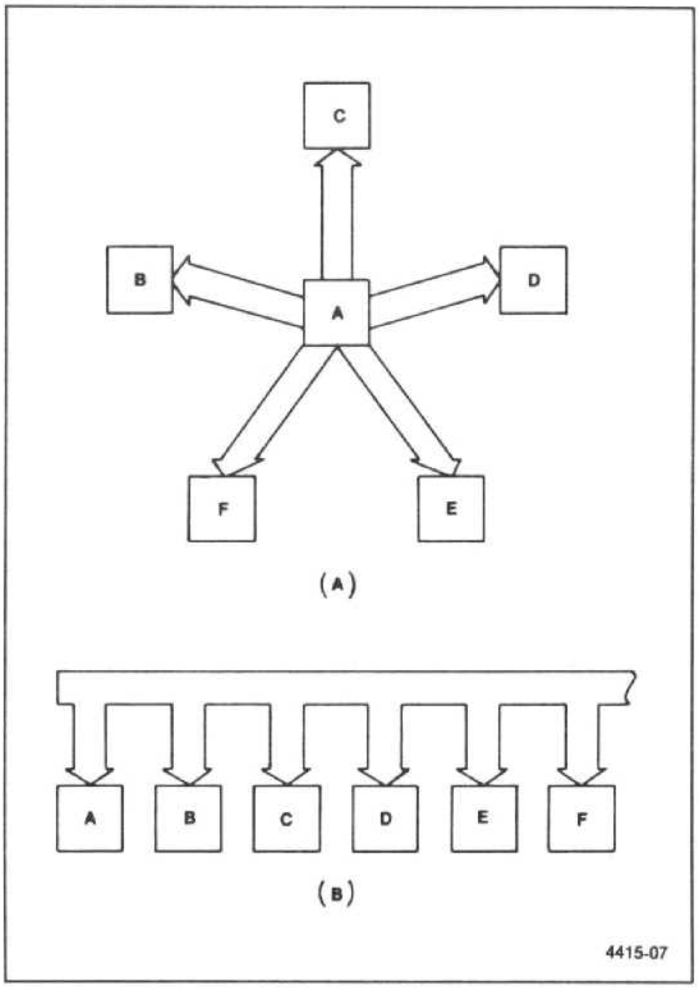


Figure 1-6. The spectrum analyzer can be connected to a GPIB system in either a star (A) or a linear (B) pattern.

DEVICE-DEPENDENT MESSAGE STRUCTURE AND EXECUTION

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.

This goal is achieved within the framework of the Tektronix Interface Standard for GPIB Codes, Formats, Conventions, and Features. This standard is intended to make messages on the bus clear and uncomplicated, while allowing the instrument to handle messages in a friendly manner (i.e., to accept variations in the message). Compatibility with existing devices is maintained as much as possible, while use of codes and data formats is encouraged to make maximum use of bus capabilities.

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 to make it shorter or send the frequency in scientific notation, but this example shows the conversational format of spectrum analyzer messages that makes them readable; therefore, human-oriented.

The spectrum analyzer device-dependent messages in this manual are downward compatible with the Tektronix 49XP-Series, 492AP, 495P, 495P Option 05, and 275XP-Series programmable spectrum analyzers, except as noted later in this section under Spectrum Analyzer Compatibility.

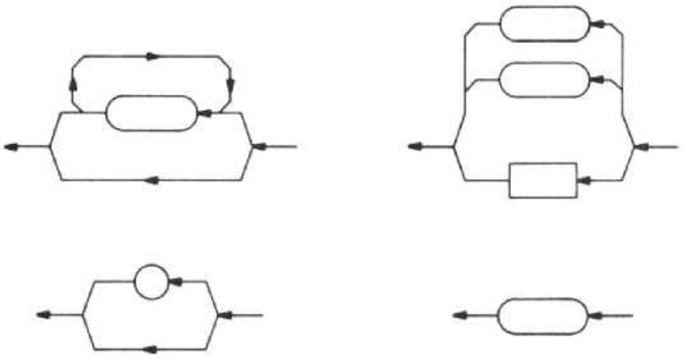
SYNTAX DIAGRAMS

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; i.e., 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 (i.e., the first three characters of the element) are shown larger than optional characters. Although mnemonics are shown all upper case, the spectrum analyzer accepts either upper-case or lower-case ASCII characters. Query response characters are 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 (i.e., 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). Some examples of such sequences follow.

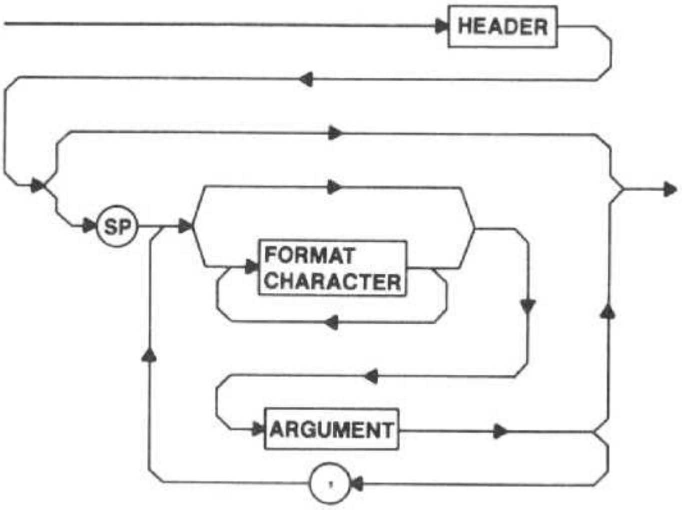


SPECTRUM ANALYZER INPUT MESSAGES

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.



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 by the rear-panel GPIB ADDRESS switch 3.

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 (i.e., 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

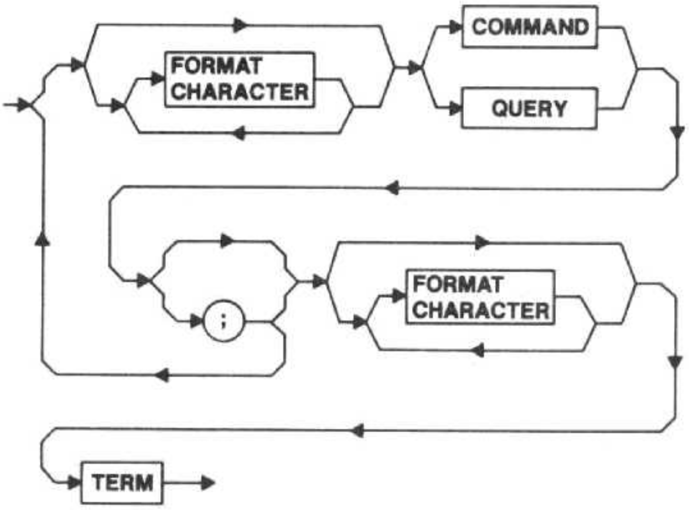
The spectrum analyzer buffers each message it receives with a capacity that exceeds that required for the SET? response. The spectrum analyzer waits until the end of the message to decode and execute it. A command error in any part of a message prevents its execution. When the instrument is under local control, commands that would conflict with local control are ignored (see Remote/Local under IEEE 488 Functions in Section 1 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 Section 9 in 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 discussed in the Interaction part of the CURVE command in Section 7, under Display Data Point Commands Interaction in Section 8, and is further expanded on under Multiple Use of Display Buffer For Waveform Processing and I/O in Section 10, all in this manual.

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.



Because the general command format for display data transfers is complicated, it is omitted; see the data I/O commands in Section 7 of this manual for the specific command syntax.

Header

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

Header Delimiter (SP)

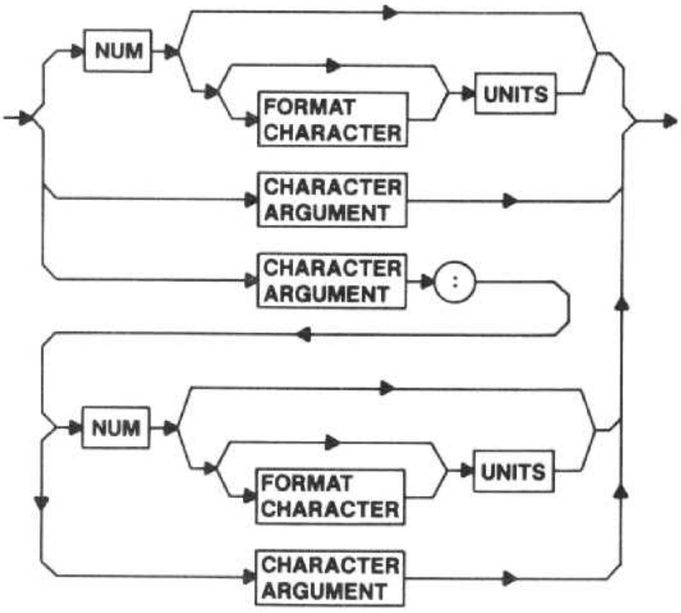
A space (SP) must separate the header from any argument(s).

Argument Delimiter (,)

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

Argument Format

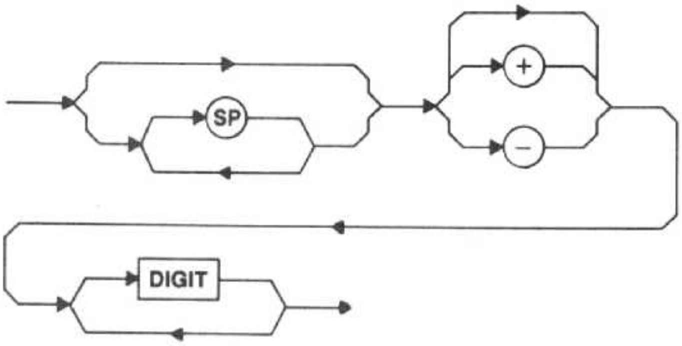
The following diagram illustrates that arguments following the header may be a number, a group of characters, or either a number or a group of characters linked to another argument.



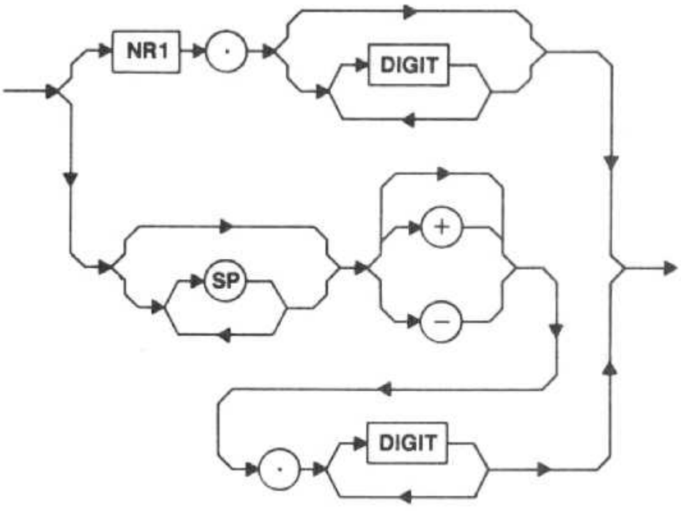
Numbers

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

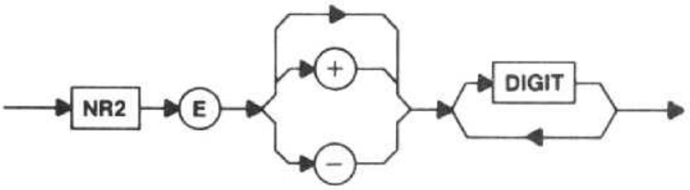
NR1 is an integer (no decimal point).



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 POINT in Section 4 in this manual 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.

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 (i.e., SEC or HZ) does not contribute to the value of the command argument and can be omitted. This does not apply to the dBm and dBmV units in use with the RLUNITS, REFLVL, and MAXPWR commands, where all letters must be used to avoid an 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, or 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 pushbuttons like VIEW B.

Link Argument

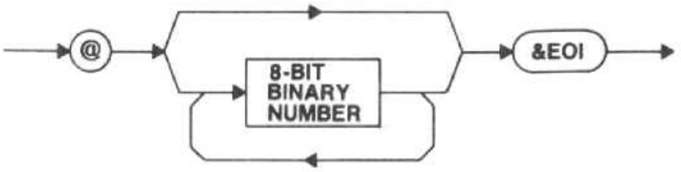
The bottom path in the argument diagram combines both character and number arguments in a link argument. The link is the colon (:), which delimits the first and second arguments. For example, the VRTDSP (vertical display) command employs link arguments to make scale factors available.

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.

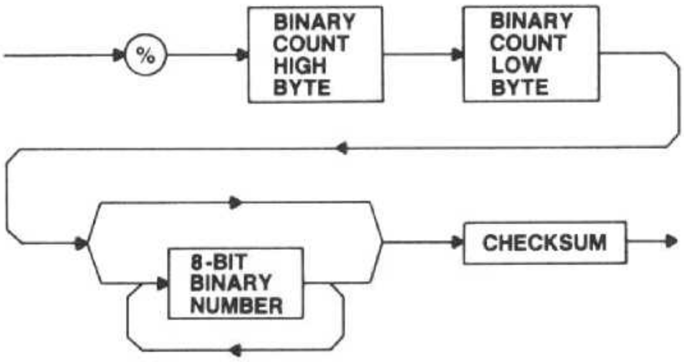
End Block

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.



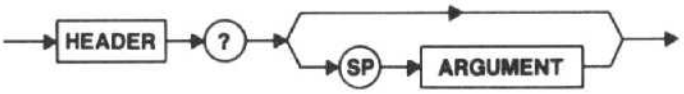
Binary Block

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.



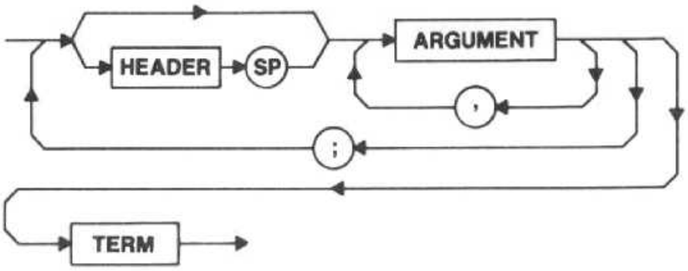
Query Format

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 setting of the EOI OR LF switch.

SPECTRUM ANALYZER COMPATIBILITY

Most of the primary modes of the controls and functions of the Tektronix 49XP-Series portable spectrum analyzers and 275XP-Series laboratory/benchtop spectrum analyzers are identical. (For purposes of feature comparison, here is some useful information: the 492AP is comparable to the 2754 and the 2755; the 495P is comparable to the 2753; and the 494AP is comparable to the 2756.) Following are some of the areas where operations or results will differ.

GPIB

494P, 492AP, 495P, 495P Option 05, 494AP — The DCL interface message is handled by interrupts and will stop execution of the command in progress.

492P, 496P — The processor is required to not be busy (e.g., executing a WAIT message) in order for DCL to be handshook in.

DEGAUS Command

494P, 492AP, 494AP — DEGAUS may be executed in any span.

492P — DEGAUS may be executed in spans 1 MHz/div or less.

IDENT Command

494P, 492AP, 494AP — The span must be ⩽50 kHz/div for coaxial bands (0–21 GHz) or ⩽50 MHz/div for waveguide bands.

492P — The span must be at 500 kHz/div.

495P, 495P Option 05 — The span must be ⩽50 kHz/div.

Readout Maximum

494P, 492AP, 495P, 495P Option 05, 494AP — Readout strings can contain up to 40 characters.

492P, 496P — Readout strings can contain up to 32 characters.

Service Requests

494P, 492AP, 495P, 495P Option 05, 494AP — RQS is the master mask for service requests, and both RQS and EOS must be on to cause end-of-sweep service requests.

492P, 496P — RQS masks error service requests and EOS masks end-of-sweep service requests. Only EOS must be on for end-of-sweep service requests.

Affect of Busy on Device-Dependent Messages

494P, 492AP, 495P, 495P Option 05, 494AP — Interface messages are processed despite busy status. If RTL interrupts a message, the programmable spectrum analyzer executes the remainder of the message before restoring local control. The response of the spectrum analyzer to interface messages depends on the manner in which they are handled. Some interface messages are handled by the GPIB interface, while others require action by the microcomputer. The latter generally involve the GPIB address, and are implemented in firmware rather than on the interface. The speed with which these commands can be handshaked depends on how fast the spectrum analyzer can service the resulting interrupt.

492P, 496P — Interface messages are processed despite busy status if the busy status occurs because the spectrum analyzer is executing a WAIT command. If RTL interrupts WAIT, the spectrum analyzer attempts to execute the remainder of the message after restoring local control and waiting for EOS.

If the busy status occurs because the spectrum analyzer is executing any device-dependent message other then WAIT, the response is handled the same as described for the 495P, 495P Option 05, 494AP, and 492AP.

GET (Group Execute Trigger)

494P, 492AP, 495P, 495P Option 05, 494AP — GET requires firmware action, so handshake occurs only when the interrupt can be handled. The effect of GET is masked by DT (Device Trigger).

492P, 496P — Handshake occurs only when the microcomputer is not executing a device-dependent message unit other than WAIT. GET is not masked.

Reference Level

494P, 492AP, 495P, 495P Option 05, 494AP — The minimum reference level is −117 dBm. The delta-amplitude range is 57.75 dB and slides depending on the reference level when the delta-amplitude mode is entered.

492P, 496P — The minimum reference level is −123 dBm. The delta-amplitude range is 63.75 dB and slides depending on the reference level when the delta-amplitude mode is entered.

RDOUT Command

494P, 492AP, 495P, 495P Option 05, 494AP — The remote-to-local transition will always return RDOUT to NORMAL (i.e., any messages sent to the crt with RDOUT commands will be replaced by the regular crt readout).

492P, 496P — If the remote-to-local transition occurs after UNT or UNL, messages sent to the crt via RDOUT may be retained on the screen. The regular crt readout will be returned by changing any control whose current condition is reported on the crt.

Compatibility-Only Commands

Most of the commands in the 494AP are compatible with the other Tektronix Spectrum Analyzer's in the 49X-series. This allows you to use the programs you created for use with other 49X-series instruments. There are two commands that have no effect on the operation of the 494AP, and their descriptions have not been included in this Programmers manual. These are FRCAL and PHSLK. Even though they have no effect, they will be accepted by the 494A.

Since phase lock cannot be turned off in the 494AP, the PHSLK command has no effect. However, if PHSLK OFF is sent while the instrument is phase locked or PHSLK ON is sent while the instrument is not phase locked, execution error message 48 will be issued. The PHSLK query will still return the present phase lock status.

GETTING STARTED

Getting started with the spectrum analyzer on the GPIB is a simple matter if you are already familiar with a GPIB controller. If not, talking to the spectrum analyzer over the bus may be the easiest way to get over any uncertainty you feel about getting started.

Refer to the Macros section for information on preparing programs that can be stored in the spectrum analyzer memory to be used without a controller.

The spectrum analyzer speaks a friendly language that includes codes for easier human understanding (mnemonics) for control of the front panel and to transfer measurement data. Put these mnemonics into GPIB input/output statements in your controller's language and you're on your way. Of course, your controller must handle details such as asserting REN, unaddressing bus devices, and addressing the spectrum analyzer to start communication; but, these are steps taken by most controllers when executing a GPIB I/O statement.

We have included some sample programs and exercises adapted for the TEKTRONIX 4041 System Controller.

Notes on the Program Examples

Many combinations of languages and controllers may be used for programming the spectrum analyzer over the GPIB. The examples in this manual are given in the BASIC language; specifically, in the version of BASIC used by the Tektronix 4041 System Controller. Other versions of BASIC will probably have similar syntax.

Some controllers may require additional statements to set up data transfer over the GPIB. For example, 4041 BASIC contains optional OPEN and SET DRIVER statements that set GPIB communications parameters. These statements are not included in the program examples because the default parameter values are suitable for simple programs, such as the examples. The statements may be helpful in writing programs that involve multiple I/O channels, such as tape files. Refer to the manual of the controller you are using for the statements needed in your application.

In these examples, the spectrum analyzer's primary address is assumed to be 1. See Section 1 of this manual for instructions on how to set the GPIB address switches.

Some of the lines in the examples extend beyond the column width limitations. When this happens the line is broken where a natural space occurs, and the remainder of the line is indented on the immediately following line. Because of this natural space, a space must be added between the two example lines when joining them to form one program line.

SETTING AND QUERYING PROGRAMMABLE CONTROLS

SETTING PROGRAMMABLE CONTROLS

We can keep this simple, because the spectrum analyzer lets you make complex spectrum measurements semi-automatically. First we'll use the front-panel pushbuttons, then perform the same measurement under remote (GPIB) control using the 4041 controller.

Front-Panel Operation — Many measurements can be made with just three front-panel settings: <blue-SHIFT> FREQ, <blue-SHIFT> SPAN/DIV, and <blue-SHIFT> REF LEVEL.

The <blue-SHIFT> FREQ setting changes the center frequency position of the spectrum window you are viewing, tuning the spectrum analyzer to change the frequency at the center of the crt.

The <blue-SHIFT> SPAN/DIV setting changes the size (width) of the window, setting the frequency span of the crt horizontal axis.

The <blue-SHIFT> REF LEVEL setting raises or lowers the window, which sets the amplitude level of the top graticule line on the crt.

Here's how to operate the spectrum analyzer to measure the CAL OUT signal, using the front-panel pushbuttons for these three settings.

1. Press the <blue-SHIFT> FREQ 1 0 0 MHZ pushbuttons to center the CAL OUT signal.

2. Span down to look more closely at the signal by pressing the <blue-SHIFT> SPAN/DIV 1 MHZ pushbuttons.

The spectrum analyzer automatically picks resolution bandwidth and time/division to fit the new span/division, unless Auto Resolution and Time Auto are cancelled. For most purposes, leave the TIME/DIV control set to AUTO so that Time Auto is in effect in either local or remote control, and leave Auto Resolution selected.

3. Press the <SHIFT> REF LEVEL 2 0 −dBM pushbuttons to set the signal to the reference level.

The spectrum analyzer automatically selects the appropriate input attenuation and IF gain for a reference level at the power level of the CAL OUT signal's fundamental frequency. The spectrum analyzer takes into account the MIN RF ATTEN dB and MIN NOISE settings when positioning the attenuation and gain.

The spectrum analyzer powers up with the automatic modes active and in MAX SPAN to display a complete frequency band. You can restore this condition at any time with the <blue-SHIFT> RESET pushbutton sequence.

4041 Controller — How do steps 1, 2, and 3 in the last example work on a Tektronix 4041 controller? The spectrum analyzer commands are inserted in the following GPIB output statement PRINT. Throughout the 4041 BASIC examples in this manual, the variable Z has been used to represent the spectrum analyzer GPIB address. A constant in the range of 1 to 30 (matching the spectrum analyzer address) can be assigned to the variable Z.

```
 80    Z=1 ! ADDRESS OF SPECTRUM ANALYZER
         EQUALS 1
100    Print #z:"FREQ 100 MHZ"
110    Print #z:"SPAN 1 MHZ"
120    Print #z:"REFLVL −20 DBM"
```

or

```
 80    Z=1
100    Print #z:"FREQ 100 MHZ;SPAN 1 MHZ;REFLVL
         −20 DBM"
```

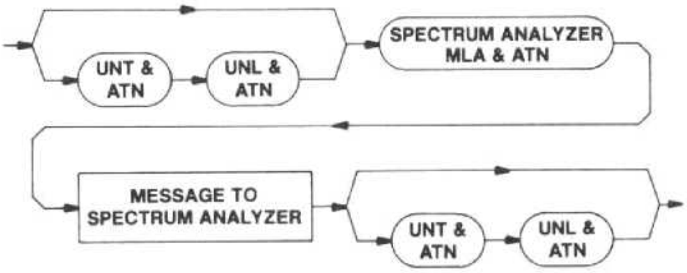
As this last statement shows, all three commands can be strung together, delimited by semicolons.

When the spectrum analyzer executes these commands, it tunes the CAL OUT signal to center screen, changes to the narrower span, and changes the reference level to display the signal peak at the top of the screen. Resolution bandwidth, time/division, input attenuation, and IF gain are changed automatically, as necessary. Because the spectrum analyzer is calibrated for this display as part of the turn-on procedure, the signal peak should occur vertically at the reference level and horizontally at the graticule center. If not, refer to the Initial Turn On procedure in the Operators Manual or Operators Handbook or, better yet, try the automatic calibration routine described in <blue-SHIFT> CAL under Display Parameters in Section 4 of the Operators Manual.

If you receive an SRQ message on the screen of the spectrum analyzer, add an SRQ handler to the program. This sequence can be added to any BASIC program example shown in this manual. The amended program would look like the following.

```
 80    Z=1
 90    On srq then call srq_hndl
100    Enable srq
110    Print #z:"FREQ 100 MHZ;SPAN 1 MHZ;REFLVL
         −20 DBM"
120    End
130 Sub srq_hndl
140    Integer status,adr
150    Poll status,adr;z
160    Print status;" -- ";
170    If status=97 then print "command error"
180    If status=98 then print "execution error"
190    If status=99 then print "internal error"
200    If status=101 then print "execution error warning"
210    If status=102 then print "internal error warning"
220    End
```

Whatever controller is used or statement is sent, the actions shown in the syntax diagram below must be taken to get a message to the spectrum analyzer.



The UNL (unlisten) and UNT (untalk) messages are optional in the previous syntax diagram of bus traffic. However, one or both are sent by most controllers when they begin transmitting and end transmitting on the bus, in order to guarantee a clear communications channel. The controller sends the GPIB address you entered as part of the controller's GPIB I/O statement. The controller either converts it to the spectrum analyzer listen address or expects to receive the listen address with the offset included (i.e., 33). The controller then sends the device-dependent message you inserted into the statement, and may finish by sending UNL and UNT. If the controller does not assert REN (remote enable) automatically for GPIB I/O, you can set it with an earlier control statement. The spectrum analyzer does not balk if REN is not set, unless you send commands that change front-panel settings or data in digital storage.

That leaves the most important part up to you; what goes in the controller statement as a device-dependent message. The spectrum analyzer control mnemonics are collected for quick reference on a Program Summary foldout chart at the back of this manual. For details on how to state each command correctly and the instrument response, turn to the command descriptions that begin in Section 4. The detailed descriptions are arranged by function; (refer to the foldout chart at the back of this manual for page numbers).

The spectrum analyzer executes the message when it sees the message terminator (either EOI or LF). Message syntax and command execution are given fuller treatment in Section 2 of this manual.

Querying Programmable Controls

The spectrum analyzer returns the state of programmable controls when queried. This takes two steps.

1. Query the spectrum analyzer. The query takes the form of the mnemonic for a function name followed by a question mark.

2. Read the response. A GPIB INPUT statement does the job in the case of most controllers. The response will be returned as a character string with or without the applicable header, depending on whether the HDR command is ON or OFF; see the Header information in Section 2 of this manual.

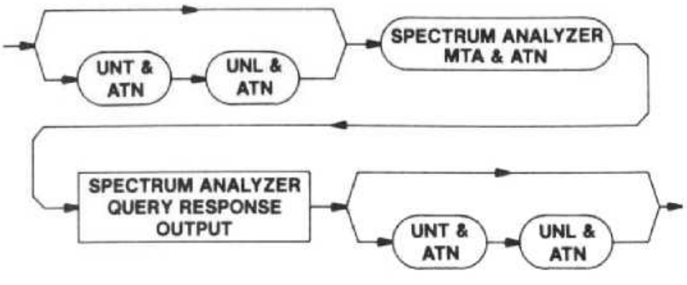
For example, the Auto Resolution mode selected a resolution bandwidth to go with a span of 1 MHz. What is that bandwidth? The query RESBW? readies the spectrum analyzer to output the answer.

The query can be included in any message to the spectrum analyzer. It is executed in its turn. This means that if RESBW? precedes the SPAN command in the previous example, the spectrum analyzer informs you of the old, rather than the new, resolution bandwidth. More than one query can be contained in a message to ask for both resolution bandwidth and, for instance, whether a video filter is on. Just add these queries into the message used in the previous example and combine the message with the controller GPIB input statement.

```
110   Print #z:"FREQ 100 MHZ;SPAN 1 MHZ;REFLVL
      -20 DBM;RESBW?;VIDFLT?"
112   Input #z:p$
114   Print p$
```

If a query that has a lengthy response (e.g., CURVE?, SET?, WFMPRE?) is included as part of this program, character string p$ must be dimensioned large enough to accommodate the message.

Whatever the controller input statement, the actions shown in the syntax diagram below must be taken to receive a message from the spectrum analyzer.


UNT & ATN
UNL & ATN
SPECTRUM ANALYZER MTA & ATN
SPECTRUM ANALYZER QUERY RESPONSE OUTPUT
UNT & ATN
UNL & ATN

The PRINT and INPUT statements above describe the two steps necessary to get output from the spectrum analyzer. The PRINT statement includes the query sent to the spectrum analyzer. The INPUT statement receives the query response. In 4041 BASIC, both steps can be accomplished by a statement such as

```
input prompt "RES?;VID?"#z:p$
```

EXERCISE ROUTINES

Talk/Listen

Now let's put the statements for message I/O together to exercise the spectrum analyzer as a listener and a talker. This routine is handy because it waits for your input and sends it, time after time. If the spectrum analyzer responds with a message, the message is printed before another message is requested from you. Enter any of the commands or queries described in Sections 4 through 9 of this manual.

An SRQ handler is included in the routine to print out any error messages.

The following routine makes use of one of the friendly features of the spectrum analyzer. When the spectrum analyzer is talked with nothing to say, it outputs a byte with all bits set to one and asserts EOI. The routine doesn't have to search the output character string for a ? (a query) and branch to input the response. Instead, the response is read after every message and printed (a blank line if the spectrum analyzer sends a byte with all ones).

The SRQ handler uses another spectrum analyzer feature. Rather than print a code for the status byte, the routine asks for the error that caused the SRQ (ERR?). This offers much more specific information about the problem. The meanings of the error and event codes are listed in Table 9-3 in Section 9 of this manual.

The routine assumes you have assigned the value of the spectrum analyzer address to variable z as previously discussed. It is also assumed your input and output character strings will fit p$ and r$, respectively. This gets further attention with regard to the instrument settings query (SET?), our next topic.

```
200   On srq then call err_hndl
210   Enable srq
220   Print "ENTER MESSAGE: ";
230   Input p$
240   Print #z:p$
250   Input #z:r$
260   Print "   ";r$
270   Goto 220
280   End
300 Sub err_hndl
305   Integer e
310   Input prompt "err?" #z:e
320   Print "   ERROR #";e
330   Resume
340   End
```

Acquiring Instrument Settings with the SET Query

The SET query enables the controller to learn spectrum analyzer settings both for reference and to be able to restore the instrument to those settings. This query readies the instrument to output a message that includes a response for each programmable function.

The format of the response allows it to be used to restore the instrument settings with no operator manipulation required. First, set up for the measurement (and try it) from the spectrum analyzer front panel. Store the message as it is transmitted by the spectrum analyzer using the SET query. Your controller must be ready for a long character string. Dimension a string variable large enough for at least 750 characters for the SET query response, although the exact size depends on the current settings. Then, perform any desired instrument operations. Finally, restore the spectrum analyzer to the original settings by transmitting the stored SET query response back to the instrument (a 4041 program follows that steps you through the operation).

```
300  Rem "This program stores/recalls
     spectrum analyzer front-panel settings"
302   Dim s$ to 750
310   Print "Press <Return> key to store settings";
320   Input k$ ! wait for return
330   Input prompt "set?" #z:s$
335   Print "    Settings stored..."
340   Print "Press <Return> to recall settings to front
      panel";
350   Input k$ ! wait for return
360   Print #z:s$
365   Print "    Settings recalled..."
370   End
```

Line 302 — Dimensions the string variable

Lines 310 through 335 — Inputs a SET? response from the spectrum analyzer.

Lines 340 through 365 — Returns a SET? response to the spectrum analyzer.

Resetting the Spectrum Analyzer and Interface Messages

The INIT command resets the instrument's programmable controls to their initial turn-on condition (see Section 9 in this manual for more on this command). INIT is sent in the same manner as other commands.

Interface message DCL (device clear) clears the instrument I/O buffers and can be used to restart bus communications with the spectrum analyzer. DCL does not interrupt message execution. If the spectrum analyzer is waiting for its talk address so it can execute an output query, output is stopped and the buffers are cleared by DCL (decimal code 20) or any device-dependent input. The decimal codes for other universal commands are 17 for LLO (local lockout), 21 for PPU (parallel poll unconfigure), 63 for UNL (unlisten), and 95 for UNT (untalk).

Addressed commands such as GTL (go to local) can also be sent to the spectrum analyzer. The codes for the addressed commands are 1 for GTL, 4 for SDC (selected device clear), 5 for PPC (parallel poll configure), and 8 for GET (group execute trigger). GET causes the spectrum analyzer to stop the current sweep and immediately start another sweep, synchronizing data acquisition with the interface message.

When the IFC (interface clear) line is asserted by the controller, the spectrum analyzer talker and listener functions are initialized (same effect as UNT and UNL).

Use the WBYTE statement to send the universal commands. For example, use this 4041 statement to send a device clear message on the bus.

```
100   WBYTE DCL
```

For addressed commands, include the primary address of the programmable spectrum analyzer being talked to. For example, this 4041 statement sends a go to local command to the spectrum analyzer at address 3.

```
100   WBYTE GTL(3)
```

Acquiring a Waveform

The waveform in digital storage can be requested as either ASCII-coded decimal numbers or a block of binary data. To keep this simple, let's discuss the ASCII here and cover the binary with the WFMPRE command in Section 8 of this manual. When power is first applied to the spectrum analyzer, it is ready to transmit waveforms in ASCII (the WFMPRE command in Section 8 explains how to change modes).

Here is a program in 4041 BASIC that acquires an ASCII waveform. It gets a Full, 1000-point waveform with A and B memories merged (a power-up condition). Array wfm$ in this program must be dimensioned to 5000.

```
100   Dim wfm$ to 5000
110   Input prompt "cur?" #z:wfm$
```

See Section 10 in this manual for help in plotting the waveform.

Getting Smarter

Signal analysis can be even easier. Put the spectrum analyzer to work to find and measure signals with its internal waveform-processing capabilities. The full set of waveform processing commands is described in Section 8 of this manual, and more instructions for their use are found in Sections 5 and 10 of this manual. To get started "getting smarter", here is a simple application.

The following 4041 program catalogs the first 10 harmonics of the CAL OUT signal. Connect a 100 MHz calibrator before running this program. If the instrument is set to other than the power-up state, precede the program with the INIT command. As in the other programs in this manual, Z is the variable that holds the value of the spectrum analyzer GPIB address.

```
 80    Z=1 ! ADDRESS OF SPECTRUM ANALYZER
 90    Print #z:"INIT"
100 !  CATALOG ROUTINE
110    Print #z:"SPAN 1 MHZ;REFLVL -20
       DBM;VIDFLT NARROW;SIGSWP"
120    For i=1 to 10
130      Print #z:"FREQ ",i*1.0E+8;";SIGSWP;WAIT"
140      Print #z:"FIBIG;TOPSIG;FREQ?;REFLVL?"
150      Input #z:r$
160      Print "SIGNAL ",i,r$
170      Print #z:"REFLVL -20 DBM"
180    Next i
```

Line 110 — Sets span/div and reference level for the start of the signal search, and selects narrow video filter to smooth the data for the routine. The single-sweep mode is selected so new data can be acquired on command.

Line 120 — Starts the loop.

Line 130 — Tunes to a harmonic of the calibrator signal, then starts a sweep to acquire new data. The WAIT guarantees digital storage is filled with updated data before proceeding.

Line 140 — FIBIG finds the calibrator harmonic (it should be the only signal on screen). TOPSIG automatically changes analyzer gain or input attenuation to bring the signal peak to the reference level (top of screen). These and other waveform processing commands allow you to analyze signals without reading in all the display data and operating on it in your controller.

Line 150 — Inputs the analyzer response.

Line 160 — Because the response to each query in line 140 begins with a mnemonic for the function, the analyzer output string acquired in line 150 is intelligible as is and the frequency and reference level readings are printed at the controller.

Line 170 — Readies the spectrum analyzer to do it again.

Line 180 — Goes around again.

The waveform processing commands and query allow you to analyze data without reading waveforms and manipulating them in your controller. More details can be found in Sections 5 and 8 of this manual, and instructions for putting spectrum analyzer waveform processing to work are given in Section 10 of this manual.

Getting Smarter Another Way

The following 4041 program will measure the amplitude and frequency of the 100 MHZ CAL OUT and the next 9 harmonics of the CAL OUT signal.

```
 80   Z=1 ! ADDRESS OF SPECTRUM ANALYZER
500   Dim m(20)
510   Print #z:"init;time auto;min 0;span 1m;ref -20"
520   Print #z:"vid nar;step 100m;sig"
530   Print #z:"pstep;sig;wai;mfbig;mlocat?;rep 9"
540   Input deln "" #z:m
550   Print m
```

Line 500 — Dimensions array m.

Line 510 — Initializes the spectrum analyzer and sets the time/division, minimum attenuation, frequency span, and reference level.

Line 520 — Sets the video filter, establishes a step size of 100 MHz, and enters the single sweep mode.

Line 530 — Increases the primary marker frequency by the step value (100 MHz), starts a sweep, waits until the end of sweep, finds the largest on-screen signal, and requests the marker frequency and marker amplitude, and then repeats the sequence 9 more times.

Line 540 — Inputs all 10 marker frequencies and marker amplitudes.

Line 550 — Prints all 10 marker frequencies and marker amplitudes.

INSTRUMENT CONTROL

Commands and queries for instrument control are grouped in this section according to the following functions (the marker-related commands and queries are in the Marker System section.)

Frequency
Frequency Span and Resolution
Vertical Display and Reference Level
Sweep Control
Digital Storage
Display Control
General Purpose
Marker System Control
Marker Positioning
Marker Finding

The mnemonics (codes) in Table 4-1 correspond to the instrument names for the front-panel pushbuttons and controls and related functions.

Table 4-1

FRONT-PANEL COMMANDS AND QUERIES

Name	Mnemonic
Frequency	
FREQ ENTRY	FREQ
CENTER/MARKER FREQUENCY	TUNE
TUNE CF	TMODE
1ST LO[a]	FIRST
2ND LO[a]	SECOND
Tracking Generator Mode[a]	TGMODE
Sideband Analyzer Mode[a]	SAMODE
Disable Tuning Corrections[a]	DISCOR
Δ Band and ∇ Band	FRQRNG
STEP ENTRY	STEP
−STEP	MSTEP
+STEP	PSTEP
COUNT	COUNT
COUNT RESOLN	CRES
Count→CF[a]	CNTCF
FREQ START STOP	STSTOP
Δ F	DELFR
Degauss[a]	DEGAUS
EXT MIXER	EXMXR
IMPEDANCE[b]	IMPED
Frequency Span and Resolution	
FREQUENCY SPAN/DIV and SPAN/DIV ENTRY	SPAN
ZERO SPAN	ZEROSP
MAX SPAN	MXSPN
RESOLUTION BANDWIDTH	RESBW
AUTO RESOLN	ARES
IDENT	IDENT
Vertical Display and Reference Level	
10dB/DIV, 2dB/DIV, LIN, and dB/DIV ENTRY	VRTDSP
REFERENCE LEVEL and REF LEVEL ENTRY	REFLVL
REF LEVEL UNITS	RLUNIT and ROFSET
CAL	CAL
Enable Calibration Factors[a]	ENCAL
FINE	FINE
MIN NOISE/MIN DISTORTION	RLMODE
Reduced Gain Mode[a]	RGMODE
MANUAL PEAK, AUTO PEAK	PEAK
MIN RF ATTEN dB	MINATT, MAXPWR, and RFATT?
PULSE STRETCHER	PLSTR
WIDE VIDEO FILTER and NARROW VIDEO FILTER	VIDFLT
Sweep Control	
FREE RUN, INT, LINE, and EXT	TRIG
SINGLE SWEEP	SIGSWP
TIME/DIV	TIME
Digital Storage	
VIEW A	AVIEW
VIEW B	BVIEW
SAVE A	SAVEA
B−SAVE A	BMINA
STORE DISP	DSTORE
RECALL	DRECAL
MAX HOLD	MXHLD
PEAK/AVERAGE	CRSOR
Display Control	
READOUT	REDOUT
GRAT ILLUM	GRAT
BASELINE CLIP	CLIP
General Purpose	
HELP?	HELP?
STORE	STORE
RECALL SETTINGS	RECALL
Register Data[a]	RDATA
PLOT?	PLOT?
Plotter Type Selection[a]	PTYPE
Plot (B−A) Reference[a]	POFSET
End of Sweep Corrections[a]	ECR
Register Valid[a]	RVALID
SRQ	SSR

[a] These commands are related to front-panel control functions; they are not actually labeled on the front panel.

[b] Available only on instruments with Option 07 installed.

Table 4-1 (cont)

Name	Mnemonic
Marker System Control	
Δ MKR and MARKER	MARKER
Marker on Trace[a]	MTRACE
ASSIGN 1	M1ASGN
ASSIGN 2	M2ASGN
dB/Hz	NSELVL
SIGNAL TRACK	SGTRAK
Marker Positioning	
Display Pointer to Marker[a]	DPMK
Marker Amplitude[a]	MAMPL?
MKR→CENTER	MCEN
1←MKR→2	MEXCHG
FREQ ENTRY[a]	MFREQ
CENTER/MARKER FREQUENCY	MKTIME
Marker to Display Pointer[a]	MKDP
Marker Location[a]	MLOCAT?
MKR→REF LVL	MTOP
Tune Marker[a]	MTUNE
Marker Finding	
Next Higher Amplitude[a]	HRAMPL
Next Lower Amplitude[a]	LRAMPL
Marker Bandwidth Number[a]	BWNUM
BANDWIDTH	BWMODE
Marker Peak Find[a]	MFBIG
Marker Left Next[a]	MLFTNX
PEAK FIND	PKFIND
Marker to Maximum and Center[a]	PKCEN
Move Marker to Maximum[a]	MMAX
Move Marker to Minimum[a]	MMIN
Marker Right Next[a]	MRGTNX
Marker Threshold[a]	THRHLD
Move Marker Left x dB[a]	MVLFDB
Move Marker Right x dB[a]	MVRTDB
Signal Type[a]	STYPE
Signal Find Error[a]	SGERR

Table 4-1 (cont)

Name	Mnemonic
Miscellaneous	
Zoom[a]	ZOOM
Zero Time[a]	ZETIME

[a] **These commands are related to front-panel control functions; they are not actually labeled on the front panel.**

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

INTENSITY
MANUAL SCAN
POSITION ◄►⬍
AMPL and LOG CAL
POWER
PEAK/AVERAGE cursor (other than fully counterclockwise or clockwise positions)

Use in Macros

Most of the instrument control commands in this section can be incorporated into macros designed to your specific needs. (No queries can be used within macros.) Since there is a total of 8k bytes of memory dedicated for macro use, it is important that you know the number of bytes used for each command, and keep this in mind while preparing macros. This maximum number of bytes used is included with the commands in this section; and there is also a table in the foldout chart at the back of this manual that lists all available instrument commands and the bytes used by each.

NUM Argument Values

Unless otherwise stated, the values for the NUM argument are

- 1 = ON
- ≥+0.5 are rounded to 1
- 0 = OFF
- <+0.5 are rounded to 0

Table 4-2
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/div
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	829	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+	2182–2183	14	1.3 GHZ
8	33–60	2072	3092–5790	10+	2182–2183	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

FREQUENCY

The commands in this group set and change the instrument center frequency (FREQ, TUNE, and STSTOP), set the tuning mode (TMODE), set the 1ST LO (FIRST) and the 2ND LO (SECOND) frequencies, enable the tracking generator (TGMODE), and sideband analyzer (SAMODE) modes, disable tuning corrections (DISCOR), select the frequency range (FRQRNG), turn the counter mode on and off (COUNT), select counter resolution (CRES), transfer signal count to center frequency (CNTCF), set frequency step size (STEP), decrease or increase center frequency (MSTEP, PSTEP), set start-stop frequencies (STSTOP), and start the delta-frequency function (DELFR), apply degaussing current to restore preselector alignment (DEGAUS), and select the EXT MIXER input (EXMXR). Instruments with Option 07 installed change between the 50-Ω and 75-Ω inputs (IMPED).

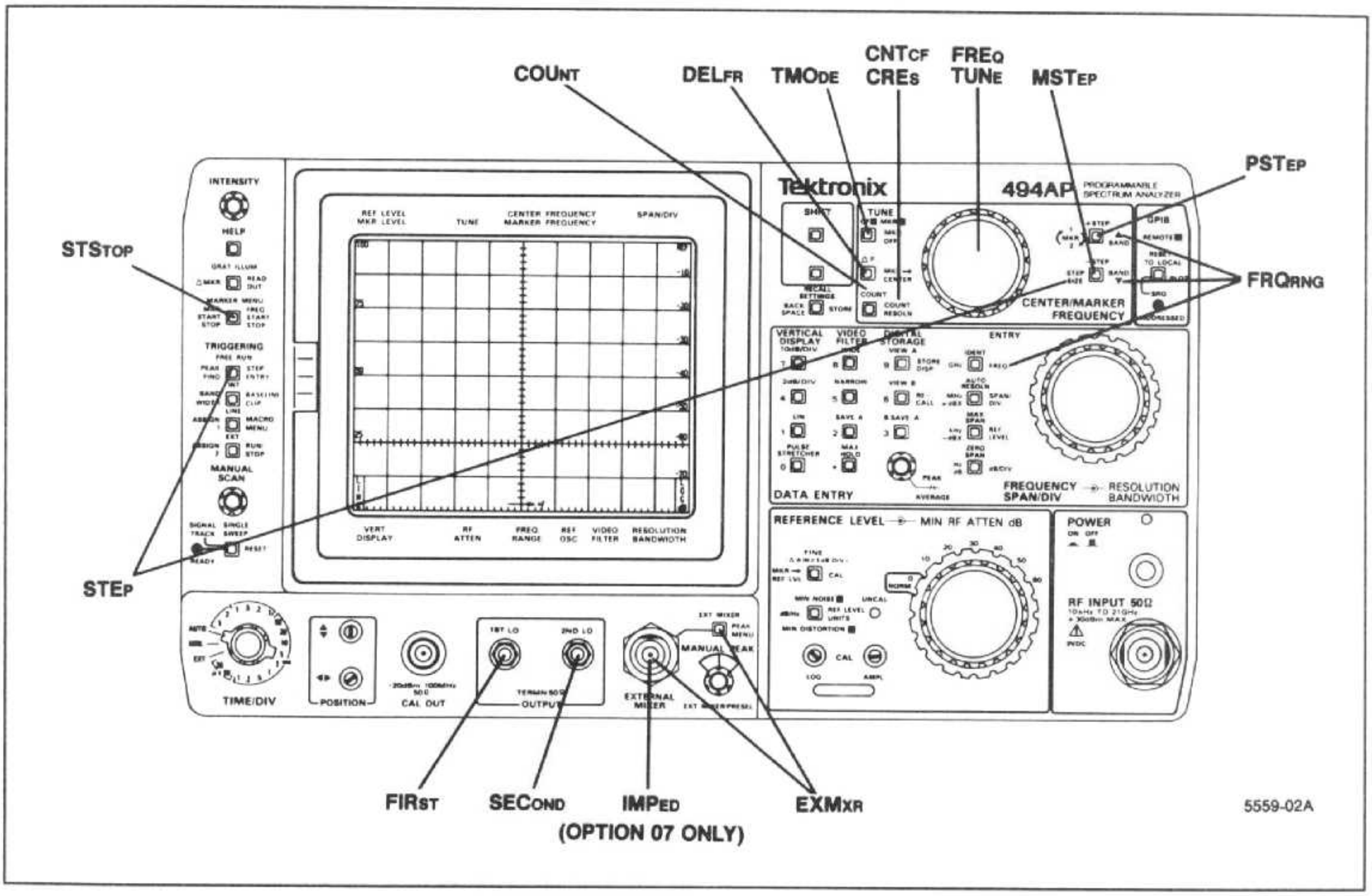
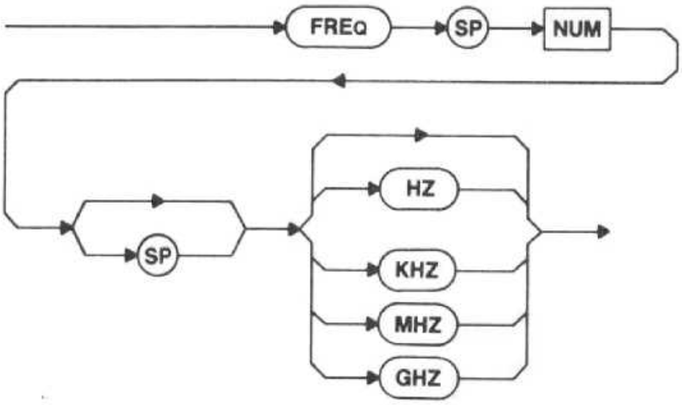


Figure 4-1. Front-panel Frequency commands.

FREQ (center frequency) command



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.

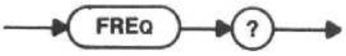
Macro Memory Used — 6 bytes.

Examples—FREQ 200MHZ
FREQ 100000
FRE 200 MHZ

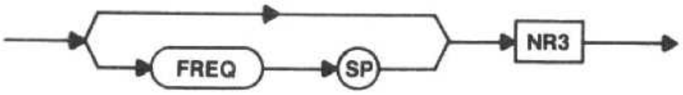
Range — 0 Hz to 325 GHz.

Power-up value — 0 MHz.

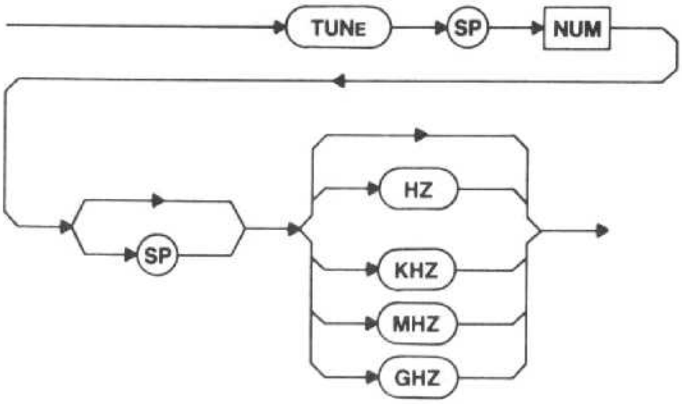
FREQ (center frequency) query



Response to FREQ query



TUNE (incremental frequency change) command



NUM — The instrument changes its center frequency by using the value of the command argument as an offset to its previous center frequency.

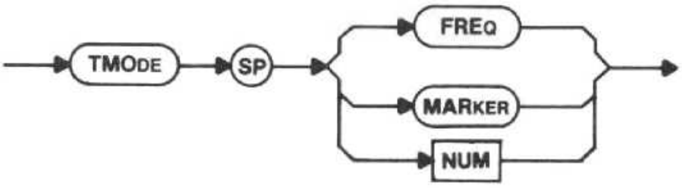
Macro Memory Used — 6 bytes.

Examples—TUN 10 MHZ
TUNE 1.0E6
TUNE 100 KHZ

Range — The frequency resulting from using the TUNE command must be within the frequency range in use. The maximum value for each freuqency range (that is, the value needed to tune from one end of the range to the other) is given in Table 4-2.

There is no TUNE query.

TMODE (set tune mode) command



MARKER — The PSTEP and MSTEP commands will change the marker frequency.

FREQ — The PSTEP and MSTEP commands will change the center frequency.

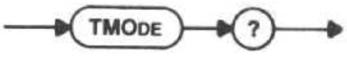
NUM—0 = FREQ
1 = MARKER

Power-up value — FREQ.

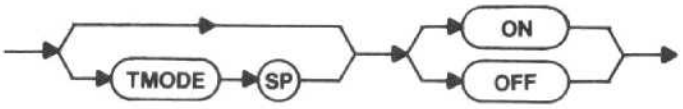
Macro Memory Used — 2 bytes.

Interaction — If MARKER is OFF, TMODE MARKER sets MARKER to SINGLE. TMODE sets the tuning mode that the front-panel CENTER/MARKER FREQUENCY knob will have when the analyzer is returned to local control.

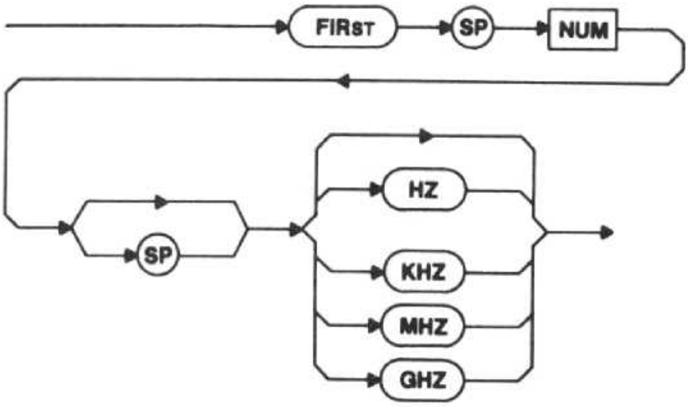
TMODE (tune mode) query



Response to TMODE query



FIRST (1ST LO frequency) command



NUM — The instrument 1ST LO is set to the requested frequency. The resulting center frequency will be displayed.

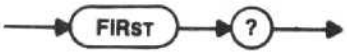
Macro Memory Used — 6 bytes.

Examples — FIR 2.8 GHZ
FIRST 2.8 GHZ

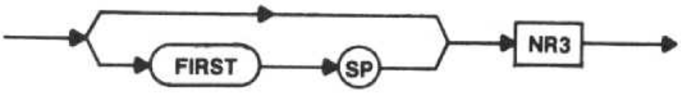
Range — Refer to Table 4-2.

Power-up value — 2072 MHz.

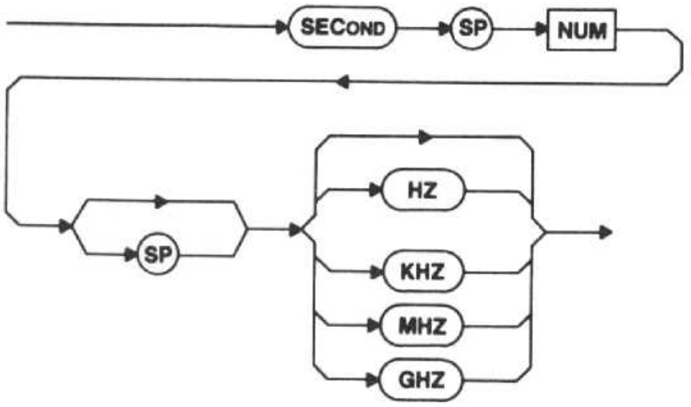
FIRST (1ST LO frequency) query



Response to FIRST query



SECOND (2ND LO frequency) command



NUM — The instrument 2ND LO is set to the requested frequency. The resulting center frequency will be displayed.

Macro Memory Used — 6 bytes.

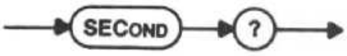
Example — SECOND 2182 MHZ

Range — Bands 1 and 5–12 is 2181–2183 MHz
Bands 2–4 is 718–720 MHz

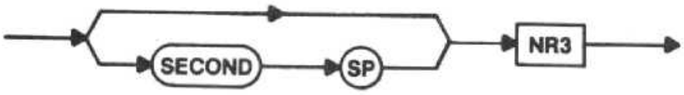
Range — Refer to Table 4-2.

Power-up value — 2182 MHz.

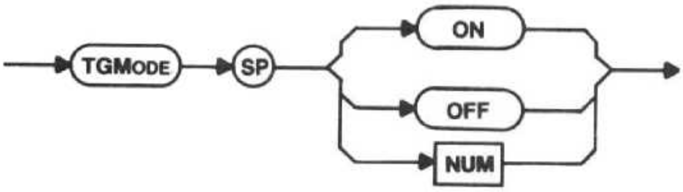
SECOND (2nd LO frequency) query



Response to SECOND query



TGMODE (tracking generator mode) command



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.

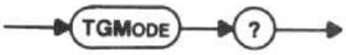
ON — The tracking generator mode is turned on.

OFF — The tracking generator mode is turned off.

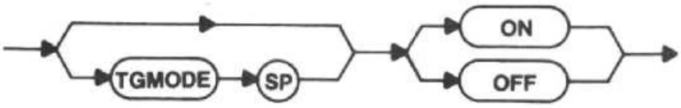
Macro Memory Used — 2 bytes.

Power-up Value — OFF.

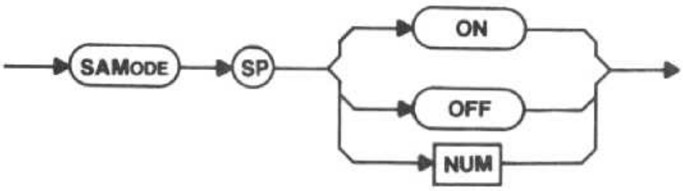
TGMODE (tracking generator mode) query



Response to TGMODE query



SAMODE (sideband analyzer mode) command



SAMODE is active in Band 1 only. When the SAMODE command is ON, the spectrum analyzer phase locks in 50 kHz/div instead of the normal 200 kHz/div. This extends the usefulness of the 1405 Sideband Analyzer, which uses only the first local oscillator of the spectrum analyzer.

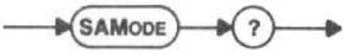
ON — The sideband analyzer mode is turned on.

OFF — The sideband analyzer mode is turned off.

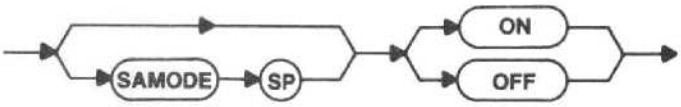
Macro Memory Used — 2 bytes.

Power-up Value — OFF.

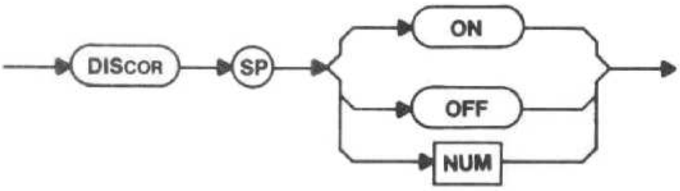
SAMODE (sideband analyzer mode) query



Response to SAMODE query



DISCOR (disable tuning corrections) command



This command is included to allow disabling of the frequency control loop in the instrument for speed or diagnostics purposes. It will also allow a fallback to low accuracy center frequency operation if the frequency control loop fails.

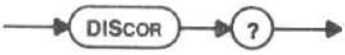
ON — Center frequency corrections are disabled.

OFF — Center frequency corrections are enabled.

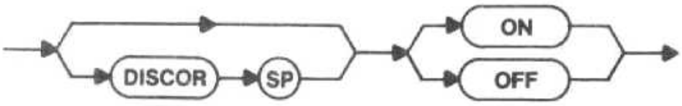
Macro Memory Used — 2 bytes.

Power-up value — OFF.

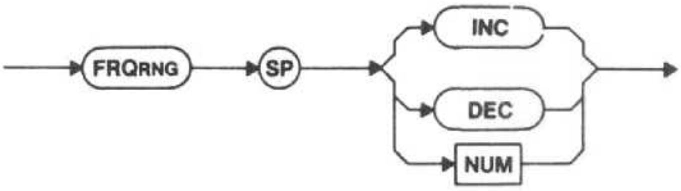
DISCOR (disable tuning corrections) query



Response to DISCOR query



FRQRNG (frequency range) command



NUM — The instrument accepts number arguments in the range of 1 through 12 and changes the frequency range accordingly. Non-integer values are rounded. If the number is too large or too small, the programmable spectrum analyzer maintains its current frequency range and reports execution error message 29.

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

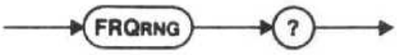
DEC — The instrument changes to the next lower frequency range, if possible.

Macro Memory Used — 2 bytes.

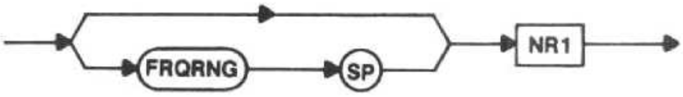
Power-up value — Frequency Range 1.

Interaction — 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.

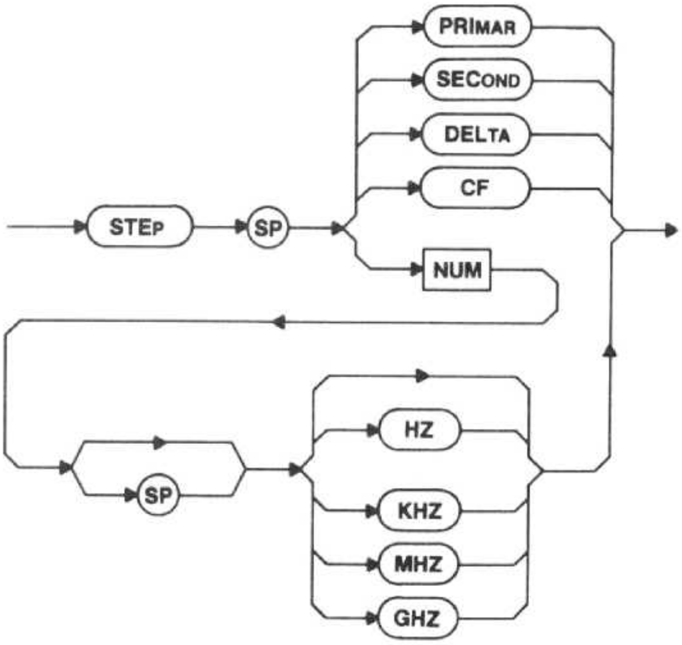
FRQRNG (frequency range) query



Response to FRQRNG query



STEP (step size) command



The STEP command sets the frequency step size used by the MSTEP and PSTEP commands.

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 input.

Macro Memory Used — 6 bytes.

Examples—STE DEL
STE PRIMAR
STEP SEC
STE 20 KHZ
STEP 100 MHZ

Range — Set STEP anywhere within the internal mixer band range. In the waveguide mixer bands, STEP can be used within a single band only. The largest acceptable value is 155 GHz (the width of band 12). If a larger value is entered, STEP will be set to 155 GHz.

Power-up value — See Interaction that follows.

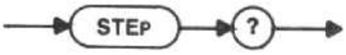
Interaction — STEP DELTA and STEP SECOND cause marker execution error message 123 to be issued if MARKER is not set to DELTA. If the frequency or delta to which the step size is to be set is larger than 155 GHz, the step size will be set to 155 GHz. If STEP has not been set, MSTEP and PSTEP set STEP to the following values.

- The absolute value of the delta frequency is put into step size if delta markers are on.
- The marker frequency is put into the step size if delta markers are off and the Tune Marker Mode is on.

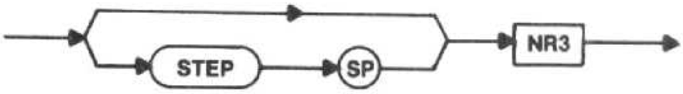
Otherwise

- The center frequency is put into the step size if delta markers are off and the Tune CF Mode is on.

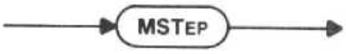
STEP (step size) query



Response to STEP query



MSTEP (minus step) command



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 — See Interaction that follows.

Interaction — 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. If STEP has not been set, MSTEP will set STEP to the following values.

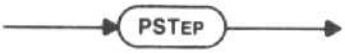
- The absolute value of the delta frequency is put into step size if delta markers are on.
- The marker frequency is put into the step size if delta markers are off and the Tune Marker Mode is on.

Otherwise

- The center frequency is put into the step size if delta markers are off and the Tune CF Mode is on.

There is no MSTEP query.

PSTEP (plus step) command



The PSTEP 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 — See Interaction that follows.

Interaction — 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. If STEP has not been set, PSTEP will set STEP to the following values.

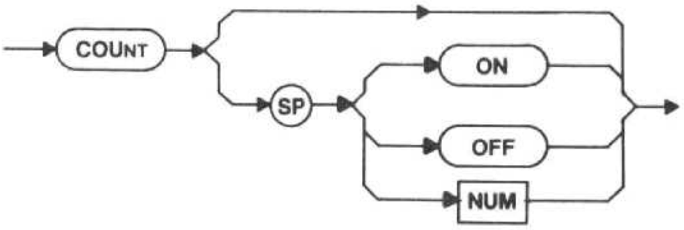
- The absolute value of the delta frequency is put into step size if delta markers are on.
- The marker frequency is put into the step size if delta markers are off and the Tune Marker Mode is on.

Otherwise

- The center frequency is put into the step size if delta markers are off and the Tune CF Mode is on.

There is no PSTEP query.

COUNT (counter) command



ON — The counter mode is turned on.

OFF — The counter mode is turned off.

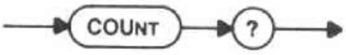
When no argument is included, an immediate signal count occurs whether or not the counter mode is on; no change occurs in the ON/OFF status of the counter mode.

Macro Memory Used — 2 bytes.

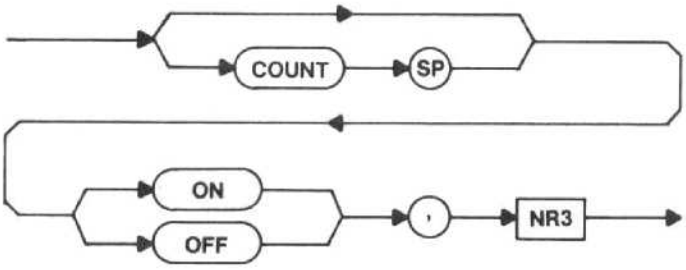
Power-up value — OFF.

Interaction — If a marker(s) is on a non-saved 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.

COUNT (counter) query

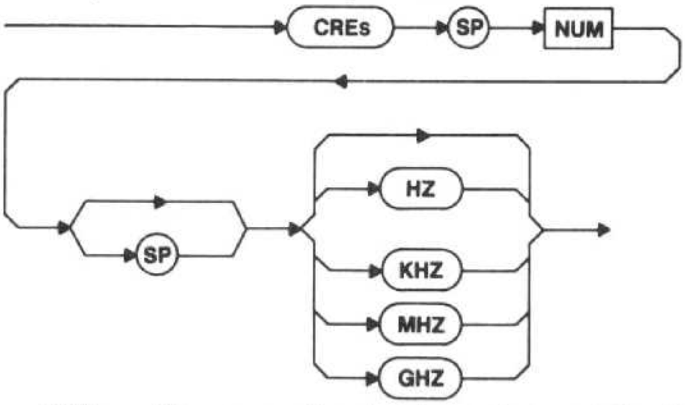


Response to COUNT query



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.

CRES (counter resolution) command



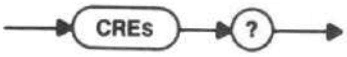
NUM — The proper decade of counter resolution is selected for use. Numbers that are not powers of ten will be set to the next lower power of ten, up to a maximum of 1 GHz resolution. If the number is out of range, execution error message 30 will be issued.

Marker Memory Used — 4 bytes.

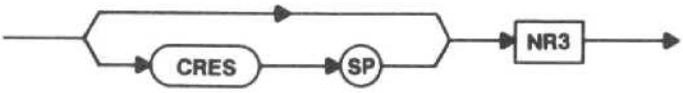
Range — 0–1 GHz.

Power-up value — 1 Hz.

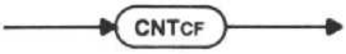
CRES (counter resolution) query



Response to CRES query



CNTCF (count to center frequency) command



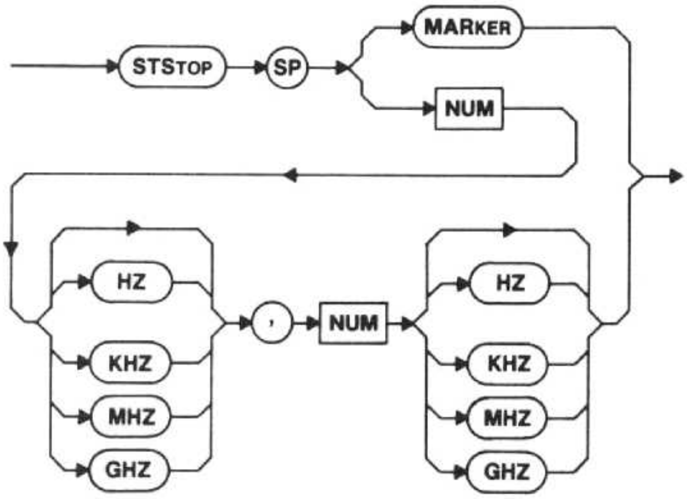
A count of the signal is taken, then this signal count result is transferred to the center frequency. This tunes the spectrum analyzer to the signal counted. Accuracy is limited by the count resolution in use when the signal count is done.

Marker Memory Used — 1 byte.

Interaction — CNTCF is not available in MXSPN or ZEROSP.

There is no CNTCF query.

STSTOP (start-stop sweep) command



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, NUM — The starting frequency of the display is set to the first NUM and 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.

Macro Memory Used — 12 bytes.

Examples — STS MARKER
STSTOP 10MHZ,130MHZ
STS 100000HZ,66MHZ

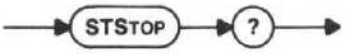
Range — Bands 1 and 6–12 — Band limits of the band the start and stop frequencies were set in, not the band the instrument is currently in.

Bands 2–5 — Any value within the bands

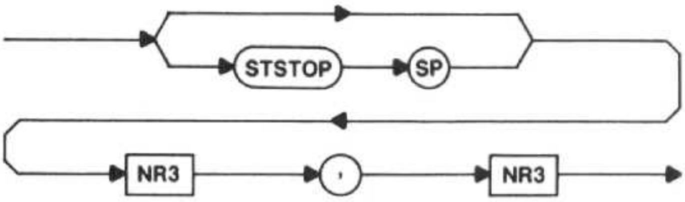
Power-up value — Start frequency, 0 Hz; stop frequency, 1.8 GHz.

Interaction — Marker execution error message 123 is issued if the STSTOP MARKER command is given when MARKER is not set to DELTA.

STSTOP (start-stop sweep) query

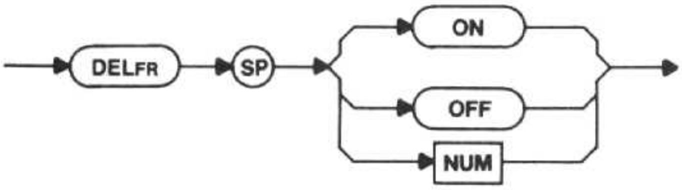


Response to STSTOP query



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.

DELFR (delta-frequency) command



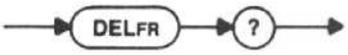
ON — The delta-frequency function is turned on. As the frequency is changed, the crt center frequency readout indicates relative frequency rather than absolute frequency. Only the readout operates differently; FREQ and FREQ? response still refer to absolute frequency. The resolution of the readout will be the lesser of the current readout resolution and the readout resolution when DELFR was turned on.

OFF — The delta-frequency function is turned off.

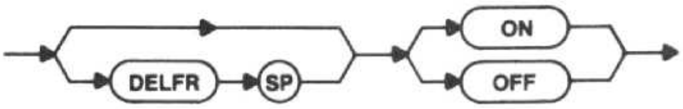
Macro Memory Used — 2 bytes.

Power-up value — OFF.

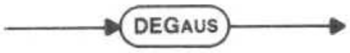
DELFR (delta-frequency) query



Response to DELFR query



DEGAUS (degauss tuning coils) command

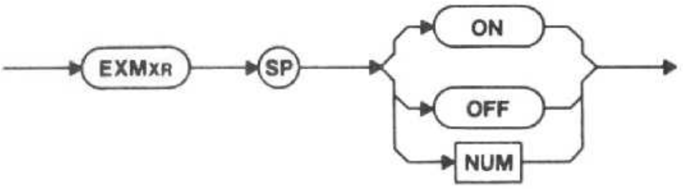


A current is momentarily turned off to remove residual magnetism in the 1st LO and preselector.

Macro Memory Used — 1 byte.

There is no DEGAUS query.

EXMXR (external mixer input) command



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

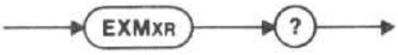
OFF — The coax RF INPUT is selected.

Macro Memory Used — 2 bytes.

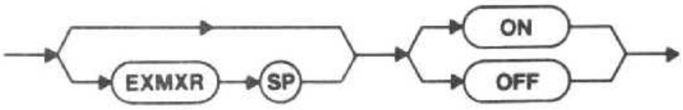
Power-up value — OFF.

Interaction — 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.

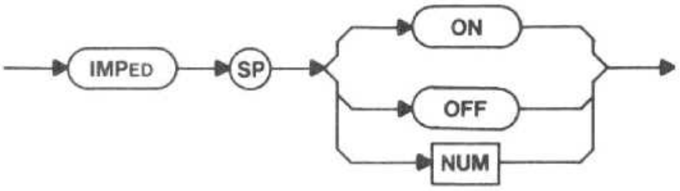
EXMXR (external mixer input) query



Response to EXMXR query



IMPED (impedance) command (Option 07 only)



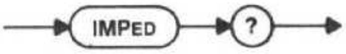
ON — The 75 Ω input is used. If Option 07 is not installed, command error message 8 will be issued.

OFF — The 50 Ω input is used.

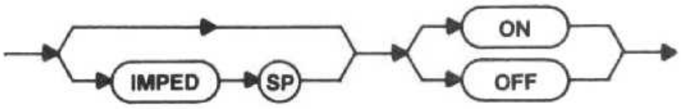
Macro Memory Used — 2 bytes.

Power-up value — OFF.

IMPED (impedance) query (Option 07 only)



Response to IMPED query (Option 07 only)



FREQUENCY SPAN AND RESOLUTION

The commands in this group control the frequency span (SPAN), the zero span mode (ZEROSP), the max span mode (MXSPN) and the resolution (RESBW and ARES) of the display. Also, true signals can be distinguished from spurious frequency conversion products (IDENT).

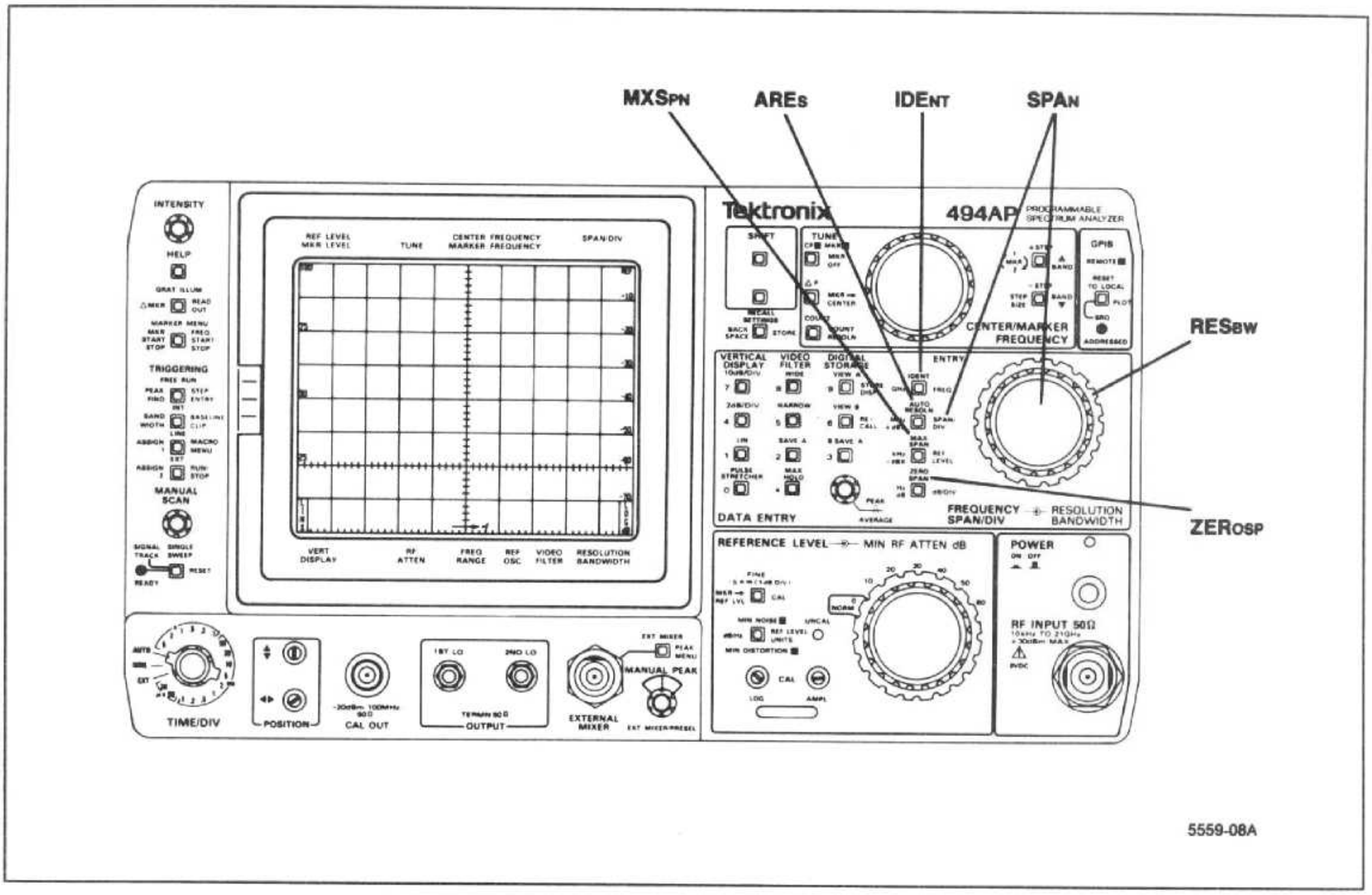
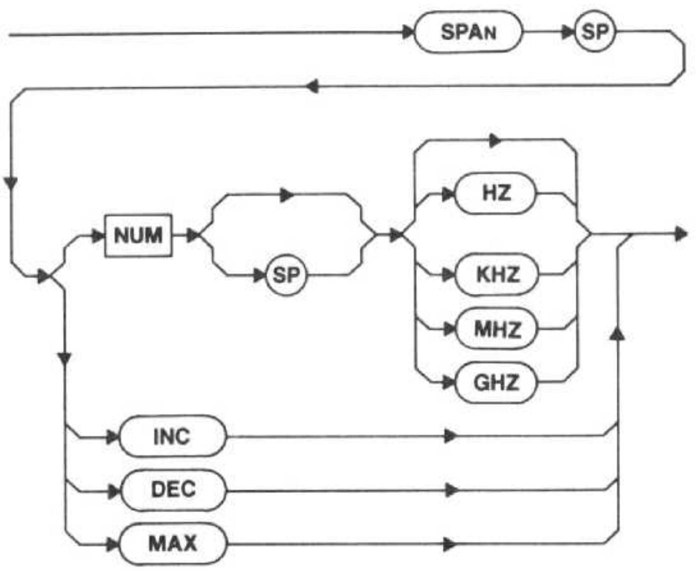


Figure 4-2. Front-panel Frequency Span and Resolution commands.

SPAN (frequency span/division) command



NUM — The span/division is selected. The value of the argument is rounded to two significant digits. 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/division is selected in the front-panel 1-2-5 sequence, if possible.

DEC — The next smaller span/division is selected in the front-panel 1-2-5 sequence, if possible.

MAX — The entire frequency range in use is swept.

Macro Memory Used — 5 bytes.

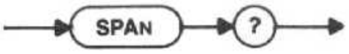
Examples — SPA 200
SPAN 50KHZ
SPA 100 MHZ
SPAN DEC

Range — Minimum value 10 Hz; maximum value for direct entry given in Table 4-2. The maximum vale for INC or DEC is the 1-2-5 step that is less-than-or-equal-to the value in Table 4-2. 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.

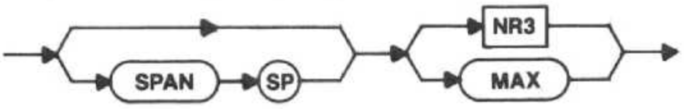
Power-up value — Maximum.

Interaction — Changing the SPAN setting turns ZEROSP OFF and MXSPN OFF.

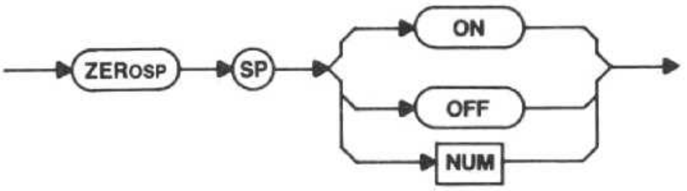
SPAN (frequency span/division) query



Response to SPAN query



ZEROSP (zero-span mode) command



ON — The instrument is converted to a time domain mode with the frequency sweep defeated. Crt readout shifts to the TIME/DIV mode on the horizontal axis instead of FREQ SPAN/DIV. The previous FREQ SPAN/DIV is saved, and it is restored when ZEROSP is turned OFF.

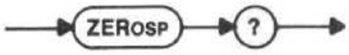
OFF — ZEROSP is cancelled, leaving the FREQ SPAN/DIV at the value previously selected.

Macro Memory Used — 2 bytes.

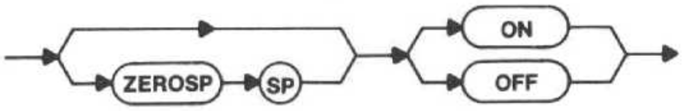
Power-up value — Off.

Interaction — Changing the SPAN setting turns ZEROSP OFF. PKCEN, MCEN, and CNTCF are not available in ZEROSP.

ZEROSP (zero span mode) query



Response to ZEROSP query

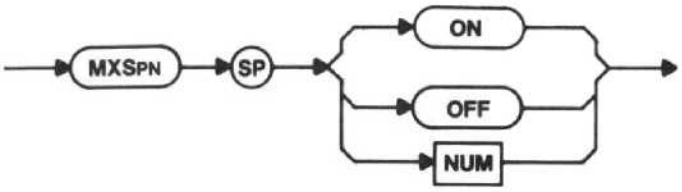


The response is the current zero span condition.

NOTE

It may be preferable to use ZEROSP rather than SPAN 0. When ZEROSP is ON, the front-panel ZERO SPAN indicator is lit so you have a positive indication that the zero span mode is set, in addition to the crt readout. When ZEROSP is turned off, the previous SPAN/DIV setting is restored.

MXSPN (max-span mode) command



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/DIV is saved, and it is restored when MXSPN is OFF.

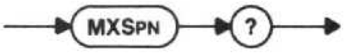
OFF — MXSPN is cancelled, leaving the FREQ SPAN/DIV at the value previously selected.

Macro Memory Used — 2 bytes.

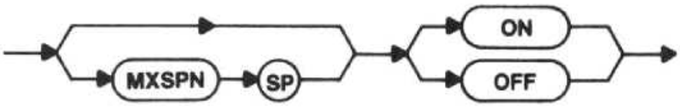
Power-up value — Off.

Interaction — Changing SPAN setting turns MXSPN OFF. PKCEN, MCEN, and CNTCF are not available in MXSPN.

MXSPN (max-span mode) query



Response to MXSPN query

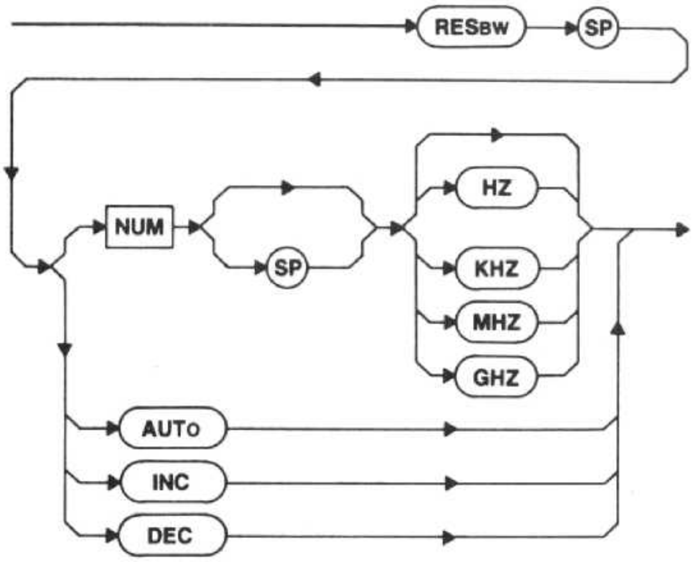


The response is the current MXSPN condition.

NOTE

It may be preferable to use MXSPN rather than SPAN MAX. When MXSPN is ON, the front-panel MAX SPAN indicator is lit so you have a positive indication that the maximum span mode is set, in addition to the crt readout. When MXSPN is turned off, the previous FREQUENCY SPAN/DIV setting is restored.

RESBW (resolution bandwidth) command



NUM — The nearest available resolution bandwidth 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 4-3 (execution error message 32 is issued if the argument is beyond the normal range).

Table 4-3

RESOLUTION BANDWIDTH SELECTION

Value[a]	Selects
3.17 Hz–31.6 Hz	10 Hz
31.7 Hz–316 Hz	100 Hz
317 Hz–3.16 kHz	1 kHz
3.17 kHz–31.6 kHz	10 kHz
31.7 kHz–316 kHz	100 kHz (Non Option 07)
31.7 kHz–316 kHz	300 kHz (Option 07 Only)
317 kHz–1.72 MHz	1 MHz
1.73 MHz–5.49 MHz	3 MHz

[a] **Values outside the ranges listed cause execution error message 32 to be issued.**

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).

Macro Memory Used — 5 bytes.

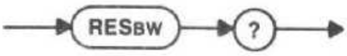
Examples—RES 100
RESBW 1KHZ
RES 1.5 MHZ
RESBW INC

Range — See Table 4-2.

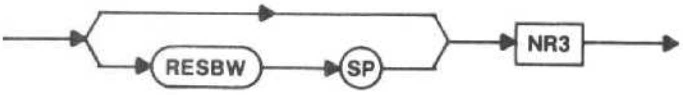
Power-up value — 3 MHz.

Interaction — Any argument except AUTO cancels ARES ON. Reducing resolution bandwidth may require a slower sweep speed (TIME) to maintain a calibrated display unless TIME/DIV is set to AUTO.

RESBW (resolution bandwidth) query

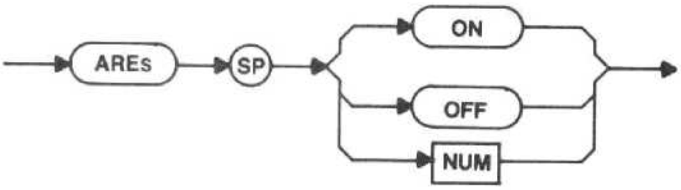


Response to RESBW query



The response to the SET query includes the AUTO argument (see SET? under Instrument Parameters in the System Commands and Queries section of this manual); because, in AUTO, the instrument can determine the bandwidth.

ARES (automatic resolution bandwidth) command



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.

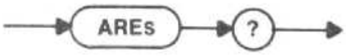
OFF — ARES ON is cancelled, leaving the resolution bandwidth at the current value.

Macro Memory Used — 2 bytes.

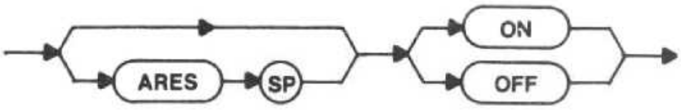
Power-up value — On.

Interaction — ARES OFF also cancels RESBW AUTO. ARES is turned off by any RESBW command except RESBW AUTO.

ARES (automatic resolution bandwidth) query

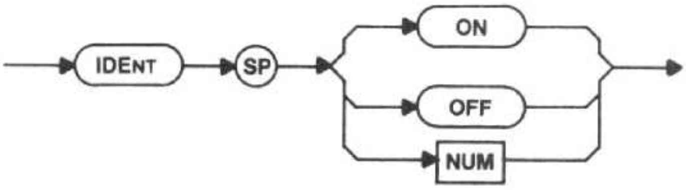


Response to ARES query



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

IDENT (identify) command



ON — The 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.

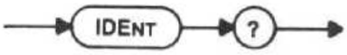
OFF — The identify function is turned off.

Macro Memory Used — 2 bytes.

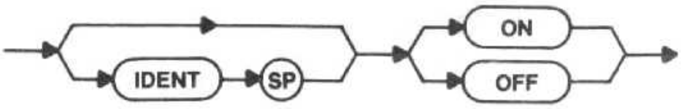
Power-up value — Off.

Interaction — The span must be 50 kHz/div or less in bands 1–5 (50 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.

IDENT (identify) query



Response to IDENT query



VERTICAL DISPLAY AND REFERENCE LEVEL

The commands in this group control the vertical scale factor (VRTDSP) and reference level (REFLVL and FINE) of the display, and select the reference level units (RLUNIT) and the reference level offset (ROFSET). The gain distribution (combination of RF attenuation and IF gain) is automatically set according to the reference level mode (RLMODE); this takes into account the least amount of RF attenuation (MINATT) allowed or maximum power (MAXPWR) expected, and the current RF attenuation is requested (RFATT?). The reduced gain mode (RGMODE enables 10dB of IF gain and RF attenuation reduction when in 10 dB/division.The largest signal in a window around the display data point can be peaked (PEAK). The pulse stretcher (PLSTR) stretches narrow or pulsed signals for acquisition and display. If a video filter (VIDFLT) is switched in, noise in the display is reduced. Calibration of the IF filters for frequency and amplitude is possible from the front panel (CAL and ENCAL).

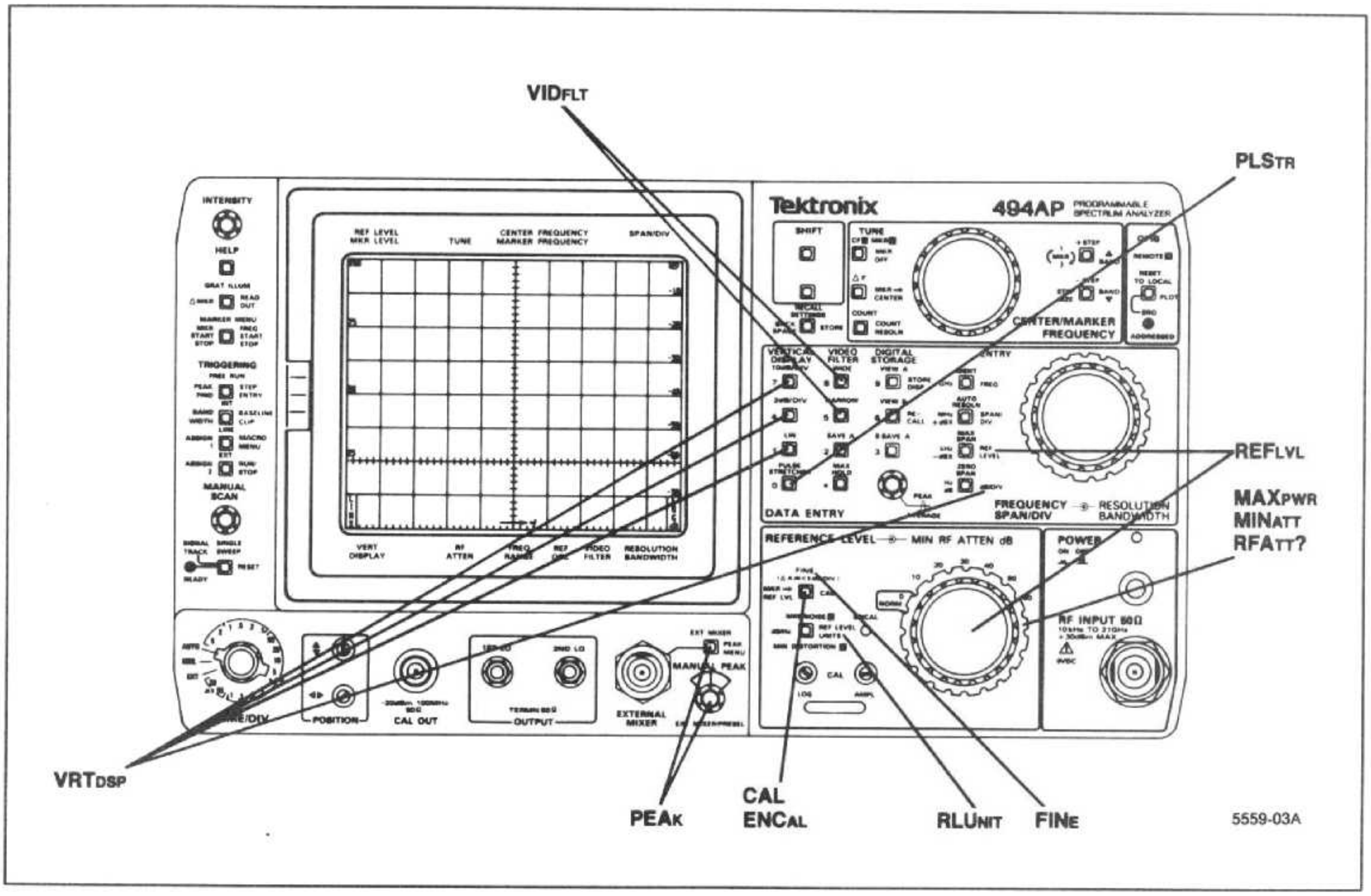
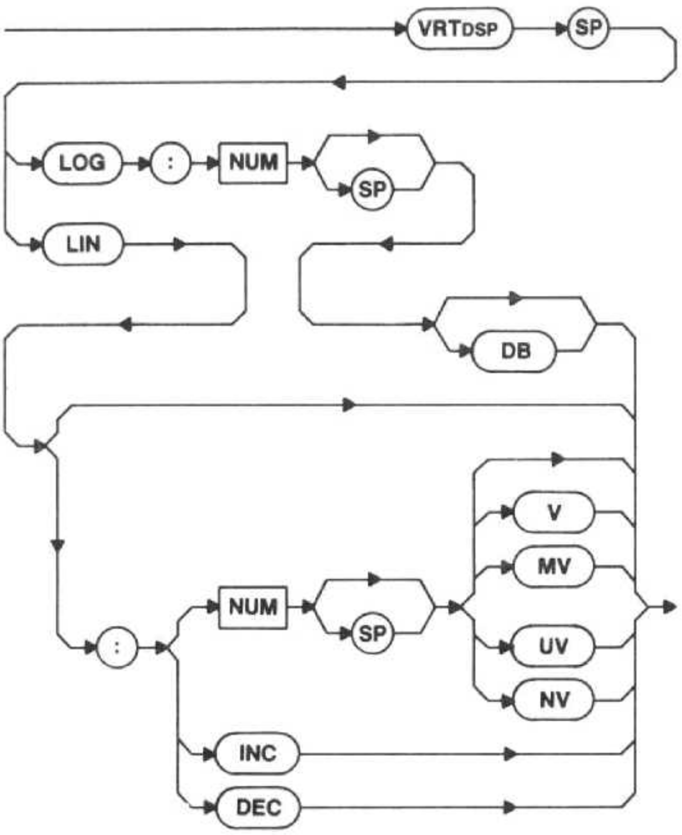


Figure 4-3. Front-panel Vertical Display and Reference Level commands.

VRTDSP (vertical display) command



LOG — The display is scaled to the dB/division specified by integers in the range 1 to 15; non-integers are rounded. 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 the volts equivalent of the nearest 1 dB/div. If NUM is omitted, the display is scaled to leave the reference level at its current value; V/D = 1/8*(volts equivalent of REFLVL). INC or DEC changes the scale factor to the next step in the 1-2-5 volts/division sequence, if possible, when FINE is OFF. When FINE is ON, the next step is determined by the 1 dB change in REFLVL that INC or DEC causes; the new scale factor is 1/8*(volts equivalent of REFLVL). Out-of-range values cause the instrument to report execution error message 35.

Macro Memory Used — 3 bytes.

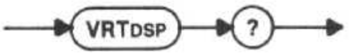
Examples — VRT LOG:3
VRTDSP LOG:2DB
VRT LOG:1 DB
VRTDSP LIN
VRT LIN:2
VRTDSP LIN:1.5MV
VRT LIN:75 NV
VRTDSP LIN:INC

Range — Log is 1–15 dBV
Lin is 39.6 nV/div to 2.8 V/div, if the reference level offset is zero

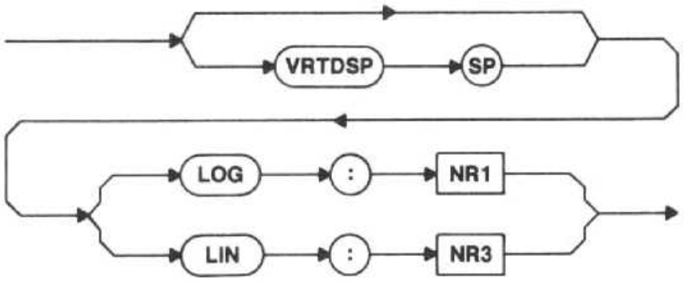
Power-up value — LOG:10 dB/division.

Interaction — The selection of 1, 2, 3, or 4 dB/div with FINE ON causes the spectrum analyzer to enter a delta-amplitude mode. See FINE for a discussion of this mode.

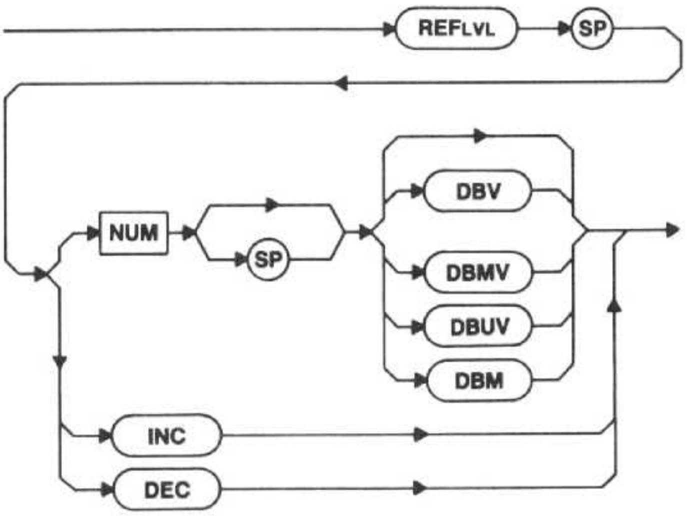
VRTDSP (vertical display) query



Response to VRTDSP query



REFLVL (reference level) command



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 required for other mnemonics.

NUM — The instrument sets the reference level to the nearest dBm step for a log vertical display (except in the Delta Amplitude mode), and to the nearest dBm step for a linear vertical display. The Delta Amplitude mode allows 0.25 dB resolution; the argument to the REFLVL command is always the absolute reference level, not an offset to the present reference level, though the crt readout shows relative amplitude in the Delta Amplitude mode only. 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.

INC or **DEC** — The reference level is stepped up or down once. The step value is determined by the value of the VRTDSP scale factor and FINE selection (refer to Table 4-4).

Macro Memory Used — 4 bytes.

Table 4-4

REFERENCE LEVEL STEPS

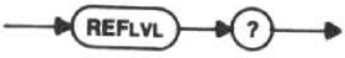
VERTDSP Scale Factor	FINE ON	FINE OFF
15 dB	1 dB	15 dB
14 dB	1 dB	14 dB
13 dB	1 dB	13 dB
12 dB	1 dB	12 dB
11 dB	1 dB	11 dB
10 dB	1 dB	10 dB
9 dB	1 dB	9 dB
8 dB	1 dB	8 dB
7 dB	1 dB	7 dB
6 dB	1 dB	6 dB
5 dB	1 dB	5 dB
4 dB	Delta-Amplitude mode (0.25 dB)	4 dB
3 dB	Delta-Amplitude mode (0.25 dB)	3 dB
2 dB	Delta-Amplitude mode (0.25 dB)	1 dB
1 dB	Delta-Amplitude mode (0.25 dB)	1 dB
LIN	1 dB	Either 6 dB or 8 dB (varies to match 1-2-5 volts/div sequence)

Examples—REF −200DBV
REFLVL −10 DBMV
REF −30DBUV
REFLVL −25 DBM
REF INC

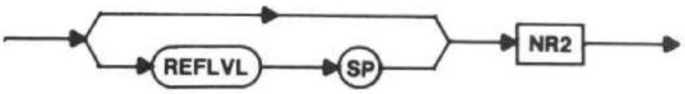
Macro Memory Used — 4 bytes.

Power-up value — 0 dBm.

REFLVL (reference level) query

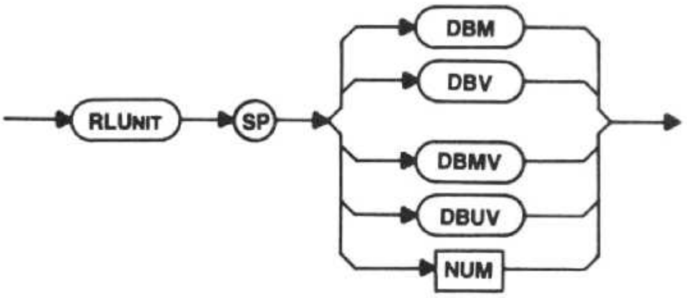


Response to REFLVL query



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, whether or not in the Delta Amplitude mode.

RLUNIT (reference level units) command



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 required for most other mnemonics.

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.

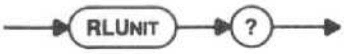
NUM—0 = dBm
1 = dBV
2 = dBmV
3 = dBuV

Macro Memory Used — 2 bytes.

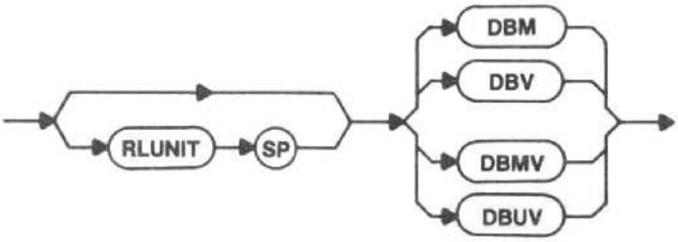
Power-up value — dBm.

Interaction — 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.

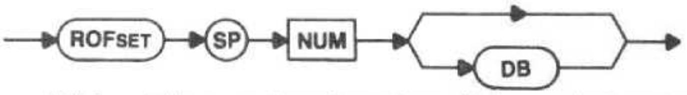
RLUNIT (reference level units) query



Response to RLUNIT query



ROFSET (reference level offset) command



NUM — This sets the offset that will be applied to the reference and marker levels.

Macro Memory Used — 3 bytes.

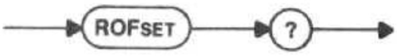
Examples — ROF 26
ROFSET 30 DB
ROF 7.5DB

Range — −30.0 dB to +30.0 dB.

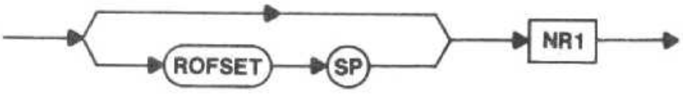
Power-up value — 0 dB.

Interaction — The offset value will affect the responses to the queries of REFLVL, MAMPL, MLOCAT, MAXPWR, and THRHLD. ROFSET must be included in the values sent with the REFLVL, THRHLD, and MAXPWR commands.

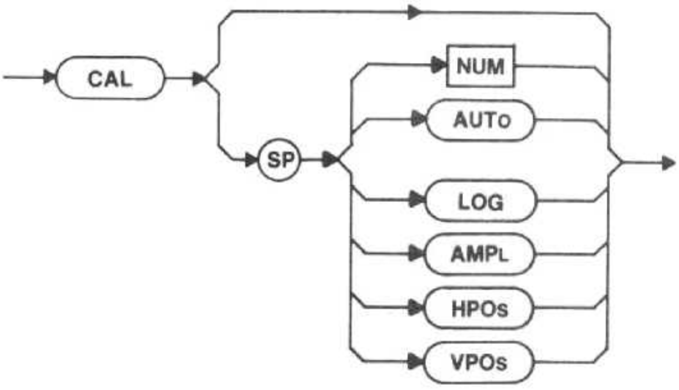
ROFSET (reference level offset) query



Response to ROFSET query



CAL command



NOTE

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

NUM—0 – AUTO
1 – LOG
2 – AMPL
3 – HPOS
4 – VPOS

CAL (without arguments) or **CAL AUTO** — 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.

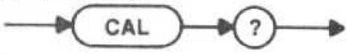
CAL LOG — The instrument is set up so you can set the front-panel CAL LOG adjustment. CAL LOG has an indefinite execution time and will operate until either a device clear (DCL) is received from the GPIB port or the spectrum analyzer is returned to local control from the instrument front panel. An instruction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

CAL AMPL — The instrument is set up so you can set the front-panel CAL AMPL 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.

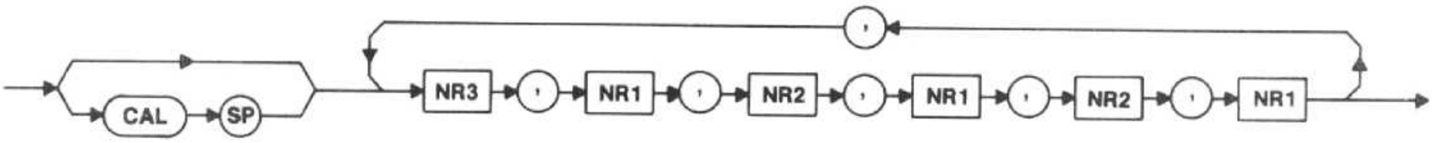
CAL HPOS — The instrument is set up so you can set the front-panel horizontal POSITION control. CAL HPOS has an indefinite execution time and will operate until either a device clear (DCL) is received from the GPIB port or the spectrum analyzer is returned to local control from the instrument front panel.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

CAL 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.

CAL query



Response to CAL query



In the CAL? response, the same data is given in succession for the 3 MHz, 1 MHz, 100 kHz (300 kHz for Option 07 instruments), 10 kHz, 1 kHz, 100 Hz, and 10 Hz filters (in that order). 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

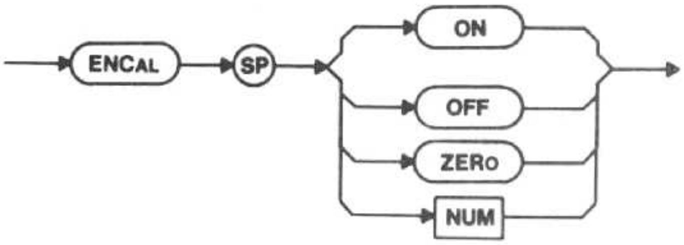
The frequency error is the difference between the measured filter frequency and 10 MHz, expressed in Hz. The level error is the difference between the measured filter level and the measured level of the 3 MHz filter, expressed in dB. The noise bandwidth is expressed as the dB correction used to normalize the filter's output to 1 Hz. Use Table 4-5 to decode the calibration code numbers.

Table 4-5

CALIBRATION CODES

Code Number	Description
0	A calibration value for this item has not been found (i.e., 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).
2	The value recorded for this item is the limit value (i.e., 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 (i.e., 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.)

ENCAL (enable calibration factors) command



OFF — The filter's amplitude and frequency are not corrected, and the nominal noise bandwidth is used.

ON — The calibration factors are used internally to correct frequency and level errors and noise bandwidth in the filters.

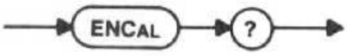
ZERO — Set calibration factors to 0; this does not affect OFF/ON status.

NUM—0 = OFF
1 = ON
2 = ZERO

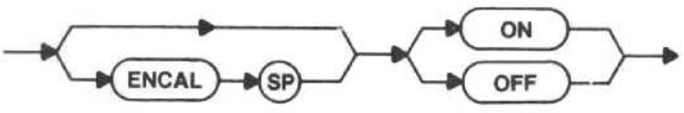
Macro Memory Used — 2 bytes.

Power-up Value — ON.

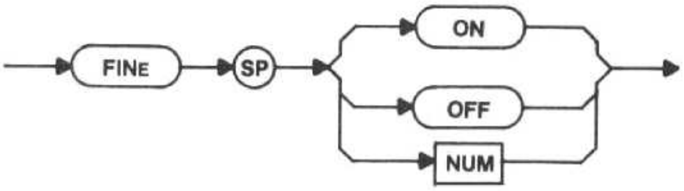
ENCAL (enable calibration factors) query



Response to ENCAL query



FINE (fine reference level steps) command



ON — Small steps are selected for the INC or DEC arguments in the reference level command (see REFLVL for details). With vertical scale factors of 1, 2, 3, and 4 dB/div, FINE ON selects the delta-amplitude mode.

Delta-Amplitude Mode — The Delta Amplitude mode is active when both FINE reference level steps and a scale factor of 4 dB/div or less are selected. In this mode, the crt VERT DISPLAY readout initializes to 0.00 dB. Changes in reference level are displayed as the difference between the initial level and the new level, not the absolute reference level. This readout is available with UPRDO?. REFLVL? returns the absolute reference level. The initial gain distribution (RF attenuation and IF gain) is not disturbed; changes in reference level are created by an offset in the display. This allows signals to be compared with inherently higher relative accuracy. The delta-amplitude range is 657.75 dB and slides depending on the reference level when the Delta Amplitude mode is entered.

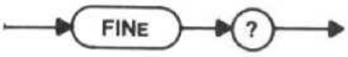
OFF — Normal steps are restored for reference level changes, which cancels the Delta Amplitude mode (if active).

Macro Memory Used — 2 bytes.

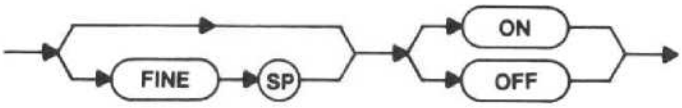
Power-up value — OFF.

Interaction — This command, along with VRTDSP, controls the spectrum analyzer response to REFLVL INC or DEC.

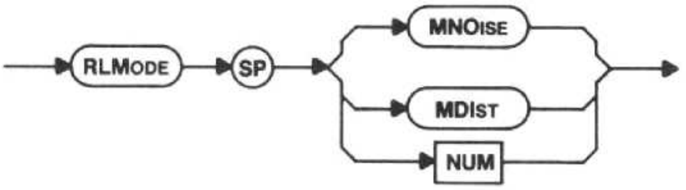
FINE (fine reference level steps) query



Response to FINE query



RLMODE (reference level mode) command



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 analyzer, hence less distortion.

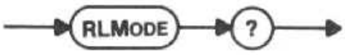
NUM—0 = MNOISE
1 = MDIST

Macro Memory Used — 2 bytes.

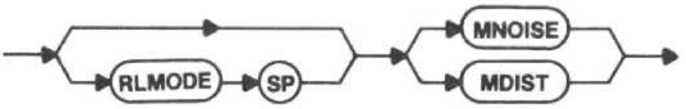
Power-up value — MNOISE

Interaction — This command affects the gain distribution obtained with the REFLVL command (see also MINATT and MAXPWR).

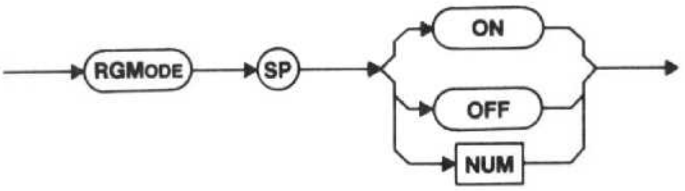
RLMODE (reference level mode) query



Response to RLMODE query



RGMODE (reduced gain mode) command



This command enables or disables 10 dB of IF gain and RF attenuation reduction when in 10 dB/division.

ON — The reduced gain mode is turned on.

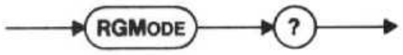
OFF — The reduced gain mode is turned off.

Interaction — 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.

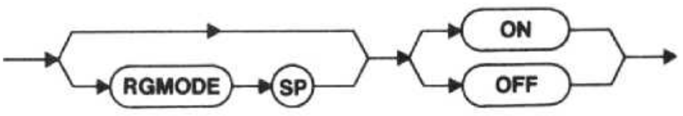
Macro Memory Used — 2 bytes.

Power-up value — OFF.

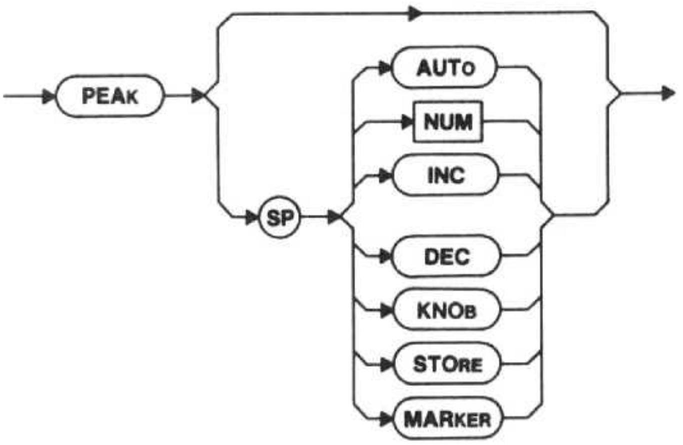
RGMODE (reduced gain mode) query



Response to RGMODE query



PEAK (peaking) command



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 the Waveform Processing section later in this manual for information on the display data point). When peakin the preselector, peaking occurs ±1 division or 0 MHz (whichever is less) on either side of the center or marker. 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/DIVISION, 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.

NUM — The number is stored in memory. Non-integers 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.

Range — 0 to 1023.

INC or **DEC** — The value of PEAK is changed ±1 from its current value, which is stored in memory, and the new value is stored in memory.

KNOB — The front-panel MANUAL PEAK control is active. You can manually peak the spectrum analyzer's response from the front panel. All other arguments switch to internal peaking and cancel KNOB.

STORE — The value stored in memory is used for the present band (with internal mixer in bands 2–5) or frequency (with external mixers or preselector).

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 off.

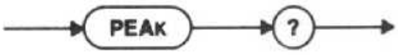
Macro Memory Used — 4 bytes.

Examples — PEAK
PEA AUTO
PEAK 512
PEA STO

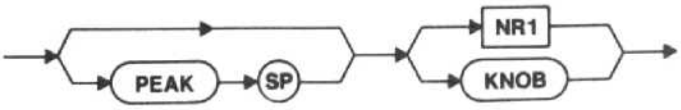
Power-up value — KNOB (MANUAL PEAK control on).

Interaction — 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, and the stored number or knob position has no effect.

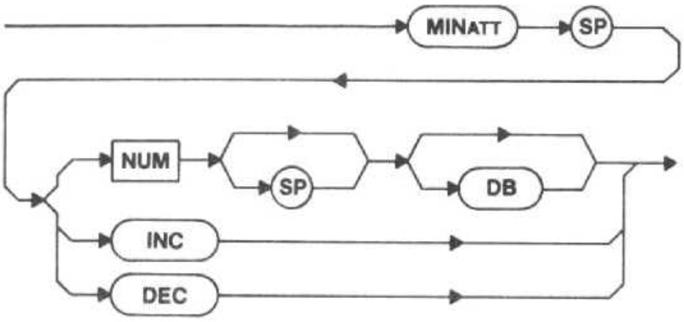
PEAK (peaking) query



Response to PEAK query



MINATT (minimum RF attenuation) command



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

Macro Memory Used — 2 bytes.

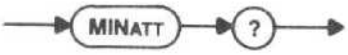
Examples — MINATT 20
MIN 42 DB
MINATT INC

Range — 0 to 60.

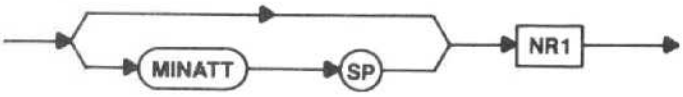
Power-up value — MIN RF ATTEN dB control setting.

Interaction — 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.

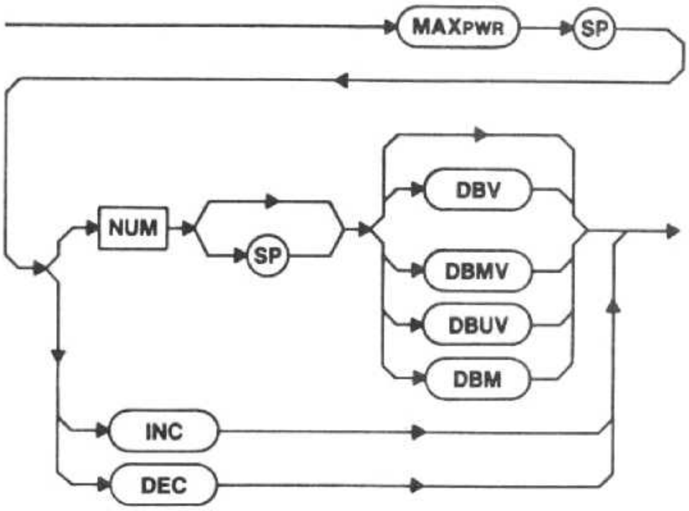
MINATT (minimum RF attenuation) query



Response to MINATT query



MAXPWR (maximum input power) command



NUM — This is an input to a instrument that 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 −18 dBm at the 1st Mixer. (This is the instrument's 1 dB compression point.) 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 required for other mnemonics.

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

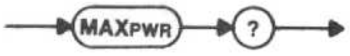
Macro Memory Used — 4 bytes.

Examples — MAXPWR +20DBMV
MAX 18 DBUV
MAXPWR DEC

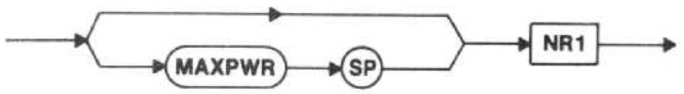
Power-up value — −18 to +42; dependent on MINATT value (−18 + MINATT value).

Interaction — 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.

MAXPWR (maximum input power) query

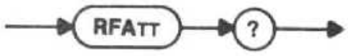


Response to MAXPWR query



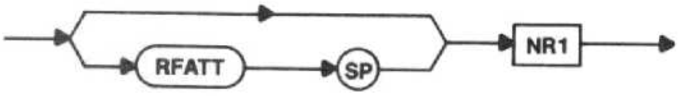
Only the number value will be returned; the units will not be indicated (the units will be the current reference level units).

RFATT (RF attenuation) query



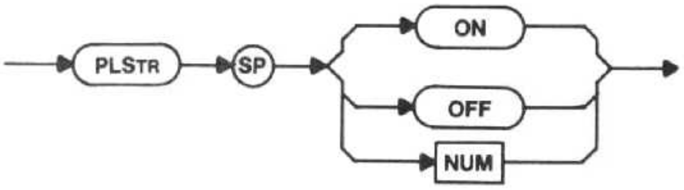
Requests the current value of RF attenuation.

Response to RFATT query



There is no RFATT command.

PLSTR (pulse stretcher) command



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.

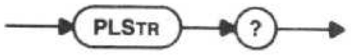
Pulse stretcher may be required to view and store fast pulsed signals. For short pulses, the signal may exist for less time than is required for a point to be digitized, causing either no value or too low a value to be stored.

OFF — The pulse stretcher is turned off.

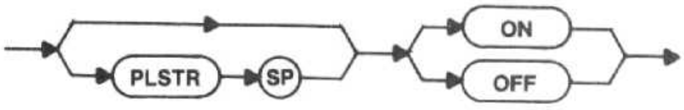
Macro Memory Used — 2 bytes.

Power-up value — OFF.

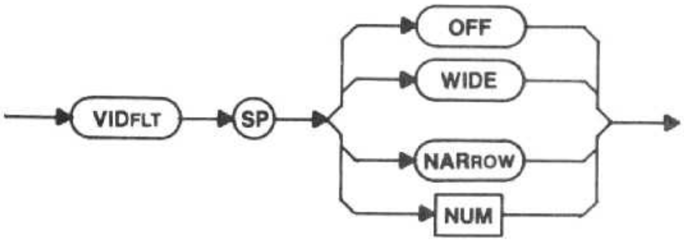
PLSTR (pulse stretcher) query



Response to PLSTR query



VIDFLT (video filter) command



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.

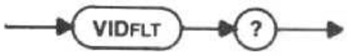
NUM—0 – OFF
1 – WIDE
2 – NARROW

Macro Memory Used — 2 bytes.

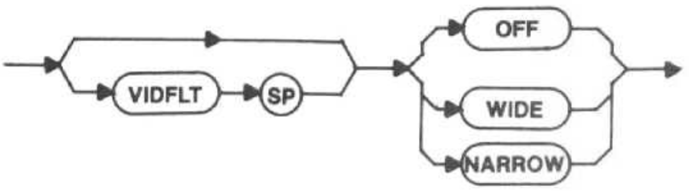
Power-up value — OFF.

Interaction — 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.

VIDFLT (video filter) query



Response to VIDFLT query



SWEEP CONTROL

Three commands control the instrument sweep, which is used both to sweep the frequency span and the crt display. These commands control the sweep triggering and mode (TRIG and SIGSWP) and sweep rate (TIME). Selection of TIME AUTO directs the instrument to automatically match the sweep to related instrument parameters. Other options include manual or external analog control of the sweep.

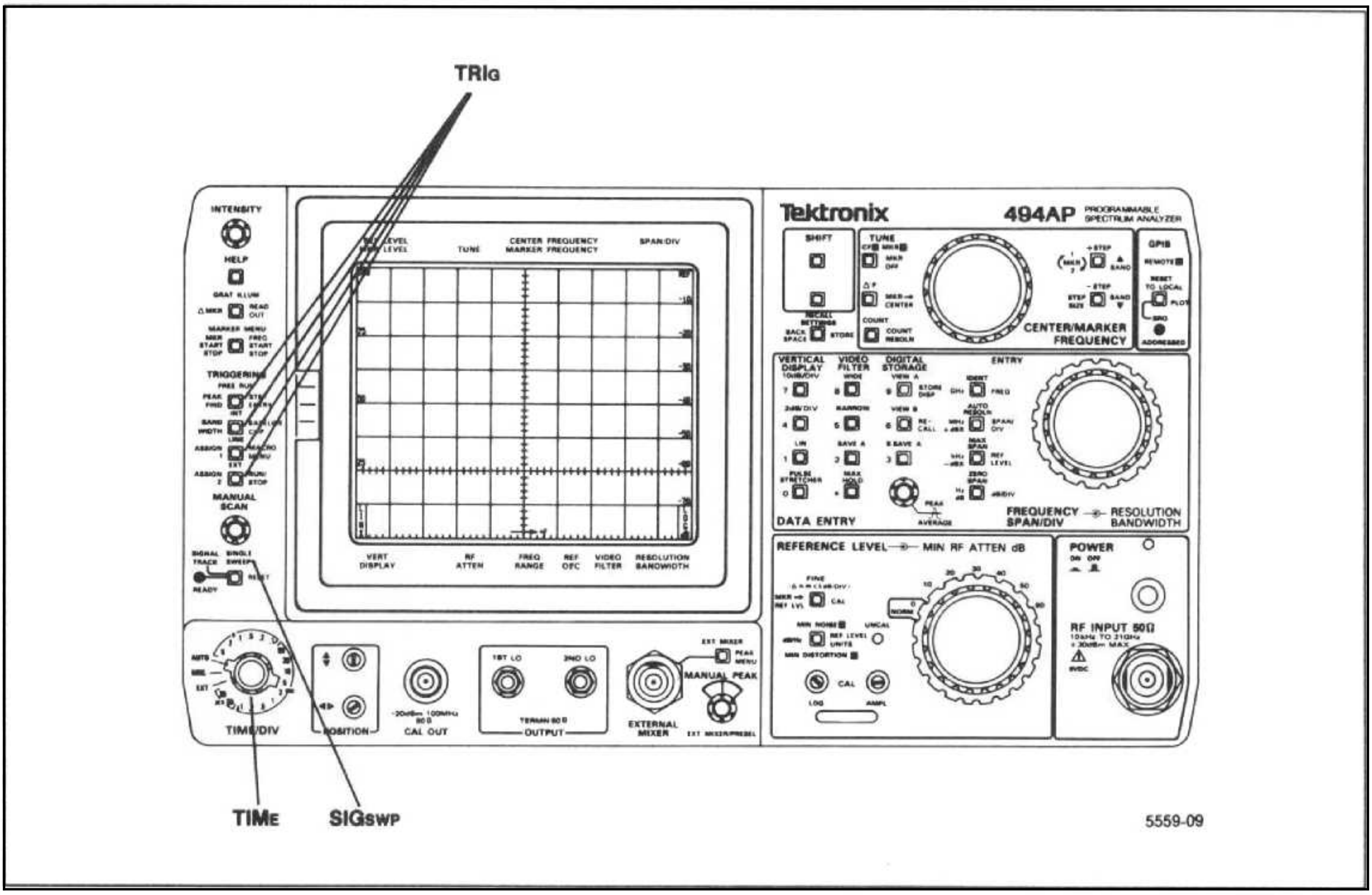
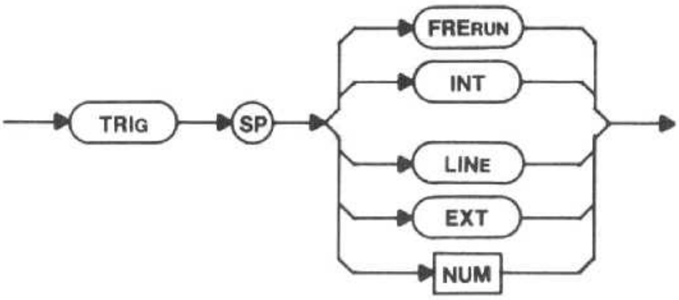


Figure 4-4. Front-panel Sweep Control commands.

TRIG (triggering) command



FRERUN — The instrument sweep is allowed to run repetitively. A trigger is not required (and 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.

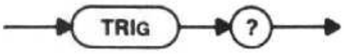
NUM — 0 = FRERUN
1 = INT
2 = LINE
3 = EXT

Macro Memory Used — 2 bytes.

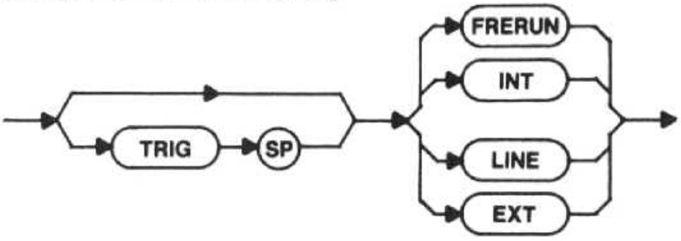
Power-up value — FRERUN.

Interaction — 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; in the time domain mode, the required frequency is the center frequency. In the frequency domain mode, the required frequency must be within the selected frequency range.

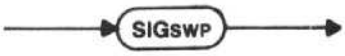
TRIG (triggering) query



Response to TRIG query



SIGSWP (single-sweep) command



On the first SIGSWP command, the instrument enters the single-sweep mode, which stops the current sweep. Once in the single-sweep mode, this command arms the sweep and lights the front-panel READY light, which remains lit 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. Refer to Programming Techniques in the Helps and Hints section later in this manual.

NOTE

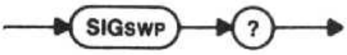
> The Single Sweep mode should be used under most programming conditions (see Programming Techniques in the Helps and Hints section later in this manual).

Macro Memory Used — 1 byte.

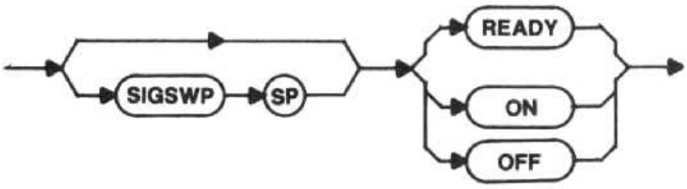
Power-up value — OFF.

Interaction — Any TRIG command cancels the single-sweep mode.

SIGSWP (single-sweep) query

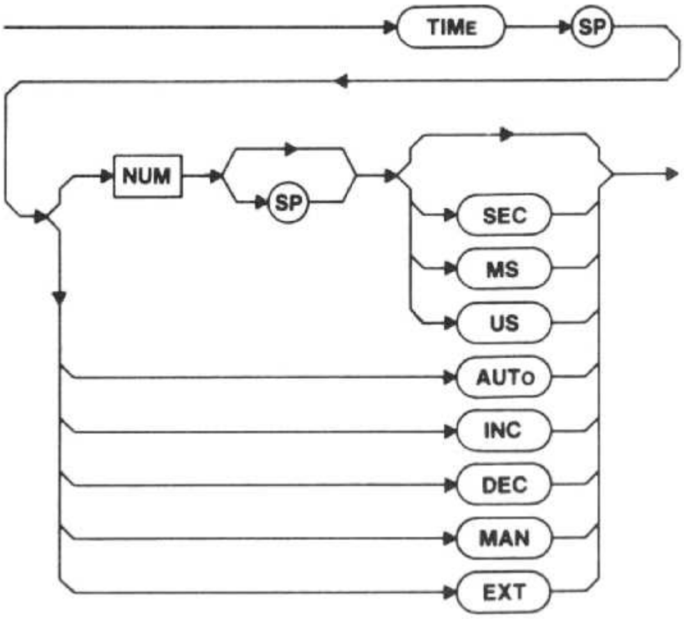


Response to SIGSWP query



The response to the SET query is omitted if single-sweep is not active (see SET? under Instrument Parameters in the System Commands and Queries section of this manual).

TIME (time/div) command



NUM — 1-2-5 sequence in the range $20.0E^{-6}$ 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 is requested to select the fastest sweep allowed for calibrated response.

INC or **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.

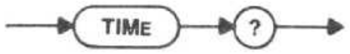
Macro Memory Used — 5 bytes.

Examples—TIME 1
TIM 10 MS
TIME MAN

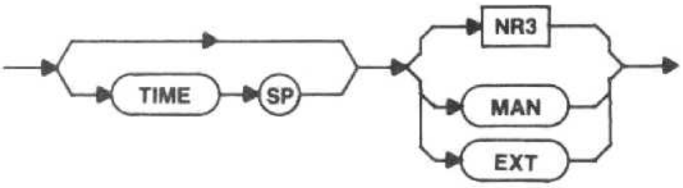
Power-up value — TIME/DIV control setting.

Interaction — Too fast a sweep speed for a given resolution bandwidth will uncalibrate the display. For digital storage to properly acquire spectrum data, 10 ms/div is the maximum usable sweep rate.

TIME (time/div) query



Response to TIME query



The SET? response includes AUTO as a possible argument (see SET? under Instrument Parameters in the System Commands and Queries section of this manual).

DIGITAL STORAGE

These commands control the digital storage functions of display (AVIEW, BVIEW), updating (SAVEA), comparison (BMINA), display storage (DSTORE), display recall (DRECAL), and digitizer control (MXHLD, CRSOR).

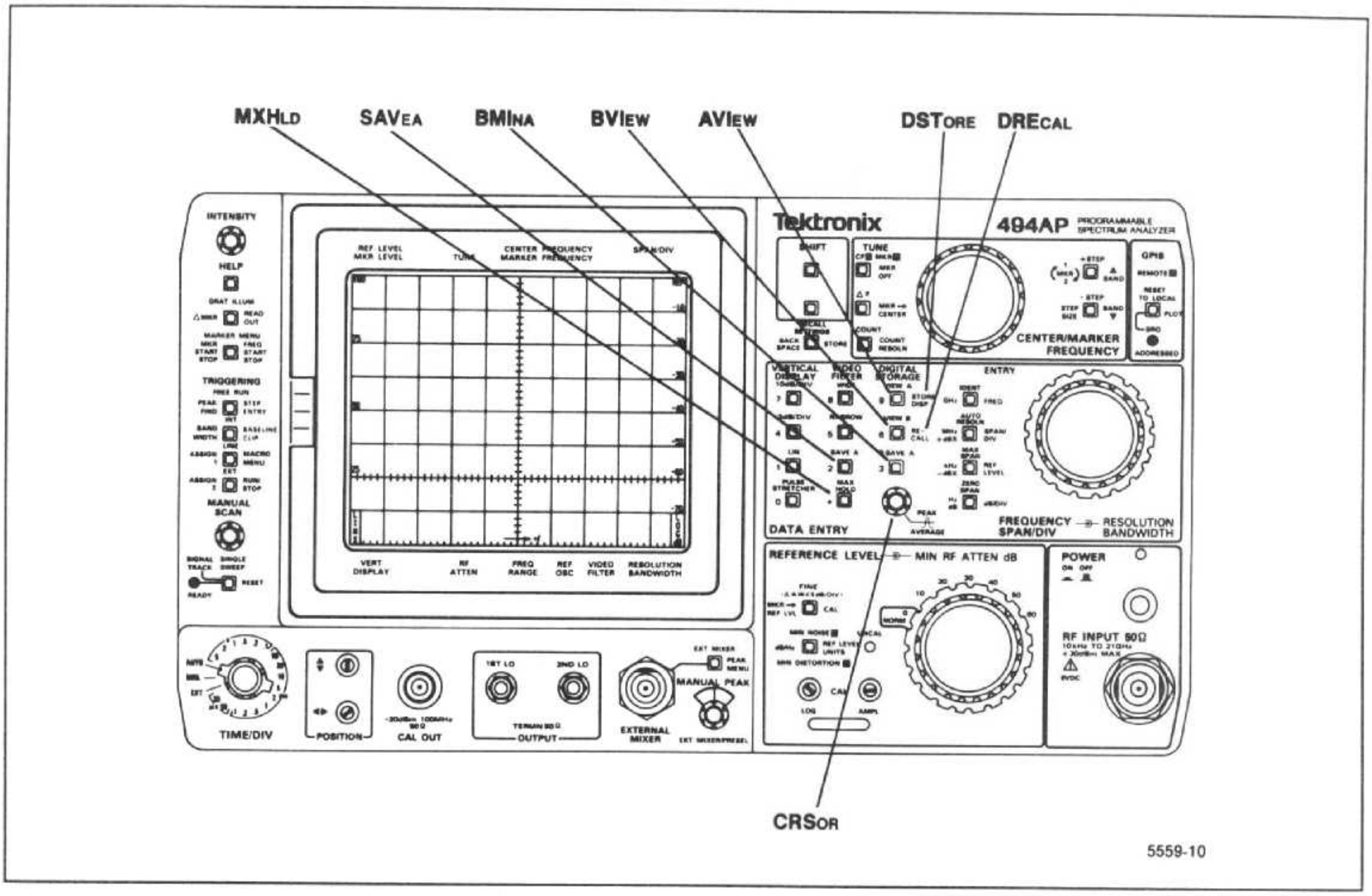
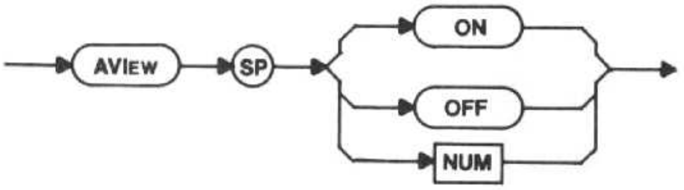


Figure 4-5. Front-panel Digital Storage commands.

AVIEW (A waveform display) command



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.

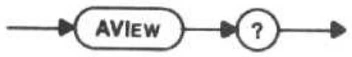
OFF — The display of the A waveform is turned off. (See the ON description for operation with SAVE A off.) If both AVIEW and BVIEW are turned off, the input signal is displayed in real time.

Macro Memory Used — 2 bytes.

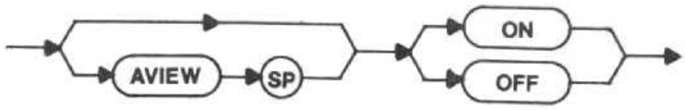
Power-up value — ON.

Interaction — While SAVEA is ON, any updating of the trace display in A is halted.

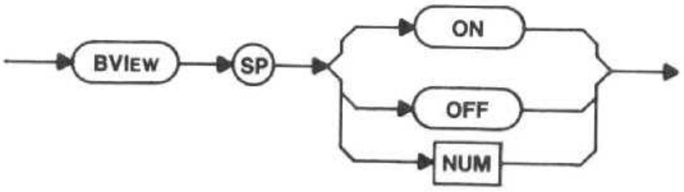
AVIEW (A waveform display) query



Response to AVIEW query



BVIEW (B waveform display) command



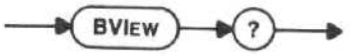
ON — The B waveform is displayed on the crt. The A and B waveforms are independent and may be displayed together or separately; however, both waveforms will be displayed if either AVIEW or BVIEW is on and SAVEA is off.

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

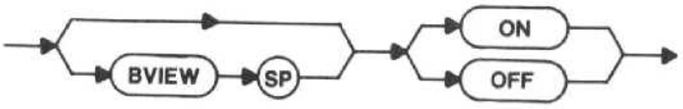
Macro Memory Used — 2 bytes.

Power-up value — ON.

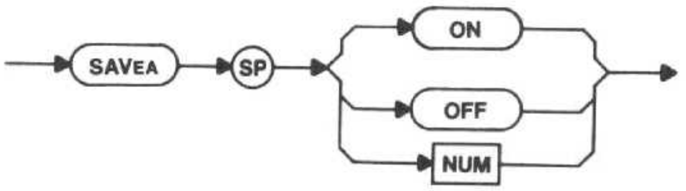
BVIEW (B waveform display) query



Response to BVIEW query



SAVEA (save A waveform) command



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 (both BVIEW and BMINA off).

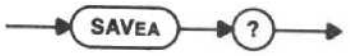
OFF — The A waveform updating is resumed.

Macro Memory Used — 2 bytes.

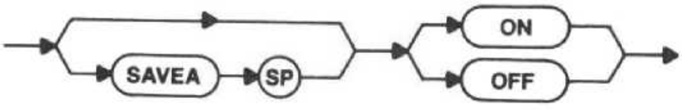
Power-up value — OFF.

Interaction — BMINA ON turns SAVEA ON. SAVEA OFF turns BMINA OFF.

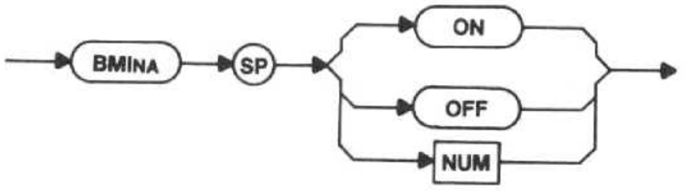
SAVEA (save A waveform) query



Response to SAVEA query



BMINA (B–A waveform display) command



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 normally set at graticule center, but may be varied with an internal switch (refer any changes to qualified service personnel).

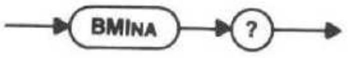
OFF — The difference display is turned off.

Macro Memory Used — 2 bytes.

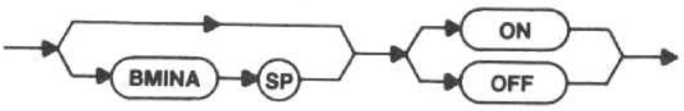
Power-up value — OFF.

Interaction — BMINA ON turns SAVEA ON. SAVEA OFF turns BMINA OFF.

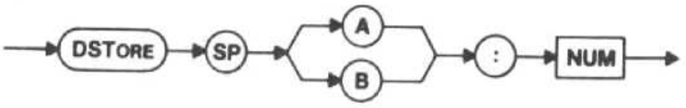
BMINA (B–A waveform display) query



Response to BMINA query



DSTORE (store display) command



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.

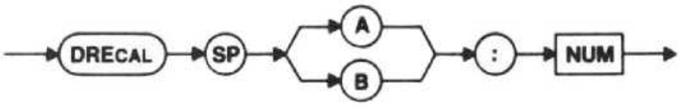
Macro Memory Used — 3 bytes.

Examples—DSTORE A:4
DST B:2

Range — 0 to 8.

There is no DSTORE query.

DRECAL (recall display) command



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.

Macro Memory Used — 3 bytes.

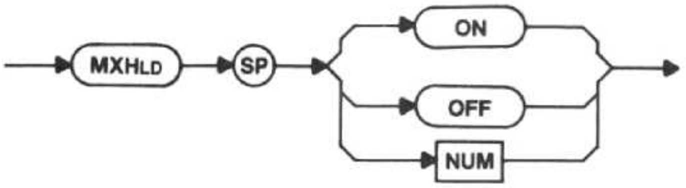
Examples—DRECAL A:4
DRE B:2

Range — 0 to 8.

Interaction — 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.

There is no DRECAL query.

MXHLD (max hold) command



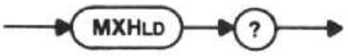
ON — Digital storage holds the maximum value obtained for each point in both the A and B waveforms; a point is updated only if the new value is greater than the current value. The A waveform is not affected if SAVEA is on.

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

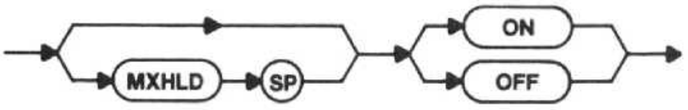
Macro Memory Used — 2 bytes.

Power-up value — OFF.

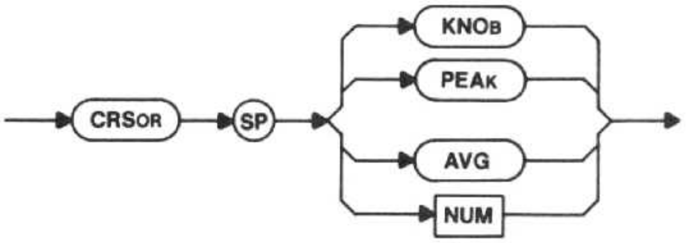
MXHLD (max hold) query



Response to MXHLD query



CRSOR (peak/average cursor) command



KNOB — The PEAK/AVERAGE control is under local control, so you can set the cursor level, which is shown by a line across the crt. Above the line, peak values are stored as each point is updated; below the line, averaged values are stored.

PEAK — The line position on the crt is not affected. The peak value digitized at each point is used to update digital storage, regardless of the cursor position last set by KNOB. This is the same as setting the cursor to its lowest (minimum) position.

AVG — The line position on the crt is not affected. Average values are used to update the waveforms, regardless of the cursor position last set by KNOB. PEAK AVG is the same as if the cursor is set to its highest (maximum) position.

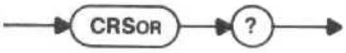
NUM—0 = KNOB
1 = PEAK
2 = AVG

Macro Memory Used — 2 bytes.

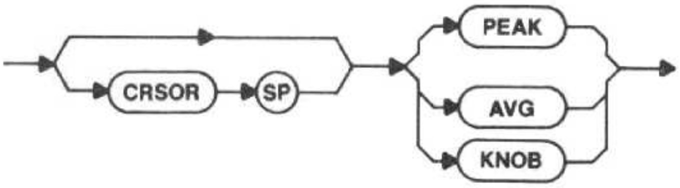
Interaction — Averaging can reduce the value in digital storage for signals with very narrow response or pulsed signals.

Power-up value — KNOB.

CRSOR (peak/average cursor) query



Response to CRSOR query



DISPLAY CONTROL

These commands control the instrument crt display functions to display the readout (REDOUT), light the graticule (GRAT), and eliminate the baseline trace (CLIP).

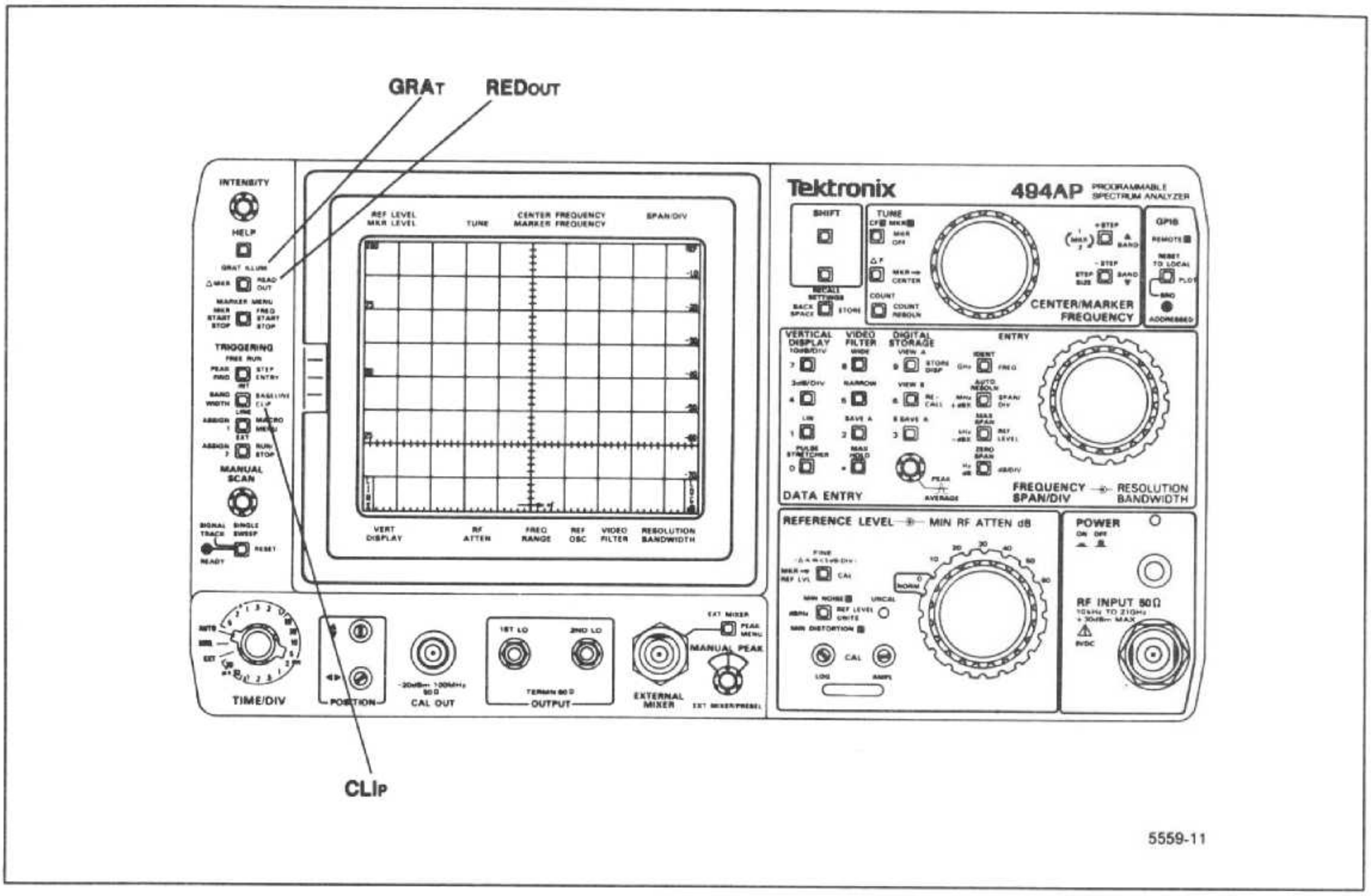
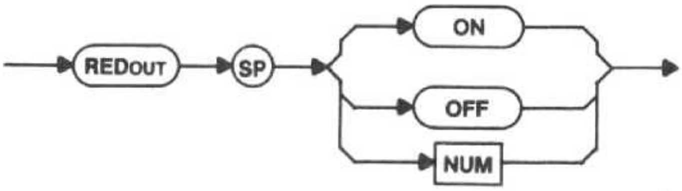


Figure 4-6. Front-panel Display Control commands.

REDOUT (readout) command



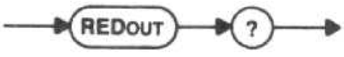
ON — The instrument settings are displayed.

OFF — The instrument settings are not displayed; the readout is blanked.

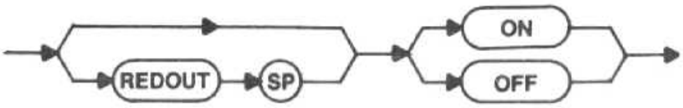
Macro Memory Used — 2 bytes.

Power-up value — ON.

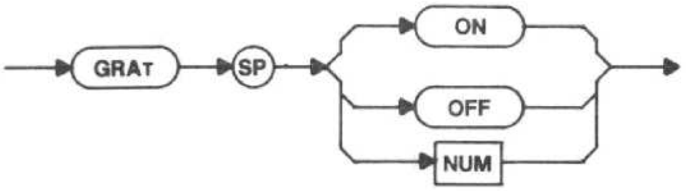
REDOUT (readout) query



Response to REDOUT query



GRAT (graticule) command



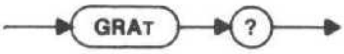
ON — The crt graticule is lighted.

OFF — The crt graticule is dark; not lighted.

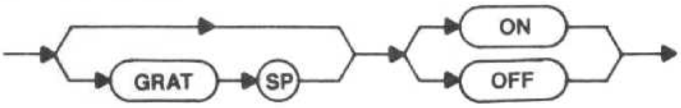
Macro Memory Used — 2 bytes.

Power-up value — OFF.

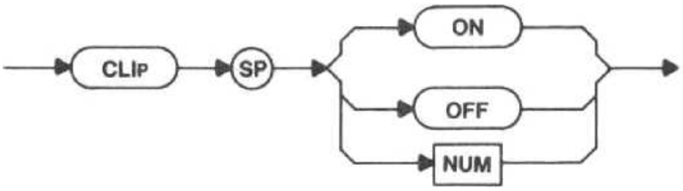
GRAT (graticule) query



Response to GRAT query



CLIP (blank baseline) command



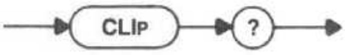
ON — Aproximately one graticule division of the screen trace is turned off at the baseline of the crt. This allows the readout at the bottom of the screen to be clearly seen when viewing or plotting the display, and it eliminates the bright baseline when photographing the display.

OFF — The full trace is displayed on the crt.

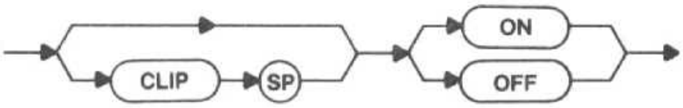
Macro Memory Used — 2 bytes.

Power-up value — OFF.

CLIP (blank baseline) query



Response to CLIP query



GENERAL PURPOSE

The general purpose commands and queries request front-panel help messages or GPIB command headers (HELP?), store settings in memory (STORE), recall settings from memory (RECALL), transfer data to and from storage memory (RDATA), plot crt information (PLOT) on a choice of plotters (PTYPE), change B–A reference for the plotter (POFSET), cause oscillator corrections at the end of every sweep (ECR), send a service request (SSR), and register valid (RVALID).

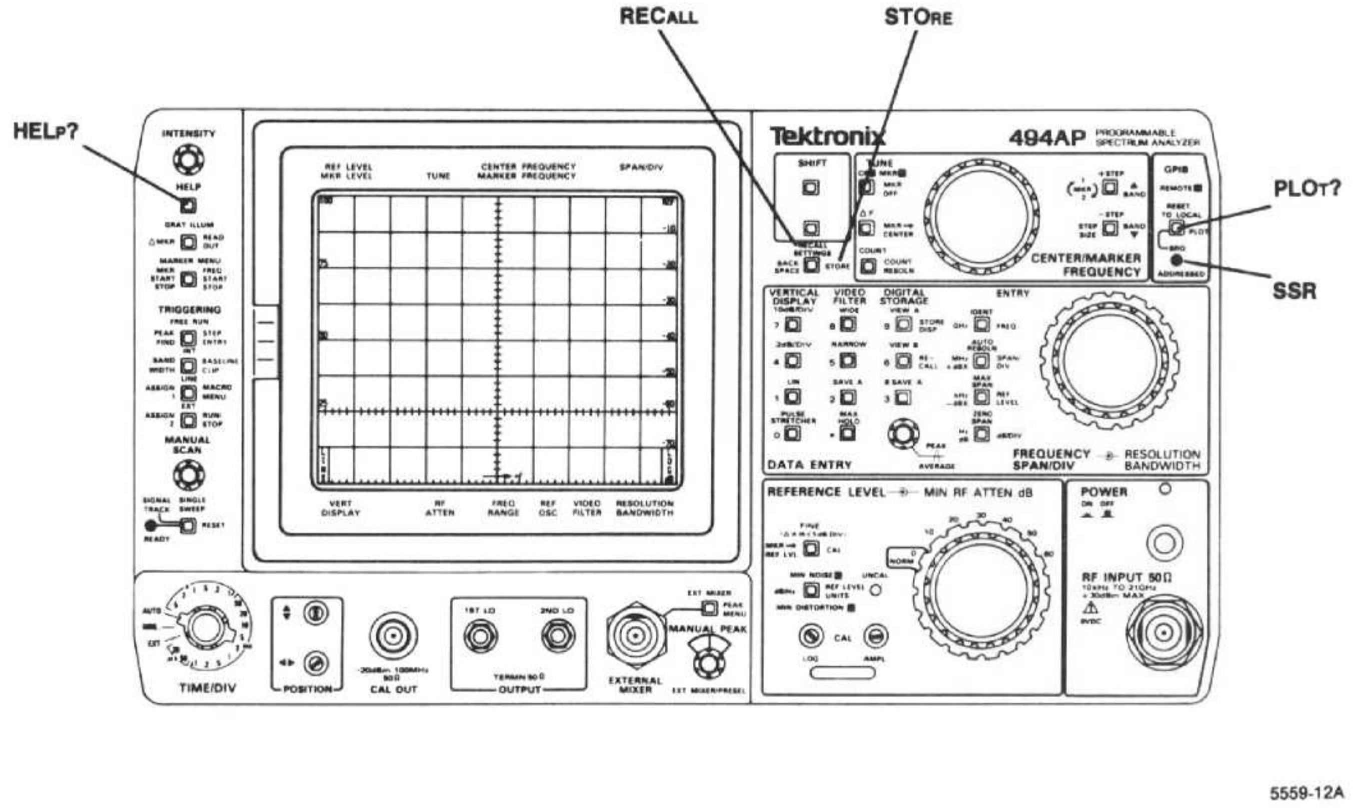
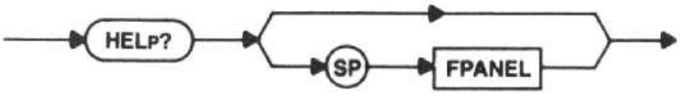
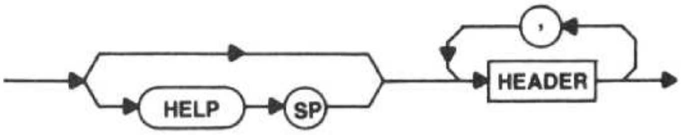


Figure 4-7. Front-panel General Purpose commands.

HELP (help) query

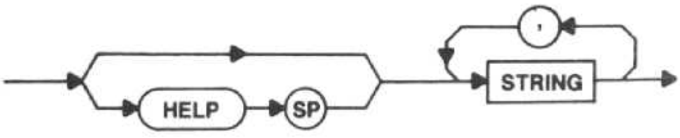


Response to HELP query (with no argument)



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

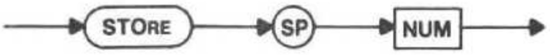
Response to HELP query (with FPANEL)



Each string represents one crt readout line of a help message. All help messages (black-, blue-, and green-labeled functions and all marker menu functions) are sent by this command.

There is no HELP command.

STORE (store settings) command



NUM — The instrument control settings are stored into the selected memory location.

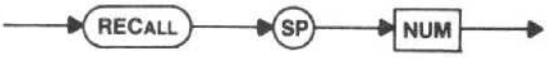
Range — 0 to 9.

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.

There is no STORE query.

RECALL (recall settings) command



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

Range — 0 to 9.

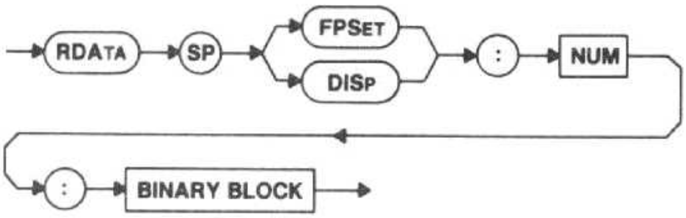
Macro Memory Used — 2 bytes.

Power-up value — The instrument STOREs its current settings in memory 0 automatically when the power is turned off, overwriting any previously-stored settings.

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

There is no RECALL query.

RDATA (register data) command



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. It is intended that this information be obtained from a previous RDATA query.

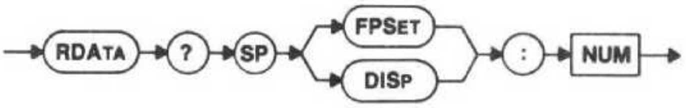
FPSET — The front-panel settings contained in the binary block are transferred to the indicated register.

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.

RDATA (register data) query



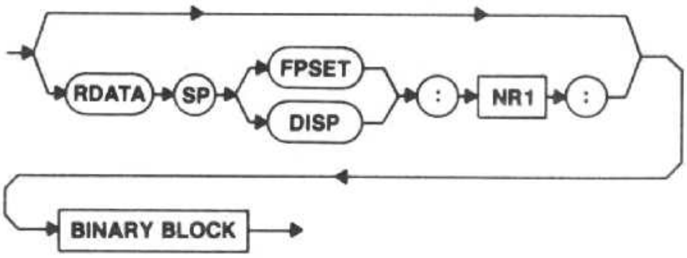
The RDATA query transfers, directly from a numbered storage register, either a front-panel setting or a waveform and associated data. The data is transferred in a coded binary format. It is intended that this information be used only as data for a subsequent RDATA command.

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.

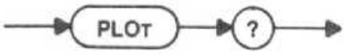
RDATA response



The length of the binary data, exclusive of the byte count and checksum (see Binary Block in Section 2 of this manual) is 128 bytes when a setting is being returned and 642 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 these cases, all binary data bytes will be 0.

PLOT query



The PLOT query sends information to plot the display on a TEKTRONIX 4662 Opt 01, 4662 Opt 31 or 4663 (emulating a 4662) Interactive Digital Plotter, a Hewlett-Packard HP7470A, HP7475A, HP7580B, HP7585B, or HP7586B plotter, or a Gould 6310 or 6320 plotter.

- If REDOUT is ON, corresponding settings will be plotted.
- If GRAT is ON, the scale down the right-hand side of the screen will be plotted, as well as the graticule information. 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).
- 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 the Tektronix 4662 Opt 31 (or the 4663 emulating the 4662), the HP7470A, HP7475A, HP7580B, HP7585B, or HP7586B, or the Gould 6310 or 6320. 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.
- Refer to Using PLOT with Macros in the Macros section of this manual for additional information.

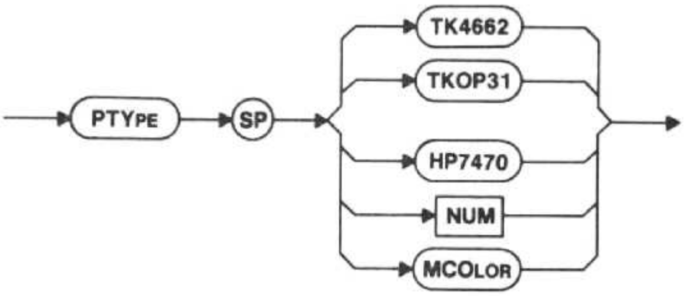
The response to PLOT? depends on the plotter in use (refer to the select plotter command (PTYPE) for a description of the plotter selections).

NOTE

> Since the GPIB languages of the Tektronix 4662 Opt 01, 4662 Opt 31 and 4663 Interactive Digital Plotters, the Hewlett-Packard, or the Gould plotters do not conform to the Tektronix Interface Standard for GPIB Codes, Formats, Conventions, and Features, this response does not follow the standard.

There is no PLOT command.

PTYPE (plotter type) command



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?.

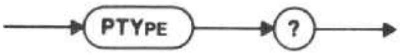
MCOLOR — Selects the Hewlett-Packard HP7475A, 7580B, 7585B, or 7586B, or the Gould 6310 or 6320 as the plotter driven by the data generated by PLOT?.

NUM—0 = TK4662
1 = TKOP31
2 = HP7470
3 = MCOLOR

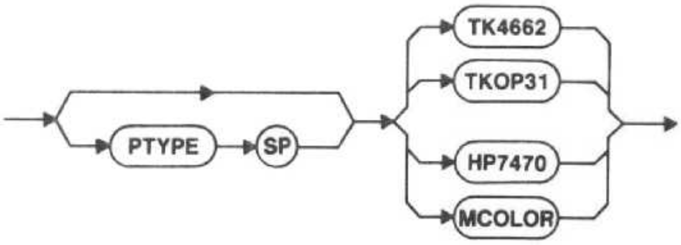
Macro Memory Used — 2 bytes.

Power-up value — The last value stored in memory.

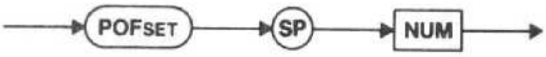
PTYPE (plotter type) query



Response to PTYPE query



POFSET (set K) command



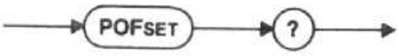
NUM — Sets the reference position for plotting B−A waveforms. Although the range is 0 to 255 the range visible on the screen is 25 at the bottom to 225 at the top. The nearest limit is used if the selected number is out of range (no error is reported).

Range — 0 to 255.

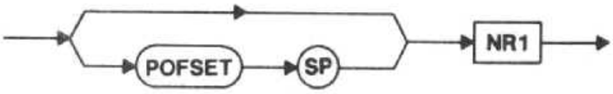
Macro Memory Used — 2 bytes.

Power-up value — The last value stored in memory.

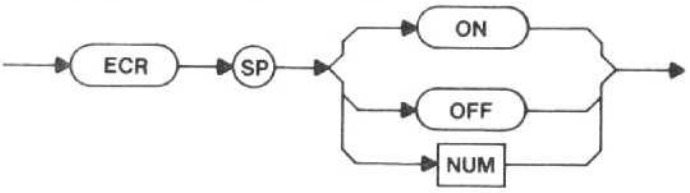
POFSET (set K) query



Response to POFSET query



ECR (end of sweep corrections) command



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.

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

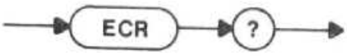
OFF — The time between oscillator corrections is determined by the drift rate of the oscillators.

Interaction — 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.

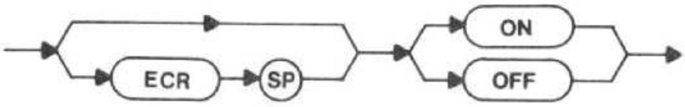
Macro memory used — 2 bytes.

Power-up value — OFF.

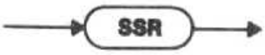
ECR query



Response to ECR query



SSR (send service request) command



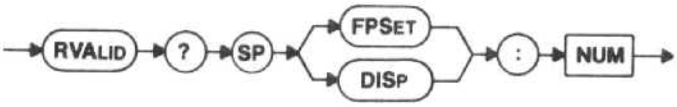
The SSR command asserts SRQ. The status byte will report this as an internal error warning, and system event number 140 will be returned.

Interaction — The service request will be asserted only if RQS is ON.

Macro Memory Used — 2 bytes.

There is no SSR query.

RVALID (register valid) query

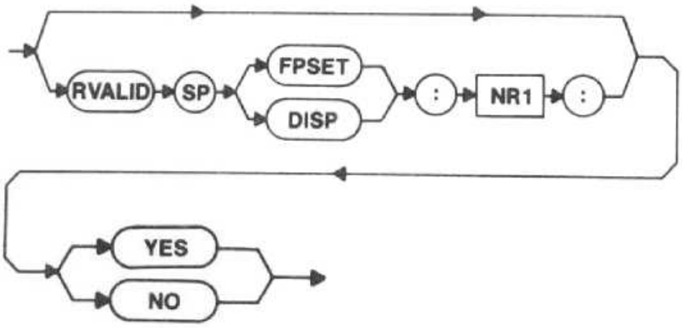


YES is returned if the indicated register contains a valid front-pannel setting or display, and NO is returned if it does not.

Examples—RVALID? FPSET:2
RVA? FPS:7
RVA? DISP:7

Range — 0–9 if FPSET is used, and 0–8 if DISP is used.

Response to RVALID query



Interaction — If HDR is OFF, thed 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.

There is no RVALID command.

MARKER SYSTEM

The digital storage functions (described in Section 4 of this manual) must be on for the marker(s) to be viewable. The Primary marker (single-marker mode) displays marker frequency and amplitude. A Secondary marker is added to the Primary marker in the delta-marker mode, and the difference in frequency and amplitude between the two markers is displayed. In the delta-marker mode, the Primary marker is the brighter of the two.

The GPIB marker commands in this section are divided into four categories; system control, marker positioning, marker finding, and miscellaneous.

Use in Macros

Most of the marker system commands in this section can be incorporated into macros designed to your specific needs. (No queries can be used within macros.) Since there is a total of 8k bytes of memory dedicated for macro use, it is important that you know the number of bytes used for each command, and keep this in mind while preparing macros. This maximum number of bytes used is included with the commands in this section; and there is also a table on the pullout page at the back of this manual that lists all available spectrum analyzer commands and the bytes used by each.

NUM Argument Values

Unless otherwise stated, the values for the NUM argument are

- 1 = ON
- ⩾+0.5 are rounded to 1
- 0 = OFF
- <+0.5 are rounded to 0

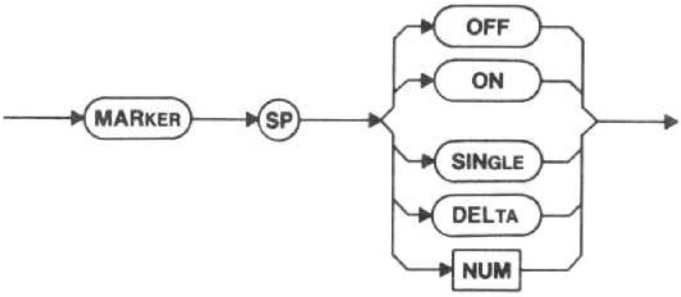
WAVEFORM FINDING

The spectrum analyzer has two sets of waveform-finding commands; five commands are described in this section and two are described in Section 8 of this manual. The MRGTNX and MLFTNX marker positioning commands move the Primary marker, and RGTNXT and LFTNXT waveform processing commands move the invisible display data point. The Primary marker is specified and reported in frequency and amplitude, and the display data point is specified and reported in screen units. The two locations (marker and data point) and the two sets of commands are independent unless the Primary marker and the display data point are coupled with the MCPOIN command. The DPMK command moves the display data point to the Primary marker location without coupling the two, and MKDP moves the Primary marker to the horizontal location of the display data point, also without coupling the locations.

SYSTEM CONTROL

The system control commands turn on the marker mode (MARKER); set the Primary or Secondary marker on a trace (MTRACE); assign a marker function to the front-panel ASSIGN 1 or ASSIGN 2 pushbutton (M1ASGN or M2ASGN); normalize the Primary marker amplitude readout to the resolution bandwidth (NSELVL); and keep the Primary marker signal at center screen (SGTRAK).

MARKER (marker mode) command



OFF — The marker is turned off.

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

DELTA — The delta-marker is turned on.

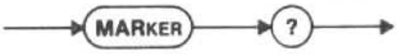
NUM—0 = OFF
1 = ON or SINGLE
2 = DELTA

Macro Memory Used — 2 bytes.

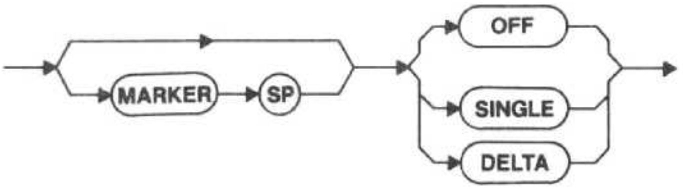
Power-up value — OFF

Interaction — 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.

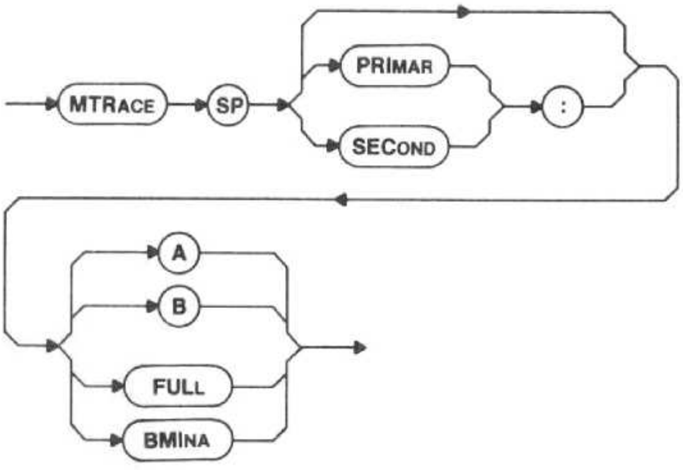
MARKER (marker mode) query



Response to MARKER query



MTRACE (marker trace position) command



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

MTRACE or **MTRACE PRIMAR** — The Primary marker is set.

MTRACE SECOND — The Secondary marker is set.

Use A, B, FULL, or BMINA to place the selected marker on the designated trace.

Macro Memory Used — 3 bytes.

Examples — MTRACE B
MTR PRI:A
MTRACE SECOND:BMINA

Power-up value — If the marker system is not turned on with the MTRACE command, the marker location will be assigned according to the settings of the digital storage commands as shown in Table 5-1.

Table 5-1

MARKER TRACE ORGANIZATION

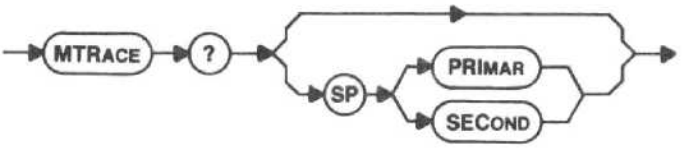
VIEW A	VIEW B	SAVE A	B–SAVE A	PRIM. MARK. ON	SECOND. MARK. ON
Off	Off	Off	Off	FULL[a]	FULL[a]
Off	Off	On	Off	A[a]	A[a]
Off	Off	On	On	B–SAVEA	B–SAVEA
Off	On	Off	Off	Full	Full
Off	On	On	Off	B	B
Off	On	On	On	B	B
On	Off	Off	Off	Full	Full
On	Off	On	Off	A	A
On	Off	On	On	B–SAVE A	B–SAVE A
On	On	Off	Off	Full	Full
On	On	On	Off	B	A
On	On	On	On	B	A

[a] **Not applicable. Since no digital storage traces are being viewed, there is no visible marker. The listed trace is that for which marker readouts are given.**

Interaction —

- 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, according to Table 5-1.
- 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.

MTRACE (marker trace position) query

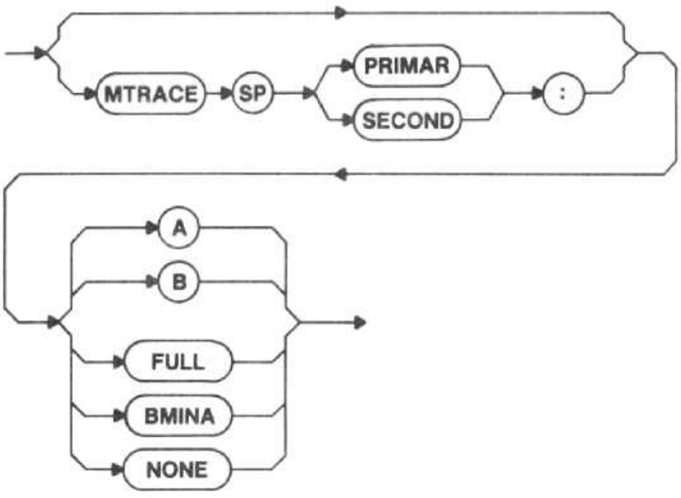


MTRACE? or MTRACE? PRIMAR — The trace containing the Primary marker is returned.

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

Examples — MTRACE?
MTR? SEC

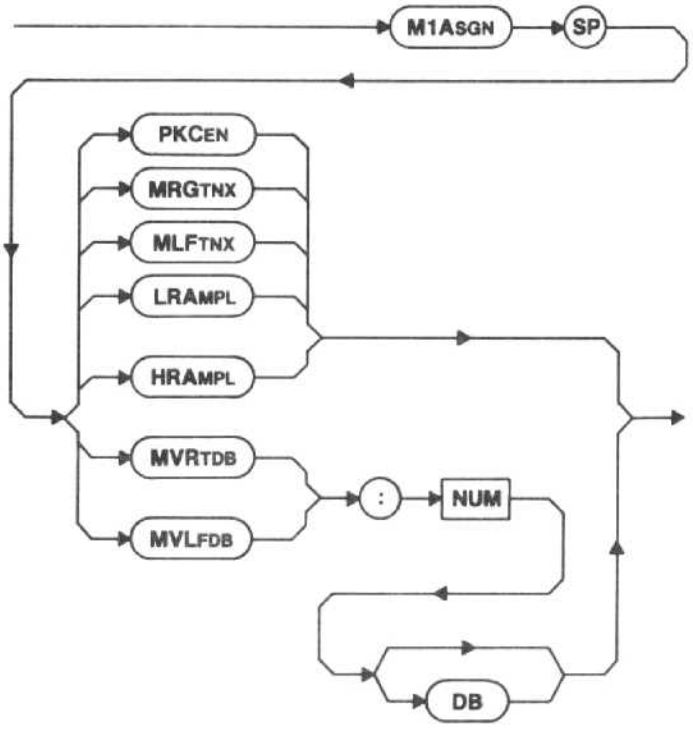
Response to MTRACE query



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.

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

M1ASGN (assign marker pushbutton 1) command



The M1ASGN command assigns the marker function that will be called up with the front-panel ASSIGN 1 pushbutton. For the description of each argument, see the description of the command whose mnemonic is the same as the argument.

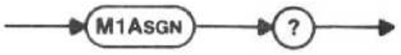
Macro Memory Used — 4 bytes.

Examples—M1ASGN MRGTNX
M1A MVR: 100
M1ASGN MVLFDB: 80DB

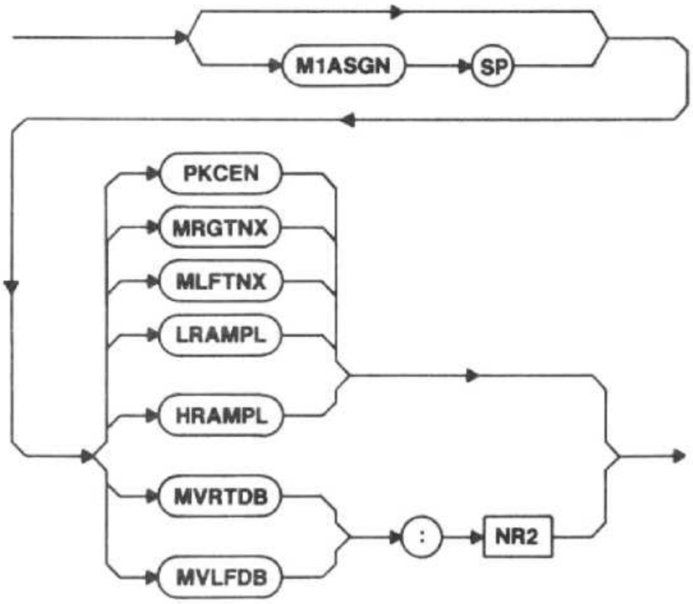
Power-up value — The assignment is stored in memory. The power-up value is the marker function that was assigned when the power to the instrument was last turned off. If no assignment has ever been made, MRGTNX is assigned.

Interaction — The M1ASGN command affects only the command that will be executed when the front-panel ASSIGN 1 pushbutton is pressed while the instrument is under local control.

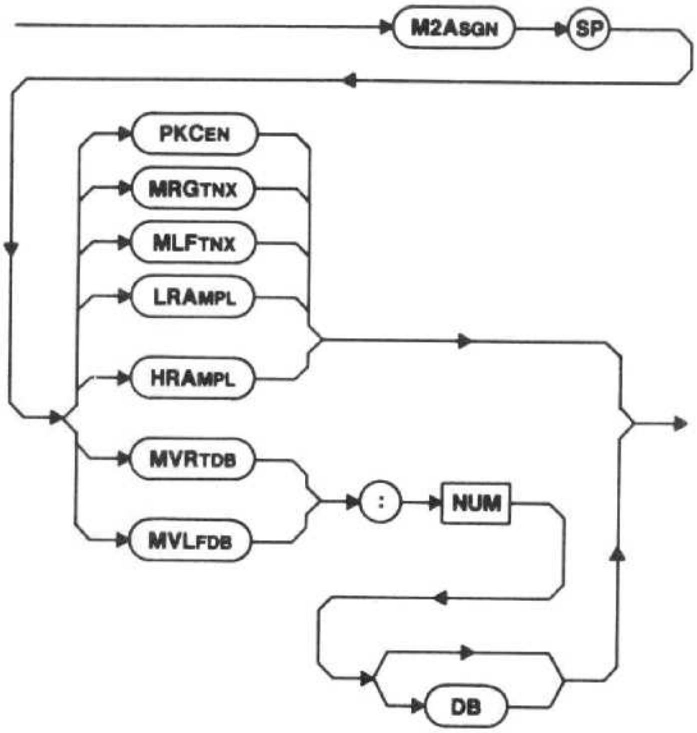
M1ASGN (assign marker pushbutton 1) query



Response to M1ASGN query



M2ASGN (assign marker pushbutton 2) command



The M2ASGN command assigns the marker function that will be called up with the front-panel ASSIGN 2 pushbutton. For the description of each argument, see the description of the command whose mnemonic is the same as the argument.

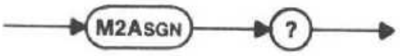
Macro Memory Used — 4 bytes.

Examples—M2ASGN MRGTNX
M2A MVR: 100
M2ASGN MVLFDB: 80DB

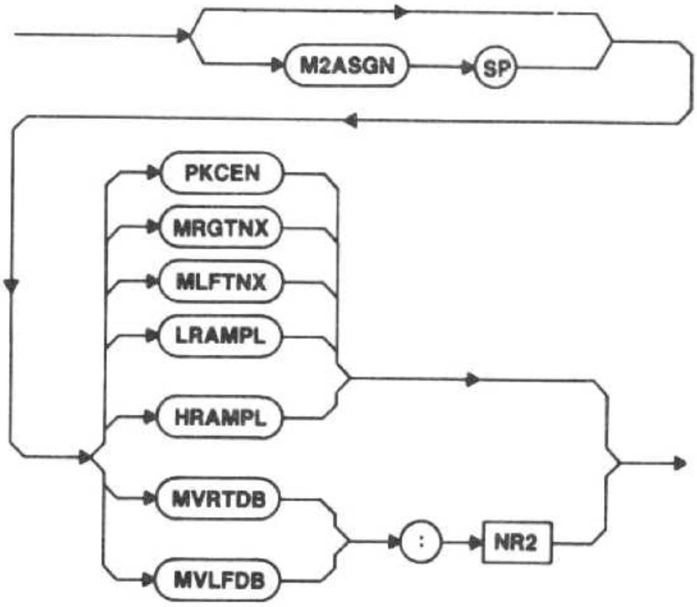
Power-up value — The assignment is stored in memory. The power-up value is the marker function that was assigned when the power to the instrument was last turned off. If no assignment has ever been made, MLFTNX is assigned.

Interaction — The M2ASGN command affects only the command that will be executed when the front-panel ASSIGN 2 pushbutton is pressed while the instrument is under local control.

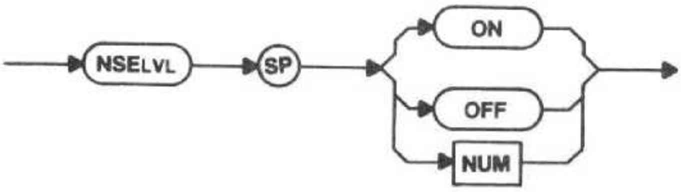
M2ASGN (assign marker pushbutton 2) query



Response to M2ASGN query



NSELVL (noise level normalization) command



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.

ON — The normalization is turned on.

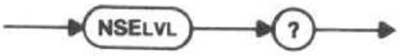
OFF — The normalization is turned off.

Macro Memory Used — 2 bytes.

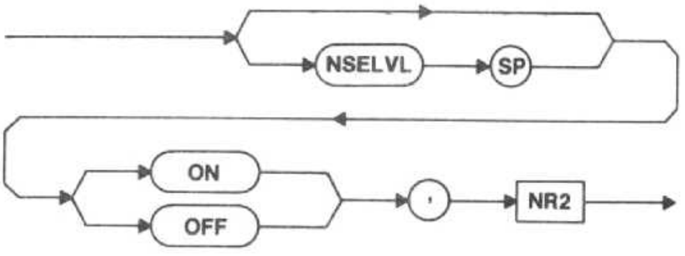
Power-up value — OFF.

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

NSELVL (noise level normalization) query



Response to NSELVL query



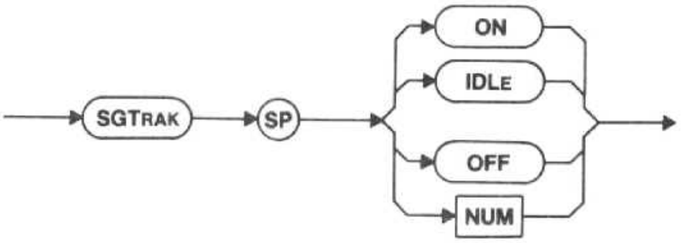
The noise level at the position of the Primary marker is returned, regardless of whether NSELVL is ON or OFF. The number is not returned with the NSELVL portion of the SET? response.

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

SGTRAK (signal track) command



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. If there is no signal at the marker location or the signal disappears, SGTRAK goes to IDLE. The signal track function takes effect at the end of the sweep after the SGTRAK command is given. SGTRAK is on during IDLE, but it is not tracking because there is no signal at the marker location.

ON or IDLE — The signal track is turned on.

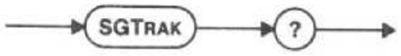
OFF — The signal track is turned off.

Macro Memory Used — 2 bytes.

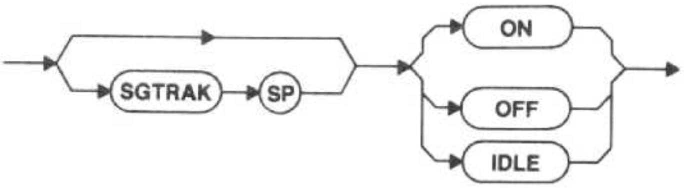
Power-up value — OFF.

Interaction — 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.

SGTRAK (signal track) query



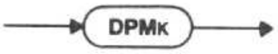
Response to SGTRAK query



MARKER POSITIONING

The marker positioning commands and queries move the display pointer to the Primary marker position (DPMK); return the amplitude of the Primary or Secondary marker or their difference (MAMPL?); tune the Primary marker frequency to center screen (MCEN); track the Primary marker with the display pointer (MCPOIN); exchange the Primary and Secondary marker positions (MEXCHG): set the Primary marker frequency (MFREQ); move the Primary marker to the display pointer horizontal location (MKDP); return the frequency and amplitude of the Primary or Secondary marker or their difference (MLOCAT?); move the marker to the reference level (MTOP); and tune the Primary marker (MTUNE).

DPMK (display pointer to marker) command



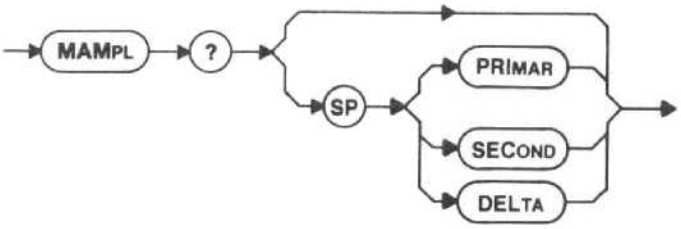
The DPMK command moves the display pointer to the Primary marker position.

Macro Memory Used — 1 byte.

Interaction — 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

There is no DPMK query.

MAMPL (marker amplitude) query



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

MAMPL? SECOND — 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

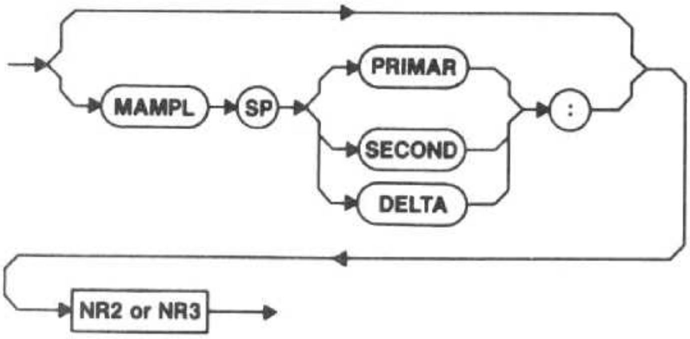
If the marker whose amplitude is being requested (or, in the case of MAMPL? DELTA, the amplitude of either marker) is out of the range of digital storage, one of the following will be returned.

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

Examples — MAMPL?
MAM? SEC

Interaction — 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 to MAMPL query



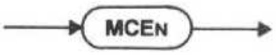
NR2 is returned in the Log Mode, and NR3 is returned in the Lin Mode.

MAMPL is not included in the response to SET?

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

There is no MAMPL command.

MCEN (marker to center) command



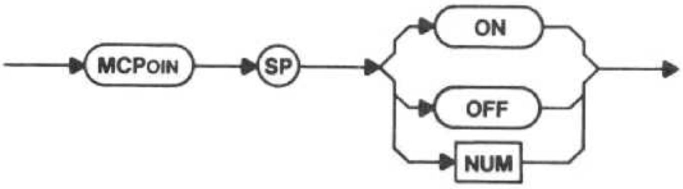
The Primary marker frequency is tuned to the center of the screen. Marker execution error message 121 is issued if the marker is not on an active trace.

Macro Memory Used — 1 byte.

Interaction — 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 in ZEROSP or MXSPN.

There is no MCEN query.

MCPOIN (marker coupled to the display pointer) command



ON — The display pointer tracks the Primary marker.

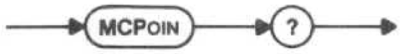
OFF — The display pointer does not track the Primary marker.

Macro Memory Used — 2 bytes.

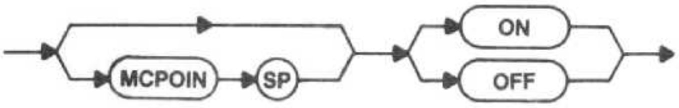
Power-up value — OFF.

Interaction — 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.

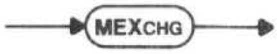
MCPOIN (marker coupled to the display pointer) query



Response to MCPOIN query



MEXCHG (marker exchange) command



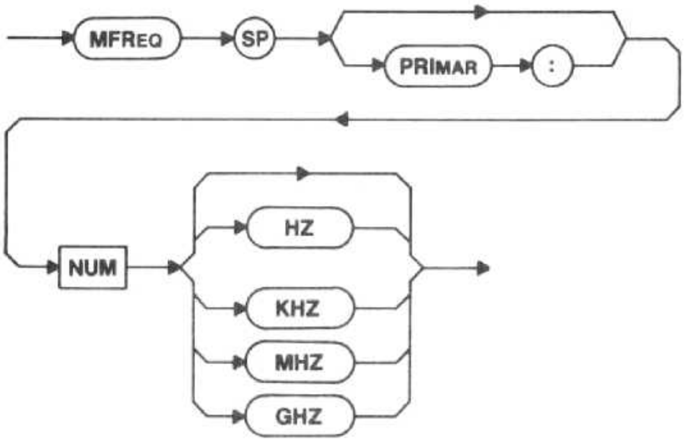
The Primary marker moves to the former location of the Secondary marker, and the Secondary marker moves to the former location of the Primary marker. If the Secondary marker is off the screen before the marker exchange, the instrument center frequency will be set to the old Secondary marker frequency, and the old Primary marker (i.e., the new Secondary marker) will be off the screen.

Macro Memory Used — 1 byte.

Interaction — MEXCHG sets MARKER to DELTA.

There is no MEXCHG query.

MFREQ (marker frequency) command



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

Macro Memory Used — 6 bytes.

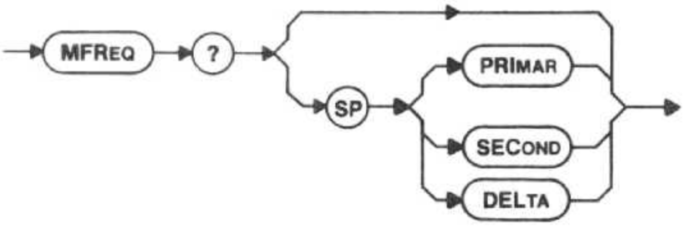
Examples — MFREQ 100000
MFR 1.8 GHZ
MFREQ PRIMAR:200 KHZ
MFR PRI:200MHZ

Range — Active trace is 0 Hz to 325 GHz
Inacative trace is limited to the edge of the screen

Power-up value — Markers are off when power is first turned on to the instrument. 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.

Interaction — 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.

MFREQ (marker frequency) query



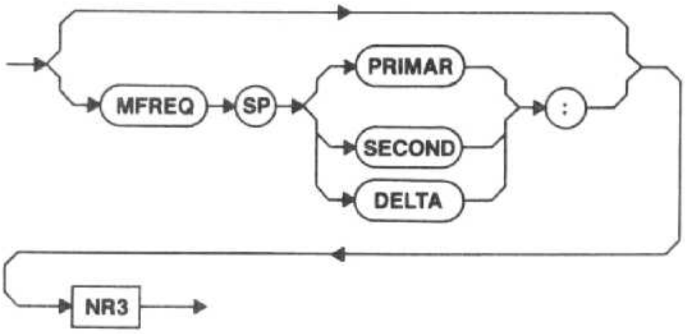
MFREQ? or **MFREQ? PRIMAR** — The frequency is returned for the Primary marker.

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.

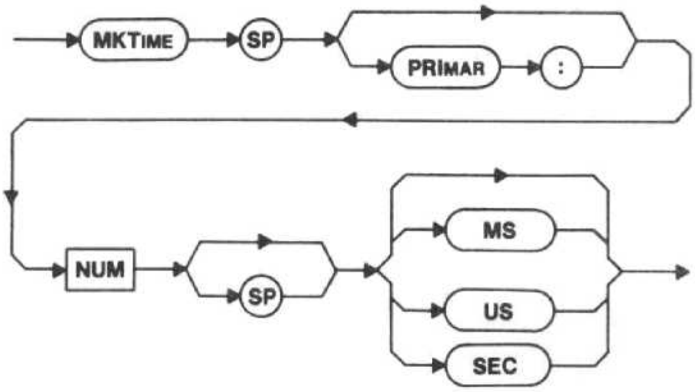
Interaction — Any MFREQ? query returns 9.999999E+99 if the requested marker or delta is not on.

Response to MFREQ query



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

MKTIME (marker time) command



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) to the value given in NUM.

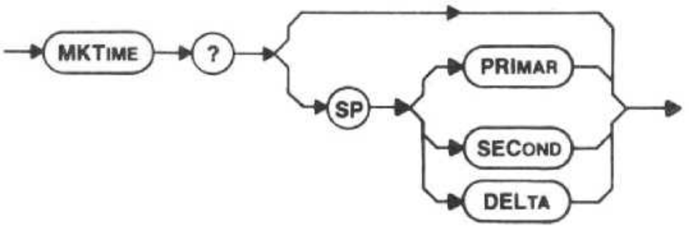
Macro Memory Used — 6 bytes.

Examples—MKTIME 1MS
MKT .1 S

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

Interaction — 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.

MKTIME (marker time) query



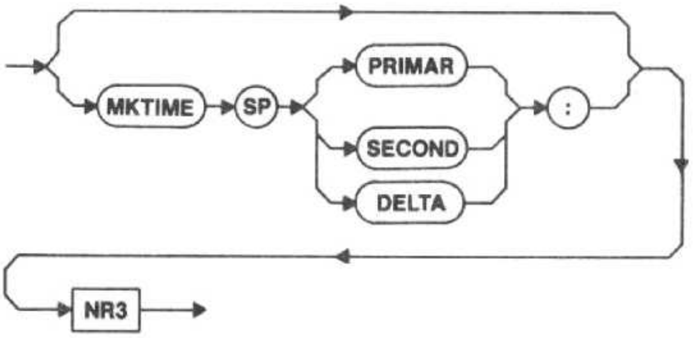
MKTIME? or **MKTIME? PRIMAR** — The time is returned for the Primary marker.

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.

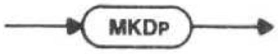
Interaction — The MKTIME query will return −200 if the time value is unavailable, +200 if the instrument is not in the zero span mode or ZETIME is OFF, and +999.99 if the marker system is off (or MARKER is not set to DELTA when the secondary or delta time is requested).

Response to MKTIME query



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

MKDP (marker to display pointer) command



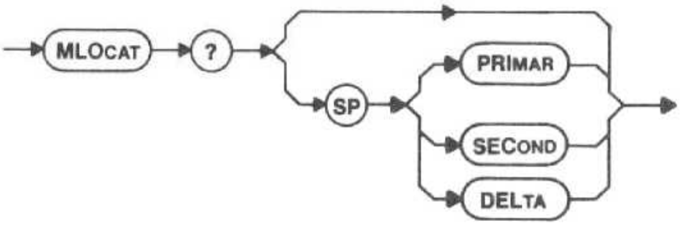
The Primary marker is moved to the same horizontal location as the display pointer.

Macro Memory Used — 1 byte.

Interaction — If MARKER is OFF, MKDP sets MARKER to SINGLE.

There is no MKDP query.

MLOCAT (marker location) query



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.

Interaction — In LOG, the amplitude is returned in reference level units. In LIN, the amplitude is returned in volts. If the frequency of the Secondary marker is off-screen, MLOCAT? SECOND and MLOCAT? DELTA use the last known Secondary marker amplitude.

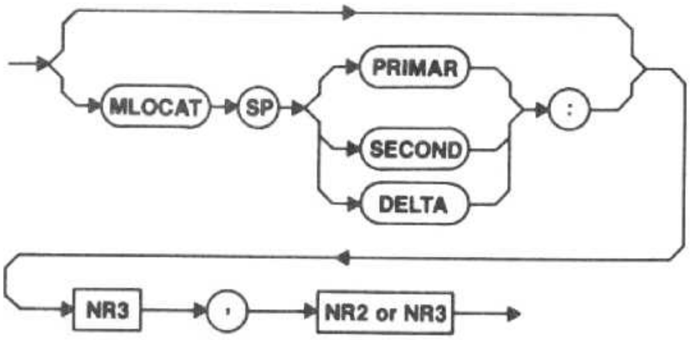
NOTE

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

- −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.

Response to MLOCAT query



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

The marker frequency is returned first, followed by the marker amplitude. If the frequency of the Secondary marker is off-screen, MLOCAT? SECOND and MLOCAT? DELTA use the last known Secondary marker amplitude.

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

There is no MLOCAT command.

MTOP (marker to reference level) command

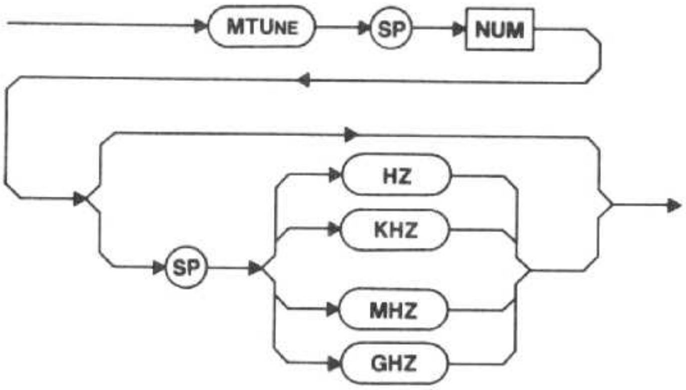
The MTOP command changes REFLVL to move the marker to the reference level (or as close as possible, given the specified vertical display and reference level accuracies).

Macro Memory Used — 1 byte.

Interaction — 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.

There is no MTOP query.

MTUNE (tune marker) command



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.

Macro Memory Used — 6 bytes.

Examples — MTUNE 100
MTU 200 MHZ

Range — The same as TUNE; refer to Table 4-2.

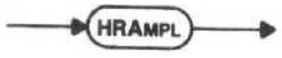
Interaction — If MARKER is OFF, MTUNE sets MARKER to SINGLE.

There is no MTUNE query.

MARKER FINDING

The marker finding commands move the Primary marker to the next higher or lower amplitude signal (HRAMPL or LRAMPL); set the bandwidth number (BWNUM); place delta markers at a given amplitude (BWMODE); move the Primary marker to the largest on-screen signal peak (MFBIG); move the Primary marker to the next signal peak to the left or the right (MLFTNX or MRGTNX); set the Primary marker to the largest or smallest vertical value in digital storage (MMAX or MMIN); set the Primary marker to the largest vertical value in digital storage that is above threshold (PKFIND); set the Primary marker to the largest vertical value in digital storage that is above threshold and tune the marker frequency to center screen (PKCEN); set the threshold for the Primary marker signal find commands MLFTNX, MRGTNX, MFBIG, HRAMPL, LRAMPL, SGTRAK, BWMODE, and PKFIND (THRHLD); move the Primary marker to the left, and down or up, or to the right, and down or up, from the present position (MVLFDB or MVRTDB); set the signal type (STYPE); and assert SRQ when the signal identification routine cannot find the requested signal (SGERR).

HRAMPL (Next higher amplitude) command

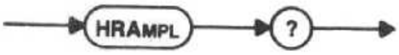


The HRAMPL command moves the Primary marker to the next higher amplitude signal on the display. If the marker is on the highest signal on the display or if no signal is found, the marker does not move.

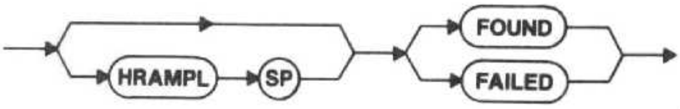
Macro Memory Used — 1 byte.

Interaction — 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.

HRAMPL (next higher amplitude) query

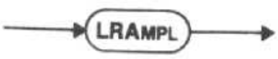


Response to HRAMPL query



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.

LRAMPL (next lower amplitude) command

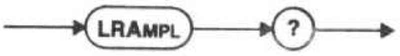


The LRAMPL command moves the Primary marker to the next lower amplitude signal on the display. If the marker is on the lowest signal on the display or a signal cannot be found, the marker does not move.

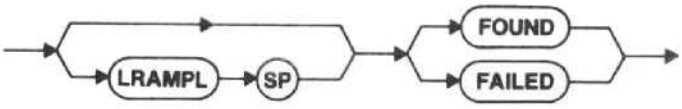
Macro Memory Used — 1 byte.

Interaction — 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.

LRAMPL (next lower amplitude) query

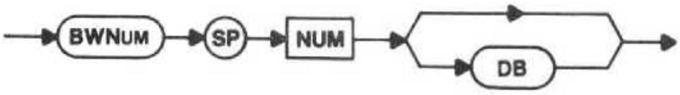


Response to LRAMPL query



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.

BWNUM (marker bandwidth number) command

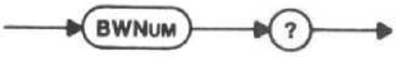


BWNUM sets the level below the signal peak used in the bandwidth mode (BWMODE) at which the bandwidth is found. This number is stored in battery-powered memory.

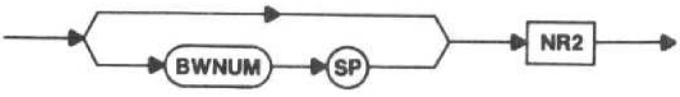
Macro Memory Used — 3 bytes.

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.

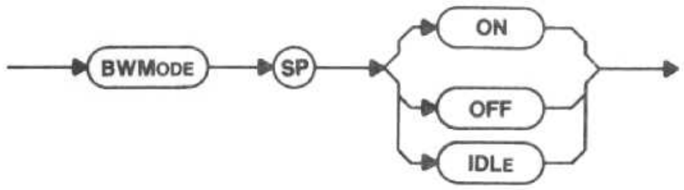
BWNUM (marker bandwidth number) query



Response to BWNUM query



BWMODE (marker bandwidth mode) command



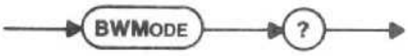
The BWMODE command moves the delta markers down from the peak of the signal that the Primary marker is on by the value set in BWNUM. (This value could also have been set from the front panel. If no value has ever been set, the value used will be 6 dB.) BWMODE moves in 1/10 dB steps. The Primary marker 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. 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. When BWMODE goes to IDLE, marker execution warning message 130 is issued if SIGERR is on.

Macro Memory Used — 2 bytes.

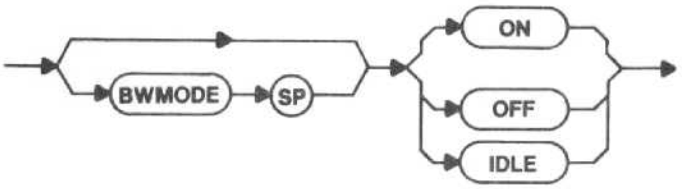
Power-up value — OFF.

Interaction — 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.

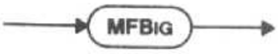
BWMODE (marker bandwidth mode) query



Response to BWMODE query



MFBIG (marker peak find) command

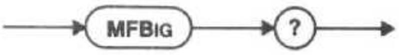


The MFBIG command moves the Primary marker to the peak of the largest on-screen signal. If no signal peak is found, the marker does not move.

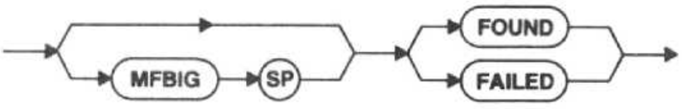
Macro Memory Used — 1 byte.

Interaction — 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.

MFBIG (marker peak find) query

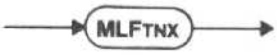


Response to MFBIG query



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.

MLFTNX (marker left next) command

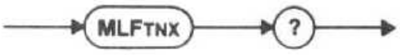


The MLFTNX command moves the Primary marker to the peak of the next signal to the left of the present marker position. If no signal peak is found, the marker does not move.

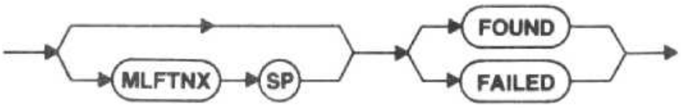
Macro Memory Used — 1 byte.

Interaction — 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.

MLFTNX (marker left next) query

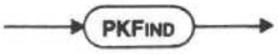


Response to MLFTNX query



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.

PKFIND (marker to maximum above threshold) command



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 the built-in intelligence. Peak B would be selected from the cluster in Figure 5-1A; peak A would be selected in Figure 5-1B because the low point (B) would stop a search from continuing to the cluster (C).

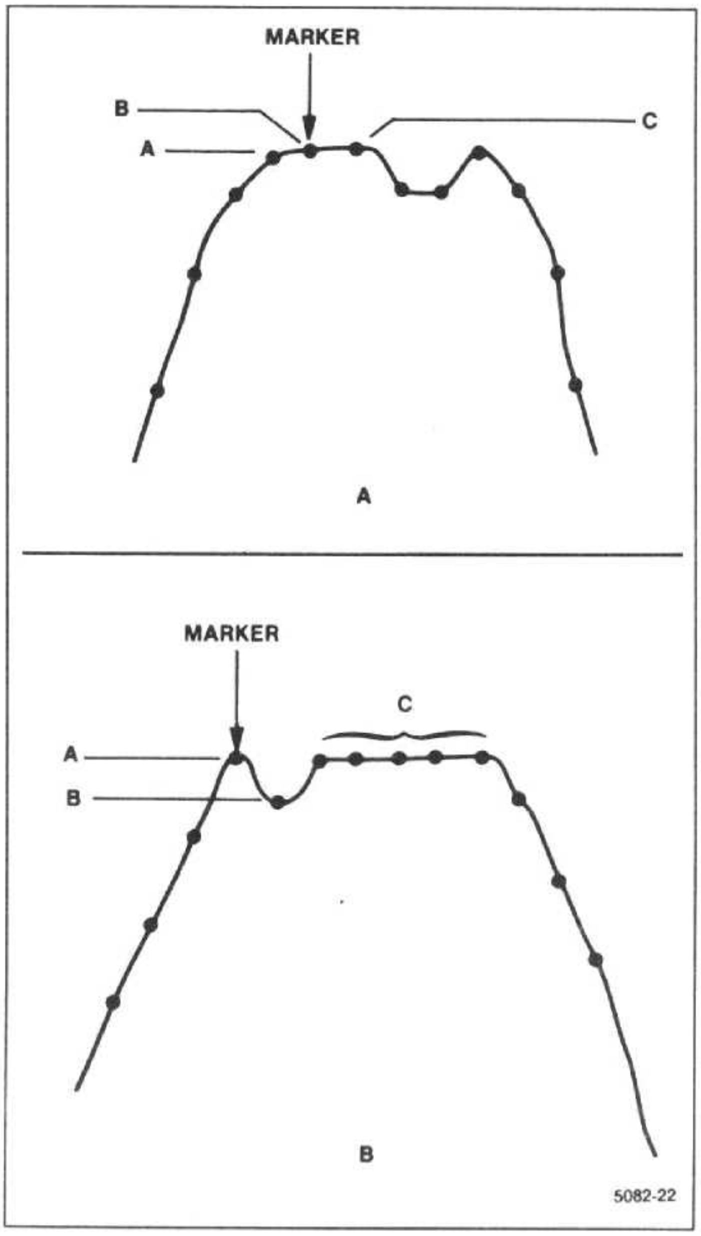
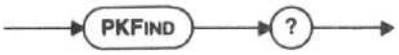


Figure 5-1. Using the PKFIND command.

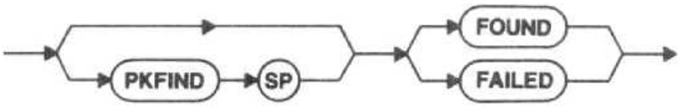
Macro Memory Used — 1 byte.

Interaction — 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.

PKFIND (marker to maximum above threshold) query

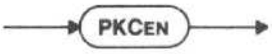


Response to PKFIND query



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.

PKCEN (marker to maximum and center) command



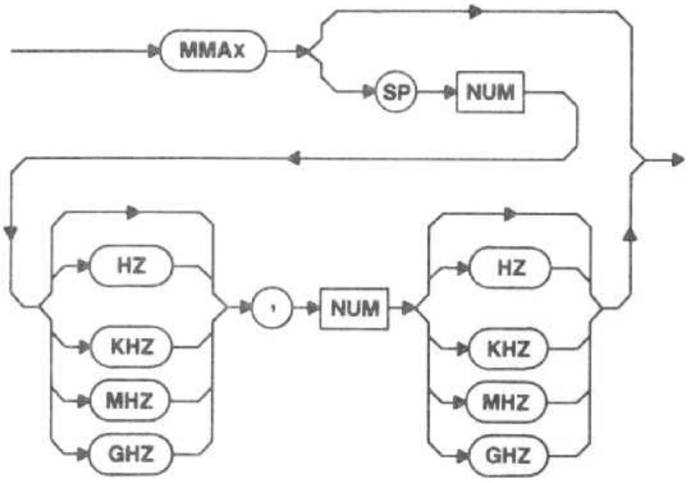
The PKCEN command is a combination of the PKFIND and MCEN commands. The Primary marker is moved to the largest left-most vertical value in digital storage if that value is above the threshold, and the marker is then tuned the marker frequency to the center of the screen. For additional information, refer to the descriptions of PKFIND and MCEN earlier in this section.

Macro Memory Used — 1 byte.

Interaction — PKCEN is not available in ZEROSP or MXSPN.

There is no PKCEN query.

MMAX (move marker to maximum) command



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 (leftmost) point is used.

NUM, 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.

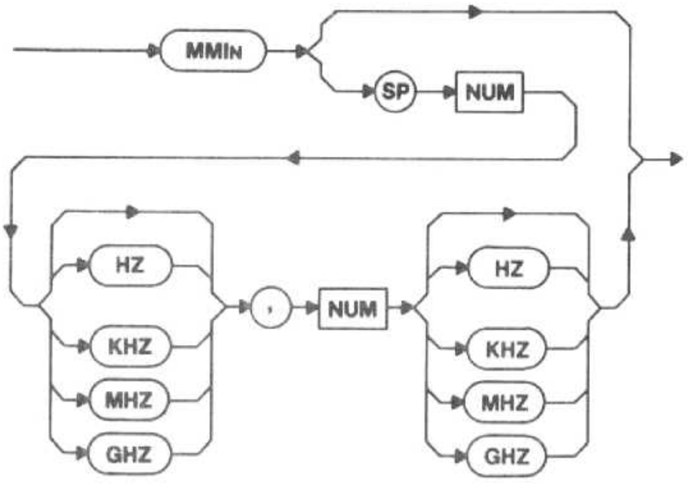
Macro Memory Used — 12 bytes.

Examples — MMAX
MMAX 15.0MHZ,19.0MHZ
MMA 15.0MHZ,19.0MHZ

Interaction — If MARKER is OFF, MMAX sets MARKER to SINGLE.

There is no MMAX query.

MMIN (move marker to minimum) command



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 (leftmost) point is used.

NUM, 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.

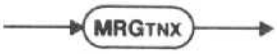
Macro Memory Used — 12 bytes.

Examples — MMIN
MMIN 15.0MHZ,19.0MHZ
MMI 15.0MHZ,19.0MHZ

Interaction — If MARKER is OFF, MMIN sets MARKER to SINGLE.

There is no MMIN query.

MRGTNX (marker right next) command

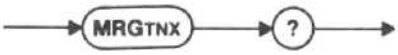


The MRGTNX command moves the Primary marker to the peak of the next signal to the right of the present marker position. If no signal peak is found, the marker does not move.

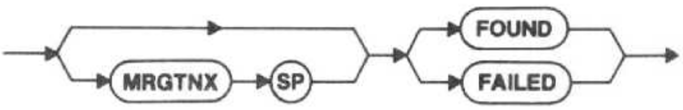
Macro Memory Used — 1 byte.

Interaction — 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.

MRGTNX (marker right next) query

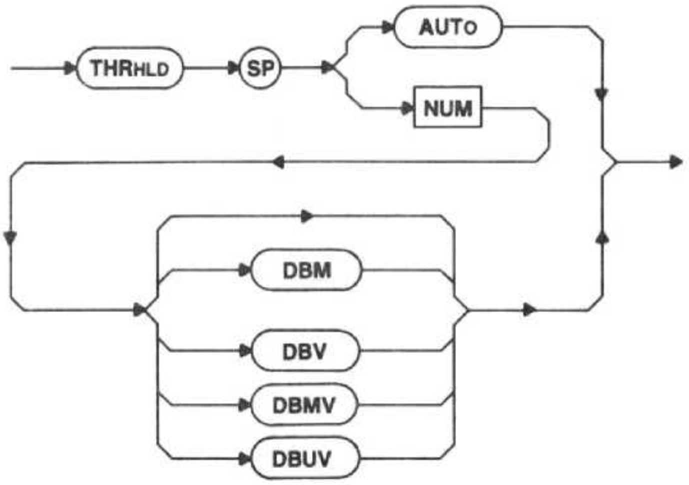


Response to MRGTNX query



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.

THRHLD (marker threshold) command



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 required 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.

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.

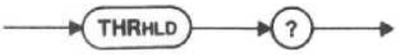
NUM — The threshold is set to level input. If no units are specified, the spectrum analyzer assumes the current reference level units.

Macro Memory Used — 4 bytes.

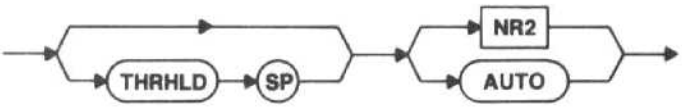
Examples — THRHLD AUTO
THRHLD −40DBMV

Power-up value — AUTO

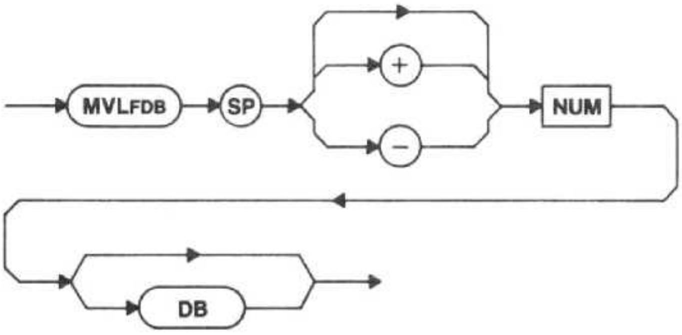
THRHLD (marker threshold) query



Response to THRHLD query



MVLFDB (move marker left x dB) command

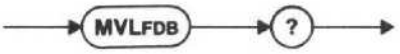


The MVLFDB command moves the Primary marker to the left and NUM DB down (negative NUM) or up (positive NUM or NUM without a sign) from the current position. If the requested amplitude cannot be found, the marker does not move.

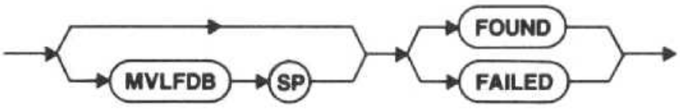
Macro Memory Used — 3 bytes.

Interaction — If MARKER is OFF, MVLFDB sets MARKER to SINGLE. If SGERR is ON, marker execution warning message 130 is issued if the requested amplitude is not found.

MVLFDB (move marker left x dB) query

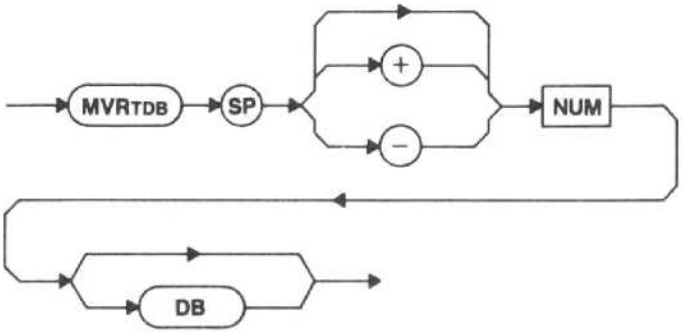


Response to MVLFDB query



FOUND is returned if the last MVLFDB command moved the marker to the requested position. FAILED is returned if the last MVLFDB command could not move the marker to the requested position. If the MVLFDB query is given before any MVLFDB command, FAILED is returned.

MVRTDB (move marker right x dB) command

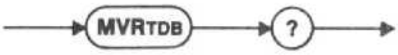


The MVRTDB command moves the Primary marker to the right and NUM DB down (negative NUM) or up (positive NUM or NUM without a sign) from the current position. If the requested amplitude cannot be found, the marker does not move.

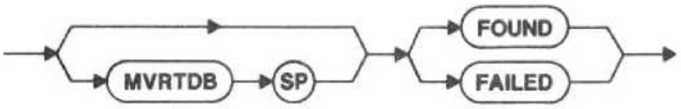
Macro Memory Used — 3 bytes.

Interaction — 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.

MVRTDB (move marker right x dB) query

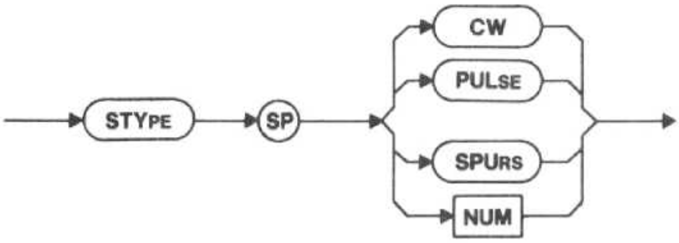


Response to MVRTDB query



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.

STYPE (signal type) command



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

Figure 5-2 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 looks at 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 5-2, 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. At the end of this Marker Functions portion are five illustrations showing the use of this signal finding command.

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.

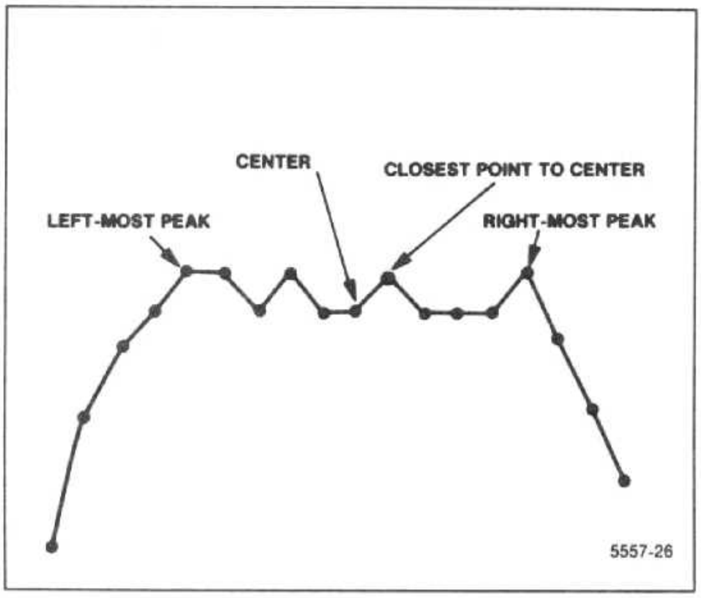


Figure 5-2. Locating the signal peak.

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 particular 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 5-3 through 5-7 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 5-3, 5-4, and 5-5 — 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 5-6 — 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 (i.e., the selections would be A, B, C, D, and E).

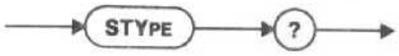
Figure 5-7 — In Figure 5-7 the threshold is assumed to be −70 dBm. If CW was selected, signals B, E, F, and G would be identified. The other signals would not be identified because they do not meet the minimum bandwidth criteria. If Pulse was selected, signals A, B, D, E, F, and G would be identified. 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.

Macro Memory Used — 2 bytes.

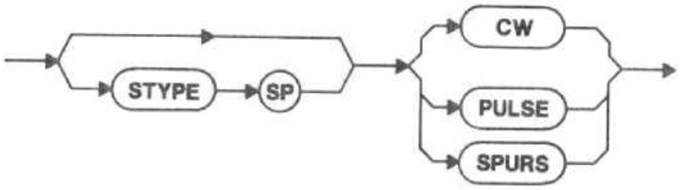
Power-up status — CW.

Interaction — The STYPE command affects the marker finding commands MFBIG, MLFTNX, MRGTNX, HRAMPL, and LRAMPL.

STYPE (signal type) query



Response to STYPE query



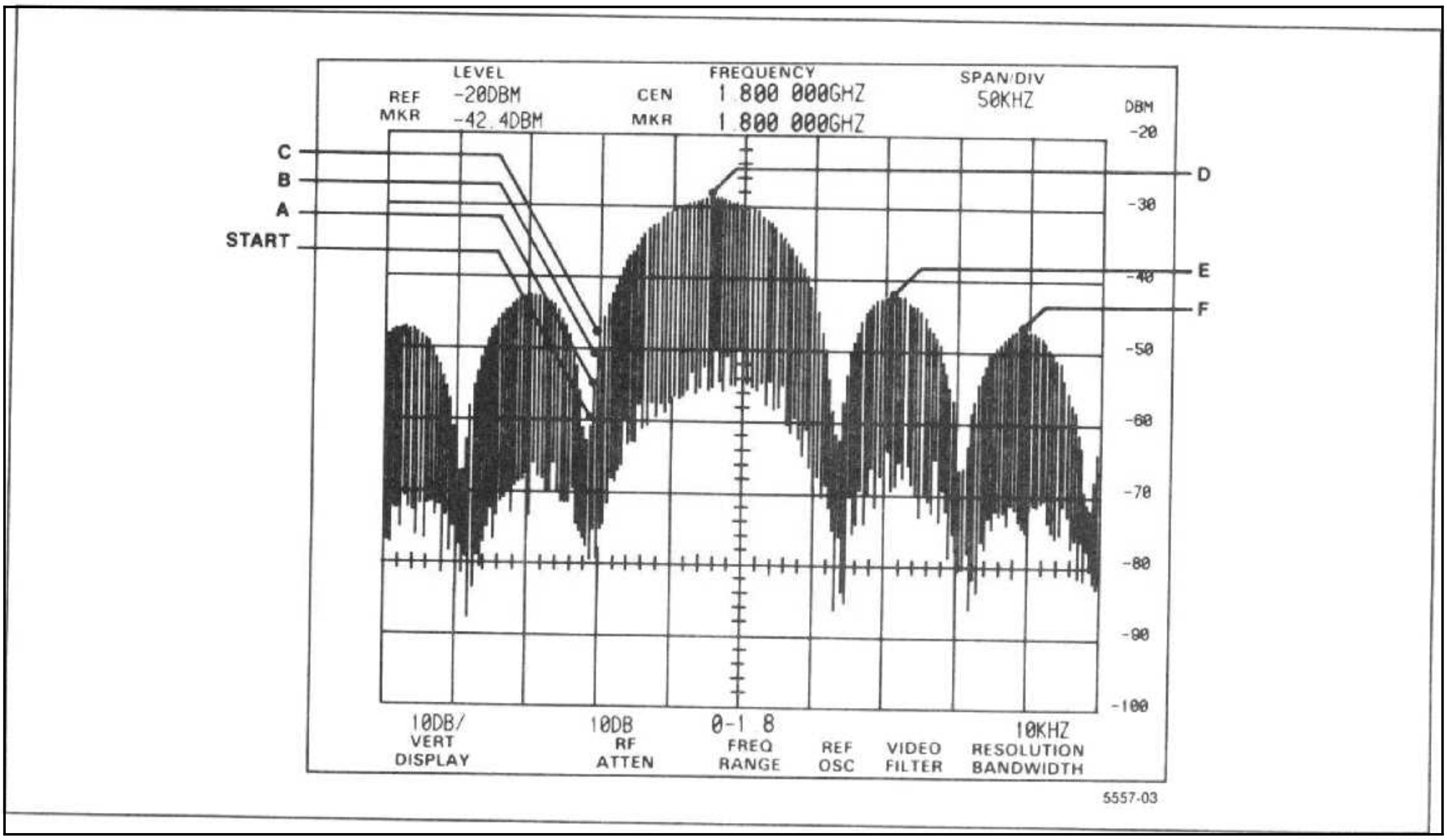


Figure 5-3. Signal finding example.

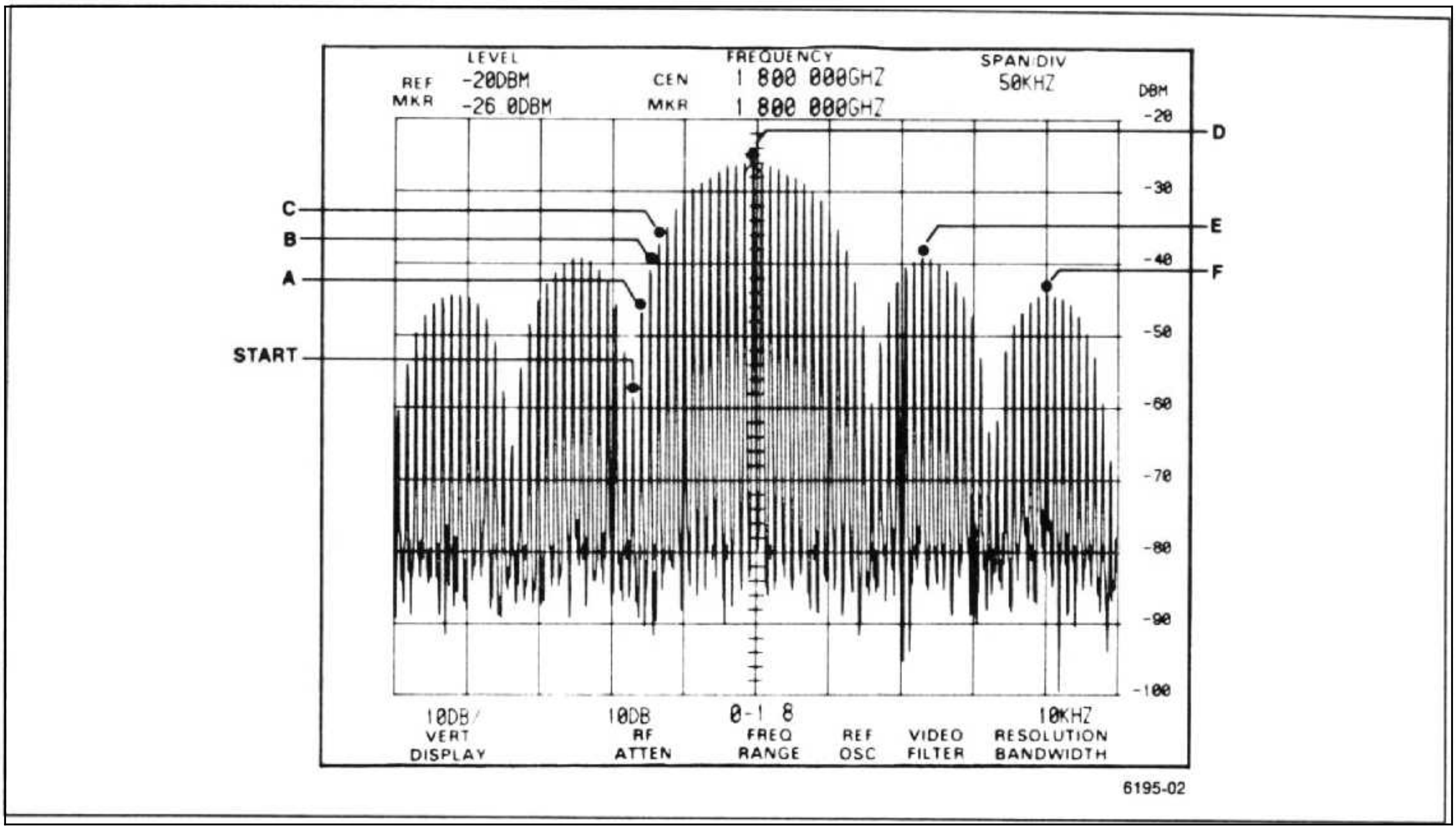


Figure 5-4. Signal finding example.

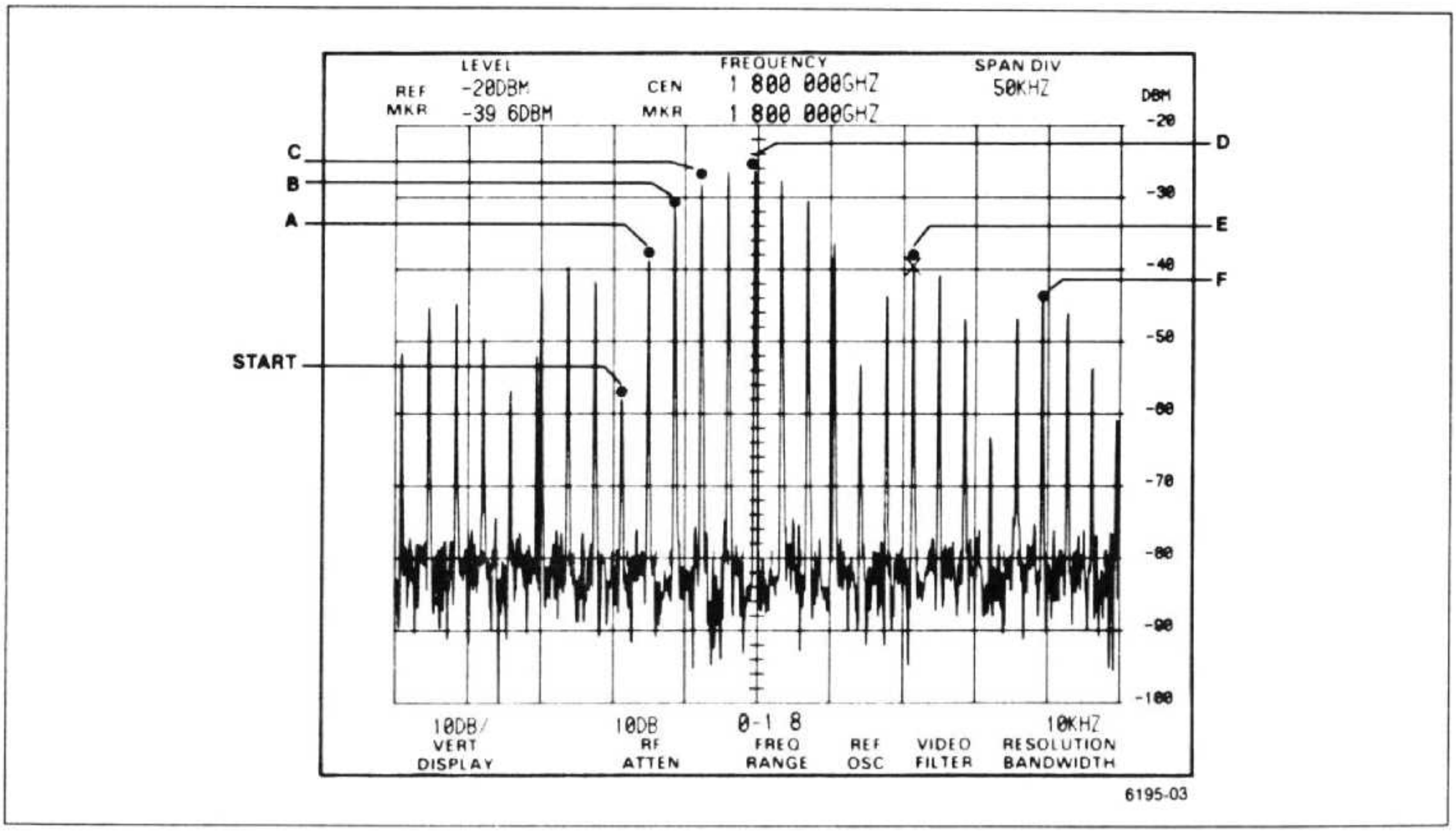


Figure 5-5. Signal finding example.

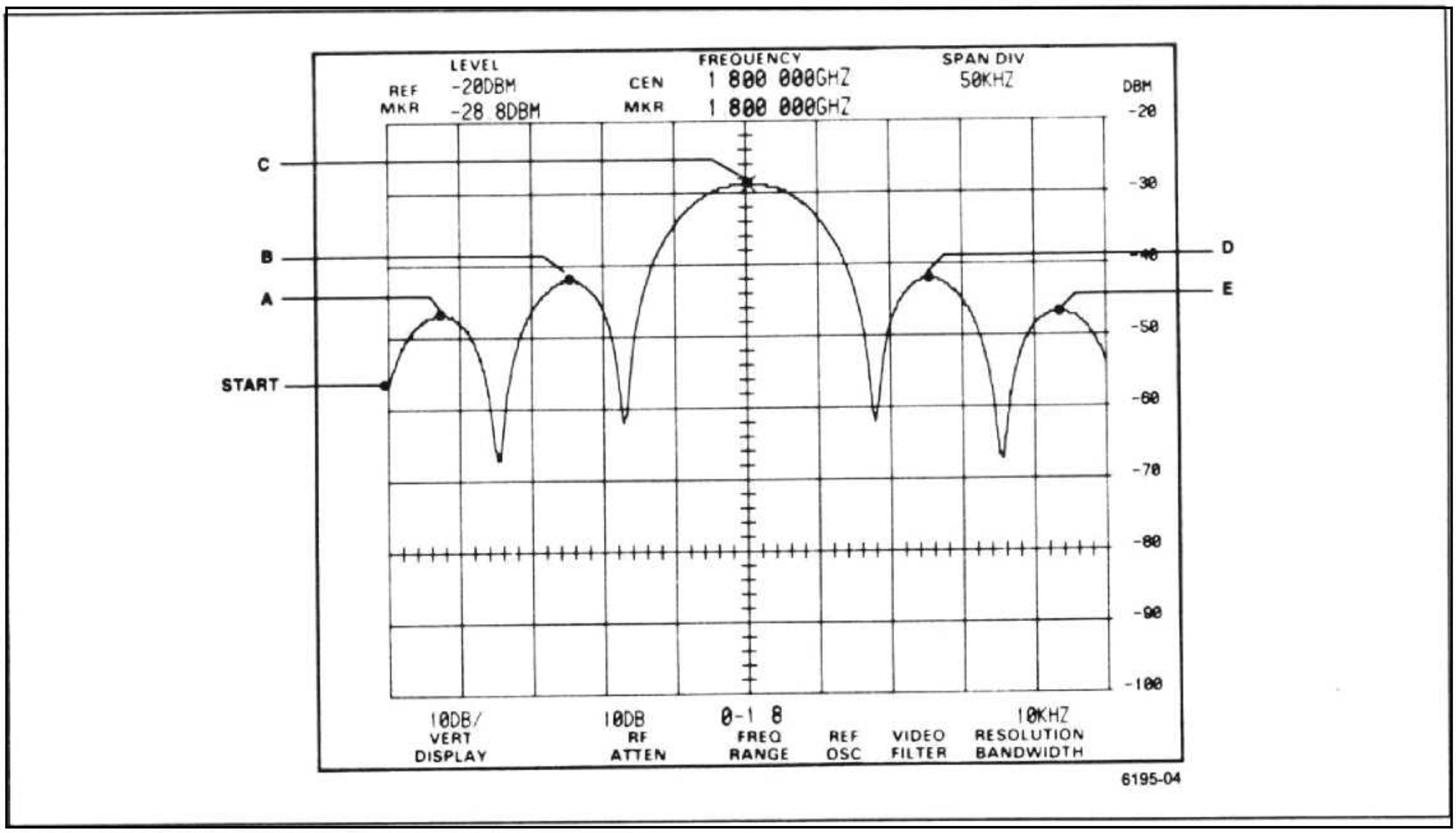


Figure 5-6. Signal finding example.

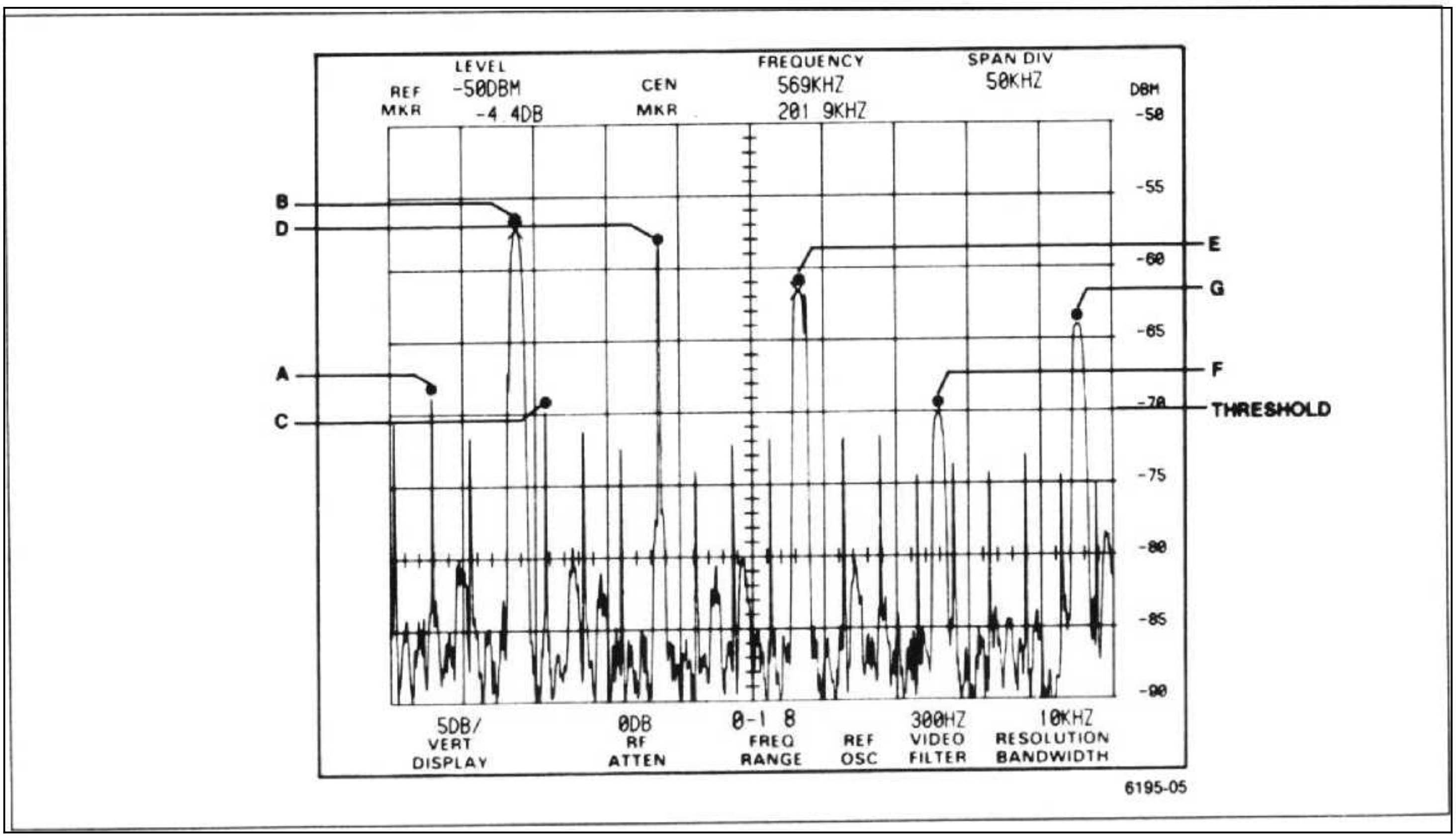
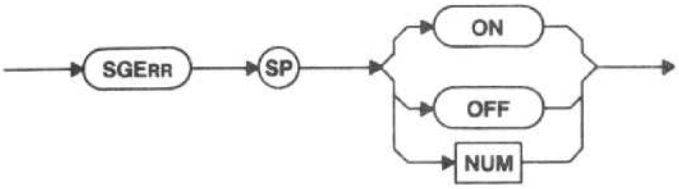


Figure 5-7. Signal finding example.

SGERR (signal find error) command



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 MVLFDB commands.
- The requested bandwidth cannot be found by the BWMODE command.

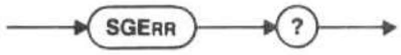
OFF — The spectrum analyzer does not assert SRQ when any of the above commands fail.

Power-up value — OFF.

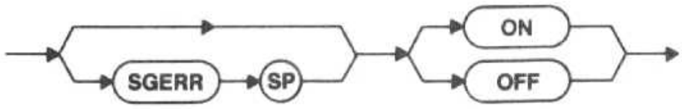
Macro Memory Used — 2 bytes.

Interaction — RQS must be on for marker execution warning message 130 to be issued.

SGERR (signal find error) query

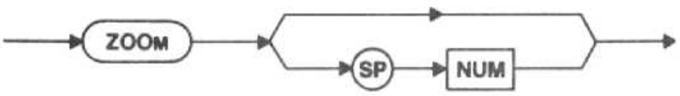


Response to SGERR query



MISCELLANEOUS

ZOOM command



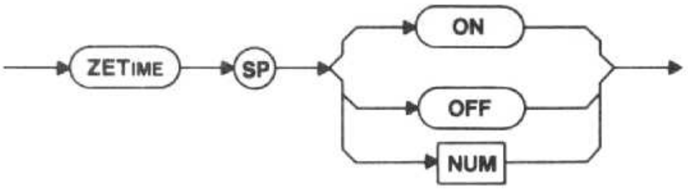
The ZOOM command moves the Primary marker frequency to the center frequency and sets the SPAN (frequency span/division) to the next smaller span/division, if possible, in the front-panel 1-2-5 sequence. If the optional number argument is given, the span/division is reduced NUM times. Numbers less than 1 are rounded to 1. Execution warning message 111 is issued if the spectrum analyzer defaults to the lowest span/division because the span could not be reduced the requested number of times.

Macro Memory Used — 2 bytes.

Interaction — 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.

There is no ZOOM query.

ZETIME (zero span) command



ON — In zero span, the marker readout is amplitude and time.

OFF — In zero span, the marker readout is amplitude and frequency.

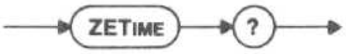
Macro Memory Used — 2 bytes.

Power-up value — OFF.

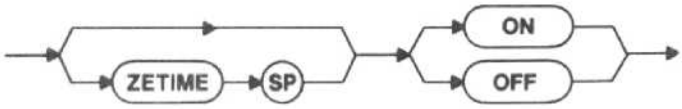
Interaction — The ZETIME command has no effect when the instrument is in a non-zero span. If the markers are on different traces, the Secondary marker will move to the Primary marker trace under either of the following conditions. Under the same conditions, if the Secondary marker is off-screen, it will 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

ZETIME (zero span) query



Response to ZETIME query



MACROS

The macroinstruction (macro) commands in this section are divided into the seven categories of math, register, branching and looping, print, data, and general commands. The branching and looping commands are available only within macros; they cannot be used outside of macros.

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 macros will be lost if the battery power to the memory is interrupted (as when the battery is removed for long-term storage).

If any GPIB command arrives over the bus while a macro is running, the macro is stopped. The RUN command (or RUN/STOP from the front panel) will restart the macro. A GPIB query has no affect on macros.

There are 8 k bytes of memory dedicated for macro use. Although there is no set limit for each macro, it is important that you know the number of bytes used for each command and keep this in mind while preparing macros. The number of bytes used for each command is included with the commands in this section; and there is also a table in the pullout pages at the back of this manual that lists all available spectrum analyzer commands and the bytes used by each.

Notes on the Program Examples

Many combinations of languages and controllers may be used for programming the spectrum analyzer over the GPIB. The examples in this manual are given in the BASIC language; specifically, in the version of BASIC used by the Tektronix 4041 System Controller. Other versions of BASIC will probably have similar syntax.

Some controllers may require additional statements to set up data transfer over the GPIB. For example, 4041 BASIC contains optional OPEN and SET DRIVER statements that set GPIB communications parameters. These statements are not included in the program examples because the default parameter values are suitable for simple programs, such as the examples. The statements may be helpful in writing programs involving multiple I/O channels, such as tape files or other GPIB-controller instruments. Refer to the manual of the controller you are using for the statements needed in your application.

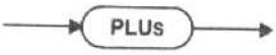
In these examples, the spectrum analyzer's primary address is assumed to be 1. See Section 1 of this manual for instructions on how to set the GPIB address switches.

Some of the lines in the examples extend beyond the column width limitations. When this happens the line is broken where a natural space occurs, and the remainder of the line is indented on the immediately following line. Because of this natural space, a space must be added between the two example lines when joining them to form one program line.

MATH COMMANDS

The math commands add the X and Y registers (PLUS); subtract the X register from the Y register (SUBT); multiply the X and Y registers (MULT); and divide the Y register by the X register (DIVIDE). In case of an overflow, the result in XREG is set to the maximum value.

PLUS (add X and Y registers) command



The PLUS command adds the contents of XREG to the contents of YREG and puts the result in XREG (e.g., XREG = XREG + YREG).

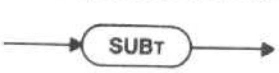
Macro Memory Used — 1 byte.

Range of Result — ±9.999 999 999 999 E+99 to ±9.999 999 999 999 E−99.

Interaction — The YREG is unchanged after the PLUS command.

There is no PLUS query.

SUBT (subtract X from Y register) command



The SUBT command subtracts the contents of XREG from the contents of YREG and puts the result in XREG (e.g., XREG = YREG − XREG).

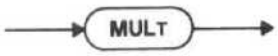
Macro Memory Used — 1 byte.

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

Interaction — The YREG is unchanged after the SUBT command.

There is no SUBT query.

MULT (multiply X and Y registers) command



The MULT command multiplies the contents of XREG with the contents of YREG and puts the result in XREG (e.g., XREG = XREG × YREG).

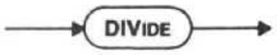
Macro Memory Used — 1 byte.

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

Interaction — The YREG is unchanged after the MULT command.

There is no MULT query.

DIVIDE (divide Y by X register) command



The DIVIDE command divides the contents of YREG by the contents of XREG and puts the result in XREG (e.g., XREG = YREG ÷ XREG).

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.

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

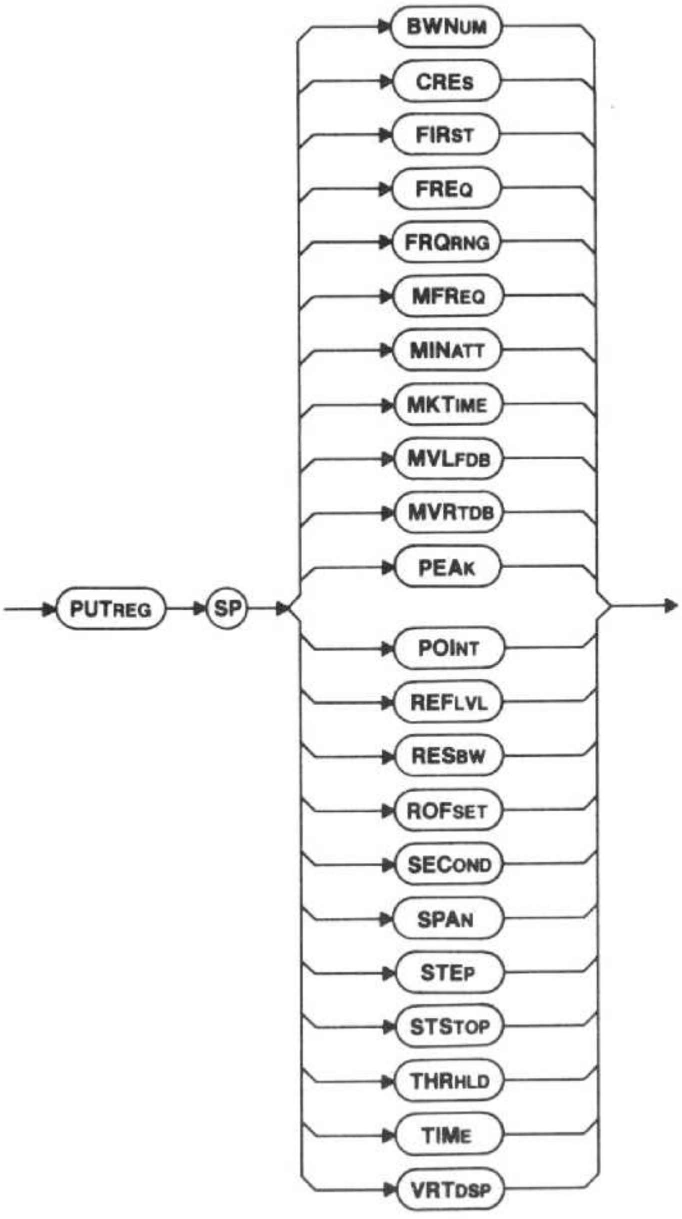
Interaction — The YREG is unchanged after the DIVIDE command.

There is no DIVIDE query.

REGISTER COMMANDS

The register commands put the number in X register into a setting (PUTREG); exchange the contents of X register and Y register (EXCHG); convert the number in X register to an integer (INTEGR); put the contents of Y register into X register (POP); and enter a value into the X register (ENTER).

PUTREG (put X register into a setting) command



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—See Example Descriptions Below

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

Example Descriptions

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

2. Sets the resolution bandwidth to the value in XREG.

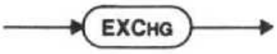
3. Puts the contents of YREG into START FREQ and the contents of XREG into STOP FREQ.

4. Puts the value of XREG into X and calls the move right X dB routine.

Macro Memory Used — 2 bytes.

There is no PUTREG query.

EXCHG (exchange X and Y registers) command



The EXCHG command exchanges the current contents in XREG and YREG.

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.

There is no EXCHG query.

INTEGR (convert X register to an integer) command

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

Examples

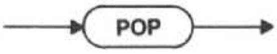
XREG CONTENTS

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.

There is no INTEGR query.

POP (put Y register into X register) command



The POP command duplicates the current contents of YREG and puts this into XREG.

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.

Interaction — YREG will remain unchanged after the POP command.

There is no POP query.

ENTER (enter value in X register) command

The ENTER command syntax diagram is on the following page.

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).

Sample Entries

ENTER POINT — The X POINT value will be put into XREG and the Y POINT value will be put into YREG.

ENTER 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 (e.g., 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 (e.g., DISBUF(NUM), DISBUF(XREG), DISBUF(YREG), DISBUF(VAR(INDEX)). Execution error message 176 is issued either if the overall index for DISBUF (the part following the first colon (:) is outside the range of 1–1000 or 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.

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

Examples—ENTER MFREQ
ENTER MAMPL:SEC
ENTER VAR:YREG
ENTER DISBUF:VAR:NUM
ENTER 100
ENTER 100MHZ

Macro Memory Used — 10 bytes.

VAR Range — 1 to 30

DISBUF range — 1 to 1000

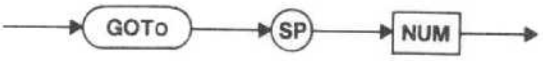
Interaction — Before the value of XREG is changed, the value in XREG is copied into YREG.

There is no ENTER query.

BRANCHING AND LOOPING COMMANDS

The branching commands will go to a LABEL (GOTO); label a point in the macro (LABEL); create macro looping (FOR); perform action if statement is true (IF,); return from a subroutine (RETURN); and go to a subroutine (GOSUB). The branching and looping commands are available only within macros; they cannot be used outside of macros. Although not discussed separately, there are companion commands included here; NEXT works with the FOR command, and ELSE and ENDI work with the IF command.

GOTO (go to LABELed line) command



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

Macro Memory Used — 5 bytes.

Range — 1 to 100.

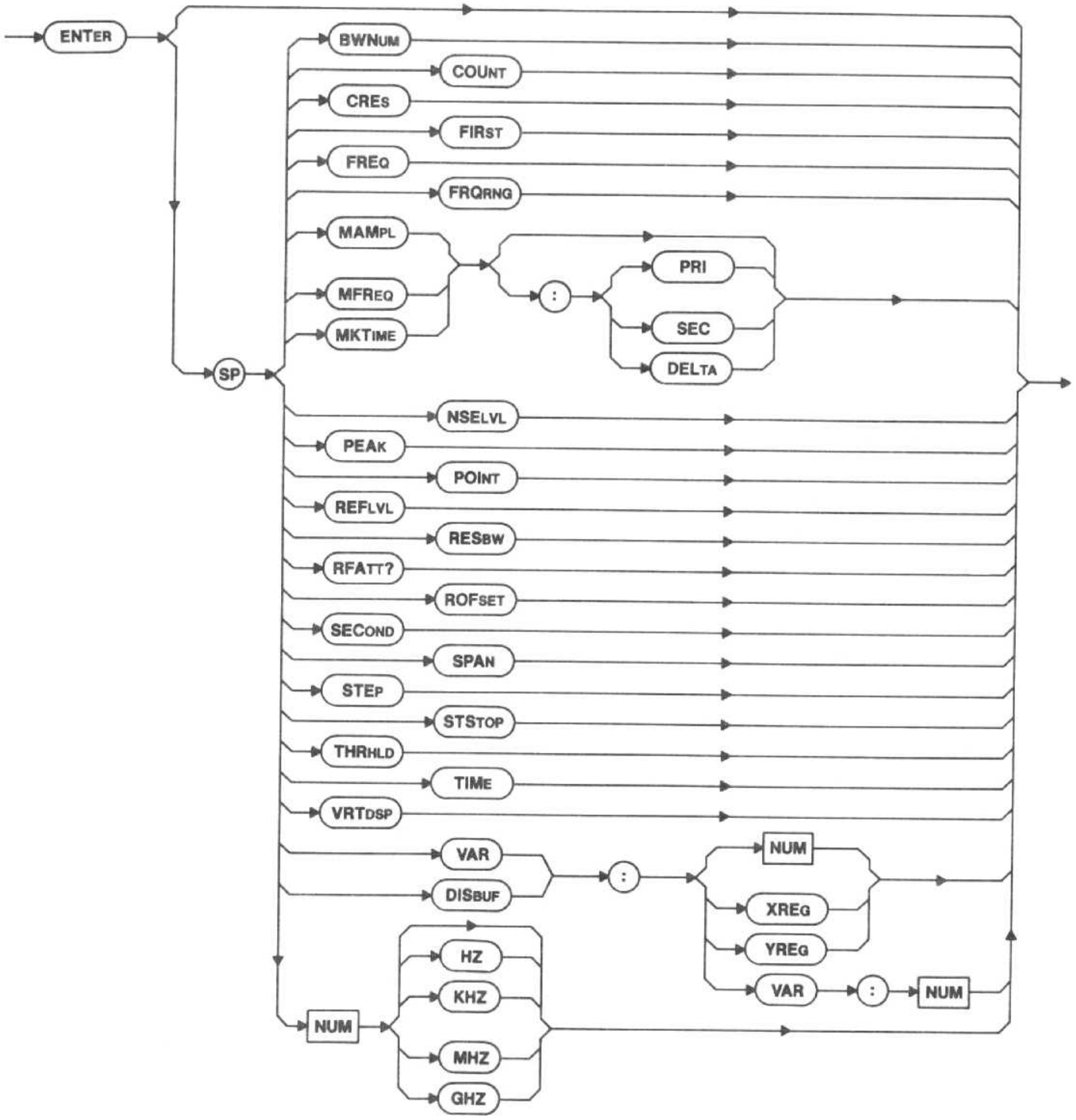
If the macro is instructed to go to a label that does not exist, macro execution error message 170 is issued. Error message 170 will only be seen after the EMAC command when the macro is compiled and errors are located.

NOTE

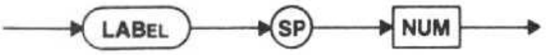
> The last command in a macro before the EMAC command must be GOTO or RETURN or DONE. If it is not, macro execution error message 178 is issued.

There is no GOTO query.

ENTER (enter value in X register) command syntax diagram



LABEL (label point in macro) command



NUM — Sets the reference point for the GOTO and GOSUB commands.

Macro Memory Used — 2 bytes.

Range — 1 to 100

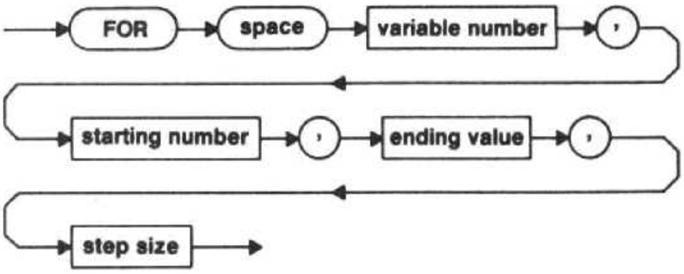
There is no LABEL query.

FOR (variable = X to Y step Z) command (and NEXT)

The FOR command syntax diagram is on the next column.

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

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

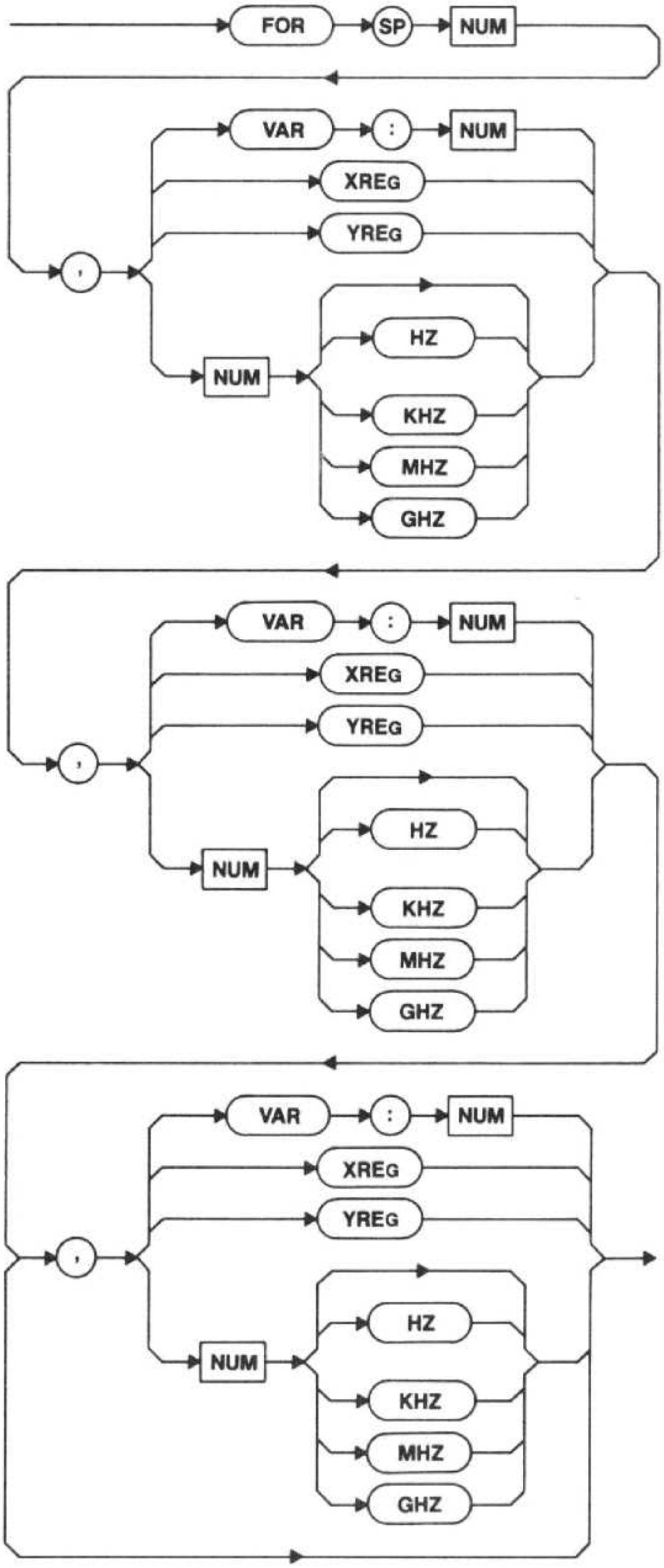


If the ending value and/or the step size is taken from the X register, the Y register value changes, or a variable change, during execution of the FOR statement have no effect. The value(s) at the start of the loop are used.

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)

FOR (variable = X to Y step Z) command syntax diagram



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(a)=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.

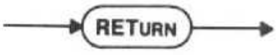
Macro Memory Used — FOR is 20 bytes, and NEXT is 5 bytes.

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

Interaction — 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.

There is no FOR query.

RETURN (return from a subroutine) command



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.

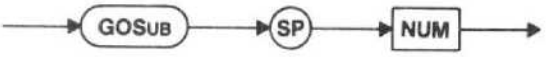
Macro Memory Used — 1 byte.

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.

There is no RETURN query.

GOSUB (go to LABELed subroutine) command



NUM — Macro control is transferred to the LABELed subroutine. The macro will return to the next command following the GOSUB when the macro finds a RETURN statement.

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 GOSUB. When the macro goes up to line 160, the macro is through running.

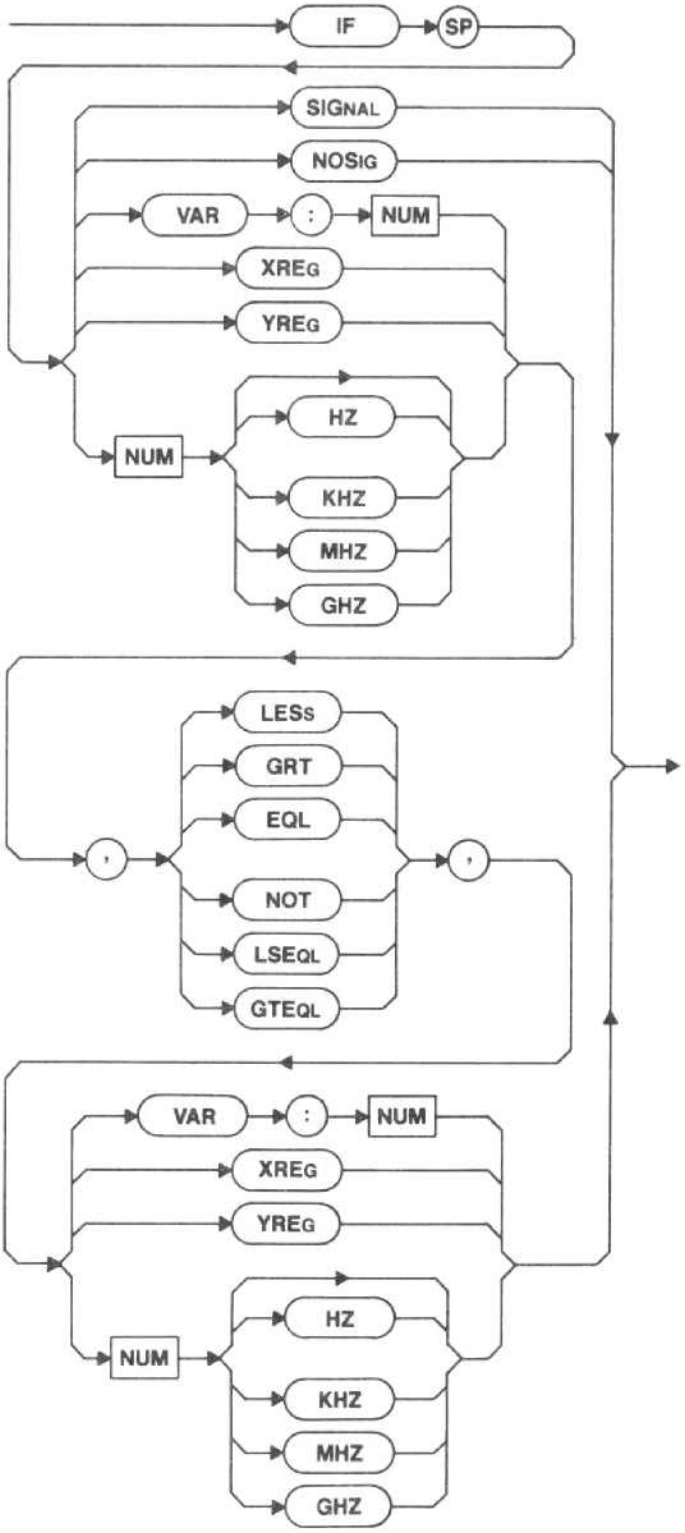
If there are more than 20 nested GOSUBs before a RETURN, macro execution error message 167 will be issued, and the macro will be aborted.

Macro Memory Used — 5 bytes.

Range — 1 to 100.

There is no GOSUB query.

IF (if statement is true) command (and ELSE and ENDI)



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 MVLFDB.

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

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.

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

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.

Line 140 — If a signal is not found, the test failed.

Line 150 — Prints the test failed message to the buffer.

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.

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

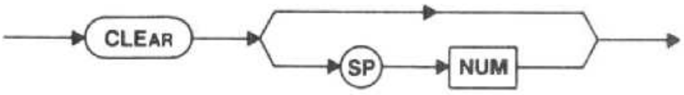
Interaction — The IF command requires an ENDI (end of the IF statement) command. If the ENDI is missing, macro execution error message 174 is issued and macro entry is aborted. The ELSE command is optional with the IF command.

There is no IF query.

PRINT COMMANDS

The print commands will clear the macro readout buffer (CLEAR); display a line (DSLINE); select data from the macro readout buffer (MRDO); and print a number or a string (PRINT).

CLEAR (clear macro readout buffer) command



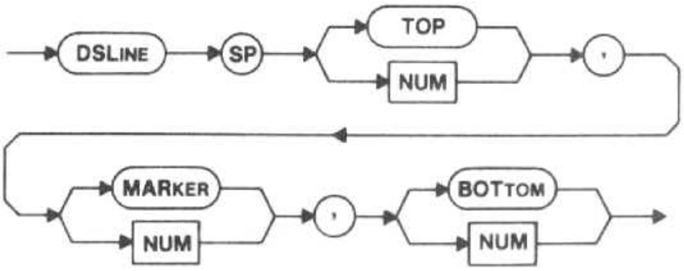
NUM — Only the numbered line (1 to 16) in the macro readout buffer will be cleared.

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

Macro Memory Used — 2 bytes.

There is no CLEAR query.

DSLINE (display line) command



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.

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.

Examples—See Example Descriptions Below

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

The following examples illustrate incorrect use of DSLINE.

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

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 (from line of the macro readout buffer) at the third line on the screen, and LINE 7 (from 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 (from line 7 of the macro readout buffer) at the seventh line on the screen, and LINE 8 (from line 8 of the macro readout buffer) at the eighth line on the screen.

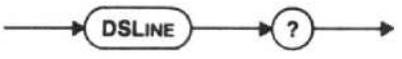
Macro Memory Used — 4 bytes.

Range — NUM is 1 to 16.

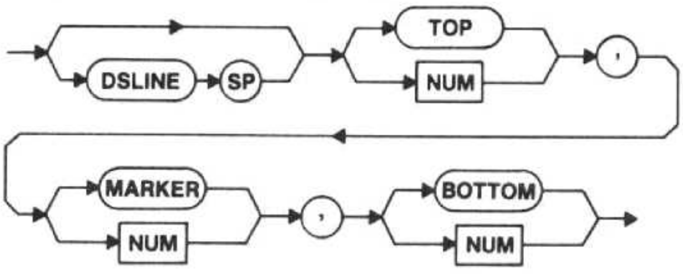
Power-up value — TOP, MARKER, and BOTTOM

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

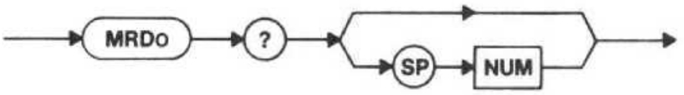
DSLINE (display line) query



Response to DSLINE query

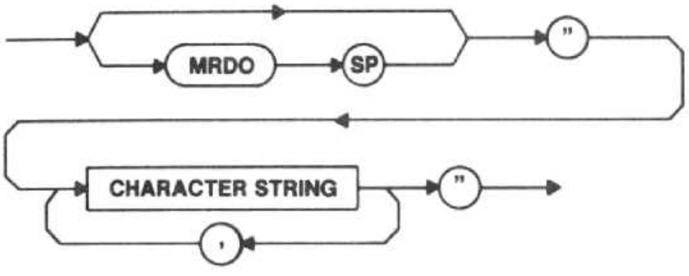


MRDO (macro readout) query



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.

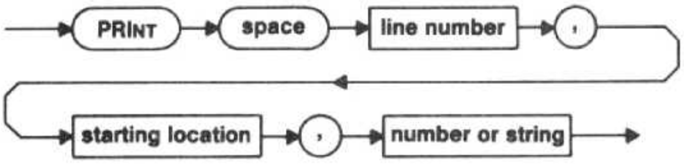
Response to MRDO query



PRINT (print number or string) command

The PRINT command syntax diagram is on the following page.

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). (A brief description follows of the five ways 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 enginnering 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 (lest 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.

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

Examples —

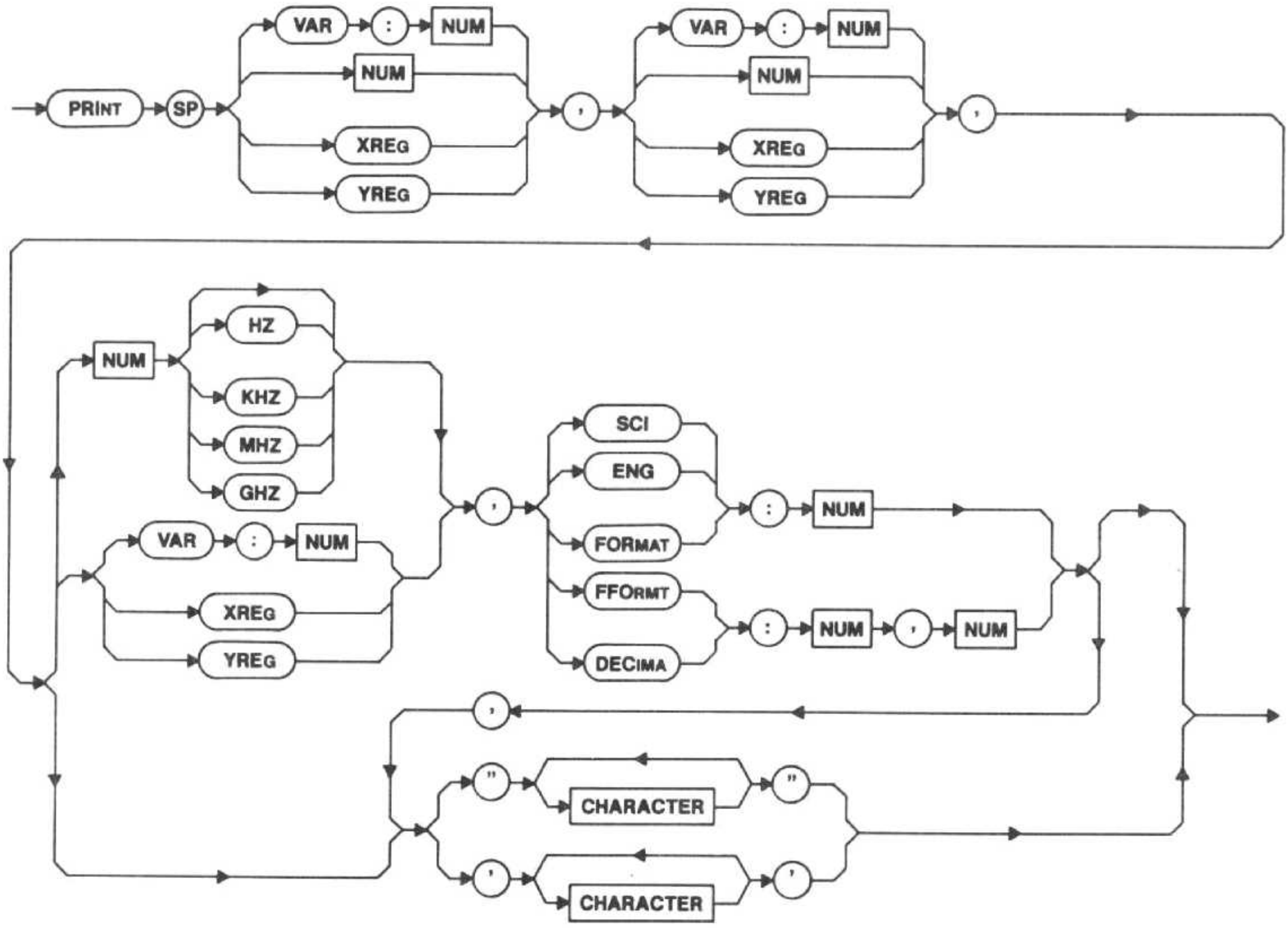
```
PRINT 1,5,XREG,SCI:14           "      3.122E+10"
PRINT 1,5,XREG,SCI:14," WATTS"  "       3.122E—3 WATTS"
PRINT 1,5,XREG,ENG:14           "      —31.12E+9"
PRINT 1,5,XREG,ENG:14," WATTS"  "       71.22E—E WATTS"
PRINT 1,5,VAR:3,DEC:14,2        "        —101.34"
PRINT 1,5,VAR:3,DEC:14,0        "          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.

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.

There is no PRINT query.

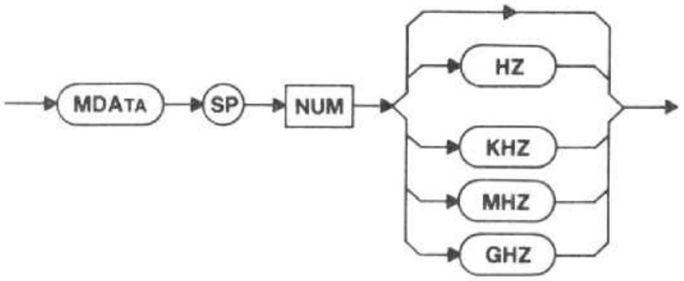
PRINT (print number or string) command syntax diagram



DATA COMMANDS

The data commands store numeric data (MDATA); read data and store it into the X register (READ); and restore the data pointer (MRESTO).

MDATA (store numeric data) command



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

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

Macro Memory Used — 9 bytes.

There is no MDATA query.

READ (read and store in XREG) command

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.

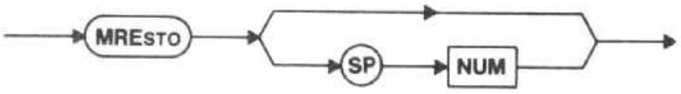
The program with the MRESTO command illustrates the use of the READ command.

Macro Memory Used — 1 byte.

Interaction — 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.

There is no READ query.

MRESTO (restore data pointer) command



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).

Examples—

MRESTO #When a READ command is used, the macro will start looking for the first MDATA command in the macro

MRESTO 1 #When a READ command is used, the macro will start looking for the first MDATA command after LABEL 1.

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 MDATA 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.

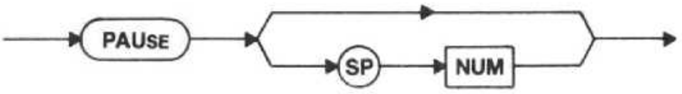
Interaction — If the macro warning message END OF READ DATA — MACRO ABORTED appears on the screen while the macro is running, the macro will be aborted at that point.

There is no MRESTO query.

GENERAL PURPOSE MACROS

The general purpose macros pause macro execution (PAUSE); start or restart macro execution (RUN); indicate when the macro is done executing (DONE); indicate when the macro entry is ended (EMAC); delete one or all macros (KILL); query macro status (MACRO?); stop macro execution (MCSTOP); tell how much memory is used for a macro or how much is left in the spectrum analyzer (MEMORY?); store the X register value into a variable (STNUM); display the requested menu (MENU); start a new sweep and wait for sweep to end (SWEEP); return the value of a variable (VAR?); tell instrument to store the following commands (STMAC); get the current waveform (GETWFM); and input a number from the DATA ENTRY pushbuttons (INPNUM).

PAUSE (macro pause) command

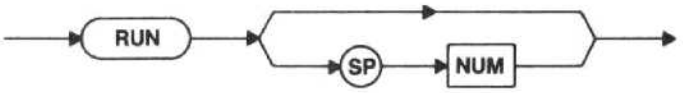


The PAUSE command pauses the current macro for 1 second, or NUM seconds if NUM is used.

Macro Memory Used — 1 byte.

There is no PAUSE query.

RUN (run macro) command



NUM — The macro number NUM will be run. Any macro currently running will be aborted and macro NUM will be started.

If any GPIB command comes while a macro is executing, the macro is stopped, and RUN will restart it.

RUN without an argument will restart a stopped macro. If a macro is not stopped and RUN is sent without an argument (or NUM is greater than 7), macro execution error message 162 (COMMAND IS ONLY AVAILABLE WHEN A MACRO IS STOPPED) is issued. If RUN is sent with a NUM (0–7) where no macro is located, macro execution error message 165 is issued.

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.

There is no RUN query.

DONE (macro execution finished) command

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).

If DONE is encountered while the instrument is under remote control, the current settings of TIME/DIV, MIN RF ATTEN dB, and PEAK/AVERAGE are held. DONE in local control resets the controls to their front-panel settings.

Macro Memory Used — 1 byte.

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.

There is no DONE query.

EMAC (end macro) command

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.

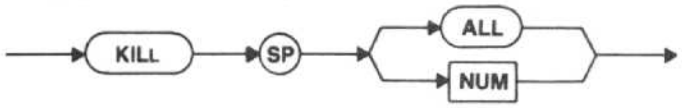
Table 6-1 lists the errors that may be sent when the macro is being compiled.

Table 6-1
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

There is no EMAC query.

KILL (delete macros) command



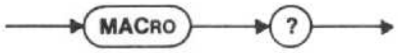
ALL — All macros will be deleted from memory.

NUM — Only macro NUM will be deleted from memory.

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

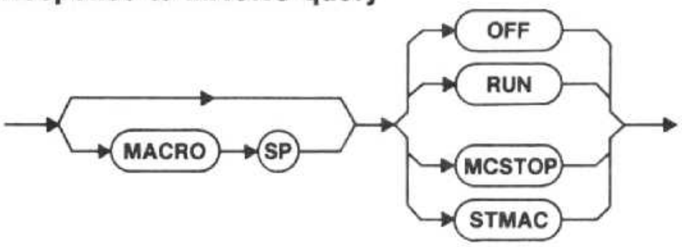
There is no KILL query.

MACRO (macro status) query



The MACRO query will return the current macro status.

Response to MACRO query



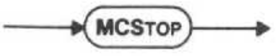
OFF — There is no macro action.

RUN — There is a macro running

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

STMAC — A macro is being stored in memory.

MCSTOP (macro stop) command



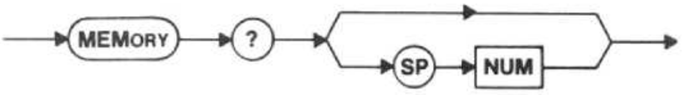
The MCSTOP command stops macro execution. The macro can be restarted by sending the RUN command over GPIB or by pressing RUN/STOP on the front panel.

MCSTOP always holds the current settings of the TIME/DIV, MIN RF ATTEN dB, and PEAK/AVERAGE.

Macro Memory Used — 1 byte.

There is no MCSTOP query.

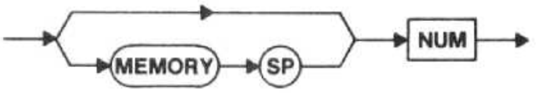
MEMORY query



The MEMORY query serves two separate purposes.

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

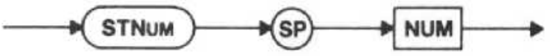
Response to MEMORY query



If a number argument that is out of range (1–30) 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.

There is no MEMORY command.

STNUM (store number) command



The STNUM command stores the 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.

Examples—

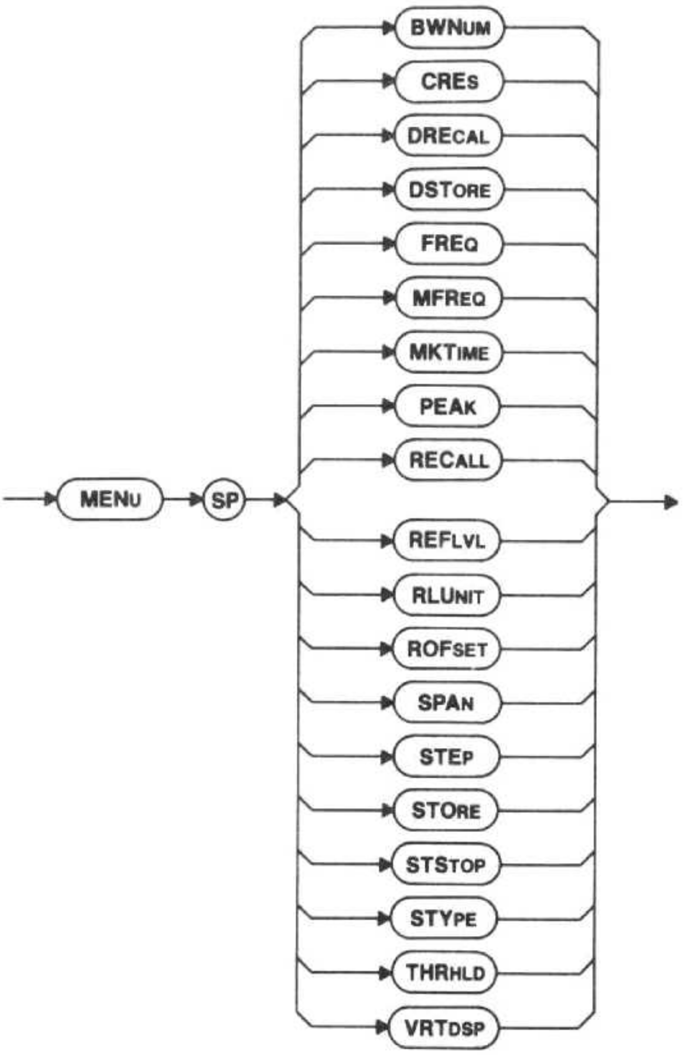
```
STNUM 1  #Copies the number in XREG to variable
  number 1.
STNUM 9  #Copies the number in XREG to variable
  number 9.
```

Macro Memory Used — 2 bytes.

Range — NUM is 1 to 30.

There is no STNUM query.

MENU (macro menu) command

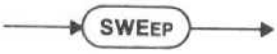


The MENU command will display the requested menu. A new value for that setting can then be input from the front panel. After the entry is complete, the macro will continue. If either the blue or green SHIFT pushbutton is pressed, macro execution will resume with no value change. If an invalid pushbutton is pressed, the macro is aborted at this point and cannot be restarted. For a description of the command arguments, see the description of the command whose mnemonic is the same as the argument.

Macro Memory Used — 2 bytes.

There is no MENU query.

SWEEP (take a sweep) command

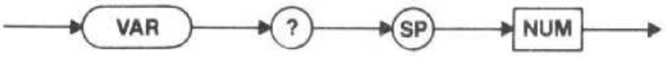


The SWEEP command will start a new sweep and wait until the sweep has finished before executing the next macro command.

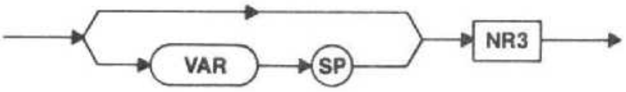
Macro Memory Used — 1 byte.

There is no SWEEP query.

VAR (variable) query



Response to VAR query

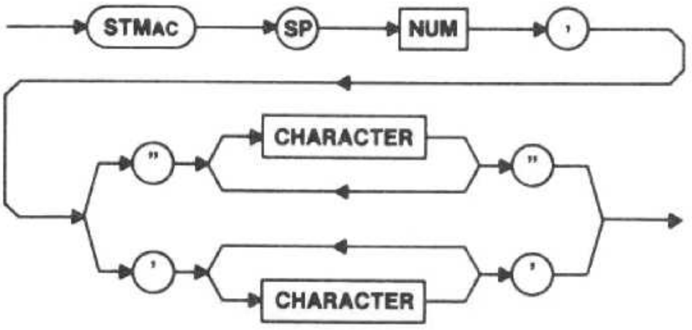


The VAR query response will return the value of variable NUM. If NUM is out of range, 0 is returned (no error message is issued).

Range — 1 to 30.

There is no VAR command.

STMAC (store macro) command



The STMAC command tells the spectrum analyzer that the following GPIB commands will be stored as macro number NUM with a title of the CHARACTER 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. The title CHARACTER string may be up to 22 characters.

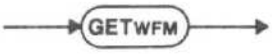
Example—STMAC 1,"HARMONIC TEST"

Range — NUM is 0 to 7.

Interaction — 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 put back in the regular operation mode.

There is no STMAC query.

GETWFM (get waveform) command



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.

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))

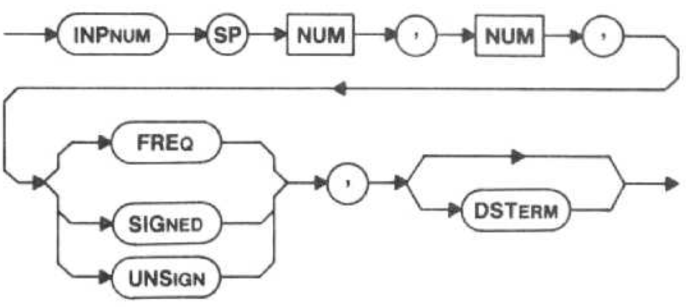
Line 140 — If YREG (old maximum value) is greater than XREG

Line 150 — Puts higher value into XREG

Macro Memory Used — 1 byte.

There is no GETWFM query.

INPNUM (input number) command



The INPNUM command allows you to input a number from the DATA ENTRY pushbuttons and store the number in XREG. The first NUM sets the line number on which the input number will be printed on the screen. The second NUM is the starting character position for the input number.

With FREQ, the GHz, MHz, KHz, and Hz terminators are valid, and the corresponding front-panel pushbuttons will light. 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, and the corresponding front-panel pushbuttons will light. +dBX will not change the input number, but −dBX will make the input number negative.

With UNSIGN, the dB terminator is valid and the front-panel dB pushbutton will light.

If DSTERM (display terminators) is used, the words END WITH and the valid terminators will be displayed two lines below the entry line with an underline on the following line.

If an undefined key is pressed, the macro will be aborted. If a terminator is pressed without a number or a SHIFT pushbutton 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 pushbutton to get that value (default)

NOTE

The message PRESS "SHIFT" TO ABORT is always displayed on line 16, so that line is not available for input.

The following example illustrates the use of the INPNUM command. The example is illustrated in Figure 6-1.

```
 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.

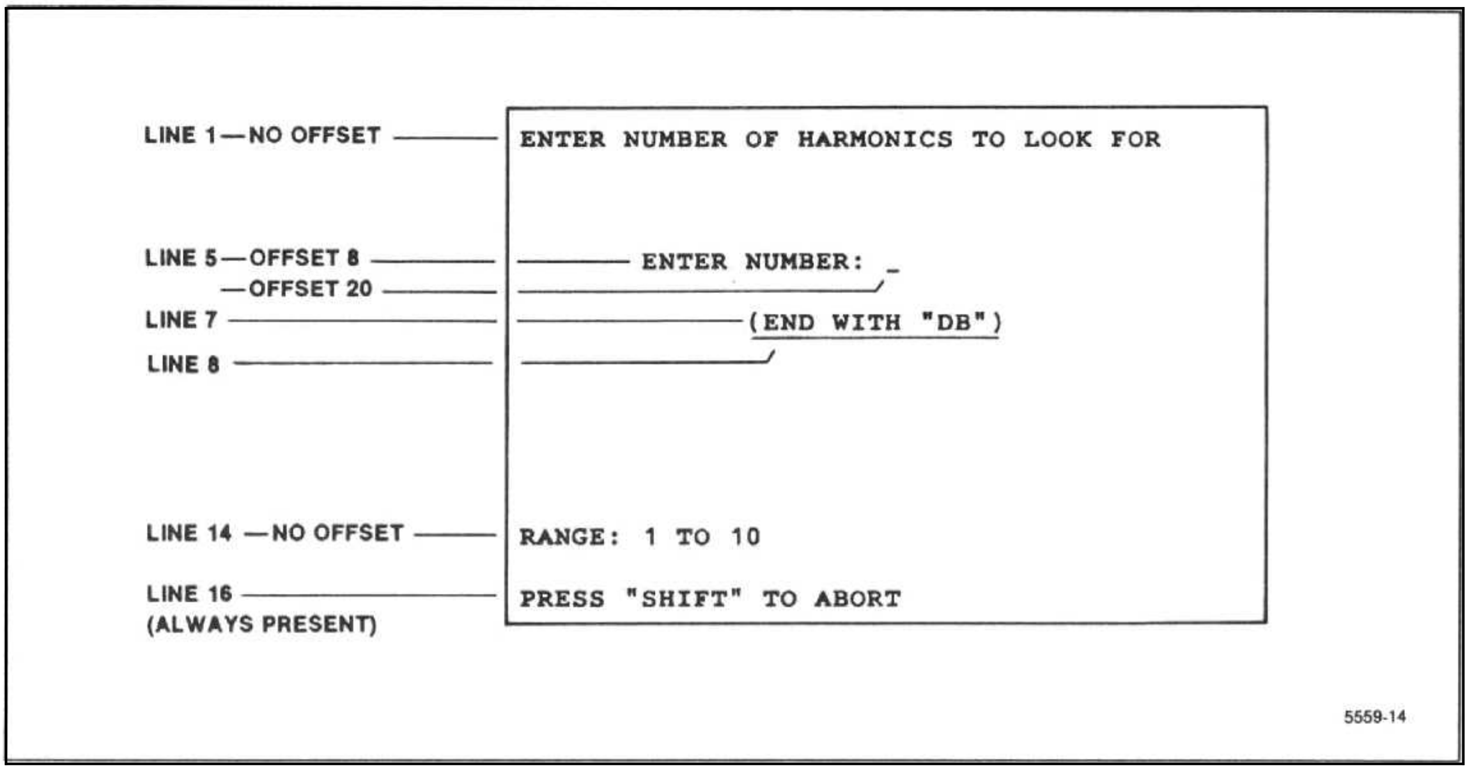
Line 140 — Prints the statement on line 14 of the screen beginning at character position 1.

Line 150 — Prepares to enter a number on line 5 of the screen, indented 19 character spaces. Since UNSIGN and DSTERM are used, the front-panel dB pushbutton will light and the words END WITH will be on the screen (on line 7 with an underline on line 8).

Macro Memory Used — 5 bytes.

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.

There is no INPNUM query.



```
ENTER NUMBER OF HARMONICS TO LOOK FOR



       ENTER NUMBER: _

                 (END WITH "DB")






RANGE: 1 TO 10

PRESS "SHIFT" TO ABORT
```

Figure 6-1. INPNUM example.

DISPLAY DATA AND CRT READOUT I/O

The spectrum analyzer follows the Tektronix Interface Standard for GPIB Codes, Formats, Conventions, and Features for waveform transfer. The commands and queries in this section transfer display and readout data to or from the spectrum analyzer and are divided into two categories; waveform transfers, and crt readout transfers.

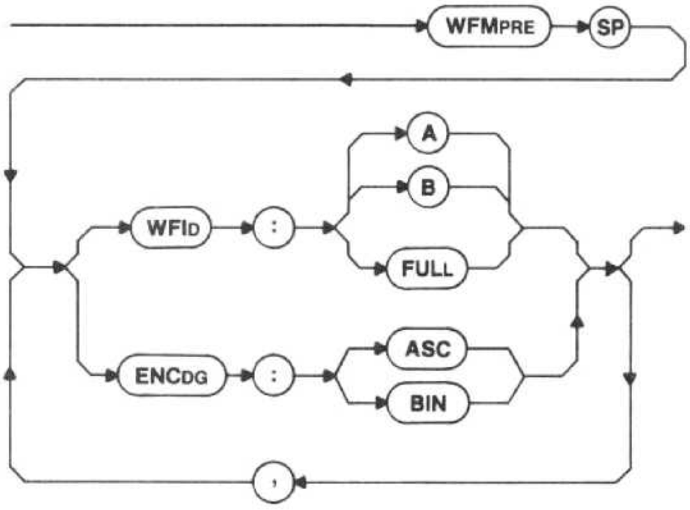
Use in Macros

Most of the I/O commands in this section can be incorporated into macros designed to your specific needs. (No queries can be used within macros.) Since there is a total of 8k bytes of memory dedicated for macro use, it is important that you know the number of bytes used for each command, and keep this in mind while preparing macros. This maximum number of bytes used is included with the commands in this section; and there is also a table in the Index at the back of this manual that lists all available spectrum analyzer commands and the bytes used by each.

WAVEFORM TRANSFERS

The waveform transfers begin with a waveform preamble (WFMPRE) that identifies and scales the data and continue with data (CURVE) that represents the waveform. A query (WAVFRM?) displays the responses to the WFMPRE and CURVE queries. The display preamble (DPRE?) contains the numeric data necessary to reproduce the display. The display units necessary to make a hard copy of the display (DCOPY?) can be transmitted to another unit. A readout command (RDOUT) displays messages on the crt in either a 2-line or a 16-line mode (TEXT). Three crt readout queries return the upper row of characters (UPRDO), the middle row (MDRDO), or the lower row (LORDO).

WFMPRE (waveform preamble) command



NOTE

The WFMPRE command has no effect on the Marker Finding commands in Section 5 of this manual.

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/div). 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.

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.

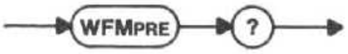
The WFMPRE command cannot be used within a macro.

Examples—WFMPRE WFID:FULL
WFMPRE ENCDG:ASC
WFM WFID:A,ENC:BIN

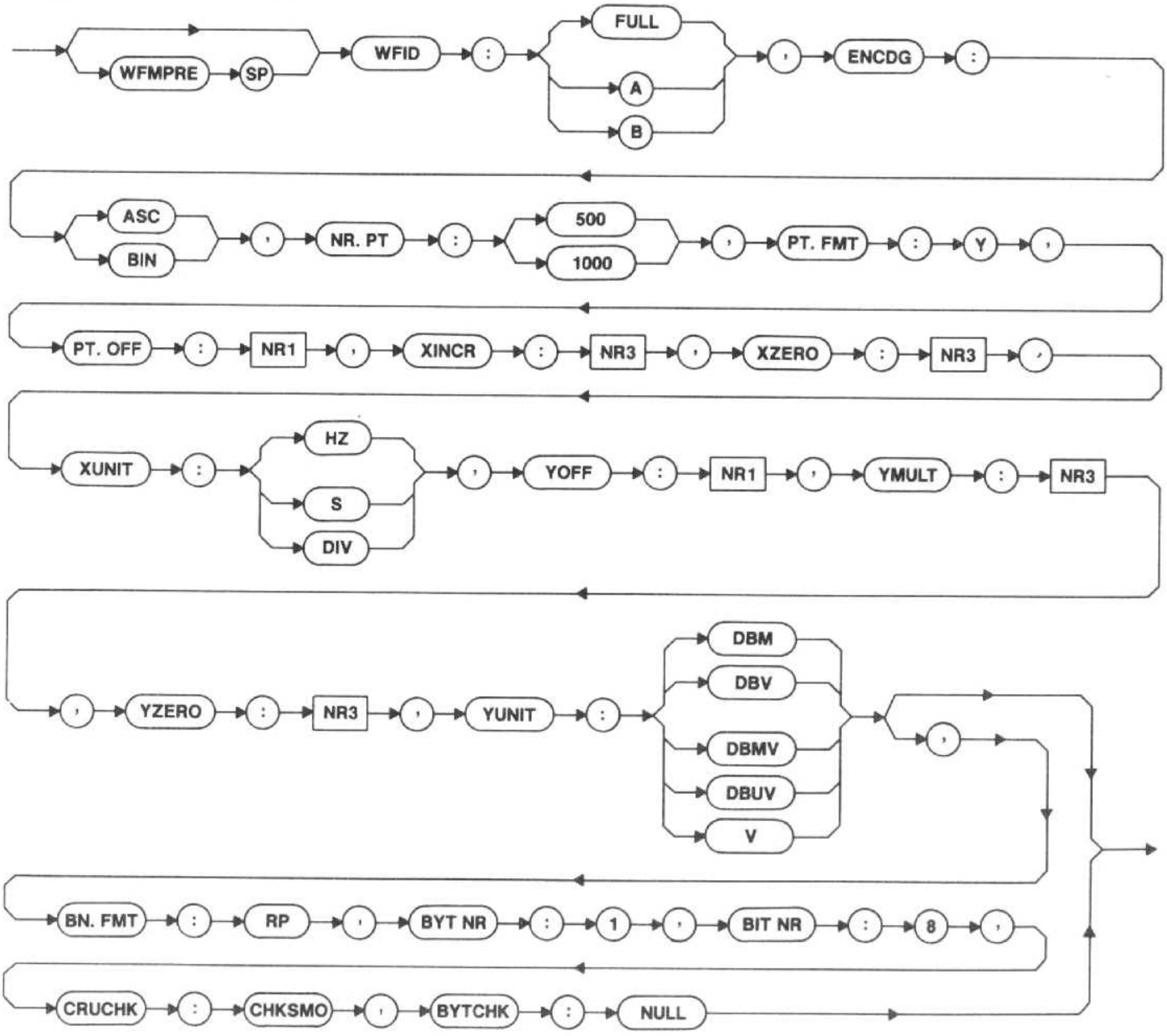
Power-up value — Full (1000 point), ASCII-coded digits.

Interaction — 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.

WFMPRE (waveform preamble) query



Response to WFMPRE query



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.

CRVCHK:CHKSM0 — Specifies that the last byte of a binary transfer is a 2's complement, modulo-256 check sum 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.

$$\mathrm{XN} = \mathrm{XZERO} + \mathrm{XINCR} * (\mathrm{N} - \mathrm{PT.OFF})$$

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

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

$$\mathrm{YN} = \mathrm{YZERO} + \mathrm{YMULT} * (\mathrm{VALN} - \mathrm{YOFF})$$

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

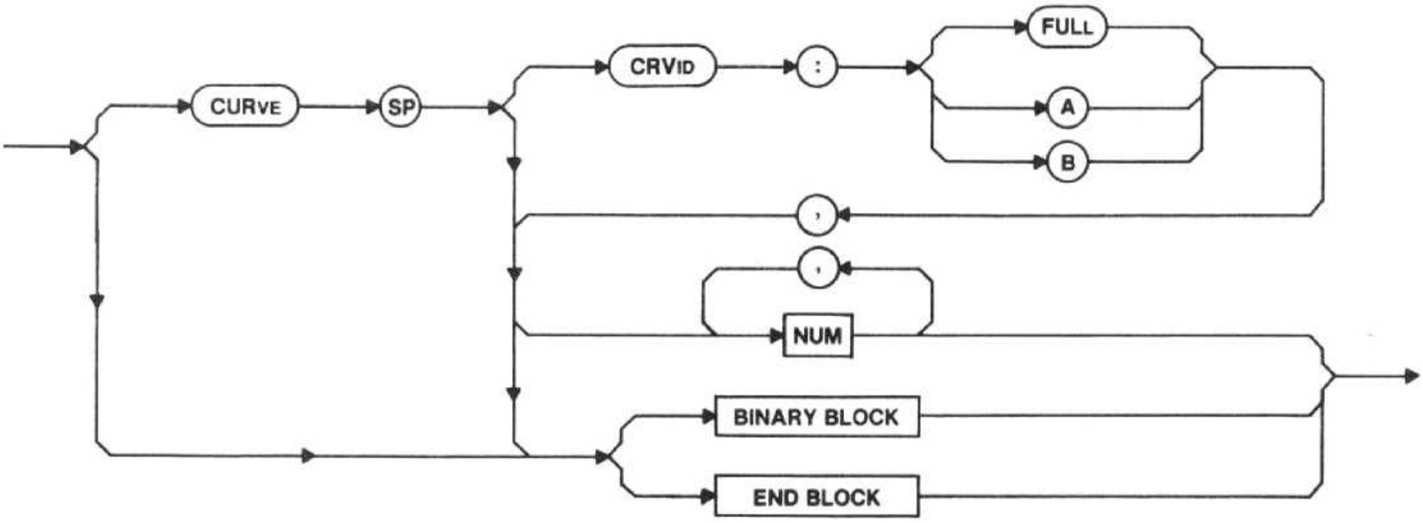
For example, data value 125 (graticule center) could have the following absolute values

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.

CURVE (display curve) command



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.

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.

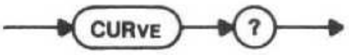
END BLOCK — End block is a sequence of binary numbers that is preceded by the ASCII code for at (@); EOI must be asserted concurrently with the last data byte. The end block can only be the last data type in the message.

The CURVE command cannot be used within a macro.

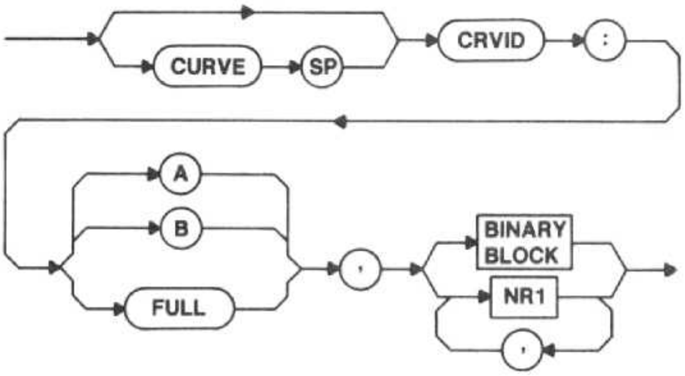
Examples—CURVE CRVID:FULL,100,100,101,99,
<996 more numbers>
CURVE <500 or 1000 numbers>
CUR <BINARY BLOCK>

Interaction — 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.

CURVE (display curve) query



Response to CURVE query



Waveform data is related to the display by Figure 7-1.

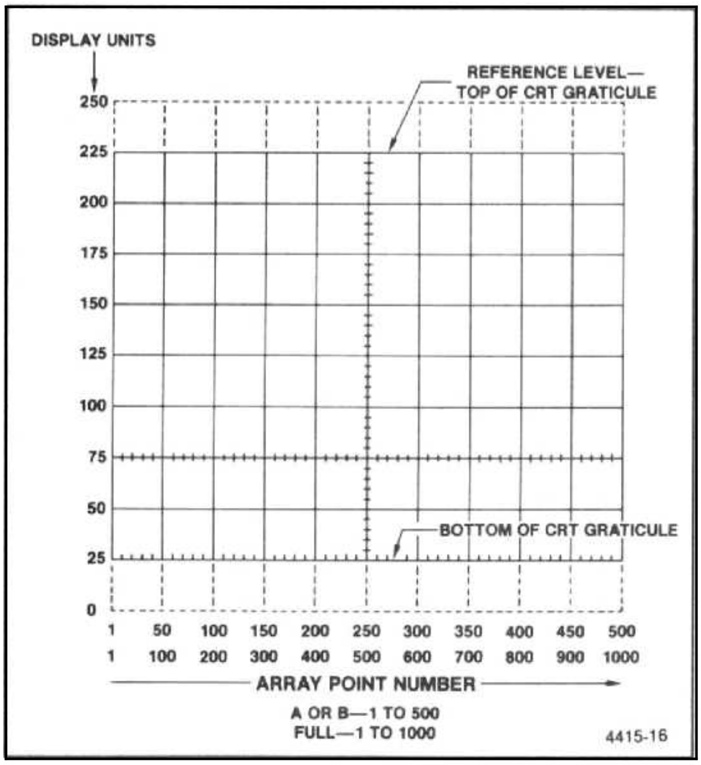
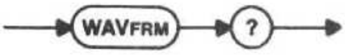


Figure 7-1. Waveform data related to the display.

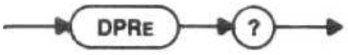
WAVFRM (waveform) query



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).

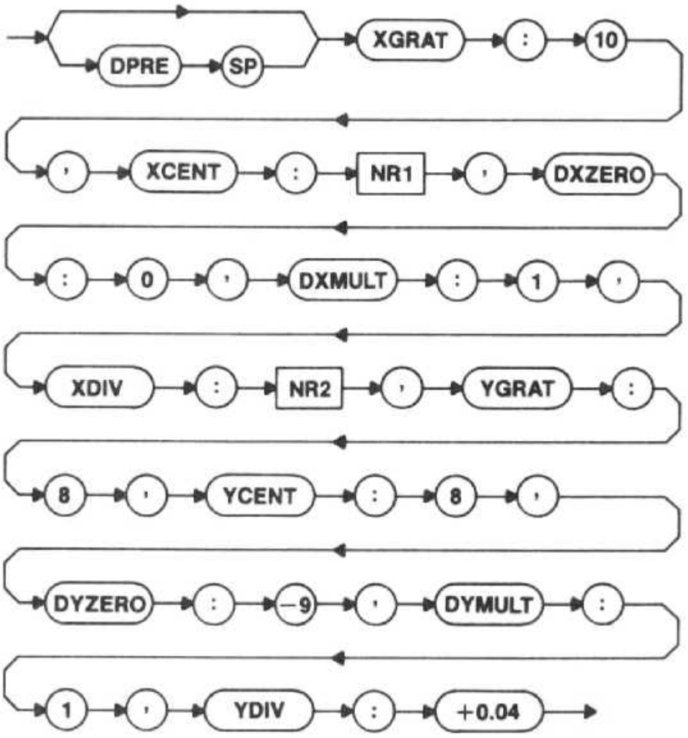
There is no WAVFRM command.

DPRE (display preamble) query



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

Response to DPRE query



XGRAT: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.

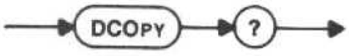
DYN = DYMULT * VALN * YDIV + YCENT
+ DYZERO

where:

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

There is no DPRE command.

DCOPY (copy display) query



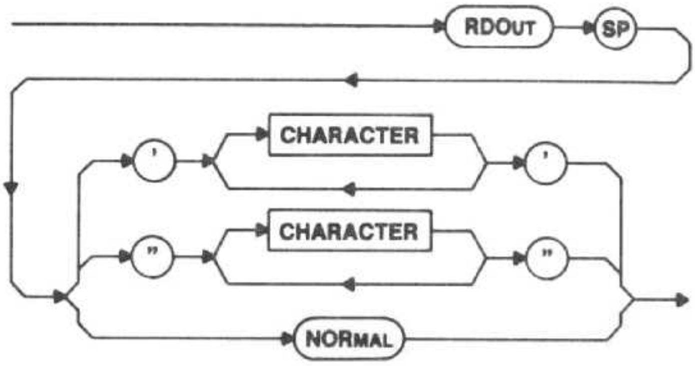
The DCOPY query response is the same as the response to ID?;WFMPRE?;DPRE?;CURVE?. It allows transmission of information from one device to another in "display" units, so that a hard copy can be made of the display.

There is no DCOPY command.

CRT READOUT TRANSFERS

Readout messages (RDOUT) can be displayed on the crt screen in either a 3-line or a 16-line mode or can be put in the macro readout buffer text mode (TEXT). Three crt readout queries return the upper row of normal readout characters (UPRDO), the middle row (MDRDO), or the lower row (LORDO).

RDOUT (readout message) command



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.

NOTE

All characters are displayed as upper-case characters.

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.

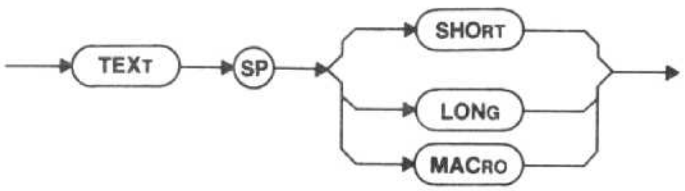
Macro Memory Used — 41 bytes.

Power-up value — Normal readout.

Interaction — 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 an control that causes the normal readout to be updated. TEXT LONG will blank the screen.

There is no RDOUT query.

TEXT (text mode) command



SHORT — The readout is switched to the normal 3-line mode with a spectrum display. When the RDOUT commands are used, the normal 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 are full of 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.

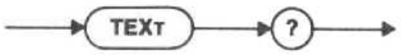
MACRO — The readout is switched to the 16-line mode without a spectrum display, and the macro readout buffer is displayed. The PRINT command is used to display data into the macro readout buffer. The CLEAR command can be used to clear the macro readout buffer.

Macro Memory Used — 2 bytes.

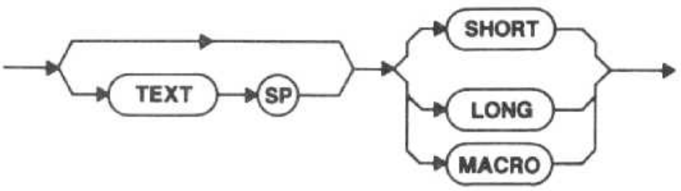
Power-up value — SHORT

Interaction — If the crt readout is not in the NORMAL mode when TEXT is executed, the readout 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.

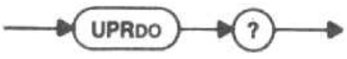
TEXT (text mode) query



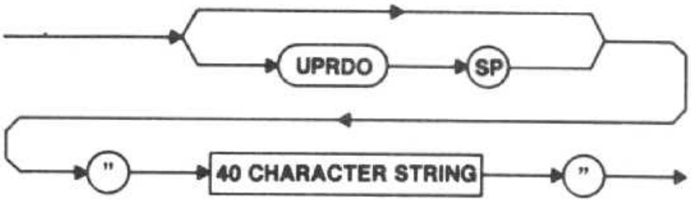
Response to TEXT query



UPRDO (upper readout) query



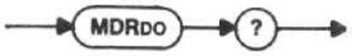
Response to UPRDO query



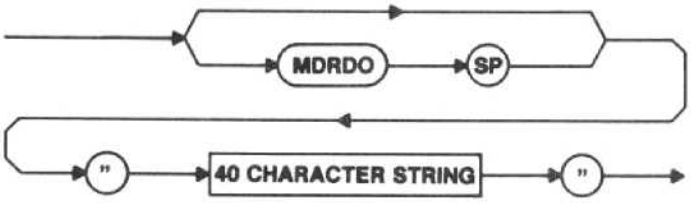
CHARACTER — Characters are from the upper row of regular crt readout. Blanks are transmitted as spaces. Regular readout that would be displayed if GPIB did not have control (whether visible on the screen or not) is the readout returned by the query, not a message sent to the instrument by RDOUT. With AVIEW and SAVEA both ON and BMINA and BVIEW both OFF, the returned readout will be the saved readout. Refer to the recall display (DRECAL) or the save A waveform (SAVEA) commands in Section 4 of this manual.

There is no UPRDO command.

MDRDO (middle readout) query



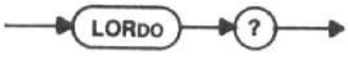
Response to MDRDO query



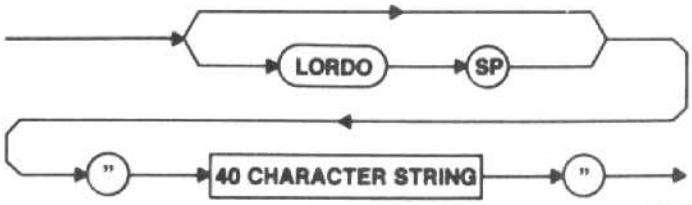
CHARACTER — Characters are from the middle row of regular crt readout. Blanks are transmitted as spaces. Regular readout that would be displayed if GPIB did not have control (whether visible on the screen or not) is the readout returned by the query, not a message sent to the instrument by RDOUT. Refer to the recall display (DRECAL) or the save A waveform (SAVEA) commands in Section 4 of this manual.

There is no MDRDO command.

LORDO (lower readout) query



Response to LORDO query



CHARACTER — Characters are from the lower 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. With AVIEW and SAVEA both ON and BMINA and BVIEW both OFF, the returned readout will be the saved readout. Refer to the recall display (DRECAL) or the save A waveform (SAVEA) commands in Section 4 of this manual.

There is no LORDO command.

WAVEFORM PROCESSING

The commands in this section allow local processing of spectrum data by the spectrum analyzer. Some of these commands operate on a display data point. This is an ordered pair (an X and a Y value) that corresponds to a point on the spectrum analyzer display. On command, the spectrum analyzer gets a display data point from the current digital storage waveform. The point is held in memory until another command updates the data point. A query requests that the spectrum analyzer report the point. Other commands change spectrum analyzer settings automatically to center it on the point.

Commands that update the display data point direct the spectrum analyzer to a new point (POINT), find the largest or nearest signal (FIBIG, LFTNXT, RGTNXT), or search for the maximum or minimum value (FMAX, FMIN). A query (POINT?) returns the X and Y values of the display data point.

This section covers how the waveform processing commands and query work. Two programs at the end of Section 2 in this manual show some of these commands in use. Waveform processing techniques are offered in Section 10 of this manual.

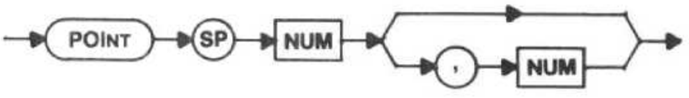
Use in Macros

Most of the waveform processing commands in this section can be incorporated into macros designed to your specific needs. (No queries can be used within macros.) Since there is a total of 8k bytes of memory dedicated for macro use, it is important that you know the number of bytes used for each command, and keep this in mind while preparing macros. This maximum number of bytes used is included with the commands in this section; and there is also a table in the foldout pages at the back of this manual that lists all available spectrum analyzer commands and the bytes used by each.

WAVEFORM FINDING

The spectrum analyzer has two sets of waveform-finding commands; two commands are described here and five are described in Section 6 of this manual. The RGTNXT and LFTNXT waveform processing commands move the invisible display data point, and the MRGTNXT and MLFTNXT marker commands move the Primary marker. The display data point is specified and reported in screen units, and the Primary marker is specified and reported in frequency and amplitude. The two locations (data point and marker) and the two sets of commands are independent unless the display data point and the Primary marker are coupled with the MCPOIN command. The DPMK command moves the display data point to the Primary marker location without coupling the two, and MKDP moves the Primary marker to the horizontal location of the display data point, also without coupling the commands.

POINT (display data point) command



First **NUM** — This is the X value of a display data point. The horizontal scale is always the same as a full, 1000-point waveform, as explained under Display Data Point Commands Interaction later in this section.

Second **NUM** — This is the Y value of a display data point. The vertical scale is the same as illustrated for the CURVE query in Section 7 of this manual.

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.

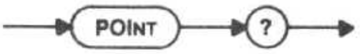
Macro Memory Used — 5 bytes.

Examples — POINT 500,150 (center screen)
POI 1,25 (screen bottom left)
POI 1000,225 (screen top right)

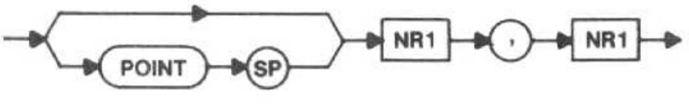
Power-up value — 500,225.

Interaction — 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. See Display Data Point Commands Interaction.

POINT (display data point) query

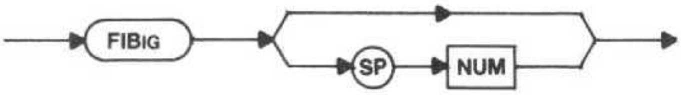


Response to POINT query



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.

FIBIG (find big) command



This command seeks to acquire the largest signal peak with a point of greater value than NUM. 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.

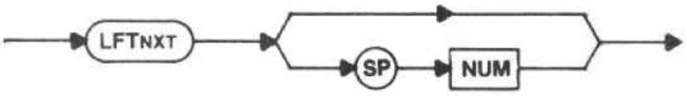
A pattern recognition routine is used to recognize signals.

Macro Memory Used — 2 bytes.

Interaction — See Display Data Point Commands Interaction.

There is no FIBIG query.

LFTNXT (left next) command



This command searches to the left of the current point to acquire the peak of a signal whose value is greater than NUM. If a signal peak greater than NUM is not found, the display data point is set to 0,0. If NUM is omitted from the command, a default value of 0 is used.

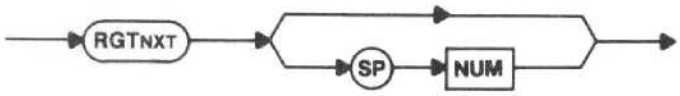
A pattern recognition routine is used to recognize signals.

Macro Memory Used — 2 bytes.

Interaction — See Display Data Point Commands Interaction.

There is no LFTNXT query.

RGTNXT (right next) command



This command searches to the right of the current point to acquire the peak of a signal whose value is greater than NUM. 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.

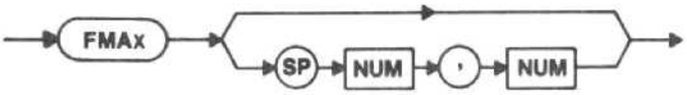
A pattern recognition routine is used to recognize signals.

Macro Memory Used — 2 bytes.

Interaction — See Display Data Point Commands Interaction.

There is no RGTNXT query.

FMAX (find maximum value) command



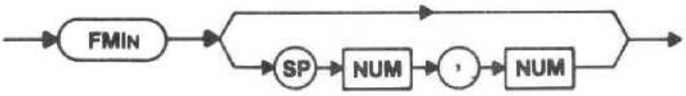
This routine sets the display data point to the point in digital storage with the largest Y value. If the largest Y value is located at more than one point, the first (leftmost) point is acquired. The optional arguments are two display X values. The FMAX command will limit its search over this X range; otherwise, the full X range (1 to 1000) will be searched.

Macro Memory Used — 5 bytes.

Interaction — See Display Data Point Commands Interaction.

There is no FMAX query.

FMIN (find minimum value) command



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 (leftmost) point is acquired. 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.

Macro Memory Used — 5 bytes.

Interaction — See Display Data Point Commands Interaction.

There is no FMIN query.

Data Point Commands Interaction

1. The waveform processing commands in this section operate only on the waveform specified by the last WFMPRE or CURVE command; either A or B or full (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.

2. The 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 two particular conditions exist together, it can cause CURVE? data output commands to be invalid;

(1) if CURVE? is followed by a command to update the display data point, and

(2) if 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.

3. 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 noted in part 2.

For further information, refer to Multiple Use of Display Buffer for Waveform Processing and I/O in Section 10 of this manual.

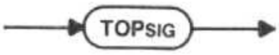
CENSIG (center signal) command

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 run, the display data point may no longer match the signal of interest.

Macro Memory Used — 1 byte.

TOPSIG (move to top of graticule) command



This command changes REFLVL to move the signal represented by the display data point to the reference level (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.

There is no TOPSIG query.

SYSTEM COMMANDS AND QUERIES

Spectrum analyzer device-dependent message units are provided to set and return parameters of use to the controller in a GPIB system. These commands and queries are described in this section in three groups related to instrument parameters, message execution, and status and error reporting.

NUM Argument

Unless otherwise stated, the values for the NUM argument are

- 1 = ON
- ≥+0.5 are rounded to 1
- 0 = OFF
- <+0.5 are rounded to 0

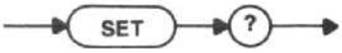
Use in Macros

Most of the system commands in this section can be incorporated into macros designed to your specific needs. (No queries can be used within macros.) Since there is a total of 8k bytes of memory dedicated for macro use, it is important that you know the number of bytes used for each command, and keep this in mind while preparing macros. This maximum number of bytes used is included with the commands in this section; and there is also a table in the foldout pages at the back of this manual that lists all available spectrum analyzer commands and the bytes used by each.

INSTRUMENT PARAMETERS

The commands and queries in this group return instrument settings (SET?), return the instrument identification parameters (ID?), initialize settings (INIT), and control return of the query response header (HDR).

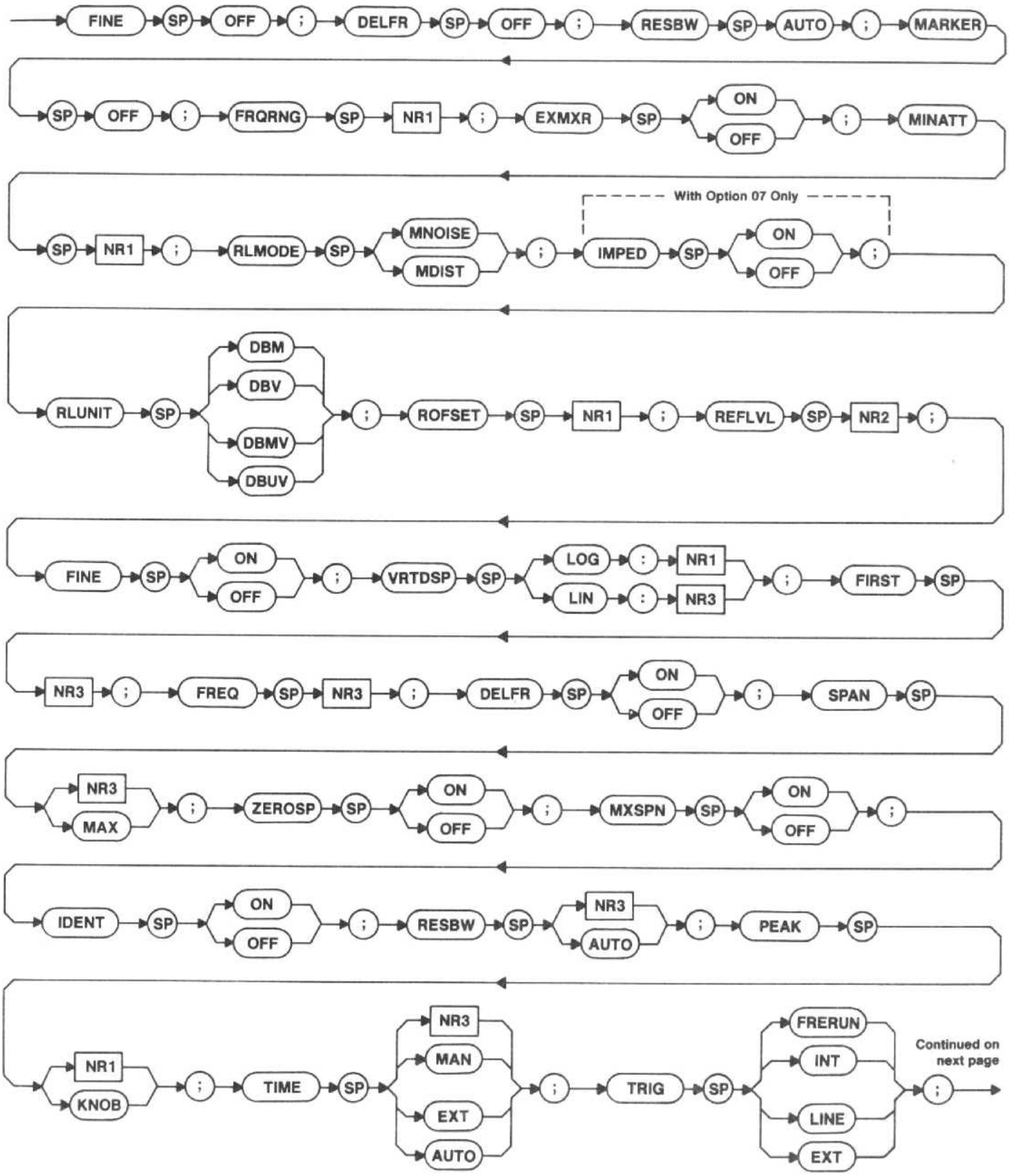
SET (instrument settings) query



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 assure no interaction 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.

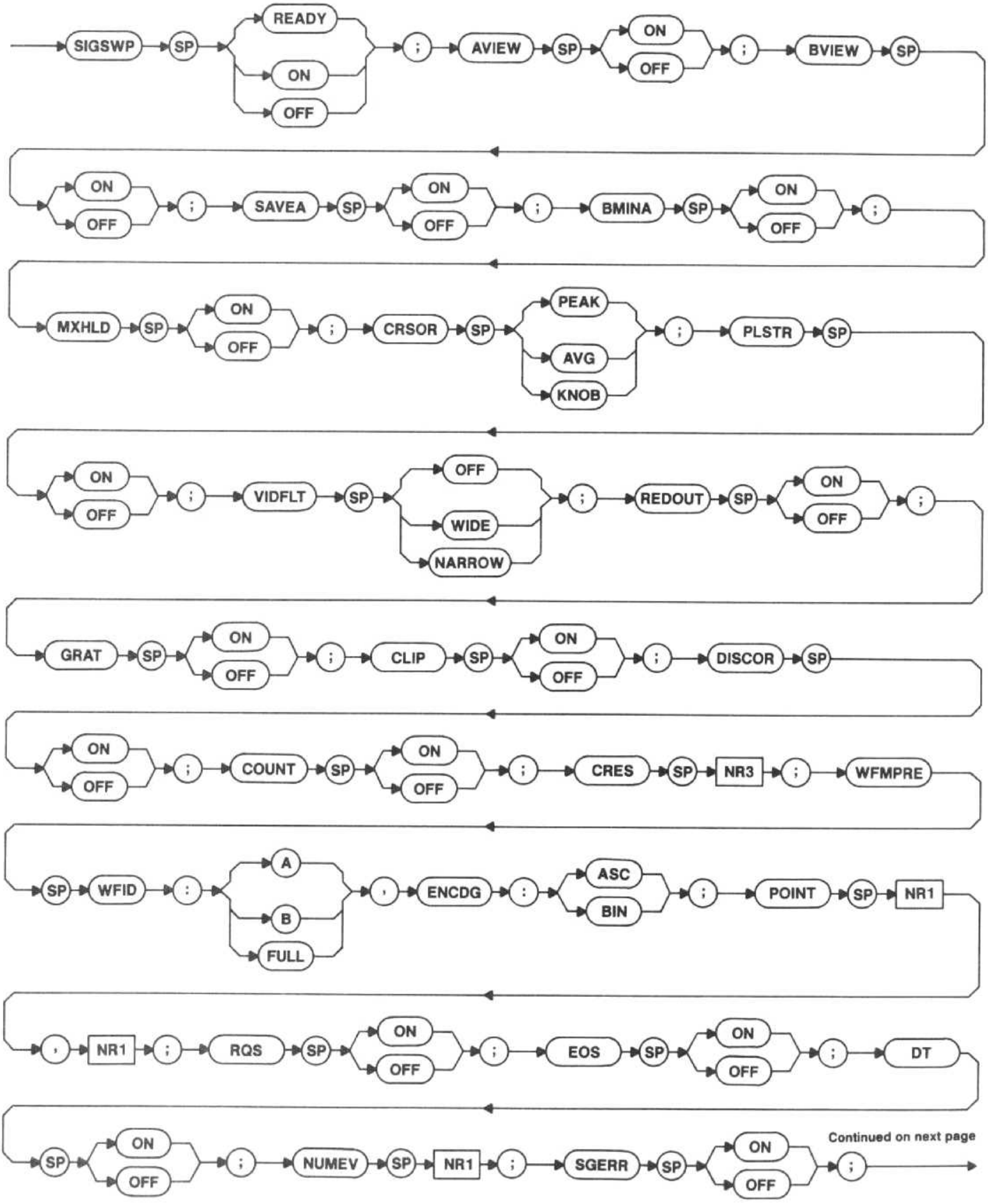
There is no SET command.

Response to SET query



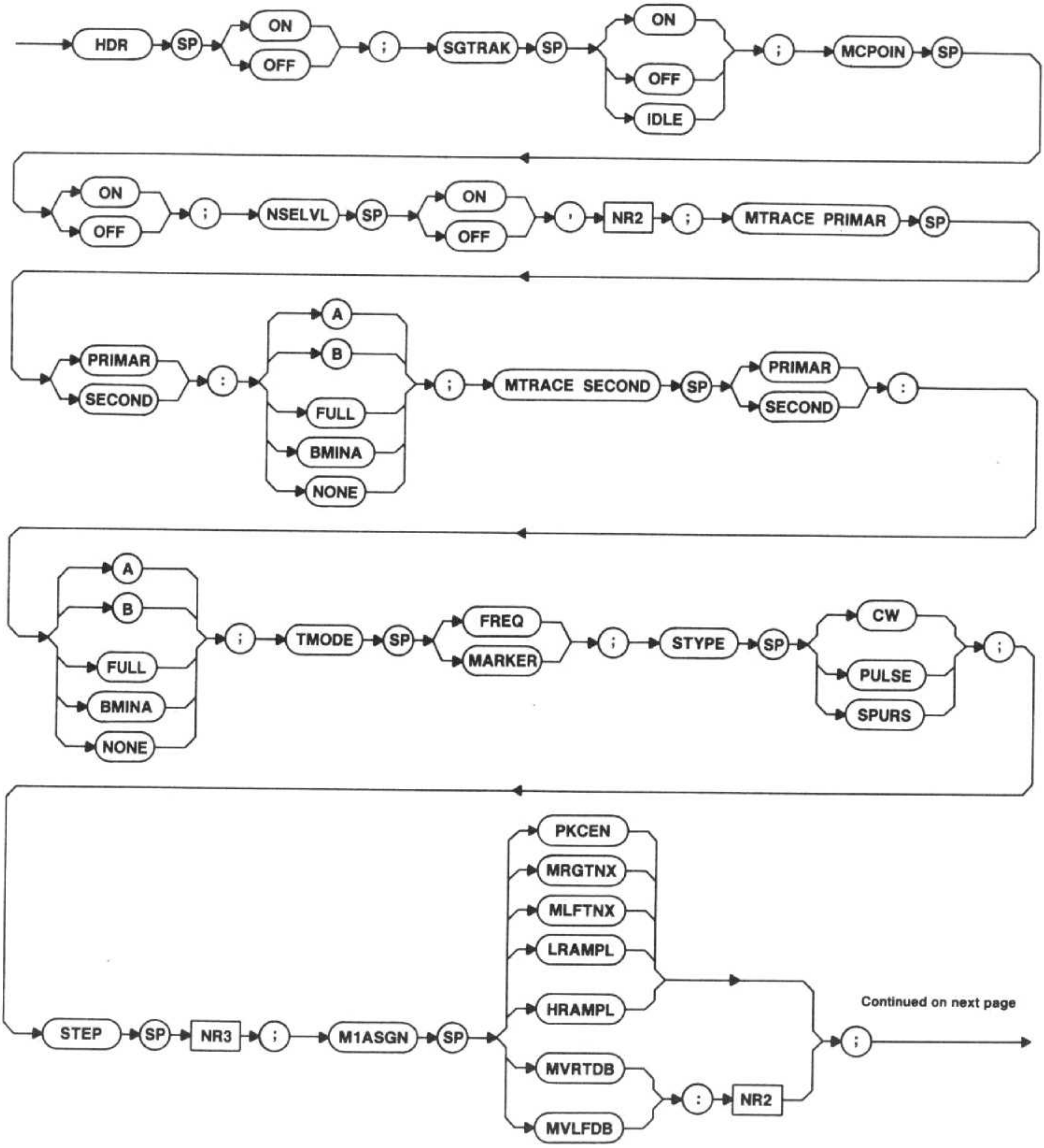
Continued on next page

Response to SET query (continued)



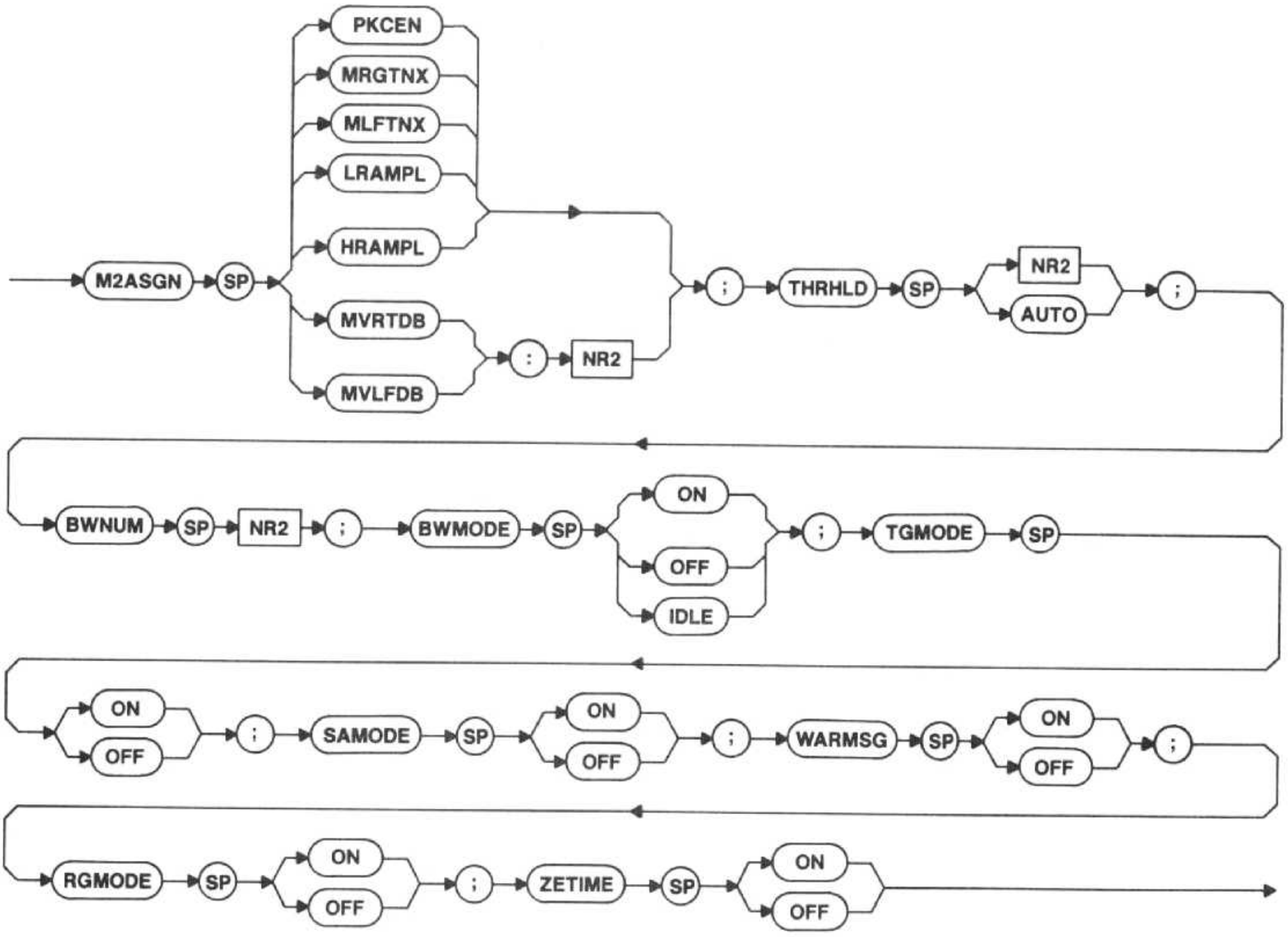
Continued on next page

Response to SET query (continued)

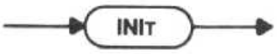


Continued on next page

Response to SET query (continued)



INIT (initialize settings) command



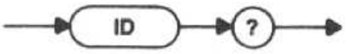
INIT resets the instrument the same as if the power was turned off, then turned back on. The instrument functions are reset as shown in Table 9-1.

Macro Memory Used — 1 byte.

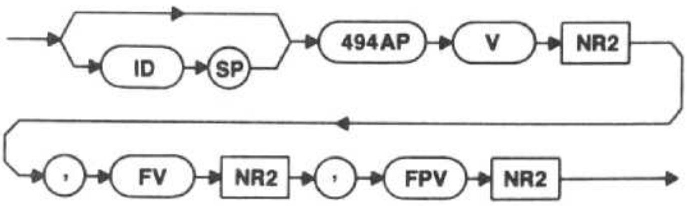
Interaction — IEEE 488 interface functions are not affected and the instrument remains under remote control. RQS is set to OFF if either the LISTEN ONLY or TALK ONLY switch is set.

There is no INIT query.

ID (identify) query



Response to ID query



494AP — The instrument type number.

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.

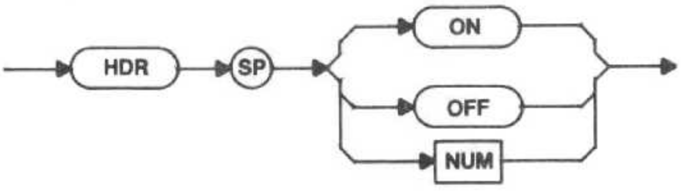
There is no ID command.

Table 9-1

INSTRUMENT FUNCTIONS

Mnemonic	INIT Value
FREQ	0
FIRST	2.972^{E+9}
SECOND	2.182^{E+9}
DISCOR	OFF
FRQRNG	1
DELFR	OFF
SPAN	MAX
ZEROSP	OFF
ARES	ON
MXSPN	OFF
PHSLK	OFF
VIDFLT	OFF
VRTDSP	LOG:10 DB
REFLVL	0 dBm
FINE	OFF
RLMODE	MNOISE
MINATT	Knob position
PLSTR	OFF
TRIG	FRERUN
SIGSWP	OFF
TIME	Knob position
AVIEW	ON
BVIEW	ON
SAVEA	OFF
BMINA	OFF
MXHLD	OFF
CRSOR	KNOB
REDOUT	ON
GRAT	OFF
PT.OFF	500
YOFF	225
CLIP	OFF
TEXT	SHORT
EOS	OFF
RQS	ON
WARMSG	ON
NUMEV	0
IMPED	OFF (50 Ω) — Option 07 Only
RLUNIT	dBm
STSTOP	0–1.8 GHz
HDR	ON
SGERR	OFF
SGTRAK	OFF
MCPOIN	OFF
NSELVL	OFF
TMODE	FREQ
STYPE	CW
MTRACE PRIMAR	NONE
MTRACE SECOND	NONE
THRHLD	AUTO
BWMODE	OFF
TGMODE	OFF
SAMODE	OFF
ECR	OFF
ZETIME	OFF
RGMODE	OFF
ROFSET	0

HDR (header) command



ON — The header for query responses is turned on.

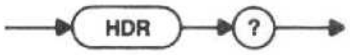
OFF — The header for query responses is turned off.

Macro Memory Used — 2 bytes.

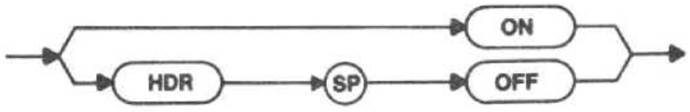
Power-up value — On.

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

HDR (header) query



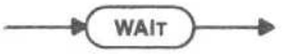
Response to HDR query



MESSAGE EXECUTION

The two following commands (WAIT and REPEAT) affect how the spectrum analyzer executes message units imbedded within other messages.

WAIT (wait for end of sweep) command



The spectrum analyzer delays action on commands in its input buffer that follow the WAIT command status byte to busy and does not input device-dependent messages. The wait condition is ended in either of two ways.

1. WAIT ends if an end-of-sweep is present. If this occurs, the controller is allowed to request updated spectrum data and be guaranteed that the data has been updated. The request message would be similar to

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.

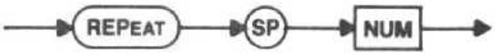
2. WAIT is ended if DCL or SDC (while listener-addressed) is received. This empties the input and output buffer so any commands that follow WAIT are discarded. (See STATUS BYTE later in this section.)

Macro Memory Used — 1 byte.

Interaction — WAIT delays completion of any portion of a message that follows until one of the ending conditions just outlined occurs.

There is no WAIT query.

REPEAT (repeat execution) command



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. For example,

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.

The REPEAT command cannot be used within a macro.

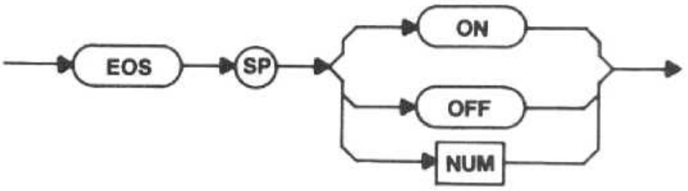
Interaction — A REPEAT loop can only be stopped by DCL. Pressing RESET TO LOCAL does not stop the loop; it only causes execution error messages to be reported if the loop contains front-panel commands. If RESET TO LOCAL is pressed while a message that includes REPEAT is being acted on, the message will only be repeated 256 times. (Since most commands are ignored after the RESET TO LOCAL button is pressed, the REPEAT loop completes quickly).

There is no REPEAT query.

STATUS AND ERROR REPORTING

Two commands (EOS and RQS) control spectrum analyzer service requests. STATUS BYTE reports instrument status in a format that includes both IEEE 488 and the Tektronix Interface Standard for GPIB Codes, Formats, Conventions, and Features. GET is enabled to trigger a new sweep (DT), two queries (ERR?, ERCNT?) return the error codes, a query (EVENT?) returns detailed information about events reported in the last serial poll status byte, two queries and one command (ALLEV?, NUMEV, EVQTY?) specify the identity and quantity of events reported.

EOS (end-of-sweep) command



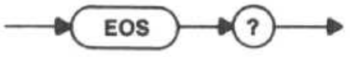
ON — The spectrum analyzer asserts SRQ (if RQS is ON) when a sweep completes.

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

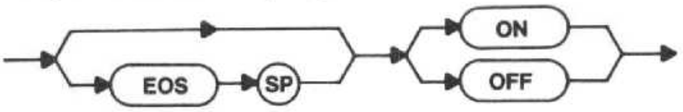
Macro Memory Used — 2 bytes.

Power-up value — Off.

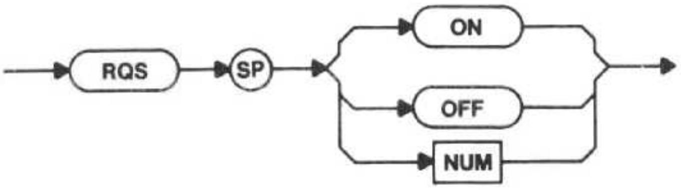
EOS (end-of-sweep) query



Response to EOS query



RQS (request service) command



ON — SRQ is asserted when abnormal status conditions occur.

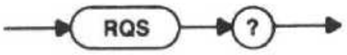
OFF — SRQ is not asserted (is masked) when abnormal status occurs.

Macro Memory Used — 2 bytes.

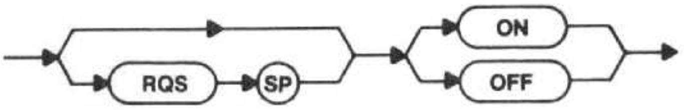
Power-up value — On.

Interaction — RQS is always OFF in the talk-only and listen-only modes.

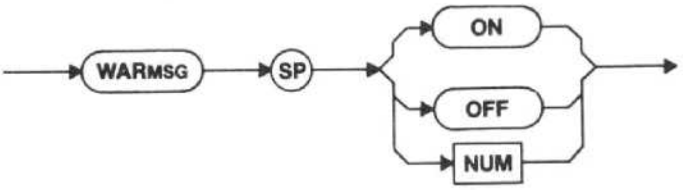
RQS (request service) query



Response to RQS query



WARMSG (warning message) command



ON — All warning messages will be issued (see Table 9-2).

OFF — No warning messages will be issued (see Table 9-2).

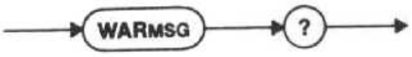
Table 9-2
WARNING MESSAGES

Number	Description
	Execution Warnings
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
110	STEP size out of range — set to maximum
111	SPAN defaulted to minimum span

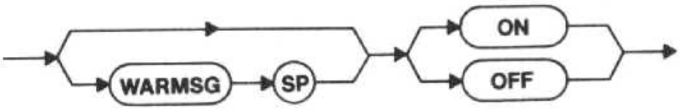
Macro Memory Used — 2 bytes.

Power-up value — On.

WARMSG (warning message) query

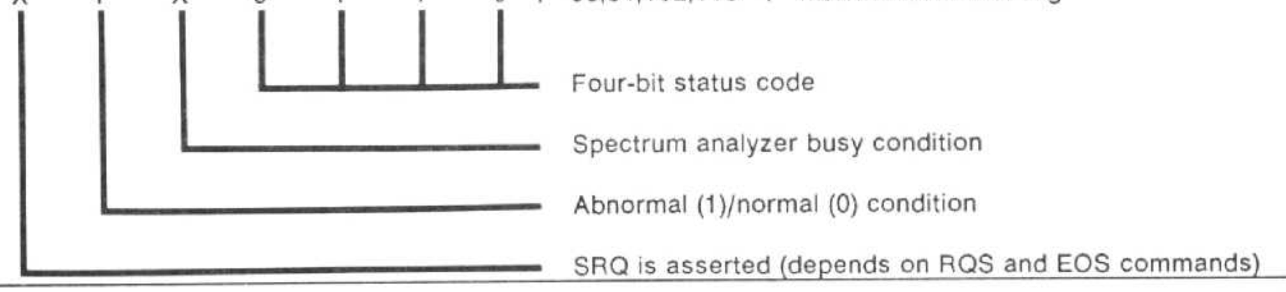


Response to WARMSG query



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)

5559-15

STATUS BYTE (response to serial poll)

When the controller addresses the spectrum analyzer as a talker and sends the SPE (Serial Poll Enable) command, the spectrum analyzer responds by sending its status byte over the bus. The following table explains the messages returned by the spectrum analyzer (1=on, 0=off, X=don't care).

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. The spectrum analyzer sets the warning status because the UNCAL indicator is lit.

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 10-2 in Section 10 of this manual). 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 rtl 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; which, in most cases, should be within a few hundred μs.

The following apply to 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 those set by the rear-panel switches 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 front-panel ADDRESSED light and 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.

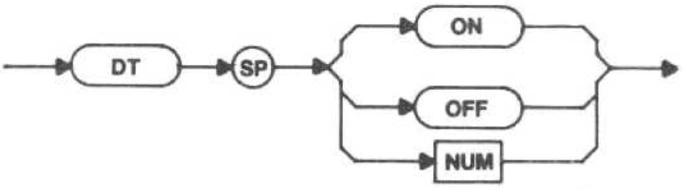
5. GET also requires spectrum analyzer action, so its handshake occurs only when the interrupt can be handled. GET is executed immediately, stopping the current sweep and rearming the sweep. If the spectrum analyzer is busy when GET is received, it will wait until the instrument is no longer busy to 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.

DT (define triggered events) command



ON — GET is enabled to trigger a new sweep.

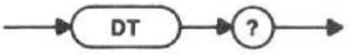
OFF — The response to GET is disabled.

Macro Memory Used — 2 bytes.

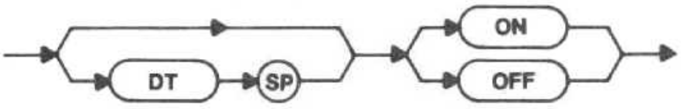
Power-up value — ON.

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

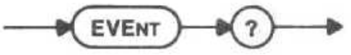
DT (define triggered events) query



Response to DT query

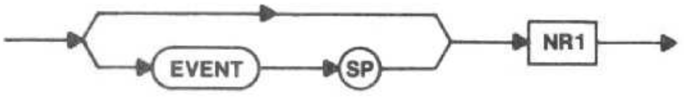


EVENT (event information) query



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).

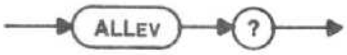
Response to EVENT query



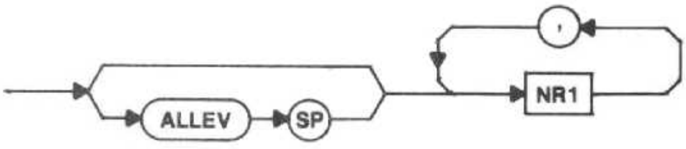
NR1 is an event code defined later in this section in Table 9-4. The EVENT is cleared when the event code is reported.

There is no EVENT command.

ALLEV (all events) query



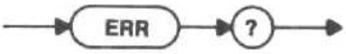
Response to ALLEV query



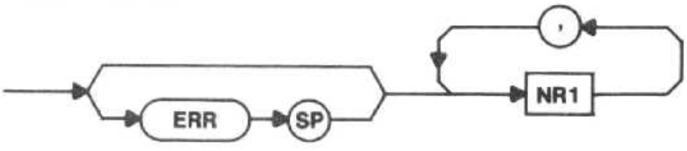
The NR1(s) is the event code defined later in this section in Table 9-4. Events are cleared when their event codes have been reported.

There is no ALLEV command.

ERR (error) query



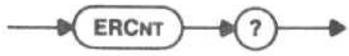
Response to ERR query



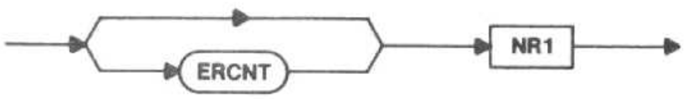
ERR? returns any current error codes, in numerical order. Reading the current code(s) clears the error response. All error codes are listed in Table 9-4.

There is no ERR command.

ERCNT (error count) query



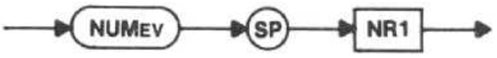
Response to ERCNT query



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

There is no ERCNT command.

NUMEV (number of events) command

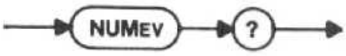


NR1 — This specifies that a fixed number of event codes is to be returned in ALLEV? If fewer events are pending when ALLEV? is executed, the response is filled with zeros to provide the specified number. A value of zero sets the spectrum analyzer to return a variable number of event codes. If the number requested is out of range (maximum number of 39), execution error message 46 will be issued.

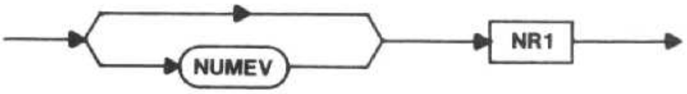
Macro Memory Used — 2 bytes.

Power-up value — 0.

NUMEV (number of events) query

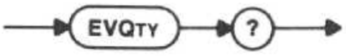


Response to NUMEV query

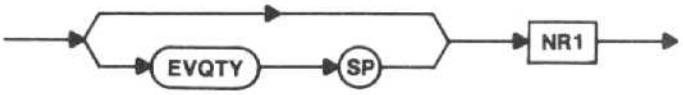


NUMEV? returns the current value for NUMEV.

EVQTY (event quantity) query



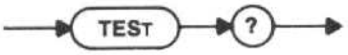
Response to EVQTY query



NR1 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.

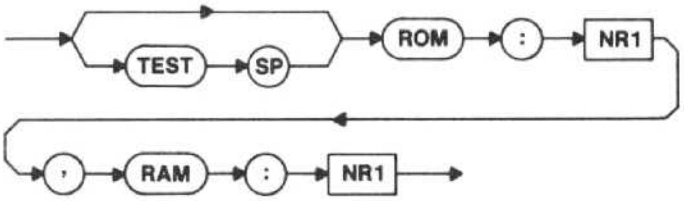
There is no EVQTY command.

TEST (internal test) query



This query checks the system ROM and RAM.

Response to TEST query



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.) After the binary numbers are determined, put them into the conversion chart in Figure 9-1 to identify the IC number. Then, use Table 9-3 to find the correct circuit number and circuit board. The following example shows how to use the conversion chart and Table 9-3. If any ROM or RAM ICs are indicated to be bad, refer this information to qualified service personnel.

Example—Print #A: "TEST?"
Input #A: R$
Print R$

If the TEST query response is

TEST ROM:4112,RAM:18

then,

ROM

1. The binary equivalent of the ROM number 4112 is 0010000001000.

2. Insert this binary number in part A of Figure 9-1 (right-justified). Blocks 6 and 2 will be 01. This indicates that both ROM 6 and ROM 2 are bad; all other ROMs are good.

3. Table 9-3 shows that ROM 6 is U1025 and that ROM #2 is U1010; both located on the GPIB board.

RAM

1. The binary equivalent of the RAM number 10 is 1010.

2. Insert this binary number in part B of Figure 9-1 (right-justified). Blocks 4 and 2 each contain a 1, which indicates that both RAMs 4 and 2 are bad; all other RAMs are good.

3. Table 9-3 shows that RAM 4 is U1020 and RAM 2 is U3020; both located on the Memory board.

There is no TEST command.

Table 9-3

TEST CONVERSION

Device	Chart Location	Circuit Board	Circuit Number
ROM	0	A54 Memory	U3060
	1	A54 Memory	U3060
	2	A56 GPIB	U1010
	3	A56 GPIB	U1010
	4	A56 GPIB	U1020
	5	A56 GPIB	U1020
	6	A56 GPIB	U1025
	7	A56 GPIB	U1025
	8	A56 GPIB	U1035
	9	A56 GPIB	U1035
	10	A56 GPIB	U3015
	11	A56 GPIB	U3015
	12	A56 GPIB	U3020
	13	A56 GPIB	U3020
	14	A56 GPIB	U3030
	15	A56 GPIB	U3030
	16	A54 Memory	U3050
	17	A54 Memory	U3050
RAM	1	A54 Memory	U1010
	2	A54 Memory	U3020
	3	A54 Memory	U1030
	4	A54 Memory	U1020

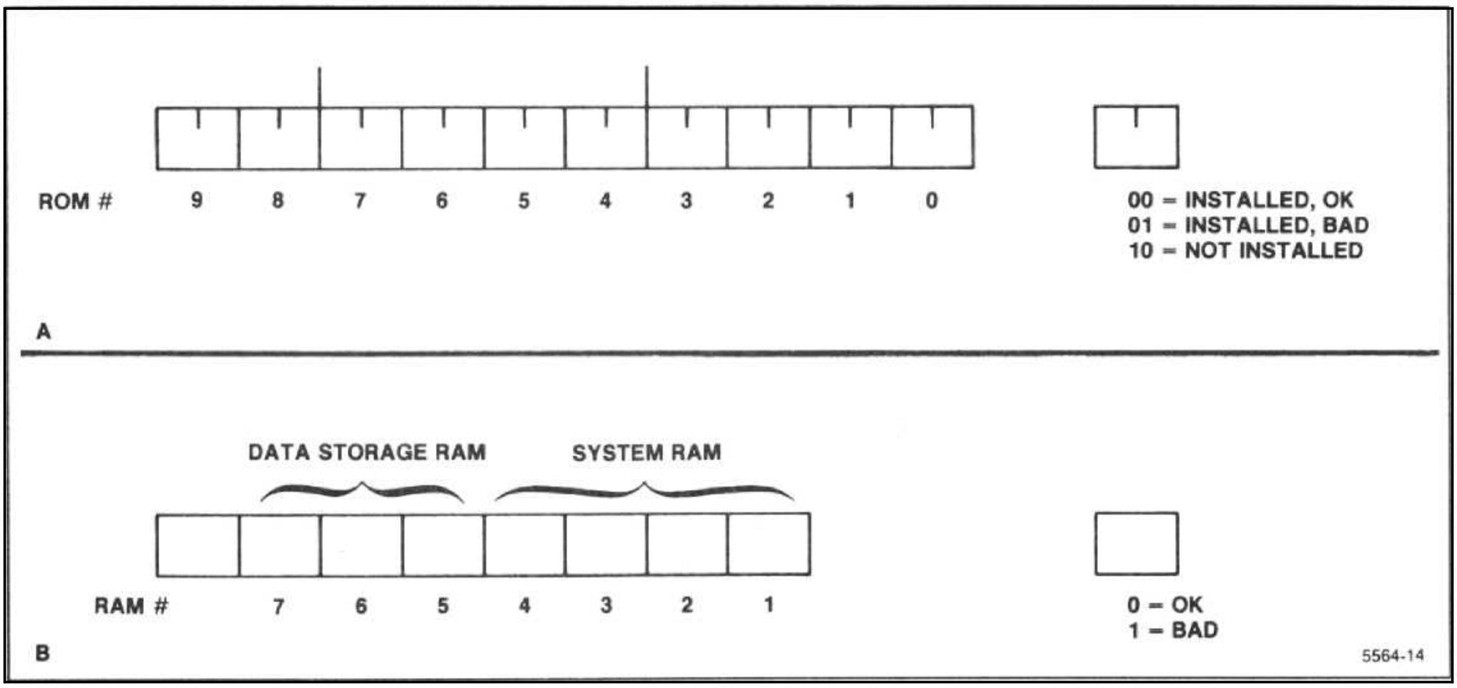


Figure 9-1. TEST Conversion Chart.

Error Codes

The Tektronix Interface Standard for GPIB Codes, Formats, Conventions, and Features specifies device-dependent Error codes by category. Table 9-4 identifies each general category and lists the codes within that category. Following the listing are the specific error messages returned by the spectrum analyzer. Error codes are returned in numerical order as they appear in Table 9-4. When the current code(s) is read, the error response is cleared.

Table 9-4

ERROR AND EVENT CODES

ERROR Code	EVENT Code	Meaning
0	0	No error
		Command Errors
	101	Command header error
	103	Command argument error
	106	Missing argument
	108	Checksum error
	109	Bytecount error
	150	Input buffer overflow
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
		Execution Errors
	201	Command not executable in Local mode
	204	Settings conflict
	205	Argument out of range
	206	Group Execute Trigger ignored (not executed)
	250	Output buffer overflow; remaining output lost
26	250	Output buffer overflow; remaining output lost
27	201	Attempt to execute command in Local mode
28[a]	205	Frequency out of range (FREQ, TUNE, FIRST, SECOND, MMAX, MMIN, MTUNE, MFREQ, STSTOP, STEP)
29[a]	205	FRQRNG out of range
30	205	CRES out of range
31[a]	205	SPAN out of range
32[a]	205	RESBW out of range
33[a]	205	MAXPWR or MINATT out of range
34[a]	205	Level out of range (REFLVL, THRHLD, BWNUM, MVRTDB, MVLFDB)
35[a]	205	VRTDSP LIN out of range
36[a]	205	VRTDSP LOG out of range
37[a]	205	TIME out of range
39[a]	204	IDENTify not allowed in this span/div
40[a]	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[a]	205	NUMEV out of range
47[a]	205	STORE, RECALL, DSTORE, DRECAL, or RDATA out of range
48	204	PHSLK cannot be turned OFF/ON directly with PHSLK command

[a] If this error message is issued, the macro in progress will be aborted.

Table 9-4 (Continued)

ERROR Code	EVENT Code	Meaning
Execution Errors (Continued)		
101[a]	204	Function not available when SGTRAK is on
102[a]	205	Frequency range limited in 75 Ω input — Option 07 only
103[a]	205	Frequency out of range after step
104[a]	204	Bandwidth mode is not available when in linear
105[a]	205	Illegal sweep range
106[a]	205	Argument out of range
109[a]	205	ROFSET out of range
Execution Warnings		
	550	FREQ change caused EXMXR change
	551	SPAN defaulted to MAX
	553	UNCAL light turned on
	554	UNCAL light turned off
	555	Multiple use of display buffer
	560	STEP defaulted to maximum
	561	SPAN defaulted to minimum span
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
54	554	UNCAL light turned off
110	560	STEP size out of range — set to maximum
111	561	SPAN defaulted to minimum span
Internal Errors		
	302	System error
	350	Tuning DAC carry operation failed
	351	Failed to lock 1st LO
	352	Lost 1st LO lock
	353	Recentering failure on unlocking of 1st LO
	354	Calibration failure
	355	Battery-operated RAM checksum error
	382	1st LO tuning system failed
	383	1st LO tuning system recovered from a failure
	386	2nd LO tuning system failed
	387	2nd LO tuning system recovered from a failure
	388	Phase lock system failed
	389	Phase lock system recovered from a failure
	394	IF count failed
	395	IF count recovered from a failure
	396	Power supply out of regulation
	397	Power supply regained regulation
	398	Frequency reference unlocked
	399	Frequency reference re-locked
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

[a] If this error message is issued, the macro in progress will be aborted.

Table 9-4 (Continued)

ERROR Code	EVENT Code	Meaning
		Internal Errors (Continued)
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
	401	Power on
	402	Operation complete
	403	User request
97	401	Power just came on
98	402	Operation complete (end of sweep)
100	403	SRQ was requested
		Marker Execution Errors
107[a]	204	MKTIME not available unless in zero span mode
108[a]	205	MKTIME out of range
120[a]	205	Frequency out of range because marker is on inactive trace
121[a]	204	Function not available when marker is on an inactive trace
122[a]	204	Function not available when marker is off
123[a]	204	Function not available when delta marker is off
124[a]	204	Function not available in BWMODE
125[a]	204	Function not available when marker is on a B−SAVEA trace
126[a]	204	Function not available when in dB/Hz
		Marker Execution Warning
	558	Signal find commands could not find a signal
130	558	Signal find commands could not find signal (MLFTNX, MRGTNX, MFBIG, HRAMPL, LRAMPL, PKFIND, MVRTDB, MVLFDB)
		Macro Execution Errors
160	204	Command not available when entering a macro
161	204	Command only available when entering a macro
162	204	Command only available when a macro is stopped
163	204	Memory full — macro entry refused
164	204	Number is already used
165	204	No macro is stored at NUM
166[a]	204	No more MDATA commands to read
167[a]	204	Too many GOSUBs with no return
168[a]	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[a]	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

[a] If this error message is issued, the macro in progress will be aborted.

Table 9-4 (Continued)

ERROR Code	EVENT Code	Meaning
		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
	556	Multiband sweep started
	557	Multiband sweep stopped
	558	Start frequency changed
150	556	Multiband sweep started
151	557	Multiband sweep stopped
152	559	Start frequency changed to 1.7 GHz because of using internal mixer

HELPS AND HINTS

This section covers some techniques for programming the spectrum analyzer. This information will speed your progress in putting the spectrum analyzer to work solving your measurement problems.

Notes on the Program Examples

Many combinations of languages and controllers may be used for programming the spectrum analyzer over the GPIB. The examples in this manual are given in the BASIC language; specifically, in the version of BASIC used by the Tektronix 4041 System Controller. Other versions of BASIC will probably have similar syntax.

Some ontrollers may require additional statements to set up data transfer over the GPIB. For example, 4041 BASIC contains optional OPEN and SET DRIVER statements that set GPIB communications parameters. These statements are not included in the program examples because the default parameter values are suitable for simple programs, such as the examples. The statements may be helpful in writing programs involving multiple I/O channels, such as tape files or other GPIB-controller instruments. Refer to the manual of the controller you are using for the statements needed in your application.

In these examples, the spectrum analyzer's primary address is assumed to be 1. See Section 1 of this manual for instructions on how to set the GPIB address switches.

Some of the lines in the examples extend beyond the column width limitations. When this happens the line is broken where a natural space occurs, and the remainder of the line is indented on the immediately following line. Because of this natural space, a space must be added between the two example lines when joining them to form one program line.

PROGRAMMING TECHNIQUES

Signal Processing

The instrument should be in the Single Sweep mode and not be sweeping during any of the signal processing commands or queries. (The signal processing commands are HRAMPL, LRAMPL, MFBIG, MLFTNX, PKFIND, MMAX, MMIN, MRGTNX, MVLFDB, MVRTDB, CURVE, WAVFRM, DPRE, DCOPY, FIBIG, LFTNXT, RGTNXT, FMAX, and FMIN.)

Wrong Way — The following is an unsuccessful command line to try to center a signal.

```
INIT;FREQ 100MHZ;SPAN 1MHZ;FMAX;POI?
```

In the preceding example, the spectrum analyzer will not have made a full sweep at 100 MHz frequency and a span of 1 MHz before the FMAX;POI? is executed because there is no WAIT included in the command, and you don't know where the sweep was when it went to 100 MHz. The following command line is better, but you will still have an error.

```
INIT;FREQ 100MHz;SPAN 1MHZ;WAIT;FMAX;POI?
```

In this example, you still don't know where the sweep was when it went to 100 MHz; the screen may not have been updated. The WAIT will tell the spectrum analyzer to wait for the end of the sweep before looking for the maximum point (FMAX). The sweep may not have swept the full 10 divisions with a span of 1 MHz and a center frequency of 100 MHz.

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 100M;SPAN 1M;SIG;SIG;WAI;FMA;POI?
```

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 executing the FMA;POI? commands.

RUNNING PROGRAMS WITHOUT A CONTROLLER (MACROS)

Most of the programming examples in this section can be incorporated into macroinstructions (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 macros will be lost if the battery-powered memory is interrupted (as when the battery is removed for long-term storage).

There are two harmonic test macros in the pullout pages at the back of this manual that were prepared to be stored in the spectrum analyzer and used without the need of a controller.

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. This is all 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 NRFD 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

```
 80    Z=1 ! ADDRESS OF SPECTRUM ANALYZER
100    For i=1 to 10
110      Print #z:"FREQ ";i;"GHZ"
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.

```
 80    Z=1 ! ADDRESS OF SPECTRUM ANALYZER
100    Dim f(10,2)
110    For i=1 to 10
120      Print #z:"FREQ ";i;"GHZ;FRQRNG?'
130      Input #Z:f(i,1)
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 assures that the spectrum analyzer updates the FRQRNG query response before handshaking out array element f(i,1) 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

```
 80    Z=1 ! ADDRESS OF SPECTRUM ANALYZER
100     Dim p(5,2)
110     Print #z:"SIGSWP"
120     For i=1 to 5
130       Print #z:"FREQ
           ";i;"GHZ;SIGSWP;WAIT;FMAX;POINT?"
140       Input #z:p(i,1),p(i,2)
150   Next i
```

Line 110 — Sets 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.

Line 130 — Illustrates 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 10-1 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 execution with the sweep is recommended, there is another method. This alternative

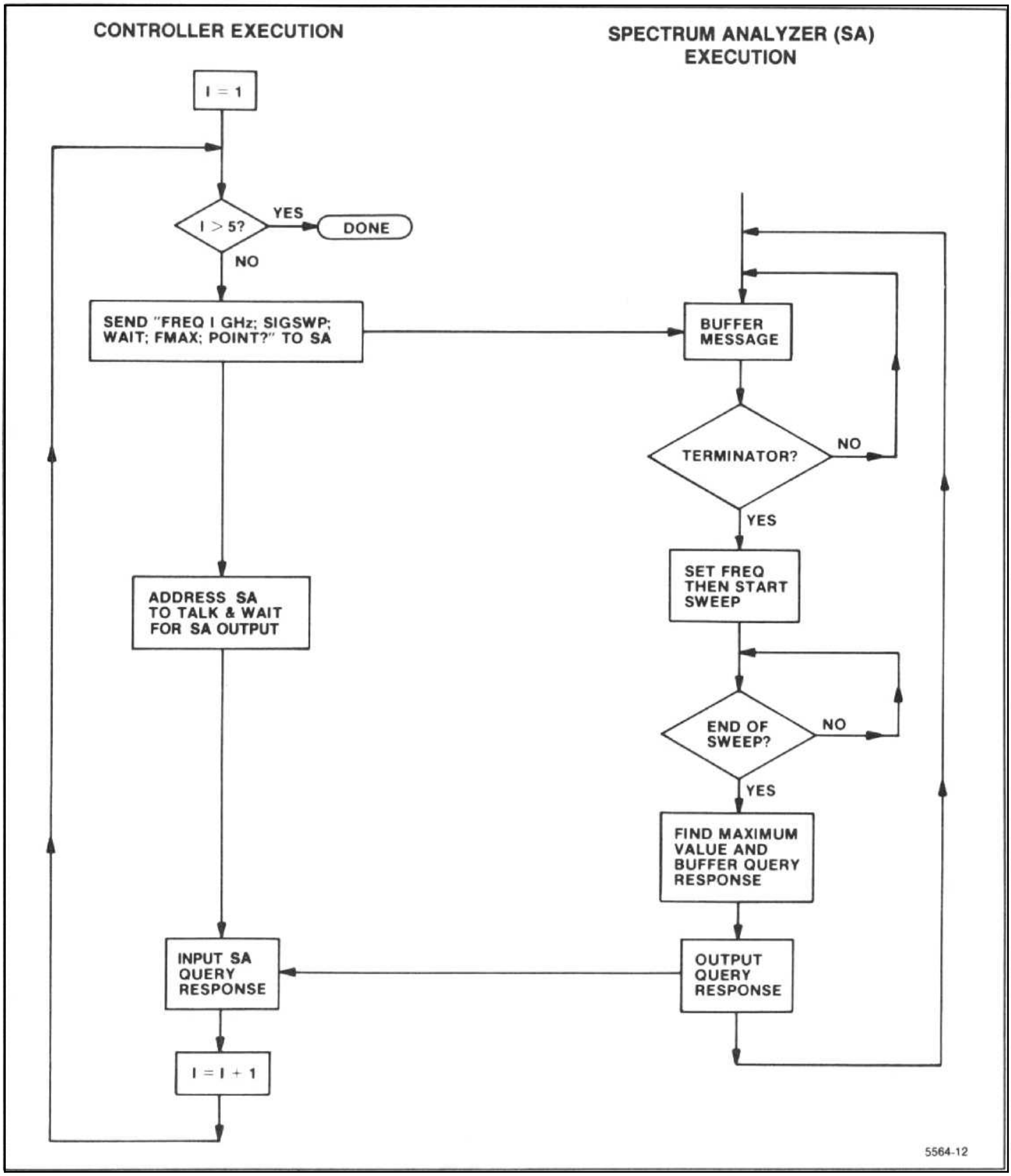

CONTROLLER EXECUTION
SPECTRUM ANALYZER (SA) EXECUTION
I = 1
I > 5?
YES
DONE
NO
SEND "FREQ I GHz; SIGSWP; WAIT; FMAX; POINT?" TO SA
ADDRESS SA TO TALK & WAIT FOR SA OUTPUT
INPUT SA QUERY RESPONSE
I = I + 1
BUFFER MESSAGE
TERMINATOR?
NO
YES
SET FREQ THEN START SWEEP
END OF SWEEP?
NO
YES
FIND MAXIMUM VALUE AND BUFFER QUERY RESPONSE
OUTPUT QUERY RESPONSE
5564-12

Figure 10-1. Synchronizing controller and spectrum analyzer for data acquisition.

may be necessary for 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 tasks while the spectrum analyzer is acquiring data. In such cases, the controller can enable the spectrum analyzer end-of-sweep SRQ to synchronize data acquisition with the sweep. The following example just shifts the WAIT from the spectrum analyzer program to the 4041 BASIC program to exercise the SRQ. It could be modified, however, to busy the controller with some other task, using the SRQ subroutine to test the status byte and perform input when end-of-sweep status is detected.

```
 80    Z=1 ! ADDRESS OF SPECTRUM ANALYZER
100    Dim p(5,2)
110    On srq then call srq_hndl
120    Enable srq
130    Print #z:"SIGSWP;EOS ON"
140    For i=1 to 5
150      Print #z:"FREQ ";i;"GHZ;SIGSWP"
160      Wait
170      Print #z:"FMAX;POINT?"
180      Input #z:p(i,1),p(i,2)
190      Next i
200    End
300 Sub srq_hndl
310    Integer status,adr
320    Poll status,adr;z
330    Resume
340    End
```

With no WAIT command following SIGSWP in line 150, the spectrum analyzer is ready to buffer another message. But, the controller does not send it immediately because of the WAIT statement in line 160. The SRQ that the spectrum analyzer asserts at the end of its sweep, which was enabled in line 130, triggers the controller to perform a serial poll (lines 300 through 340) and then send the message in line 170.

INPUT: An SRQ Alternative

An INPUT 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 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).

```
 80    Z=1 ! ADDRESS OF SPECTRUM ANALYZER
100    Dim p(5,2)
110    Print #z:"SIGSWP"
120    For i=1 to 5
130      Print #z:"FREQ ";i;"GHZ;SIGSWP;WAIT"
140      Input #z:d$
150      Print #z:"fmax;poi?"
160      Input #z:p(i,1),p(i,2)
170    Next i
180    End
```

Here the WAIT is put back into the spectrum analyzer message and the INPUT statement in line 140 stalls the controller while the spectrum analyzer makes a sweep. d$ serves the purpose of a dummy string; the data is not input until line 160.

BINARY WAVEFORM TRANSFER

These examples cannot be used as presented here in customized macros because they contain the CURVE and WFMPRE commands

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 10-1. The gains possible by using binary are not hard to achieve. Here's how.

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 4041 routine that handles block binary, enter the following.

```
 80    Z=1 ! ADDRESS OF SPECTRUM ANALYZER
500 !  GET SPECTRUM ANALYZER CURVE OUTPUT
510    Integer w(1000) ! Declare integer
       array
520    Print #z:"WFMPRE ENC:BIN"
530    Input prompt "CURVE?" using
       "FA,+8%"dels "," #z:header$,w
540 !  Waveform heading is in header$,
       data in w array
```

Line 510 — Declares an integer array large enough for a full 1000 points.

Line 520 — Requests the spectrum analyzer to format data in a binary format.

Line 530 — Uses the input prompt to request a curve. The free field input for the header string is delimited from the data a comma with the dels clause, and enters the array w in an 8-bit word format with the image specifier (+8%). The byte count words and checksum are input and checked automatically. Checksum failure results in a trapable error interrupt to the 4041.

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.

```
 80   Z=1 ! ADDRESS OF SPECTRUM ANALYZER
600 !  TRANSMIT A BINARY CURVE TO THE
       SPECTRUM ANALYZER
610    Print using "+8@" #z:w
```

Line 610 — Sends the image specifier +8@ followed by the binary numbers in array w, after which it asserts EOI. EOI causes the spectrum analyzer to act on the message. The CURVE header is omitted; it is not required, but would be accepted if sent. The spectrum analyzer buffers the last byte (hex FF), but does not put it into digital storage.

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.

```
100 Z=1  ! ADDRESS OF SPECTRUM ANALYZER
110 Integer w(130)
120 Print #z:"STORE 1"
130 Input prompt "RDATA? FPSET:1" using "FA,FA,+8%"
    dels ":" #z: h1$, h2$, w
140 Print #z: "RDATA FPSET:2:";
150 Print using "+8%" z: w
```

Line 110 — Dimensions an integer array to hold the settings data. This must be an integer array.

Line 120 — Places present instrument settings in memory 1 to insure there is something to transfer.

Line 130 — Read the settings from the instrument. The delimiter is set to a colon (:) with the dels clause. Since there is a link argument in the returned header before the binary data, the header will be input as two strings (h1$ and h2$ as specified by FA,FA in the using clause). The image specifier for the array (+8%) selects binary block format and places each 8-bit byte in one location in array w. The byte count and checksum are input and checked automatically. A checksum error results in a trappable interrupt to the 4041.

Line 140 — Send the command header to the instrument. The terminating semi-colon (;) stops the 4041 from asserting EOI, so the header and the data that follows will be received by the spectrum analyzer as a single message.

Line 150 — Send the settings data to the instrument. The image specifier (+8%) selects binary block format and sends the right-most 8 bits of each element of w to the spectrum analyzer. The percent sign (%) and byte count preceding the data and the checksum following the data are sent automatically by the 4041.

SCALING, SAVING, AND GRAPHING WAVEFORM DATA

The spectrum analyzer waveform data outputs 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 spectrum analyzer binary output program given above. 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. (This example cannot be used in customized macros as presented here because it contains the CURVE and WFMPRE commands.)

```
100  ! GET AND SCALE SPECTRUM ANALYZER
       BINARY CURVE OUTPUT
110     Delete var w,m
120     Print #1:"WFMPRE?"
130     Input deln "" #1:n,x3,x2,x2,y3,y2,y1
140     Goto 150
150     Dim m(n,2)
160     Integer w(n)
170     Gosub wave_in
180     For i=1 to n
190       m(i,1)=x1+x2*(i-1-x3)
200       m(i,2)=y1+y2*(w(i)-y3)
210       Next i
220     Stop
230  Wave_in: print #z:"WFM ENC:BIN"
240     Input prompt "CURV?" using "FA,+8% dels ","
        #z:head$,w
250     Return
```

Line 110 — Clears the waveform arrays.

Line 120 — Requests the waveform preamble and a binary curve.

Line 130 — Inputs the spectrum analyzer WFMPRE response, storing the first seven numbers it finds as variables n, x3, x2, etc.

Lines 150, 160, and 170 — Calls subroutine wave_in 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 and 240 — Uses the input prompt to request a curve. The field input for the header string is delimited from the data by a comma (,) with the dels clause, and enters the array w in an 8-bit word format with the image specifier (+8%). The byte count words and checksum are input and checked automatically. Checksum failure results in a trapable error interrupt to the 4041.

Saving the Scaled Array

When you transfer an array to tape, first find and mark an adequate tape file. These statements do the job.

```
100  Open #2:"WAVEFORM (OPEN=REP,CLIP=YES)"
110  Print #2:m
120  Close #2
```

These statements return the data from the tape.

```
100  Open #2:"WAVEFORM"
110  Input #2:m
120  Close #2
```

Storing Settings

If a particular series of settings are commonly used in an application, it is recommended that these settings be stored within the spectrum analyzer using the STORE command. This practice will save program storage in the controller, programming time, and bus transfer time. The settings can be recalled whenever needed with the RECALL command. If you need to store more settings than can be held in the spectrum analyzer at one time, the settings data can be conveniently exchanged between the controller and the analyzer with the RDATA command and query.

Using PLOT

The spectrum analyzer can generate a plot of the display directly on a

- Tektronix 4662 Option 01
- Tektronix 4662 Option 31
- Tektronix 4663 (in the 4662 emulation mode)
- Hewlett-Packard HP7470A
- Hewlett-Packard HP7475A
- Hewlett-Packard HP7580B
- Hewlett-Packard HP7585B
- Hewlett-Packard HP7586B
- Gould 6310
- Gould 6320

All selected waveform, graticule, marker, and crt readout data can be plotted. PLOT? sent to the spectrum analyzer causes it to output the plotter code when addressed as a talker. Address the plotter as a listener, then monitor the EOI line to allow the controller to cause a plot to be generated. The following routine assumes the spectrum analyzer is at address z, the plotter is at address P, and that the plotter type has been selected with the PTYPE command.

```
 70  Z=1 ! ADDRESS OF SPECTRUM ANALYZER
 80  P=3 ! ADDRESS OF PLOTTER
100  Print #z:"GRAT ON;RED ON;PLOT?"
110  WBYTE ATN(64+z,32+p)
120  On EOI(5) then gosub 200
130  Enable EOI(5)
140  Wait
150  Stop
200  WBYTE ATN(unl,unt)
210  Resume
```

Line 100 — Illuminates the graticule, turns the crt readout on, and readies the spectrum analyzer to send a waveform to the plotter.

Line 110 — Makes the spectrum analyzer a talker and the plotter a listener.

Line 200 — Untalks the spectrum analyzer and unlistens the plotter.

Using PLOT With Macros

Macro Readout Buffer — When the macro readout buffer is displayed, it alone will be plotted—nothing else will plot, even the graticule or waveform.

Using DSLINE — If DSLINE is used to turn off the normal top, second, or bottom readout lines or replace them with information from the macro readout buffer, no labels associated with the lines will be plotted.

With DSLINE ON, any middle line to be plotted will always be plotted in the location where the normal second line would be.

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. (This occurs in Section 7 under the Interaction part of the CURVE command and under Display Data Point Commands Interaction in Section 8).

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. (This example cannot be used in customized macros as presented here because it contains the CURVE command.)

```
 80   Z=1 ! ADDRESS OF SPECTRUM ANALYZER
100 !  BUFFER DEMO
110    Print #z:"SIGSWP"
120    Print #z:"CURVE ";a,";FIBIG;POINT?"
130    Input #z:B1,B2
```

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 from overwriting the CURVE commmand.

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 10-2. 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 destinations branching from the left of the buffer.

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.

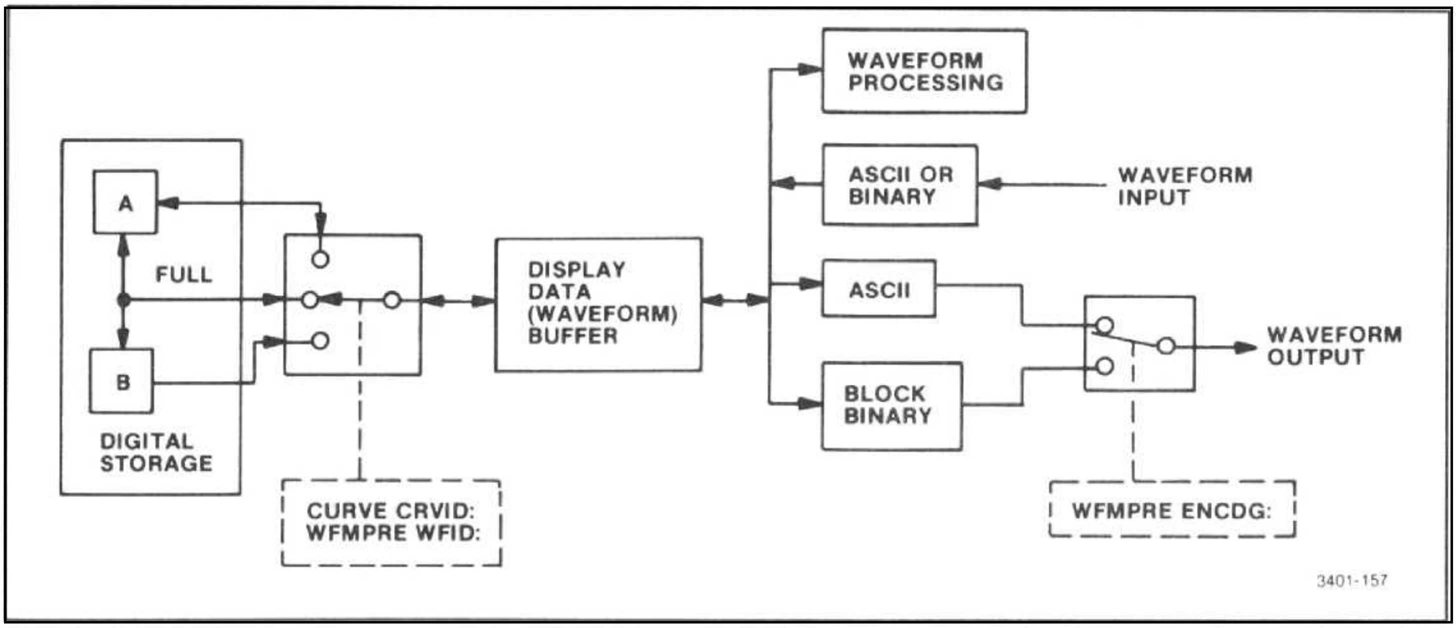


Figure 10-2. How multiple use of the display data buffer is controlled.

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.

```
 80   Z=1 ! ADDRESS OF SPECTRUM ANALYZER
100 ! Waveform processing and I/O
110   Dim b(1000)
120   Print #z:"CURVE?"
130   Input #z:b
140   Print #z:"CURVE ";A;
150   End
```

Line 120 — Requests a curve, which the spectrum analyzer buffers.

Line 130 — Inputs the curve before it is overwritten by line 140. The semicolons enclosing A at the end of line 140 instruct the controller to squeeze out unneeded spaces between the numbers as the PRINT statement transmits array A. Without a semicolon immediately after the array variable, this line does not run properly. With this semicolon, the controller places a space between numbers; the spectrum analyzer accepts the space (or other format characters), as well as a comma, for a delimiter.

FINDING SIGNALS WITH WAVEFORM PROCESSING

The waveform processing functions in the spectrum analyzer allow 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 try to help you gain that understanding and to use the power of the spectrum analyzer in your application with more accurate and predictable results.

Understanding How Waveform Processing Works

There are two sets of signal processing commands; commands that move the marker and commands that move the display data point. (See Sections 5 and 8 of this manual for more information.) Each set of commands uses a different signal-finding algorithm. Commands that move the display data point use the same algorithm as the commands used in previous 490-series spectrum analyzers. This algorithm looks for the signal shape produced by a continuous wave (cw) signal passing through the resolution filters of the spectrum analyzer. Complex spectra, such as those produced by amplitude, frequency, or pulse modulation or by signals with significant amounts of drift or phase noise, may be missed.

In contrast, the marker-related signal processing commands allow, through the STYPE command, a choice of signal-processing 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 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 particular 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.

Another difference between the display-point commands and the marker commands is the output units. The POI? query reports in screen units; 1–1000 horizontally and 25–225 vertically. The marker queries report the actual frequency and amplitude of the marker.

Two other factors affect the results. The first of these 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.

After a signal is found, it is often moved to center screen and to the reference level with either the CENSIG or TOPSIG commands (for the display-point-related functions). The accuracy of the vertical display and the reference level determine how closely the point of interest can be moved to the reference level. For CENSIG the span/division determines how closely the point of interest can be moved to center screen. On an active trace, the oscillator frequencies are counted at the marker, and the values are used to set the center frequency for the MCEN command. Thus, MCEN will generally be more accurate than CENSIG. These commands may be applied repetitively for greater accuracy.

Setting the Threshold

Both sets of 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-point commands 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.

The value of THRHLD can also be set 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 instrument's 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.

Acquiring Data for Waveform Processing

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.

To use digital averaging, select CRSOR AVG or set the peak/average line above the noise floor and select CRSOR KNOB. Successive samples are averaged to obtain the value displayed at each digital storage point. The number of samples averaged increases as the sweep is slowed. This number is given by (Time/div)/100/9 μs.

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 or the RGTNXT command. Find successive signals by repeating the same command. All signals in the display have been found when the RGTNXT command sets the display data point to 1001,0 or the MRGTNX command generates an SRQ message of NO SIGNAL FOUND. If the service request is disabled, either through SGERR OFF or RQS OFF, the MRGTNX? query can be used to determine whether the MRGTNX command found (FOUND will be returned) or did not find (FAILED will be returned) a signal.

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.

Because the oscillators are counted on an active trace, the marker-related commands are slower on an active trace than on a stored trace. Speed will be increased, at the expense of some accuracy, by turning BVIEW OFF and commanding SAVEA ON after the single sweep and before the search is started. SAVEA OFF should be sent before the next sweep.

A signal search program using RGTNXT is given in connection with the description of the REPEAT command.

Measuring Signal Frequency with COUNT

To measure the frequency of a signal, center the signal on screen or move the marker to the signal. This can be done by using a FMAX;CENSIG combination or 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 number returned will have a least significant digit of the current count resolution (CRES command).

```
100   Print #z:"SIG;WAI;FMA;CEN"
110   Print #z:"SIG;WAI;FMA;POI?;FMIN;POI?"
120   Input #z:X,Y,X1,Y1
```

```
130   If Y−Y1 <50 then 200
140   Print #z:"COUNT;COUNT?"
150   Input #z:"f

200 ! CODE TO HANDLE LOW SIGNAL LEVEL
      CONDITION
```

Using COUNT→CF

To go to a narrow span from a wide span, without having to span down or re-center the signal, etc., use the CNTCF command. Position the signal as indicated previously for COUNT, check for 20 dB above the noise level, and send CNTCF. Now select the desired span/div. It is important to make sure sufficient count resolution (CRES) is selected for the desired span/div. The CRES selected should be less than 0.1 of the span/div desired. If many different span/divs are being used, it is more convenient to leave CRES set to 1 Hz.

```
100   Print #z:"SIG;WAI;FMA;CEN"
110   Print #z:"SIG;WAI;FMA;POI?;FMI;POI?"
120   Input #z:X,Y,X1,Y1
130   If Y−Y1 <50 then 200
140   Print#z:CNTCF;SPAN ";s

200 ! CODE TO HANDLE LOW SIGNAL LEVEL
      CONDITION
```

Using MCEN

To span down around a signal without having to manually span down, re-center the signal, etc., use the MCEN command after first having selected the desired signal with one of the marker signal finding commands described previously. Then select the desired span/div.

Higher Center Frequency Drift Rate After Tuning

After a large center frequency change (more than 500 MHz), oscillator drift can cause center frequency errors. This is especially noticeable at slow sweep speeds at spans just above the point where the 1st LO is phase locked (which occurs at either 200 kHz/div or 50 kHz/div, depending upon band). Additional delay time after a large frequency change, beyond that built into the spectrum analyzer firmware, may be needed to reduce the drift rate.

Noise

Noise can be overcome in several ways. To reduce noise peaks, smooth the data by averaging in digital storage. Averaging is enabled by the CRSOR command; use CRSOR AVG or CRSOR KNOB and set the cursor above the noise by turning the front-panel knob. Further smoothing is possible by slowing the sweep (TIME command) so that the amount of data averaged for each point in digital storage is increased.

Noise peaks can also be reduced (smoothed) by the video filters.

There is an alternative to smoothing the data. The signal search commands can include a parameter 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. But, how do you find this level that is not too high and not too low? There is no one level that will work in every case. 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.

In practice, a combination of these methods may be applied to handle varying conditions. For example, smooth the data with the narrow video filter and average it as it is acquired to enable the search routine to find signals close to the noise floor.

USING REPEAT FOR SIGNAL TRACKING AND SEARCHES

NOTE

The REPEAT command cannot be used within customized macros because it causes a loop condition. Therefore, the examples shown for this discussion cannot be used within macros.

The REPEAT command adds another dimension to spectrum analyzer messages; the number of times a loop through a string of commands or queries. Several uses are suggested here.

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.

```
 80   Z=1 ! ADDRESS OF SPECTRUM ANALYZER
100 !   SPECTRUM SEARCH
```

```
110     Print #z:"POINT 0"
120     Print #z:"RGTNXT;POINT?;REPEAT 20"
130 !   *****************
140 !   INSERT ANY OTHER TASKS
150 !   ******************
300     Dim p(20,2)
310     Input #z:p
```

Line 120 — The spectrum analyzer buffers the query responses as it executes the loop.

Line 310 — Inputs the signal points as a string, delimited by semicolons.

The number of query responses that can be buffered depends on the query and the message sent to the spectrum analyzer. Messages and responses share buffer space. Long messages will leave less space for responses than short messages.

Messages on the Crt Using RDOUT

The spectrum analyzer accepts either a single set or a double set of quote marks to delimit the crt message. With the 4041 controller, use single quotes around the message inside the RDOUT command.

```
100  Print #z:"RDOUT 'SET THE PEAK/AVERAGE
    CONTROL'"
```

This is necessary because the 4041 uses a double set of quotes to set off the message following the colon in the Print statement. A variation gets around this if you want quote marks to appear on the spectrum analyzer crt.

```
100     Print #Z:"RDOUT 'PRESS ""RETURN TO
       LOCAL""'"
```

The controller strips off the first set in each double set of quote marks and transmits the second set of each double set for the display.

The RDOUT message continues to be displayed if the spectrum analyzer remains under remote control. To demonstrate the above messages by themselves, add the following statement.

```
110     Goto 110
```

To scroll the RDOUT message to the top of the spectrum analyzer screen, insert the following statement.

```
105     Print #z:"RDOUT''"
```

Using CAL Over the Bus

The CAL function activated from the front panel does two things. It directs the operator in the adjustment of the four screwdriver adjustments (AMPL and LOG CAL, and Vertical and Horizontal POSITION). CAL also performs an automatic calibration of the relative amplitude, frequency, and noise bandwidth of the resolution bandwidth filters. The CAL command allows these functions to be separated so that either the adjustments or the automatic calibration will be done.

To recreate the front-panel calibration function over the bus requires that five commands be sent to the instrument; four for the adjustments and one for the automatic calibration. User prompt messages are displayed on the crt screen. Since the spectrum analyzer is in the Remote mode, the controller keyboard must be used to go to the next adjustment step. A device clear is sent to the spectrum analyzer to terminate the execution of each adjustment command. Note that HPOS could be deleted without affecting the other adjustments, but the others should be done in the recommended order.

```
 80     Z=1 ! ADDRESS OF SPECTRUM ANALYZER
100     Print #z:"RDO 'CONNECT CAL OUT TO RF
        INPUT'"
110     Print #z:"RDO 'HIT RETURN AFTER
        COMPLETING EACH STEP'"
120     Input b$
130     Print #z:"CAL AMPL"
140     Input b$
150     WBYTE DCL
160     Print #z:"CAL HPOS"
170     Input b$
180     WBYTE DCL
190     Print #z:"CAL VPOS"
200     Input b$
210     WBYTE DCL
220     Print #z:"CAL LOG"
230     Input b$
240     WBYTE DCL
250     Print #z:"CAL AUTO"
260     End
```

COMPARING FREQ AND TUNE

The FREQ command argument is in absolute frequency, while the TUNE command argument is in 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 limited to the current frequency range.

USING THE TIME MEASUREMENT FEATURE

This instrument employs a special time measurement feature that is available whenever the instrument is in the zero span mode of operation or ZEROSP or ZETIME are 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/division. 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 1 ms/division, 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 while ZEROSP is on.

The MKTIME command and query that set and read the marker time, are only valid with ZEROSP ON 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 −200 if the time value is unavailable, +200 if ZEROSP is OFF or ZETIME if 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 – 21 GHz). 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. The multiband function is also available if the external mixer is in use; however, if the external mixer is being used, the range that can be swept is 10 KHz – 21 GHz. In the waveguide bands, the sweep range is restricted to a single band, since each band normally requires a different mixer.

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 and 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. During multiband sweep operation, MULTIBD is displayed at the bottom center of the screen.

If the start frequency is less than 3 GHz and the stop frequency is greater than 7.1 GHz, the 3 GHz – 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 from normal. 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/division value refers to a division of the sweep that gathered the data, not to a division of 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.

- Using the FREQ command to enter a frequency
- Recalling a setting with RECAL with a sweep that falls within one band
- Entering a sweep that falls within one band, using the STSTOP command
- Entering FRQRNG
- Changing 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 Section 9 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; i.e., 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 10-1). This is the time the spectrum analyzer is busy following receipt of the end-of-message terminator (EOI or LF, depending on the switch). 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 (e.g., 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 10-1

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
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 preselector/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 string)	3.0 s
ASCII (input as numbers)	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
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

IEEE STD 488 (GPIB) SYSTEM CONCEPTS

The General Purpose Interface Bus (GPIB) is a digital control bus that allows efficient communications between self-contained instruments or devices interconnected in an instrumentation system. The GPIB is an interface system independent of the stimulus or measurement functions incorporated in any instrument.

Instruments or devices designed to operate on the digital control bus must be developed according to the specifications contained in IEEE Std 488-1978, "IEEE Standard Digital Interface for Programmable Instrumentation." At Tektronix, the IEEE 488 digital interface is commonly known as the General Purpose Interface Bus (GPIB). This section discusses the basic concepts of the GPIB. (For complete specifications, refer to the IEEE Std 488-1978 standard, published by the Institute of Electrical and Electronics Engineers, Inc.).

The GPIB has four elements; mechanical, electrical, functional, and operational. Of these four, only the last is device-dependent. Operational elements state the way in which each instrument reacts to a signal on the bus.

MECHANICAL ELEMENTS

The IEEE Std 488 defines the GPIB connector and cable assembly as the mechanical elements of the instrumentation system. Standardizing the connector and cable assembly ensures that GPIB-compatible instruments can be physically linked together with complete pin compatibility. The connector has 24 pins; sixteen active signal lines, seven interlaced grounds, and 1 shield connection. Standard connector pin arrangement and nomenclature for the digital control signals are illustrated in Figure A-1.

The cable that attaches to the GPIB connector must be no longer than 20 meters maximum with no more than fifteen peripheral devices (including a GPIB controller) connected at one time. The interconnecting cable assembly, which is offered as an optional accessory to the spectrum analyzer, is provided with both a plug and receptacle connector type at each end of the cable to allow either a star or linear bus structure. Contact your local Tektronix Field Office or representative for cable ordering information. Connectors may be rigidly stacked, using standard counter-bored captive screws.

ELECTRICAL ELEMENTS

The voltage and current values required at the connector nodes on the bus are based on TTL technology. The power source is not to exceed +5.25 V referenced to logic ground. The standard defines the logic levels as follows.

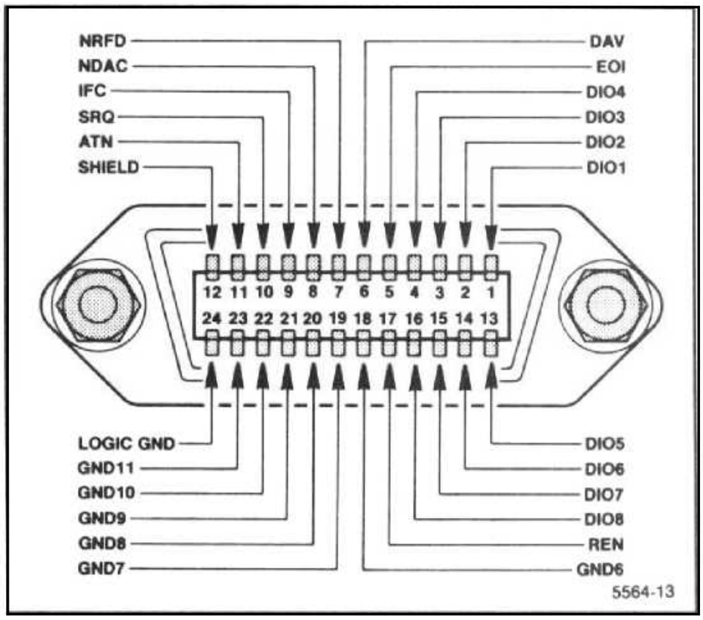


Figure A-1. IEEE Std 488 (GPIB) connector.

1. Logical 1 is a true state, low voltage level (⩽+0.8 V), signal line is asserted.

2. Logical 0 is a false state, high voltage level (⩾+2.0 V), signal line is not asserted.

Messages can be sent over the GPIB as either active-true or passive-true signals. Passive-true signals occur at a high voltage level and must be carried on a signal line using open-collector devices. Active-true signals occur at a low voltage level.

FUNCTIONAL ELEMENTS

The functional elements of the GPIB cover three areas.

1. The ten major interface functions of the GPIB, are listed in Table A-1. Each interface function is a system element that provides the basic operational facility through which an instrument can receive, process, and send messages over the GPIB.

2. The second functional element is the specific protocol by which the interface functions send and receive their limited set of messages.

3. The logical and timing relationships between allowable states for all interface functions is the third area covered.

Table A-1

MAJOR GPIB INTERFACE FUNCTIONS

Interface Function	Symbol
Source Handshake	SH
Acceptor Handshake	AH
Talker or Extended Talker	T or TE
Listener or Extended Listener	L or LE
Service Request	SR
Remote-Local	RL
Parallel Poll	PP
Device Clear	DC
Device Trigger	DT
Controller	C

NOTE

The IEEE Std 488 standard defines the ten interface functions, the specific protocol, and timing relationships by the use of state diagrams. Not every instrument on the bus will have all ten interface functions incorporated, because only those functions important to a particular instrument's purpose need be implemented.

A TYPICAL GPIB SYSTEM

A typical GPIB instrumentation system is illustrated in Figure A-2, and it includes the nomenclature for the sixteen active signal lines. Only four instruments are shown connected directly to the control bus, but the GPIB can support up to fifteen instruments connected directly to the bus. However, more than fifteen devices can be interfaced to a single bus if they do not connect directly to the bus, but are interfaced through a primary device. Such a scheme can be used for programmable plug-ins housed in a mainframe where the mainframe is addressed with a primary address code and the plug-ins are addressed with a secondary address code.

To maintain the electrical characteristics of the bus, a device load should be connected for each two meters of cable length. Although instruments are usually spaced no more than two meters apart, they can be separated farther apart if the required number of device loads are lumped at any given point. For proper operation, at least two-thirds of the instruments connected directly to the bus must be in the power-on state.

TALKERS, LISTENERS, AND CONTROLLERS

A talker is an instrument that can send messages and data over the bus; a listener is an instrument that can accept messages and data from the bus. An instrument can be a talker only, listener only, or be both a talker and a listener. Unless a device is in the talk-only or listen-only mode, it can only communicate with other devices on the bus when it is enabled to do so by the controller in charge of the instrumentation system.

A controller is an instrument that determines, by software routines, which instrument will talk and which instruments will listen during any given time interval. The controller has the ability to assign itself as a talker or a listener whenever the program routine requires it. In addition to designating the current talker and listeners for a particular communication sequence, the controller is assigned the task of sending special codes and commands (called interface control messages) to any or all instruments on the bus. A complete operating system may contain more than one controller. The IEEE standard has provisions for a system controller that operates with another controller in charge of the bus. The controller that is in charge of the bus can take control only when it is directed to do so by the system controller. The system controller itself may be, but is not necessarily, the controller in charge of the bus.

INTERFACE CONTROL MESSAGES

The two types of interface control messages are multi-line messages sent over the data bus and uni-line messages.

A message that shares a group of signal lines with other messages, in some mutually exclusive set, is called a multi-line message (only one multi-line message (message byte) can be sent at one time).

A message sent over a single line is called a uni-line message (two or more of these messages can be sent concurrently.)

Only multi-line messages are discussed here; uni-line messages are discussed later under GPIB Signal Line Definitions.

The interface control messages (refer to Figure A-3) are sent and received over the data bus only with the ATN (attention) line asserted (true). Interface message coding can be related to the ISO (International Standards Organization) 7-bit code by relating data bus lines DIO1 through DIO7 to bits B1 through B7, respectively, in the Bits column in Figure A-3.

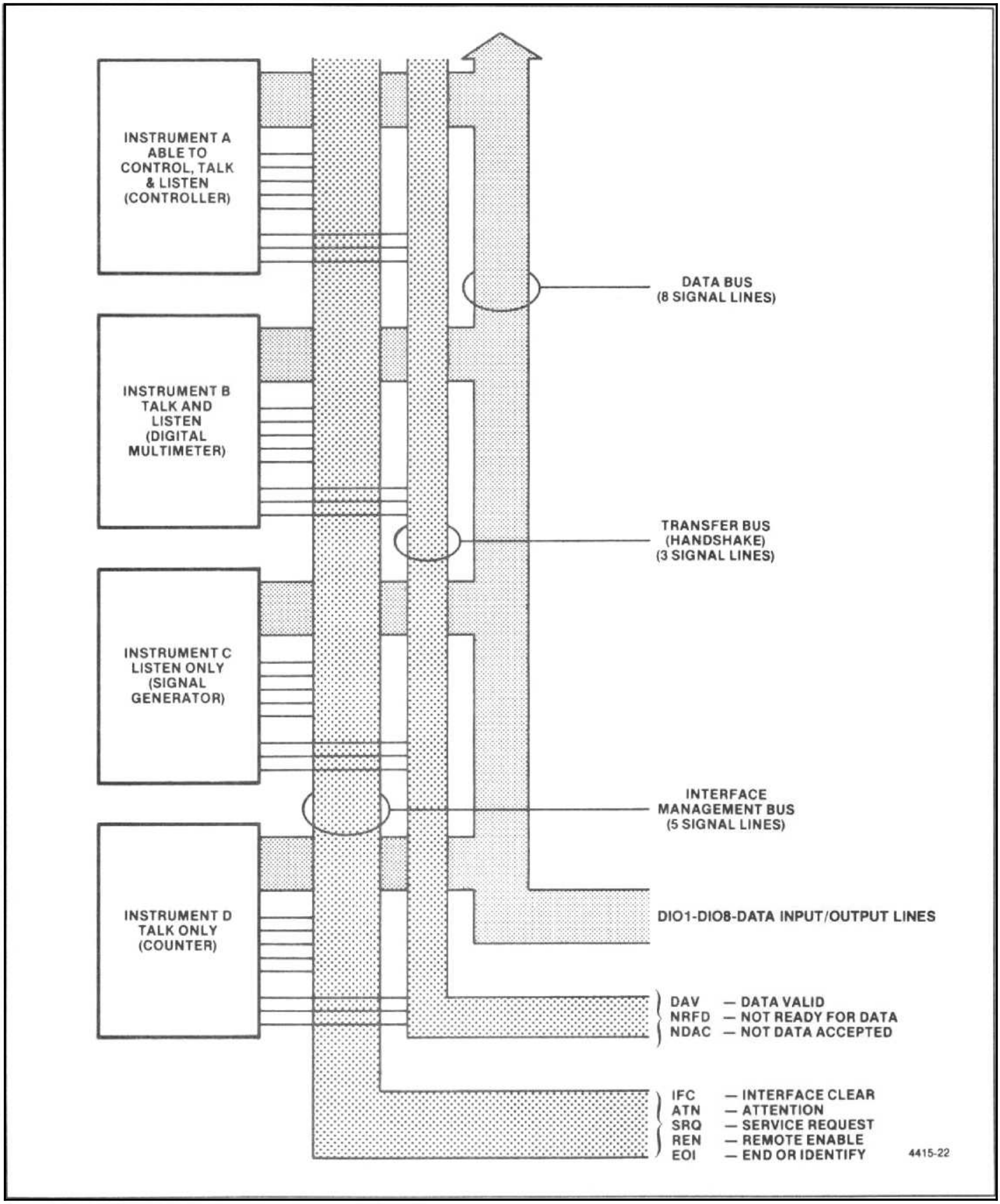


Figure A-2. A typical GPIB system.

Interface control messages (refer to Table A-2) include the primary talk and listen addresses for instruments on the bus, addressed commands (only instruments previously addressed to listen will respond to these commands), universal commands (all instruments, whether they have been addressed or not, will respond to these commands), and secondary addresses for devices interfaced through a primary instrument. Parallel Poll Enable (PPE) messages are derived from the characters in the first column under Lower Case letters in Figure A-3 (decimal coded characters 96 through 111). The standard recommends the use of decimal code 112 (lower case letter p) for the Parallel Poll Disable (PPD) command. All parallel poll configured instruments respond with status information at the same time when the EOI line is asserted with ATN true.

Table A-2

INTERFACE MESSAGES (REFERRED TO IN THIS APPENDIX) AND FUNCTIONS

Mnemonic	Message	Interface Function
	Remote Messages Received	
ATN	Attention	AH,C,L,LE,PP,SH,T,TE
DAC	Data Accepted	SH
DAV	Data Valid	AH
DCL[a]	Device Clear	DC
GET[a]	Group Execute Trigger	DT
GTL[a]	Go To Local	RL
IFC	Interface Clear	C,L,LE,T,TE
LLO[a]	Local Lockout	RL
MSA[a]	My Secondary Address	LE,TE
MTA[a]	My Talk Address	T,TE
PPC[a]	Parallel Poll Configure	PP
PPD[a]	Parallel Poll Disable	PP
PPE[a]	Parallel Poll Enable	PP
PPU[a]	Parallel Poll Unconfigure	PP
REN	Remote Enable	RL
RFD	Ready For Data	SH
SDC[a]	Selected Device Clear	DC
SPD[a]	Serial Poll Disable	T,TE
SPE[a]	Serial Poll Enable	T,TE
SRQ	Service Request	(via C)
TCT[a]	Take Control	C
UNL[a]	Unlisten	L,LE
	Remote Messages Sent	
ATN	Attention	C
DAC	Data Accepted	AH
DAV	Data Valid	SH
DCL[a]	Device Clear	(via C)
GET[a]	Group Execute Trigger	(via C)
GTL[a]	Go To Local	(via C)
IFC	Interface Clear	C
LLO[a]	Local Lockout	(via C)
MSA[a]	My Secondary Address	(via C)
MTA[a]	My Talk Address	(via C)
PPC[a]	Parallel Poll Configure	(via C)
PPD[a]	Parallel Poll Disable	(via C)
PPE[a]	Parallel Poll Enable	(via C)
PPU[a]	Parallel Poll Unconfigure	(via C)
REN	Remote Enable	C
RFD	Ready For Data	AH
SDC[a]	Selected Device Clear	(via C)
SPD[a]	Serial Poll Disable	(via C)
SPE[a]	Serial Poll Enable	(via C)
SRQ	Service Request	SR
TCT[a]	Take Control	(via C)
UNL[a]	Unlisten	(via C)
UNT[a]	Untalk	(via C)

[a] Multi-line messages.

DEVICE-DEPENDENT MESSAGES

The IEEE standard does not specify the coding of device-dependent messages (messages that control the internal operating functions of a device). After addressing a talker and the required number of listeners via interface control messages, the controller unasserts the ATN line (false) on the bus. When ATN becomes false (high), any commonly understood 8-bit binary code may be used to represent a device-dependent message.

However, the standard recommends that the alphanumeric codes associated with the numbers, symbols, and upper case characters (decimal 32 to decimal 94) in the ASCII Code Chart (Figure A-3) be used to compose device-dependent messages. One example of a device-dependent message could be the following ASCII character string:

MODE V;VOLTS 2.5E-3;FREQ 1.0E3

The ASCII character string, sent with the ATN line unasserted, tells the instrument to set its front panel controls to the voltage mode and output a 2.5 mV signal at a frequency of 1000 Hz.

When 8-bit binary codes other than the ISO 7-bit are used for device-dependent messages, the most significant bit should be on data line DI08 (for bit-8).

B7 B6 B5 BITS	Ø Ø Ø	Ø Ø 1	Ø 1 Ø	Ø 1 1	1 Ø Ø	1 Ø 1	1 1 Ø	1 1 1
B4 B3 B2 B1	CONTROL		NUMBERS SYMBOLS		UPPER CASE		LOWER CASE	
Ø Ø Ø Ø	0 NUL 0 0	20 DLE 10 16	40 0 SP 20 32	60 16 0 30 48	100 0 @ 40 64	120 16 P 50 80	140 0 ‵ 60 96	160 16 p 70 112
Ø Ø Ø 1	1 GTL SOH 1 1	21 LLO DC1 11 17	41 1 ! 21 33	61 17 1 31 49	101 1 A 41 65	121 17 Q 51 81	141 1 a 61 97	161 17 q 71 113
Ø Ø 1 Ø	2 STX 2 2	22 DC2 12 18	42 2 " 22 34	62 18 2 32 50	102 2 B 42 66	122 18 R 52 82	142 2 b 62 98	162 18 r 72 114
Ø Ø 1 1	3 ETX 3 3	23 DC3 13 19	43 3 # 23 35	63 19 3 33 51	103 3 C 43 67	123 19 S 53 83	143 3 c 63 99	163 19 s 73 115
Ø 1 Ø Ø	4 SDC EOT 4 4	24 DCL DC4 14 20	44 4 $ 24 36	64 20 4 34 52	104 4 D 44 68	124 20 T 54 84	144 4 d 64 100	164 20 t 74 116
Ø 1 Ø 1	5 PPC ENQ 5 5	25 PPU NAK 15 21	45 5 % 25 37	65 21 5 35 53	105 5 E 45 69	125 21 U 55 85	145 5 e 65 101	165 21 u 75 117
Ø 1 1 Ø	6 ACK 6 6	26 SYN 16 22	46 6 & 26 38	66 22 6 36 54	106 6 F 46 70	126 22 V 56 86	146 6 f 66 102	166 22 v 76 118
Ø 1 1 1	7 BEL 7 7	27 ETB 17 23	47 7 ’ 27 39	67 23 7 37 55	107 7 G 47 71	127 23 W 57 87	147 7 g 67 103	167 23 w 77 119
1 Ø Ø Ø	10 GET BS 8 8	30 SPE CAN 18 24	50 8 (28 40	70 24 8 38 56	110 8 H 48 72	130 24 X 58 88	150 8 h 68 104	170 24 x 78 120
1 Ø Ø 1	11 TCT HT 9 9	31 SPD EM 19 25	51 9) 29 41	71 25 9 39 57	111 9 I 49 73	131 25 Y 59 89	151 9 i 69 105	171 25 y 79 121
1 Ø 1 Ø	12 LF A 10	32 SUB 1A 26	52 10 * 2A 42	72 26 : 3A 58	112 10 J 4A 74	132 26 Z 5A 90	152 10 j 6A 106	172 26 z 7A 122
1 Ø 1 1	13 VT B 11	33 ESC 1B 27	53 11 + 2B 43	73 27 ; 3B 59	113 11 K 4B 75	133 27 [5B 91	153 11 k 6B 107	173 27 { 7B 123
1 1 Ø Ø	14 FF C 12	34 FS 1C 28	54 12 , 2C 44	74 28 < 3C 60	114 12 L 4C 76	134 28 \ 5C 92	154 12 l 6C 108	174 28 | 7C 124
1 1 Ø 1	15 CR D 13	35 GS 1D 29	55 13 – 2D 45	75 29 = 3D 61	115 13 M 4D 77	135 29] 5D 93	155 13 m 6D 109	175 29 } 7D 125
1 1 1 Ø	16 SO E 14	36 RS 1E 30	56 14 . 2E 46	76 30 > 3E 62	116 14 N 4E 78	136 30 ∧ 5E 94	156 14 n 6E 110	176 30 ~ 7E 126
1 1 1 1	17 SI F 15	37 US 1F 31	57 15 / 2F 47	77 UNL ? 3F 63	117 15 O 4F 79	137 UNT — 5F 95	157 15 o 6F 111	177 DEL (RUBOUT) 7F 127
	ADDRESSED COMMANDS	UNIVERSAL COMMANDS	LISTEN ADDRESSES		TALK ADDRESSES		SECONDARY ADDRESSES OR COMMANDS	

KEY

octal	25 PPU	GPIB code
	NAK	ASCII character
hex	15 21	decimal

REF: ANSI STD X3. 4-1977
IEEE STD 488-1978
ISO STD 646-1973

4415-23

Figure A-3. ASCII & GPIB Code Chart.

To summarize the difference between interface control messages and device-dependent messages on the data bus, remember that any message sent or received when the ATN line is asserted (low) is an interface control message. Any message (data bytes) sent or received when the ATN line is unasserted (high) is a device-dependent message.

GPIB SIGNAL LINE DEFINITIONS

Figure A-2 shows how the sixteen active signal lines on the GPIB are functionally divided into three component buses: an eight-line data bus, a three-line data byte transfer control (handshake) bus, and a five-line general interface management bus.

The data bus contains eight bi-directional signal lines, DI01 through DI08. Information, in the form of data bytes, is transferred over this bus. A handshake timing sequence between the enabled talker and the enabled listeners on the three-line data transfer control bus transfers one data byte (eight bits) at a time. These data bytes are sent and received in a byte-serial, bit-parallel fashion.

Since the handshake sequence is an asynchronous operation (no clock signal on the bus), the data transfer rate is only as fast as the slowest instrument involved in a data byte transfer. A talker cannot place data bytes on the bus faster than the slowest listener can accept them.

Figure A-4 illustrates the flow of data bytes on the bus when a typical controller sends ASCII data to an assigned listener. The first data byte (decimal 44) enables an instrument at address 12 as a primary listener. The second data byte (decimal 108) is optional; for example, enabling a plug-in device at secondary address 12 as the final destination of the data to follow. The data is the two ASCII characters A and B (decimal 65 and decimal 66). Note that the ATN line is asserted for the first two data bytes and unasserted for the device-dependent character to indicate the last data byte in the message.

The controller activates the ATN line again and sends the universal unlisten (UNL) and untalk (UNT) commands to clear the bus. Six handshake cycles on the data transfer control bus are required to send the six data bytes.

Transfer Bus (Handshake)

Each time a data byte is transferred over the data bus, an enabled talker and all enabled listeners execute a handshake sequence via signal lines DAV, NRFD, and NDAC (see Figure A-5 — the ATN line is shown to illustrate the controller's role in the process).

DAV (Data Valid). The DAV signal line is asserted by the talker after the talker places a data byte on the data bus. When asserted (low), DAV tells each assigned listener that a new data byte is on the bus. The talker is inhibited from asserting DAV as long as any listener holds the NRFD signal line asserted.

NRFD (Not Ready For Data). An asserted NRFD signal line indicates one or more of the assigned listeners are not ready to receive the next data byte from the talker. When all of the assigned listeners for a particular data byte transfer have released NRFD, the NRFD line becomes unasserted (high). When NRFD goes high, the RFD message (Ready For Data) tells the talker it may place the next data byte on the data bus.

NDAC (Not Data Accepted). Each assigned listener holds the NDAC signal line asserted until the listener accepts the data byte currently on the bus. When all assigned listeners have accepted the current data byte, the NDAC signal line becomes unasserted (high), telling the talker to remove the data byte from the bus. The DAC message (Data Accepted) tells the talker that all assigned listeners have accepted the current data byte.

NOTE

One handshake cycle transfers one data byte; then, the listeners reset the NRFD line high and the NDAC line low before the talker asserts DAV for the next data byte transfer. Both NRFD and NDAC high at the same time is an invalid state on the bus.

Management Bus

The management bus is a group of five signal lines that are used to control the operation of the IEEE Std 488 (GPIB) Digital Interface.

IFC (Interface Clear). The system controller is the only instrument on the bus allowed to assert IFC. IFC is asserted for greater-than 100 μs to place all instruments in a predetermined state. While IFC is being sent, only the DCL (Device Clear), LLO (Local Lockout), PPU (Parallel Poll Unconfigure), and REN (Remote Enable) interface messages (universal commands) will be recognized.

ATN (Attention). The controller in charge is the only instrument on the bus allowed to assert ATN. ATN is asserted when an instrument connected to the bus is being enabled as a talker or listener, or when sending other interface control messages. As long as the ATN line is asserted (low), only instrument address codes and interface control messages are sent over the bus. When the ATN line is unasserted, only those instruments enabled as a talker and listener can send and receive data over the bus.

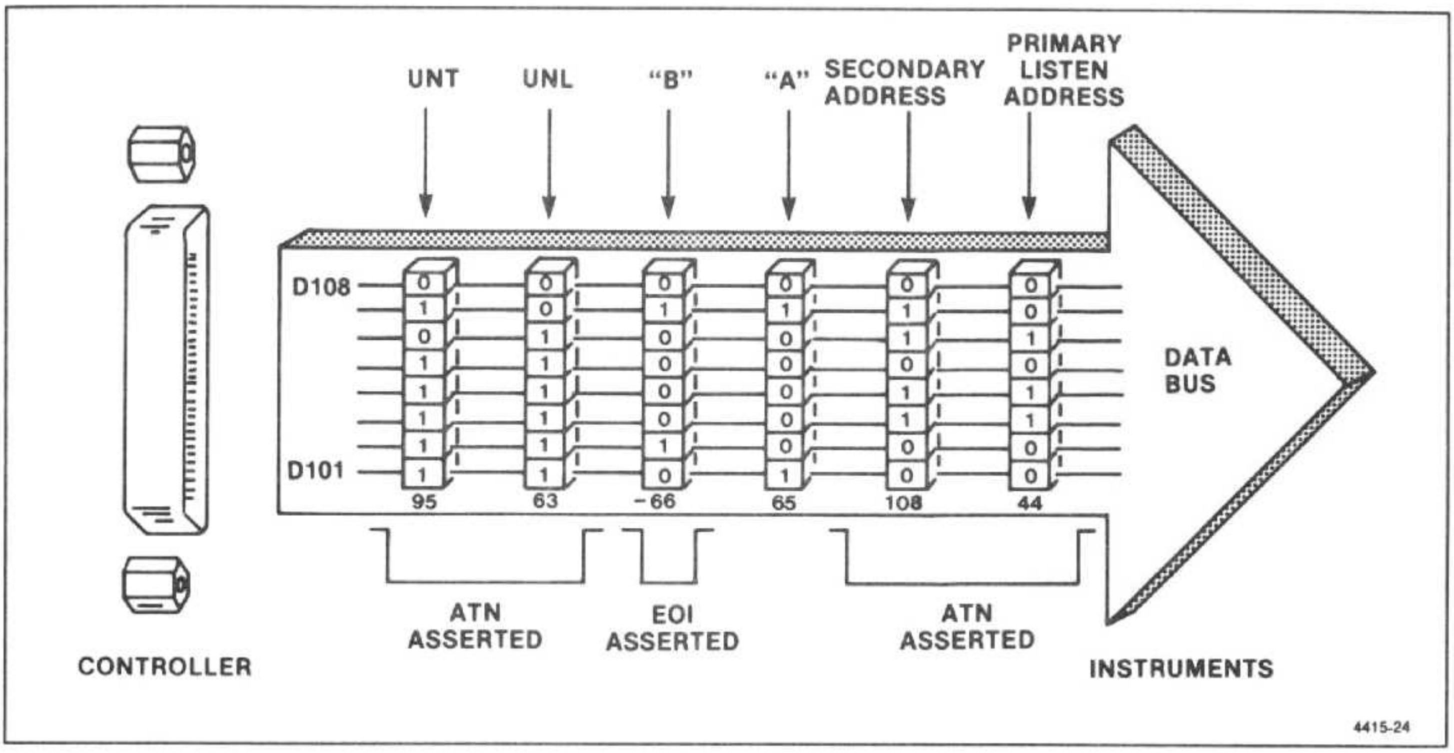


Figure A-4. An example of data byte traffic on the GPIB.

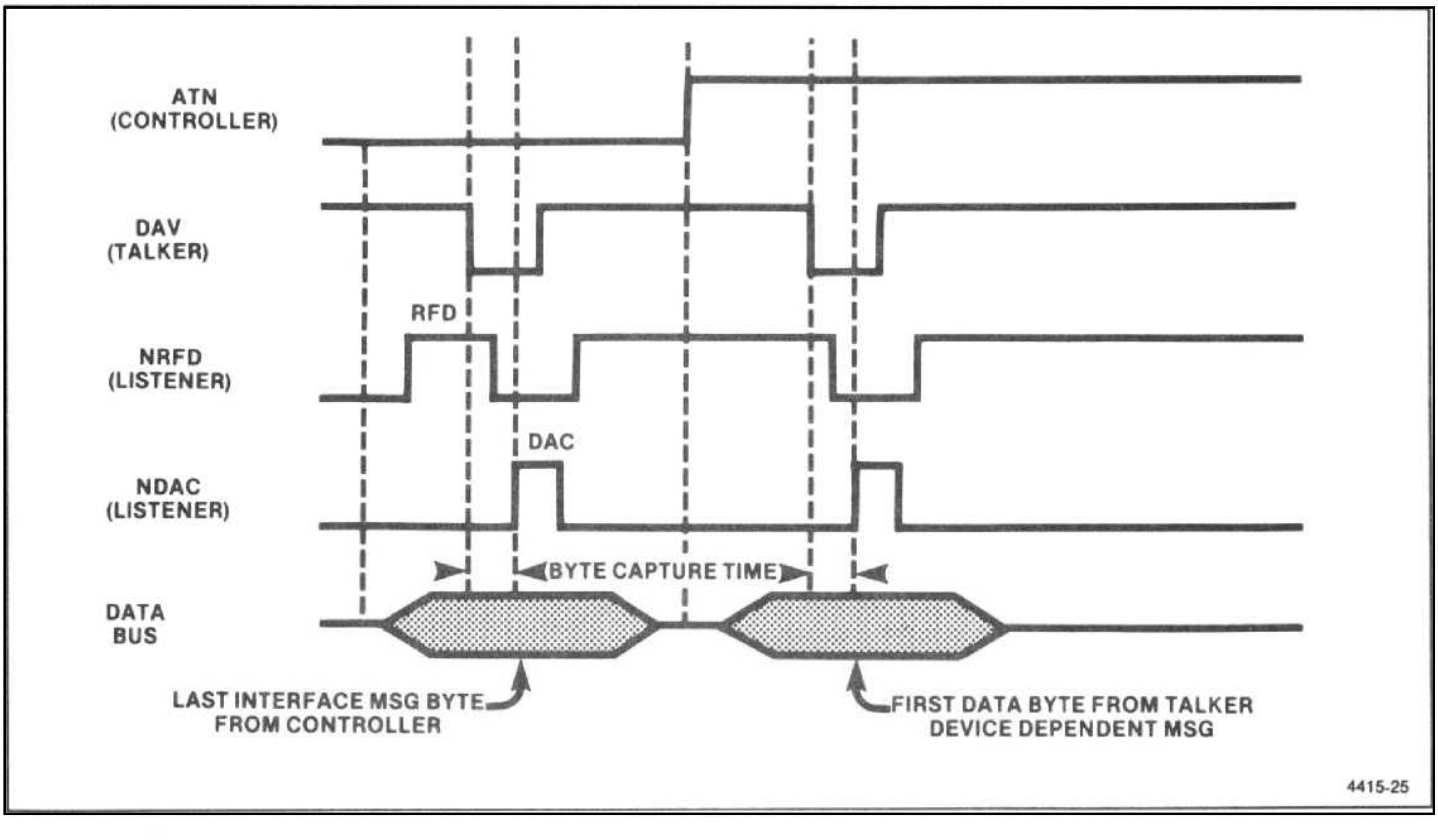


Figure A-5. A typical handshake timing sequence (idealized). Byte capture time is dependent on the slowest instrument involved in the handshake. RFD means Ready For DATA; DAC means Data Accepted.

SRQ (Service Request). Any instrument connected to the bus can request the controller's attention by asserting the SRQ line. The controller responds by asserting ATN and executing a serial poll routine to determine which instrument is requesting service. The instrument requesting service responds with a device-dependent status byte with bit seven asserted. When the instrument requesting service is found, program control is transferred to a service routine for that instrument. When the service routine is completed, program control returns to the main program. (The controller does not have to see the SRQ line asserted to perform a polling routine; it may do so whenever a program requires it).

REN (Remote Enable). The system controller asserts the REN signal line whenever the interface system operates under remote program control. Used with other interface control messages, such as LLO (Local Lockout) or GTL (Go To Local), the REN signal causes an instrument on the bus to select between two alternate sources of programming data. A remote-local interface function indicates to an instrument that the instrument will use either information input from the interface (remote) or to information input by the operator via the front panel controls (local).

EOI (End Or Identify). A talker can use the EOI signal line to indicate the end of a data transfer sequence. The talker asserts EOI as the last byte of data is transmitted. In this case, the EOI line is essentially a ninth data bit and must observe the same settling time as the data on the data bus. When an instrument controller is listening, it assumes that a data byte sent with EOI asserted is the last data byte in the complete message. When the instrument controller is talking, it may assert the EOI signal line as the last data byte is transferred. The EOI line is also asserted with the ATN line true if the controller conducts a parallel polling sequence on the bus. The EOI line is not used for a serial polling sequence.

INTERFACE FUNCTIONS AND MESSAGES

The ten major interface functions listed in Table A-1 provide a variety of capabilities and options for an instrumentation system. These functions may be implemented in, or for, any particular instrument with instrument hardware or with a programming routine (software).

Only those functions necessary for an instrument's purpose need be implemented by the instrument's designer; it is not likely that one single instrument will have all ten interface functions. For example, an instrument generally doesn't need to implement the Parallel Poll (PP) function if the instrument can respond to a serial polling sequence from the controller in charge of the GPIB system.

The following discusses the interface functions and their relationship to the interface control messages shown in Figure A-3. All the interface control messages discussed are sent and received over the GPIB with the ATN line asserted (low).

RL (Remote-Local Function)

The RL function provides an instrument with the capability to select between two sources of input information. This function indicates to the instrument that its internal device-dependent functions are to respond to information input from the front panel (Local) or to corresponding programming information from the GPIB (Remote). Only the system controller is permitted to assert the REN (Remote Enable) line, whether or not it is the controller in charge at the time.

When the system controller asserts the REN line, an instrument on the GPIB goes to a remote mode when it is addressed as a listener with its listen address, not before. An instrument remains in a remote mode until the REN line is released (high), or an optional front-panel switch on the instrument is activated to request the local mode, or a GTL (Go To Local) command is received while the instrument is enabled as a listener.

However, the controller can disable the instrument's front-panel "return to local" switch(es) by sending a LLO (Local Lockout) command. The LLO command must be preceded or followed by a listen address (MLA) to cause the instrument to go to a remote mode with front-panel lockout. The UNL (Unlisten) command does not return an instrument to the local mode.

When the REN line goes false, it must be recognized by all instruments on the bus and they must go to the local mode within 100 μs. If data bytes are still being placed on the bus when REN goes false, the system program should assure that the data bytes are sent and received with the knowledge that the system is in a local mode, not remote.

T/TE and L/LE (Talker and Listener Functions)

NOTE

> Although discussed under one heading, the T/TE and L/LE functions are independent of each other.

The T (Talker) and TE (Talker Extended) functions provide an instrument and its secondary devices, if any, with the capability to send device-dependent data over the GPIB (or, in case of a controller, the capability to send device-dependent program data) over the GPIB.

The Talker (T) function is a normal function for a talker and uses only a one-byte primary address code called MTA (My Talk Address). The Talker Extended (TE) function requires a two-byte address code; an MTA code followed by the second byte called MSA (My Secondary Address).

Only one instrument in the GPIB system can be in the active talker state at any given time. A non-controller commences talking when ATN is released and continues its talker status until an Interface Clear (IFC) message occurs or an Untalk (UNT) command is received from the controller in charge. The instrument will stop talking and listen any time that the controller in charge asserts ATN.

One or more instruments on the bus can be programmed for the L (Listener) function by use of their specific primary listen address (called MLA). Some of the instruments interfaced to the bus may be programmed for the LE (Listener Extended) function, if implemented. The LE function requires a two-byte address code. No L or LE function is active during the time that ATN is asserted.

All talker and listener functions must respond to ATN within 200 ns. They must also respond to IFC in less than 100 µs.

An instrument may be a talker only, a listener only, or implement all functions. In any case, its address code has the form X10TTTTT for a talker and X01LLLLL for a listener. For instruments with both T and L functions, the T-bit binary values are usually equal to the binary value of the L bits. Before applying power to the system, the system operator sets these five least significant bits by means of an address switch on each instrument. The controller's address code may be implemented in software.

The system program, run from the controller, designates the primary talker and primary listener status of the desired instruments by coding data bits 6 and 7; 1, 0, respectively, for a talker and 0, 1, respectively, for a listener. Secondary talk and listen addresses (or commands) are represented by the controller sending both data bits (6 and 7) as a logical 1. The controller may listen to bus traffic without actually addressing itself over the bus.

SH and AH (Source and Acceptor Handshake Functions)

NOTE

Although discussed under one heading, the SH and AH functions are independent of each other.

The SH (Source Handshake) function guarantees proper transmission of data, while the AH (Acceptor Handshake) function guarantees proper reception of data. The interlocked handshake sequence between these two functions guarantees asynchronous transfer of each data byte. The handshake sequence is performed via the NRFD, DAV, and NDAC signal lines on the bus (see Figure A-5). Both functions must respond to ATN within 200 n.

The SH function must wait for the RFD (Ready For Data) message plus wait at least 2 µs before asserting DAV; this allows the data to settle on the data bus. If three-state drivers are used, the settling time is reduced to RFD plus 1.1 µs. Faster settling times are allowed under special conditions and warning notes in the standard. The time it takes for the AH function to accept an interface message byte is dependent on how the designer implemented the function.

DC (Device Clear Function)

The DCL (Device Clear) function allows the controller in charge to "clear" any or all instruments on the bus. The controller (under program direction) asserts ATN and sends either the universal DCL (Device Clear) command or the SDC (Selected Device Clear) command.

When the DCL message is received, all instruments on the bus must clear or initialize their internal device functions. When the controller sends the SDC command, only those instruments that have been previously addressed to listen must respond. The IEEE 488 standard does not specify the settings an instrument must go to as a result of receiving the DCL or SDC command. (In general, these commands are used only to clear the GPIB interface circuits within an instrument.)

DT (Device Trigger Function)

The DT (Device Trigger) function allows the controller in charge to start the basic operation specified for an instrument or group of instruments on the bus. The IEEE 488 standard does not specify the basic operation an instrument is to perform when it receives the GET (Group Execute Trigger) command. To issue the GET command, the controller asserts ATN, sends the listen addresses of the instruments that are to respond to the trigger, and then sends the GET message.

Once an instrument starts its basic operation in response to GET, the instrument must not respond to subsequent trigger-state transitions until the current operation is complete. Only after completing the operation can the instrument repeat its basic operation in response to the next GET message. Thus, the basic operating time is the major factor that determines how fast the instrument(s) can be repeatedly "triggered" by commands from the bus.

C, SR, and PP (Controller, Service Request, and Parallel Poll Functions)

The C (Controller) function provides the capability to send primary talk and listen addresses, secondary addresses, universal commands, and addressed commands to all instruments on the bus. The Controller function also provides the capability to respond to a service request message (SRQ) from an instrument or to conduct a parallel poll routine to determine the status of any or all instruments on the bus that have the Parallel Poll (PP) function implemented.

If an instrumentation system has more than one controller, only the system controller is allowed to assert the IFC (Interface Clear) and REN (Remote Enable) lines at any time during system operation, whether or not it is the controller in charge at the time.

If a controller requests system control from another controller and it receives a message from another controller to send REN, the system controller must verify that the REN line remains unasserted (false) for at least 100 μs before asserting REN. The time interval that REN is asserted depends on the remote programming sequence and will vary with the program. The IFC line must be asserted for at least 100 μs.

The Controller function has specified time intervals for certain operations. For example, the execution time for parallel polling instruments on the bus cannot be less than 2 μs. If the controller is in the controller active wait state and does not receive an internal message to conduct a parallel poll, it must wait for at least 1.5 μs before going to the controller active state in order to give the NRFD, NDAC, and EOI lines sufficient time to assume their valid states.

The controller must also have a delay of at least 2 μs (1.1 μs for tri-state drivers) in order for the instruments to see the ATN line asserted before the controller places the first data byte on the bus.

Taking Control (Asynchronous or Synchronous)

All data bytes transmitted over the GPIB with the ATN line asserted are interpreted as system control information. Asserting ATN directly at any moment is an asynchronous operation with respect to the bus and may cause loss of data if a handshake cycle is in progress. To prevent loss of data, a controller can take control synchronously, that is, it can monitor the Transfer Bus and only assert ATN when DAV is unasserted (false).

Passing Control

As a controller in charge, the system controller (program) may relinquish control to any other instrument in the system capable of acting as a controller. The controller in charge first addresses the other controller as a talker and then sends the TCT (Take Control) command. The other controller then becomes the controller in charge when ATN is released.

Performing a Serial Poll

The controller-in-charge may conduct a serial poll at any time, whether or not an instrument on the bus has asserted the SRQ line. (Most, but not all, instruments have the Service Request (SR) function.)

To perform a serial poll, the controller first asserts ATN and issues the Untalk (UNT) and Unlisten (UNL) commands. The controller then sends the Serial Poll Enable (SPE) command, followed by the talk address of the first instrument to be polled. The controller then releases ATN and the addressed talker responds by sending its status byte over the bus. If the addressed talker has requested service, it must assert bit seven of the status byte and encode the remaining seven bits to indicate the reason for asserting SRQ. Status bytes are device-dependent and are not specified in the IEEE 488 standard. An addressed instrument will release its SRQ line when serial polled, but other instruments may still hold it asserted. When the controller has read the status byte of an addressed instrument, it reasserts ATN and addresses the next instrument to talk, then releases ATN and receives the instrument's status byte. The routine continues until the controller no longer detects the SRQ line asserted. At this time, the controller should send the Serial Poll Disable (SPD) message and, optionally, send the UNT message to release the last active talker.

Performing A Parallel Poll

The Parallel Poll (PP) function provides an instrument with the capability to present one, and only one, bit of status information to the controller without being previously addressed to talk. The parallel polling capability requires a commitment by the system program to periodically conduct a parallel poll sequence.

When an instrument responds to a parallel poll, the single data bit presented to the controller may or may not indicate a need for service. If the data bit is used as a service request indication, the controller should perform a serial poll in order to obtain a complete status byte with more information (if the device has the SR function implemented).

Before an instrument can respond to a parallel poll, the GPIB system must first be configured. In a typical sequence, the controller first sends an UNL command to clear the bus of listeners, then the listen address of the device to be configured. Following this, the controller sends the PPC (Parallel Poll Configure) command followed by a PPE (Parallel Poll Enable) message. The PPE message contains coded information that tells the selected instrument which data line will carry the PP status bit for that device. This entire sequence is repeated for each instrument to be configured.

The PPE message(s) sent by the controller has the form X110SPPP. Bit 4 (S) is called the sense bit and the three least significant bits (PPP) represent an octal number (0 through 7) that corresponds to a specific line on the data bus that an instrument must assert if its internal status has the same value as the sense bit (S may equal 1 or 0).

The actual parallel poll takes place after each instrument has been completely configured. The concept is to have the controller receive one data byte that contains status information on all of the addressed instruments. To receive this status byte, the controller asserts the EOI line and the ATN line. The assertion of EOI may be coincident with ATN or later, so long as both are asserted. This may occur any time after the last PPE message. The controller then reads the data bus lines while ATN and EOI are asserted to interpret the status of all selected instruments.

To conclude the parallel poll, the controller releases EOI and then ATN. The instrument(s) do not need to be reconfigured for each subsequent parallel poll. The PPU (Parallel Poll Unconfigure) command will clear all device configurations and prevent them from responding to future polls. The PPD (Parallel Poll Disable) command accomplishes essentially the same thing, except that the PP function remains in the "configured" state. PPU is a universal command (all instruments) while PPD is used with PPC and becomes an addressed command (only those devices selected with PPC will accept PPD.)

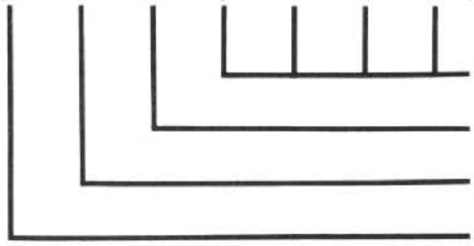
NOTES

- To assure correct and expected results during signal processing, use the Single Sweep triggering mode while programming. The signal processing commands are HRAMPL, LRAMPL, MFBIG, MLFTNX, PKFIND, MMAX, MMIN, MRGTNX, MVLFDB, MVRTDB, CURVE, WAVFRM, DPRE, DCOPY, FIBIG, LFTNXT, RGTNXT, FMAX, and FMIN.
- Only the first three letters of a mnemonic are required (e.g., ARE for AREs); except, all of the letters of the units arguments for REFLVL and MAXPWR must be entered.
- Form a query by adding a question mark to the header of a Display Data I/O, Marker System, Macro, or Front-Panel header (AREs?) unless no query is indicated. POInt? is the only Waveform Processing query.
- The header will be returned with query responses with HDR ON, or the header will not be returned with HDR OFF.
- NUM is a decimal number; integer, floating point, or scientific notation.
- NUM may be substituted for ON or OFF; 1=ON, 0=OFF.
- NUM may be followed by units in engineering notation for frequency, time, and amplitude; 100 MHz, 10 μs, −60 dBm.
- Be sure to keep copies of all macros so they can be more easily recreated in case the memory is interrupted (as in the case of removing the battery for long-term storage).
- Keep an accurate account of the memory used in each macro; use the MEMORY query to find out how much memory a particular macro uses, or to find out how much memory is left for additional macros.
- Queries are acted upon as soon as they are received and cannot be used in macros.
- Once a macro is entered into memory, it cannot be changed. The macro must be removed by number (KILL NUM) and completely re-entered.

PROGRAMMING SUMMARY

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)

5559-15

Q, STSTOP, STEP)

INTERFACE MESSAGES

DCL (20) — Clears OPI/O buffer and status byte
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

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
		Execution Errors
26	250	Output buffer overflow; remaining output lost
27	201	Attempt to execute command in Local mode
28[a]	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[a]	205	SPAN out of range
32[a]	205	RESBW out of range
33[a]	205	MAXPWR or MINATT out of range
34[a]	205	Level out of range (REFLVL, THRHLD, BWNUM, MVRTDB, MVLFDB)
35[a]	205	VRTDSP LIN out of range
36[a]	205	VRTDSP LOG out of range
37[a]	205	TIME out of range
39[a]	204	IDENTify not allowed in this span/div
40[a]	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[a]	205	NUMEV out of range
47[a]	205	STORE, RECALL, DSTORE, DRECAL, or RDATA out of range
48	204	PHSLK cannot be turned OFF/ON directly with PHSLK command
101[a]	204	Function not available when SGTRAK is on
102[a]	205	Frequency range limited in 75 Ω input — Option 07 only
103[a]	205	Frequency out of range after step
104[a]	204	Bandwidth mode is not available when in linear
105[a]	205	Illegal sweep range
106[a]	205	Argument out of range
109[a]	205	ROFSET out of range
		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

[a] If this error message is issued, the macro in progress will be aborted.

ERR? RESPONSES
In Numerical Order For Quick Reference

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[a]	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[a]	SPAN out of range
32[a]	RESBW out of range
33[a]	MAXPWR or MINATT out of range
34[a]	Level out of range (REFLVL, THRHLD, BWNUM,MVRTDB, MVLFDB)
35[a]	VRTDSP LIN out of range
36[a]	VRTDSP LOG out of range
37[a]	TIME out of range
39[a]	IDENTify not allowed in this span/div
40[a]	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[a]	NUMEV out of range
47[a]	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
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

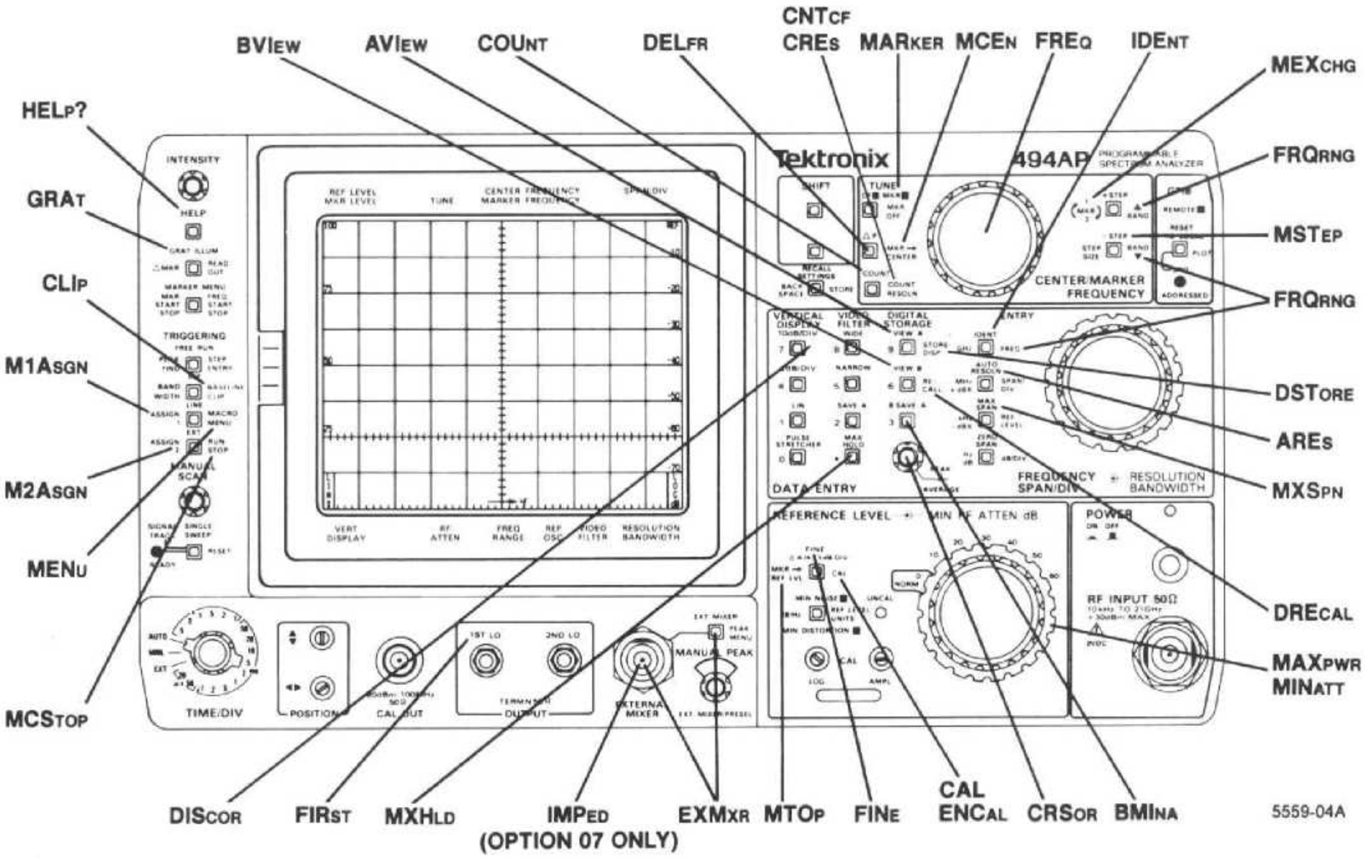
[a] If this error message is issued, the macro in progress will be aborted.

ERR? RESPONSES (Continued)
In Numerical Order For Quick Reference

Error Code	Meaning
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
99	Unrecognized event occurred
97	Power just came on
98	Operation complete (end of sweep)
100	SRQ was requested
101[a]	Function not available when SGTRAK is on
102[a]	Frequency range limited in 75 Ω input — Option 07 only
103[a]	Frequency out of range after step
104[a]	Bandwidth mode is not available when in linear
105[a]	Illegal sweep range
106[a]	Argument out of range
107[a]	MKTIME not available unless in zero span mode
108[a]	MKTIME out of range
109[a]	ROFSET out of range
110	STEP size out of range — set to maximum
111	SPAN defaulted to minimum span
120[a]	Frequency out of range because marker is on inactive trace
121[a]	Function not available when marker is on an inactive trace
122[a]	Function not available when marker is off
123[a]	Function not available when delta marker is off
124[a]	Function not available in BWMODE
125[a]	Function not available when marker is on a B−SAVEA trace
126[a]	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
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[a]	No more MDATA commands to read
167[a]	Too many GOSUBs with no RETURN
168[a]	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[a]	Index out of range (DISBUF:INDEX VAR:INDEX, or STNUM INDEX)
177	FOR index is already used
178	Missing DONE or RETURN or GOTO

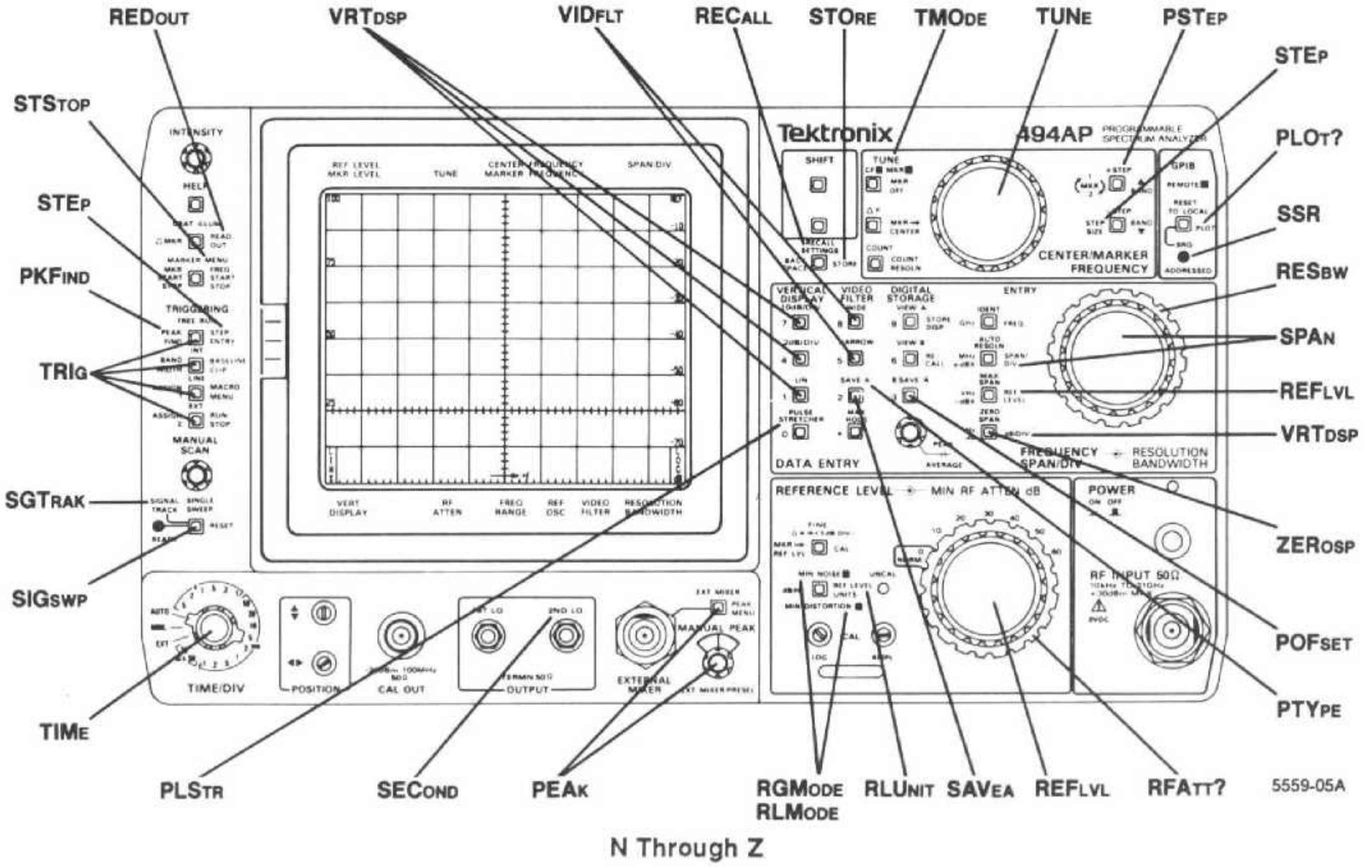
[a] If this error message is issued, the macro in progress will be aborted.

FRONT-PANEL RELATIONSHIPS TO MNEMONICS



5559-04A

A Through M



5559-05A

N Through Z

QUICK REFERENCE TO COMMANDS AND QUERIES

Mnemonic	Argument[a]	Query Available	Macro Memory Used	Page
ALLEV?[b]	None	Yes	N/A	9-10
ARES	On/Off	Yes	2 bytes	4-14
AVIEW	On/Off	Yes	2 bytes	4-27
BMINA	On/Off	Yes	2 bytes	4-28
BVIEW	On/Off	Yes	2 bytes	4-28
BWMODE	On/Off	Yes	2 bytes	5-11
BWNUM	Number	Yes	3 bytes	5-10
CAL[b]	Special	Yes	N/A	4-18
CENSIG	None	No	1 byte	8-3
CLEAR	Number	No	2 bytes	6-9
CLIP	On/Off	Yes	2 bytes	4-32
CNTCF	None	No	1 byte	4-9
COUNT	On/Off	Yes	2 bytes	4-8
CRES	Number	Yes	4 bytes	4-8
CRSOR	Special	Yes	2 bytes	4-29
CURVE[b]	Special	Yes	N/A	7-4
DCOPY?[b]	None	Yes	N/A	7-6
DEGAUS	None	No	1 byte	4-10
DELFR	On/Off	Yes	2 bytes	4-9
DISCOR	On/Off	Yes	2 bytes	4-6
DIVIDE	None	No	1 byte	6-2
DONE[c]	None	No	1 byte	6-13
DPMK	None	No	1 byte	5-5
DPRE?[b]	None	Yes	N/A	7-5
DRECAL	Special	No	3 bytes	4-29
DSLINE	Special	Yes	4 bytes	6-9
DSTORE	Special	No	3 bytes	4-28
DT	On/Off	Yes	2 bytes	9-10
ECR	On/Off	Yes	2 bytes	4-36
ELSE[c]	None	No	4 bytes	6-7
EMAC[c]	None	No	0 bytes	6-14
ENCAL	On/Off	Yes	2 bytes	4-19
ENDI[c]	None	No	1 byte	6-7
ENTER	Special	No	10 bytes	6-4
EOS	On/Off	Yes	2 bytes	9-7
ERCNT?[b]	None	Yes	N/A	9-11
ERR?[b]	None	Yes	N/A	9-11
EVENT?[b]	None	Yes	N/A	9-10
EVQTY?[b]	None	Yes	N/A	9-11
EXCHG	None	No	1 byte	6-3
EXMXR	On/Off	Yes	2 bytes	4-10
FIBIG	Number	No	2 bytes	8-2
FINE	On/Off	Yes	2 bytes	4-20
FIRST	Number	Yes	6 bytes	4-5

[a] Arguments are divided into the following four categories

- None — No arguments are required
- On/Off — The arguments ON (or num 1) or OFF (or num 2) are required
- Number — Number(s) argument required (terminator optional)
- Special — Specific command arguments, character strings, or any combination of these categories are required

[b] This command cannot be used within a macro.

[c] This command can only be used within a macro; it cannot be used outside a macro.

QUICK REFERENCE TO COMMANDS AND QUERIES (Continued)

Mnemonic	Argument[a]	Query Available	Macro Memory Used	Page
RLUNIT	Special	Yes	2 bytes	4-17
ROFSET	On/Off	Yes	3 bytes	4-18
RQS	On/Off	Yes	2 bytes	9-7
RUN[b]	Number	No	N/A	6-13
RVALID?[b]	None	Yes	N/A	4-36
SAMODE	On/Off	Yes	2 bytes	4-5
SAVEA	On/Off	Yes	2 bytes	4-28
SECOND	Number	Yes	6 bytes	4-5
SET?[b]	None	Yes	N/A	9-1
SGERR	On/Off	Yes	2 bytes	5-19
SGTRAK	Special	Yes	2 bytes	5-5
SIGSWP	None	Yes	1 byte	4-25
SPAN	Special	Yes	5 bytes	4-12
SSR	Number	No	2 bytes	4-36
STATUS BYTE[a]	N/A	N/A	N/A	9-8
STEP	Special	Yes	6 bytes	4-7
STMAC[b]	Special	No	N/A	6-16
STNUM	Number	No	2 bytes	6-15
STORE	Number	No	2 bytes	4-34
STSTOP	Special	No	12 byte	4-9
STYPE	Special	Yes	2 bytes	5-15
SUBT	None	No	1 byte	6-1
SWEEP	None	No	1 byte	6-15
TEST?[b]	None	Yes	N/A	9-11
TEXT	Special	Yes	2 bytes	7-7
TGMODE	On/Off	Yes	2 bytes	4-5
THRHLD	Special	Yes	4 bytes	5-14
TIME	Special	Yes	5 bytes	4-25
TMODE	Special	Yes	2 bytes	4-4
TOPSIG	None	No	1 byte	8-3
TRIG	Special	Yes	2 bytes	4-24
TUNE	Number	No	6 bytes	4-4
UPRDO?[b]	None	Yes	N/A	7-7
VAR?[b]	None	Yes	N/A	6-16
VIDFLT	Special	Yes	2 bytes	4-23
VRTDSP	Special	Yes	3 bytes	4-16
WAIT	None	No	1 byte	9-6
WARMSG	On/Off	Yes	2 bytes	9-8
WAVFRM?[b]	None	Yes	N/A	7-5
WFMPRE[b]	Special	Yes	N/A	7-1
ZEROSP	On/Off	Yes	2 bytes	4-12
ZETIME	On/Off	Yes	2 bytes	5-19
ZOOM	Number	No	2 bytes	5-19

[a] Arguments are divided into the following four categories

- None – No arguments are required
- On/Off – The arguments ON (or num 1) or OFF (or num 2) are required
- Number – Number(s) argument required (terminator optional)
- Special – Specific command arguments, character strings, or any combination of these categories are required

[b] This command cannot be used within a macro.

[c] This command can only be used within a macro; it cannot be used outside a macro.

HARMONIC TEST MACRO

MACRO PREPARATION

The folowing program shows the 4041 Data statements necessary for the HARMONIC TEST program. This macro will use less than 10% of the total available 8K bytes of memory set aside for macro storage.

```
 80  Z=1 ! ADDRESS OF SPECTRUM ANALYZER
100  Print #z:STMAC 1,"HARMONIC TEST"
110  Print #z:MARKER SINGLE
120  Print #z:RDOUT NORMAL;CLEAR
130  Print #z:PRINT 1,1,"HARMONIC TEST"
140  Print #z:SWEEP;MFBIG
150  Print #z:IF SIGNAL
160    Print #z:MCEN
170    Print #z:PRINT 3,1,"FUNDAMENTAL
      FREQUENCY,              AMPLITUDE"
180    Print #z:ENTER MFREQ;STNUM 3
190    Print #z:PRINT 4,1,XREG,FFORMT:18,0
200    Print #z:ENTER MAMPL;STNUM 2
210    Print #z:PRINT 4,28,XREG,DEC:6,1
220    Print #z:PRINT 5,1,"----------------------------------------"
230    Print #z:PRINT 6,1,"HARM #    LEVEL     DBC"
240    Print #z:STEP MARKER
250    Print #z:FOR 1,1,10
260      Print #z:ENTER 6;ENTER
       VAR:1;PLUS;STNUM 5
270      Print #z:PRINT VAR:5,2,VAR:1,DEC:2,0
280      Print #z:PSTEP;SWEEP;MFBIG
290      Print #z:IF SIGNAL
300        Print #z:ENTER MAMPL
310        Print #z:PRINT VAR:5,15,XREG,DEC:6,1,"DBM"
320        Print #z:ENTER VAR:2
330        Print #z:SUBT
340        Print #z:PRINT VAR:5,33,XREG,DEC:6,1
350      Print #z:ELSE
360        Print #z:PRINT VAR:5,6,"NO SIGNAL:
370      Print #z:ENDI
380    Print #z:NEXT
390    Print #z:TEXT MACRO
400    Print #z:FREQ VAR:3
410  Print #z:ELSE
420  Print #z:PRINT 4,2,"NO SIGNAL FOUND"
430  Print #z:PRINT 6,10,"MACRO KILLED"
440  Print #z:ENDI
450  Print #z:DONE
460  Print #z:EMAC
```

HARMONIC TEST MACRO (Continued)

Line 100 — Store the following macro in menu position 1 and title it HARMONIC TEST.

Line 110 — Turn on a single marker.

Line 120 — Turn on the normal readout, and clear the macro readout buffer.

Line 130 — Print HARMONIC TEST on line 1 beginning 1 space from the left.

Line 140 — Take a sweep, then move the marker to the signal peak.

Line 150 — Do a harmonic test.

Line 160 — Center the marker.

Line 170 — Print the fundamental frequency and amplitude on line 3 beginning 1 space from the left.

Line 180 — Enter the current marker frequency into the X register.

Line 190 — Print the following on the fourth line of the screen beginning 1 space from the left—18 characters of the contents of the X register in frequency format to 0 resolution.

Line 200 — Enter the current marker amplitude into the X register, and store this information into variable number 2.

Line 210 — Print the following on the fourth line of the screen beginning 28 spaces from the left—6 characters of the contents of the X register in decimal format to 1 digit.

Line 220 — Print a separating line on the fifth line of the screen beginning 1 space from the left.

Line 230 — Print the current harmonic information on the sixth line of the screen beginning 1 space from the left.

Line 240 — Put the marker frequency into step size.

Line 250 — For variable(1)=1 to 10.

Line 260 — Enter 6 into the X register, enter the contents of variable 1 into the X register (copying the 6 to the Y register), add 6 to the loop counter, and store the display line number.

Line 270 — Print the contents of variable 1 on the fifth line of the screen beginning 2 spaces from the left. This information will be 2 characters printed in decimal format to 0 digits.

Line 280 — Set plus step, do a sweep,and put the marker on the highest signal.

Line 290 — If a signal is present, display the level.

Line 300 — Enter the marker amplitude.

Line 310 — Print on the line held in variable 5 (line 7) indented 15 spaces from the left. The information to be printed will be 6 characters of the contents of the X register in decimal format to 1 digit, followed by DBM units designator.

Line 320 — Enter the contents of variable 2 into the X register.

Line 330 — Subtract the contents of the Y register (harmonic amplitude) from the contents of the X register (fundamental amplitude). The result is put in the X register.

Line 340 — Again print on the line held in variable 5 (line 7), but indented 33 spaces from the left. The information to be printed will be 6 characters of the contents of the X register in decimal format to 1 digit.

Line 360 — Print NO SIGNAL on the line held in variable 5, indented 6 spaces.

Line 380 — Go to the next harmonic number.

Line 390 — Display the macro readout buffer.

Line 400 — Tune back to the fundamental frequency.

Line 420 — Print NO SIGNAL FOUND on the fourth line of the screen indented 2 spaces.

Line 430 — Print MACRO KILLED on the sixth line of the screen indented 10 spaces.

Line 450 — The macro is done running.

Line 460 — This is the end of the macro entry.

HARMONIC TEST MACRO (Continued)

```
STMAC 1,"HARMONIC TEST"                                     # Store as macro number 1 and
                                                            # title it HARMONIC TEST
MARKER SINGLE                                               # Turn on a single marker
RDOUT NORMAL                                                # Readout normal
CLEAR                                                       # Clear macro readout buffer
SWEEP                                                       # Do a sweep
MFBIG                                                       # Put marker on highest signal
IF SIGNAL                                                   # Do harmonic test if signal was
                                                            # found
    PRINT 1,1,"HARMONIC TEST"                               # Print title
    PRINT 3,1,"FUNDAMENTAL FREQUENCY,   AMPLITUDE"
    ENTER MFREQ                                             # Enter primary frequency into
                                                            # register
    STNUM 3                                                 # Store fundamental frequency
    PRINT 4,1,XREG,FFORMT:18,0                              # Display fundamental frequency
    ENTER MAMPL                                             # Enter primary amplitude into
                                                            # register
    STNUM 2                                                 # Store fundamental amplitude
    PRINT 4,28,XREG,DEC:6.1                                 # Display fundamental amplitude
    PRINT 5,1,"------------------------------------------"
    PRINT 6,1,"HARM #          LEVEL            DBC"
    STEP PRIMAR                                             # Put marker frequency into
                                                            # step size
    FOR 1,1,10                                              # For var(1)=1 to 10
        ENTER 6
        ENTER VAR:1
        PLUS                                                # Add 6 to loop counter
        STNUM 5                                             # Store display line number
        PRINT VAR:5,2,VAR:1,DEC:2,0                         # Display harmonic number
        PSTEP                                               # Plus step
        SWEEP                                               # Do a sweep
        MFBIG                                               # Put marker on highest signal
        IF SIGNAL                                           # If signal, then display level
            ENTER MAMPL
            PRINT VAR:5,15,XREG,DEC:6,1,"DBM"               # Display harmonic level
            ENTER VAR:2                                     # Enter fundamental amplitude
            SUBT                                            # Xreg = fundamental amplitude
                                                            # - harmonic amplitude
            PRINT VAR:5,33,XREG,DEC:6,1                     # Display dbc level
        ELSE
            PRINT VAR:5,6,"NO SIGNAL   ---               ---"# Display "no signal"
        ENDI
    NEXT                                                    # Next harmonic number
    TEXT MACRO                                              # Display macro readout buffer
    ENTER VAR:3                                             # Enter fundamental frequency
    PUTREG FREQ                                             # Tune back to fundamental
                                                            # frequency
ELSE
DSLINE TOP,4,BOTTOM                                         # Display normal top and bottom
                                                            # readout and line 4 in the
                                                            # macro readout buffer

PRINT 4,2,"N O  S I G N A L  F O U N D"
ENDI
DONE                                                        # Done with macro
EMAC                                                        # End macro entry
```

OUTPUT FOR HARMONIC MACRO TEST

(displayed on the CRT of the 49X)

```
HARMONIC TEST

FUNDAMENTAL FREQUENCY,               AMPLITUDE
    100.000 238MHZ                   -10.0DBM
-----------------------------------------
HARM #         LEVEL               DBC
  1          -30.0DBM            -20.0
  2          -35.0DBM            -25.0
  3          -36.2DBM            -26.2
  4          -42.0DBM            -32.0
  5          -54.3DBM            -44.3
  6          -56.0DBM            -56.0
  7  NO SIGNAL  ---                ---
  8          -72.4DBM            -62.4
  9          -79.9DBM            -69.9
 10          -87.0DBM            -77.0
```

PASS/FAIL HARMONIC TEST MACRO

MACRO PREPARATION

The following program shows the 4041 Data statements necessary for the PASS/FAIL HARMONIC TEST program. This macro will display TEST PASSED on line 4 of the crt if the fundamental's 1st harmonic is less than 60 dBc from the fundamental's amplitude. If it is not, TEST FAILED will be displayed on line 4 of the crt. This macro will use less than 5% of the total available 8K bytes of memory set aside for macro storage.

```
 10           Z=1
 20         ! MACRO NUMBER 0   "HARMONIC PASS/FAIL" test
 30           Data "stmac 0,'HARMONIC PASS FAIL'"
 40           Data "marker single;sweep;mfbig;clear 4;dsline top,4,bottom"
 50           Data "if signal;mcen;step primar;marker delta;pstep"
 60           Data "sweep;thrhld -60;mfbig;if nosig;print 4,10,'test passed'"
 70           Data "else;print 4,12,'T E S T   F A I L E D'"
 80           Data "endi;mstep;thrhld auto;else"
 90           Data "print 4,2'N O   S I G N A L   F O U N D';endi;marker off;done;emac"
100         ! MACRO NUMBER 1   "HARMONIC TEST"
110           Data "stmac 1,'HARMONIC TEST'"
120           Data "marker single;rdout normal;clear"
130           Data "sweep;mfbig;if signal;print 1,1,'harmonic test'"
140           Data "print 3,1,'fundamental frequency,   amplitude'"
150           Data "enter mfreq;stnum 3;print 4,1,xreg,fformt:18,0"
160           Data "enter mampl;stnum 2;print 4,28,xreg,dec:6,1"
170           Data "print 5,1'----------------------------------------'"
180           Data "print 6,1,'harm #        level          dbc';step primar"
190           Data "for 1,1,10"
200           Data "enter 6;enter var:1;plus;stnum 5"
210           Data "print var:5,2,var:1,dec:2,0;pstep;sweep;mfbig;if signal"
220           Data "enter mampl;print var:5,15,xreg,dec:6,1,'DBM'"
230           Data "enter var:2;subt;print var:5,33,xreg,dec:6,1"
240           Data "else;print var:5,6,'no signal   ---          ---';endi"
250           Data "next"
260           Data "text macro;enter var:3;putreg freq;else"
270           Data "dsline top,4,bottom;print 4,2,'N O  S I G N A L  F O U N D'"
280           Data "endi;done;emac"
300         ! MACRO LOADING PROGRAM
310           On error(501) then gosub mdone       ! When end of data goto "mdone"
320           Restore                              ! Point to 1st DATA item
330           Read d$                              ! Read data
340           S1=1                                 ! Start at 1st character position
350           E1=1en(d$)                           ! Length of string
360           A1=pos(d$,";",s1)                    ! Position of ";"
370           If a1=0 then goto 420                ! No ";" , send string
380           D1$=seg$(d$,s1,a1-s1)                ! Segment out just one command
390           Gosub send_str                       ! Send the one command
400           S1=a1+1                              ! Set new starting position in string
410           Goto 360
420           D1$=seg$(d$,s1,e1)                   ! Segment out command
430           Gosub send_str                       ! Send string
440           Goto 330                             ! Go read the next data statement
450       Send_str:   print d1$                    ! Print string
460           Print #z:d1$                         ! Send string to spectrum analyzer
470           Print #z:"ERR?"                      ! Check for error
480           Input #z:err$
490           If err$="ERR 0" then return          ! If no error then return
500           Print err$                           ! Print error
510           Stop                                 ! Stop
520       Mdone:   print "All done.  No errors^G"
```

PASS/FAIL HARMONIC TEST

Line 100 — Store the following macro in menu position 2 and title it PASS/FAIL HARMONIC TEST.

Line 110 — Turn on a single marker, do a sweep, and move the marker to the signal peak.

Line 120 — Clear line 4 in the macro readout buffer, and display the normal top and marker readout and line 4 in the macro readout buffer.

Line 140 — Center the marker, put the marker frequency into the step size, turn on delta markers, and do a plus step.

Line 150 — Take a sweep, set the threshold to −60 dBm, and move the marker to the signal peak.

Line 170 — If there is no signal, print TEST PASSED on the fourth line of the screen indented 10 spaces from the left.

Line 190 — If there is a signal, print TEST FAILED on the fourth line of the screen indented 12 spaces from the left.

Line 210 — Move the marke- back to the fundamental, and set the threshold to auto.

Line 230 — If no fundamental frequency was found, print NO SIGNAL FOUND on the fourth line of the screen indented 2 spaces

Line 250 — Set the Triggering to Frerun, and turn the markers off.

Line 260 — The macro is done running.

Line 270 — This is the end of the macro entry.

PASS/FAIL HARMONIC TEST

```
IC PASS/FAIL"                  # Store the following GPIB commands as macro
                               # number 1 (in the macro menu) and title it
                               # HARMONIC PASS/FAIL.
                               # NOTE: The following commands will not be
                               # executed, just stored in NVRAM.
                               # Turn on single markers
                               # Take a sweep
                               # Move marker to the biggest peak
                               # Clear line 4 in macro readout buffer
TTOM                           # Display normal top and bottom readout and
                               #line 4 in the macro readout buffer

                               # Center marker (signal)
.                              # Put marker frequency into step size
A                              # Turn on delta markers
                               # Plus step (tune to harmonic)
                               # Take a sweep
                               # Set threshold to -60
                               # Move marker to peak
                               # If no signal was found (above -60) then the
                               # test passed
,10,"TEST PASSED"              # Print TEST PASSED on line 4.
                               # A signal was found, so test failed
,12,"T E S T   F A I L E D"    # Print TEST FAILED on line 4
                               # Done with IF statement
                               # Move marker back to fundamental frequency
                               # (MINUS STEP)
                               # Reset the signal threshold to auto
                               # No fundamental frequency was found
N O   S I G N A L   F O U N D"# Print NO SIGNAL FOUND on line 4

                               # Turn markers off
                               # Done with macro
                               # End of macro entry
```

MANUAL CHANGE INFORMATION

At Tektronix, we continually strive to keep up with latest electronic developments by adding circuit and component improvements to our instruments as soon as they are developed and tested.

Sometimes, due to printing and shipping requirements, we can't get these changes immediately into printed manuals. Hence, your manual may contain new change information on following pages.

A single change may affect several sections. Since the change information sheets are carried in the manual until all changes are permanently entered, some duplication may occur. If no such change pages appear following this page, your manual is correct as printed.