TEK PROGRAMMERS MANUAL

Part No. 070-4415-00
Product Group 26

494P
SPECTRUM ANALYZER

Please Check for
CHANGE INFORMATION
at the Rear of This Manual

First Printing OCT 1983
Revised MAR 1985

Tektronix®
COMMITTED TO EXCELLENCE

PREFACE

This manual is one of a set for the TEKTRONIX 494P and non-programmable 494 Spectrum Analyzers. The manuals that are available at this time in addition to this 494P Programmers are the 494/494P Operators, 494/494P Operators Handbook (a small manual that fits in the instrument front cover), and 494/494P Service, Volumes 1 and 2.

Refer to the 494/494P Operators manual for a full description of instrument functions and front-panel controls. The Operators manual also contains the full specification of instrument performance.

This manual describes the programmable functions of the 494P and how to use them for remote operation.

Sections 1 and 2 help you get started using the 494P on the IEEE Std 488 General Purpose Interface Bus (GPIB). Programming examples are included here as well as throughout the manual. Some examples are given for a variety of GPIB controllers, but most are in BASIC as implemented on TEKTRONIX 4050-Series controllers. Comments are included to help you translate if you are using another controller.

Sections 3 through 7 are a reference to the language used to set and read 494P functions and transfer spectrum data acquired by the 494P. Section 3 defines device-dependent message format and execution. Sections 4 through 7 cover the commands and queries by function; front-panel, display data I/O, waveform processing, and system operation.

Section 8 is a how-to section for making programmable measurements with the 494P.

Appendix A will help you understand the GPIB and the IEEE 488 standard on which it is based.

Appendix B includes two foldout pages containing an index and a comprehensive list of remote control commands and queries and error/event responses.

Document Standards and References Used

Terminology used in this manual is in accordance with industry practice. Abbreviations are in accordance with ANSI Y1.1-1972, with exceptions and additions explained in parentheses after the abbreviation. Graphic symbology is based on ANSI Y32.2-1975. Logic symbology is based on ANSI Y32.14-1973 and the manufacturer's data books or sheets. A copy of these ANSI standards may be obtained from the Institute of Electrical and Electronic Engineers, 345, 47th Street, New York, NY 10017.

Change/History Information

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

History information, with the updated data, is integrated 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.

TABLE OF CONTENTS

TABLE OF CONTENTS (cont)

TABLE OF CONTENTS (cont)

TABLE OF CONTENTS (cont)

LIST OF TABLES

LIST OF ILLUSTRATIONS

SAFETY SUMMARY

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

The 494P complies with the following Industry Safety Standards and Regulatory Requirements.

Safety

CSA: Electrical Bulletin

FM: Electrical Utilization Standard Class 3820
ANSI C395 — 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.

PRECAUTIONS

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 receptacle before connecting to the product input or output terminals. A protective ground connection by way of the grounding conductor in the power cord is essential for safe operation.

Danger Arising 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.

Refer cord and connector changes to qualified service personnel. For detailed information on power cords and connectors, see the Maintenance section in the 494/494P Service manual, Volume 1.

Use the Proper Fuse

To avoid fire hazard, use only the fuse of correct type, voltage rating and current rating (as specified in the Replaceable Electrical Parts list in Volume 2 of the 494/494P Service manual) for your product.

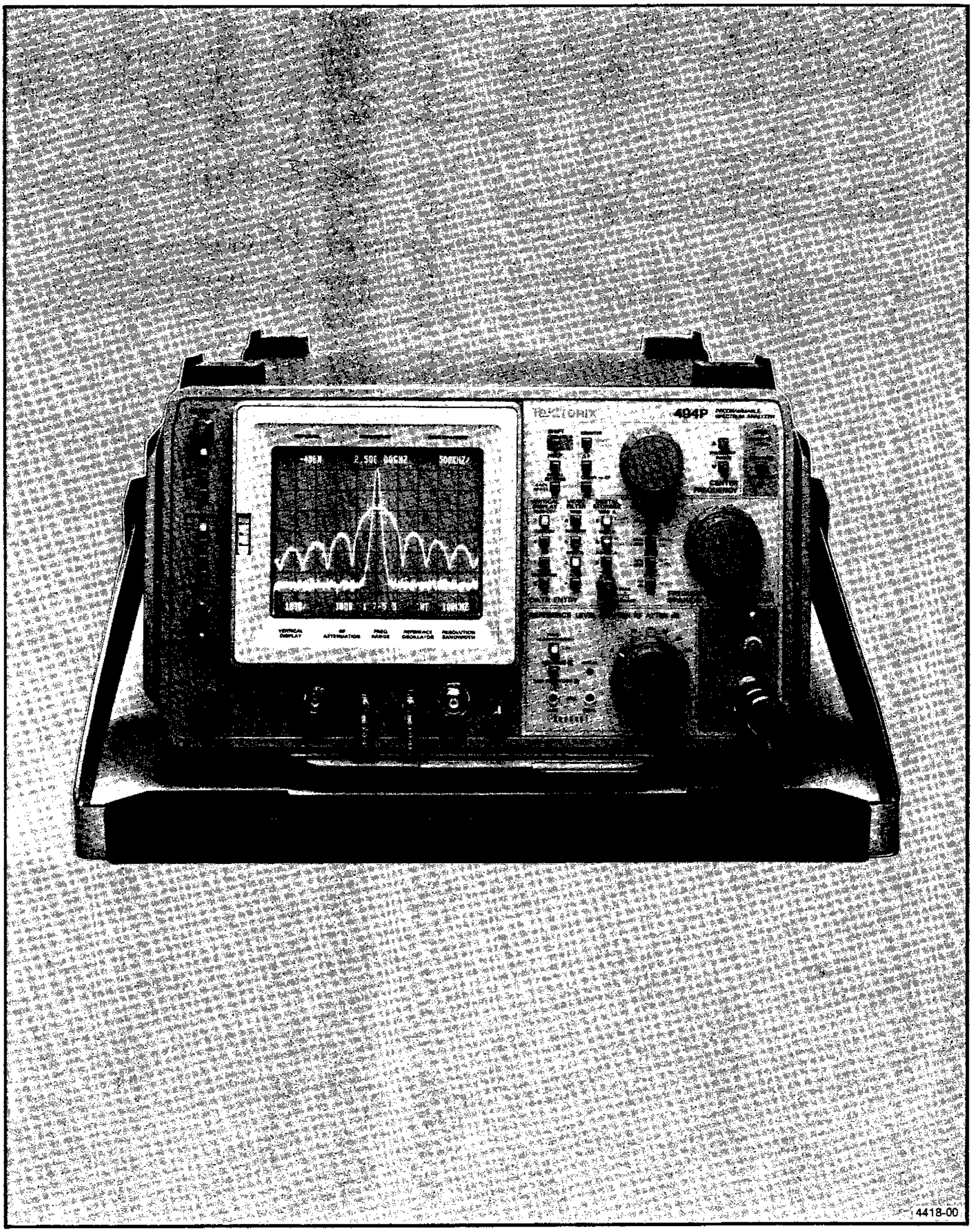
Refer fuse replacement to qualified service personnel.

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. Do not operate the product without the covers and panels properly installed.



The TEKTRONIX 494P Programmable Spectrum Analyzer.

INTRODUCTION TO GPIB OPERATION

INTRODUCTION

The TEKTRONIX 494P adds remote control and automated spectrum data acquisition and analysis to the performance and portability features of the TEKTRONIX 494 Spectrum Analyzer. The 494P 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 added in the 494P allows it to be used with a wide variety of systems and controllers. This versatility is accomplished because the 494P is implemented according to the Tektronix Interface Standard for GPIB Codes, Formats, Conventions, and Features. This standard promotes ease of operation, and makes the 494P compatible with other Tektronix instruments and, as much as possible, with GPIB instruments from other manufacturers.

NOTE

Some of the lines of input in examples of controller programs in this section extend beyond the column width limitations. Where this occurs, the overrun information is indented on the immediately-following line.

Important—whenever a line is broken, it is always where a natural space occurs. So, be sure to add a space when inputting the program.

GPIB CONTROLS AND INDICATORS (Figure 1-1)

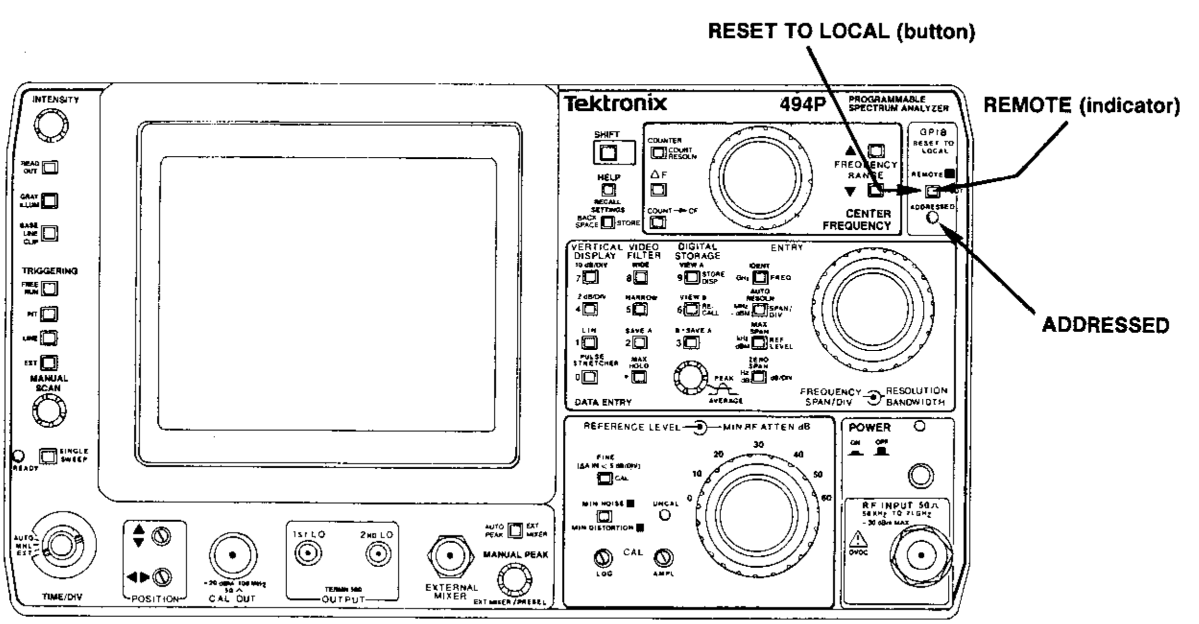


Figure 1-1. GPIB control and indicators.

RESET TO LOCAL/REMOTE

This button is lighted when the analyzer is under control of the GPIB controller. While under remote control, the other 494P front-panel controls are not active, but indicators will still reflect the current state of all front-panel functions.

This button is not lighted when the analyzer is under local operator control. While under local control, the 494P does not execute GPIB messages that would conflict with front-panel controls or change the waveforms in digital storage.

When the button 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 internal microcomputer flashes the instrument and front-panel firmware version numbers and the GPIB address on the crt when the button is pressed. The microcomputer also updates the GPIB primary address if the GPIB ADDRESS switches have been changed.

For another function of this button when in the talk-only mode, refer to Talk-Only/Listen-Only Operation later in this section.

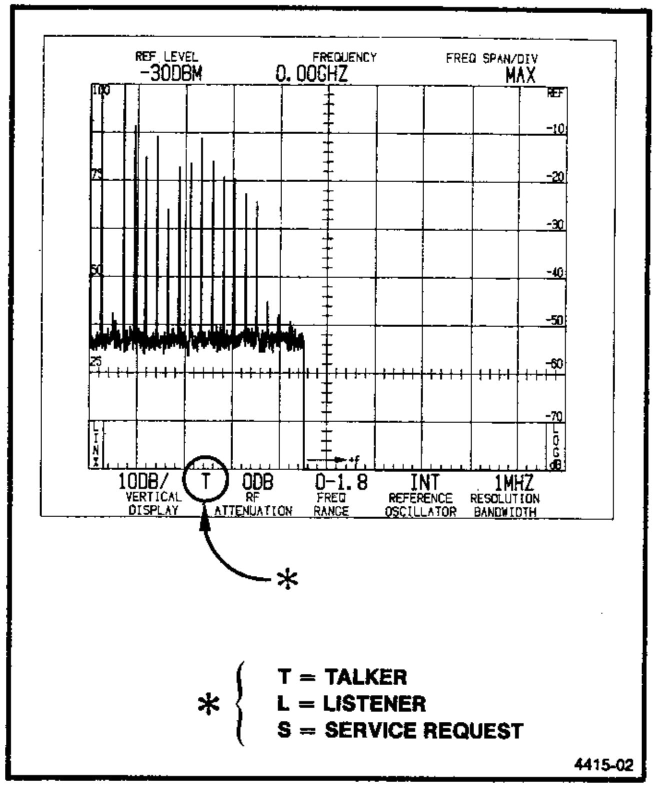


Figure 1-2. Status of active GPIB functions.

<SHIFT> PLOT

If this button is pressed when the 494P is in the talk-only or listen-only mode (see the TALK ONLY, LISTEN ONLY switch descriptions later in this section), it causes the 494P to send the appropriate commands over the GPIB to a plotter connected to the bus. The 494P display can be recreated on a TEKTRONIX 4662 or 4662 Opt 31 Interactive Digital Plotter (or a 4663 in the 4662 emulation mode) or a HP7470A plotter. The plotter must be in the listen-only mode. A bus controller is not required.

ADDRESSED

This button lights when the analyzer is addressed to listen or talk.

GPIB Function Readout

A single character appears in the lower crt readout when the 494P is talking (T), listening (L), or requesting service (S); see Figure 1-2. Two characters will appear in this location if the 494P is talking or listening and also requesting service.

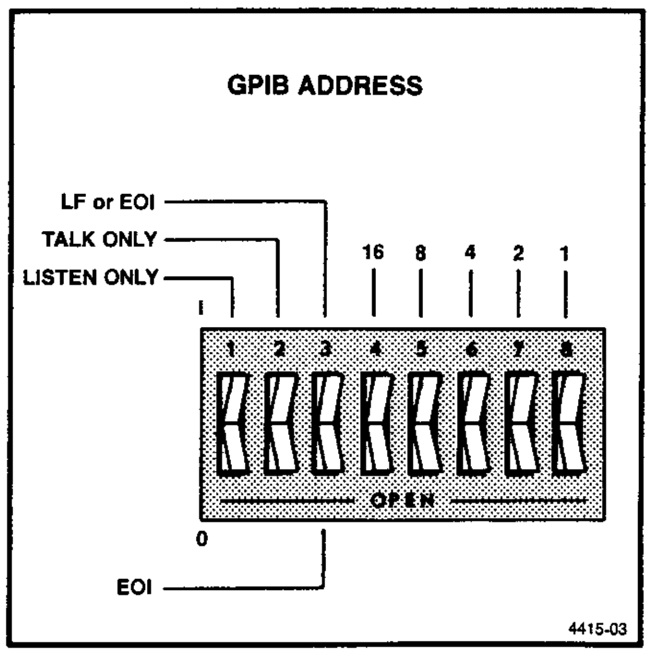


Figure 1-3. Rear-panel GPIB ADDRESS switches. The LF OR EOI switch (message terminator) and the TALK ONLY and LISTEN ONLY switches are part of the same switch bank.

Setting the GPIB ADDRESS Switches

The rear-panel GPIB ADDRESS switches shown in Figure 1-3 set the value of the 494P GPIB addresses (refer to Table 1-1). The instrument's primary address (0 through 31) is the value of the lower five bits, which are switches 4 through 8. The listen and talk addresses are set by the two left-most switches. The internal microcomputer reads these switches at power-up and again each time the RESET TO LOCAL button is pressed or the <SHIFT> PLOT sequence is performed.

Table 1-1
BUS ADDRESSES

16	8	4	2	1	Primary Address	Listen Address	Talk Address
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

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 494P, so are ignored.

Set the switches as desired, but do not use 0 with 4050-Series controllers; they reserve this address for themselves. Selecting a primary address of 31 logically removes the 494P from the bus; it does not respond to any GPIB address, but remains both unlistened and untalked. Remember, if you change these switches after the 494P is already powered-up, you must press RESET TO LOCAL or <SHIFT> PLOT to cause the microcomputer to update the primary address.

Setting the LF OR EOI Switch

Switch 3 of the rear-panel GPIB ADDRESS switch (see Figure 1-3) selects the terminator for messages on the bus. If LF OR EOI is selected (switch up, 1), the 494P 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 494P interprets the byte sent with the end message (EOI asserted) as the end of a message.

This switch also selects the output terminator. Set to LF OR EOI, the 494P adds CR and LF (with EOI asserted concurrently) after the last byte of the message. Set to EOI, the 494P 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 some other controllers, such as the Hewlett-Packard 9826A. A change in this switch takes immediate effect.

TALK ONLY, LISTEN ONLY Switches

The 494P switches for talk-only and listen-only operation are part of the GPIB ADDRESS switch bank shown in Figure 1-3. Set either or both switches; an extension of the IEEE 488 standard allows you to enable both talk-only and listen-only operation. If 494P power is on, press RESET TO LOCAL or <SHIFT> PLOT to cause a change in these switches to take effect. Both the TALK ONLY and LISTEN ONLY switches must be off (down) when the 494P is used with any controller.

Set the LF OR EOI switch to EOI (down) for use with Tektronix equipment. The switches marked 1, 2, 4, 8, and 16 may be set to any combination except all ones (decimal 31), which logically disconnects the 494P from the bus.

Talk Only/Listen Only Operation

The 494P can be operated as a talker only or a listener only on the GPIB under local control. This requires only the 494P and a talker or listener. Such a system uses the TEKTRONIX 4924 Digital Cartridge Tape Drive with the 494P.

This system can be used to save spectrum measurements for later display on the 494P or for analysis by a controller. This system will also save and restore analyzer control settings. Follow the Data Logging instructions under 494P Operation with the 4924.

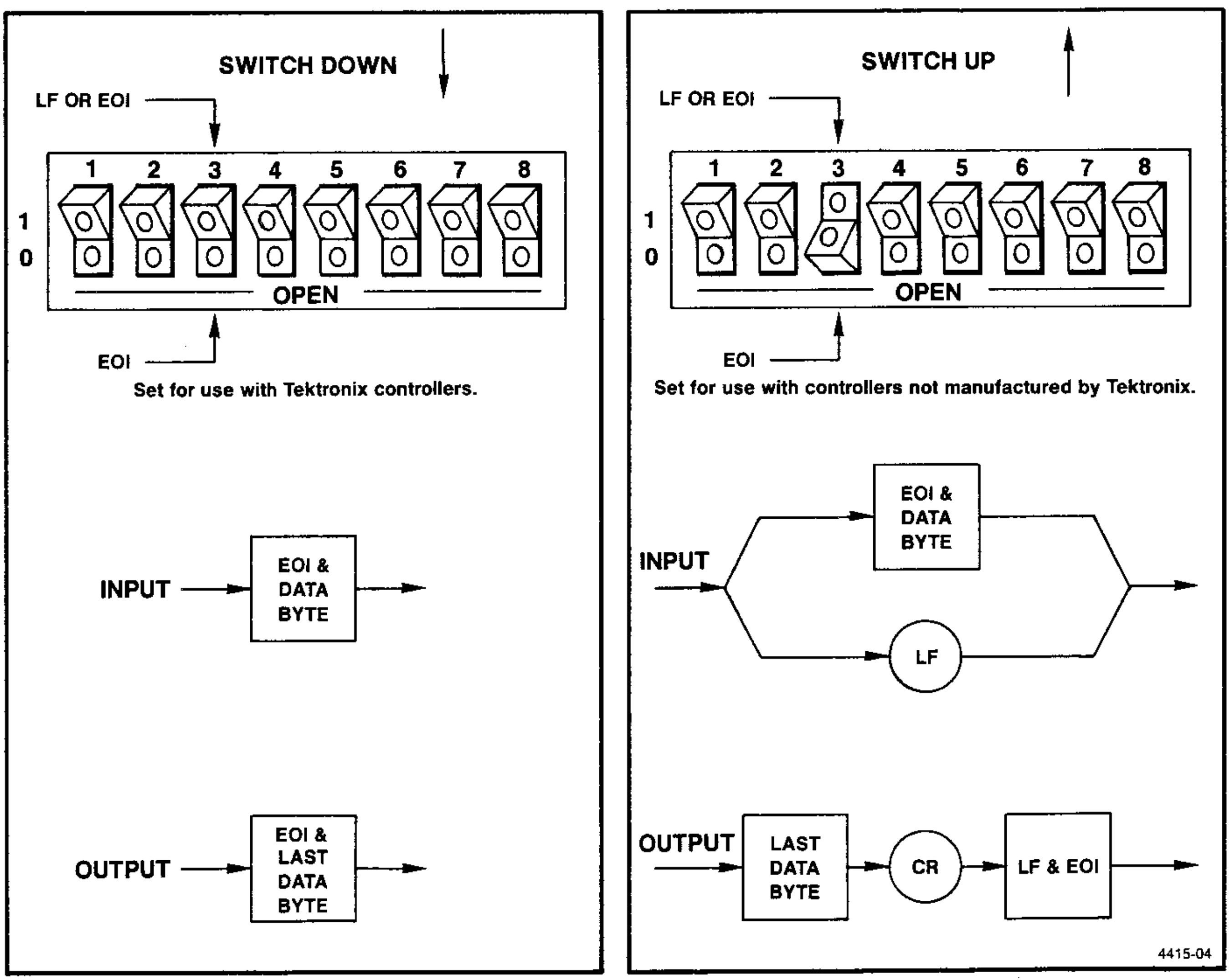


Figure 1-4. Effect of message terminator switch for input and output.

494P OPERATION WITH THE 4924

Data Logging

To operate with the 494P, the Mode Control switches on the 4924 rear panel must be set as a pair in any combination except SW1 OFF and SW2 ON. Set SW1 to ON and SW2 to OFF (same as for operation with the 4051) or set both switches to the same position (both SW1 and SW2 ON or OFF).

With the 494P TALK ONLY switch set (down), you can write spectrum data onto a tape in the 4924 using the controls shown in Figure 1-5.

1. Insert a marked tape into the 4924. The tape must be previously marked for the size and number of files you expect to record.

Use the MARK command to mark the tape in a 4050-series controller.

```
MARK n,4500
```

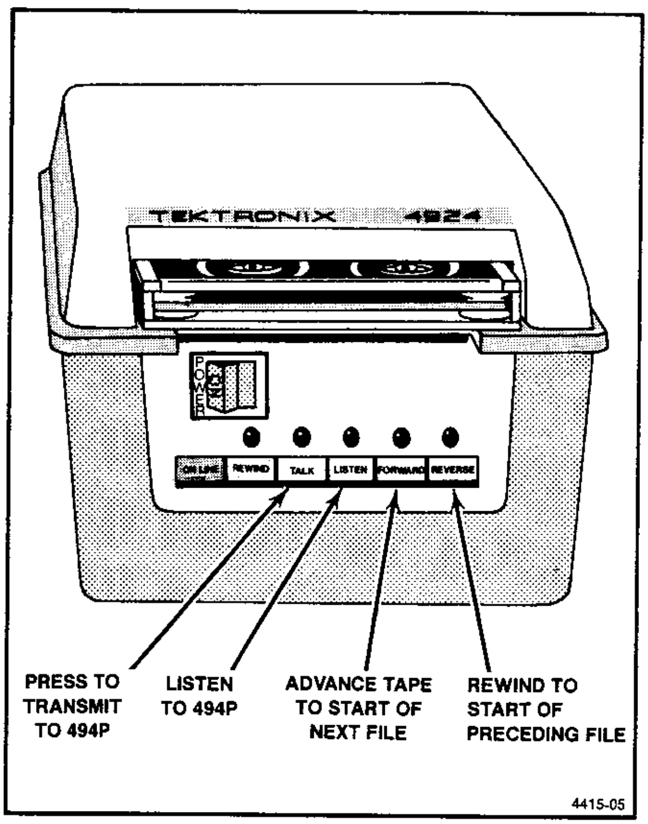


Figure 1-5. 4924 controls used for data transfers.

This command marks n (a number you choose) files to be big enough to store both a waveform and its control settings on each file.

2. Connect the 4924 and 494P with a GPIB cable after both are powered up.

3. Set the 4924 On Line switch out (off line).

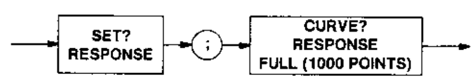
4. Rewind the tape.

5. Press the 4924 Forward button to advance the tape to the beginning of file 1. To reach a file further into the tape, press Forward again as many times as desired.

6. To save the current 494P control settings and waveform in digital storage, press Listen on the 4924 and RESET TO LOCAL on the 494P.

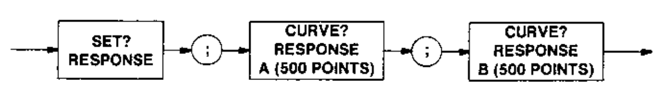
Press the RESET TO LOCAL button on the analyzer to transmit the instrument settings and waveform. The message is formatted so that when it is played back to the analyzer, it restores the settings and display.

The message is a combination of the responses to the SET and CURVE queries. If SAVEA is OFF, A and B are transmitted as a full waveform (A and B memories merged for 1000 points).



4415-88

If SAVEA is ON, A and B are transmitted as separate waveforms (500 points each).



4415-89

The analyzer transmits waveform data as ASCII-coded decimal numbers unless changed to binary by the ENCDG argument in a WFMPRE command. You'll find the full CURVE? response syntax diagram in Section 5. See Section 7 for the full SET? response syntax diagram.

NOTE

If an internal switch is changed, the analyzer reports control settings only when RESET TO LOCAL is pressed. Refer questions about setting this internal switch to qualified service personnel.

The 4924 keeps listening (or talking if Talk is pressed) until the message transfer is completed; there is no reset switch except Power. Once the 494P starts talking, it keeps talking until it is finished, and cannot be interrupted except by turning off the power. (This is true only if the 494P begins transmitting; if there is no listener, the message NO LISTENER is flashed to the operator and the 494P returns to local control.)

7. To move to the next 4924 file, press Forward. To move to the previous file, press Reverse. To move to the beginning of the same file, press Reverse, then Forward.

Restoring Control Settings and the Display From Tape

With the LISTEN ONLY switch set, the 494P buffers and executes device-dependent messages, except for interrupt control commands EOS and RQS. Since the remote/local state diagram in the IEEE 488 standard does not cover the listen-only mode, this mode is implemented in the 494P so it goes to the remote state after buffering a message. This makes the listen-only mode consistent with the nonlisten-only mode. The nonlisten-only mode requires that the 494P be under remote control to execute commands that change front-panel settings or waveform data in digital storage.

To restore control settings and a previously recorded display, find the file on the tape using Forward or Reverse on the 4924 and press Talk. The 494P goes to remote to execute the message and then returns to local control.

The listen-only mode can be used for a comparison test. Settings and a waveform previously recorded with SAVEA on can be played back to the analyzer. The analyzer automatically sets up to make the same measurement (turning on SAVEA), and saves the comparison waveform in A memory. If B−SAVEA is selected, the current spectrum data being acquired in B memory can be compared to the waveform saved in A memory.

Putting a Counter on the Tape

How do you keep track of where you are on the tape? You can keep track of files going by when you use the 4924 Forward and Reverse buttons. Or you can record a message on every other tape file as a marker. With the latter scheme, press Forward and then Talk after either recording or playing back a file. This causes a message with the number of the next file to appear on the 494P crt. Here's a 4050-Series program to mark and record the tape for this purpose.

```
100 REMARK PROGRAM TO MARK TALK/LISTEN TAPE
110 FOR I=1 TO N
120 FIND 2*I-1
130 MARK 1,500
140 FIND 2*I-1
150 PRINT @33:"RDOUT 'FILE ";I;" IS
    NEXT-PRESS FORWARD';RDOUT""
160 FIND 2*I
170 MARK 1,4500
180 NEXT I
```

This program works on a tape that has been previously marked for any purpose. For a new tape, enter the following 4050-Series commands before running the program.

```
FIND 1
MARK 1,500
```

Line 110: Substitute the number of files you want to mark for N.

Line 150: The RDOUT command replaces the crt readout with a message showing the file count and a reminder to move to the beginning of the file.

Line 170: Each time through the loop, the file is marked to be used for recording.

As an alternative, the PRINT statement in line 150 could be expanded to include an instrument setup. To do this, make up a character string to hold both the control settings and a crt message. When the analyzer receives this longer string, it will restore its controls to make the desired measurement and display a message to the operator.

IEEE 488 FUNCTIONS

The 494P is compatible with IEEE Standards 488-1975 and 488-1978. The connector and the signal levels at the connector follow the specifications in the IEEE 488 standards. Table 1-2 lists 494P interface capabilities, as defined in the standards.

Table 1-2
494P 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	CO

Source Handshake (SH1)

The 494P has complete capability for transfer of 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 494P has complete capability to receive messages on the bus.

Talker (T5)

The 494P employs the complete talker function including serial poll; unaddresses as a talker when addressed as a listener. The analyzer operates in a simple system in a talk-only mode if the TALK ONLY switch is set to 1.

Listener (L3)

The 494P employs the complete listener function; unaddresses as a listener when addressed as a talker. The analyzer operates in a simple system in a listen-only mode if the LISTEN ONLY switch is set to 1.

Service Request (SR1)

The 494P employs the complete service request function; asserts SRQ for the conditions indicated under Status Byte in Section 7 and reports the corresponding status when polled.

Remote/Local (RL1)

The 494P employs the complete remote/local function. The front-panel RESET TO LOCAL button 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 7 for the affect of busy status on remote/local transitions.

The current value of all programmable functions is maintained when switching from local to remote control. Only the value of TIME, MINATT, and CRSOR may change to match the front-panel control settings when switching from remote to local control, since they don't conflict with local control. In either case, all front-panel indicators show the current value of the functions.

The 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 goes false. The analyzer changes settings for which there is no local control and outputs data while under local control.

Parallel Poll (PP1)

The 494P responds to a parallel poll to indicate if service is requested.

Device Clear (DC1)

The 494P 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 7 for more on power-up status and for the effect of busy status on the execution of DCL and SDC.

Device Trigger (DT1)

The 494P device trigger function is implemented so the group execute trigger (GET) message causes the instrument to abort 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 494P must be in the Remote mode for GET to have any effect.

Controller (CO)

The 494P does not act as a controller.

CONNECTING TO A SYSTEM

The 494P can be connected directly to a GPIB system with the cable supplied with the instrument. The IEEE STD 488 PORT is shown in Figure 1-6. To avoid interference on the bus, connect the 494P after turning on power or while the controller on the bus is turned off.

The GPIB is a flexible system that works either in a star or linear configuration as shown in Figure 1-7. Up to 15 devices can be connected. 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 494P, etc.), and at least two thirds of the devices connected must be powered up. (Appendix A details the IEEE STD 488 GPIB System Concepts.)

If an internal switch is changed, the 494P asserts SRQ on power-up. This requires immediate action by some controllers, such as Tektronix 4050-Series, so is not recommended for these controllers. Other internal switches select self-test modes at power-up; changing these switches prevents the 494P from operating normally. Because changing these switches requires that the cover be removed, refer this task to qualified service personnel.

A turn-on procedure is provided in both the 494/494P Operators Manual and the 494/494P Operators Handbook. Refer to those books for instructions on how to begin operating the instrument.

The power-up condition of all programmable functions is restored by the INIT command. Refer to Section 7 for more on this command and a list of the power-up parameters.

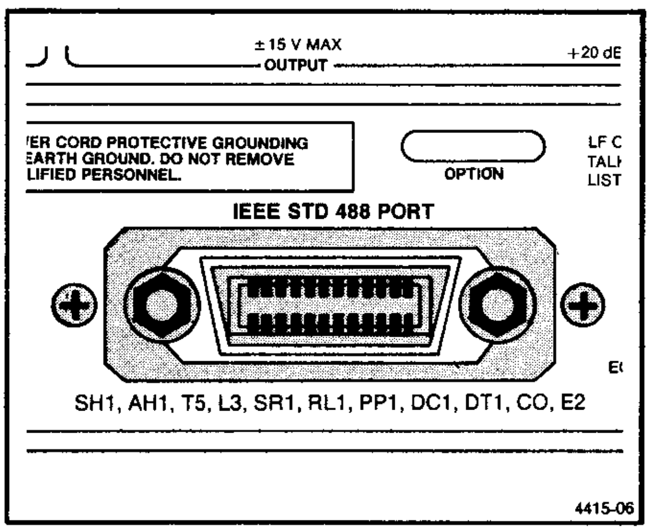


Figure 1-6. The rear-panel IEEE STD 488 PORT (GPIB).

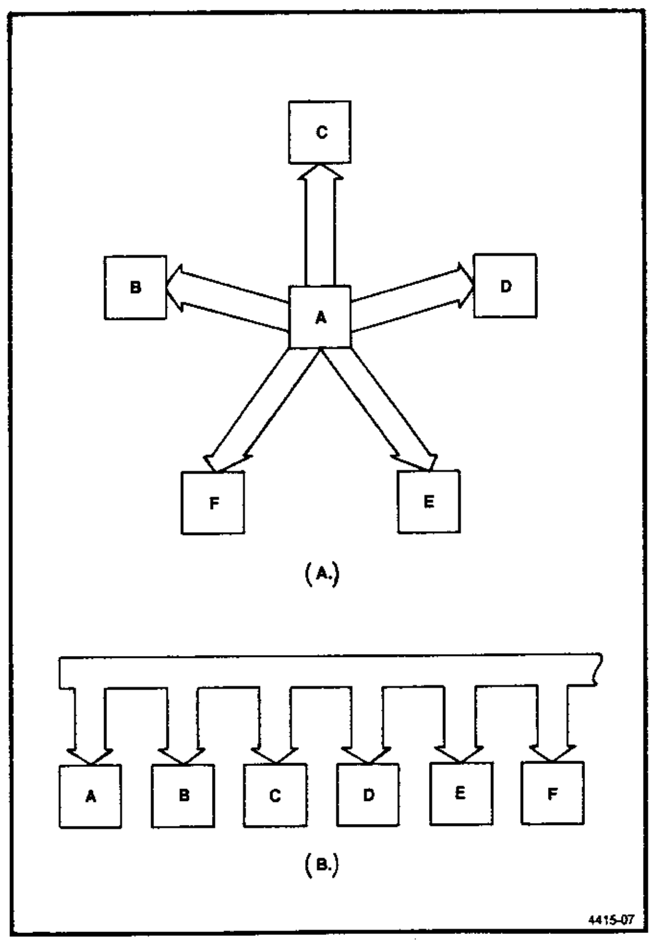


Figure 1-7. The 494P can be connected to a GPIB system in either a star (A) or a linear (B) configuration.

GETTING STARTED

INTRODUCTION

Getting started with the 494P on the GPIB is a simple matter if you are already familiar with a GPIB controller. If not, talking to the 494P over the bus may be the easiest way to get over any uncertainty you feel about getting started.

The 494P speaks a friendly language that includes mnemonics for control of the front panel and other parameters 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 494P 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 4050-Series controllers (in BASIC), Tektronix CP1100 and CP4100-Series controllers (in TEK SPS BASIC), and the Hewlett-Packard 9826A controller.

NOTE

Some of the lines of input in examples of controller programs in this section extend beyond the column width limitations. Where this occurs, the overrun information is indented on the immediately-following line.

Important—whenever a line is broken, it is always where a natural space occurs. So, be sure to add a space when inputting the program.

SETTING AND QUERYING PROGRAMMABLE CONTROLS

SETTING PROGRAMMABLE CONTROLS

We can keep this simple, because the 494P lets you make complex spectrum measurements semi-automatically. Many measurements can be made with just three front-panel settings; FREQ, SPAN/DIV, and REF LEVEL.

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

The SPAN/DIV setting changes the size (width) of the window, setting the frequency calibration of the crt horizontal axis.

The REF LEVEL setting raises or lowers the window, which sets the amplitude calibration of the top graticule line on the crt.

Here's how to program the 494P to measure the CAL OUT signal, using the front-panel pushbuttons for these three settings.

1. The CAL OUT signal can be centered by pressing the <SHIFT> FREQ 1 0 0 MHZ pushbuttons.

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

The 494P 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.

3. Set the signal to the reference level by pressing the <SHIFT> REF LEVEL 2 0 −dBM pushbuttons.

The 494P 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 494P microcomputer takes into account the MIN RF ATTEN dB and MIN NOISE settings when positioning the attenuation and gain.

The 494P powers up with the automatic modes active and in MAX SPAN to display all the frequencies. You can restore this condition at any time with the INIT (initialize) command.

4050-Series Controller

How do steps 1, 2, and 3 in the last example work on a Tektronix 4050-Series controller? The 494P commands are inserted in the following GPIB output statement PRINT. Throughout the 4050-Series BASIC examples in this manual, the variable Z has been assigned to the value of the 494P GPIB address. Any constant can be used to represent the number for the GPIB address.

```
100 PRINT @Z: "FREQ 100 MHZ"
110 PRINT @Z: "SPAN 1 MHZ"
120 PRINT @Z: "REFLVL -20 DBM"
```

or

```
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 494P executes these commands, it tunes the CAL OUT signal to center screen, magnifies the narrower span, and changes the reference level to display the signal peak at the top of the screen. Frequency range, resolution bandwidth, time/division, input attenuation, and IF gain are changed automatically, as necessary. Because the 494P 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 either the 494/494P Operators Manual or 494/494P Operators Handbook or, better yet, try the automatic CAL function provided and described in the Operators manual and handbook.

If you receive an SRQ message on the screen of the 4050-Series controller, add a SRQ handler sequence to the program. This sequence can be added to any 4050-Series program example shown in this manual. The amended program would look like the following.

```
90 ON SRQ THEN 140
100 PRINT @Z: "FREQ 100 MHZ"
110 PRINT @Z: "SPAN 1 MHZ"
120 PRINT @Z: "REFLVL -20 DBM"
130 POLL Q1,Q2;Z
140 PRINT "SRQ ";Q2
150 RETURN
```

or

```
90 ON SRQ THEN 110
100 PRINT @Z: "FREQ 100 MHZ;SPAN 1 MHZ;REFLVL
    -20 DBM"
110 POLL Q1,Q2;Z
120 PRINT "SRQ ";Q2
130 RETURN
```

CP1100 and CP4100-Series Controllers

The story is the same for two other Tektronix controllers. Only the names change for the I/O statements. With TEK SPS BASIC in Tektronix CP1100 and CP4100-Series Controllers, the output statement is PUT.

```
100 PUT "FREQ 100 MHZ;SPAN 1 MHZ;REFLVL
    -20 DBM" INTO @N,LZ
```

where N is the number of the GPIB interface in the controller and LA is the 494P listen address (primary address plus 32).

9826A Controller

For the Hewlett-Packard 9826A Desktop Computer, use the write statement.

```
1: wrt "z","freq 100 mhz;span 1 mhz;reflvl
   -20 dbm"
```

A 9826A device statement is used to assign "z" to the 494P.

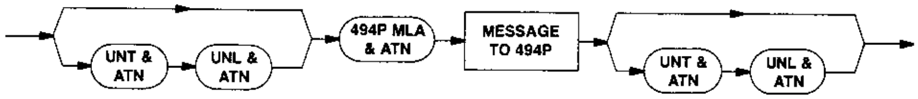
```
0: dev "z",701
```

The device statement assigns address 701 to "z". The 01 in 701 assumes the 494P GPIB ADDRESS switches are set to 1, but could be changed to any number between 00 and 30. (Or "z" could be replaced in the write statement by 700 + the 494P primary address.)

The 9826A should be equipped with a GPIB Interface, the General I/O and Extended I/O ROM, and the String and Advanced Programming ROM to operate with the 494P.

Summation

Whatever controller is used or statement being sent, the actions shown in the following syntax diagram must be taken to get a message to the 494P.



4415-86

The unlisten (UNL) and untalk (UNT) messages are optional in the previous syntax diagrams 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 494P listen address or expects to receive the listen address with the offset included (i.e., 32). The controller then sends the device-dependent message you insert into the statement, and may finish by sending UNL and UNT. If the controller does not assert REN automatically for GPIB I/O, you can set it with an earlier control statement. The 494P does not balk if REN is not set, except if 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 494P 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; the front-panel functions are in Section 4 with other functions covered in following sections (refer to the Index for page numbers).

The 494P executes the message when it sees the message terminator (either EOI or LF). Message syntax and command execution is given fuller treatment in Section 3.

QUERYING PROGRAMMABLE CONTROLS

The 494P returns the state of programmable controls when queried. This takes two steps.

1. Query the 494P. 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.

For example, the auto resolution mode selected a resolution bandwidth to go with a span of the selected 1 MHz. What is that bandwidth? The query RESBW? readies the 494P to output the answer.

The query can be inserted in any message to the 494P. It is executed in its turn. This means that if RESBW? precedes the SPAN command in the previous example, the 494P 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.

4050-Series Controller

```
100 PRINT @Z: "FREQ 100 MHZ;SPAN 1 MHZ;REFLVL
    -20 DBM;RESBW?;VIDFLT?"
110 INPUT @Z:P$
120 PRINT P$
```

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

CP1100 and CP4100-Series Controllers

```
100 PUT "FREQ 100 MHZ;SPAN 1 MHZ;REFLVL
    -20 DBM;RESBW?;VIDFLT?" INTO @N,LZ
110 GET P$ FROM #N,TZ
```

TZ is the 494P talk address derived in the same manner as LZ (i.e., primary address + 64).

9826A Controller

```
1: wrt "z","freq 100 mhz;span 1 mhz;reflvl
   -20 dbm;resbw?;vidflt?"
2: red "z",P$
```

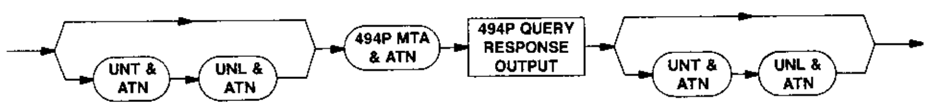
With the 9826A, you must dimension P$ before using it, for example,

```
0: dim P$[500]
```

Summation

Whatever the controller input statement, the actions shown in the syntax diagram below must be taken to receive a message from the 494P.

The syntax diagram to receive a message could be appended to the end of the one shown to send a message. Together, they describe the two steps necessary to obtain output from the 494P. The message in the first diagram would include the query, and the response in the second diagram would come from the 494P to answer that query.



4415-87

EXERCISE ROUTINES

LISTEN/TALK

Now let's put the statements for message I/O together to exercise the 494P as a listener and a talker. This routine is handy because it waits for your input and sends it, time after time. If the 494P 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 7. (The HELP query will return an alphabetized list of the available commands and queries.)

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

The following routines make use of one of the friendly features of the 494P. When the 494P 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 494P sends a byte with all ones).

The SRQ handler employs another 494P feature. Rather than print a code for the status byte, the routine asks for the error that caused the SRQ (EVENT?). This offers much more specific information about the problem. The meaning of the event codes is listed in Table 7-4 in Section 7.

The routines assume you have assigned the value of the 494P address to variable Z (or LZ/TZ or z) as previously discussed. It is also assumed your input and output character strings will fit P$. This gets further attention with regard to the instrument settings query (SET?), our next topic.

In the routine for the CP1100 and CP4100-Series controllers, lines 10 and 20 are added to load the software driver and disable the bus timeout feature in this driver.

4050-Series Controller

```
100 ON SRQ THEN 200
110 PRINT "ENTER MESSAGE"
120 INPUT P$
130 PRINT @Z:P$
140 INPUT @Z:P$
150 GO TO 110
200 POLL Q1,Q2;Z
210 PRINT @Z: "EVENT?"
220 INPUT @Z: P$
230 PRINT P$
240 PRINT "SRQ ";Q2
250 RETURN
```

CP1100 and CP4100-Series Controllers

```
10 LOAD "GPI"
20 SIFTO @N,1
100 WHEN @N HAS "SRQ" GOSUB 200
110 PRINT "ENTER MESSAGE"
120 INPUT P$
130 PUT P$ INTO @N,LZ
140 GET P$ FROM @N,TZ
150 PRINT P$
160 GO TO 110
200 POLL @N,Q2,Q1,Q0;TZ
210 PUT "EVENT?" INTO #N,LZ
220 GET Q$ FROM #N,TZ
230 PRINT Q$
240 RETURN
```

9826A Controller

```
0: dim P$[500]
1: oni 7, "srq"
2: eir 7
3: ent "enter message",P$
4: wrt "z"P$
5: red "z",P$
6: prt P$
7: gto 3
8: "srq":rds("z")rl
9: wrt "z",""event?"
10: red "z",P$
11: eir 7;iret
```

ACQUIRING INSTRUMENT SETTINGS WITH SET?

The SET query enables the 494P to learn instrument 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 494P front panel. Store the message as it is transmitted by the 494P using the SET query. Your controller must be ready for a long character string. Dimension a string variable large enough for at least 500 characters for SET?, although the exact size depends on the current settings. Then, perform any desired instrument operations. Finally, restore the 494P to the original settings by transmitting back to the instrument the stored SET? response. (A 4050-Series program follows that steps you through the operation.) If you wish to transmit instrument settings from battery-powered memory to a controller, first recall the desired settings, then execute the SET? sequence that follows.

4050-Series Controller

Here is a 4050-Series BASIC program that allows you to store two instrument set-ups by pressing User Definable keys. (Only keys 1 and 2 are used in this program, but others could be used in a similar manner.) To restore the set-ups, simultaneously press SHIFT and the key used to store the settings.

NOTE

Be sure to use the same line numbers we used in the example for all input messages on lines 1 through 50. Many of the line numbers directly relate to the special characteristics of the 4050-Series User Definable keys (refer to the 4050-Series Operators manual for more information). Also, portions of this main program will be expanded for use in Acquiring A Waveform later in this section. Save this program as it is here for later use.

```
1 REMARK SETTINGS PROGRAM
2 GO TO 100
4 K=1
5 GOSUB 1000
6 RETURN
8 K=2
9 GOSUB 1000
10 RETURN
44 K=1
45 GOSUB 2000
46 RETURN
48 K=2
49 GOSUB 2000
50 RETURN
100 DIM K$ (500 )
110 DIM L$ (500 )
120 SET KEY
130 GO TO 130
1000 REMARK LEARN INSTRUMENT SETTINGS
1010 PRINT "494P SETTINGS NOW LEARNED"
1020 PRINT @Z: "SET? "
1030 GO TO K OF 1040,1060
1040 INPUT @Z:K$
1050 GO TO 1070
1060 INPUT @Z:L$
1070 RETURN
2000 REMARK RESTORE INSTRUMENT SETTINGS
2010 PRINT "494P SETTINGS NOW RESTORED"
2020 GO TO K OF 2030,2050
2030 PRINT @Z:K$
2040 GO TO 2060
2050 PRINT @Z:L$
2060 RETURN
2070 END
```

Lines 4 through 50: Call subroutines when 4050-Series User Definable keys 1 or 2 are pressed. They set a parameter to indicate whether you are asking for learn string #1 or #2, then jump to subroutines that perform the transfer of settings to or from the 494P.

Lines 100 through 130: Exercise the subroutines, dimension the string variables, and arm the User Definable keys (SET KEY). During the idle time established by line 130, the controller waits for the first input of settings from the 494P.

Lines 1000 through 1070: Input a SET? response from the 494P and stores it.

Lines 2000 through 2060: Return a SET? response to the 494P.

RESETTING THE 494P AND INTERFACE MESSAGES

The INIT command resets the 494P programmable controls to their power-up state (see Section 7 for more on this command). INIT is sent in the same manner as other commands.

Interface message DCL clears the 494P I/O buffers and can be used to restart bus communications with the analyzer. DCL does not interrupt message execution except for the WAIT command. If the 494P is waiting for its talk address so it can execute an output query, output is aborted 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).

For addressed commands such as GTL (go to local) to be executed, precede the decimal code with the 494P listen address. The codes for the addressed commands are 1 for GTL, 5 for PPC (parallel poll configure), and 8 for GET (group execute trigger). GET causes the analyzer to abort the current sweep and immediately start another sweep, synchronizing data acquisition with the interface message.

When the IFC line is asserted by the controller, as when the BASIC statement INIT is executed, the 494P talker and listener functions are initialized (same effect as UNT and UNL).

4050-Series Controller

Use the WBYTE statement to send the universal commands.

```
100 WBYTE @20:
```

For addressed commands, precede the decimal code with the 494P listen address.

```
100 WBYTE @L,1,63:
```

where L = GPIB address + 32.

CP1100 and CP4100-Series Controllers

Use the SIFCOM statement to send the commands, entering the mnemonic for the interface message.

```
100 SIFCOM @N,LZ,"GTL"
```

9826A Controller

For the 9826A, use commands that transfer the interface messages. For example; clr7 sends DCL, trg7 sends GET, lcl701 sends GTL to the device with primary address 1, and llO7 sends LLO. The RESET statement asserts IFC.

LZ is the 494P listen address and is necessary only for addressed commands. The SIFLIN statement is used to assert IFC.

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 in the WFMPRE command in Section 5. The 494P powers up ready to transmit waveforms in ASCII (the WFMPRE command in Section 5 explains how to change modes).

4050-Series Controller

Let's define another 4050-Series User Definable key for this job. Incorporate the following input messages into the main program established under Acquiring Instrument Settings With SET? earlier in this section.

When you press User Definable key 6, the following subroutine inputs a full, 1000-point waveform with A and B memories merged (a power-up condition). Array W in this program must be dimensioned to 1000.

```
24 GOSUB 5000
25 RETURN
115 DIM W(1000)
5000 REMARK 4050-SERIES WAVEFORM INPUT
     SUBROUTINE
5010 PRINT @Z:"CURVE?"
5020 INPUT @Z:W
5030 RETURN
```

Lines 24 and 25: Subroutines called when User Definable key 6 is pressed. They jump to the subroutine that transfers the CURVE data.

Line 115: Dimensions array W to receive a full, 1000-point waveform.

Line 5010: Requests waveform (CURVE) data.

Line 5020: Ignores the ASCII characters that the 494P sends at the beginning of its CURVE? response. The INPUT statement then fills array W with the 1000 numbers transmitted by the 494P.

See Section 8 for help in plotting the waveform.

CP1100 and CP4100-Series Controllers

The following TEK SPS BASIC statements do the same job as the 4050-Series waveform INPUT subroutine.

```
5000 REMARK TEK SPS BASIC WAVEFORM INPUT
     SUBROUTINE
5010 PUT "CURVE?" INTO @N,LZ
5020 RASCII W FROM @N,TZ
5030 RETURN
```

Array W must be dimensioned using INT(1000). RASCII operates as noted for the 4050-Series INPUT statement; i.e., it ignores non-numeric characters and proceeds to fill the array when it receives numbers.

9826A Controller

Here's a self-contained program for input of a 494P waveform. So the program can operate with minimum memory, it inputs a half-resolution waveform (500 points).

```
0: wti 0,7
1: dim W[500]
2: wrt "z","wfmpre wfid:A;curve?"
3: rdb ("z")rl
4: if rl#44;rdi 4rl;rdi 4rl;jmp 0
5: wait 1
6: 1I
7: for I=1 to 500
8: red "z",A
9: AW[I]
10: next I
```

Line 2: The wfmpre command selects memory A for transfer.

Lines 3 and 4: Read the ASCII bytes in the curve? response until the comma that precedes the first number is detected.

Lines 7 through 9: The loop inputs the waveform data. Of course, you must have assigned "z" to the 494P with a device statement as explained earlier in this section under Setting Programmable Controls.

GETTING SMARTER

Signal analysis can be even easier. Put the 494P microcomputer to work to find and measure signals with its internal waveform processing capabilities. The full set of waveform processing commands is described in Section 6, and more instructions for their use are found in Section 8. To get started "getting smarter", here is a simple application.

4050-Series Controller

The following 4050-Series program catalogs the first 10 harmonics of the CAL OUT signal. If the instrument is set to other than the power-up state, precede the program with the INIT command. As in the other 4050-Series programs in this manual, Z is the variable that holds the value of the 494P GPIB address switches.

```
90 PRINT @Z:"INIT"

100 REMARK 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 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 Section 6, and instructions for putting 494P waveform processing to work are given in Section 8.

DEVICE-DEPENDENT MESSAGE STRUCTURE AND EXECUTION

INTRODUCTION

The goal of the 494P 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 unambiguous, 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.

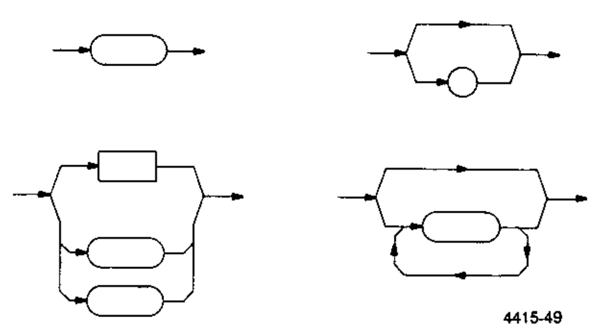
To make 494P messages easy to understand and write, ordinary engineering terms are used. Message mnemonics are chosen to be short, yet remind the user of their function. For example, to set the 494P center frequency to 500.000 MHz, the message FREQ 500.000 MHZ could be sent over the bus after the 494P 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 494P messages that makes them readable; therefore, human-oriented. The 494P device-dependent messages are upward compatible with the 492P, except as noted later under 494P/492P Compatibility.

SYNTAX DIAGRAMS

494P 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 are used to contain the mnemonics for literal (terminal entry) elements; i.e., a character or string of characters that must be sent exactly as shown. Because most mnemonics may be shortened, the command and query characters required in a 494P literal element (i.e., the first three characters of the element) are shown larger than optional characters. Although mnemonics are shown all upper case, the 494P 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. 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.



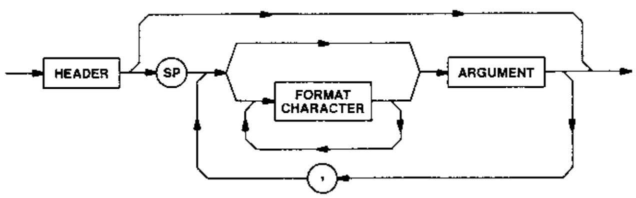
4415-49

494P INPUT MESSAGES

A remote control message to the 494P comprises one or more message units of two types. The message units either consist of commands that the 494P inputs as control or measurement data, or they consist of queries that request the 494P to output data.

One or more message units can be transmitted as a message to the 494P. Message units contain ASCII characters (binary may also be used for waveforms). The 494P accepts either upper-case or lower-case characters for the mnemonics shown in the syntax diagrams.

Input Message Format



4415-50

Message Unit Delimiter (;)

Message units are separated by the ASCII code for 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 space (SP), carriage return (CR), and line feed (LF), as well as all other ASCII control characters and comma (,). At some points in a message, the 494P may accept other non-alphanumeric characters.

Input Buffering and Execution

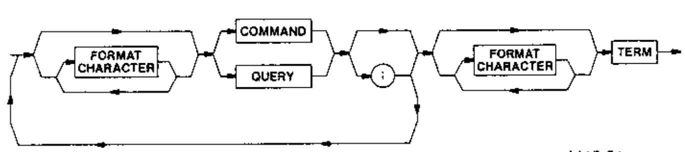
The 494P buffers each message it receives with a capacity that exceeds that required for the SET? response. The 494P waits until the end of the message to decode and execute it. A command error in any part of a message prevents its execution. When the instrument is under local control, commands that would conflict with local control are ignored (see Remote/Local under IEEE 488 Functions in Section 1).

If the message contains multiple message units, none are acted on until the 494P sees the end-of-message terminator. When the 494P sees the terminator, it executes the commands in the message in the order they were received. The 494P remains busy until it is done executing the commands in the buffer, unless the process is aborted by the DCL (Device Clear) or SDC (Selected Device Clear) interface messages. While busy, further input is not accepted (see Status Byte in Section 7 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 5, under Display Data Point Commands Interaction in Section 6, and is further expanded on under Multiple Use of Display Buffer For Waveform Processing and I/O in Section 8.

Command Format

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



4415-51

Because the general command format for display data transfers is complicated, it is omitted; see the data I/O commands in Section 5 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)

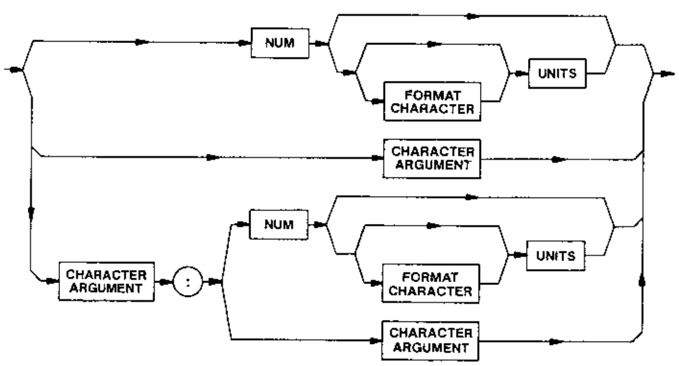
A space (SP) must separate the header from any argument(s).

Argument Delimiter (,)

A comma (,) must separate multiple arguments.

Argument Format

The following diagram illustrates that arguments following the header may be numbers, groups of characters, or either linked to a character argument.

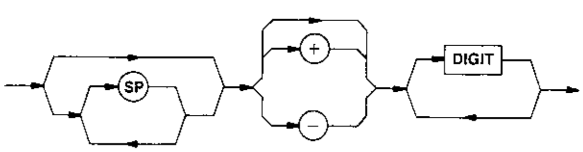


4415-52

Numbers

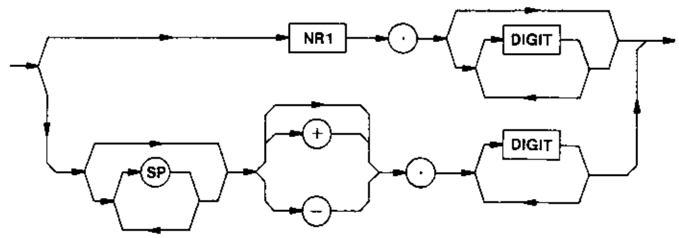
The defined element NUM is a decimal number in any of three formats; NR1, NR2, or NR3.

NR1 is an integer (no decimal point).



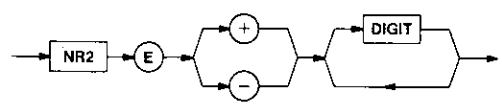
4415-53

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



4415-54

NR3 is a floating point number in scientific notation.



4415-55

NUM arguments may serve two functions. The first is to select the value of a continuous function (for example, the center frequency via FREQ). In this case, if NUM exceeds the range of the function, the 494P microcomputer does not execute the command, but issues an error message (see PEAK and POINT in Sections 4 and 6, respectively, for exceptions). 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 494P accepts arguments in engineering notation; that is, engineering units may be appended to a number argument. The 494P microcomputer 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. Although more than one format character may precede the units, only a space (SP) is shown in the command syntax figures in this manual.

Character Argument

Arguments may be either words or mnemonics. ON and OFF, for instance, are arguments for the commands that correspond to 494P front-panel pushbuttons like Vertical Display or Digital Storage.

Link Argument

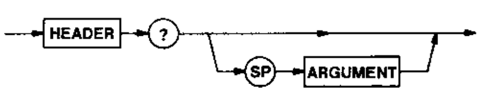
The bottom path in the argument diagram combines both character and number arguments in a link argument. The link is the colon (:), which delimits the first and second arguments.

String Argument

A string argument is used when a message is to be displayed on a printer, 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.

Query Format

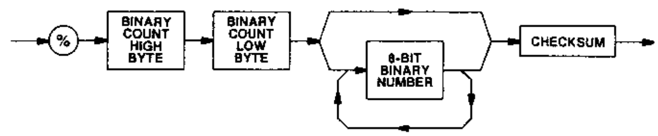
A query message unit requests either function or display data from the instrument. The query message unit format is shown below.



4415-56

Binary Block

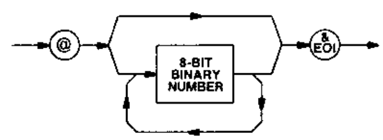
Binary block is a sequence of binary numbers that is preceded by the ASCII code for percent (%) and a two-byte binary integer representing the number of binary numbers plus one (the extra byte is the checksum) and followed by the checksum. The checksum is the 2's-complement of the modulo-256 sum of all 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.



4415-59

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.



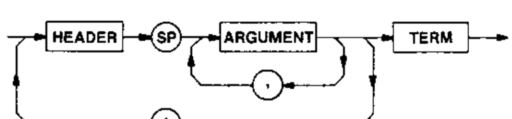
4415-58

494P OUTPUT MESSAGES

When the 494P executes a query, it buffers an output message unit that is a response to the query. Output message units contain ASCII characters (except when a binary waveform 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.



4415-57

Output Message Execution

The analyzer begins output when talked, and it continues until it reaches the end of the information in its buffer or is interrupted by a device clear (DCL), untalk (UNT), or interface clear (IFC) message. If the analyzer is interrupted and the buffer is not cleared, the analyzer will resume output if it is retalked. The buffer may be cleared by the DCL message, or if it is listened, by the SDC message or any device-dependent message. If not interrupted, the analyzer terminates the output according to the setting of the EOI OR LF switch.

494P/492P COMPATIBILITY

Most of the primary modes of the 494P controls and functions are identical to those of the 492P. Following are some of the areas where operations or results will differ.

GPIB

494P — The DCL interface message is handled by interrupts and will stop execution of the command in progress.

492P — 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 — DEGAUS may be executed in any span.

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

IDENT Command

494P — The span must be ⩽50 kHz for coaxial bands (0 - 21 GHz) or ⩽50 MHz for waveguide bands.

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

PEAK Command

494P — A PEAK value is stored for each band.

492P — The same value is used for all bands.

Readout Maximum

494P — Readout strings can contain up to 40 characters.

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

Service Requests

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

492P — 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 — Interface messages are processed despite busy status. If RTL interrupts a message, the microcomputer executes the remainder of the message before restoring local control. The response of the 494P 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 494P GPIB address, and are implemented in microcomputer firmware rather than on the interface. The speed with which these commands can be handshaked depends on how fast the microcomputer can service the resulting interrupt.

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

If the busy status occurs because the microcomputer is executing any device-dependent message other then WAIT, the response is handled the same as described for the 494P.

GET (Group Execute Trigger)

494P — GET requires microcomputer action, so handshake occurs only when the microcomputer can handle the interrupt. The effect of GET is masked by DT (Device Trigger).

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

EVENT?/ERR? Codes

494P — Bit 5 reflects the current condition, and a serial poll clears the EVENT? status that was reported. Only one Command Error is saved (i.e., the category code of the first Command Error will be reported, and any succeeding Command Errors will be ignored). All errors from other categories (Execution Errors, Internal Errors, System Events, Execution Warnings, and Internal Warnings) are saved and reported. Refer to Section 7 for information on the 494P Status and Error Reporting.

492P — Bit 5 reflects the current condition, and a serial poll clears the ERR? status that was reported. All errors, regardless of category, are saved.

Preserving Frequencies

494P — Band changes will attempt to preserve oscillator frequencies. They will set to a band limit if the oscillator frequencies would cause an out-of-band center frequency. If a 494P is not tuned after a band change and is returned to the original band, the center frequency will be the same as that before the band changes.

492P — The original center frequency will not be preserved. The 492P will just check to see if the oscillator frequencies will cause an out-of-band frequency or not.

MINATT Command

494P — MINATT always set from MAXPWR by limiting the power at the first mixer to −18 dBm.

492P — MINATT set using −10 dBm for non-preselected band and −18 dBm for the preselected bands.

Reference Level

494P — The minimum reference level is −123 dBm. The Δ-amplitude range is 63.75 dB and slides depending on the reference level when the Δ-amplitude mode is entered. The range slides from a +10 to −57.75 range to a +19 to −38.75 dB range. If the entry level is on a 10 dB step, the range is 0 to −57.75 dB. If the entry level is not on a 10 dB step, the range is (next 10 dB step-entry level) to 57.75 dB lower (e.g., entering Δ amplitude at −19 dBm will cause the range to be +9.00 to −48.75 dB).

492P — The minimum reference level is −123 dBm. The Δ-amplitude range is 63.75 dB and fixed, usually at +10 to −53.75 dB. However, the range may be 0 to −63.75 in the minimum noise mode.

RDOUT Command

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

492P — 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.

INIT Command

494P — Whenever the INIT command appears in a program, it sets TIME/DIV to the front-panel setting. If INIT is sent when TIME/DIV is at a setting that is not compatible with the following input (e.g., a high 1 μs sweep speed), the processor will recognize that an uncalibrated condition occurred and an uncal error message will be issued. Including a new TIME/DIV setting in the input will not alone eliminate this. To get around the error message, SRQ must be turned off before INIT is sent; for example,

```
RQS O; INI; TIM AUT;MIN O;ERR?;RQS 1
```

492P — If a TIME/DIV setting is included in an INIT statement, no SRQ will be issued; for example,

```
INI; TIM AUT;MIN O
```

FRONT-PANEL CONTROL

INTRODUCTION

Commands and queries for front-panel control (refer to Figure 4-1) are grouped in this section according to the following seven functions.

Frequency
Frequency span and resolution
Vertical display and reference level
Sweep control
Digital storage
Display control
General purpose

NOTE

Some of the lines of input in examples of controller programs in this section extend beyond the column width limitations. Where this occurs, the overrun information is indented on the immediately-following line.

Important—whenever a line is broken, it is always where a natural space occurs. So, be sure to add a space when inputting the program.

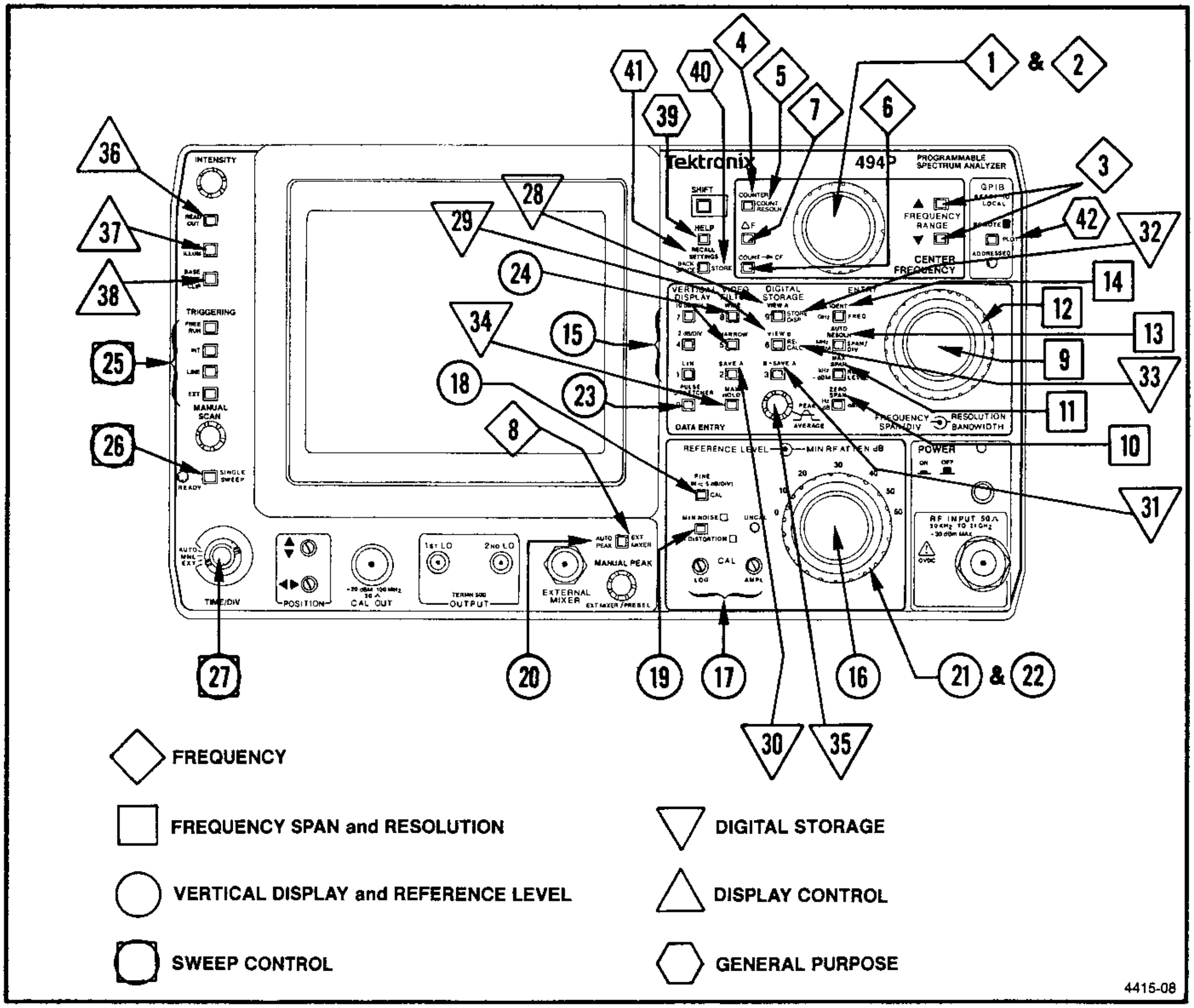


Figure 4-1. Front-panel control commands and queries.

The mnemonics in Table 4-1 correspond to the 494P commands for front-panel controls and related functions.

Table 4-1
FRONT-PANEL COMMANDS AND QUERIES

Control	Mnemonic	Identification Number[b]
Frequency		
CENTER FREQUENCY	FREQ	1
& FREQ ENTRY	TUNE	2
1ST LO[a]	FIRST	
2ND LO[a]	SECOND	
Disable Tuning Corrections (<SHIFT> FREERUN)	DISCOR	
FREQUENCY RANGE	FRQRNG	3
COUNTER	COUNT	4
COUNT RESOLN	CRES	5
COUNT→CF	CNTCF	6
ΔF	DELFR	7
Degauss[a]	DEGAUS	
EXT MIXER	EXMXR	8
Frequency Span and Resolution		
FREQUENCY SPAN/DIV & SPAN/DIV ENTRY	SPAN	9
ZERO SPAN	ZEROSP	10
MAX SPAN	MXSPN	11
RESOLUTION BANDWIDTH	RESBW	12
AUTO RESOLN	ARES	13
IDENT	IDENT	14
Vertical Display and Reference Level		
10dB/DIV 2dB/DIV LIN & dB/DIV ENTRY	VRTDSP	15
REFERENCE LEVEL & REF LEVEL ENTRY	REFLVL	16
CAL	CAL	17
FINE	FINE	18
MIN NOISE/MIN DISTORTION	RLMODE	19
MANUAL PEAK, AUTO PEAK	PEAK	20
MIN RF ATTEN dB	MINATT	21
	MAXPWR	22
PULSE STRETCHER	PLSTR	23
WIDE VIDEO FILTER NARROW VIDEO FILTER	VIDFLT	24

Table 4-1 (cont)

Control	Mnemonic	Identification Number[b]
Sweep Control		
FREE RUN INT LINE EXT	TRIG	25
SINGLE SWEEP	SIGSWP	26
TIME/DIV	TIME	27
Digital Storage		
VIEW A	AVIEW	28
VIEW B	BVIEW	29
SAVE A	SAVEA	30
B−SAVE A	BMINA	31
STORE DISP	DSTORE	32
RECALL	DRECAL	33
MAX HOLD	MXHLD	34
PEAK/AVERAGE	CRSOR	35
Display Control		
READOUT	REDOUT	36
GRAT ILLUM	GRAT	37
BASELINE CLIP	CLIP	38
General Purpose		
HELP?	HELP?	39
STORE	STORE	40
RECALL SETTINGS	RECALL	41
PLOT?	PLOT?	42
Plotter Type Selected[a]	PTYPE	
Plot (B−A)+K Formula[a]	POFSET	

[a]Command related to front-panel control functions, not actually a 494P labeled front-panel control.

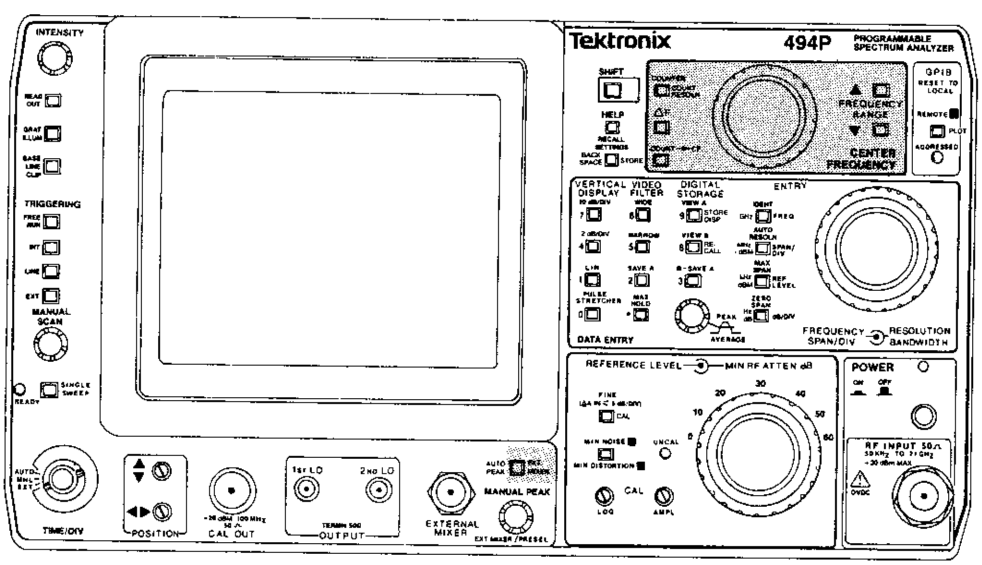
[b]Refers to Figure 4-1.

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

INTENSITY
MANUAL SCAN
POSITION
AMPL and LOG CAL
POWER
PEAK/AVERAGE cursor; other than fully counter-clockwise or clockwise positions.

FREQUENCY (Figure 4-2)

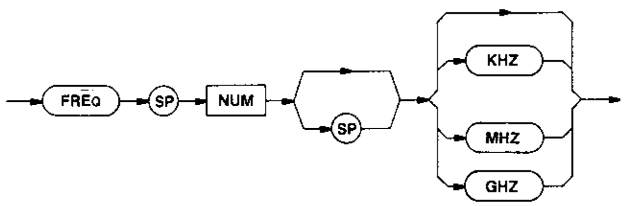
The commands in this group set and change the 494P center frequency (FREQ and TUNE), set the 1st LO (FIRST) and the 2nd LO (SECOND) frequencies, disable frequency corrections (DISCOR), select the frequency range (FRQRNG), turn the counter mode on and off (COUNT), select counter resolution (CRES), transfer signal count to center frequency (CNTCF), and activate the Δ-frequency function (DELFR). Degaussing current (DEGAUS) can be applied to restore preselector alignment. Another command (EXMXR) selects the EXT MIXER input.



4415-09

Figure 4-2. Front-panel Frequency Control commands.

FREQ (center frequency) command



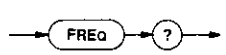
4415-90

NUM — The analyzer centers its span about the value in the command argument. If that frequency is not within the current band, the analyzer selects the nearest band to the current band that encompasses the value. The range of values and resolution of the instrument response are the same as for front-panel operation.

Power-up value — 0 MHz.

Interaction — An automatic degauss is done when FREQ is changed >1 GHz to improve amplitude accuracy.

FREQ (center frequency) query

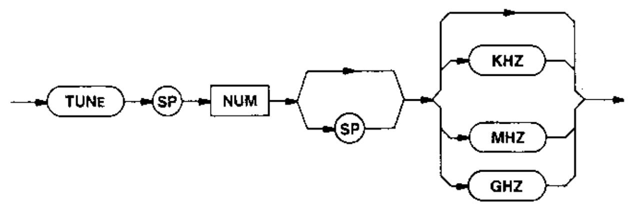


4415-91

Response to FREQ query

4415-92

TUNE (incremental frequency change) command

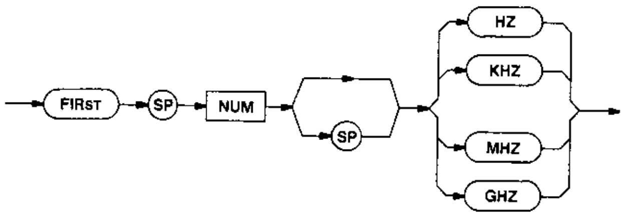


4415-93

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

There is no TUNE query.

FIRST (1st LO frequency) command



4415-94

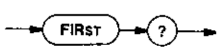
NOTE

The FIRST and SECOND commands are included primarily to allow the SET? response to exactly re-create the 494P settings.

NUM — The 494P 1st LO is set to the requested frequency. The resulting center frequency will be displayed.

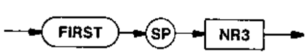
Power-up value — 2072 MHz.

FIRST (1st LO frequency) query



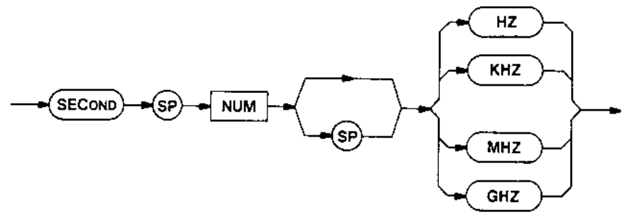
4415-95

Response to FIRST query



4415-96

SECOND (2nd LO frequency) command



4415-97

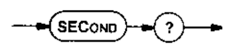
NOTE

The FIRST and SECOND commands are included primarily to allow the SET? response to exactly re-create 494P settings.

NUM — The 494P 2nd LO is set to the requested frequency. The resulting center frequency will be displayed.

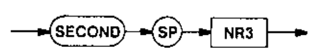
Power-up value — 2182 MHz.

SECOND (2nd LO frequency) query



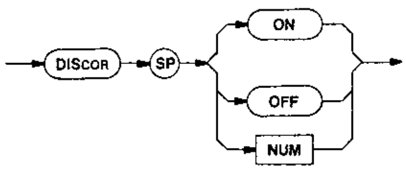
4415-98

Response to SECOND query



4415-99

DISCOR (disable frequency corrections) command



4415-100

This command allows the 494P frequency control loop to be disabled for diagnostic purposes. It also allows a fallback to low accuracy center frequency operation if the frequency control loop fails.

ON — Center frequency corrections are disabled.

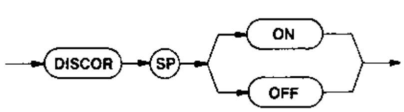
OFF — Center frequency corrections are enabled.

NUM — 1 equals ON; numbers $\geqslant +0.5$ are rounded to 1. Number 0 equals OFF; numbers less than $+0.5$ are rounded to 0.

DISCOR (disable frequency corrections) query

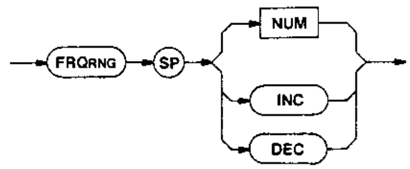
4415-101

Response to DISCOR query



4415-102

FRQRNG (frequency range) command



4415-103

NUM — The analyzer 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 analyzer maintains its current frequency range and reports an error.

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

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

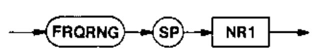
Power-up value — Frequency Range 1.

Interaction — The 494P microcomputer automatically selects the lowest frequency range that encompasses the frequency setting that responds to the FREQ command.

FRQRNG (frequency range) query

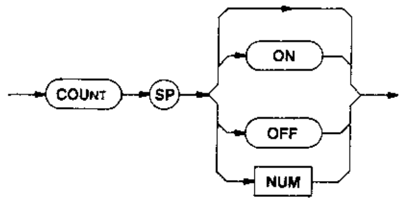
4415-104

Response to FRQRNG query



4415-105

COUNT (counter) command



4415-106

ON — The counter mode is turned on.

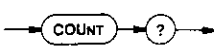
OFF — The counter mode is turned off.

NUM — 1 equals ON; numbers ⩾+0.5 are rounded to 1. Number 0 equals OFF; numbers less than +0.5 are rounded to 0.

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.

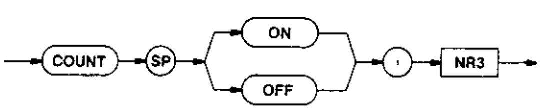
Power-up value — Off.

COUNT (counter) query



4415-107

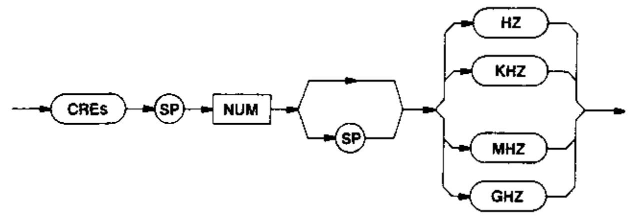
Response to COUNT query



4415-108

The number returned in this response is the result of the last count, regardless of whether COUNT is ON or OFF.

CRES (counter resolution) command



4415-109

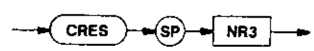
NUM — The proper decade of counter resolution is selected for use. Numbers that are not powers of ten will be set to the next lower power of ten, up to a maximum of 1 GHz resolution.

Power-up value — 1 Hz.

CRES (counter resolution) query

4415-110

Response to CRES query



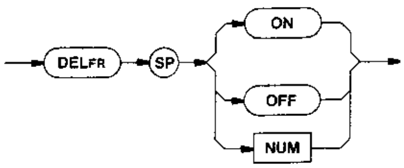
CNTCF (count to center frequency) command

4415-111

A count of the signal is taken at center screen, then this signal count result is transferred to the center frequency. This tunes the 494P to the signal counted. Accuracy is limited by the count resolution in use when the signal count is done.

There is no CNTCF query.

DELFR (Δ-frequency) command



4415-112

ON — The Δ-frequency function is turned on. As the frequency is changed, the crt 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 the DELFR mode was turned on.

OFF — The Δ-frequency function is turned off.

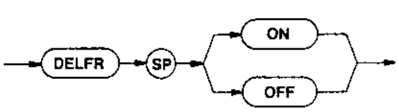
NUM — 1 equals ON; numbers ≥+0.5 are rounded to 1. Number 0 equals OFF; numbers less than +0.5 are rounded to 0.

Power-up value — Off.

DELFR (Δ-frequency) query

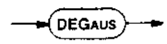
4415-113

Response to DELFR query



4415-114

DEGAUS (degauss tuning coils) command

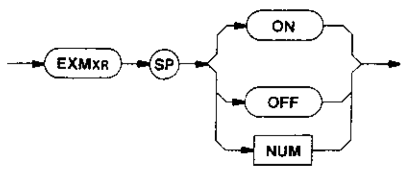


4415-115

A degaussing current is turned on to remove residual magnetism in the 1st LO and preselector tuning coils. This improves preselector tracking amplitude accuracy when the preselector is not PEAK'd at each frequency. This function is performed automatically by the 494P when FRQRNG is changed and when CENTER FREQUENCY is changed by more than 1 GHz.

There is no DEGAUS query.

EXMXR (external mixer input) command



4415-116

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

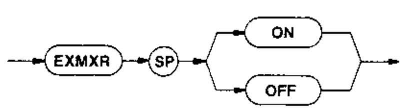
OFF — The coax RF INPUT is selected.

NUM — 1 equals ON; numbers ≥ +0.5 are rounded to 1. Number 0 equals OFF; numbers less than +0.5 are rounded to 0.

Power-up value — Off.

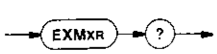
Interaction — The EXT MIXER input is automatically selected for the waveguide bands (FRQRNG 6 and above) and cannot be defeated by EXMXR OFF. When active, this input bypasses the input attenuator, so it is up to the operator to prevent input overload.

EXMXR (external mixer input) query



4415-118

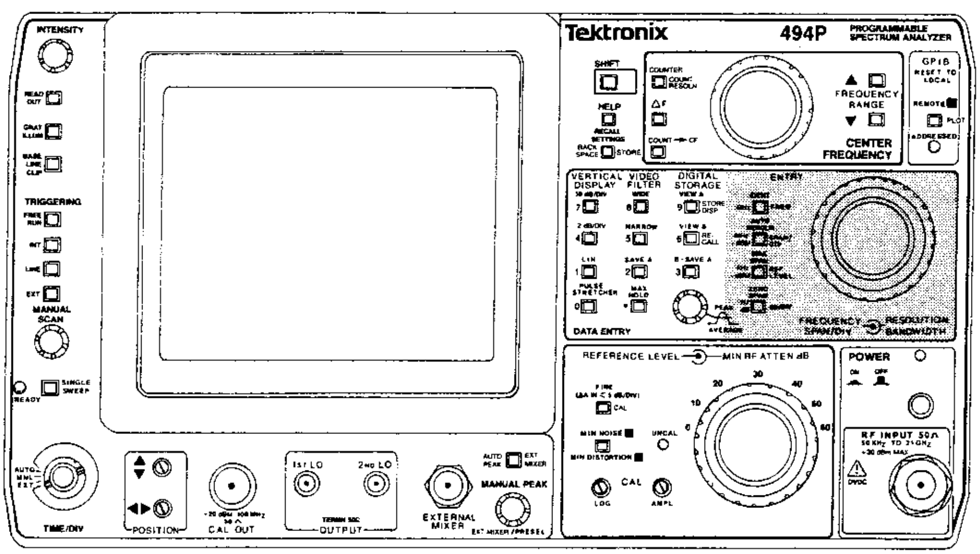
Response to EXMXR query



4415-117

FREQUENCY SPAN AND RESOLUTION (Figure 4-3)

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).



4415-10

Figure 4-3. Front-panel Frequency Span and Resolution commands.

AUTO — Auto resolution is selected (equivalent to ARES ON).

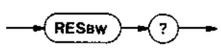
INC — The next larger step is selected (if possible).

DEC — The next smaller step is selected (if possible).

Power-up value — 1 MHz.

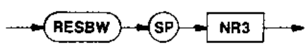
Interaction — Any argument except AUTO cancels ARES ON. Reducing resolution bandwidth may require a slower sweep speed (TIME).

RESBW (resolution bandwidth) query



4415-129

Response to RESBW query



4415-130

The response to the SET query includes the AUTO argument (see SET? under System Commands And Queries, Section 7).

VERTICAL DISPLAY AND REFERENCE LEVEL (Figure 4-4)

The commands in this group control the vertical scale factor (VRTDSP) and reference level (REFLVL and FINE) of the display. The microcomputer automatically selects the gain distribution (combination of RF attenuation and IF gain) according to the reference level mode (RLMODE); this takes into account the least amount of RF attenuation (MINATT) allowed or maximum power (MAXPWR) expected. The microcomputer also automatically selects the peak (PEAK) analyzer response. The pulse stretcher (PLSTR) stretches narrow or pulsed signals for acquisition or 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).

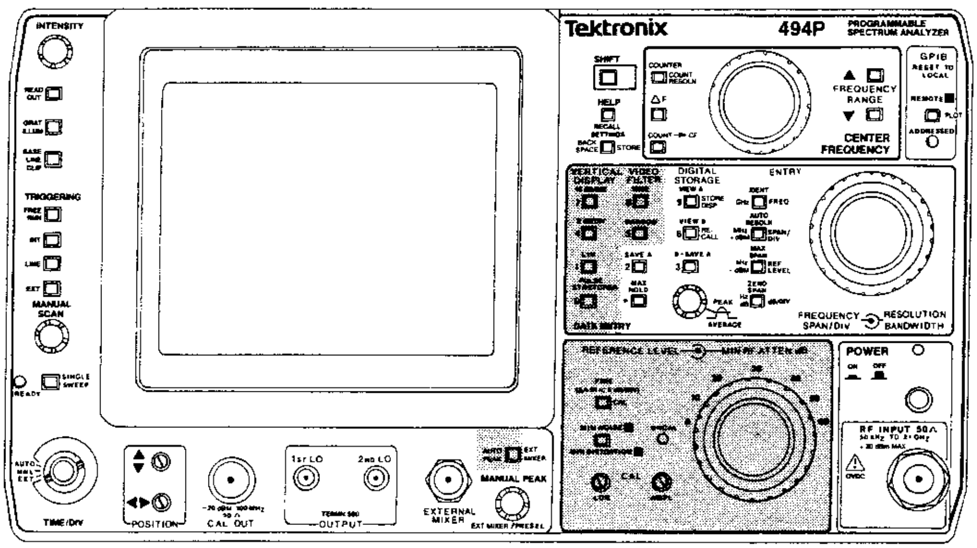
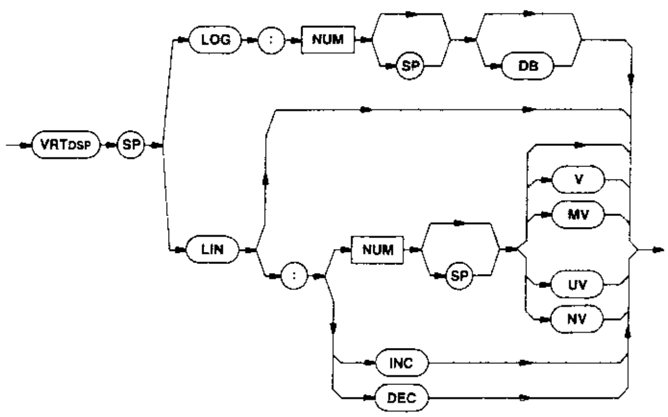


Figure 4-4. Front-panel Vertical Display and Reference Level commands.

VRTDSP (vertical display) command



4415-137

LOG — The display is scaled to the dB/div specified by integers in the range 1 to 15 (non-integers are rounded). Values outside this range cause the 494P to report an error.

LIN — The display is scaled in volts/div. 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/div 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).

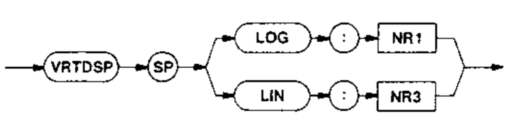
Power-up value — 10 dB/division.

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

VRTDSP (vertical display) query

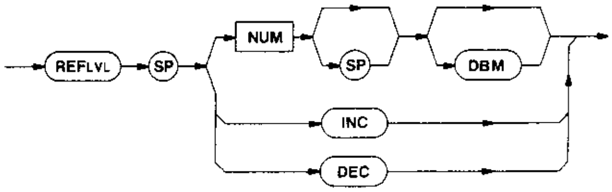
4415-138

Response to VRTDSP query



4415-139

REFLVL (reference level) command



4415-140

NUM — The analyzer sets the reference level to the nearest dBm step for a log vertical display (except in the Δ A mode) and to the nearest dBm step for a linear vertical display. The argument to the REFLVL command is always an absolute reference level and not an offset to the present reference level, though the crt readout shows relative amplitude.

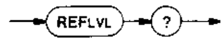
INC or DEC — The reference level is stepped up or down once. The step value is determined by the value of the VRTDSP scale factor and FINE selection (refer to Table 4-3).

Table 4-3
REFERENCE LEVEL SETTINGS

	Step Size	
VERTDSP Scale Factor	**FINE ON**	**FINE OFF**
15 dB	1 dB	Equal to VRTDSP scale factor
14 dB	1 dB	Equal to VRTDSP scale factor
13 dB	1 dB	Equal to VRTDSP scale factor
12 dB	1 dB	Equal to VRTDSP scale factor
11 dB	1 dB	Equal to VRTDSP scale factor
10 dB	1 dB	Equal to VRTDSP scale factor
9 dB	1 dB	Equal to VRTDSP scale factor
8 dB	1 dB	Equal to VRTDSP scale factor
7 dB	1 dB	Equal to VRTDSP scale factor
6 dB	1 dB	Equal to VRTDSP scale factor
5 dB	1 dB	Equal to VRTDSP scale factor
4 dB	Δ A mode (0.25 dB)	Equal to VRTDSP scale factor
3 dB	Δ A mode (0.25 dB)	Equal to VRTDSP scale factor
2 dB	Δ A mode (0.25 dB)	1 dB
1 dB	Δ A mode (0.25 dB)	1 dB
LIN	1 dB	Either 6 dB or 8 dB (varies to match 1-2-5 volts/div sequence)

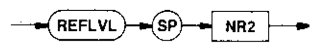
Power-up value — +30 dBm.

REFLVL (reference level) query



4415-141

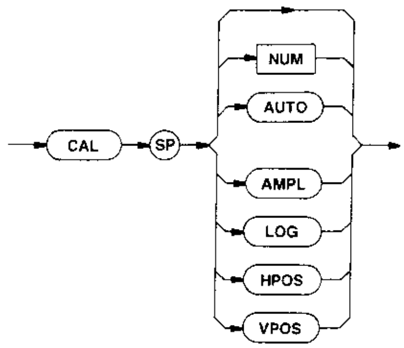
Response to REFLVL query



4415-142

The value returned is the absolute reference level, whether or not in the Δ A mode.

CAL (cal) command



4415-143

To request the 494P to perform an internal calibration, input the following program.

```
RQS O;CAL;ERR?;RQS 1
```

NUM — 0 equals AUTO, 1 equals LOG, 2 equals AMPL, 3 equals HPOS, and 4 or higher equals VPOS.

CAL (without arguments) or **CAL AUTO** — The resolution bandwidth filter frequencies are calibrated with respect to 10 MHz, and levels relative to the 1 MHz filter level (within a range of +2, −4 dB). During operation, the word CALIBRATING appears on the screen.

CAL LOG — The instrument is set up to enable the operator to adjust the front-panel LOG CAL control. CAL LOG has an indeterminate execution time, and will operate until either a device clear (DCL) is received via the GPIB port or the 494P is returned to local control via the instrument front panel. An instruction message appears on the screen.

CAL AMPL — The instrument is set up to enable the operator to adjust the front-panel AMPL CAL control. CAL AMPL has an indeterminate execution time, and will operate until either a device clear (DCL) is received via the GPIB port or the 494P is returned to local control via the instrument front panel. An instruction message appears on the screen.

CAL HPOS — The instrument is set up to enable the operator to adjust the front-panel horizontal POSITION control. Horizontal POSITION has an indeterminate execution time, and will operate until either a device clear (DCL) is received via the GPIB port or the 494P is returned to local control via the instrument front panel. An instruction message appears on the screen.

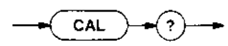
CAL VPOS — The instrument is set up to enable the operator to adjust the front-panel vertical POSITION control. Vertical POSITION has an indeterminate execution time, and will operate until either a device clear (DCL) is received via the GPIB port or the 494P is returned to local control via the instrument front panel. An instruction message appears on the screen.

Power-up value — Off.

Interaction — Due to the internal procedures required to perform CAL, CAL AUTO, CAL LOG, and CAL AMPL, an error message (ERR 52 or ERR 54) will be delivered upon completion of these operations. To eliminate the message, input the following program.

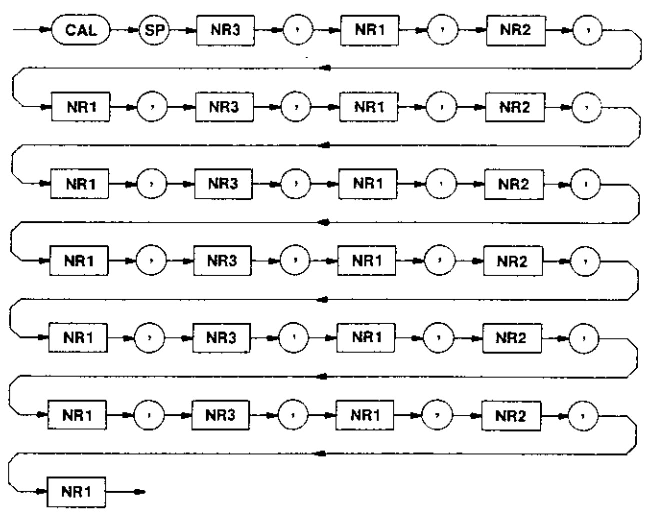
```
RQS O;CAL;ERR?;RQS 1
```

CAL (cal) query



4415-144

Response to CAL query



4415-145

In the CAL? response, the same data is given in succession for the 1 MHz, 100 kHz, 10 kHz, 1 kHz, 100 Hz, and 30 Hz filters (in that order). The center frequency and reference level accuracy calibration adjustment sequence is amplitude calibration, horizontal position, vertical position, and log calibration. The data given for each filter is the frequency error, the frequency cal code, the level error, and the level cal code.

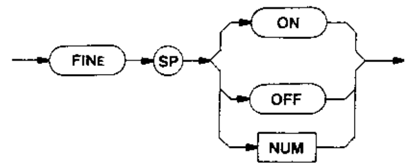
The frequency error is the difference between the measured filter frequency and 10 MHz, expressed in Hz. The level error is the difference between the measured filter level and the measured level of the 1 MHz filter, expressed in dB.

Use Table 4-4 to decode the calibration code numbers.

Table 4-4
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).
1	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 level 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 level calibration only.)
4	The last calibration attempt for this item succeeded.

FINE (fine reference level steps) command



4415-146

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 Δ A mode.

Δ A Mode — The Δ A 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. 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 over a range of at least 0 dB to −48 dB from the initial level, without an overload to the analyzer input. This readout is available with UPRDO?. REFLVL? returns the absolute reference level.

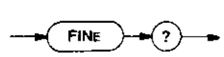
OFF — Normal steps are restored for reference level changes, which cancels the Δ A mode (if active).

NUM — 1 equals ON; numbers ⩾ +0.5 are rounded to 1. Number 0 equals OFF; numbers less than +0.5 are rounded to 0.

Power-up value — Off.

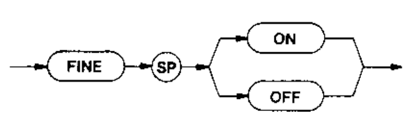
Interaction — This command, along with VRTDSP, controls the analyzer response to REFLVL INC or DEC.

FINE (fine reference level steps) query



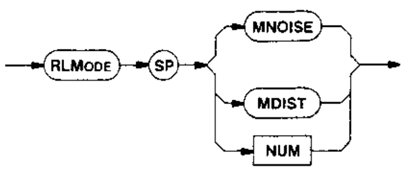
4415-147

Response to FINE query



4415-148

RLMODE (reference level mode) command



4415-149

MNOISE — The microcomputer 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).

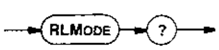
MDIST — Normal RF attenuation is requested for a given reference level. Generally, this yields 10 dB more RF attenuation than the MNOISE argument and results in lower signal levels in the analyzer, hence less distortion.

NUM — 0 equals MNOISE, 1 equals MDIST, and other numbers are rounded to 0 or 1.

Power-up value — MDIST

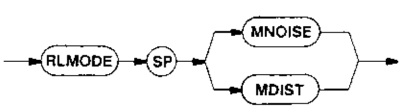
Interaction — This command affects the gain distribution obtained with the REFLVL command (see also MINATT and MAXPWR).

RLMODE (reference level mode) query



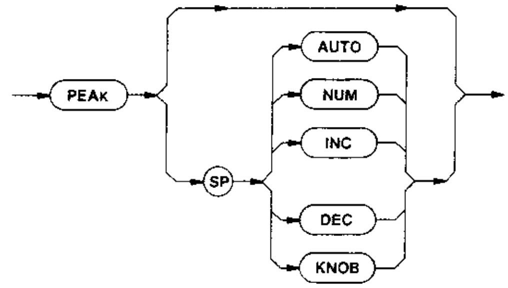
4415-150

Response to RLMODE query



4415-151

PEAK (peaking) command



4415-152

AUTO — During several sweeps, the 494P microcomputer automatically tunes the PEAK control to peak the largest signal in a window ±1 division around the display data point. The peak code that results is stored in battery-powered RAM, with a unique code retained for each band (with the exception of the lowest frequency band, in which peaking is not used). These codes are used whenever AUTO PEAK is on. Thus, once a signal is peaked in a band, signals at other frequencies in that band will be within a few dB of being peaked as well. If a signal is not found within the ±1 division window, the previously-acquired peaking code stored in battery-powered RAM is used. End-of-sweep interrupts are not issued and triggering, time/division, and the max hold digital storage parameters may be changed by the 494P while PEAK is being executed. The previous parameters are restored when execution is complete. Although this command uses digital storage, it does not overwrite the A portion if SAVEA is ON. The AUTO PEAK light is turned on.

NUM — Numbers in the range 0 to 1023 set the value of the PEAK control, and are stored in battery-powered memory. Non-integers or numbers outside this range are rounded to the nearest integer in the range; no warning is issued.

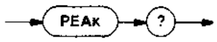
INC or DEC — The value of PEAK is changed ±1 from its current value, and is stored in battery-powered RAM.

KNOB — The front-panel MANUAL PEAK control is active. AUTO PEAK is turned off so the operator can manually peak the analyzer's response from the front panel.

The omission of the argument is equivalent to AUTO PEAK.

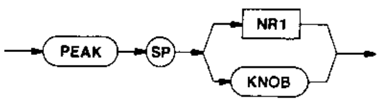
Power-up value — KNOB (AUTO PEAK off, MANUAL PEAK on).

PEAK (peaking) query



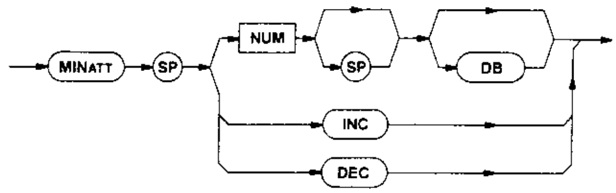
4415-153

Response to PEAK query



4415-154

MINATT (minimum RF attenuation) command



4415-155

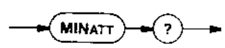
NUM — The gain distribution set by the microcomputer is limited; RF attenuation may not be reduced below the attenuator step in the number argument. If NUM is not an even decade from 0 to 60, the next higher step (0, 10, 20, ... 60) is selected.

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

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

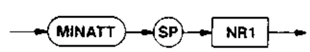
Interaction — The range of RF attenuation is limited in response to the REFLVL command, which limits the range of the REFLVL command. The previous limit set by either MINATT or MAXPWR is cancelled.

MINATT (minimum RF attenuation) query



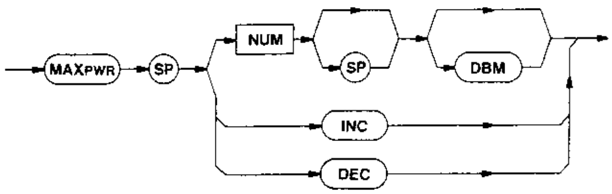
4415-156

Response to MINATT query



4415-157

MAXPWR (maximum input power) command



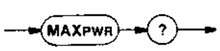
4415-158

NUM — This is an input to a 494P microcomputer algorithm that protects the RF INPUT from overload at the expected maximum power level. The microcomputer selects a minimum RF attenuation so that the NUM signal level is reduced to −18 dBm at the 1st Mixer. (This is the analyzer's 1 dB compression point.) The maximum nondestructive power level that can be connected to the RF INPUT is +30 dBm.

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

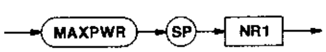
Interaction — The range of RF attenuation is limited in response to the REFLVL command, which limits the range of the REFLVL command. MAXPWR cancels the previous limit set by either MINATT or MAXPWR.

MAXPWR (maximum input power) query



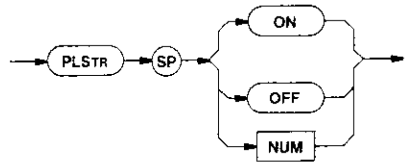
4415-159

Response to MAXPWR query



4415-160

PLSTR (pulse stretcher) command



4415-161

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 or 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.

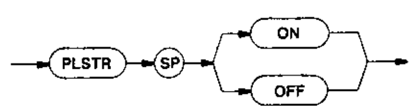
NUM — 1 equals ON; numbers $\geq +0.5$ are rounded to 1. Number 0 equals OFF; numbers less than +0.5 are rounded to 0.

Power-up value — Off.

PLSTR (pulse stretcher) query

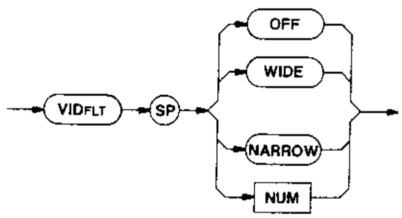
4415-162

Response to PLSTR query



4415-163

VIDFLT (video filter) command



4415-164

OFF — Both 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.

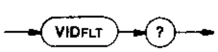
NARROW — The effect of wide filter is magnified by a factor of 10. The narrow video filter reduces video bandwidth to about 1/300 of the selected resolution bandwidth.

NUM — 0 equals OFF; numbers less than +0.5 are rounded to 0. Number 1 equals WIDE; numbers $\geq +0.5$ are rounded to 1. Number 2 equals NARROW; numbers $\geq +1.5$ are rounded to 2.

Power-up value — Off.

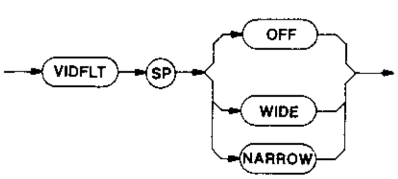
Interaction — It may be necessary to reduce sweep speed (TIME), because the analyzer's overall bandwidth is reduced by video filtering.

VIDFLT (video filter) query



4415-165

Response to VIDFLT query



4415-166

SWEEP CONTROL (Figure 4-5)

Three commands control the 494P 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 auto sweep directs the 494P to automatically match the sweep to related analyzer parameters. Other options include manual or external analog control of the sweep.

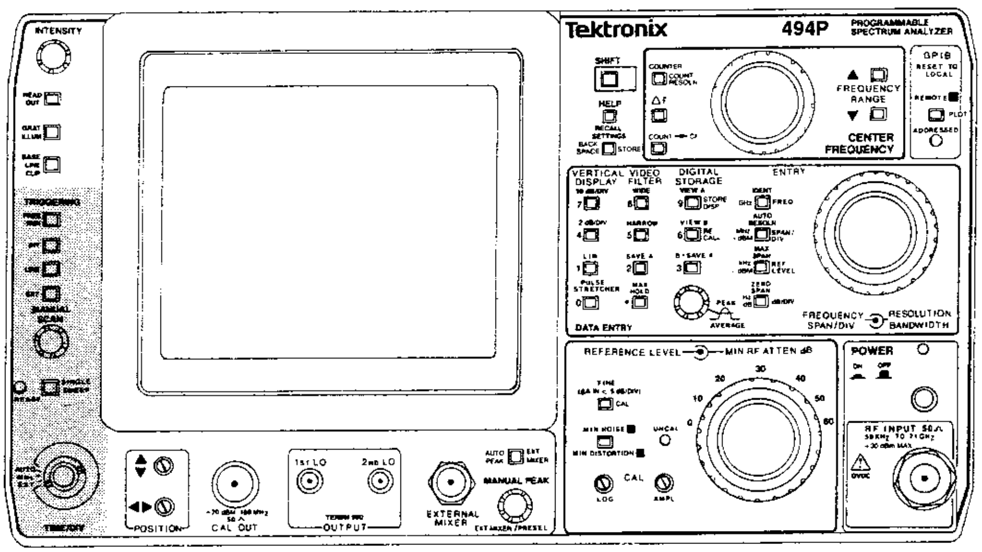
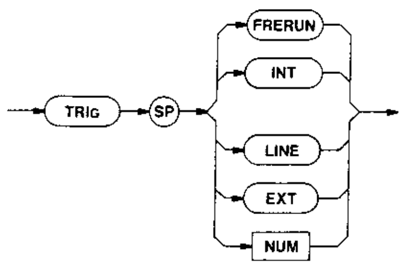


Figure 4-5. Front-panel Sweep Control commands.

TRIG (triggering) command



4415-167

FRERUN — The analyzer sweep is allowed to run repetitively. A trigger is not required (and is ignored), so the analyzer 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 analyzer generates a sweep only when it is triggered by an input signal at its center frequency. 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 zero-span mode (ZEROSP).

LINE — The power line input is selected as the trigger signal (useful in both the frequency and time domain modes for signals with components related to the power line frequency).

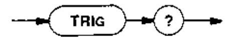
EXT — The sweep is triggered for pulsed signals by a signal with an amplitude of at least +1.0 V peak connected to EXT IN TRIG on the 494P rear panel.

NUM — 0 equals FRERUN, 1 equals INT, 2 equals LINE, and 3 equals EXT. Numbers not equal to these values are rounded to the nearest valid integer.

Power-up value — Free-run.

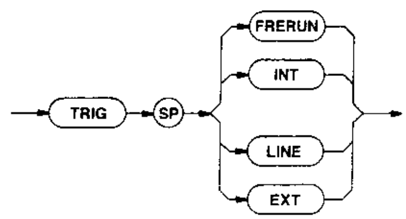
Interaction — The signal frequency required for internal trigger is related to the center frequency. In the frequency domain mode, the required frequency corresponds to 1/2 division to the left of the left graticule edge; in the time domain mode, the required frequency is the center frequency. In the frequency domain mode, the required frequency must be within the selected band (e.g., ≥1.7 GHz in band 2).

TRIG (triggering) query



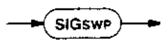
4415-168

Response to TRIG query



4415-169

SIGSWP (single-sweep) command



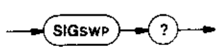
4415-170

On the first SIGSWP command, the analyzer enters the single-sweep mode, which aborts 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 analyzer makes a single sweep of the selected spectrum when the conditions determined by the TRIG command are met. Correction of the center frequency is done periodically and when SIGSWP is executed (before the sweep is armed).

Power-up value — Off.

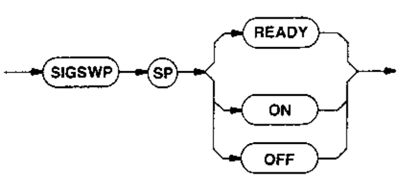
Interaction — Any TRIG command cancels the single-sweep mode.

SIGSWP (single-sweep) query



4415-171

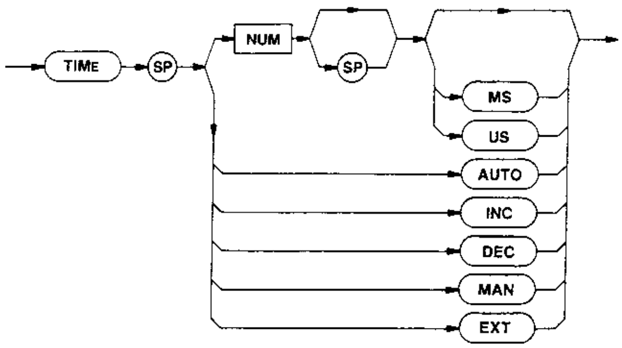
Response to SIGSWP query



4415-172

The response to the SET query is omitted if single-sweep is not active (see SET? under System Commands And Queries, Section 7).

TIME (time/div) command



4415-173

NUM — 1-2-5 sequence in the range 200^{6} to 10. Numbers not in this sequence are rounded to the nearest step.

AUTO — The 494P microcomputer 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 the operator can manually scan the spectrum. As the control is rotated, the horizontal position of the crt beam and the analyzer front-end tuning are varied from the center of the sweep and the center of the selected spectrum. Correction of the center frequency is done periodically.

EXT — The sweep is coupled to EXT IN HORIZ on the 494P rear panel. The horizontal position of the crt beam and the analyzer front-end tuning are varied by an external signal. A signal in the range 0 to +10 V scans the spectrum. Correction of the center frequency is done periodically.

Power-up value — TIME/DIV control setting.

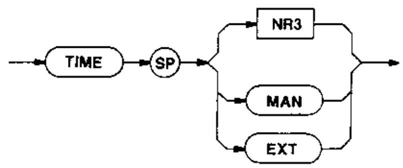
Interaction — Too fast a sweep speed for a given resolution bandwidth will uncalibrate the display. For digital storage to properly acquire spectrum data, 2 ms/div is the maximum usable sweep rate.

TIME (time/div) query

4415-174

Response to TIME query

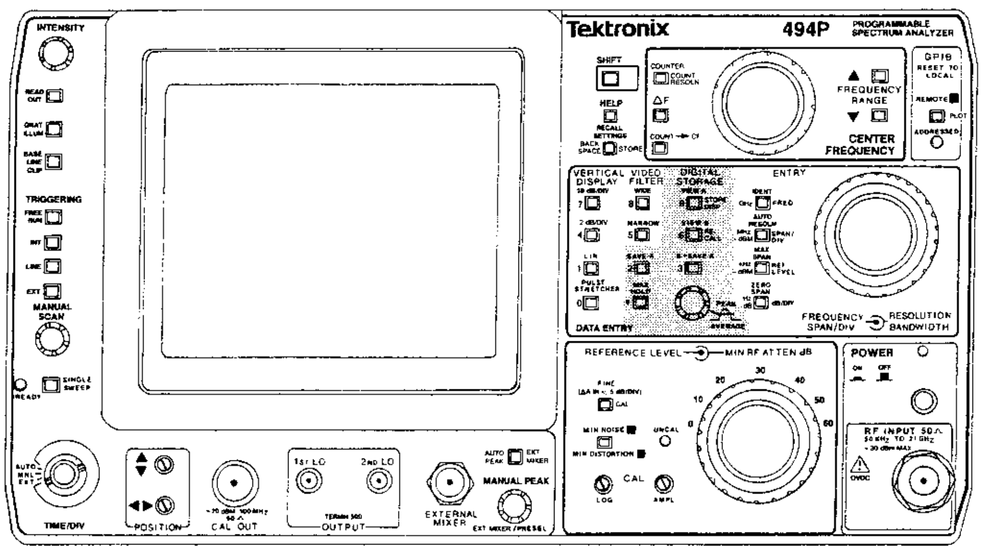
The SET? response includes AUTO as a possible argument (see SET? under System Commands And Queries, Section 7).



4415-175

DIGITAL STORAGE CONTROL (Figure 4-6)

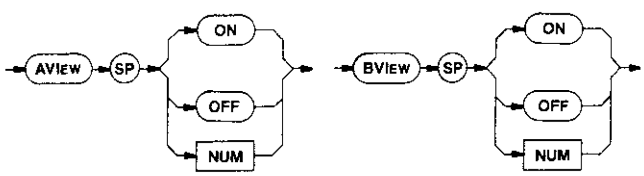
These commands control the 494P digital storage functions of display (AVIEW, BVIEW), updating (SAVEA), display and updating (BMINA), display storage (DSTORE), recalling the display (DRECAL), and digitizer control (MXHLD, CRSOR).



4415-13

Figure 4-6. Front-panel Digital Storage Control commands.

AVIEW and BVIEW (A and B waveform display) commands



4415-176

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

OFF — The display of the specified waveform is turned off. (Refer to the ON description for operation with SAVEA off.) If both waveforms are turned off, the input signal is displayed in real time.

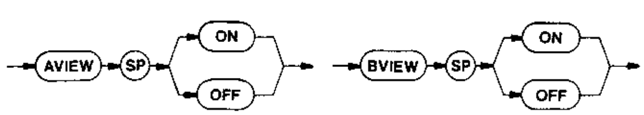
NUM — 1 equals ON; numbers ≥ +0.5 are rounded to 1. Number 0 equals OFF, numbers less than +0.5 are rounded to 0.

Power-up value — Both A and B on.

AVIEW and BVIEW (A and B waveform display) queries

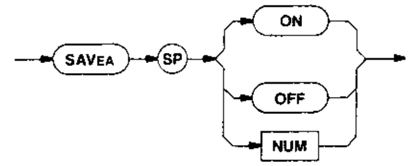
4415-177

Response to AVIEW and BVIEW queries



4415-178

SAVEA (save A waveform) command



4415-179

ON — The A waveform updating is discontinued 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 analyzer settings if only AVIEW is on (both BVIEW and BMINA off).

OFF — The A waveform updating is resumed.

NUM — 1 equals ON; numbers ≥ +0.5 are rounded to 1. Number 0 equals OFF; numbers less than +0.5 are rounded to 0.

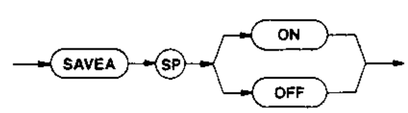
Power-up value — Off.

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

SAVEA (save A waveform) query

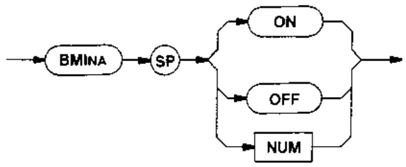
4415-180

Response to SAVEA query



4415-181

BMINA (B−A waveform display) command



4415-182

ON — The 494P microcomputer turns on SAVEA 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 internally.

OFF — The difference display is turned off.

NUM — 1 equals ON; numbers ≥ +0.5 are rounded to 1. Number 0 equals OFF; numbers less than +0.5 are rounded to 0.

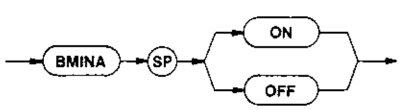
Power-up value — Off.

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

BMINA (B−A waveform display) query

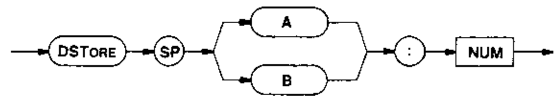
4415-183

Response to BMINA query



4415-184

DSTORE (store display) command



4415-185

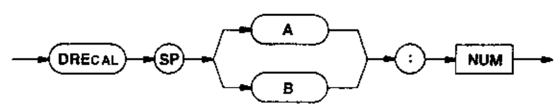
The readout associated with the display is stored with the display.

A — The A waveform is stored in the battery-powered memory indicated by NUM (range: 0-8).

B — The B waveform is stored in the battery-powered memory indicated by NUM (range: 0-8).

There is no DSTORE query.

DRECAL (recall display) command



4415-186

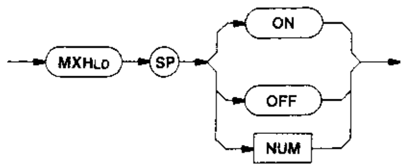
A — A waveform from the memory specified by NUM (0-8) is recalled and put in the A waveform display.

B — A waveform from the memory specified by NUM (0-8) is recalled and put in the B waveform display. If BVIEW or BMINA is on, the readout associated with a recalled B waveform is displayed. If AVIEW is on, and BVIEW and BMINA are OFF, the readout associated with a recalled B waveform is displayed. In all other cases, the current instrument readout is displayed.

There is no DRECAL query.

Interaction — DRECAL turns SAVEA ON. If SIGSWP is off, the current instrument sweep readout is displayed.

MXHLD (max hold) command



4415-187

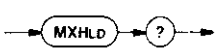
ON — Digital storage holds the maximum value obtained for each point in both the A and B waveforms; a point is updated only if the new value is greater than the current value. The A waveform is not affected if SAVEA is on.

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

NUM — 1 equals ON; numbers $\geq +0.5$ are rounded to 1. Number 0 equals OFF; numbers less than +0.5 are rounded to 0.

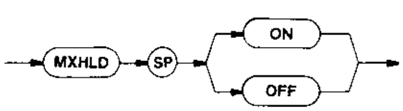
Power-up value — Off.

MXHLD (max hold) query



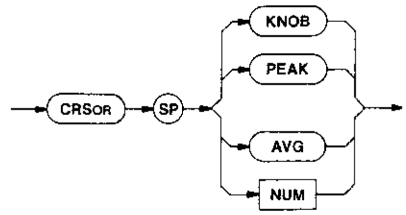
4415-188

Response to MXHLD query



4415-189

CRSOR (peak/average cursor) command



4415-190

KNOB — The PEAK/AVERAGE knob is under local control. The operator 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 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.

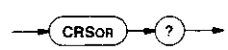
AVG — Average values are used to update the waveforms, regardless of the cursor position last set by KNOB. PEAK AVG is the same as if the cursor is set to its highest (maximum) position.

NUM — 0 equals KNOB; numbers less than +0.5 are rounded to 0. 1 equals PEAK; numbers $\geq +0.5$ and less than +1.5 are rounded to 1. Number 2 equals AVG; numbers $\geq +1.5$ are rounded to 2.

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

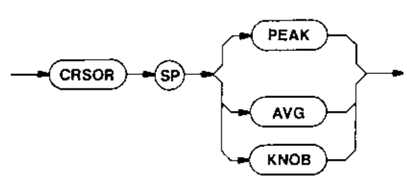
Power-up value — Knob.

CRSOR (peak/average cursor) query



4415-191

Response to CRSOR query



4415-192

DISPLAY CONTROL (Figure 4-7)

These commands control the 494P crt display functions to display the readout (REDOUT), light the graticule (GRAT), and clip the baseline trace (CLIP). Refer to Table 4-5. Each function can be turned on or off and queried.

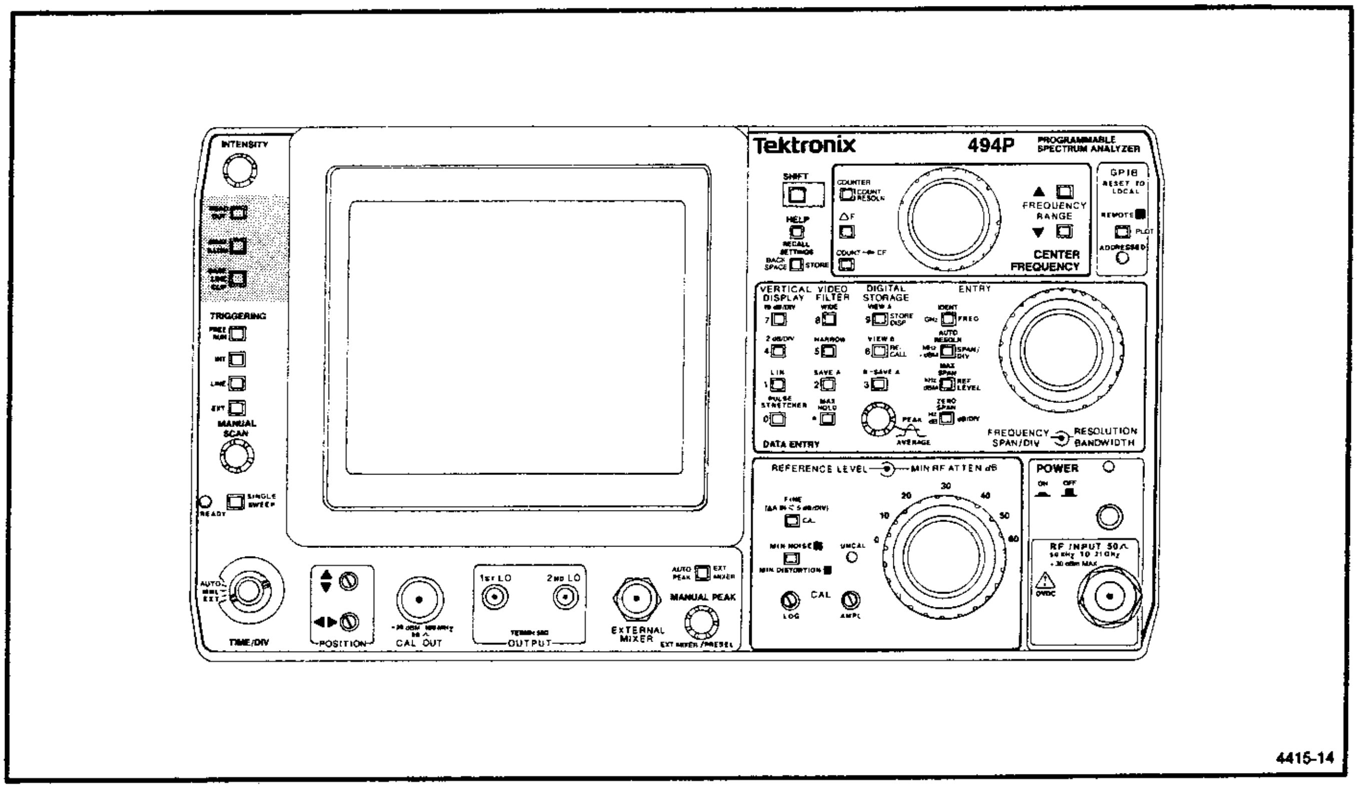
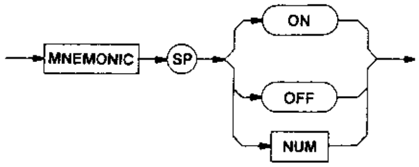


Figure 4-7. Front-panel Display Control commands.

Display control command



4415-193

Table 4-5
DISPLAY CONTROL

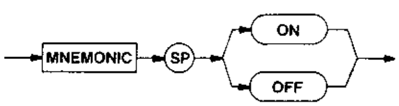
Mnemonic	ON	OFF	Power-up Value
REDOUT	Display instrument control characters	Blank readout	On
GRAT	Lighted graticule	Dark graticule	Off
CLIP	Blank trace at bottom of crt	Full trace	Off

NUM — 1 equals ON; numbers ≥ +0.5 are rounded to 1. 0 equals OFF; numbers less than +0.5 are rounded to 0.

Display control query

4415-194

Response to display control query



4415-195

GENERAL PURPOSE (Figure 4-8)

The general purpose commands and queries request the command list or front-panel help messages (HELP?), store settings in memory (STORE), recall settings from memory (RECALL), plot crt information (PLOT) on a choice of plotters (PTYPE), and plot the (B−A)+K formula (POFSET).

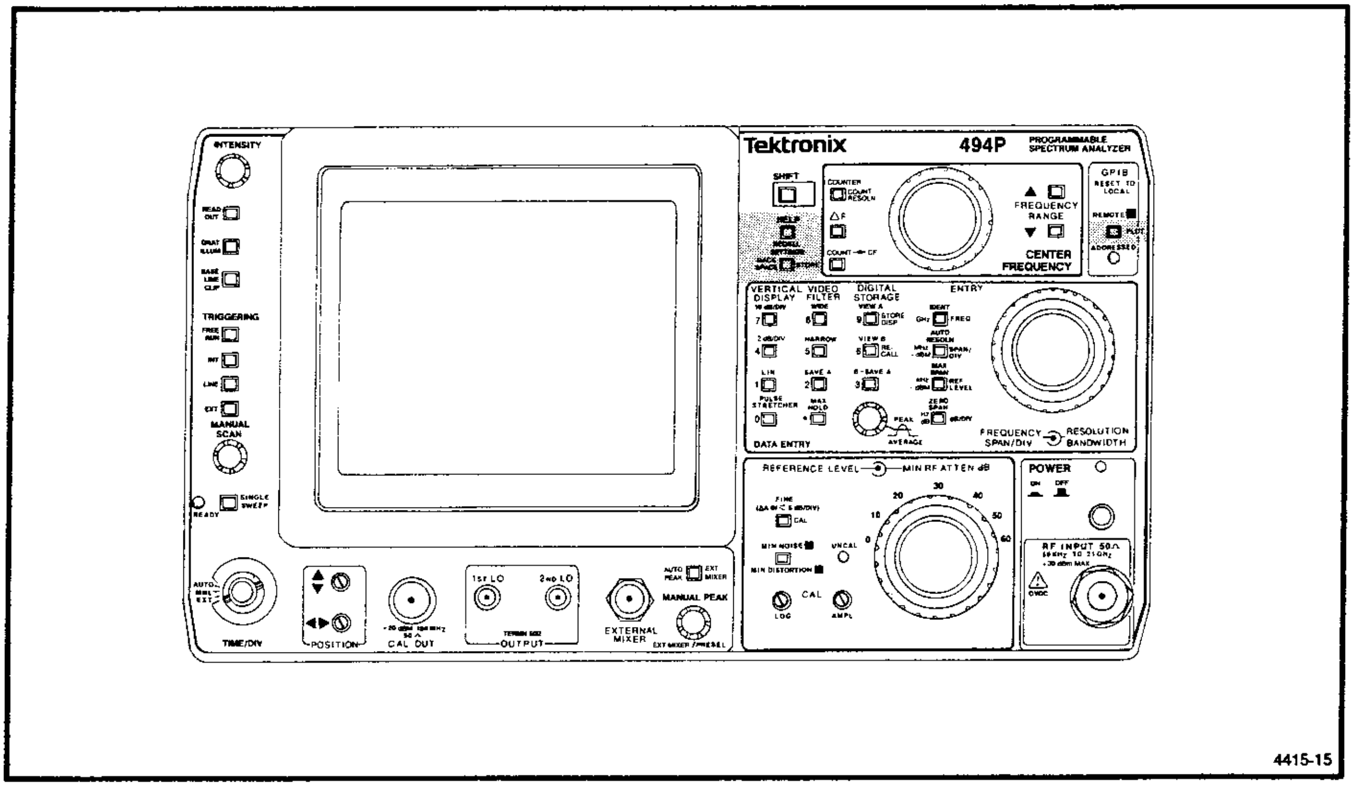
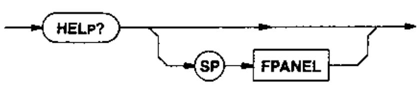


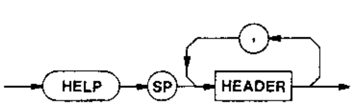
Figure 4-8. Front-panel General Purpose commands and queries.

HELP (send help list of command list) query



4415-196

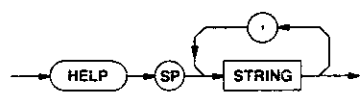
Response to HELP query (with no argument)



4415-197

The response includes a list of all command headers in the 494P GPIB language.

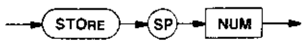
Response to HELP FPANEL query



4415-198

Each string represents one crt readout line of a help message. All help messages are sent in sequence by this command.

STORE (store settings) command



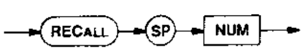
4415-199

NUM — The 494P control settings are loaded into the selected battery-powered memory (range is 0-9).

Power-up value — The 494P automatically STOREs its power down settings in memory 0 when the power is turned off, overwriting previously-stored settings in memory 0.

There is no STORE query.

RECALL (recall settings) command



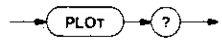
4415-200

NUM — The 494P control settings are recalled from the selected battery-powered memory (range is 0-9).

Power-up value — The 494P automatically STOREs its power down settings in memory 0 when the power is turned off, overwriting previously-stored settings in memory 0.

There is no RECALL query.

PLOT (plot data) query



4415-201

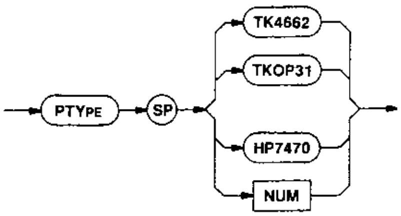
PLOT? sends information to plot the 494P display on the TEKTRONIX 4662 or 4662 Opt 31 Interactive Digital Plotter (or the 4663 in the 4662 emulation mode) or the Hewlett-Packard HP7470A Plotter.

The response to PLOT? depends on the plotter in use.

NOTE

Since the GPIB languages of the 4662, 4662 Opt 31, and 4663 Interactive Digital Plotters or the HP7470A Plotter do not conform to the Tektronix Interface Standard for GPIB Codes, Formats, Conventions, and Features, this response does not follow the standard.

PTYPE (plotter type) command



4415-202

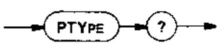
TK4662 — The TEKTRONIX 4662 is selected as the plotter to be driven by the data generated by PLOT?.

TKOP31 — The TEKTRONIX 4662 Opt 31 is selected as the plotter to be driven by the data generated by PLOT?.

HP7470 — The Hewlett-Packard HP7470A is selected as the plotter to be driven by the data generated by PLOT? (the rear-panel LF OR EOI switch must be in the LF OR EOI position, up).

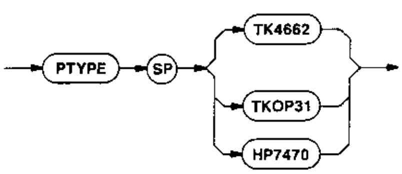
NUM — 0 equals TK4662, 1 equals TKOP31, and numbers ⩾2 equal HP7470A.

PTYPE (plotter type) query



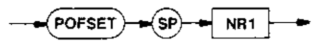
4415-203

Response to PTYPE query



4414-204

POFSET (set K) command



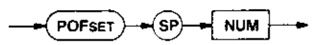
4415-205

NUM — Sets K in the (B−A)+K formula for plotting B−A waveforms using PLOT?. NUM is limited to the 0 to 255 range, and sets K to the nearest limit if out of range (no error reported).

POFSET (set K) query

4415-206

Response to POFSET query



4415-207

DISPLAY DATA AND CRT READOUT I/O

INTRODUCTION

The commands and queries (refer to Table 5-1) in this section transfer display and readout data to or from the 494P. Additional information on waveform transfer and storage is available in Section 8.

NOTE

Some of the lines of input in examples of controller programs in this section extend beyond the column width limitations. Where this occurs, the overrun information is indented on the immediately-following line.

Important—whenever a line is broken, it is always where a natural space occurs. So, be sure to add a space when inputting the program.

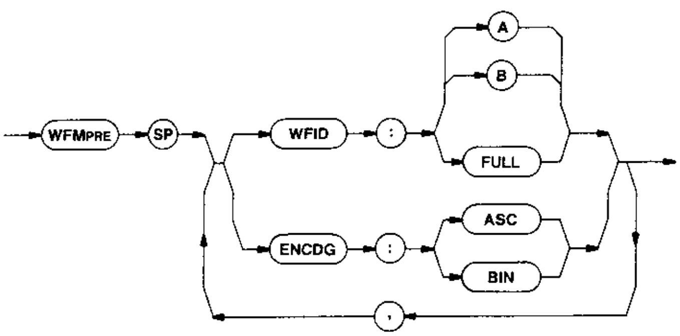
Table 5-1
DISPLAY AND READOUT DATA MNEMONICS

Message Unit	Function
	Waveform Transfers
WFMPRE	Selects A/B/FULL, ASCII/binary data transfers
WFMPRE?	Requests waveform parameters from 494P
CURVE	Sends waveform data to 494P
CURVE?	Requests waveform data from 494P
WAVFRM?	Requests waveform parameters and data from 494P
DPRE?	Requests display parameters from 494P
DCOPY?	Returns model, firmware version, and options; requests waveform and data parameters, and display data
	Crt Readout Transfers
RDOUT	Sends one line of crt readout to 494P
TEXT	Selects SHORT/LONG page for RDOUT
TEXT?	Requests page size of RDOUT
UPRDO?	Requests top line of crt readout from 494P
LORDO?	Requests bottom line of crt readout from 494P

WAVEFORM TRANSFERS

The 494P follows the Tektronix Interface Standard for GPIB Codes, Formats, Conventions, and Features for waveform transfer. Waveform transfers begin with a waveform preamble (WFMPRE) that identifies and scales the data, and data (CURVE) that represents the waveform. A command (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 a display (DCOPY?) can be transmitted to another unit.

WFMPRE (waveform preamble) command



4415-35

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 494P.

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 (B0, A0, B1, A1, B2, A2, ...).

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, 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 half-resolution 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.

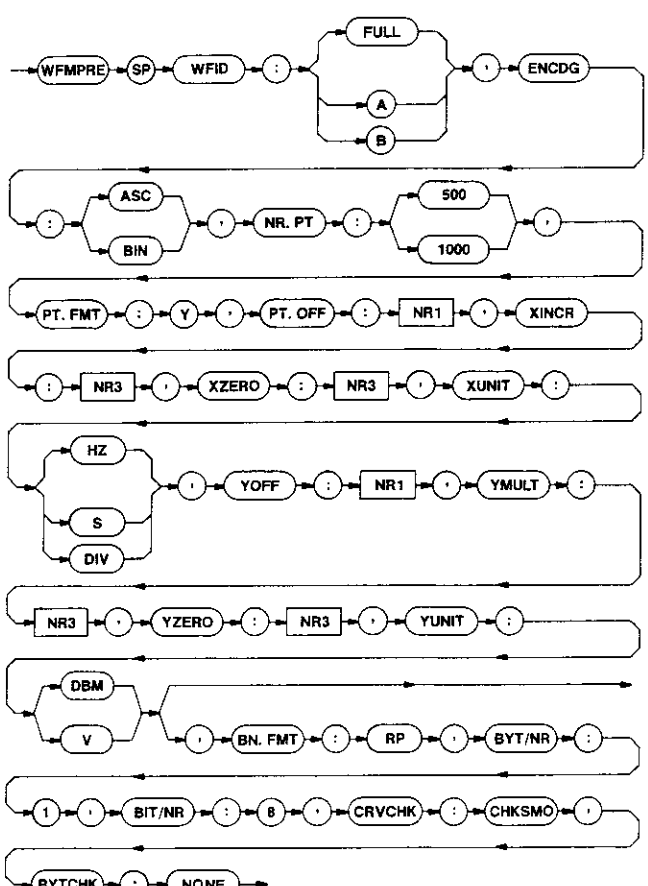
Example — `WFMPRE WFID:FULL`
`WFMPRE ENCDG:ASC`
`WFM WFID:A,ENC:B`

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

WFMPRE (waveform preamble) query

4415-36

Response to WFMPRE query



4415-37

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, either hertz or seconds.

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, either dBm 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 checksum for the preceding bytes (except for the first byte, which is a percent sign parser).

BYTCHK:NULL — Indicates no byte check is appended to binary data transfers.

X-Axis Scaling

X-axis specifications XINCR, PT.OFF, and XZERO are used to interpret the position of the ordered points as absolute X values.

XN = XZERO + XINCR * (N−PT.OFF)

where:

XN is the value in XUNITS on the X axis

XZERO is the center frequency, except in the following cases;

XZERO = 0 for time-domain data (ZEROSP)
XZERO = frequency at graticule center for SPAN MAX

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 = 445 MHz for FULL with SPAN MAX
XN = 2 ms for FULL with SPAN 0 and TIME 2 ms

Y-Axis Scaling

Y-axis specifications YMULT, YZERO, and YOFF are used to interpret the data as the absolute value of the ordered data points

YN = YZERO + YMULT * (VALN − YOFF)

where:

YN is the value in YUNITS of point number N

YZERO is the reference level in log vertical display mode and 0 in linear vertical display mode

YMULT is the scale factor divided by 25

VALN is the unscaled integer data at point N

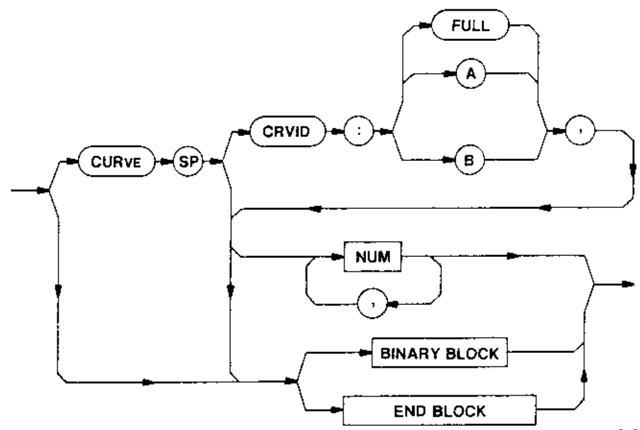
YOFF is 225 (top edge of graticule) in log vertical display mode and 25 (bottom edge of graticule) in linear vertical display mode

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



4415-38

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.

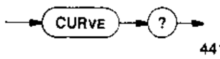
END BLOCK — End block is a sequence of binary numbers that is preceded by the ASCII code for at (@); EOI must be asserted concurrently with the last data byte. The end block can only be the last data type in the message.

Example —

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

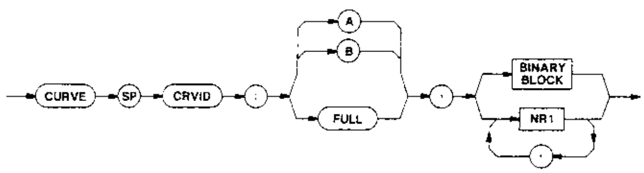
Interaction — A waveform sent in a CURVE command is overwritten in the display I/O buffer if preceded by a CURVE query in the same message. This causes the queried display data to be put back into digital storage.

CURVE (display curve) query



4415-39

Response to CURVE query



4415-40

Waveform data is related to the display by Figure 5-1.

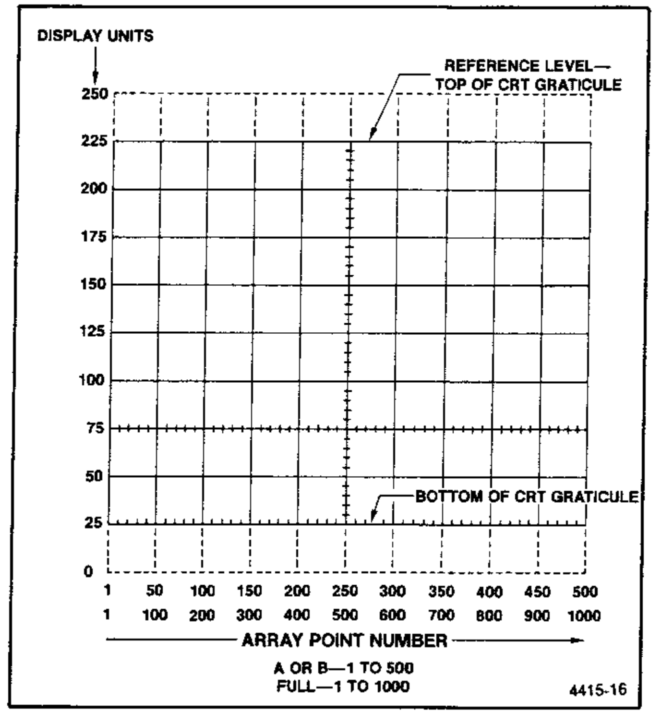


Figure 5-1. Waveform data related to the display.

WAVFRM (waveform) query

4415-33

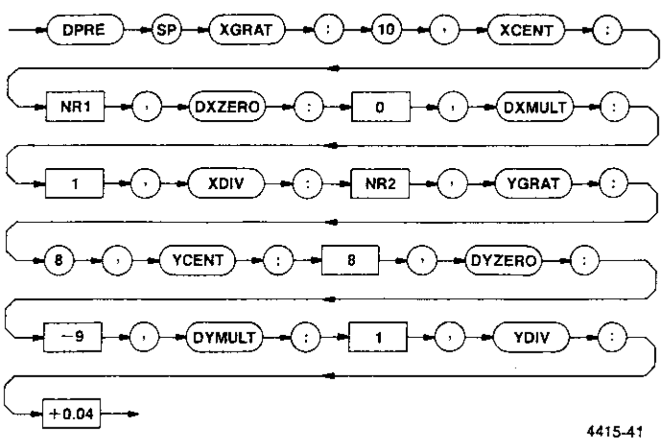
The WAVFRM query response is the same as WFMPRE?;CURVE?. The most recent WFID and CRVID arguments select whether A, B, or both memories are selected for data transfers and waveform processing in ASCII or binary numbers (refer to both the WFMPRE and CURVE queries).

DPRE (display preamble) query

4415-34

DPRE? elicits 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



4415-41

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 — Displays the X offset in divisions relative to XCENT.

DXMULT — Displays the X multiplier.

XDIV — Displays X divisions/unit.

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

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

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

DYMULT — Displays the Y multiplier.

YDIV — 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.

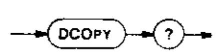
DYN = CYMULT * VALN * YDIV + YCENT + DYZERO

where:

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

DCOPY (copy display) query

The DCOPY query response is the same as 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.

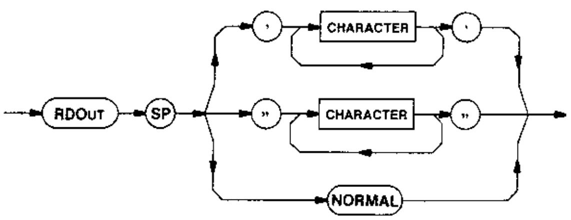


4415-42

CRT READOUT TRANSFERS

Readout messages (RDOUT) can be displayed on the screen in either a 2-line or a 16-line mode (TEXT). Two crt readout queries return the upper row of normal readout characters (UPRDO) or the lower row (LORDO).

RDOUT (readout message) command



4415-43

CHARACTER — In the TEXT SHORT mode, the spectral display remains on the crt, the readout is cleared, and the first 40 ASCII-coded characters are displayed across the bottom of the 494P 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 494P readout to scroll. Lower-case characters are displayed as upper-case characters.

In the TEXT LONG mode, the screen is completely blanked and 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.

Use single quote marks (') instead of double quote marks (") to delimit RDOUT messages in 4050-Series BASIC statements. Reserve double quote marks to enclose the entire message sent by the PRINT statement, as

```
100 PRINT @Z:"RDOUT 'SET CONTROLS AS DESIRED'"
```

NORMAL — Normal 494P readout is restored.

Example —

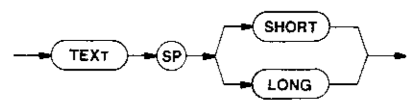
```
RDOUT "TEKTRONIX 494P"
RDO 'SET CONTROLS AS DESIRED'
RDOUT NOR
```

Power-up values — Normal readout.

Interaction — If a crt message sent with a RDOUT command remains on the screen after the analyzer is returned to local control, normal readout can be restored by changing any control that causes the normal readout to be updated. REDOUT ON is required to see any crt readout. The TEXT command switches between 2-line and 16-line modes.

There is no RDOUT query.

TEXT (change readout mode) command



4415-44

SHORT — The GPIB-accessible readout is switched to the 2-line mode with a spectral display. RDOUT commands will load characters to these two lines.

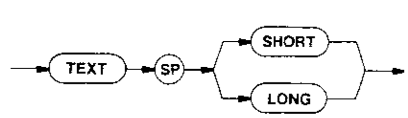
LONG — The GPIB-accessible readout is switched to the 16-line mode without a spectral display. RDOUT commands will load to the top line first, then to successive lines until all are filled. When all 16 lines are filled, the entire screen scrolls up. TEXT LONG will clear the page of readout, and initiate entry of characters into the top line again.

Power-up value — SHORT

TEXT (change readout mode) query

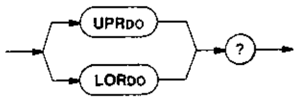
4415-45

Response to TEXT query



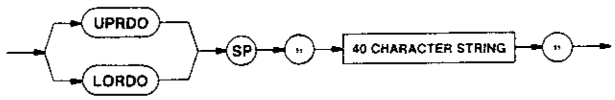
4415-46

UPRDO (upper readout) and LORDO (lower readout) queries



4415-47

Response to UPRDO and LORDO queries



4415-48

CHARACTER — ASCII characters appear in the upper or 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 command (DRECAL) in Section 4.

WAVEFORM PROCESSING

INTRODUCTION

The commands in this section (refer to Table 6-1) allow local processing of spectrum data by the 494P microcomputer. 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 494P display. On command, the 494P microcomputer acquires a display data point from the current digital storage waveform. The point is held in microcomputer memory until another command updates the data point. A query requests that the 494P report the point. Other commands change analyzer settings automatically to center it exactly on the point.

Table 6-1
WAVEFORM PROCESSING COMMANDS AND QUERY

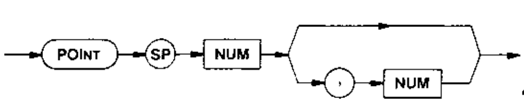
Message Unit	Function
POINT	Directs the microcomputer to a new display data point
POINT?	Requests X and Y values of the display data point
FIBIG	Seeks the largest signal peak
LFTNXT	Seeks the signal peak to the left of the current point
RGTNXT	Seeks the signal peak to the right of the current point
FMAX	Finds the maximum Y value in digital storage
FMIN	Finds the minimum Y value in digital storage
CENSIG	TUNES the frequency to center the signal
TOPSIG	Moves the signal to the reference level

The commands in this section update the display data point direct the microcomputer 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.

Commands that change the center frequency or the reference level to zero in on a signal use the X value (CENSIG) or the Y value (TOPSIG).

This section covers how the waveform processing commands and query work. Two programs at the end of Section 2 show some of these commands in use. Waveform processing techniques are offered in Section 8.

POINT (display data point) command



4415-26

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 5.

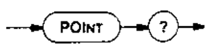
If the second number is omitted, the microcomputer interrogates digital storage for the value of the waveform at X (the first number). This makes the display data point correspond to a point in digital storage. If the second number is supplied in the POINT command, the display data point may not correspond to any point in digital storage.

Example — 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



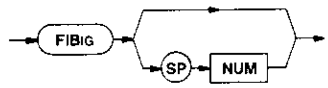
4415-27

Response to POINT query

4415-28

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, LFTNXT, RGTNXT (signal search) commands



4415-29

FIBIG (find big) — 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.

LFTNXT (left-next) — 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.

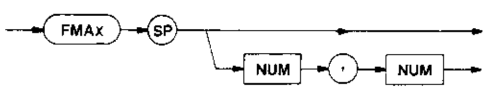
RGTNXT (right-next) — This command searches to the right of the current point to acquire the peak of a signal whose value is greater than NUM. If a signal peak greater than NUM is not found, the display data point is set to 1001,0.

A pattern recognition routine and a threshold value are employed to recognize signals. If the threshold value is omitted from the command, a default value of 0 is used.

Example — FIBIG 100
LFTNXT
RGT

Interaction — See Display Data Point Commands Interaction.

FMAX (find maximum value) command

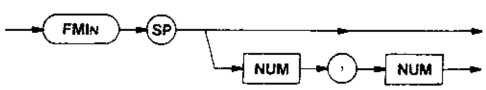


4415-30

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

Interaction — See Display Data Point Commands Interaction.

FMIN (find minimum value) command



4415-31

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

Display Data Point Commands Interaction

1. The preceding waveform processing commands 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 preceding waveform processing commands 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.

a. The simultaneous existance of two conditions 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.

b. If CURVE is preceded by a command to update the display data point, the display data output commands may be invalid. In this case, the curve data is loaded into the buffer as it is received from the GPIB, but it is overwritten when the display data point is updated.

This overwriting causes the data already loaded into the buffer from digital storage to be written back into digital storage when the CURVE command is then executed with the updated data. The overwriting also causes the curve data sent to the instrument to be lost.

One exception to the potential interaction just described is when a Y as well as an X value is sent with the POINT command. In that case, since both values are established by POINT, the microcomputer does not read digital storage data into the buffer and the interaction does not occur.

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 8.

CENSIG, TOPSIG (center or move signal) commands

4415-32

CENSIG (center signal) — 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).

TOPSIG (move to top of graticule) — 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).

These commands do not acquire a new display data point or digital storage waveform. Therefore, if a new waveform is acquired after CENSIG or TOPSIG is executed, the display data point may no longer match the signal of interest.

SYSTEM COMMANDS AND QUERIES

INTRODUCTION

494P 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 listed in Table 7-1 and described in this section in three groups related to instrument parameters, message execution, and status and error reporting.

Table 7-1
DEVICE-DEPENDENT COMMANDS AND QUERIES

Message Unit	Function
	Instrument Parameters
SET?	Returns values of programmable parameters
INIT	Resets programmable parameters to power-up values
ID?	Returns model and firmware versions
	Message Execution
WAIT	Synchronizes message execution with sweep
REPEAT	Repeats execution of previous units in message
	Status and Error Reporting
EOS,EOS?	Turns on/off and queries end-of-sweep SRQ function
RQS,RQS?	Turns on/off and queries RQS message function
Status Byte	Serial poll response
DT,DT?	Defines events triggered by GET
EVENT?	Returns error condition reported in last serial poll byte
ALLEV?	All events query
NUMEV,NUMEV?	Specifies fixed number of events returned in ALLEV?
EVQTY?	Specifies number of events returned in next ALLEV?
ERR?	Returns a code for the current error report
ERCNT?	Returns the number of errors to be reported
TEST?	Initiates self-test routine

INSTRUMENT PARAMETERS

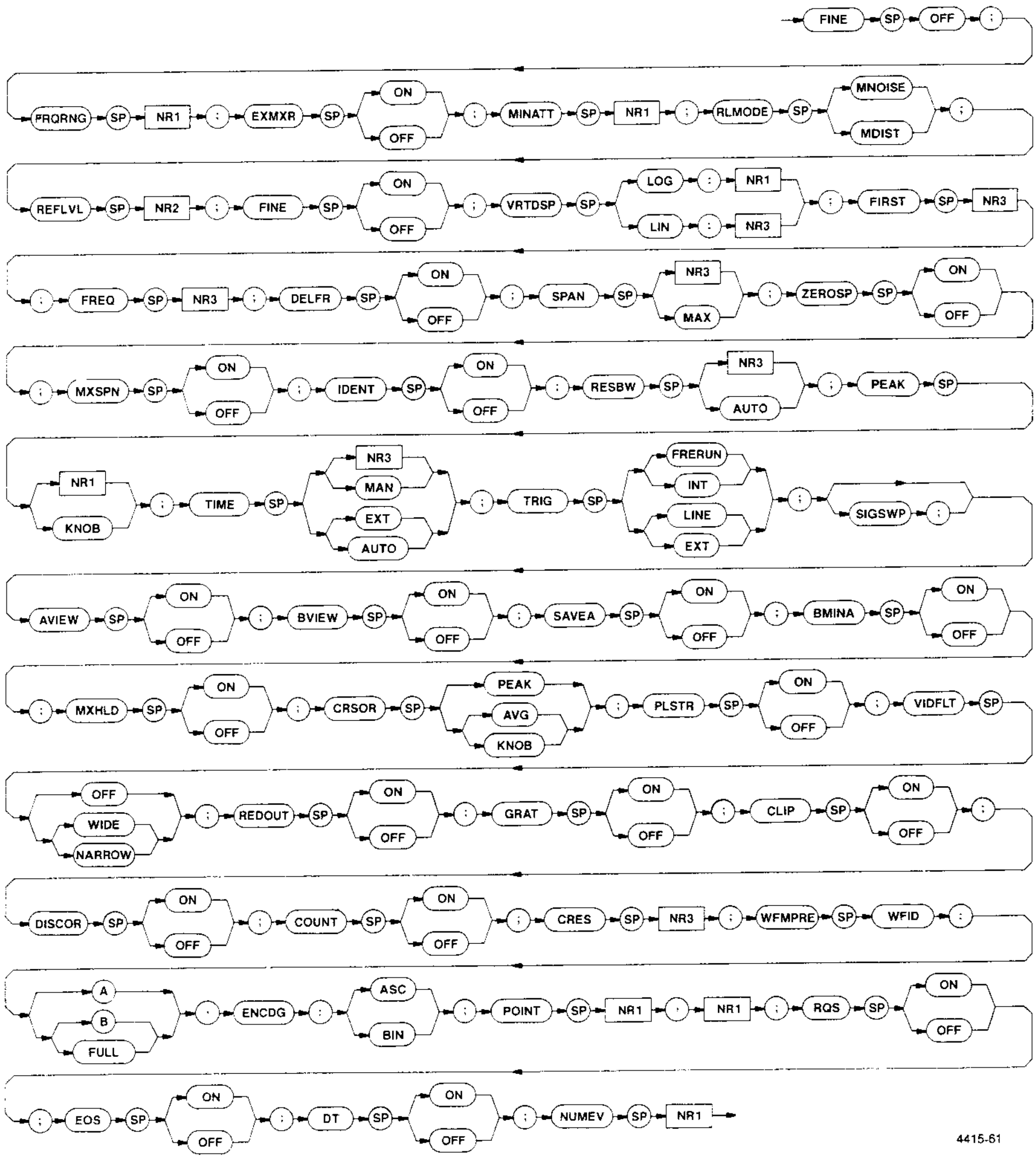
The queries (SET? and ID?) and command (INIT) in this group return settings and identification parameters and initialize settings.

SET (instrument settings) query

4415-60

Response to SET query

The instrument returns a string of commands that can be "learned" for later transfer to the 494P when the same setup is desired. The response includes only those functions necessary for such a setup. To assure no interaction with the Δ A mode that might alter the setup, FINE OFF precedes the string to turn FINE off before the setup begins.



4415-61

INIT (initialize settings) command

4415-62

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 7-2.

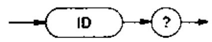
Table 7-2
INSTRUMENT FUNCTIONS

Mnemonic	INIT Value
FREQ	0
FIRST	2.072E+9
SECOND	2.182E+9
DISCOR	OFF
COUNT	OFF
CRES	1 Hz
DELFR	OFF
ZEROSP	OFF
RESBW	1 MHZ
ARES	ON
MXSPN	OFF
PHSLK	OFF
VIDFLT	OFF
VRTDSP	LOG:10 dB
REFLVL	+30 dBm
CAL	OFF
FINE	OFF
RLMODE	MDIST
PEAK	KNOB (AUTO PEAK off)
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
NUMEV	0

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



4415-63

Response to ID query

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

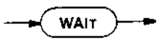
FV[NR2] — Instrument firmware version number.

FPV[NR2] — Front-panel processor firmware version number.

MESSAGE EXECUTION

The following two commands (WAIT and REPEAT) affect how the 494P microcomputer executes message units imbedded within other messages.

WAIT (wait for end of sweep) command



4415-64

The 494P microcomputer delays execution of commands in its input buffer that follow the WAIT command. While the microcomputer waits, it sets its GPIB status byte to busy and does not input device-dependent messages. The wait condition is terminated in either of two ways.

1. WAIT is terminated 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 analyzer to the single-sweep mode if it was previously in a repetitive-sweep mode. The next SIGSWP arms the sweep, and WAIT suspends further execution until the sweep completes. The message ends by the request of a waveform preamble and data. (The analyzer should be triggered or set to FRERUN.)

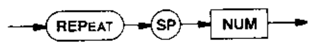
If the sweep is in the single-sweep mode and is not armed (the READY light is on) when the microcomputer encounters WAIT, the microcomputer continues to execute the message in the buffer and does not wait.

2. WAIT is terminated if DCL or SDC (while listener-addressed) is received. This results in flushing the input and output buffer so any commands that follow WAIT are aborted. (See Status Byte later in this section.)

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

There is no WAIT query.

REPEAT (repeat execution) command



4415-65

NUM — This determines the number of times the microcomputer is to repeat the execution of commands or queries that precede REPEAT.

NUM range — 0 to 16,777,215 ($2^{24}-1$).

Since REPEAT may itself be one of the commands that precedes a REPEAT, the nested REPEAT will only be executed on the first pass through the commands that precede the second REPEAT. For example,

```
RGTNXT;FREQ?;REPEAT 10;FREQ 15 GHZ;REPEAT 1
```

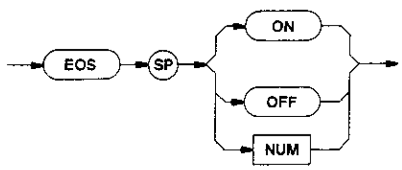
This causes the 494P to output 12 frequency values after it executes the message, because it only executes the frequency query once on its second pass through the entire message.

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

STATUS AND ERROR REPORTING

Two commands (EOS and RQS) control 494P service requests. The status byte reports instrument status in a format that implements both IEEE 488 and the Tektronix Interface Standard for GPIB Codes, Formats, Conventions, and Features. GET is enabled to trigger a new sweep (DT). A query (EVENT?) returns detailed information about events reported in the last serial poll status byte. Two queries and one command (ALLEV?, NUMEV, and EVQTY?) specify the identity and quantity of events reported. Two queries (ERCNT? and ERR?) are included for 492P compatibility to provide more detailed information on errors related to abnormal status conditions. One query (TEST?) checks the 494P ROMs and RAM.

EOS (end-of-sweep) command



4415-66

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

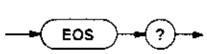
OFF — The analyzer does not assert SRQ for the EOS conditon.

NUM — 1 equals ON; numbers ≥ +0.5 are rounded to 1. 0 equals OFF; numbers less than +0.5 are rounded to 0.

Power-up value — Off.

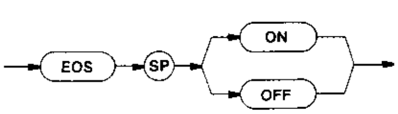
Interaction — EOS is always OFF in the talk-only and listen-only modes.

EOS (end-of-sweep) query



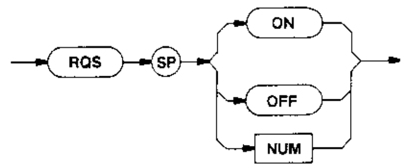
4415-67

Response to EOS query



4415-68

RQS (request service) command



4415-69

ON — SRQ is asserted when abnormal status conditions occur.

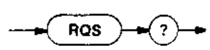
OFF — SRQ is not asserted (is masked) when abnormal status occurs.

NUM — 1 equals ON; numbers ≥ +0.5 are rounded to 1. 0 equals OFF; numbers less than +0.5 are rounded to 0.

Power-up value — On.

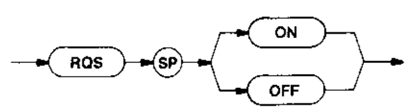
Interaction — RQS is always OFF in the talk-only and listen-only modes.

RQS (request service) query



4415-70

Response to RQS query



4415-71

Status Byte (response to serial poll)

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

- Bits 4–1: Four-bit status code
- Bit 5: 494P microcomputer busy condition
- Bit 6: Abnormal (1)/normal (0) condition
- Bit 7: SRQ is asserted (depends on RQS and EOS commands)

Power-on is reported only if an internal switch is set to request this status.

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 494P 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 parsed or recognized.

Execution error — This results when a message is parsed and is recognized, but cannot be executed, such as FREQ 999 GHZ.

Internal error — This indicates that the 494P microcomputer has discovered a malfunction that could cause the instrument to operate incorrectly.

Execution error warning — This results from a command that the 494P executes, but has a potential for error. An example is RESBW 10 KHZ in the max span mode. The 494P sets the warning status because the UNCAL indicator is lit.

Internal error warning — This reports that a non-fatal operating condition has been detected by the 494P microcomputer.

Busy — This is reported whenever the 494P microcomputer executes a message in its input buffer. This includes the WAIT command; while waiting, the microcomputer reports busy status.

Affect of Busy on Device-Dependent Messages

The microcomputer will not accept any further device-dependent messages while the busy condition exists; if made a listener, it asserts NRFD. Commands that require microcomputer interaction with the hardware can keep the microcomputer busy for a second or more (significant to some bus controllers); for instance, commands such as DEGAUS and INIT. The waveform processing commands and PEAK AUTO can also require significant processor time. Of course, long messages such as the SET? response take a while to execute (see Execution Times, Table 8-1 in Section 8). Although output operations, such as the CURVE? response, may take a long time to complete, the microcomputer is busy only for the time it takes to load the output buffer.

Affect 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 the execution of a message, the microcomputer attempts to execute the remainder of the message after local control is restored. At that time, commands that attempt to change a front-panel function will result in error SRQs, because they conflict with local control.

The response of the 494P 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 494P GPIB address, and are implemented in microcomputer firmware rather than on the interface. The speed with which these commands can be handshaked depends on how fast the microcomputer can service the resulting interrupt; which, in most cases, should be within a few hundred μs.

The following considerations apply to interface messages received by the 494P.

1. Universal commands LLO, SPE, and SPD are handshaked and acted on by the interface, so they are unaffected by the microcomputer's 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 microcomputer can get involved. The microcomputer 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 microcomputer can execute the programs that update the addressed status.

Because the microcomputer gets involved when a current address is received, addressed commands are impacted by the speed at which the microcomputer can service interrupts. Serial poll is similarly affected if MTA preceded SPE.

3. GTL is handshaked immediately by the interface. If the 494P is already listen-addressed, the microcomputer returns the 494P 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 microcomputer action that will hold up the handshake if the microcomputer is busy. If the 494P is listen-addressed, the microcomputer treats SDC in the same manner. These two device-clear messages are executed as soon as they are accepted.

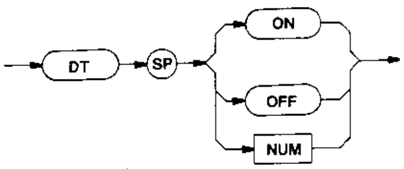
5. GET also requires action by the microcomputer, so its handshake occurs only when the microcomputer can handle the interrupt. GET is executed immediately, aborting the current sweep and rearming the sweep.

6. Parallel polls are handled by the microcomputer, so PPC, PPE, PPD, and PPU must wait for the microcomputer to service the interrupts before they can be executed. This assumes that the 494P was addressed for the parallel poll sequence.

Busy and end-of-sweep are independent. Busy exists only while the microcomputer is executing a command, and end-of-sweep indicates that sweep and data-updating are complete. If a single-sweep command is sent, the microcomputer remains busy only until it can initiate the sweep, while end-of-sweep does not occur until the operation is complete.

When polled, the 494P reports a status code related to its SRQ, if any. Bit 5 always reflects 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 (device trigger enable) command



4415-72

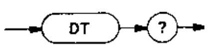
ON — GET is enabled to trigger a new sweep.

OFF — The response to GET is disabled.

NUM — 1 equals ON; numbers ≥ +0.5 are rounded to 1. 0 equals OFF; numbers less than +0.5 are rounded to 0.

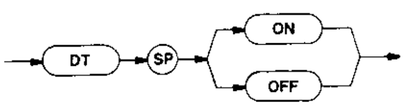
Power-up value — On.

DT (device trigger enable) query



4415-73

Response to DT query



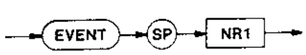
4415-74

EVENT (event information) query

4415-75

The EVENT query returns more detailed information about the event that was reported in the last serial poll status byte. It also allows a controller to get information about events when the controller's RQS assertion capability has been disabled by RQS OFF.

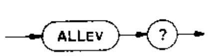
Response to EVENT query



4415-76

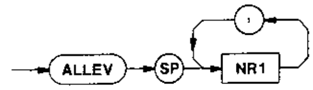
NR1 represents an event code that is defined at the end of this section in the ERROR and EVENT Codes, Table 7-4. The event is cleared when the event code is reported.

ALLEV (all events) query



4415-77

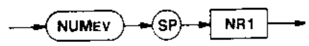
Response to ALLEV query



4415-78

The NR1s represent the event codes that are defined in the EVENT and ERROR Codes, at the end of this section, Table 7-4. The events are cleared when their event codes have been reported.

NUMEV (number of events) command



4415-79

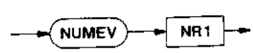
NR1 — This specifies that a fixed number of event codes is to be returned in ALLEV?. If fewer events are pending when ALLEV? is executed, the response is filled with zeros to provide the specified number. The zero value sets the 494P to return a variable number of event codes.

Power-up value — 0.

NUMEV (number of events) query

4415-80

Response to NUMEV query

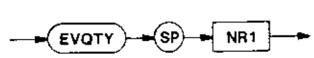


4415-81

EVQTY (event quantity) query

4415-82

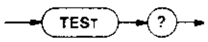
Response to EVQTY query



4415-83

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.

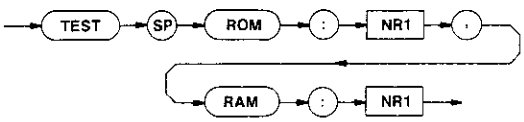
TEST (internal test) query



4415-84

This command checks the system ROMs and RAM.

Response to TEST query



4415-85

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, respectively. (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 charts in Figure 7-1 to identify the IC number. Then, use Table 7-3 to find the correct circuit number and circuit board. The following example shows how to use the conversion charts and Table 7-3. If any ROM or RAM ICs are indicated to be bad, refer this information to qualified service personnel.

Example: Enter `100 PRINT @A: "TEST?"`

```
100 INPUT @A: R$
120 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 01000000010000.

2. Insert this binary number in part A of Figure 7-1 (right-justified). Blocks 6 and 2 will be 0 1. This indicates that both ROM #6 and ROM #2 are bad; all other ROMs are good.

3. Table 7-3 shows that ROM #6 is U2018 and that ROM #2 is U1018, both located on the GPIB board.

RAM

1. The binary equivalent of the RAM number 18 is 10010.

2. Insert this binary number in part B of Figure 7-1 (right-justified). Blocks 5 and 2 each contain a 1, which indicates that both RAMs 5 and 2 are bad; all other RAMs are good.

3. Table 7-3 shows that RAM #5 is U2044 and RAM #2 is U1027, both located on the Memory board.

Table 7-3
TEST CONVERSION
(Version 670-8431-00)

Device	Chart Location	Circuit Board	Circuit Number
ROM	0	A54 Memory	U2015
	1	A56 GPIB	U1012
	2	A56 GPIB	U1018
	3	A56 GPIB	U1022
	4	A56 GPIB	U1024[a]
	5	A56 GPIB	U2013[b]
	6	A56 GPIB	U2018[c]
	7	A56 GPIB	U2022[d]
	8	A54 Memory	U1021
	9	A54 Memory	U1014
RAM	1	A54 Memory	U1032
	2	A54 Memory	U1027
	3	A54 Memory	U2014
	4	A54 Memory	U2021
	5	A54 Memory	U2044
	6	A54 Memory	U2039
	7	A54 Memory	U2032

[a] ROM #4 was U2013 in version 670-7896-00.
[b] ROM #5 was U2018 in version 670-7896-00.
[c] ROM #6 was U2022 in version 670-7896-00.
[d] ROM #7 was not installed in version 670-7896-00.

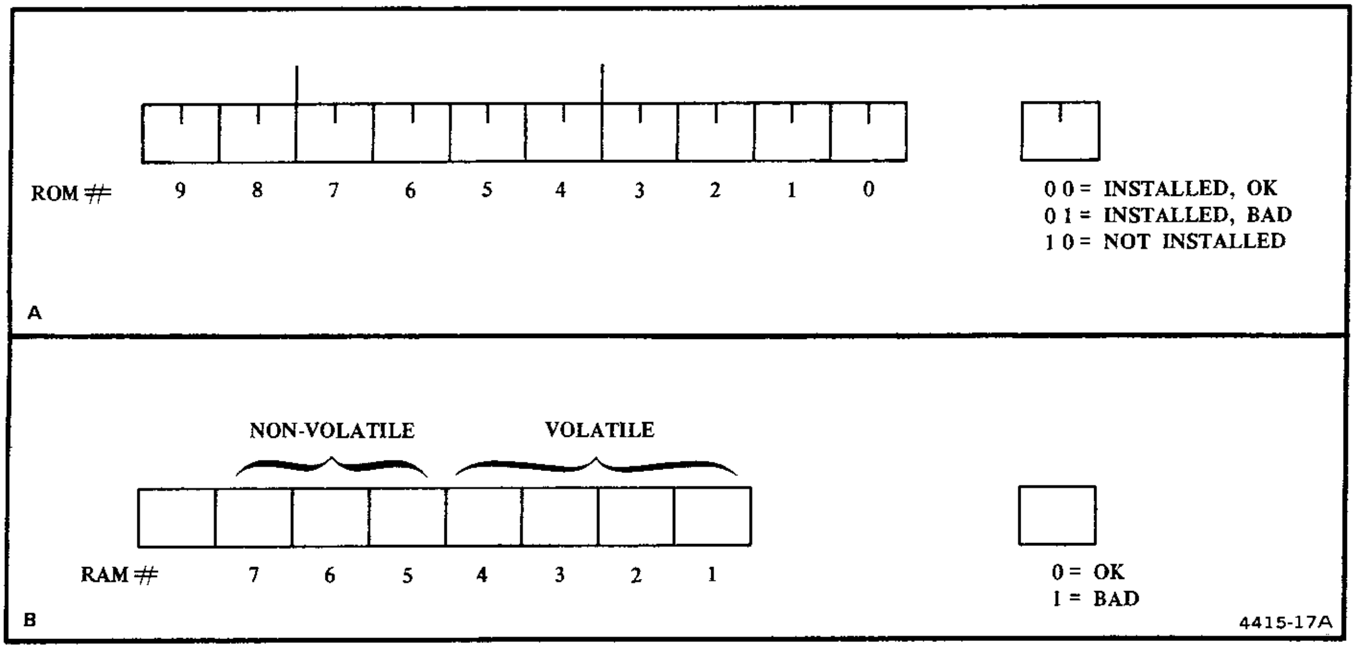


Figure 7-1. TEST Conversion Chart.

Error and Event Codes

The Tektronix Interface Standard for GPIB Codes, Formats, Conventions, and Features specifies device-dependent Error and Event codes by category. Table 7-4 identifies each general category and lists the codes within that category. Following the listing are the specific error messages returned by the 494P. Error codes are returned in numerical order as they appear in the table. When the current code(s) is read, the error response is cleared.

Table 7-4
494P 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	Binary block 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 for this header
11	103	Numeric argument not allowed for this header
12	103	String argument not allowed for this header
13	103	Binary argument not allowed for this header
14	103	Link not allowed for this argument
20	103	Special argument type not recognized
21	103	Special argument not allowed for this header
22	103	Character argument not recognized
24	150	Input buffer overflow

Table 7-4 (cont)

Error Code	Event Code	Meaning
		Execution Errors
	201	Command not executable in Local mode
	204	Settings conflict
	205	Argument out of range
	206	GET (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	205	FREQ, TUNE, FIRST, or SECOND out of range
29	205	FRQRNG out of range
30	205	CRES out of range
31	205	SPAN out of range
32	205	RESBW out of range
33	205	MAXPWR or MINATT out of range
34	205	REFLVL out of range
35	205	VRTDSP LIN out of range
36	205	VRTDSP LOG out of range
37	205	TIME out of range
39	204	IDENTIFY not allowed in this span/div
40	204	Signal finds not allowed in this span/div
45	206	GET (Group Execute Trigger) ignored (not executed)
46	205	NUMEV out of range
47	205	STORE, RECALL, DSTORE, or DRECAL out of range
48	204	PHSLK cannot be turned OFF/ON directly with PHSLK command
		Execution Warnings
	550	FREQ change caused EXMXR change
	551	SPAN defaulted to MAX
	552	SPAN defaulted to 0
	553	UNCAL light turned on
	554	UNCAL light turned off
	555	Multiple use of display buffer
49	550	FREQ change caused EXMXR change
50	551	SPAN defaulted to MAX
51	552	SPAN defaulted to 0
52	553	UNCAL light turned on
53	555	Multiple use of display buffer
54	554	UNCAL light turned off
		Internal Warnings
	650	Frequency reference changed to INT
	651	Frequency reference changed to EXT
55	650	Frequency reference changed to INT
56	651	Frequency reference changed to EXT

Table 7-4 (cont)

Error Code	Event Code	Meaning
		Internal Errors
	302	System error
	350	Tuning DAC carry operation failure
	351	Failed to lock 1st LO
	352	Lost 1st LO lock
	353	Recentering failure on unlocking of 1st LO
	354	Calibration failure
	355	Battery-powered 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
	396	Power supply out of regulation
	397	Power supply regained regulation
	398	Frequency reference unlocked
	399	Frequency reference relocked
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-powered RAM checksum error
72	382	1st LO tuning system failed
74	386	2nd LO tuning system failed
75	388	Phase lock system failed
78	394	IF count failed
79	396	Power supply out of regulation
80	398	Frequency reference unlocked
82	383	1st LO tuning system recovered from a failure
84	387	2nd LO tuning system recovered from a failure
85	389	Phase lock system recovered from a failure
88	395	IF count recovered from a failure
89	397	Power supply regained regulation
90	399	Frequency reference relocked
		System Events
	401	Power on
	402	Operation complete
97	401	Power just came on
98	402	Operation complete (end of sweep)
		Internal Error
99	302	Unrecognized event occurred

HELPS AND HINTS

INTRODUCTION

This section covers some techniques for programming the 494P, using 4050-Series BASIC for examples. We hope this information will speed your progress in putting the 494P to work solving your measurement problems.

NOTE

In the examples throughout this section, the 494P primary address is assumed to be 1. See Section 1 for how to set the GPIB ADDRESS switches.

Some of the lines of input in examples of controller programs in this section extend beyond the column width limitations. Where this occurs, the overrun information is indented on the immediately-following line.

Important—whenever a line is broken, it is always where a natural space occurs. So, be sure to add a space when inputting the program.

DATA ACQUISITION

When the 494P is acquiring spectrum data under program control, there are two programs running, not just one. One program is running in the controller, and a second in the 494P. The key to success is synchronizing the execution of the two programs.

In addition, the execution of the two programs must be synchronized with the event that accomplishes data acquisition; in this case, the sweep.

Synchronizing Controller and 494P

Program execution in the controller can be synchronized with 494P execution of messages it receives over the GPIB. This is all done within the 494P by the way it buffers and executes messages.

When the 494P receives a message, it waits until the end of the message to begin execution. While busy executing the message, the 494P does not accept any other device-dependent messages. When it is finished executing the message, the 494P is ready to handshake another message, which it then executes, and so on. You can depend on the 494P to assert NRFD on the GPIB while it is busy; this prevents the execution of a controller GPIB output statement that would send further instructions to the 494P.

For example, enter

```
100 FOR I=1 TO 10
110 PRINT @Z:"FREQ ";I;"GHZ"
120 NEXT I
```

Watch the 494P 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 4050-Series controller step through the loop at a more deliberate pace? It must wait at line 110 (after the first pass through the loop) for the 494P to execute the previous FREQ command.

A controller GPIB input statement can also be used to synchronize the controller and the 494P. The controller could make a table of frequency ranges for the frequencies covered by the previous loop, filling the table only after the FREQ command is executed.

```
100 DIM F(10,2)
110 FOR I=1 TO 10
120 PRINT @Z:"FREQ ";I;"GHZ;FRQRNG?"
130 INPUT F(I,1)
140 F(I,2)=I*1E+9
150 NEXT I
```

Line 130: Addresses the 494P to talk; however, the 494P does not begin talking until it finishes executing the message in line 120. This assures that the 494P 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 494P. 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 494P to acquire, process, and output data, all in the same message. If the commands or queries you add to process or output data follow WAIT, the results will be based on data acquired by the SIGSWP command.

For example, enter

```
100 DIM P(5)
110 PRINT @Z:"SIGSWP"
120 FOR I=1 TO 5
130 PRINT @Z:"FREQ ";I;"GHZ;SIGSWP"
140 PRINT @Z:"WAIT;FMAX;POINT?"
150 INPUT @Z:P(I)
160 NEXT I
```

Line 110: Sets the 494P to the single-sweep mode (if the 494P is not already in the single-sweep mode). Succeeding SIGSWP commands arm the sweep.

Lines 130 and 140: Illustrate how to use WAIT. WAIT follows SIGSWP and precedes the command and query that ready the 494P to output the updated data. The 494P does not handshake out the data in line 150 until it finishes executing the message in lines 130 and 140.

This simple routine only gets the X variable of the display data point. In this case, it is the horizontal location of the point with the largest value in digital storage. The Y variable is lost each time through the loop when the 494P receives further input before it can handshake out the second POINT argument.

Figure 8-1 further illustrates the two-program concept (one in the controller and one in the 494P) 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/494P execution with the sweep is recommended, there is another method. This alternative may be necessary for some operating systems or application programs that allow a short response time when the 494P is made a talker or that must take care of other tasks while the 494P is acquiring data. In such cases, the controller can enable the 494P end-of-sweep SRQ to synchronize data acquisition with the sweep. The following example just shifts the WAIT from the 494P program to the 4050-Series 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.

```
100 DIM P(5)
110 ON SRQ THEN 200
120 PRINT @Z:"SIGSWP;EOS ON"
130 FOR I=1 TO 5
140 PRINT @Z:"FREQ ";I;"GHZ;SIGSWP"
150 WAIT
160 PRINT @Z:"FMAX;POINT?"
170 INPUT @Z:P(I)
180 NEXT I

200 POLL Q1,Q2;Z
210 RETURN
```

With no WAIT command following SIGSWP in line 140, the 494P is ready to buffer another message. But, the controller does not send it immediately because of the WAIT statement in line 150. The SRQ that the 494P asserts at the end of its sweep, which was enabled in line 120, triggers the controller to perform a serial poll (lines 200 and 210) and then send the message in line 160.

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 494P 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).

```
100 DIM P(5)
110 PRINT @Z:"SIGSWP"
120 FOR I=1 TO 5
130 PRINT @Z:"FREQ ";I;"GHZ"
140 PRINT @Z:"FREQ ";I;"GHZ;SIGSWP;WAIT"
150 INPUT @Z:D$
160 PRINT @Z:"FMAX;POINT?"
170 INPUT @Z:P(I)
180 NEXT I
```

Here the WAIT is put back into the 494P message and the INPUT statement in line 150 stalls the controller while the 494P makes a sweep. D$ serves the purpose of a dummy string; the data is not input until line 170.

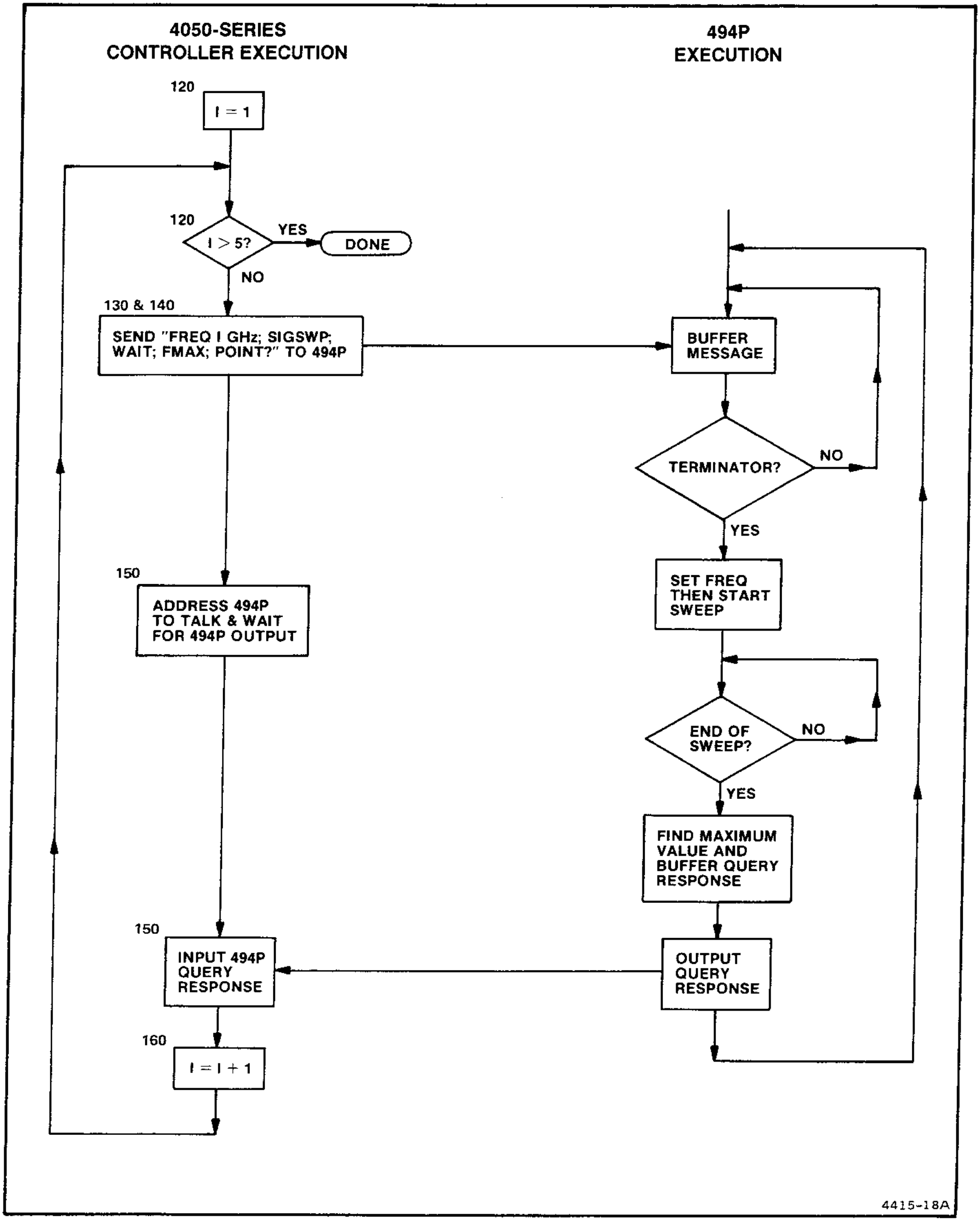


Figure 8-1. Synchronizing controller and 494P for data acquisition.

BINARY WAVEFORM TRANSFER

Selecting binary, rather than ASCII-coded decimal, speeds up waveform transfers. Neither the controller nor the 494P 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. The gains possible by using binary are not hard to achieve. Here's how.

Getting 494P Binary CURVE Output

The 494P encloses binary waveform data values in some other items in the binary block format. For details, see the syntax diagrams in Section 5.

For a 4050-Series routine that handles block binary, enter the following.

```
500 REM GET 494P BINARY CURVE OUTPUT
510 DIM W(100)
520 PRINT @37,0:37,255,255
530 PRINT @Z: "WFMPRE ENC:BIN;CURVE?"
540 INPUT %Z:H$
550 WBYTE @65:
560 RBYTE A,B,W,D
570 WBYTE @95:
```

Line 520: Sets the second processor status byte in the 4050-Series controller to an alternate delimiter, ASCII 37 (%). The percent sign in line 540 instructs the controller to use the alternate delimiter for H$. This maneuver inputs the header in the 494P response (CURVE CRVID:FULL,) and stops at the block binary delimiter (%), which is discarded.

Line 550: Makes the 494P a talker.

Line 560: First inputs the initial two bytes in the block binary format (the byte count) into A and B. This routine does not make use of the byte count, but could be expanded to count the bytes as an error check. Line 560 next inputs the binary waveform data to fill array W. The RBYTE statement completes by inputting the check sum into D. Statements could be added to this routine to keep a running 8-bit sum of the bytes in the binary block; such a sum could be added to the check sum byte as an error check (two added together, disregarding the carry, should equal zero).

Line 570: Untalks the 494P.

Sending a Binary CURVE to the 494P

The following routine employs end block format to transfer a waveform to the 494P. Array W is transferred; if not already created by the preceding routine, W should be dimensioned to 1000 and filled with data in the range 0 to 255.

```
600 REMARK SEND BINARY CURVE TO 494P
610 WBYTE @33:64,W,-255
620 WBYTE @63:
```

Line 610: Sends the 494P listen address, followed by the binary number for 63, which is the ASCII code for the end-block delimiter (@). Line 610 then sends the binary numbers in array W, after which it asserts EOI asserted concurrently with 255 (a byte with all bits set to one). EOI causes the 494P to act on the message. The CURVE header is omitted; it is not required, but would be accepted if sent. The 494P buffers the last byte (hex FF), but does not put it into digital storage.

Line 620: Sends UNL.

SCALING, SAVING, AND GRAPHING WAVEFORM DATA

The 494P 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 494P binary output program given previously. This version transfers whatever portion of memory you have 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 n-scaled binary array so you can transfer it back into 494P digital storage if you wish.

```
500 REMARK GET AND SCALE 494P BINARY CURVE
    OUTPUT
510 DELETE W,M
520 PRINT @Z: "WFMPRE ENC:BIN;WFMPRE?CURVE? "
530 PRINT @37,0:37,255,255
540 INPUT %1:N,X3,X2,X1,Y3,Y2,Y1
550 DIM W(N ),M(N,2 )
560 WBYTE @65:
570 RBYTE A,B,W,D
580 WBYTE @95:
590 FOR I=1 TO N
600 M(I,1 )=X1+X2*(I-1-X3 )
610 M(I,2 )=Y1+Y2*(W(I )-Y3 )
620 NEXT I
```

Line 510: Clears the waveform arrays.

Line 520: Requests the waveform preamble and a binary curve.

Line 530: Sets the alternate delimiter to the block binary delimiter (%).

Line 540: Makes the 494P a talker and inputs the 494P WFMPRE and CURVE response until it reaches the percent sign, storing the first seven numbers it finds as variables N, X3, X2, etc. These numbers are the waveform parameters sent in the WFMPRE query response.

Line 550: Dimensions the arrays to fit the incoming waveform.

Lines 560 — 580: Talk the 494P, input the elements in the block binary format, and untalk the 494P.

Lines 590 — 620: Scale the waveform integers and fill array M with the result. The first number in each element of the array is a frequency, 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).

Saving the Scaled Array

A single statement (WRITE) transfers an array to tape. First, however, you must find and mark an adequate tape file. These statements do the job (insert the number of the last tape file for N).

```
FIND N
MARK 1,20000
FIND N
WRITE @33:M
```

These statements return the data from the tape.

```
FIND N
READ @33:M
```

Storing Settings

If a particular series of settings are commonly used in an application, it is recommended that these settings be stored within the 494P 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.

Waveform Plotting

A simple routine plots array W, the integers output by the 494P. The plot (see Figure 8-2) is embellished by labels derived from the waveform preamble obtained by the previous program.

```
700 REMARK SIMPLE WAVEFORM PLOT
710 VIEWPORT 10,110,10,90
720 WINDOW 1,N,1,250
730 PAGE
740 AXIS N/10,25,N/2,75
750 MOVE 1,W(1 )
760 FOR I=2 TO N
770 DRAW I,W(I )
780 NEXT I
790 MOVE N/2,5
800 PRINT X1;" HZ"
810 MOVE N,75
820 PRINT Y1+Y2*(75-Y3 );" DBM"
```

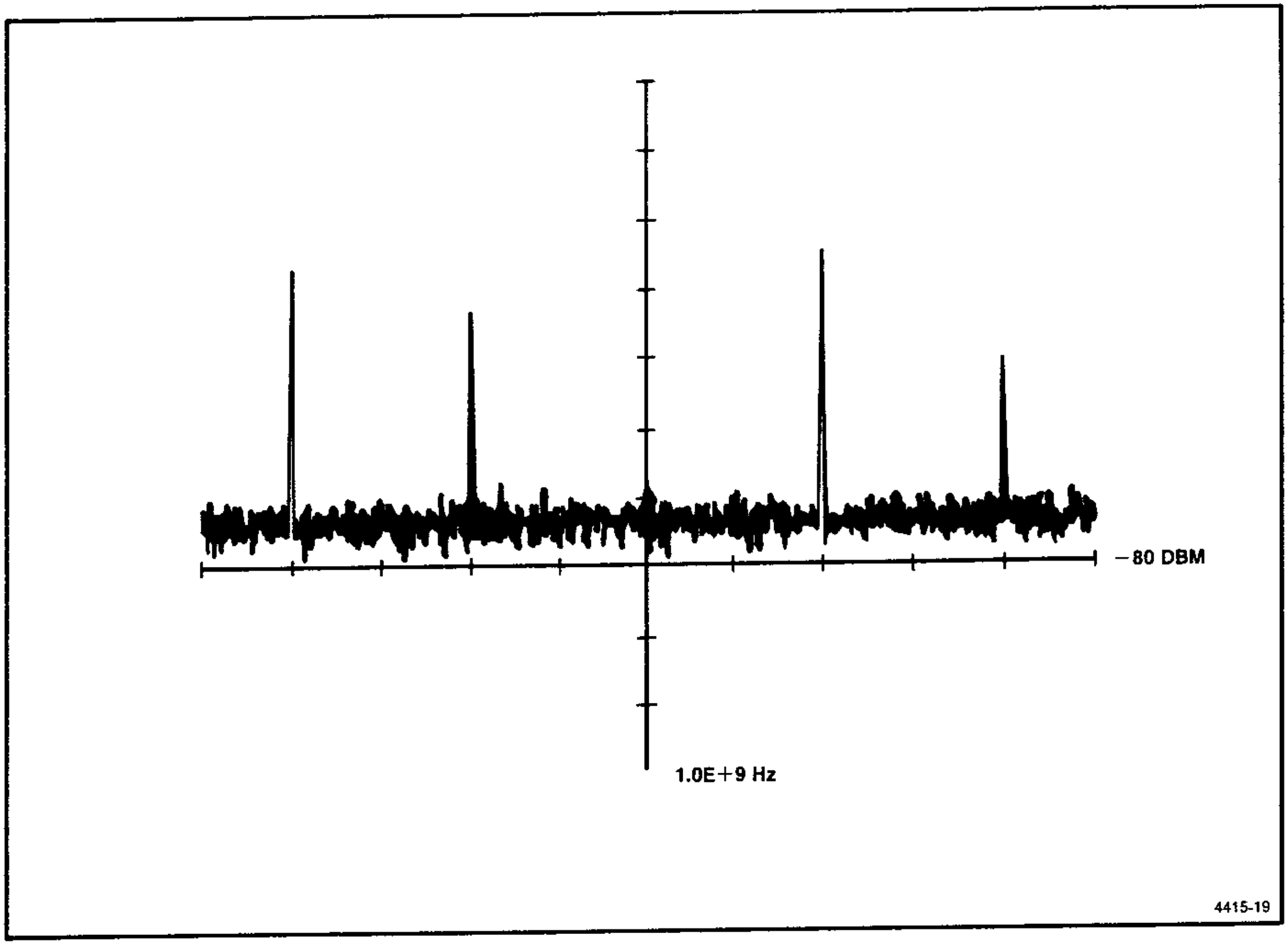


Figure 8-2. A simple plot of a spectrum acquired from the 494P.

Line 710: Shrinks the plot slightly to leave room for labels.

Line 720: Sizes the plot for the data.

Line 740: Marks off cross hairs as a reference for the plot. Add more AXIS statements if you wish to mark the reference level or fill in other parts of the graticule.

Lines 750 — 780: Draw the plot, point-by-point.

Lines 790 and 800: Label the X-axis center marked by the vertical cross hair with the center frequency.

Lines 810 and 820: Label the amplitude marked by the horizontal cross hair (six divisions below the reference level). The vertical cross hair extends over the full 10 divisions of vertical data, including a full division above and below what you see on the 494P crt. If there is data in digital storage from these areas outside the 494P display, the data is acquired and plotted.

Other MOVE and PRINT statement pairs could be added to label other points on the plot.

For instance, to label the start frequency (using the scaling formula), enter

```
830 MOVE 0,0
840 PRINT X1+X2*(1X3); "HZ"
```

Using PLOT?

The 494P can generate a plot of the display directly on the Tektronix 4662 or 4662 Opt 31 (or a 4663 in the 4662 emulation mode) or the Hewlett Packard HP7470A Plotter. All selected waveform, graticule, and crt readout data can be plotted. PLOT? sent to the 494P 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 without further intervention. The following routine assumes the 494P is at address Z, the plotter is at address P, and that the plotter type has been selected with the PTYPE command.

```
100 PRINT @Z: "GRAT ON;CLIP ON;RDOUT ON;PLOT?"
110 WBYTE @64+Z,32+P:
120 ON EOI THEN 200
200 WBY @63.95:
210 RETURN
```

Line 100: Illuminates the graticule, turns the baseline clip and the crt readout on, and readies the 494P to send a waveform to the plotter.

Line 110: Makes the 494P a talker and the plotter a listener.

Line 200: Untalks the 494P and unlistens the plotter.

MULTIPLE USE OF DISPLAY BUFFER FOR WAVEFORM PROCESSING AND I/O

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, at several points in this manual, you are informed of possible interaction involving waveform processing and waveform data I/O executed in the same 494P message. (This occurs in Section 5 under the Interaction part of the CURVE command and under Display Data Point Commands Interaction in Section 6.)

There is no conflict in many cases because the 494P buffers the message you send and then executes it in the order you sent it.

For example, you can use the 494P as a waveform processor for spectrum data you previously acquired in array A by entering

```
100 REMARK BUFFER DEMO
110 PRINT @Z: "CURVE ";A, ";FIBIG;POINT?"
120 INPUT @Z:B1,B2
```

In this case, the 494P does what you ask; it loads a waveform into digital storage and returns the point at the peak of the largest signal.

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 when more than one of these message units is executed in the same message:

```
CURVE
CURVE?
POINT (if Y argument omitted)
FIBIG
LFTNXT
RGTNXT
FMAX
FMIN
```

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 8-3. 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.

Order-Dependent Conflicts

Conflicts in the use of the buffer take place depending on the order in which waveform processing and I/O occurs. The CURVE query and display data point commands, by contrast, simultaneously load the buffer as they execute. 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 494P 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.

Here is an example combining both CURVE output and input in the same message; this is a way to talk to yourself.

```
100 REMARK WRONG WAY
110 DIM B(1000)
120 PRINT @Z:"CURVE?;CURVE ";A;
130 INPUT @Z:B
```

This program attempts to obtain a 494P waveform and replace it with a waveform residing in controller array A. But that's not what happens. The 494P does buffer the CURVE data transmitted from array A by line 120, but then the 494P overwrites the data in the buffer when it executes the message in line 120. This occurs because the CURVE query is executed first, transferring the contents of digital storage to the buffer. When the 494P executes the CURVE command that follows, it writes the contents of the buffer back into digital storage. As a result, the controller gets the digital storage waveform it requested and stores it in array B (as it executes line 130). However, the data from array A is lost and does not replace the original digital storage waveform.

Instead of the previous example, try

```
100 REMARK RIGHT WAY
110 DIM B(1000)
120 PRINT @Z:"CURVE?"
130 INPUT @Z:B
140 PRINT @Z:"CURVE ";A;
```

Line 120: Requests a curve, which the 494P 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 in some 4050-Series controllers. With this semicolon, the controller places a space between numbers; the 494P accepts the space (or other format characters), as well as a comma, for a delimiter.

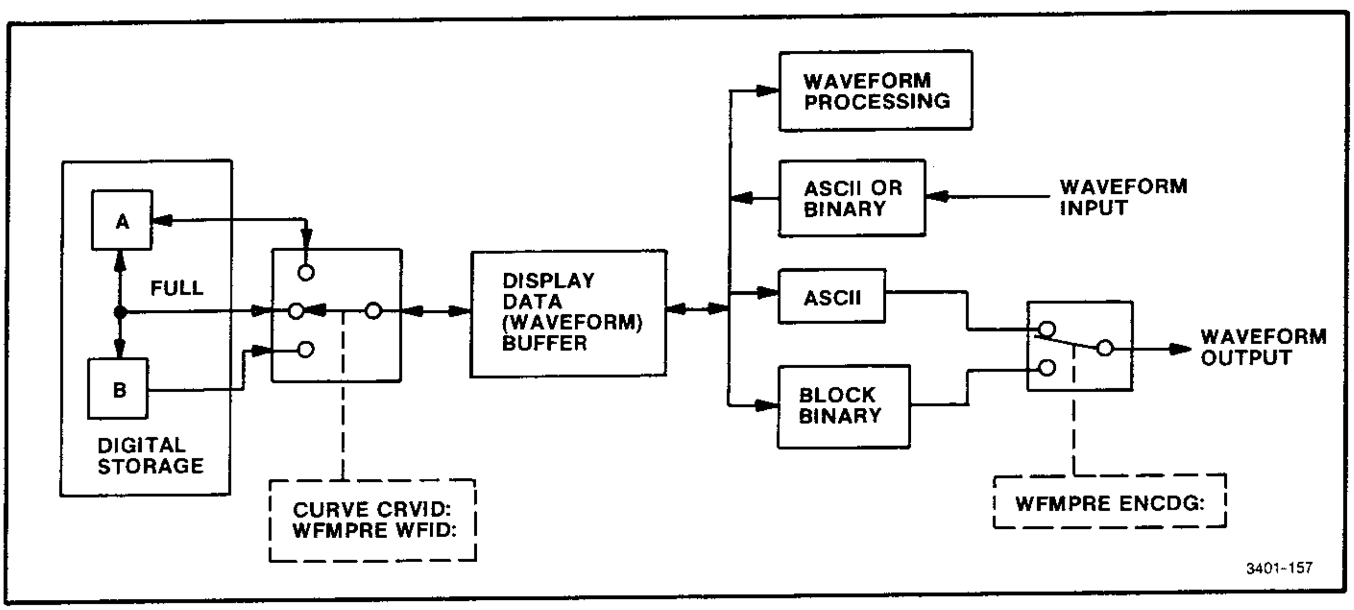


Figure 8-3. How multiple use of the display data buffer is controlled.

FINDING SIGNALS WITH 494P WAVEFORM PROCESSING

The waveform processing resident in the 494P packs a lot of power into a portable analyzer. This power can be better realized 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 suggest how to apply 494P waveform processing in your application with more accurate and predictable results.

Understanding How Waveform Processing Works

The signal-finding commands (FIBIG, LFTNXT, and RGTNXT) are programmed to recognize a shape in the stored waveform that is characteristic of a continuous wave (cw) signal. This means complex signals, such as those created by frequency modulation, may be overlooked, depending on their relative amplitude and spacing. (This limitation also relates to phase or frequency noise.) It may be necessary to transfer the entire waveform and process these signals externally.

Two other factors affect how the signal finding commands perform. The first is separation. Because the shape of the signal response is an important factor, a definite notch must be present between adjacent signals for both to be recognized. The second factor is noise. Because the signal search routine is sensitive enough to detect small signals, it also detects noise peaks that appear to be small signals. Both of these factors are subject to your control, so you can improve the results of waveform processing by practicing the following suggestions.

Once a signal is found, the analyzer can be instructed to change its center frequency to match the signal at the display data point. The analyzer can also be instructed to change its reference level to match the amplitude of the signal at the display data point. The commands that do this (CENSIG and TOPSIG) rely on the span/division and vertical display scale factors when computing how far to change the center frequency and reference level, respectively. The accuracy of span/division and vertical display, along with the accuracy of REFLVL, determine how closely the signal peak is moved to the center and top of the graticule. You may apply the waveform processing commands again for greater accuracy.

Acquiring Data for Waveform Processing

The results of waveform processing depend to a great extent on the data in digital storage. Both the resolution and noise factors mentioned previously can be controlled during data acquisition.

Signal resolution can be improved by selecting a narrower resolution bandwidth (RESBW command). You may need to slow the sweep so the data is calibrated (done automatically in the TIME AUTO mode).

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, setting the cursor above the noise by turning the front-panel knob. Further smoothing is possible by slowing the sweep (TIME command) so that the number of data averaged for each point in digital storage is increased.

Noise peaks can also be reduced (smoothed) by the video filters. The narrow video filter (VDFLT NARROW) is recommended for acquiring data for most waveform processing applications.

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.

Spectrum Search

The RGTNXT and LFTNXT commands support spectrum search applications. Begin with the display data point at the left edge of the screen (POINT 1). Acquire a waveform with the SIGSWP command; don't update digital storage with successive sweeps while the search is under way. Let successive RGTNXT commands pick off the signals on the display. To continue, tune up in frequency by an amount equal to the span/div multiplied by 10, take a single sweep, and continue with RGTNXT.

LFTNXT could be used by starting at the right of the screen (POINT 1000) and tuning down in frequency, rather than up.

A way to let the 494P loop through a message that uses RGTNXT to search for a spectrum is presented in connection with REPEAT, later in this section.

Measuring Signal Frequency with COUNT

To measure the frequency of a signal, center the signal on screen. This can be done by using a FMAX;CENSIG combination. The signal level at center screen (point 500) 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 REMARK - 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, 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 REMARK - Code to handle low signal level condition
```

Higher Center Frequency Drift Rate After Tuning

After a large center frequency change (more than a few 100 MHz), oscillator drive will cause center frequency errors. This is especially noticeable using slow sweep speeds when the time is long between the end of sweeps and the frequency corrections. Thus, delay time, in addition to that built into the 494P, may be needed to allow the oscillator drift to reduce in magnitude.

USING REPEAT FOR SIGNAL TRACKING AND SEARCHES

The REPEAT command adds another dimension to 494P messages; the number of times to loop-through a string of commands or queries. Several uses are suggested here.

Tracking a Signal

The 494P can track a signal while keeping it centered on the display. To do this, the 494P updates its frequency. At the same time, waveform processing is used to find the peak value after each sweep and center it on the display. The REPEAT command causes the 494P to execute the waveform processing message repeatedly.

```
100 REMARK TRACK A SIGNAL
110 PRINT @Z:"WAIT;FMAX;CENSIG;SIGSWP;REPEAT
    10E6"
```

Line 110: Causes the microcomputer to wait until the sweep is completed before it processes the digital storage waveform, then it arms the sweep to acquire a new waveform. (This WAIT ... SIGSWP sequence does not hang the first time through the loop, even if a sweep is not in progress or the instrument is not in single-sweep mode; see WAIT in Section 7 for more details.) The REPEAT command causes the microcomputer to continue to execute the message.

This REPEAT loop can be aborted by a DCL interface message. Even though the microcomputer is busy executing the loop, it recognizes DCL, stops executing the message, and flushes it from the input buffer.

NOTE

Use DCL to abort a REPEAT loop. Pressing RESET TO LOCAL does not abort the loop. It does, however, prevent execution of front-panel commands, if any, in the loop, which causes execution error status to be reported.

REPEAT can also be applied to a similar task; keeping a signal centered as span/division is reduced to focus on the signal. Here's a routine that quickly spans down from 100 kHz to 10 kHz.

```
100 REMARK SPAN DOWN
110 PRINT @Z: "SIGSWP"
120 PRI @Z: "CEN,SPA DEC;SIG;WAI;POI 400;RGT
    125;REP 5"
```

Line 110: Guarantees the analyzer is in the single-sweep mode. This routine assumes the signal is already identified by the display data point (as a result of FIBIG, RGTNXT, etc., or POINT). If the signal of interest is centered, adding POINT 500 to the message in line 110 would do the job.

Line 120: Centers the signal, steps the span down once, arms the sweep, and finds the signal in the new data. A threshold value of center screen is used (rather than video filtering) to overcome noise in the data. This speeds up the sweep, but requires that the signal be less than four divisions below the reference level. The mnemonics in line 120 are squeezed to three letters to save space on the line. REP 5 at the end causes the message to be executed a total of six times, spanning down by two decades.

Spectrum Search Using REPEAT

The 494P 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 494P and input the results. The following routine works on a waveform in digital storage that is not updated during processing.

```
100 REMARK SPECTRUM SEARCH
110 PRINT @Z: "POINT 0"
120 PRINT @Z: "RGTNXT;POINT?;REPEAT 20"
130 REMARK*****************
140 REMARK INSERT ANY OTHER TASKS
150 REMARK******************
300 DIM P(20,2)
310 INPUT @Z:P
```

Line 120: The 494P 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 494P. The buffer can handle 176 (FRE?;REP 175) responses to the FREQ query, which includes the frequency in scientific notation; but, the buffer can handle 293 (PHS?;REP 292) of the shorter PHSLK query responses (PHSLK ON or PHSLK OFF). Messages and responses share buffer space. Long messages will leave less space for responses than short messages.

MESSAGES ON THE CRT USING RDOUT

The 494P accepts either a single set or a double set of quote marks to delimit the crt message. With 4050-Series controllers, use a single set of quotes around the message inside the RDOUT command.

```
100 PRINT @Z: "RDOUT 'SET THE PEAK/AVERAGE
    KNOB'"
```

This is necessary because the 4050-Series controller 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 494P 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, as shown in Figure 8-4.

The RDOUT message continues to be displayed if the 494P remains under remote control. To demonstrate the above messages by themselves, add the statement

```
110 GOTO 110
```

To scroll the RDOUT message to the top of the 494P screen, insert

```
105 PRINT @Z: "RDOUT ''"
```

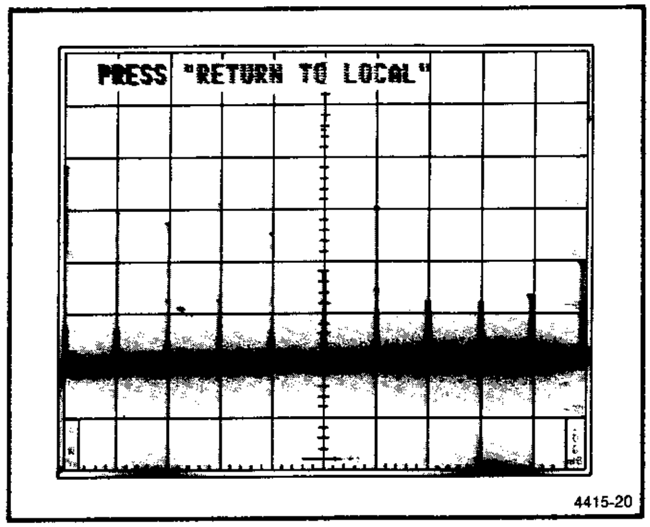


Figure 8-4. Quote marks can be used in messages on the 494P crt.

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 amplitudes of the resolution bandwidth filters. The CAL command allows these functions to be separated so that either the adjustments need to be made, 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. The order of the first four adjustments is AMPL, HPOS, VPOS, and LOG. User prompt messages are displayed on the crt screen. Since the 494P 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 494P 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. The program must contain a message instructing the operator to connect the calibrator.

```
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 WBY@20:
160 PRINT @Z: "CAL HPOS"
170 INPUT B
180 WBY@20:
190 PRINT @Z: "CAL VPOS"
200 INPUT B
210 WBY@20:
220 PRINT @Z: "CAL LOG"
230 INPUT B
240 WBY@20:
250 PRINT @Z: "CAL AUTO"
```

COMPARING THE STATUS BYTE AND THE ERR?/EVENT? RESPONSE

The 494P status byte and ERR?/EVENT? responses described in Section 7 play complementary roles in GPIB system programming. The status byte is the 494P response to a serial poll. The ERR?/EVENT? response is the 494P 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?/EVENT? response details the cause of abnormal status; i.e., what kind of error or warning prompted the 494P to assert SRQ and report abnormal status.

Status bytes and EVENT? responses 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).

The status byte is cleared by a serial poll of the instrument. Event codes are cleared when the event code is reported. Reading the status byte does not clear the error codes, and vice versa. DCL and SDC (if addressed) clear both the status byte and event codes.

FIRMWARE OPERATING NOTES

Following are exceptions to normal instrument operation that relate to the different firmware versions. The instrument displays its version number for approximately 3 seconds whenever instrument power is turned on or the RESET TO LOCAL/REMOTE pushbutton is pressed.

Version 2.2: Plotting on a 4662 Option 31 Plotter

The 4662 Option 31 plotter does not respond to the move command issued by the 494P immediately following the initial pen selection or a pen change command. This results in the pen drawing from its "home" position rather than the selected location. To avoid this, use only a one-pen (one-color) configuration or manually select the second-color pen to be used.

Follow these steps to make a two-color plot.

1. Switch GRAT ILLUM on and turn the crt READOUT and Digital Storage off.

2. Manually select a pen and plot the information.

3. Turn off the GRATicule ILLUMination and turn on the crt READOUT and Digital Storage.

4. Manually select the second-color pen and plot the stored display and crt readout.

Version 2.2: Auto Peaking Function with External Mixers

When using external mixers, the automatic peaking function does not always succeed in setting the correct peak value. To be sure of having the correct value, set peaking manually with the front-panel PEAK/AVERAGE control or use the PEAK command with specific numeric arguments.

EXECUTION AND TRANSFER TIMES

The 494P microcomputer system typically takes 10 to 25 ms to execute commands received over the bus. This is the time the 494P 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 microcomputer and hardware or a wait to allow hardware response. These cases are noted in Table 8-1.

Table 8-1
EXECUTION TIMES

Command	Time
FREQ	
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	
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
DEGAUSS (cf=100 MHz)	
@ 5 MHz/div	700 ms
@ 50 kHz/div	820 ms
@ 500 Hz/div	980 ms
EXMXR and FRQRNG if input 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

Table 8-1 (cont)

Command	Time
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
CAL AUTO	25 s
PEAK AUTO	10.6 s @ 10 ms/div
DSTORE (display)	145 ms
DRECAL (display)	290 ms
STORE (settings)	14 ms
RECALL (settings)	425 ms
CURVE	100 ms
CURVE?	60 ms
POINT (X argument only)	56 ms
FIBIG	760 ms
LFTNXT, RGTNXT	100 ms * (signal separation in div)
FMAX, FMIN	83 ms
SET? (command execution time)	520 ms to 6.8 s
INIT	500 ms
TEST	6 s

Because of the way the 494P handles output, the microcomputer is free after it loads an output buffer. The additional time for the transfer is related to the listener for cases where the 494P is faster. For instance, with 4051 and 4050-Series controllers, the following transfer times have been observed (Table 8-2).

Table 8-2
TRANSFER TIMES

Transfer	4041	4051	4052
Data Output			
SET? response	480 ms	480 ms	480 ms
CURVE? response			
Binary (input as numbers)	2.26 s	2.6 s	830 ms
ASCII (input as a string)	7.1 s	7.4 s	7.1 s
ASCII (input as numbers)	9.5 s	11 s	7.8 s
Display Data Input			
CURVE			
Binary (from number array	1.4 s	1.1 s	1.1 s
ASCII (as a string)	9.5 s	8.1 s	8.1 s
ASCII (as numbers)	12.4 s	20.0 s	11.5 s

IEEE STD 488 (GPIB) SYSTEM CONCEPTS

INTRODUCTION

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., 345 East 47th Street, New York, New York 10017).

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 a standard accessory to the 494P, 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. Connectors may be rigidly stacked, using standard counterbored 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.

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

2. Logical 0 is a false state, high voltage level ($\geq +2.0$ 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.

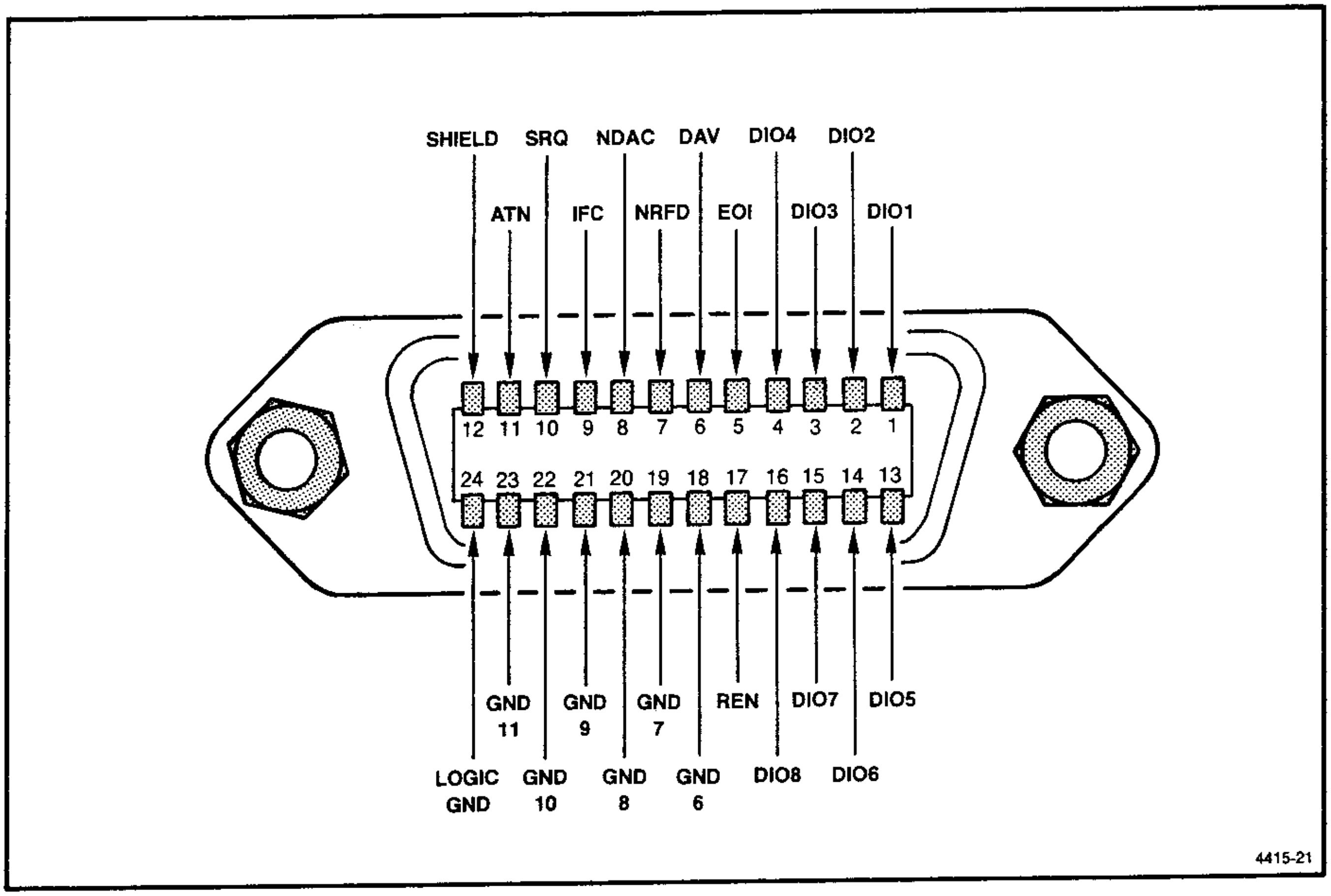


Figure A-1. IEEE Std 488 (GPIB) connector.

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.

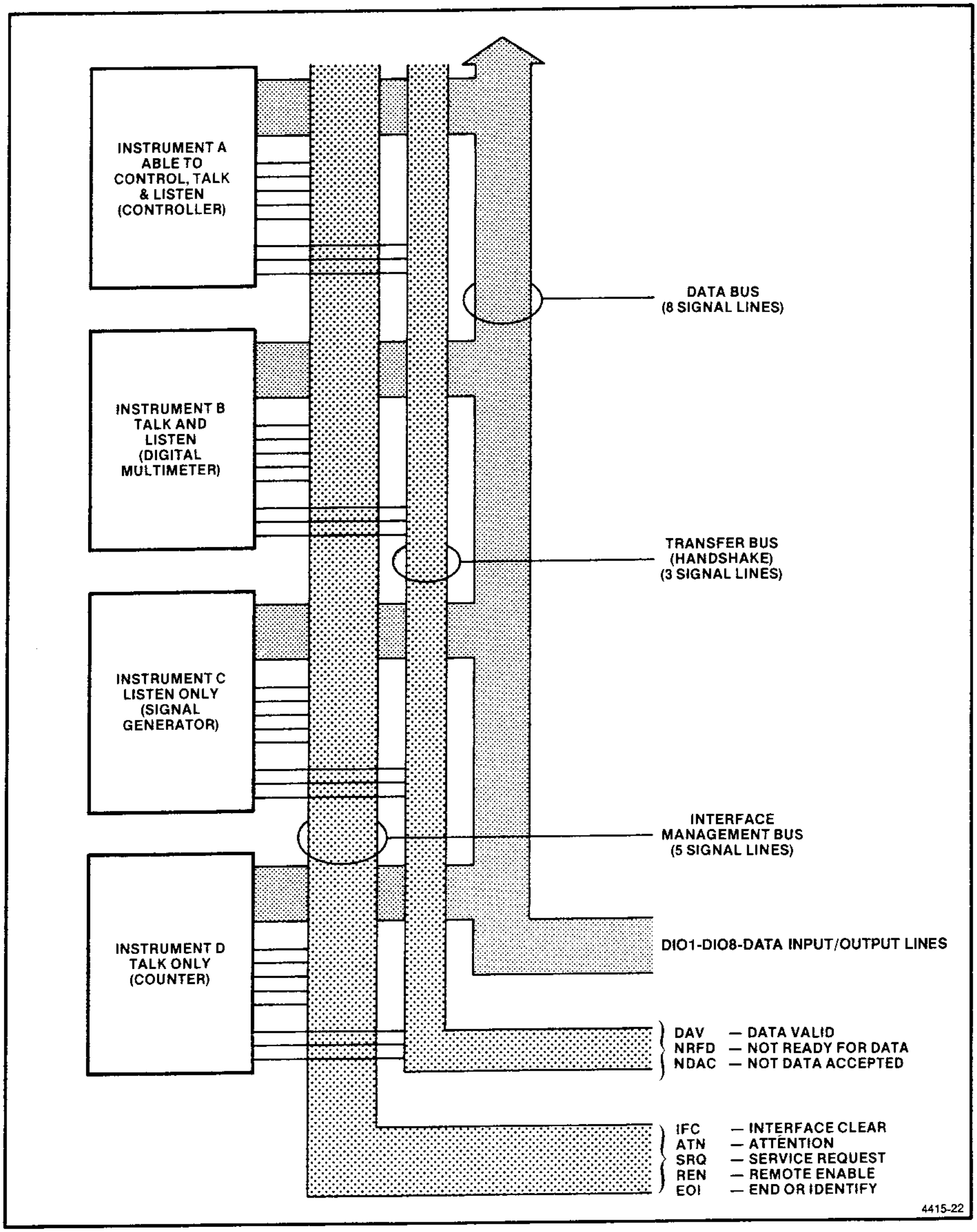


Figure A-2. A typical GPIB system.

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.

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 of Figure A-3.

Interface control messages (refer to Table A-2) include 1) the primary talk and listen addresses for instruments on the bus; 2) addressed commands (only instruments previously addressed to listen will respond to these commands); 3) universal commands (all instruments, whether they have been addressed or not, will respond to these commands); 4) 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.

B7 B6 B5 →	Ø Ø Ø	Ø Ø 1	Ø 1 Ø	Ø 1 1	1 Ø Ø	1 Ø 1	1 1 Ø	1 1 1
BITS 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.

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.

The standard recommends that the alphanumeric codes associated with the numbers, symbols, and upper case characters (decimal 32 to 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).

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 asychronous 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.

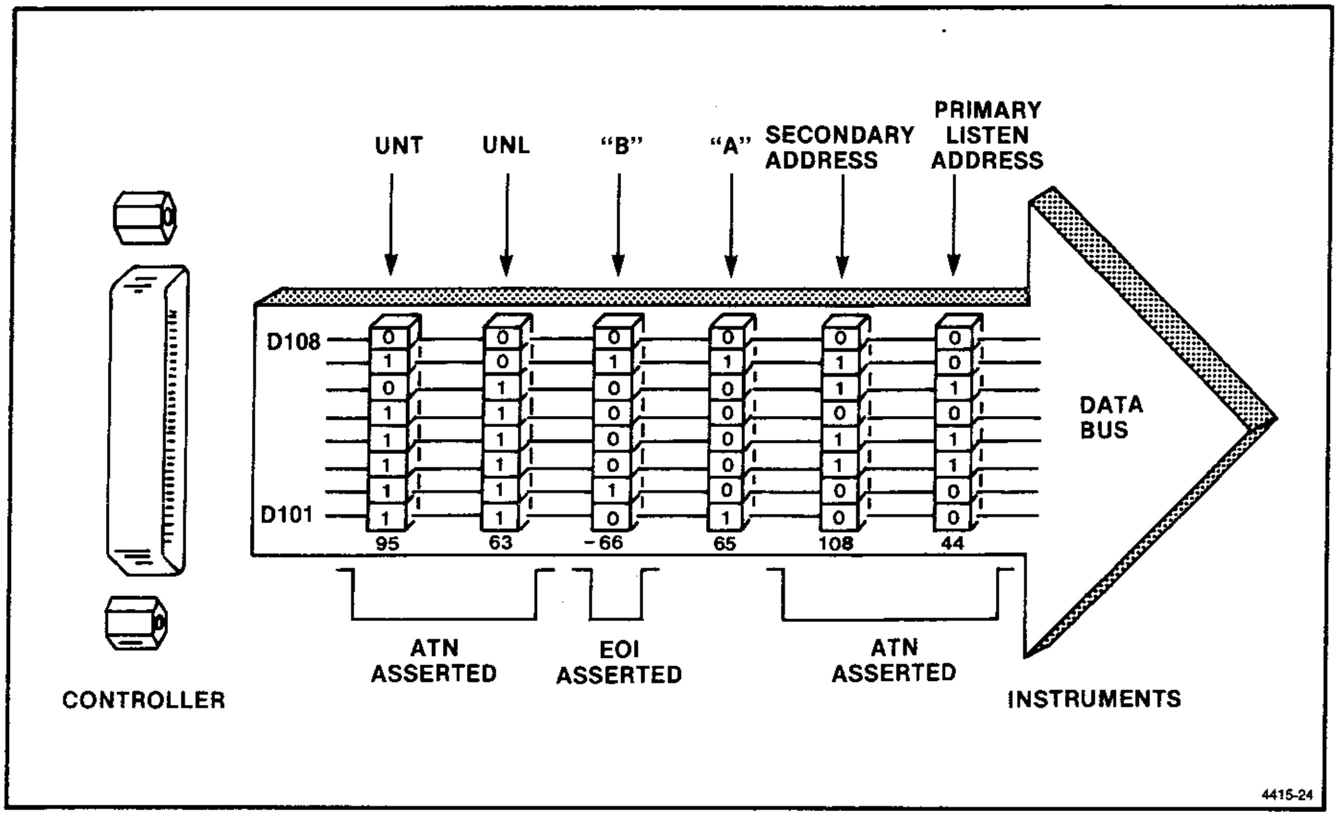


Figure A-4. An example of data byte traffic on the GPIB.

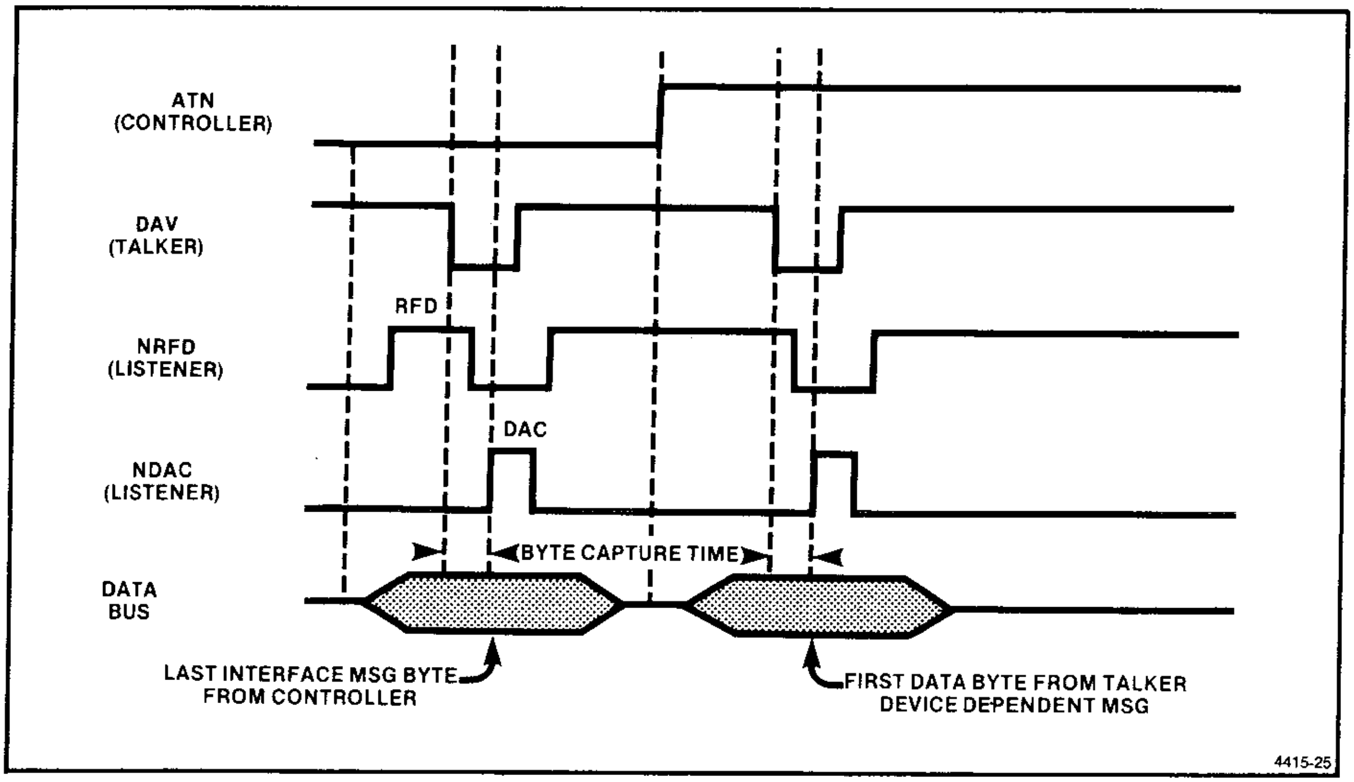


Figure A-5. A typical handshake timing sequence (idealized). Byte capture time is dependent on the slowest instrument involved in the handshake. RFD means Ready For DATA; DAC means Data Accepted.

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 $> 100\,\mu s$ 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(s) can send and receive data over the bus.

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

Introduction

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. In the 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 asychronous 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 ns.

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 as noted 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.

DCL (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 1) send primary talk and listen addresses, secondary addresses, universal commands, and addressed commands to all instruments on the bus; 2) respond to a service request message (SRQ) from an instrument; 3) 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. This is true 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. This gives 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 asychronous 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) does 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).

494P COMMANDS AND QUERIES

Front Panel (refer to Figure B-1)

Header	Argument	Function
AREs	ON, OFF	AUTO RESOLN bandwidth
AVIew	ON, OFF	VIEW A
BMInA	ON, OFF	B−SAVE A
BVIew	VIEW B	
CAL	AUTo, AMPl, LOG, HPOs, VPOs	<SHIFT> CAL
CLIp	ON, OFF	BASELINE CLIP
CNTcf		COUNT → CF (no query)
COUnt	ON, OFF	COUNTER
CREs	NUM	COUNT RESOLN
CRSor	KNOb, PEAk, AVG	PEAK/AVERAGE cursor
DEGaus		Degaus tuning coils (no query)
DELfr	ON, OFF	ΔF
DIScor	ON, OFF	<SHIFT> FREERUN
DREcal	A, B	<SHIFT> RECALL (no query)
DSTore	A, B	<SHIFT> STORE DISP (no query)
EXMxr	ON, OFF	<SHIFT> EXT MIXER
FINe	ON, OFF	FINE reference level steps
FIRst	NUM	1ST LO
FREq	NUM	CENTER FREQUENCY
FRQrng	NUM, INC, DEC	FREQUENCY RANGE
GRAt	ON, OFF	GRAT ILLUM
HELp?		HELP
IDEnt	ON, OFF	IDENT
Xpwr	NUM, INC, DEC	Maximum input power
MINatt	NUM, INC, DEC	MIN RF ATTEN dB
MXHld	ON, OFF	MAX HOLD
MXSpn	ON, OFF	MAX SPAN
PEAk	AUTo, NUM, INC, DEC, KNOb	MANUAL PEAK
PLOt?		<SHIFT> PLOT
PLStr	ON, OFF	PULSE STRETCHER
POFset	NUM	<SHIFT> B−SAVE A
PTYpe	TK4662,TKOP31, HP7470	<SHIFT> SAVE A
RECall	NUM	RECALL SETTINGS (no query)
REDout	ON, OFF	READOUT
REFlvl	NUM, INC, DEC	REFERENCE LEVEL
RESbw	NUM,AUTo, INC, DEC	RESOLUTION BANDWIDTH
RLMode	MNOise, MDIst	MIN NOISE/MIN DISTORTION
SAVea	ON, OFF	SAVE A
SECond	NUM	2ND LO
SIGswp		SINGLE SWEEP
SPAn	NUM, INC, DEC, MAX	FREQUENCY SPAN/DIV
STOre	NUM,	<SHIFT> STORE (no query)
TIMe	NUM, AUTo, INC, DEC, MAN, EXT	TIME/DIV
TRIg	FRErun, INT, LINe, EXT	TRIGGERING
TUNe	NUM	Incremental frequency change (no query)
VIDflt	OFF, WIDe, NARrow	VIDEO FILTER
VRTdsp	LOG: NUM, LIN: NUM, INC, DEC	VERTICAL DISPLAY
ZERosp	ON, OFF	ZERO SPAN

Display Data I/O

Header	Argument	Function
CURve	CRVid and integers or block binary	Display curve I/O
DCOpy?		Copy display
DPRe?		Display preamble
LORdo?		Lower crt readout
RDOut	Up to 40 characters or NORmal	Crt readout message (no query)
TEXt	SHOrt, LONg	Change readout mode
UPRdo?		Upper crt readout
WAVfrm		Waveform
WFMpre	WFId and ENCdg	Waveform preamble

System

Header	Argument	Function
ALLev?		All events
DT	ON, OFF	Device triggered enable
EOS	ON, OFF	End-of-sweep SRQ control
ERCnt?		Error count
ERR?		Errors
EVEnt?		Event information
ID?		Identify instrument
INIt		Initialize settings
NUMev	NUM	Number of events
REPeat	NUM	Repeat execution
RQS	ON, OFF	Turn on or off abnormal SRQ mask
SET?		Request instrument settings
TESt?		Internal test
WAIt		Wait for end of sweep (no query)

Waveform Processing

Header	Argument	Function
CENsig		Center signal
FIBig	Threshold NUM	Find biggest signal
FMAx		Find maximum value
FMIn		Find minimum value
LFTnxt	Threshold NUM	Find left-next signal
POInt?	X NUM, Y NUM	Display data point
POInt?		Only waveform processing query
RGTnxt	Threshold NUM	Find right-next signal
TOPsig		Change reference level to move signal to top of graticule

NOTES

Only the first three letters of a mnemonic are required; e.g., **ARE** for **AREs**.

Form a query by adding a question mark to the header of a Display Data I/O, System, or Front-Panel header (**AREs?**) unless no query is indicated. **POInt?** is the only Waveform Processing query.

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.

INTERFACE MESSAGES

DCL (20) — Clears I/O buffer and status byte
GET (8) — Aborts and then rearms sweep
GTL (a) — Go to local control
IFC (IFC line) — Initializes talker and listener functions
LLO (17) — Lock 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

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

Bits 4–1: Four-bit status code
Bit 5: 494P microcomputer busy condition
Bit 6: Abnormal (1)/normal (0) condition
Bit 7: SRQ is asserted (depends on RQS and EOS commands)

Power-on is reported only if an internal switch is set to request this status.

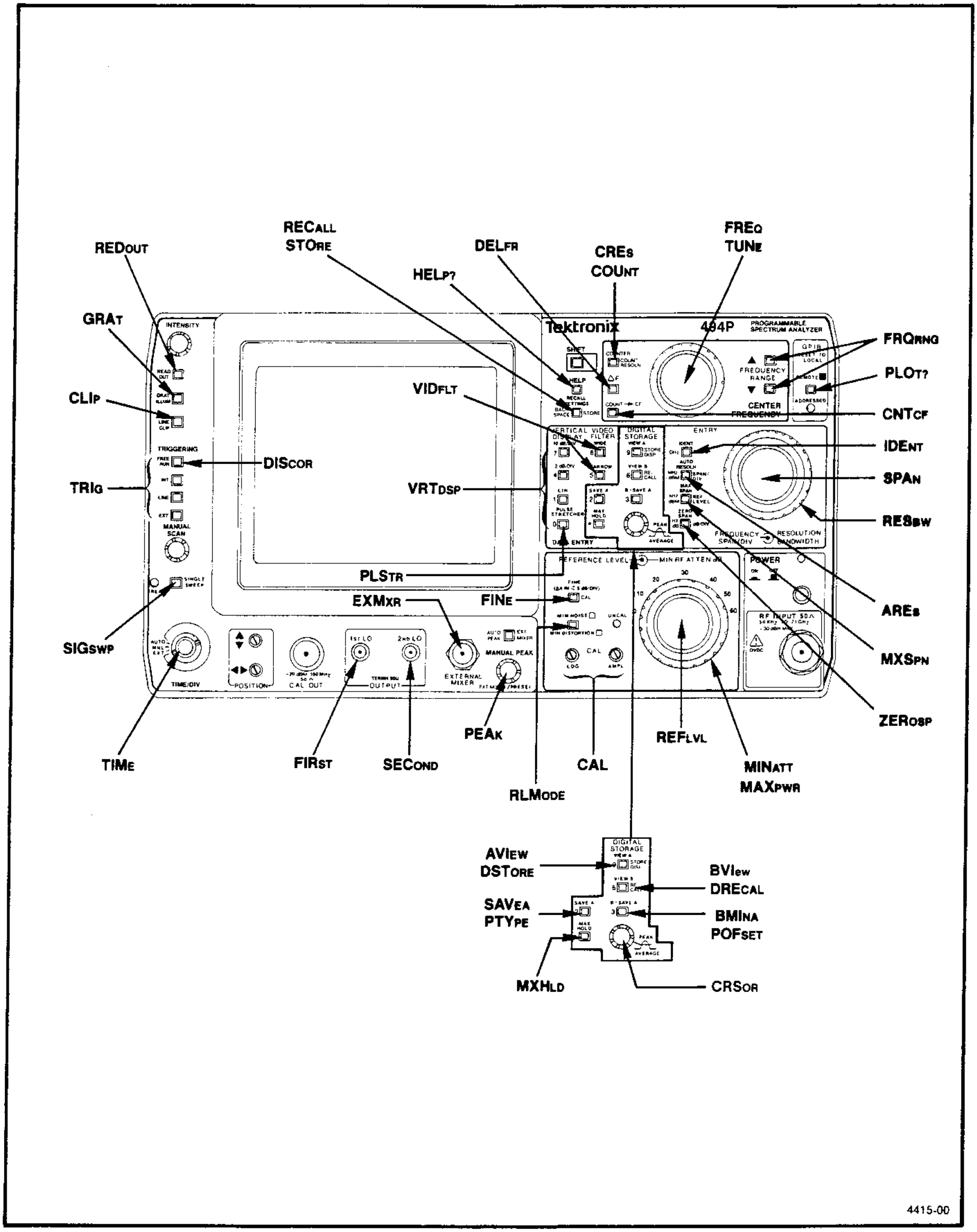


Figure B-1. 494P Front-panel controls.

ERR? RESPONSE

Command Errors

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 for this header
11	Numeric argument not allowed for this header
12	String argument not allowed for this header
13	Binary argument not allowed for this header
14	Link not allowed for this argument
20	Special argument type not recognized
21	Special argument not allowed for this header
22	Character argument not recognized
24	Input buffer overflow

Execution Errors

26	Output buffer overflow; remaining output lost
27	Attempt to execute command in local mode
28	FREQ, TUNE, FIRST, or SECOND out of range
29	FRQRNG out of range
30	CRES out of range
31	SPAN out of range
32	RESBW out of range
33	MAXPWR or MINATT out of range
34	REFLVL out of range
35	VRTDSP LIN out of range
36	VRTDSP LOG out of range
37	TIME out of range
39	IDENTify not allowed in this span/div
40	Signal finds not allowed in this span/div
45	GET (Group Execute Trigger) ignored (not executed)
46	NUMEV out of range
47	STORE, RECALL, DSTORE, or DRECAL out of range
48	PHSLK cannot be turned OFF/ON directly with PHSLK command

Execution Warnings

49	FREQ change caused EXMXR change
50	SPAN defaulted to MAX
51	SPAN defaulted to 0
52	UNCAL light turned on
53	Multiple use of display buffer
54	UNCAL light turned off

Internal Warnings

55	Frequency reference changed to INT
56	Frequency reference changed to EXT

Internal Errors

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-powered RAM checksum error
72	1st LO tuning system failed
74	2nd LO tuning system failed
75	Phase lock system failed
78	IF count failed
79	Power supply out of regulation
80	Frequency reference unlocked
82	1st LO tuning system recovered from a failure
84	2nd LO tuning system recovered from a failure
85	Phase lock system recovered from a failure
88	IF count recovered from a failure
89	Power supply regained regulation
90	Frequency reference relocked
99	Unrecognized event occurred

System Events

97	Power just came on
98	Operation complete (end of sweep)

494P APPENDIX B

COMMAND AND QUERY DESCRIPTIONS