Tektronix®
COMMITTED TO EXCELLENCE

PLEASE CHECK FOR CHANGE INFORMATION AT THE REAR OF THIS MANUAL.

496P
SPECTRUM ANALYZER

PROGRAMMERS

INSTRUCTION MANUAL

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

Serial Number ____________

070-3484-00
Product Group 26

First Printing APRIL 1981
REV JAN 1983

PREFACE

About the 496P Programmable Spectrum Analyzer Manuals

Local control and maintenance of the 496P are described in combined 496/496P manuals. These manuals are: 496/496P Operators, 496/496P Operators Handbook (a small manual that fits in the 496P front cover), and 496/496P Service, Volumes 1 and 2.

Refer to the 496/496P Operators Manual for a full description of 496P functions and front-panel controls. The Operators Manual also contains the full specification of instrument performance.

About This Programmer's Manual

This Programmer's Manual describes the programmable functions of the 496P and how to use them for remote operation.

Sections 1 and 2 help you get started using the 496P on the GPIB (General Purpose Interface Bus). Programming examples are given here and 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 help you translate if you are using another controller.

Sections 3 through 7 are a reference to the language used to set and read 496P functions and transfer spectrum data acquired by the 496P. 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-do-it section for making programmable measurements with the 496P.

An appendix helps you understand the GPIB and the IEEE 488 standard on which it is based.

A command summary foldout provides a complete list of remote control messages and features of the 496P.

TABLE OF CONTENTS

		Page
Preface		i
Operators Safety Summary		vii
Service Safety Summary		ix
Section 1	**INTRODUCTION TO GPIB OPERATION**	
	GPIB Controls and Indicators	1-1
	RESET TO LOCAL (REMOTE)	1-2
	ADDRESSED	1-2
	GPIB Function Readout	1-2
	Setting the GPIB ADDRESS Switches	1-3
	Setting the LF or EOI Switch	1-5
	Syntax Diagrams	1-6
	Talk/Listen-Only Operation	1-7
	TALK ONLY, LISTEN ONLY Switches	1-7
	Data Logging	1-8
	Restoring Control Settings and the Display	1-10
	Putting a Counter on the Tape	1-11
	IEEE 488 Functions	1-11
	Connecting To a System	1-13
Section 2	**GETTING STARTED**	
	Setting Programmable Controls	2-1
	4050-Series Graphic Systems	2-2
	Two Other Controllers	2-2
	In General	2-3
	Querying Programmable Controls	2-3
	Listen/Talk Routine	2-5
	Acquiring Instrument Settings with SET?	2-6
	Resetting the 496P and Interface Messages	2-8
	Acquiring a Waveform	2-9
	4050-Series	2-9
	TEK SPS BASIC	2-10
	9825	2-10
	Getting Smarter	2-10
	AutoCal—A Programmable Assist in Calibrating the Front Panel	2-12

TABLE OF CONTENTS (cont)

Section 3 DEVICE-DEPENDENT MESSAGE STRUCTURE AND EXECUTION **Page**

496P Input Messages 3-1
Input Message Format 3-2
Input Buffering and Execution 3-2
Command Format 3-3
Argument Format 3-4
Query Format 3-6
496P Output Messages 3-6
Output Message Format 3-6
Output Message Execution 3-6

Section 4 FRONT-PANEL CONTROL

Frequency 4-2
FREQ (center frequency) 4-3
TUNE (increment frequency) 4-3
DELFR (delta-frequency) 4-4
FRCAL (frequency cal) 4-5
DEGAUS (degauss tuning coils) 4-5
Frequency Span and Resolution 4-6
SPAN (frequency span/division) 4-7
RESBW (resolution bandwidth) 4-8
ARES (automatic resolution bandwidth) 4-9
PHSLK (phaselock) 4-10
ZEROSP (zero span mode) 4-10
Vertical Display and Reference Level 4-12
VRTDSP (vertical display) 4-13
REFLVL (reference level) 4-14
FINE (fine reference level steps) 4-15
RLMODE (reference level mode) 4-16
MINATT (min RF attenuation) 4-17
MAXPWR (maximum input power) 4-18
PLSTR (pulse stretcher) 4-19
VIDFLT (video filter) 4-20
Sweep Control 4-21
TRIG (triggering) 4-22
SIGSWP (single sweep) 4-23
TIME (time/div) 4-24
Digital Storage 4-25
AVIEW, BVIEW (A and B memory display) 4-26
SAVEA (save A memory) 4-27
BMINA (B−A memory display) 4-28
MXHLD (max hold) 4-29
CRSOR (peak/average cursor) 4-30
Display Control 4-31
REDOUT, GRAT, CLIP (display control) 4-32

TABLE OF CONTENTS (cont)

Section 5 **DISPLAY DATA AND CRT READOUT I/O** Page

Waveform Transfers 5-1
WFMPRE (waveform preamble) 5-2
CURVE (display curve) 5-6
CRT Readout Transfers 5-9
RDOUT (readout message) 5-9
UPRDO, LORDO (upper or lower readout) 5-10

Section 6 **WAVEFORM PROCESSING**

POINT (display data point) 6-1
FIBIG, LFTNXT, RGTNXT (signal search) 6-2
FMAX (find maximum value) 6-3
FMIN (minimum value) 6-3
CENSIG, TOPSIG (center or move signal) 6-5

Section 7 **SYSTEM COMMANDS AND QUERIES**

Instrument Parameters 7-1
SET (instrument settings) 7-1
INIT (initialize settings) 7-3
ID (identify) 7-4
Message Execution 7-4
WAIT (wait to execute) 7-4
REPEAT (repeat execution) 7-5
Status and Error Reporting 7-6
EOS (end-of-sweep) 7-6
RQS (request service) 7-7
Status Byte 7-8
Effect of Busy on Device-Dependent Messages 7-9
Effect of Busy on Interface Messages 7-9
ERCNT (error count) 7-10
ERR (error) 7-11
Command Errors 7-11
Execution Errors 7-12
Execution Warnings 7-12
Internal Errors 7-13

Section 8 **HELPS AND HINTS**

Data Acquisition 8-1
Synchronizing Controller and 496P 8-1
Synchronizing with the Sweep 8-2
Using the End-of-Sweep SRQ 8-4
INPUT: An SRQ Alternative 8-4

TABLE OF CONTENTS (cont)

	Page
Binary Waveform Transfers	8-5
Getting 496P Binary CURVE Output	8-5
Sending a Binary CURVE to the 496P	8-6
Scaling, Saving, and Graphing Waveform Data	8-6
Saving the Scaled Array	8-7
Waveform Plotting Made Easy	8-7
Multiple Use of Display Buffer for Waveform Processing and I/O	8-9
Buffer Data Flow	8-9
Order-Dependent Conflicts	8-10
Finding Signals with 496P Waveform Processing	8-11
Understanding How Waveform Processing Works	8-11
Acquiring Data for Waveform Processing	8-12
Spectrum Search	8-12
Using REPEAT for Signal Tracking and Searches	8-13
Tracking a Signal	8-13
Spectrum Search Using REPEAT	8-14
Messages on the Crt Using RDOUT	8-14
What is the Difference Between FREQ and TUNE?	8-15
Power-Up Bus Glitches	8-16
Comparing the Status Byte and ERR? Response	8-16
Execution and Transfer Times	8-16

Appendix A IEEE 488 (GPIB) SYSTEM CONCEPTS

Introduction	A-1
Mechanical Elements	A-1
Electrical Elements	A-1
Functional Elements	A-2
A Typical GPIB System	A-3
Talkers, Listeners, and Controllers	A-3
Interface Control Messages	A-4
Device-Dependent Messages	A-5
Typical GPIB Events When Transferring Data	A-5
GPIB Signal Line Definitions	A-8
Data Transfer Control Bus	A-8
Management Bus	A-9
INTERFACE FUNCTIONS AND MESSAGES	A-10
Introduction	A-10
Remote-Local Function (RL)	A-10
Talker and Listener Functions (T/TE and L/LE)	A-11
Source and Acceptor Handshake Functions (SH and AH)	A-11
Device Clear Function (DC)	A-12
Device Trigger Function (DT)	A-12
Controller (C), Service Request (SR), and Parallel Poll (PP)	A-12
Taking Control (Asychronous or Synchronous)	A-13
Passing Control	A-13
Performing a Serial Poll	A-13
Performing a Parallel Poll	A-14
Programming Summary	A-15

LIST OF ILLUSTRATIONS

Figure No.		Page
	The TEKTRONIX 496P Programmable Spectrum Analyzer	x
1-1	GPIB control and indicators on the front panel	1-1
1-2	Status of GPIB functions indicated when active	1-2
1-3	GPIB ADDRESS switches on the rear panel	1-3
1-4	Effect of message terminator switch for input and output	1-5
1-5	The TEKTRONIX 4924 and 496P in talk/listen-only system	1-6
1-6	Controls used for talk/listen-only data transfers	1-8
1-7	The 496P GPIB port on the rear panel	1-13
1-8	The 496P can be connected in either a star or linear manner	1-14
4-1	Front-Panel Control Commands and Queries	4-1
4-2	Front-Panel Control—Frequency commands	4-2
4-3	Front-Panel Control—Frequency Span and Resolution commands	4-6
4-4	Front-Panel Control—Vertical Display and Reference Level commands	4-12
4-5	Front-Panel Control—Sweep Control commands	4-21
4-6	Front-Panel Control—Digital Storage commands	4-25
4-7	Front-Panel Control—Display Control commands	4-31
8-1	Synchronizing controller and 496P for data acquisition	8-3
8-2	A simple plot of a spectrum acquired from the 496P	8-8
8-3	How multiple use of the display data buffer is controlled	8-10
8-4	Quote marks can be used in messages on the 496P crt	8-15
A-1	Standard IEEE 488 (GPIB) connector	A-2
A-2	A Typical GPIB System	A-4
A-3	ASCII & IEEE 488 (GPIB) Code Chart	A-6
A-4	An example of data byte trafic on the GPIB	A-7
A-5	A typical handshake timing sequence	A-8
A-6	496P Front Panel Controls	A-15

LIST OF TABLES

Table No.		Page
1-1	Bus Address	1-4
1-2	496P IEEE 488 Interface Functions	1-11
8-1	Execution Times	8-17
8-2	Transfer Times	8-17
A-1	Major GPIB Interface Functions	A-3

OPERATORS SAFETY SUMMARY

The general safety information in this part of the 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.

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.

Power source

This product is intended to operate from a power source that will not apply more than 250 volts 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.

For detailed information on power cords and connectors, see maintenance section.

Refer cord and connector changes to qualified service personnel.

Use the Proper Fuse

To avoid fire hazard, use only the fuse of correct type, voltage rating and current rating as specified in the parts list 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.

SERVICE SAFETY SUMMARY

FOR QUALIFIED SERVICE PERSONNEL ONLY

Refer also to the preceding Operators Safety Summary.

Do Not Service Alone

Do not perform internal service or adjustment of this product unless another person capable of rendering first aid and resuscitation is present.

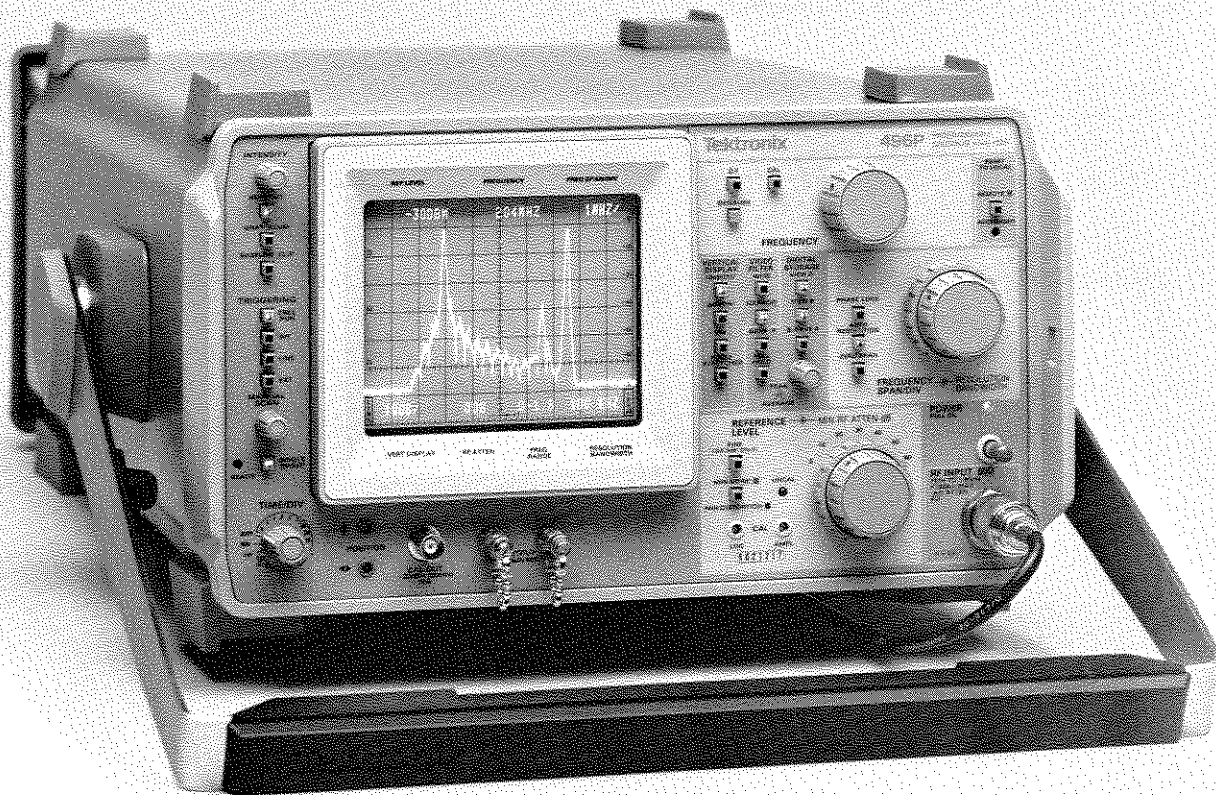
Use Care When Servicing With Power On

Dangerous voltages exist at several points in this product. To avoid personal injury, do not touch exposed connections and components while power is on.

Disconnect power before removing protective panels, soldering, or replacing components.

Power Source

This product is intended to operate from a power source that will not apply more than 250 volts 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.



3484-9

The TEKTRONIX 496P Programmable Spectrum Analyzer.

INTRODUCTION TO GPIB OPERATION

The 496P adds remote control to the performance and portability features of the 496 Spectrum Analyzer. The front panel can be controlled remotely (except for those controls intended for local use only, such as INTENSITY). Some controls are enhanced by remote operation; the SPAN command has greater resolution than the FREQ SPAN/DIV control, for instance, and the vertical display scale factor can be set from 1 to 15 dB/div in 1 dB increments.

The 496P can be programmed for automated spectrum data acquisition and analysis. Waveform processing functions are added to the 496P so it can do some spectrum analysis, such as find a signal, report its position and amplitude, and change analyzer settings to zoom in on the signal.

The GPIB (IEEE Std 488) port added in the 496P allows use with a wide variety of systems and controllers, because the 496P is implemented according to the Tektronix Codes & Formats standard to promote compatibility and ease of operation.

GPIB CONTROLS AND INDICATORS

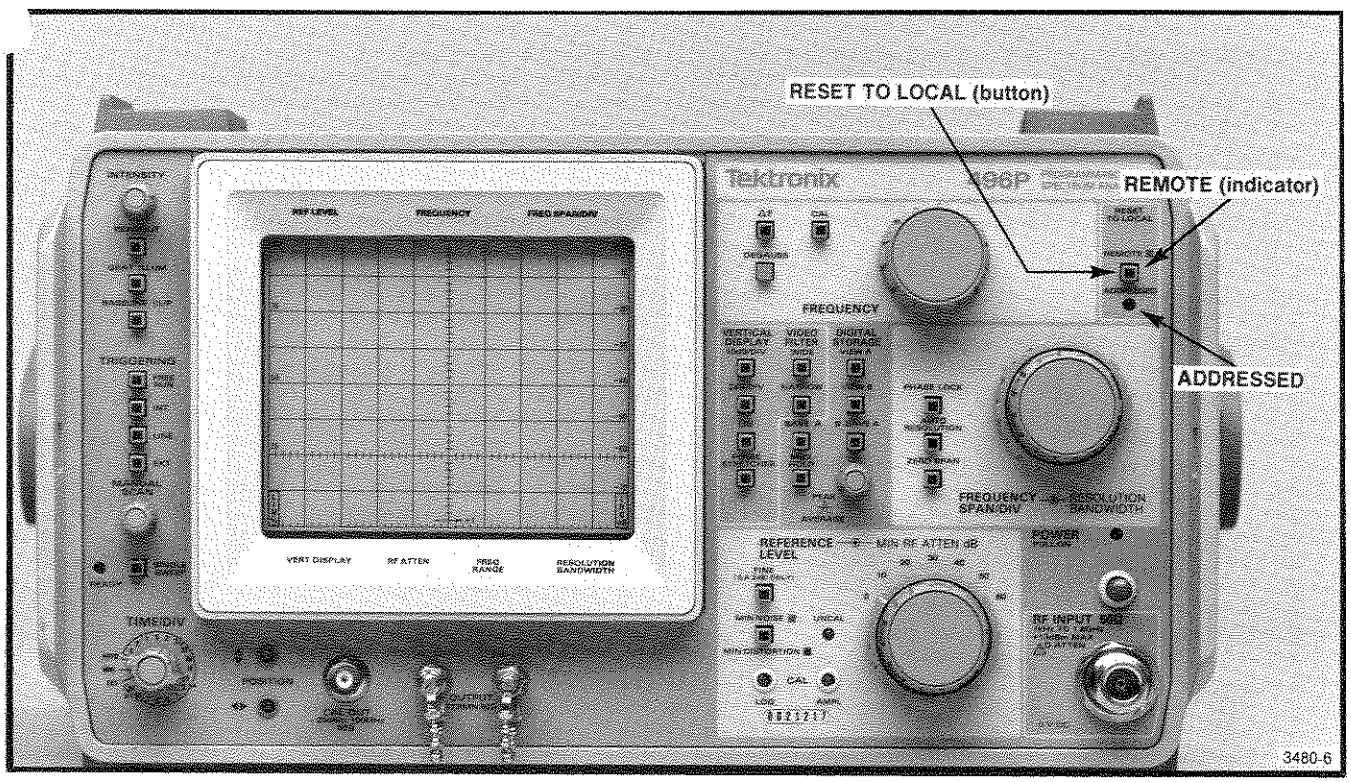


Fig. 1-1. GPIB control and indicators on the front panel.

RESET TO LOCAL (REMOTE)

This button is lighted when the GPIB controller takes remote control of the analyzer. While the 496P is under remote control, its other front-panel controls are not active, but indicators still reflect the current state of front-panel functions.

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

Pressing this button restores local control 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, and PEAK/AVERAGE.

The internal 496P microcomputer flashes the firmware version number and GPIB address on the crt when the button is pressed. This also causes the microcomputer to update the GPIB primary address if the GPIB ADDRESS switches have been changed.

This button has another function in talk-only mode. See Talk/Listen-Only Operation later in this section.

ADDRESSED

Lights when analyzer is addressed to listen or talk.

GPIB Function Readout

A single character appears in the crt readout when the 496P is talking, listening, or requesting service.

The character appears in the position shown in Fig. 1-2, but only while the 496P is addressed to talk or listen or is asserting SRQ.

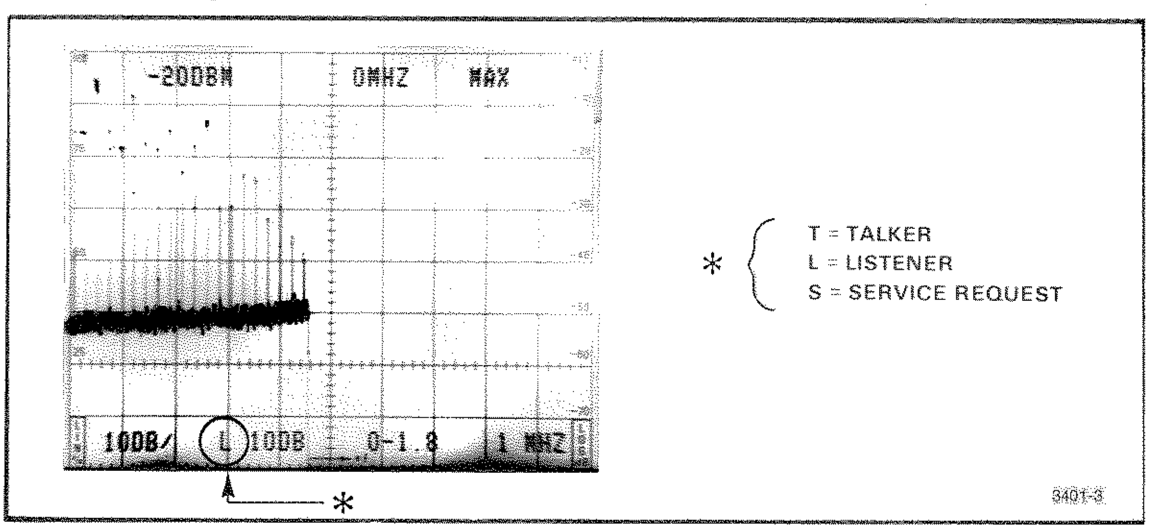


Fig. 1-2. Status of GPIB functions indicated when active.

Setting the GPIB ADDRESS Switches

Switches on the rear panel (Fig. 1-3) set the value of the lower five bits of the instrument's GPIB addresses. The value of these switches is called the instrument's primary address, which corresponds to talk and listen addresses shown in Table 1-1. The 496P microcomputer reads these switches at power-up and again each time the RESET TO LOCAL button is pressed.

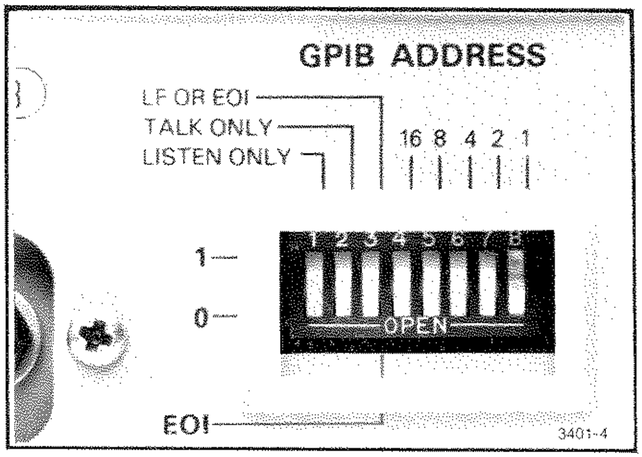


Fig. 1-3. GPIB ADDRESS switches on the rear panel. The LF OR EOI switch (message terminator) and the TALK ONLY and LISTEN ONLY switches are part of the same switch bank.

Table 1-1
BUS ADDRESSES

Switches 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 actually seven bits wide. Bits 1 through 5 are the primary address shown in Table 1-1, while bits 7 and 8 determine whether it is a listen address (32 + primary address) or talk address (64 + primary address). Secondary addresses (both bits 6 and 7 set) are not used by the 496P so are ignored.

Set the switches as desired, but don't use 0 with 4050-Series controllers—they reserve this address for themselves. Selecting a primary address of 31 logically removes the 496P 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 496P is already powered-up, you must press RESET TO LOCAL to cause the microcomputer to update the primary address.

Setting the LF or EOI Switch

A rear-panel switch (also shown in Fig. 1-3) selects the terminator for messages on the bus. If LF OR EOI is selected, the 496P interprets either the data byte LF or the end message (EOI asserted concurrently with a data byte) as the end of a message. If EOI is selected, the 496P 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 496P adds CR and LF (with EOI asserted concurrently) after the last byte of the message. Set to EOI, the 496P 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 for Tektronix controllers. The other position of this switch is provided to accomodate some other controllers, such as the Hewlett-Packard 9825. A change in this switch takes effect immediately.

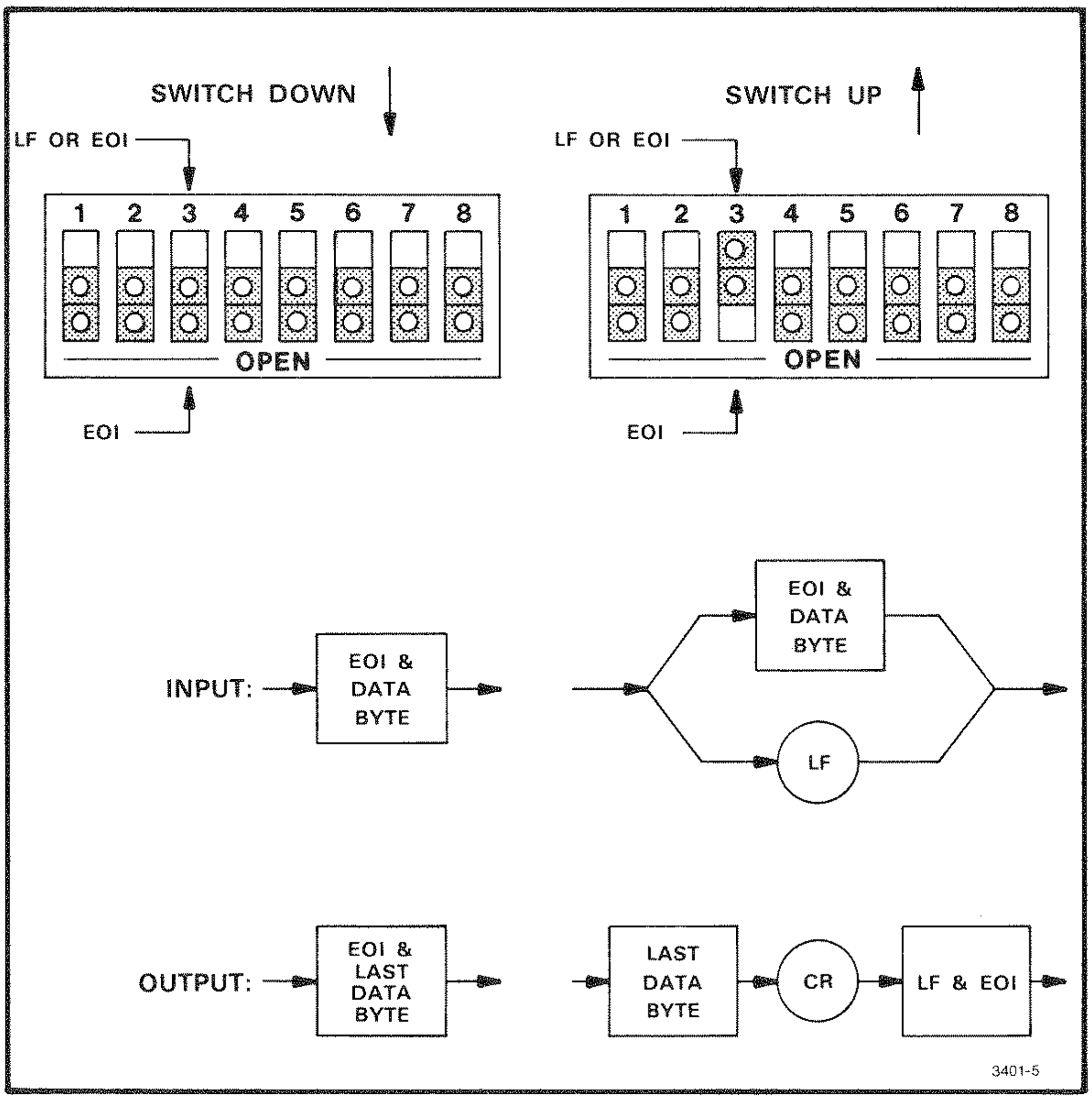


Fig. 1-4. Effect of message terminator switch for input and output.

SYNTAX DIAGRAMS

496P messages are presented 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 for literal (terminal) elements—a character or string of characters that must be sent as shown. Because most mnemonics may be shortened, characters that the 496P requires in a literal element are larger than optional characters.

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

Elements of the syntax diagram are connected together by arrows that show the possible paths through the diagram—the sequence of elements that may 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. Examples of such sequences are:

TALK/LISTEN-ONLY OPERATION

The 496P can be operated as a talker only or a listener only on the GPIB under local control. A simple system requires only the 496P and a talker or listener. Such a system using the 4924 Digital Cartridge Tape Drive is shown in Fig. 1-5.

This system can be used to save spectrum measurements for later display on the 496P or analysis by a controller. You can also use this system to save and restore analyzer control settings.

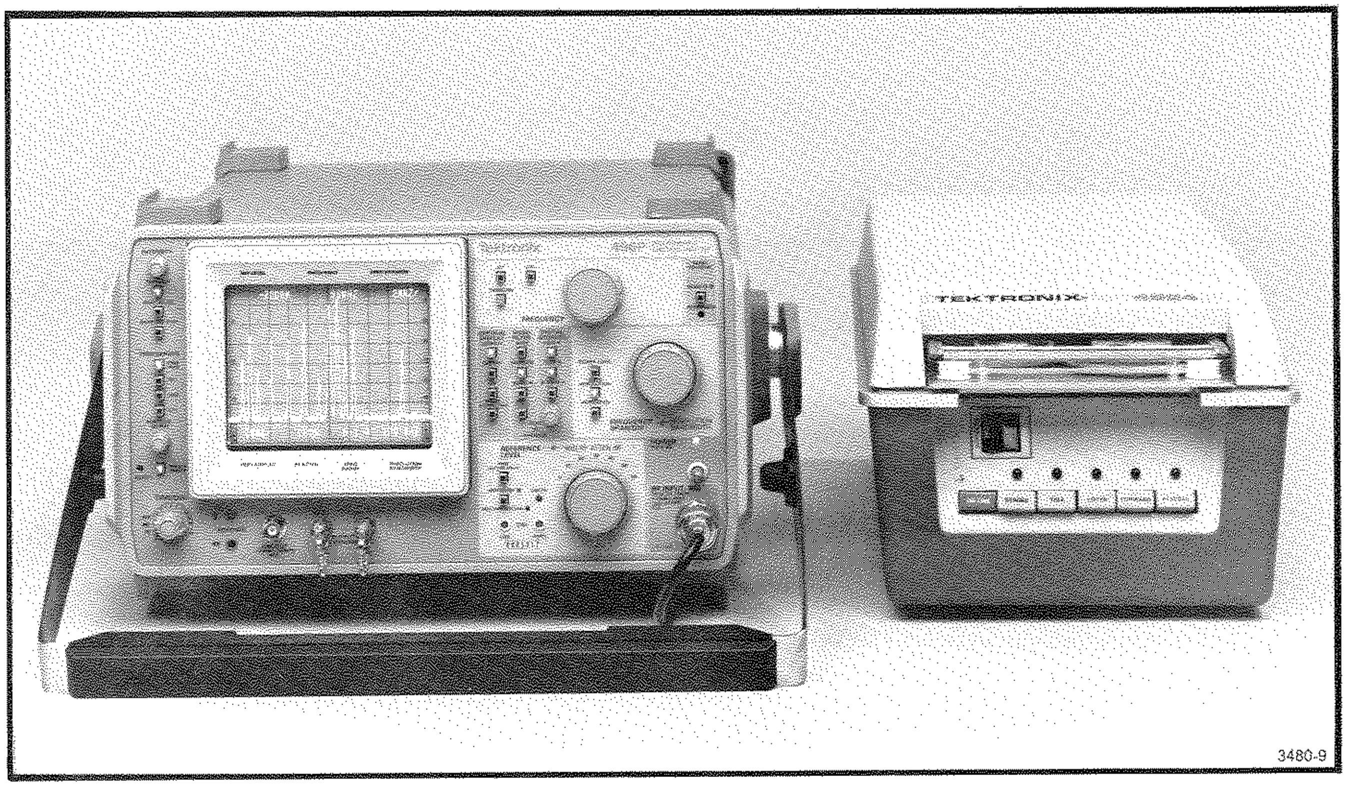


Fig. 1-5. The TEKTRONIX 4924 Digital Cartridge Tape Drive and 496P Programmable Spectrum Analyzer in a talk/listen-only system.

TALK ONLY, LISTEN ONLY Switches

The 496P switches for talk-only and listen-only operation are part of the GPIB ADDRESS switch bank (Fig. 1-3). Set either or both switches—an extension of the IEE 4888 standard allows you to enable both talk-only and listen-only operation. If 496P power is already on, press RESET TO LOCAL to cause a change in these switches to take effect.

Set the LF OR EOI switch to EOI 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 496P from the bus.

The MODE CONTROL switches on the 4924 rear panel must be set as a pair to operate with the 496P. Set SW1 to ON and SW2 to OFF (same as for operating with the 4051) or set both switches to the same position (both SW1 and SW2 ON or OFF).

Data Logging

With the TALK ONLY switch set, you can write spectrum data onto a tape in the 4924 using the controls shown in Fig. 1-6.

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

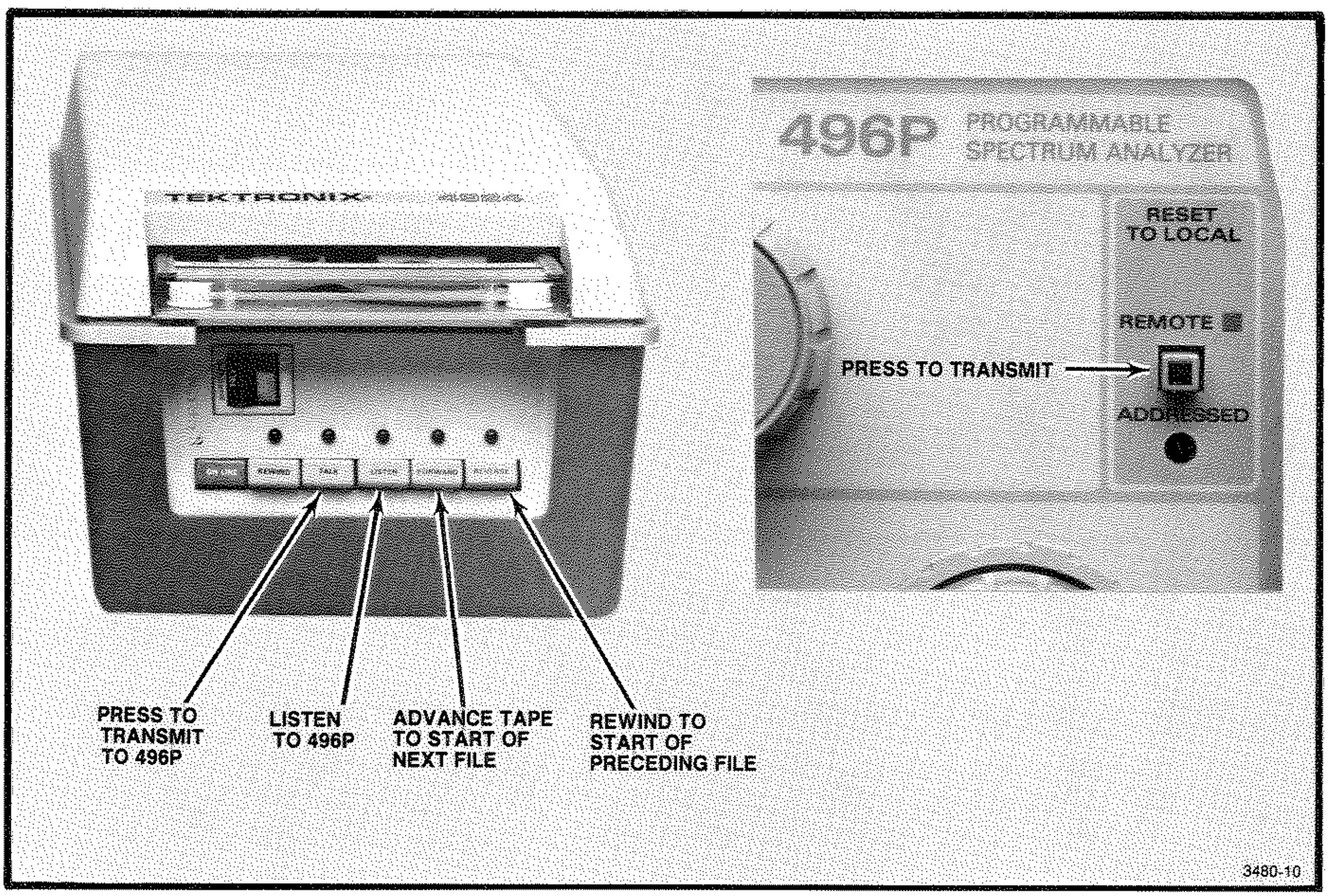


Fig. 1-6. Controls on the 4924 and 496P used for talk/listen-only data transfers.

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

2. Connect the 4924 and 496P with a GPIB cable after both are powered up.

3. Set the 4924 ON LINE switch out (off line).

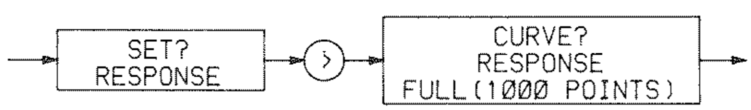
4. Rewind the tape.

5. Press FORWARD to advance to file 1. Press FORWARD again, as desired, to reach a file further into the tape.

6. To save the current control settings and waveform in digital storage, press LISTEN on the 4924 and RESET TO LOCAL on the 496P.

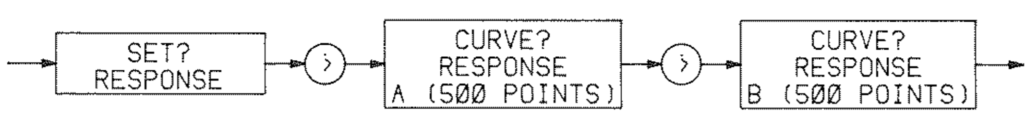
Pressing the RESET TO LOCAL button causes the analyzer to transmit instrument settings and a waveform. The message is formatted so that when it is played back to the analyzer, it restores the settings and display. The message is a combination of the responses to the SET and CURVE queries.

If SAVEA is OFF, A and B are transmitted as a full waveform (A and B memories merged for 1000 points):



3401-8

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



3401-9

The analyzer transmits waveform data as ASCII-coded decimal numbers unless changed by the ENCDG argument in a WMPRE 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 only control settings 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 ends—there is no reset switch except for POWER. The 496P, once it starts talking, keeps talking until finished. It also cannot be interrupted except by turning off the power. (This is true only if the 496P begins transmitting—if there is no listener, it flashes a message to the operator and returns to local control.)

7. To move to the next 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

With the LISTEN ONLY switch set the 496P 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, we have chosen to implement this mode so the 496P goes to remote state after buffering a message. This makes listen-only mode consistent with the nonlisten-only mode, which requires that the 496P be under remove control to execute commands that change front-panel settings or waveform data in digital storage.

To restore control settings and a display previously recorded:

1) find the file on the tape using FORWARD or REVERSE on the 4924, and

2) press TALK. The 496P goes to remote to execute the message and then returns to local control.

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 operator can compare the current spectrum data being acquired in B memory to the saved waveform 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 FORWARD and REVERSE switches. Or you can record a message on every other tape file as a marker. With the latter scheme, just press FORWARD and then TALK after either recording or playing back a file and a message with the number of the next file is placed on the 496P 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
```

Substitute the number of files you want to mark for N in line 110. Line 150 uses the RDOUT command to replace the crt readout with a message showing the file count and a reminder to move to the beginning of the file. Each time through the loop, line 170 marks the file 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 496P is compatible with IEEE Standards 488-1975 and 488-1978. The connector and signal levels at the connector follow the specifications in the IEEE 488 standards. Table 1-2 lists 496P interface capabilities, as defined in the standards.

Table 1-2
496P IEEE 488 INTERFACE FUNCTIONS

Function	Implemented As
Source handshake	SH1
Acceptor handshake	AH1
Talker	T5
Listener	L3
Service Request	SR1
Remote local	RL1
Parallel poll	PP1
Device clear	DC1
Device trigger	DT1
Controller	C0

Source Handshake (SH1)

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 microseconds.

Acceptor Handshake (AH1)

Complete capability to receive messages on the bus.

Talker (T5)

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

Listener (L3)

Complete listener function; unadddresses as a listener when addressed as a talker. The analyzer operates in a simple system as a listener-only if the LISTEN ONLY switch is set to 1.

Service Request (SR1)

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)

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 effect of busy status on remote/local transitions.

The current value of all programmable functions is maintained when switching from local to remote; only the value of TIME, MINATT, PEAK, and CRSOR may change to match the front-panel control when switching from remote to local. 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 does change settings for which there is no local control and does output data while under local control.

Parallel Poll (PP1)

Responds to a parallel poll to indicate if service is requested.

Device Clear (DC1)

The instrument 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 device trigger function is implemented so the group execute trigger message causes the instrument to abort the current sweep and rearm the sweep. The new sweep does not begin until the triggering conditions are met.

Controller (C0)

The 496P does not act as a controller.

CONNECTING TO A SYSTEM

The 496P can be connected directly to a GPIB system with the cable supplied with the instrument. The port is shown in Fig. 1-7. To avoid interference on the bus, connect the 496P after turning on power or while the controller on the bus is turned off.

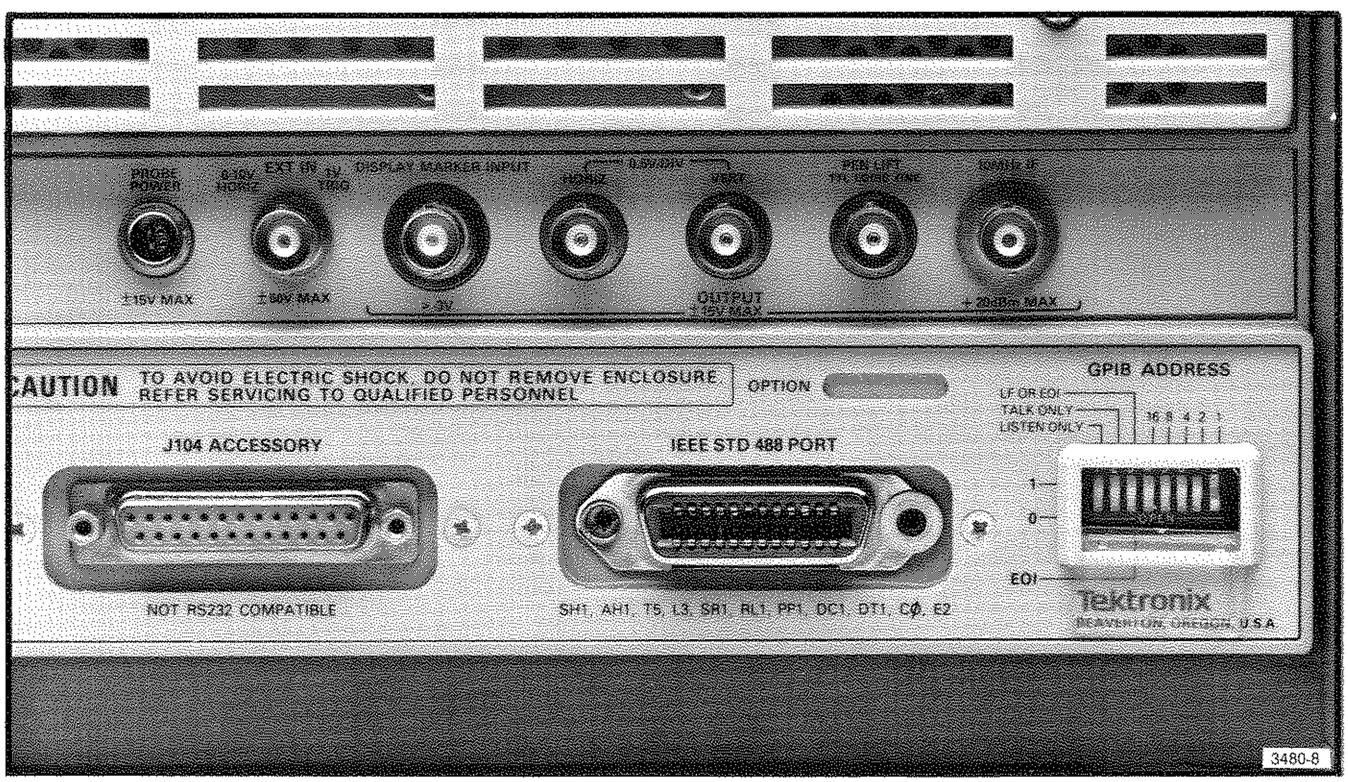


Fig. 1-7. The 496P GPIB port on the rear panel.

The GPIB is a flexible system—it works either in a star or linear configuration as shown in Fig. 1-8. 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 496P, etc.), and more than half the devices connected should be powered up.

If an internal switch is changed, the 496P asserts SRQ on power-up. This requires immediate action by some controllers, such as 4050-Series, so is not recommended for these controllers. Other internal switches can be used to select self-test modes at power-up; changing these switches prevents the 496P 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 496/496P Operators Manual and the 496/496P Operators Handbook. Turn there for instructions on how to begin operating the instrument.

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

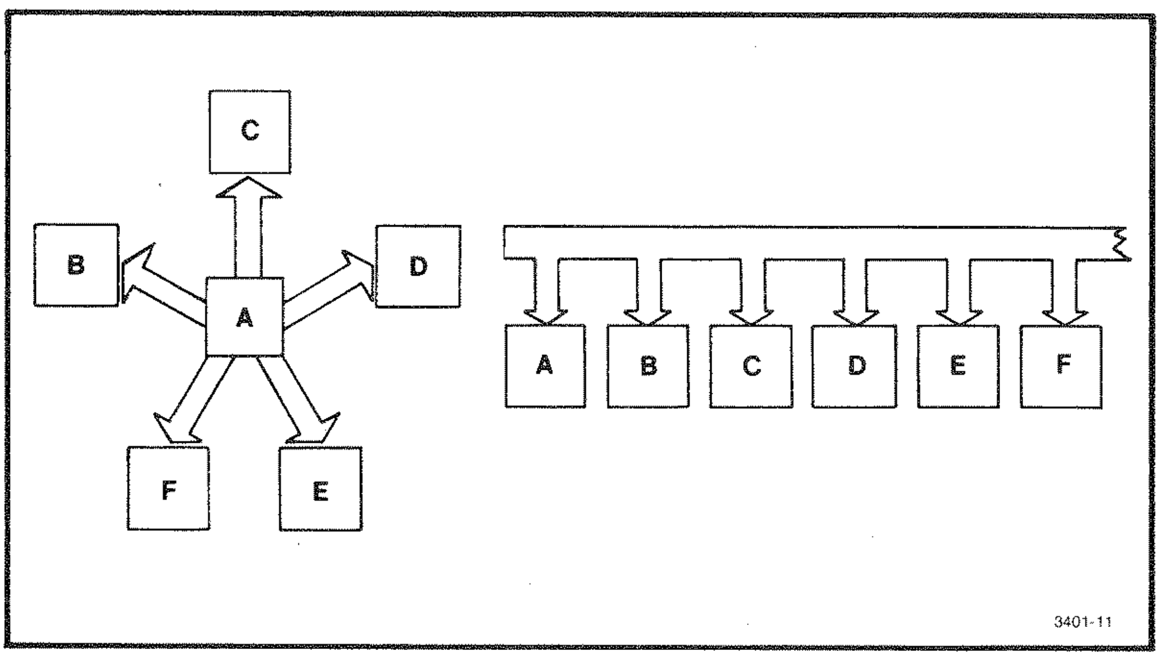


Fig. 1-8. The 496P can be connected to a GPIB system in either a star or linear manner.

@

GETTING STARTED

Getting started with the 496P on the GPIB is simple—if you are already on good terms with a GPIB controller. If not, talking to the 496P over the bus may be the easiest way to get over any uncertainty you feel in getting started.

The 496P speaks a friendly language that includes mnemonics to control 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 496P to start communication, but these are steps taken by most controllers when executing a GPIB i/o statement.

SETTING PROGRAMMABLE CONTROLS

Let's keep it simple. We can, because the 496P lets you make complex spectrum measurements semiautomatically. Many measurements can be made with just three controls: FREQUENCY, FREQUENCY SPAN/DIV, and REFERENCE LEVEL.

FREQUENCY changes the position of the spectrum window you are viewing, tuning the analyzer to change the frequency at the center of the crt. DEGAUSS may be used after significant frequency jumps to remove the effects of residual magnetism in the analyzer front end.

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

REFERENCE LEVEL raises or lowers the window, setting the amplitude calibration of the top graticule line on the crt.

Here's how to program the 496P using just these three controls to measure the CAL OUT signal.

1. The CAL OUT signal can be centered by FREQ 100 MHZ. Note that the 496P accepts engineering notation (MHZ for megahertz). It also accepts integers, floating point, or scientific notation. Thus, FREQ 100000000, FREQ 100000000., and FREQ 100E+6 also set the center frequency to 100 MHz.

2. Span down to look more closely at the signal with SPAN 1 MHZ.

The 496P automatically picks resolution bandwidth and time/division to fit the new span/division, unless AUTO RESOLUTION and TIME AUTO are canceled. 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 REFLVL −20 DBM.

The 496P automatically selects appropriate input attenuation and IF gain for a reference level at the power level of the CAL OUT signal's fundamental frequency. The 496P microcomputer takes into account the MIN RF ATTEN and MIN NOISE controls when setting attenuation and gain.

The 496P powers up with automatic modes active and in MAX SPAN to display all the frequencies. You can restore this condition at any time with the INIT command.

4050-Series Graphic Systems

How do steps 1, 2, and 3 above work on a TEKTRONIX 4050-Series controller? The 496P commands are inserted in a GPIB output statement, PRINT:

```
PRINT @A:"FREQ 100 MHZ"
PRINT @A:"DEGAUS"
PRINT @A:"SPAN 1 MHZ"
PRINT @A:"REFLVL −20 DBM"
```

```
PRINT @A:"FREQ 100 MHZ;DEGAUS;SPAN 1 MHZ;REFLVL −20 DBM"
```

As this last statement shows, all four commands can be strung together, delimited by semicolons. A is a variable that holds the value of the 496P GPIB address switches (stored previously). A can be replaced by a constant—the number for the GPIB address—in any statement as:

```
PRINT @1:"FREQ 100 MHZ;DEGAUS;SPAN 1 MHZ;REFLVL −20 DBM"
```

When the 496P 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 496P 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 496/496P Operators Manual or 496/496P Operator's Handbook or, better yet, try the Auto Cal routine at the end of this section.

Two Other Controllers

The store is the same for other 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:

```
PUT "FREQ 100 MHZ;DEGAUS;SPAN 1 MHZ;REFLVL −20 DMB" INTO @N,LA
```

where N is the number of the GPIB interface in the controller and LA is the 496P listen address (primary address plus 32).

For the Hewlett-Packard 9825 Dekstop Computer, use a write statement:

```
wrt "a","freq 100 mhz;degaus;span 1 mhz;reflvl −20 dbm"
```

A 9825 device statement is used to assign "a" to the 496P:

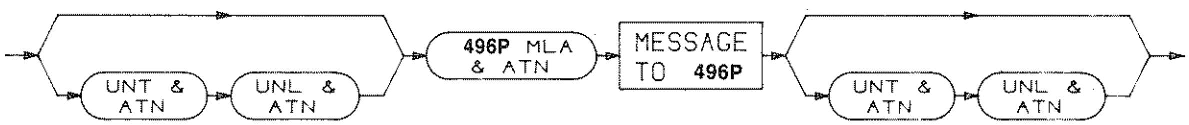
```
dev "a",701
```

The device statement assigns address 701 to "a". The 01 in 701 assumes the 496P GPIB ADDRESS switches are set to 1, but could be changed to any number between 00 and 30. (Or "a" could be replaced in the write statement by 700 + the 496P primary address.)

The 9825 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 496P.

In General

Whatever the controller or statement, it must take the following actions to get a message to the 496P:



The unlisten (UNL) and untalk (UNT) messages are optional in this syntax diagram of bus traffic. However, one or both are sent by most controllers when they begin transmitting and end transmitting on the bus, in order to guarantee a clear communications channel. The controller sends the GPIB address you entered as part of the controller's GPIB i/o statement. The controller either converts it to the 496P listen address or expects to receive the listen address with the offset included (33 instead of 1 as noted for TEK SPS BASIC above). 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 496P 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 496P programmable control mnemonics are collected for quick reference on a foldout chart at the back of the 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.

The 496P 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 496P returns the state of programmable controls when queried. This takes two steps:

1) query the 496P. The query takes the form of the mnemonic for a function name followed by a question mark. Send it in the same way as a command described above;

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 1 MHz selected above. What is that bandwidth? The query RESBW? readies the 496P to output the answer.

The query can be inserted in any message to the 496P. It is executed in its turn, which means if RESBW? precedes the SPAN command-in the above example, the 496P 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 for an example above and combine with the controller GPIB input statement.

4050-Series:

```
PRINT @A:"FREQ 100 MHZ;SPAN 1 MHZ;REFLVL -20 DMB;RESBW?VIDFLT?"
INPUT @A:P$
```

TEK SPS BASIC:

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

TA is the 496P talk address derived in the same manner as LA—primary address + 64.

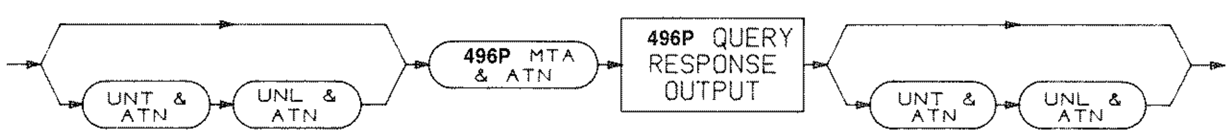
9825:

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

With the 9825, you must dimension P$ before using it. For example:

```
dim P$[80]
```

Whatever the controller input statement, it must take the following actions to receive a message from the 496P:



@

This syntax figure could be appended to the end of one above for sending a message. Together, they describe the two steps necessary to obtain output from the 496P. The message in the first figure would include the query and the response in the second figure would come from the 496P to answer that query.

LISTEN/TALK ROUTINE

Now let's put the above statements for message i/o into a simple routine to exercise the 496P as a listener and a talker. This routine is handy because it waits for your input and sends it, time after time. If the 496P responds with a message, the routine prints it before requesting another message from you.

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

4050-Series BASIC:

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

TEK SPS BASIC:

```
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,LA
140 GET P$ FROM @N,TA
150 PRINT P$
160 GO TO 110
200 POLL @N,Q2,Q1,Q0;TA
210 PUT "ERR?" INTO #N,LA
220 GET Q$ FROM #N,TA
230 PRINT Q$
240 RETURN
```

Lines 10 and 20 are added to load the software driver and to disable the bus timeout feature in this driver.

9825:

```
0: dim P$[80]
1: oni 7, "srq"
2: eir 7
3: ent "enter message",P$
4: wrt "a"P$
5: red "a",P$
6: prt P$
7: gto 3
8: "srq":rds("a")→r1
9: wrt "a,""err?"
10: red "a",P$
11: eir 7;iret
```

This routine makes use of a friendly feature of the 496P: when talked with nothing to say, the 496P 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 routine reads a response after every message and prints the response (a blank line if the 496P sends a byte with all ones).

The SRQ handler employs another 496P feature. Rather than print a code for the status byte, the routine asks for the error that caused the SRQ (ERR?)—it's more specific about the problem. The meaning of the error codes is listed under ERR? in Section 7. In the case of the 4050-Series and the 9825, the print statement in the loop takes care of errors that result from messages you send. In the case of TEK SPS BASIC, a print statement in the SRQ routine does the job.

The above routines assume you have assigned the value of the 496P address to variable A (or LA/TA or a) as discussed above. 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.

ACQUIRING INSTRUMENT SETTINGS WITH SET?

A query is provided to learn instrument settings for reference or for restoring the instrument to those settings. This query (SET?) 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 to the same settings. No further manipulation is required, just store the message as it is transmitted by the 496P. Your controller must be ready for a long character string: dimension a string variable large enough for 345 characters, although the exact size depends on the current settings.

The instrument settings need not have been programmed. This allows you to operate you system in a learn mode:

1) set up for the measurement (and try it) from the 496P front panel;

2) teach these settings to the controller using the SET query, storing the response;

3) restore the 496P to the same settings by transmitting back to the instrument the stored SET? response.

Here are 4050-Series BASIC program segments that allow you to store two instrument set-up settings by pressing a user-definable key. (Only keys 1 and 2 are used, but others could be used in a similar manner.) Pressing SHIFT and the key restores either of the instrument settings earlier stored by pressing that key.

```
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


1000 REMARK LEARN INSTRUMENT SETTINGS
1010 PRINT @A:"SET?"
1020 GO TO K of 1030,1050
1030 INPUT @A:K$
1040 GO TO 1060
1050 INPUT @A:L$
1060 RETURN


2000 REMARK RESTORE INSTRUMENT SETTINGS
2010 GO TO K OF 2020,2040
2020 PRINT @A:K$
2030 GO TO 2050
2040 PRINT @A:L$
2050 RETURN
```

Lines 4 through 50 are subroutines called when user-definable keys 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. Lines 1000 to 1060 input a SET? response and store it. Lines 2000 through 2050 return a SET? response to the 496P.

These subroutines work with a program, which need only be a loop that occupies the processor until interrupted by a key. Here is a simple 4050-Series program that allows you to exercise these subroutines. It takes care of such details as dimensioning the string variables and arming the user-definable keys (SET KEY):

```
1 REMARK SETTINGS PROGRAM
2 GO TO 100

100 DIM K$(350)
110 DIM L$(350)
120 SET KEY
130 GO TO 130
```

RESETTING THE 496P AND INTERFACE MESSAGES

A command, INIT, resets the 496P programmable controls to power-up state (see Section 7 for more on this command). It is sent in the same manner as other commands in device-dependent messages.

An interface message, DCL, clears the 496P 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, as is illustrated later in this section in an Auto Cal routine. If the 496P is waiting for its talk address so it can execute an output query, output is aborted and the buffers cleared by DCL (or any device-dependent input).

Use the WBYTE statement in 4050-Series BASIC to send DCL:

```
WBYTE @20:
```

The decimal code for other universal commands can be substituted for 20, such as 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 496P listen address:

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

where L = GPIB address + 32.

Other addressed commands are 5 for PPC (parallel poll configuration) 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.

In TEK SPS BASIC, use the SIFCOM statement, entering the mnemonic for the interface message:

```
SIFCOM @N,LA,"GTL"
```

LA is the 496P listen address and is necessary only for addressed commands.

With the 9825, 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.

When the IFC line is asserted by the controller, as when the 4050-Series BASIC statement INIT is executed, the 496P talker and listener functions are initialized (same effect as UNT and UNL). The 9825 RESET statement also asserts IFC. In TEK SPS BASIC, 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 tackle the former here and reserve the latter for Section 5. The 496P powers-up ready to transmit waveforms in ASCII (Section 5 explains how to change modes).

4050-Series

Let's define another 4050-Series controller key for this job:

```
24 GOSUB 5000
25 RETURN
```

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

```
5000 REMARK 4050-SERIES WAVEFORM INPUT SUBROUTINE
5010 PRINT @A:"CURVE?"
5020 INPUT @A:W
5030 RETURN
```

Line 5010 requests waveform (CURVE) data. The INPUT statement in line 5020 ignores the ASCII characters that the 496P sends at the beginning of its CURVE? response. The INPUT statement then fills array W with the 1000 numbers transmitted by the 496P.

These program segments can be added to those listed above for acquiring instrument settings with the SET query. For this subroutine to run, however, you must add a line to dimension W:

```
115 DIM W(1000)
```

See Section 8 for help in plotting the waveform.

TEK SPS BASIC

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,LA
5020 RASCII W FROM @N,TA
5030 RETURN
```

Array W must be previously dimensioned: use INT(1000). RASCII operates as noted above for INPUT; it ignores nonnumeric characters and proceeds to fill the array when it receives numbers.

9825

Here's a self-contained program for 9825 input of a 496P 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 "a","wfmpre wfid:A;curve?"
3: rdb("a")→r1
4: if r1#44;rdi 4→r1;rdi 4→r1;jmp 0
5: wait 1
6: 1→I
7: for I=1 to 500
8: red "a",A
9: A→W[I]
10: next I
```

The WFMPRE command in line 2 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. The loop in lines 7 through 9 then inputs the waveform data. Of course, you must have assigned "a" to the 496P with a device statement as explained earlier.

GETTING SMARTER

Signal analysis can be even easier. Put the 496P microcomputer to work to find and measure signals with its internal waveform processing. The full set of commands is described in Section 6 and some more instructions for their use are found in Section 8, but to get started "getting smarter", here is a simple application.

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:

```
PRINT @A:"INIT"
100 REMARK CATALOG ROUTINE
110 PRINT @A:"SPAN 10 MHZ;REFLVL -20 DBM; VIDFLT NARROW;SIGSWP"
120 FOR I=1 TO 10
130 PRINT @A:"FREQ ",I*100E6;",DEGAUS;SIGSWP:WAIT"
140 PRINT @A:"FIBIG;CENSIG;TOPSIG"
150 PRINT @A:"SPAN 1 MHZ;SIGSWP;WAIT;FMAX;CENSIG"
160 PRINT @A:"SIGSWP;WAIT;FIBIG;CENSIG;FREQ?REFLVL?"
170 INPUT @A:R$
180 PRINT @A:"SIGNAL ",I,R$
190 PRINT @A:"SPAN 10 MHZ;REFLVL -20 DBM"
200 NEXT I
```

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

Line 120: Loop start.

Line 130: Tunes to a harmonic of the calibrator signal, degausses to improve tuning accuracy, 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). CENSIG changes the analyzer's frequency to center the signal. 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: Spans down and takes another sweep to analyze the new data. To reduce execution time, we use FMAX to find the maximum value, expecting it to be the signal peak. CENSIG acts on the updated data.

Line 160: The analyzer is asked to take another sweep, to recenter itself, and then to report its frequency and reference level, which are also the frequency and amplitude of the signal component.

Line 170: Inputs the analyzer response.

Line 180: Because the response to each query in line 160 begins with a mnemonic for the function, the analyzer output string acquired in line 170 is intelligible as is.

Line 190: Ready the analyzer to do it again.

Line 200: Go 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; instructions for putting 496P waveform processing to work are given in Section 8.

AUTOCAL—A PROGRAMMABLE ASSIST IN CALIBRATING THE FRONT PANEL

How about letting your controller do the routine part of setting the frequency readout and vertical gain of the analyzer? Freed of turning the knobs, you can concentrate on making accurate settings.

```
100 REMARK AUTOCAL ROUTINE
110 PRINT @A:"INIT"
120 PRINT @A:"SPAN 20 KHZ;VRTDSP LOG:2 DBM"
130 PRINT @A:"RDOUT 'SET DOT ON CENTER LINE WITH HORZ'"
140 PRINT @A:"RDOUT 'POSITION, THEN PRESS RETURN KEY'"
150 INPUT X$
160 PRINT @A:"RDOUT 'SET TRACE ON BOTTOM LINE WITH'"
170 PRINT @A:"RDOUT 'VERT POSITION, THEN PRESS RETURN'"
180 INPUT X$
190 PRINT @A:"RDOUT NORMAL"
200 PRINT @A:"SPAN 1 MHZ;FREQ 100 MHZ;DEGAUS;SIGSWP"
210 PRINT @A:"REFLVL -10 DBM;VRTDSP LOG:10 DBM;SIGSWP;WAIT"
220 PRINT @A:"FIBIG 100;CENSIG;SIGSWP;WAIT;REPEAT 2"
230 PRINT @A:"FRCAL 100 MHZ"
240 PRINT @A:"SPAN 200 KHZ;SIGSWP;WAIT;FIBIG 100;CENSIG"
250 PRINT @A:"SPAN 20 KHZ;SIGSWP;WAIT;FIBIG 100;CENSIG"
260 PRINT @A:"REFLVL -20 DBM;TRIG FRERUN"
270 PRINT @A:"RDOUT 'SET PEAK TO TOP GRATICULE LINE'"
280 PRINT @A:"RDOUT 'WITH LOG CAL, THEN PRESS RETURN'"
290 INPUT X$
300 PRINT @A:"RDOUT 'SET TRACE PEAKS EQUAL WITH AMPL'"
310 PRINT @A:"RDOUT 'CAL, THEN PRESS RETURN KEY'"
320 PRINT @A:"VRT LOG:10;SIG;WAI;VRT LOG;2,SIG;WAI;REP 1E+6"
330 INPUT X$
340 WBYTE @20:
350 PRINT @A:"RDOUT 'SET TRACE PEAK TO TOP LINE WITH'"
360 PRINT @A:"RDOUT 'LOG CAL, THEN PRESS RETURN KEY'"
370 PRINT @A:"TRIG FRERUN"
380 INPUT X$
390 FOR I=1 TO 40
400 PRINT @A:"RDOUT 'FRONT-PANEL CALIBRATION COMPLETE';RDOUT ''"
410 NEXT I
420 PRINT @A:"RDOUT NORMAL"
```

Line 110: Resets power-up conditions.

Lines 120-180: Prepare display for setting the front-panel POSITION adjustments. The RDOUT commands display instructions on the 496P crt. The INPUT statements make the controller wait for you to make the adjustments.

Lines 200-230: Find and center the 100 MHz CAL OUT signal and calibrate the Frequency readout. Single-sweep mode is selected and WAIT commands are inserted to guarantee the sweep completes each time before the waveform processing commands act on the new data. The REPEAT command causes the 496P to take a sweep and center itself on the biggest signal three times before proceeding to input line 230.

Lines 240-290: The analyzer spans down, resetting Frequency (with CENSIG) as necessary. The display is prepared and you are instructed to set the CAL OUT signal to the reference level (top graticule line).

Lines 300-330: Now comes the tricky part. The RDOUT commands display a message asking you to make equal the peaks of the CAL OUT signal in both 10 dB/div and 2 dB/div. The 496P repeatedly switches its display between 10 dB/div and 2 dB/div so you can see the effect of varying AMPL CAL.

While making this adjustment, you may need to change LOG CAL to keep the two trace peaks on-screen.

Lines 340-380: Interrupt the loop the 496P is executing (with the DCL interface message) and restore a free-running trace. You are asked again to set the peak to the reference level.

Lines 390-420: A flashing message assures you that you are finished and restores the readout to show display parameters.

DEVICE-DEPENDENT MESSAGE STRUCTURE AND EXECUTION

The 496P device-dependent message structure is based on two oft-times conflicting goals: to enhance compatibility with a variety of GPIB systems, yet be simple and obvious to use.

These goals are achieved within the framework of the Tektronix GPIB Codes and Formats standard. This standard is intended to make messages on the bus unambiguous, while making instrument handling of messages friendly; that is, accepting of variations in the message. As much as possible, compatibility with existing devices is maintained, while encouraging use of codes and data formats that make maximum use of bus capabilities.

To make 496P 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 496P center frequency to 500 megahertz, the message "FREQ 500 MHZ" would be sent over the bus after addressing the 496P 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 496P messages that makes them readable; therefore, human-oriented.

496P INPUT MESSAGES

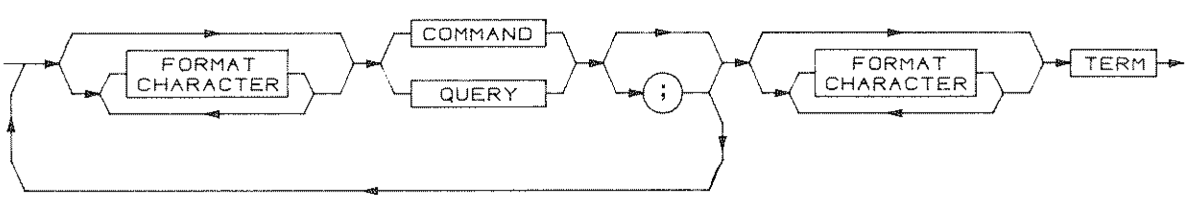
A remote control message to the 496P comprises one or more message units. Message units are of two types:

1) commands that the 496P inputs as control or measurement data, or

2) queries that request the 496P to output data.

One or more message units can be transmitted as a message to the 496P. Message units contain ASCII characters (binary may also be used for waveforms). The 496P accepts either upper or lower-case characters for the mnemonics shown in the syntax diagrams.

Input Message Format



3401-15

Message Unit Delimiter (;)

Message units are separated by the ASCII code for semicolon (;). A semicolon is optional following a query or the last message unit.

Message Terminator (TERM)

The end-of-message terminator may be either:

1) the END message (EOI asserted concurrently with the last data byte), or

2) the ASCII code for line feed (LF) sent as the last data byte.

The active terminator is selected by a rear-panel switch.

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 with no bypass. Allowable format characters are: space (SP), carriage return (CR), and line feed (LF), as well as all other control characters and comma (,). At some points in a message, the 496P may also accept other nonalphanumeric characters.

Input Buffering and Execution

The 496P buffers each message it receives with a capacity that exceeds that required for the SET? response. The 496P waits until the end of the message to decode and execute it. A command error in any part of a message prevents its execution. Commands that would conflict with local control are ignored if the instrument is not under remote control—see Remote/Local under IEEE 488 Functions in Section 1.

If the message contains multiple message units, none are acted on until the 496P sees the end-of-message terminator. When the 496P sees the terminator, it executes the commands in the message in the order received. It generally remains busy until done executing the commands in the buffer (see Status Byte in Section 7 for more on busy status). While busy, further input is not accepted. Output, if requested, is begun only after the entire input message is executed.

Message execution can be aborted by the DCL or SDC interface messages.

Because display (measurement) data input and output and waveform processing share the same buffer, conflicts can arise. This is discussed in connection with the CURVE command in Section 5 and waveform processing in Section 6 and is further expanded on in Section 8.

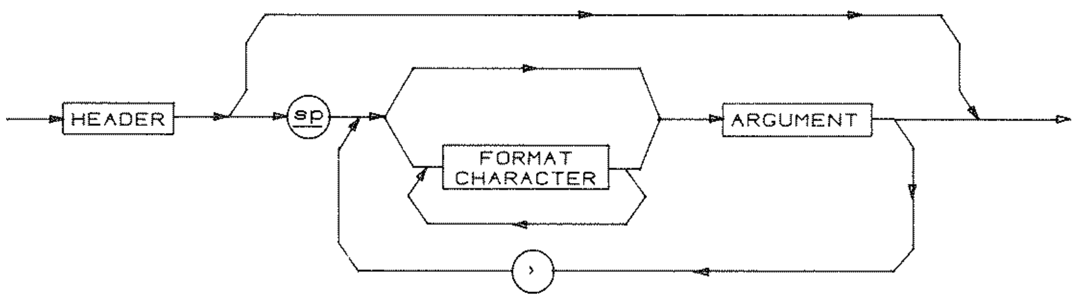
Command Format

A command message unit either:

1) sets an operating mode or parameter; or

2) transfers display data to the instrument.

The command format to set a mode or parameter includes these possible paths:



3401-16

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

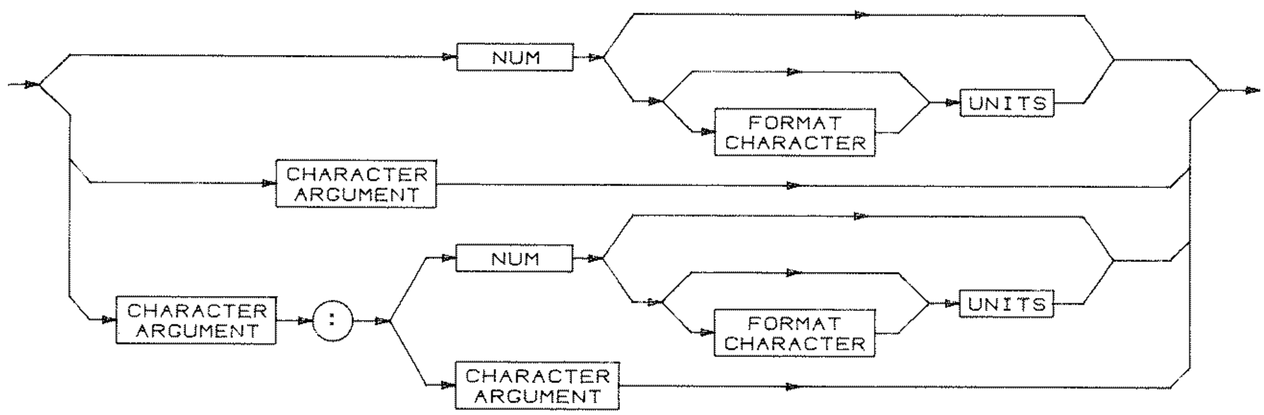
A space (SP) must separate the header from any argument(s).

Argument Delimiter (,).

A comma (,) must separate multiple arguments.

Argument Format

Arguments following the header may be either a number or a group of characters, or either linked to a character argument:

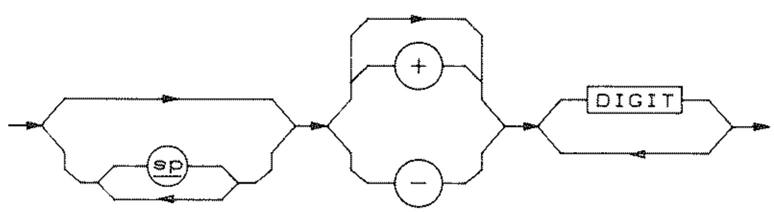


3401-17

Numbers.

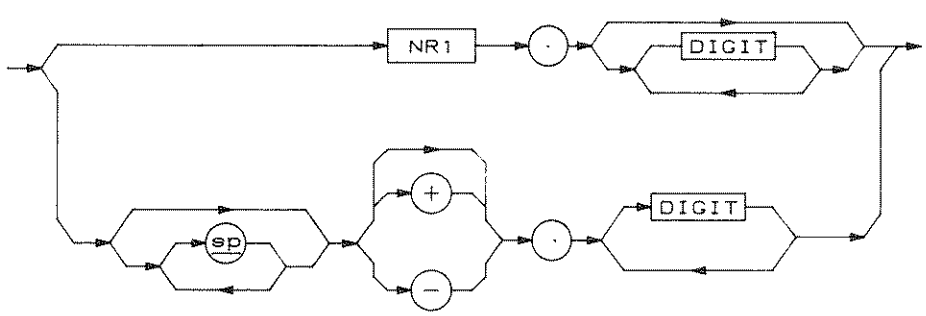
The defined element NUM is a decimal number in any of three formats; NR1, NR2, or NR3.

NR1 is an integer (no decimal point):



3401-18

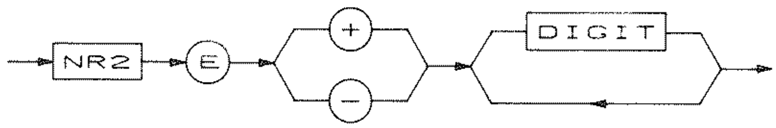
NR2 is a floating point number (decimal point required):



3401-19

@

NR3 is a floating point number in scientific notation:



3401-20

If NUM exceeds the range of the function, the 496P microcomputer does not execute the command, but issues an error (see PEAK and POINT for exceptions). Numbers within the range are rounded. NUM is used in a special way in commands for front-panel push buttons. A number may be used instead of ON (1) or OFF (2). Numbers other than 1 or 0 are rounded, but must be negative.

Units.

The 496P accepts arguments in engineering notation; that is, engineering units may be appended to a number argument. The 496P 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=1E3 or 1E+6, depending on the function: MSEC stands for 1E−3 (milliseconds) in the TIME (time/div) command, while MHZ stands for 1E+6 (megahertz) in the SPAN (span/div) command. The rest of the units element 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 that follow.

Character Argument.

Words or mnemonics may be arguments. ON and OFF, for instance, are arguments for the commands that correspond to 496P front-panel push buttons like those for vertical display or digital storage.

Link Argument.

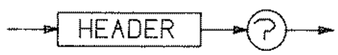
The bottom path in the argument diagram combines character and number arguments in a link argument. The line is the color (:) delimiting the first and second arguments. Two arguments can expand front-panel control via the GPIB compared to that allowed locally. For example, the VRTDSP (vertical display) command employs link arguments to make more scale factors available than those for the 10 and 2 dB/div buttons and the default values for the LIN button.

String Argument.

A special character argument is used for a few commands that transfer characters to be used as is. The characters are enclosed in quotes to delimit them as a string argument.

Query Format

A query message unit requests data from the instrument, either from a function or the display. The query message unit format is:



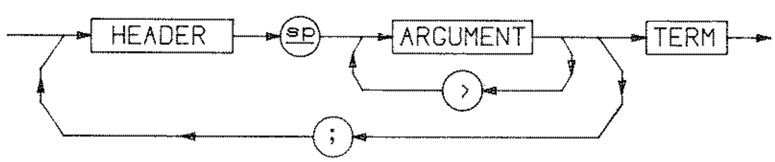
3401-21

496P OUTPUT MESSAGES

When the 496P 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 agrument(s). Message units are combined if the output includes a response to the SET query or to more than one query:



3401-22

Output Message Execution

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

@

FRONT-PANEL CONTROL

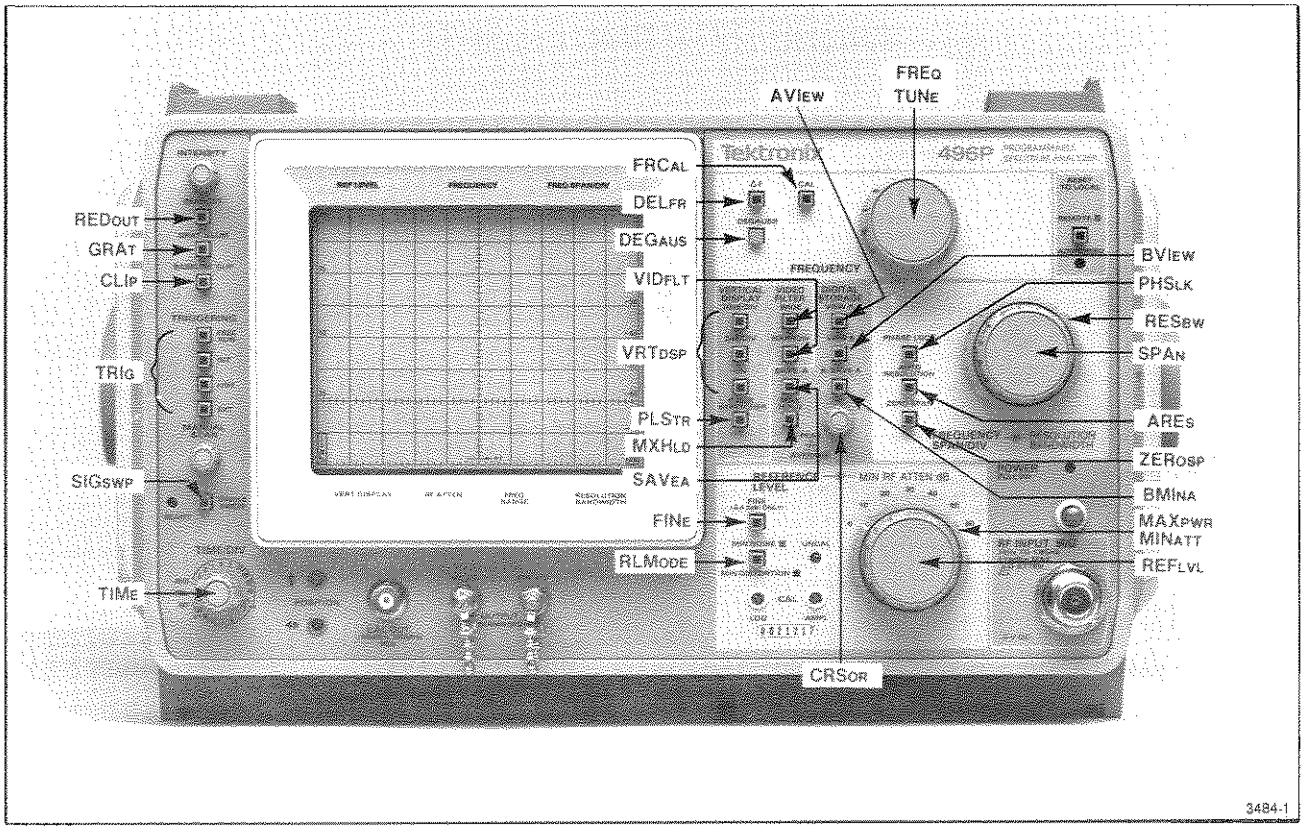


Fig. 4-1. Front-Panel Control Commands and Queries.

Commands and queries for front-panel control are grouped in this section according to these functions:

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

Some controls, however, are operated only from the front panel (no remote control). These controls are:

INTENSITY
MANUAL SCAN
POSITION
AMPLITUDE and LOG CAL
POWER

Also, the PEAK/AVERAGE cursor can be set by remote control to the existing knob position, to minimum, or maximum.

FREQUENCY (see Fig. 4-2)

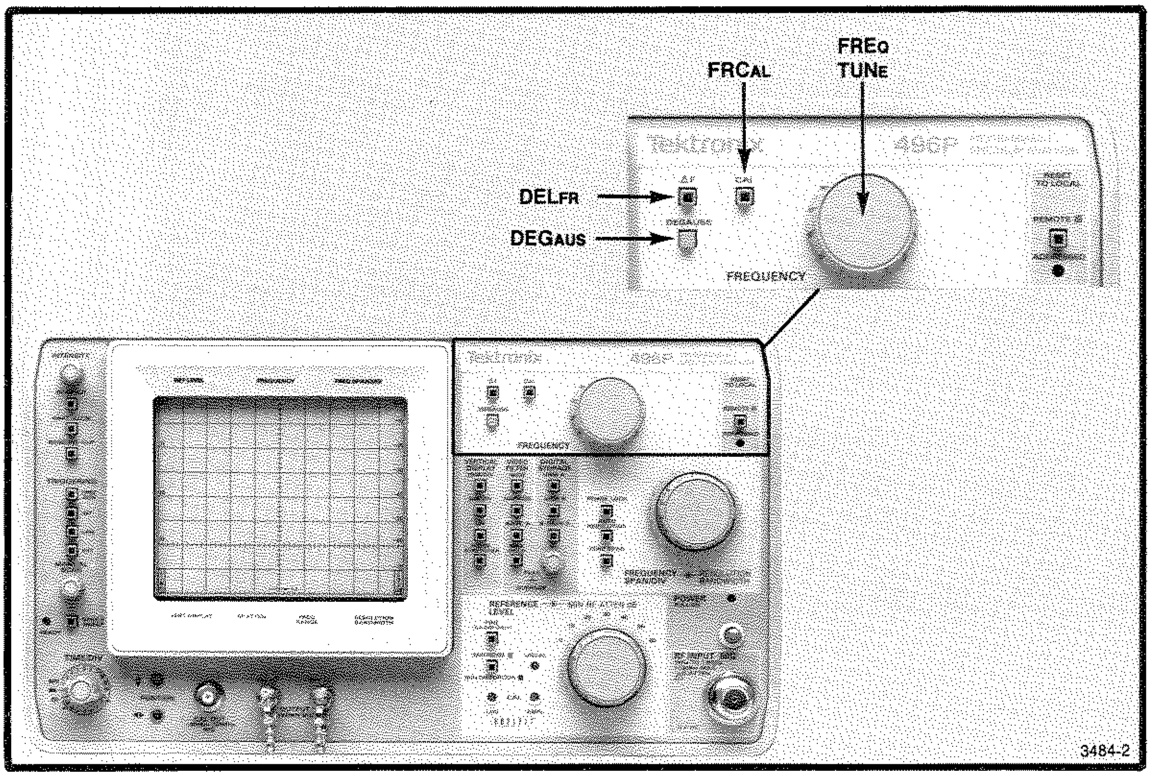
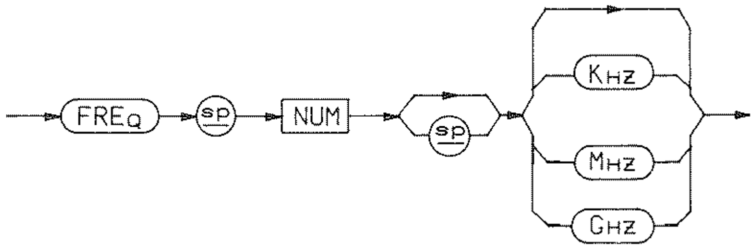


Fig. 4-2. Front-Panel Control—Frequency commands.

Commands in this group set and change the 496P center frequency (FREQ and TUNE), activate the delta-frequency function (DELFR), and calibrate the center frequency readout (FRCAL). Degaussing current (DEGAUS) can be applied to restore front-end alignment.

FREQ (center frequency) command:



3401-25

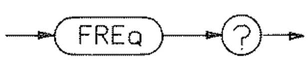
NUM: The analyzer centers its span about the value in the command argument. The range of values and resolution of the instrument's response are the same as for front-panel operation.

While setability is the same as from the front panel, repeatability under remote control is also specified. It is within ±(2 MHz + 10% of span/div) or ±(0.1% of frequency + 10% of span/div) of previous settings to the same frequency—whichever is greater. Ambient temperature change must be within 10°C.

Power-up value: −56 MHz

Interaction: Use DEGAUS to improve accuracy if frequency is changed significantly.

FREQ (center frequency) query:

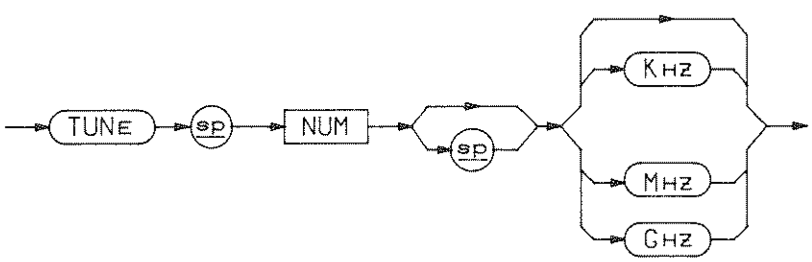


3401-94

Response to FREQ query:

3401-26

Tune (increment frequency) command:



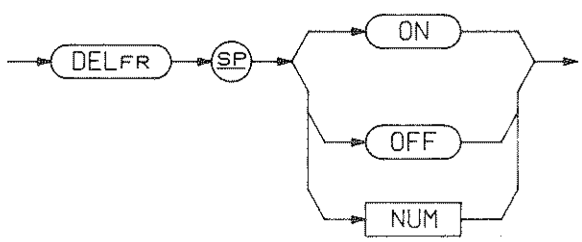
3401-27

NUM: The analyzer changes its center frequency using the value of the command argument as an offset to its previous center frequency. This command takes advantage of the analyzer's high relative accuracy. If the new frequency lies outside the analyzer's frequency range or, in narrow spans, the tuning range of the 2nd LO, the analyzer maintains its current center frequency and reports an error.

The overall TUNE accuracy is (±7% or ±150 kHz, whichever is greater) when the span/div is >50 kHz. When the span/div is ≤50 kHz, the TUNE accuracy is (±7% or ±100 Hz, whichever is greater).

There is no TUNE query.

DELFR (delta-frequency) command:



3401-31

ON: Turns on the delta-frequency function. 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.

OFF: Turns off the delta-frequency function.

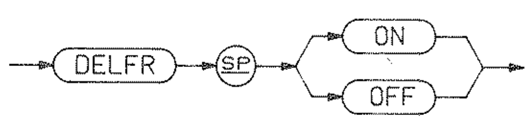
NUM: 1 equals ON; 0 equals OFF. Other numbers are rounded to 1 or 0.

Power-up value: Off.

DELFR (delta-frequency) query:

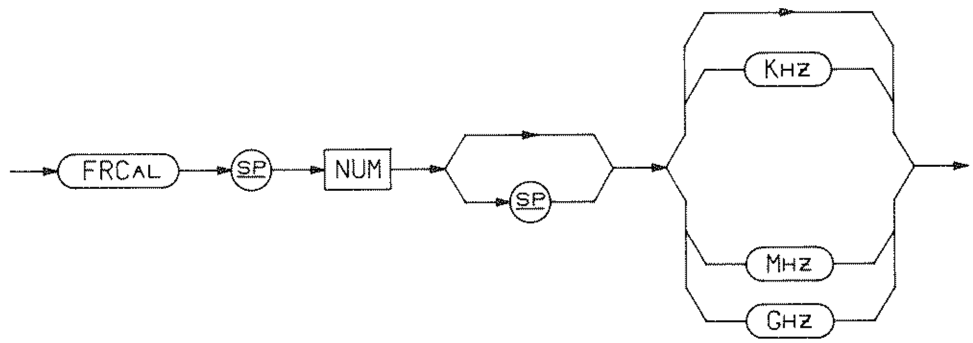
3401-32

Response to DELFR query:



3401-33

FRCAL (frequency call) command:



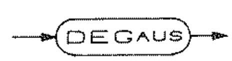
3401-34

NUM: The center-frequency readout assumes the value of NUM without retuning the analyzer local oscillator, calibrating the readout.

Interaction: Precede FRCAL with DEGAUS.

There is no FRCAL query.

DEGAUS (degauss tuning coils) command:



3401-35

DEGAUS: Turns on a degaussing current to remove residual magnetism in the tuning coils for the 1st LO. This command is recommended after a significant frequency change to improve center frequency and amplitude accuracy. Also, use DEGAUS before calibrating center frequency readout with FRCAL.

There is no DEGAUS query.

FREQUENCY SPAN AND RESOLUTION (see Fig. 4-3)

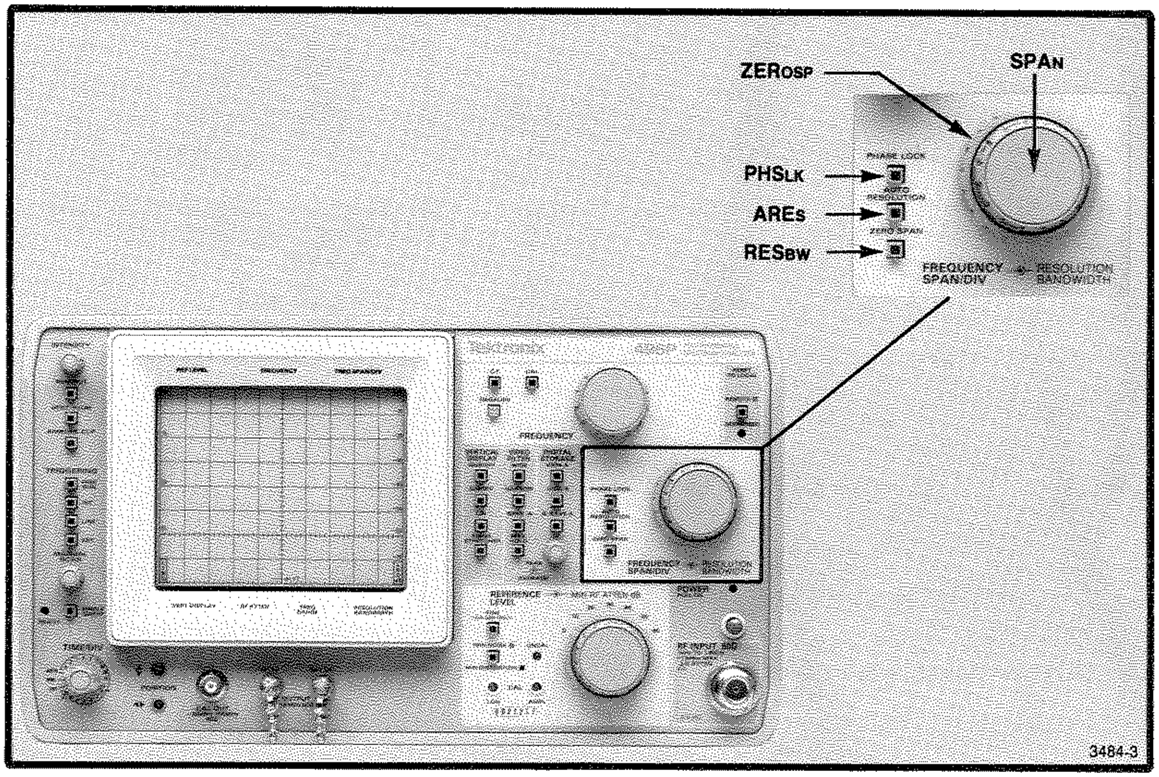
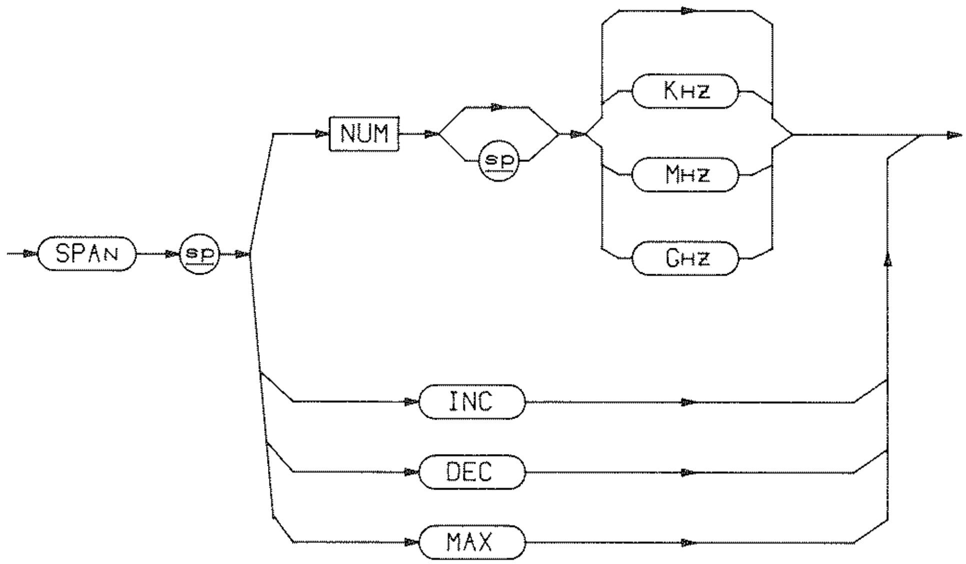


Fig. 4-3. Front-Panel Control—Frequency Span and Resolution commands.

Commands in this group control the frequency span (SPAN) and resolution (RESBW and ARES) of the display. Automatic phaselock at narrow spans can be overridden (PHSLK).

@

SPAN (frequency span/division) command:



3401-43

NUM: Selects the span/division (not limited to the 1-2-5 steps on the front panel); the value of the argument is rounded to two significant digits. Zero converts the analyzer to the time domain; in zero-span mode, the analyzer displays signals within its band-pass (RESBW) about its center frequency (FREQ).

An execution error is issued if the number is too small or too large.

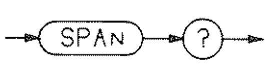
INC: Selects the next larger span/div in the front-panel 1-2-5 sequence, if possible. If the analyzer defaults to MAX SPAN, an execution warning error is issued.

DEC: Selects the next smaller span/div in the front-panel 1-2-5 sequence, if possible. If the analyzer defaults to zero span, an execution warning error is issued.

MAX: Sweeps the entire frequency range.

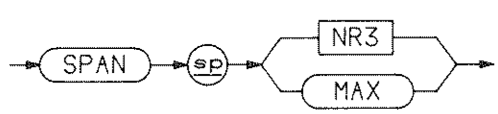
Power-up value: MAX.

SPAN (frequency span/division) query:



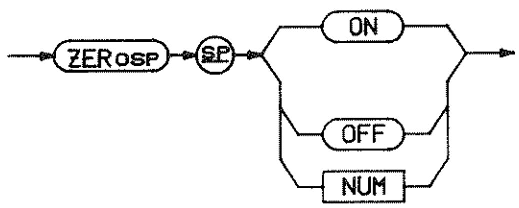
3401-44

Response to SPAN query:



3401-45

ZEROSP (zero span mode) command:



ON: Converts the 496P to a continuous-tune mode with the sweep defeated. Display shifts to the TIME/DIV on the horizontal axis instead of FREQ SPAN/DIV. RESBW (resolution bandwidth) is not affected.

OFF: Cancels ZEROSP ON, leaving the FREQ SPAN/DIV at the previously selected value.

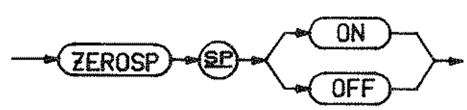
NUM: 1 equals ON; 0 equals OFF. Other numbers are rounded to 1 or 0.

Power-up value: Off.

Interaction: Changing SPAN setting turns ZEROSP off.

ZEROSP (zero span) query:

Response to ZEROSP query:

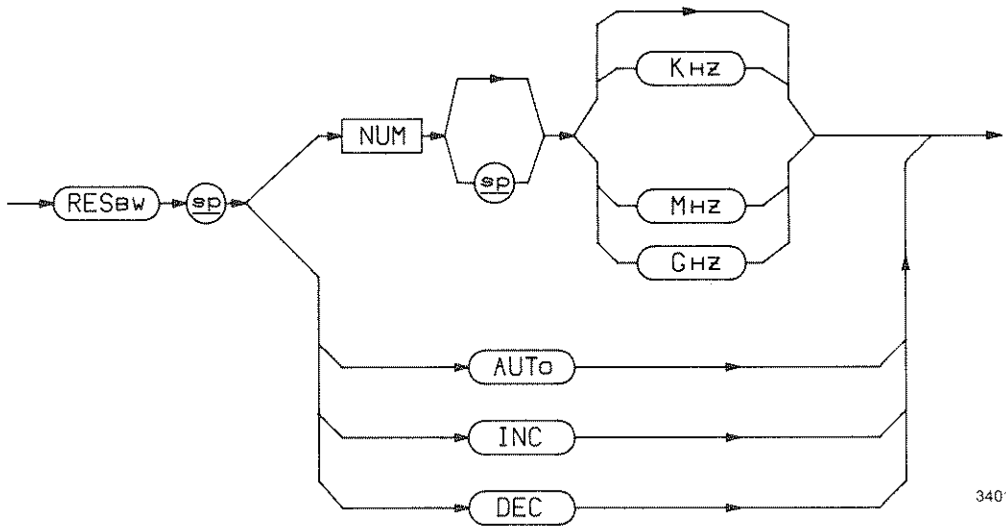


The response is the current zero span condition.

NOTE

It may be preferrable to ZEROSP, rather than SPAN 0 because enabling ZEROSP turns the front-panel indicator on to give positive indication (in addition to the readout). Current resolution bandwidth setting is not destroyed. When ZEROSP is disabled, the previous setting is restored.

RESBW (resolution bandwidth) command:



NUM: Selects the nearest available resolution bandwidth; numbers between bandwidths that can be selected from the front panel are rounded by a geometric algorithm. Positive numbers above or below the range of bandwidth steps are rounded to the nearest step (an eror is issued if the argument is beyond the normal range). Zero and negative numbers cause an error. The geometric algorithm rounds the value to a single significant digit; if the digit is above the breakpoint, the next higher bandwidth step is selected; if the digit is equal to or less than the breakpoint, the lower step is selected. For values above 100, the breakpoint is 3; for values below 100, the breakpoint is 5. For example:

VALUE	Selects
349 kHz	100kHz
350 kHz	1 MHz
55 Hz	1 kHz
55 Hz	100 Hz
19 MHz	Frequency not changed, error reported

AUTO: Selects auto resolution (equivalent to ARES ON).

INC: Selects next larger step (if possible).

DEC: Selects next smaller step (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:

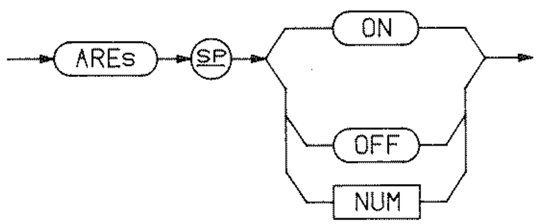
3401-47

Response to RESBW query:

3401-48

Response to SET query includes the AUTO argument (see SET? under System Messages).

ARES (automatic resolution bandwidth) command:



3401-49

ON: The microcomputer matches the current span with an appropriate resolution bandwidth that maintains calibrated performance for the current sweep speed. If automatic sweep speed is active, the microcomputer selects an appropriate resolution bandwidth and changes sweep speed to the fastest sweep allowing calibrated performance. Canceled by any RESBW command except RESBW AUTO.

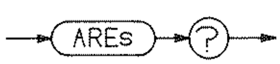
OFF: Cancels ARES ON, leaving resolution bandwidth at current value.

NUM: 1 equals ON; 0 equals OFF. Other numbers are rounded to 1 or 0.

Power-up value: ON.

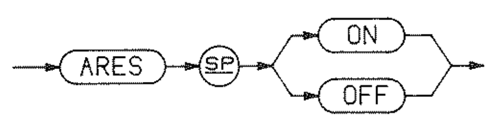
Interaction: ARES OFF also cancels RESBW AUTO.

ARES (automatic bandwidth resolution) query:



3401-50

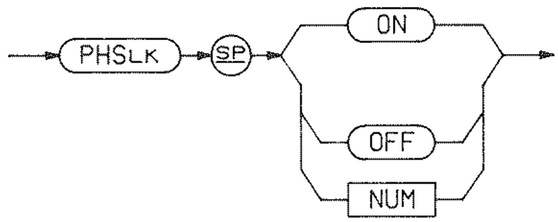
Response to ARES query:



3401-51

ARES is not included in response to SET query.

PHSLK (phaselock) command:



3401-52

OFF: Overrides automatic 1st LO phaselock. PHSLK OFF has an effect only when automatic phaselock is active in spans of 50 kHz or less.

ON: Restores automatic phaselock only if it is currently off and span/div is 50 kHz or less.

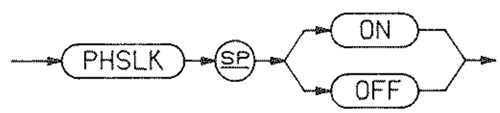
NUM: 1 equals ON; 0 equals OFF. Other numbers are rounded to 1 or 0.

Power-up value: Off.

PHSLK (phaselock) query:

3401-53

Response to PHSLK query:



3401-54

The response is the current phaselock condition.

VERTICAL DISPLAY AND REFERENCE LEVEL (see Fig. 4-4)

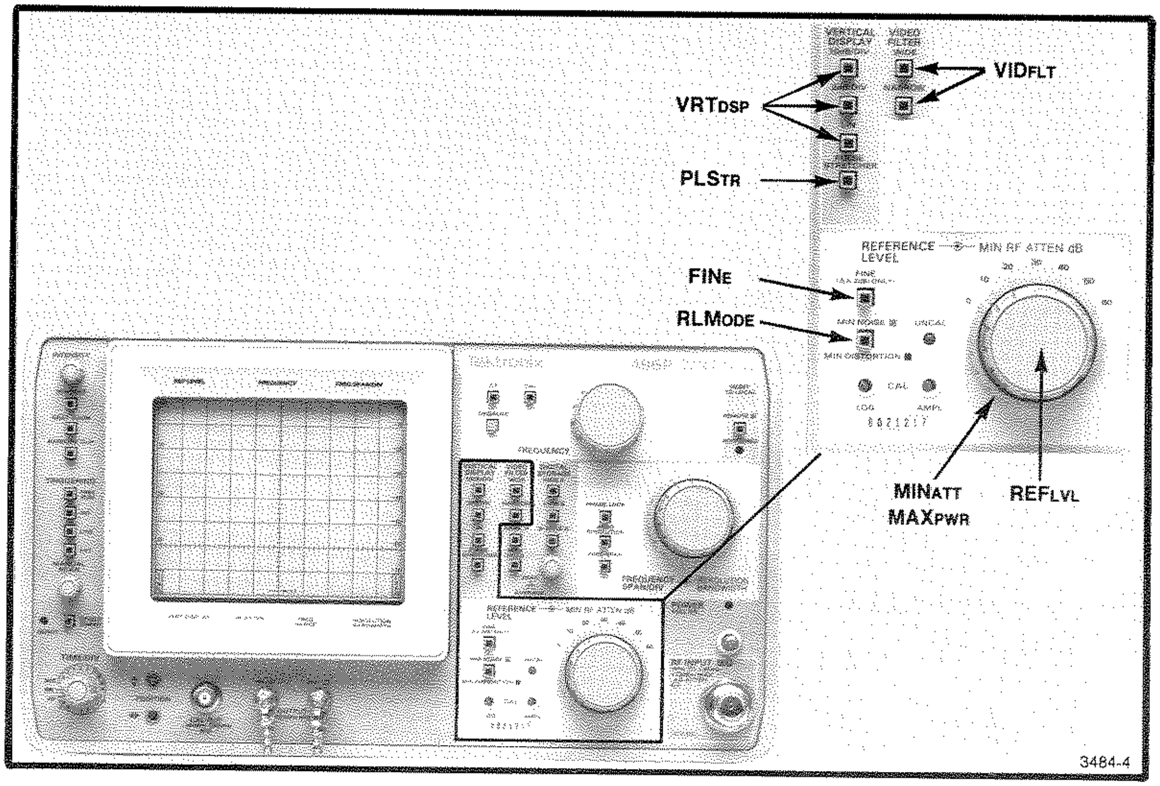
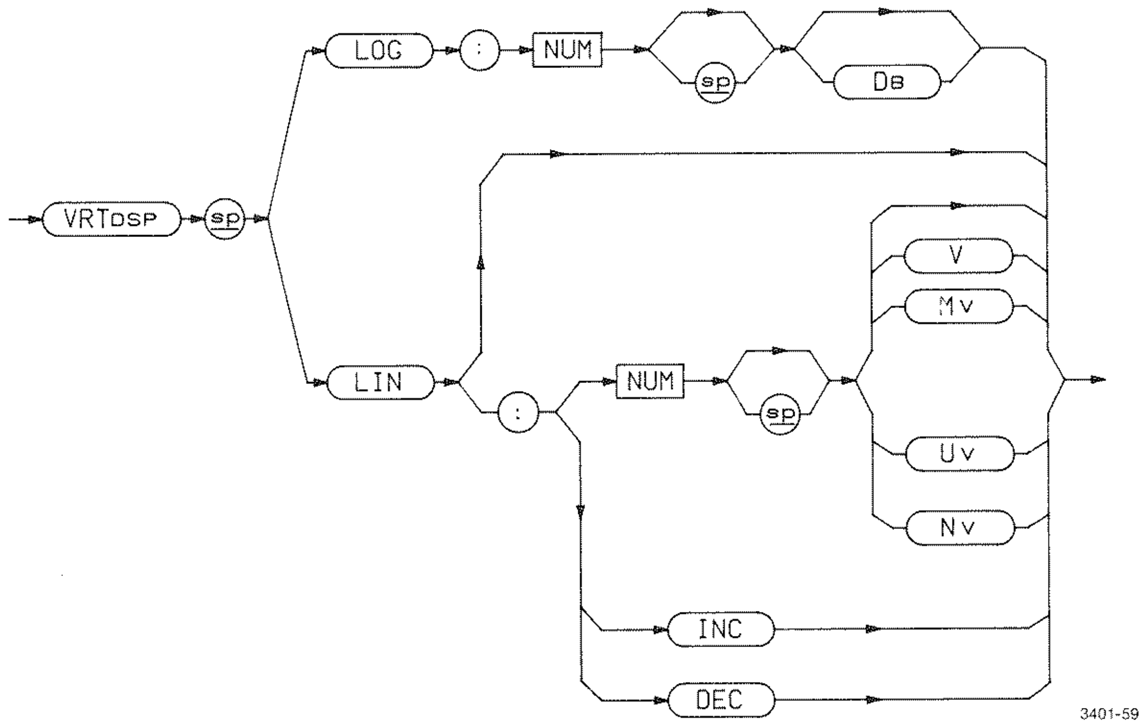


Fig. 4-4. Front-Panel Control—Vertical Display and Reference Level commands.

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), taking into account the least amount of RF attenuation (MINATTI) allowed or maximum power (MAXPWR) expected. 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.

VRTDSP (vertical display) command:



LOG: Scales the display to the dB/division specified by integers in the range 1 to 15 (nonintegers are rounded). Values outside this range cause the 496P to report an error.

LIN: Scales the display in volts/division. NUM is adjusted to the volts equivalent of the nearest 1 dB/division. 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 bumps 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).

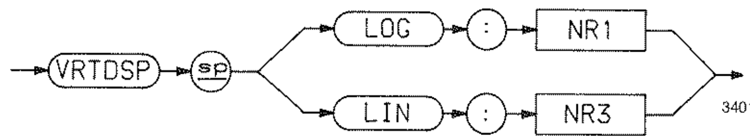
Power-up value: 10 dB/division.

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

VRTDSP (vertical display) query:

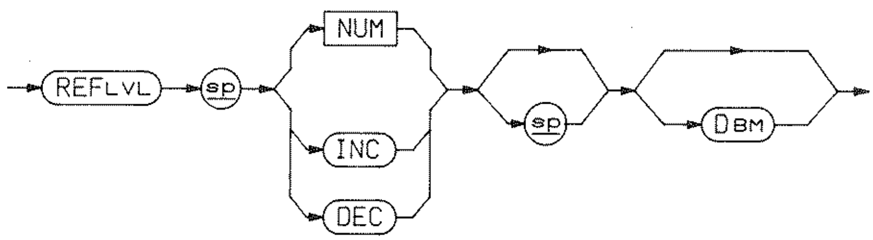
3401-60

Response to VRTDSP query:



3401-61

REFLVL (reference level) command:



3401-62

NUM: The analyzer sets the reference level to the nearest dBm step for a log vertical display (except in Δ A mode) and to the nearest dBm step for a linear vertical display. Delta A mode allows 1/4 dB resolution; the argument is an absolute reference level, not an offset to the present reference level, though the crt readout shows relative amplitude.

INC,DEC: Reference level is stepped up or down once. The step value is determined by the value of VRTDSP and FINE:

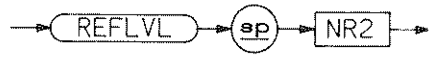
VRTDSP	**Step size**	
Scale factor	**FINE ON**	**FINE OFF**
15 dB	1 dB	Equal to VRTDSP scale factor
14 dB	↓	↓
↓		
4 dB	0.25 dB (Δ A)	
3 dB		
2 dB		1 dB
1 dB		1 dB
LIN	1 dB	6 or 8 dB (varies to match VOLTS/DIV in 1-2-5 sequence)

Power-up value: 30 dBm.

REFLVL (reference level) query:

3401-63

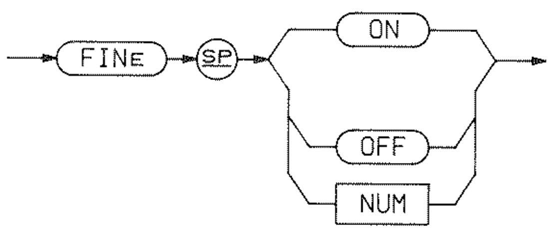
Response to REFLVL query:



3401-64

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

FINE (fine reference level steps) command:



3401-65

ON: Selects small steps for INC/DEC arguments in the reference level and vertical display commands—see REFLVL and VRTDSP for details. With vertical scale factors of 1, 2, 3, and 4 dB/div, FINE ON selects the Δ A mode.

Δ A Mode: Active when both fine resolution 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 log amplifier. This allows signals to be compared with inherently higher relative accuracy over a range of at least +10 dB to −40 dB from the initial level without overloading the analyzer input. The asterisk in the readout remains until the Δ A mode is canceled and the initial gain distribution is changed. This readout is available with UPRDO? while REFLVL? returns the absolute reference level.

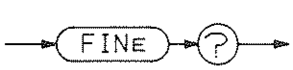
OFF: Restores normal steps for reference level changes; cancels Δ A mode (if active).

NUM: 1 equals ON; 0 equals OFF. Other numbers are rounded to 1 or 0.

Power-up value: Off.

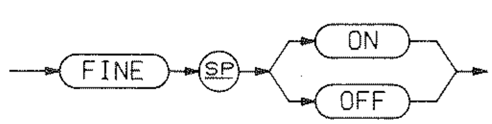
Interaction: This command, in conjunction with VRTDSP, controls the analyzer response to REFLVL INC/DEC.

FINE (find reference level steps) query:



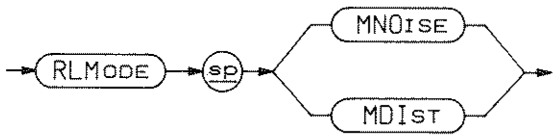
3401-66

Response to FINE query:



3401-67

RLMODE (reference level mode) command:



3401-68

MNOISE: Requests that the microcomputer 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 amplifier noise (but may increase distortion).

MDIST: Requests normal RF attenuation for a given reference level. Generally, this yields 10 dB more RF attenuation than the MNOISE argument and results in lower signal level at the mixer, hence less distortion.

NUM: 0 equals MNOISE, 1 equals MDIST.

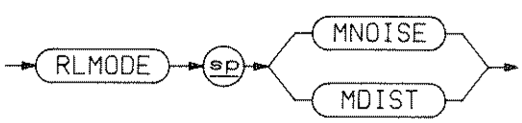
Power-up value: Min distortion.

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

RLMODE (reference level mode) query:

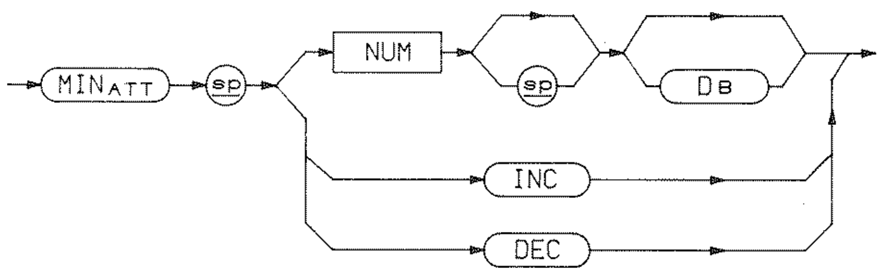
3401-69

Response to RLMODE guery:



3401-70

MINATT (min RF attenuation) command:



3401-71

NUM:	Limits the microcomputer in setting gain distribution, RF attenuation may not be reduced below the attenuator step in the number argument. If NUM is not an even decade from zero to 60, the next higher step (10,20,...60) is selected.
INC,DEC:	Changes the minimum RF attenuation to the next higher or lower step, if any.
Power-up value:	MIN RF ATTEN dB control setting.
Interaction:	Limits range of RF attenuation in response to REFLVL command. Cancels previous limit set by either MINATT or MAXPWR.

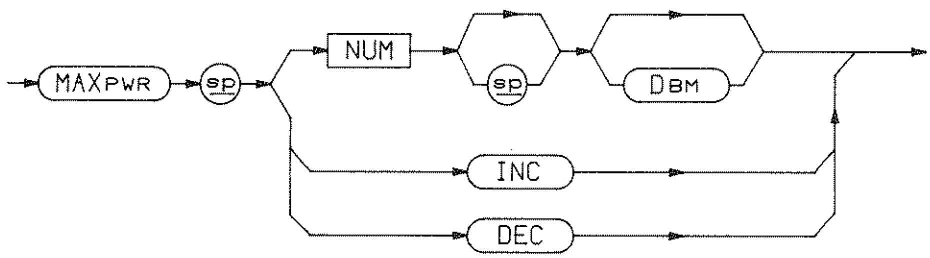
MINATT (minimum RF attenuation) query:

3401-71

Response to MINATT query:

3401-72

MAXPWR (maximum input power) command:



3401-73

NUM: An input to a 496P microcomputer algorithm that protects the RF INPUT from overload at the expected maximum power level. The microcomputer selects a minimum RF attenuation setting such that the NUM signal level is reduced to −18 dBm at the first 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,DEC: Changes the minimum RF attenuation to the next higher or lower step, if any.

Interaction: Limits range of RF attenuation in response to REFLVL command. Cancels previous limit set by either MINATT or MAXPWR.

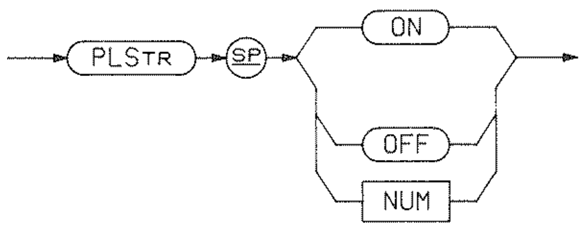
MAXPWR (maximum input power) query:

3401-74

Response to MAXPWR query:

3401-75

PLSTR (pulse stretcher) command:



3401-76

ON: Increases the fall time of detected signals 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 high pulse repetition rates, the rate of change may be too great for the digitizer; the pulse stretcher lengthens the fall time of the signals, allowing accurate measurement.

OFF: Turns off the pulse stretcher.

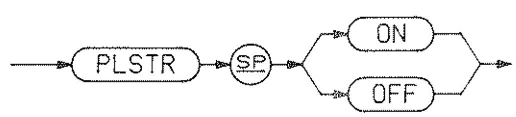
NUM: 1 equals ON; 0 equals OFF. Other numbers are rounded to 1 or 0.

Power-up value: Off.

PLSTR (pulse stretcher) query:

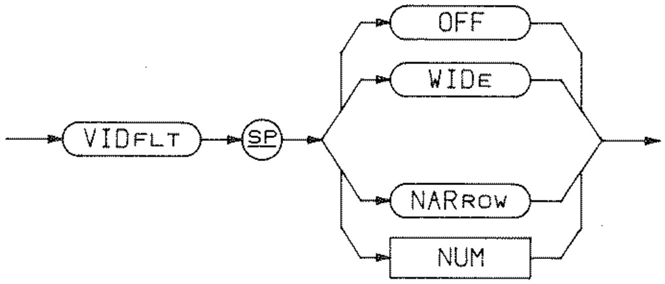
3401-77

Response to PLSTR query:



3401-78

VIDFLT (video filter) command:



3401-79

OFF: Turns off both video filters.

WIDE: Turns on a filter in the video amplifier (after the detector) to reduce noise in the display. The wide filter reduces video bandwidth to about 1/30 of the selected resolution bandwidth.

NARROW: Magnifies effect of wide filter 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; 1 equals WIDE; 2 equals NARROW. Other numbers are rounded.

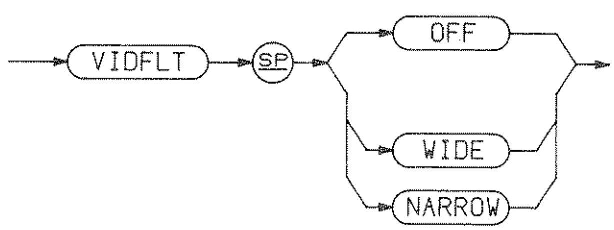
Power-up value: Off.

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:

3401-80

Response to VIDFLT query:



3401-81

SWEEP CONTROL (see Fig. 4-5)

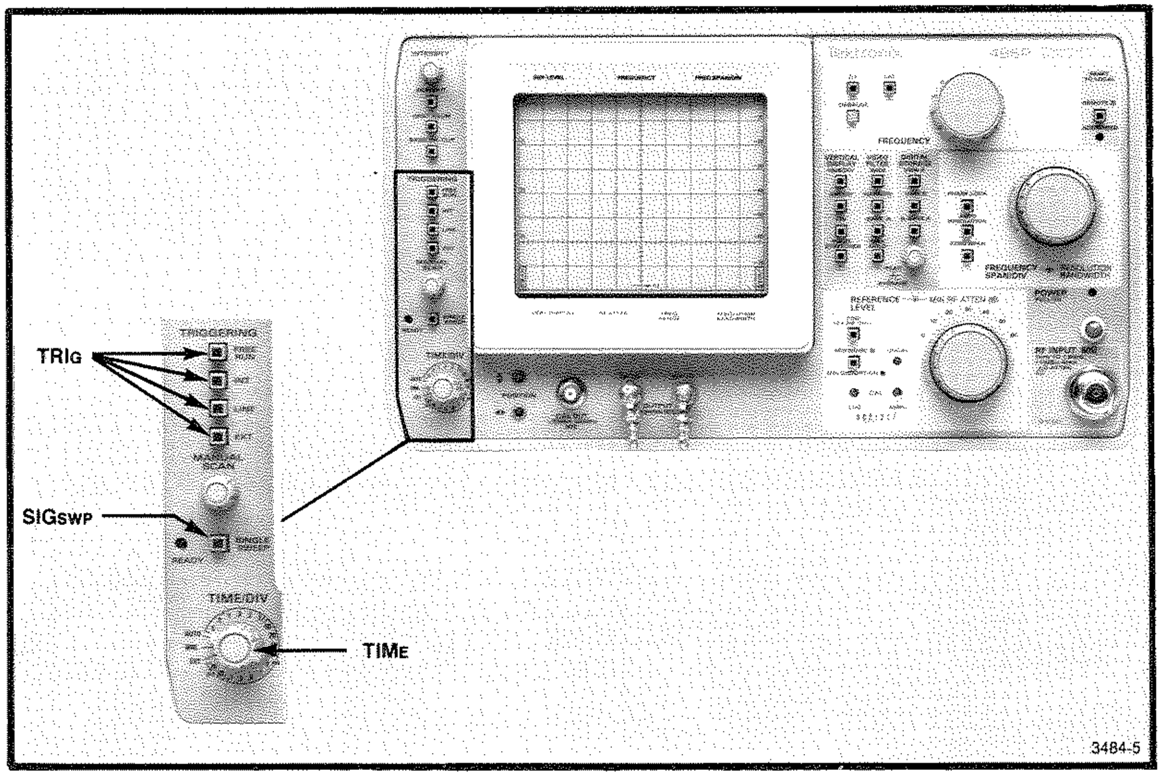
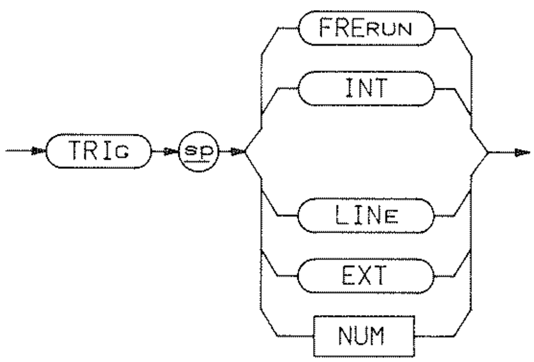


Fig. 4-5. Front-Panel Control—Sweep Control commands.

Three commands control the 496P 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). By selecting a free-running sweep with automatic sweep rate, the programmer can direct the 496P to match the sweep to related analyzer parameters automatically. Other options include manual or external analog control of the sweep.

TRIG (triggering) command:



3401-83

FRERUN: Allows the analyzer sweep to run repetitively; a trigger is not required (and is ignored), so the analyzer generates a sweep immediately after the hold-off period following the previous sweep. This is a simple and common setup for acquiring a spectrum.

INT: The analyzer generates a sweep only when triggered by an input signal at the frequency to which the analyzer is tuned. A signal amplitude of at least 2 divisions is required and must occur after the hold-off period following the previous sweep. This sweep mode is often used for examining time-domain signals in zero-span mode (ZEROSP).

LINE: Selects the power line input as the trigger signal—useful in both the frequency and time domain for signals with components related to the power line frequency.

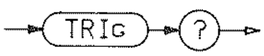
EXT: A signal with an amplitude of at least +1.0 volt connected to EXT IN TRIG on the rear panel triggers the sweep for pulsed signals.

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.

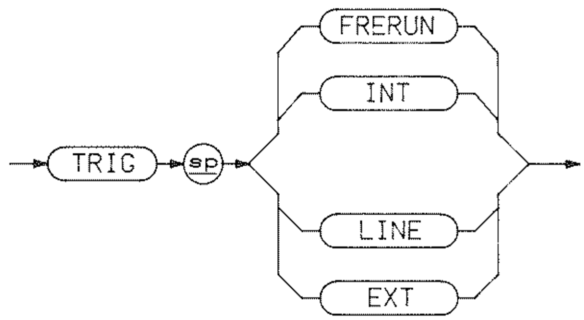
Interaction: The signal frequency required for internal trigger is related to the center frequency. In the frequency domain, the required frequency corresponds to the left graticule edge; in the time domain, the required frequency is the center frequency. In the frequency domain, the triggering frequency must be less than 1.8 GHz.

TRIG (triggering) query:



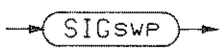
3401-84

Response to TRIG query:



3401-85

SIGSWP (single-sweep) command:



3401-86

SIGSWP: On the first SIGSWP command, the analyzer enters single-sweep mode, aborting the current sweep. Once in a single-sweep mode, this command arms the sweep and lights READY on the front panel, which remains lighted until the sweep completes. The analyzer makes a single sweep of the selected spectrum when the conditions determined by the TRIG command are met.

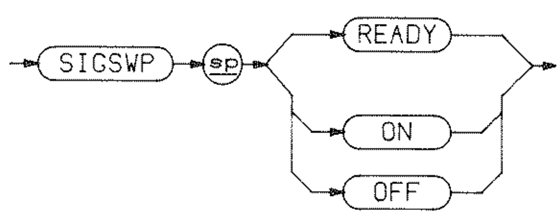
Power-up value: Off:

Interaction: Any TRIG command cancels single-sweep mode.

SIGSWP (single-sweep) query:

3401-87

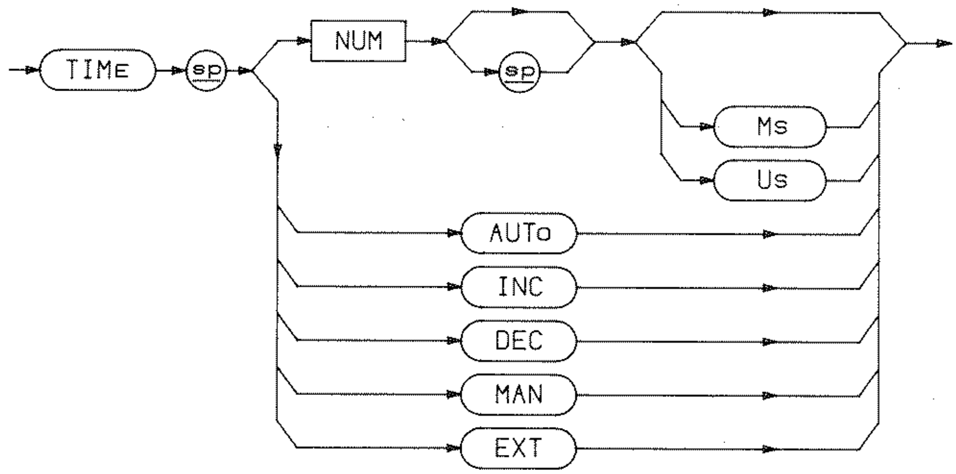
Response to SIGSWP query:



3401-88

The single-sweep function is omitted in the SET query response if single-sweep is not active (see SET? in Section 7).

TIME (time/div) command:



3401-89

NUM: 1-2-5 sequency in the range 20 E−6 to 10, numbers not in this sequence are rounded to the nearest step.

AUTO: Requests the 496P microcomputer to select the fastest sweep allowed for calibrated response.

INC,DEC: The sweep rate is bumped ±1 in the sequence, if possible.

MAN: The sweep is coupled to the MANUAL SCAN control; the horizontal position of the crt beam and the analyzer front-end tuning are varied from the center of the sweep and center of the selected spectrum as the control is rotated. The operator can manually scan the spectrum.

EXT: The sweep is coupled to EXT IN SWEEP on the 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 volts scans the spectrum.

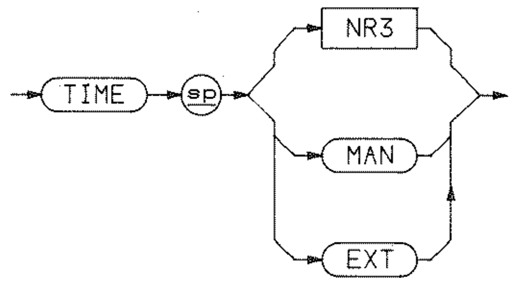
Power-up value: TIME/DIV control setting.

Interaction: Too fast a sweep speed for a given reoslution bandwidth uncalibrates the display. For digital storage, 2 milliseconds/division is the maximum usable sweep rate.

TIME (time/div) query:

3401-90

Response to TIME query:



3401-91

The SET? response includes AUTO as a possible argument (see SET? in Section 7).

DIGITAL STORAGE (see Fig. 4-6)

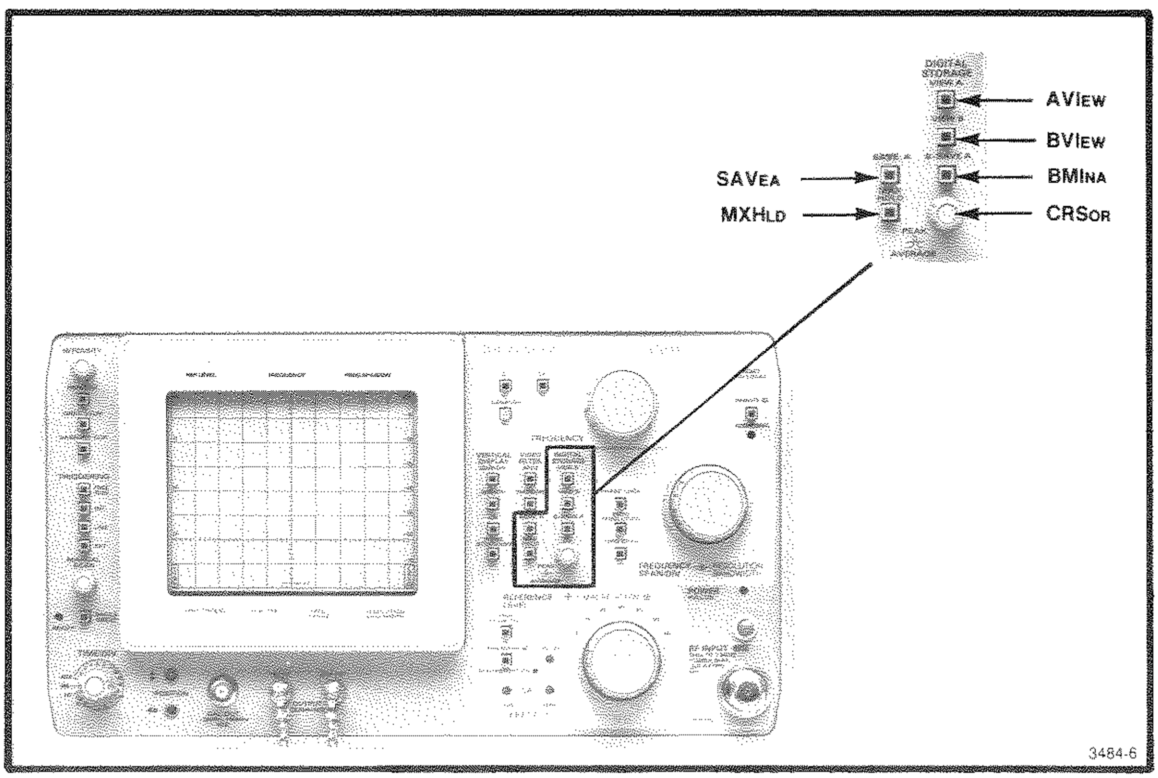
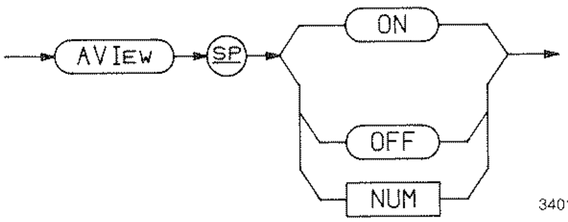


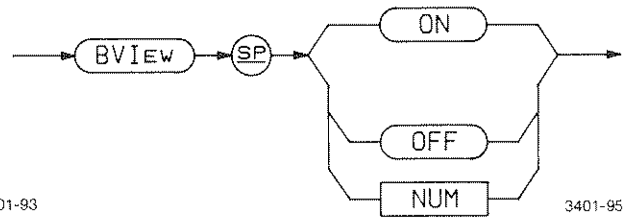
Fig. 4-6. Front-Panel Control—Digital Storage commands.

These commands control the following 496P digital storage functions: display (AVIEW, BVIEW), updating (SAVEA), display and updating (BMINA), and digitizer control (MXHLD, CRSOR).

AVIEW, BVIEW (A and B memory display) commands:



3401-93



3401-95

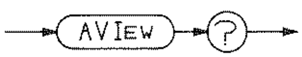
ON: The contents of the specified memory are displayed on the crt. The memories are independent and may be displayed together or separately.

OFF: The display of the specified memory is turned off for a real-time display of the input signal.

NUM: 1 equals ON; 0 equals OFF. Other numbers are rounded to 1 or 0.

Power-up value: Both A and B on.

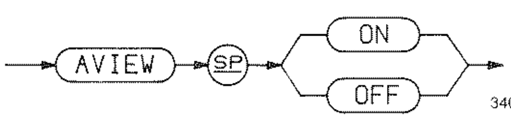
AVIEW and BVIEW (A and B memory display) queries:



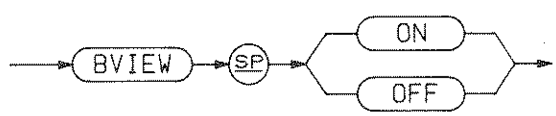
3401-96

3401-97

Response to AVIEW and BVIEW queries:

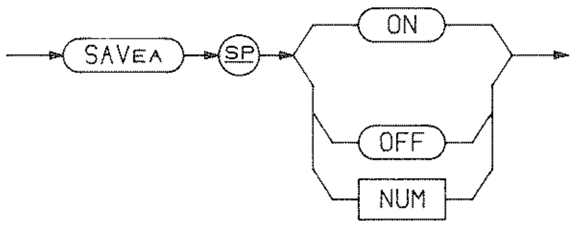


3401-98



3401-99

SAVEA (save A memory) command:



3401-100

ON: Stops updating of A memory; current contents are saved. This allows comparison with B memory, which is continuously updated.

OFF: Resumes updating of A memory.

NUM: 1 equals ON; 0 equals OFF. Other numbers are rounded to 1 or 0.

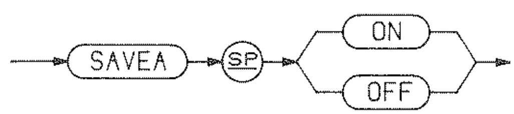
Power-up value: On.

Interaction: Set to ON by BMINA ON. SAVEA OFF cancels BMINA ON.

SAVEA (save A memory) query:

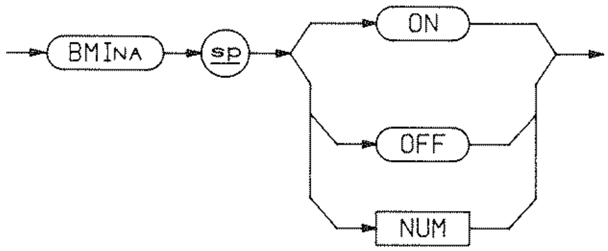
3401-101

Response to SAVEA query:



3401-102

BMINA (B−A memory display) command:



3401-103

ON: The 496P microcomputer turns on SAVEA and then turns on a display of the difference between the A memory and the B memory, which is continuously updated. Normally the difference trace baseline is set at graticule center, but may be varied internally.

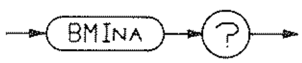
OFF: Turns off the difference display.

NUM: 1 equals ON; 0 equals OFF. Other numbers are rounded to 1 or 0.

Power-up value: Off.

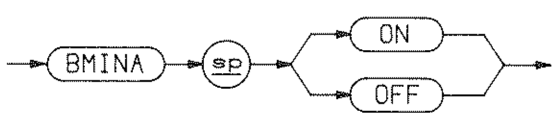
Interaction: BMINA ON turns on SAVE A. SAVEA OFF turns off BMINA.

BMINA (B−A memory display) query:



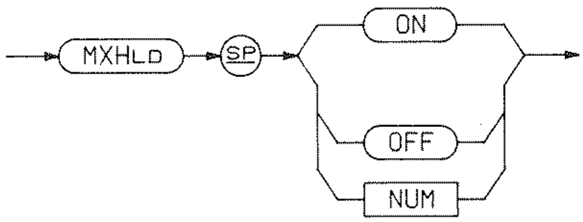
3401-104

Response to BMINA query:



3401-105

MXHLD (max hold) command:



3401-106

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

OFF: B memory is continuously updated; A memory is updated if SAVEA is off.

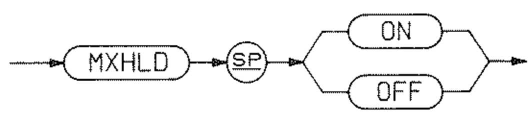
NUM: 1 equals ON; 0 equals OFF. Other numbers are rounded to 1 or 0.

Power-up value: Off.

MXHLD (max hold) query:

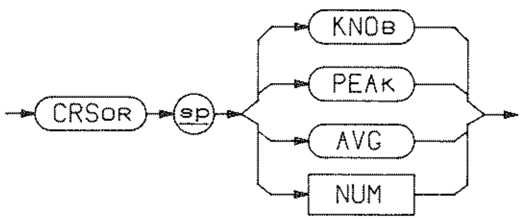
3401-107

Response to MXHLD query:



3401-108

CRSOR (peak/average cursor) command:



3401-109

KNOB: Activates the PEAK/AVERAGE knob so the operator can set the cursor level, which is shown by a line across the crt. Above the line, peak values are stored in memory 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 position of the cursor 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 memories, regardless of the position of the cursor last set by KNOB. Each value stored is the mean of the data digitized at a single point during a sweep. PEAK AVG is the same as setting the cursor to its highest (maximum) position.

NUM: 0 equals KNOB; 1 equals PEAK; 2 equals AVG. Other numbers are rounded to 0, 1, or 2.

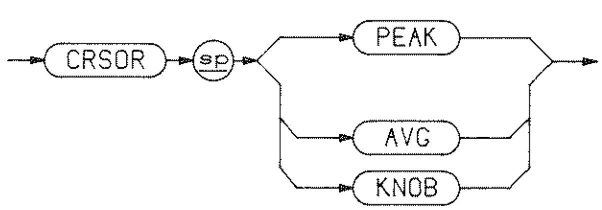
Interaction: Averaging can reduce the value in digital storage for signals with very narrow response or pulsed signals.

Power-up value: Knob.

CRSOR (peak/average cursor) query:

3401-110

Response to CRSOR query:



3401-111

DISPLAY CONTROL (see Fig. 4-7)

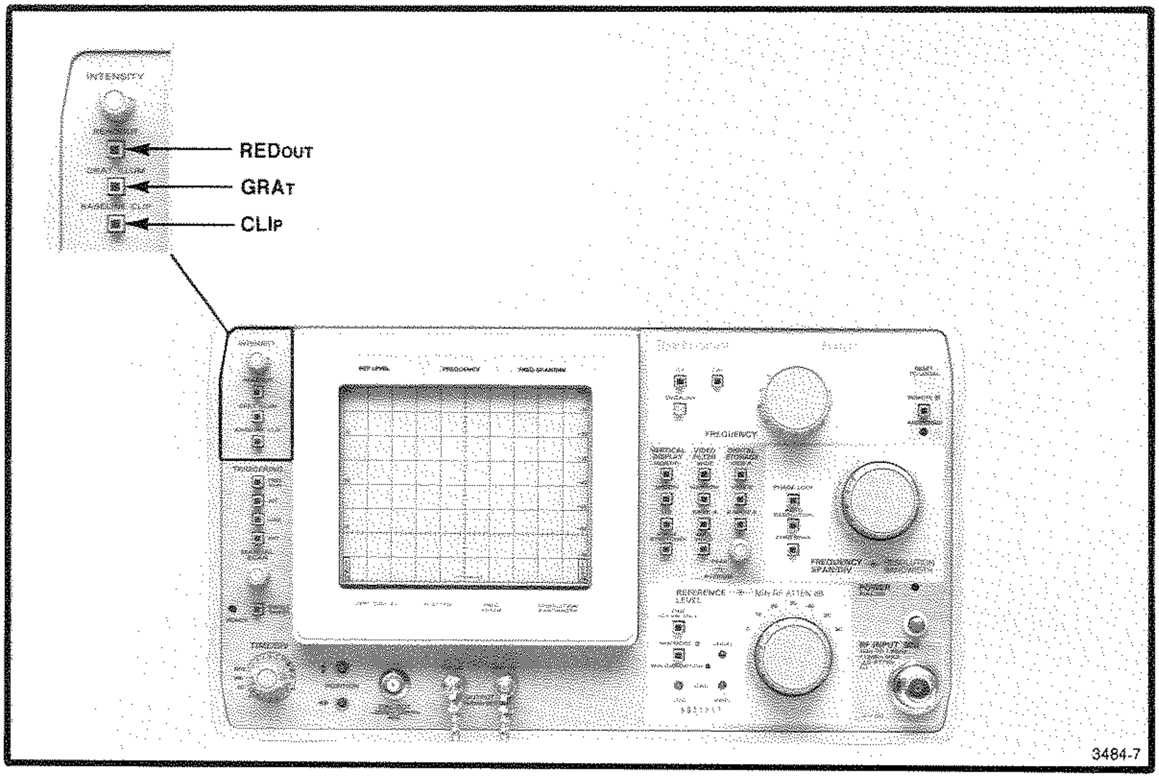
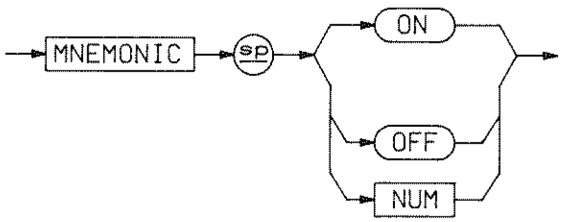


Fig. 4-7. Front-Panel Control—Display Control commands.

These commands control crt display functions: REDOUT (readout display), GRAT (graticule light), and CLIP (baseline clipper). Each can be turned on or off and queried.

Display control command:



3401-113

The mnemonics are:

Mnemonic	ON	OFF
RED or REDOUT	Display readout characters	Blank readout
GRA or GRAT	Light graticule	Dark graticule
CLI or CLIP	Blank trace at crt bottom	Full trace

NUM: 1 equals ON; 0 equals OFF. Other numbers are rounded to 1 or 0.

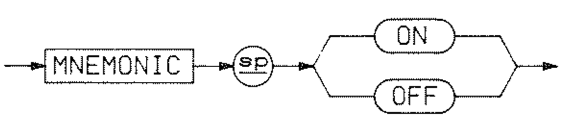
Power-up values:

Command	Value
REDOUT	ON
GRAT	OFF
CLIP	OFF

Display control query:

3401-114

Response to display control:



3401-115

DISPLAY DATA AND CRT READOUT I/O

These commands and queries transfer display (measurement) data and readout data to or from the 496P.

Message Unit	Function
WFMPRE	Selects A/B/FULL, ASCII/binary data transfers
WFMPRE?	Requests waveform parameters from 496P
CURVE	Sends display (trace) data to 496P
CURVE?	Requests display (trace) data from 496P
RDOUT	Sends one line of crt readout to 496P
UPRDO?	Requests top line of crt readout from 496P
LORDO?	Requests bottom line of crt readout from 496P

WAVEFORM TRANSFERS

The 496P follows the Tektronix Codes & Formats standard procedure for waveform transfers. The transfer is handled in two parts: a preamble (WFMPRE) that identifies and scales the data; and the data that represents the waveform (CURVE). The preamble allows two kinds of choices:

1) either A or B memory or both (FULL); and

2) ASCII-coded decimal or binary trace data.

The 496P defaults at power-up to FULL and ASCII.

Depending on the memory contents, a half- or full-resolution waveform is obtained, or two different waveforms. This follows from the way digital storage is partitioned.

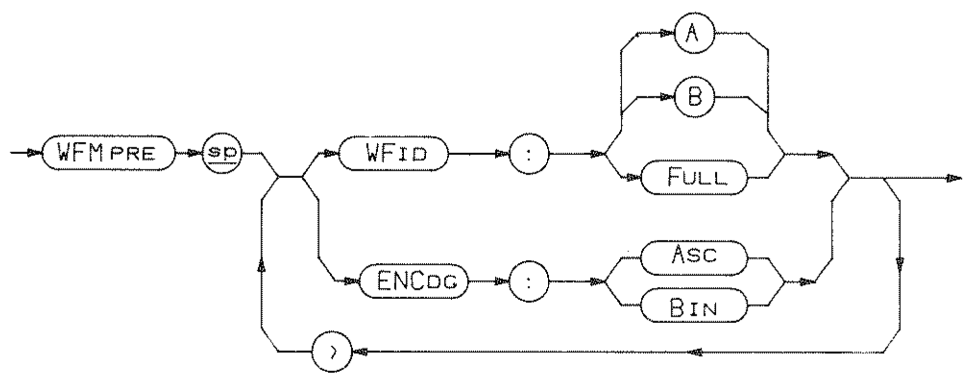
The B memory is updated with each sweep; the A memory is updated only if SAVEA is OFF. The values stored in each memory are alternate points on the current display (B0, A0, B1, A1, B2, A2,...).

With SAVEA OFF, each memory is a half-resolution replica of data from the last sweep (A data points offset from corresponding B data points by 1). Full-resolution (FULL) transfers merge the two memories for 1000 data points (100 points/div), and half-resolution transfers (A or B) divide the memories for 500 data points (50 points/div). Asking for only A or B halves the amount of data to be transferred (saving time, memory, etc.) but risks losing spectrum detail. For example, a signal resolved to a single point (with very narrow resolution bandwidth compared to span) would appear in either A or B, but not both.

With SAVEA ON, only B is filled with data from the current sweep. This divides digital storage, allowing two waveforms to be acquired, so half-resolution transfers can involve two unrelated waveforms.

See Section 8 for more discussion of waveform transfers and waveform storage and display using 4050-Series controllers.

WFMPRE (waveform preamble) command:



3401-116

WFID: Selects either A or B memory or both A and B (FULL) memory for following data transfers and waveform processing.

ENCDG: Selects either ASCII-coded decimal numbers or binary numbers for data transfer.

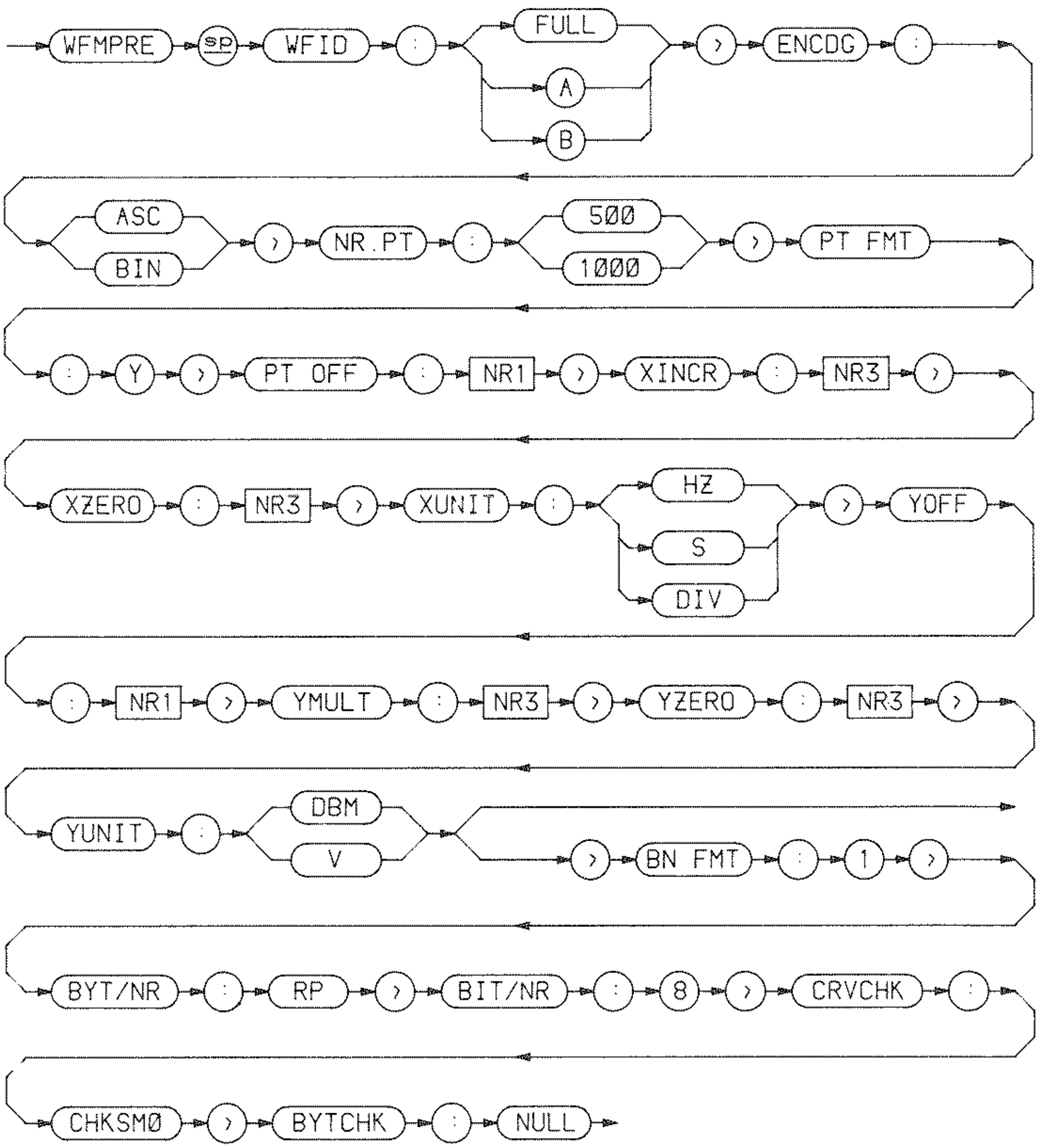
Examples:
WFMPRE WFID:FULL
WFMPRE ENCDG:ASC
WFM WFID:A,ENC:B

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

WFMPRE (waveform preamble) query:

3401-117

Response to WFMPRE query:



3401-118

Items that follow the waveform identification and coding specify other data packet parameters such as 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 are Y (display vertical) values. The data are 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:CHKSMO	Specifies that the last byte of a binary transfer is a 2's-complement, modulo-256 checksum for the preceding bytes (except for the percent-sign parser at the beginning).
BYTCHK:NULL	Indicates no byte check is appended to binary data transfers.

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

$$XN = XZERO + XINCR * (N - PT.OFF)$$

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 = 448 MHZ for FULL with SPAN MAX.
XN = 2 milliseconds 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:

$$YN = YZERO + YMULT * (VALN - YOFF)$$

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.

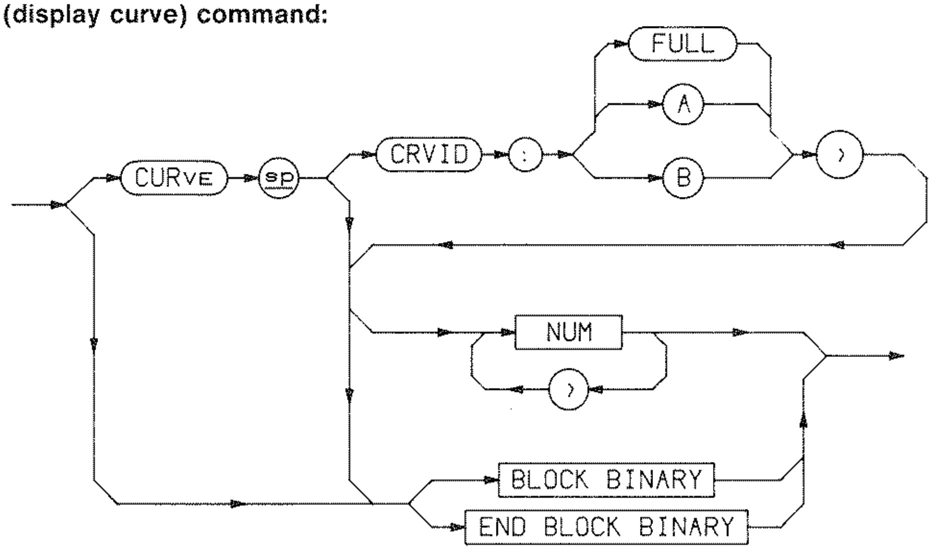
YOFF — Is 225 (top edge of graticule) in log vertical display mode and 25 (bottom edge of graticule) in linear vertical display mode.

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

YN = −40 at 10 dB/div with a reference level of 0 dBm.
YN = 0.112 in linear mode with a reference level of 0 dBm.

The WFMPRE poriton of the SET? response includes only the WFID and ENCDG arguments.

CURVE (display curve) command:



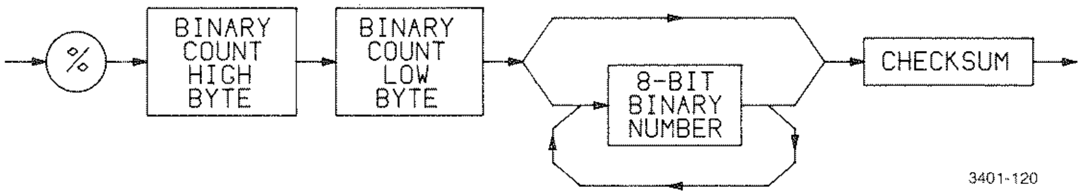
3401-119

CRVID: Specifies portion of display data memory for input. If this argument is omitted, the last CRVID in a CURVE command or WFMID in a WFMPRE command is assumed. A or B indicates a 500-point transfer; FULL indicates 1000 points.

NUM: A sequence of ASCII-coded digits, delimited by commas between successive numbers.

@

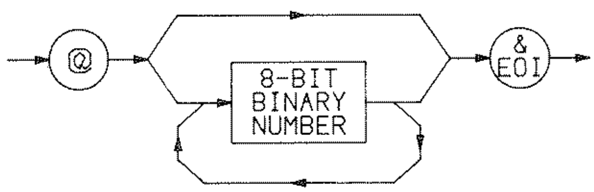
BLOCK BINARY:



3401-120

Block binary is a sequence of binary numbers, preceded by the ASCII code for "%" and a two-byte binary integer representing the number of binary numbers plus one (the extra byte being 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-providing an error-check of the binary block transfer.

END BLOCK:



3401-121

End block binary is a sequence of binary numbers, preceded by the ASCII code for "@"; EOI must be asserted concurrently with the last data byte, which must end the message.

Examples:

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

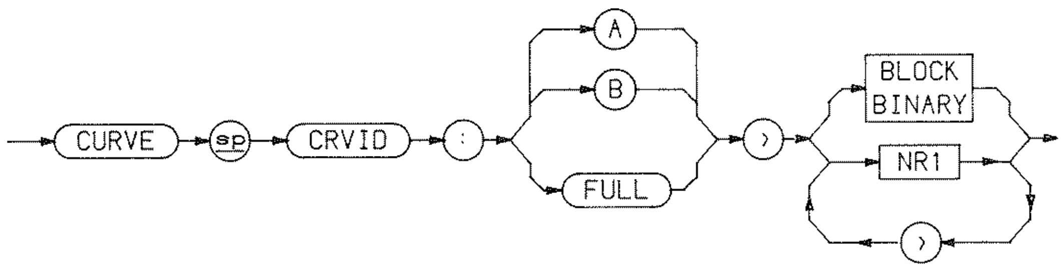
Interaction:

CURVE input is overwritten in the display i/o buffer if preceded by a CURVE query in the same message, causing the old data to be put back into digital storage. See Section 8 for more on waveform storage and i/o.

CURVE (display curve) query:

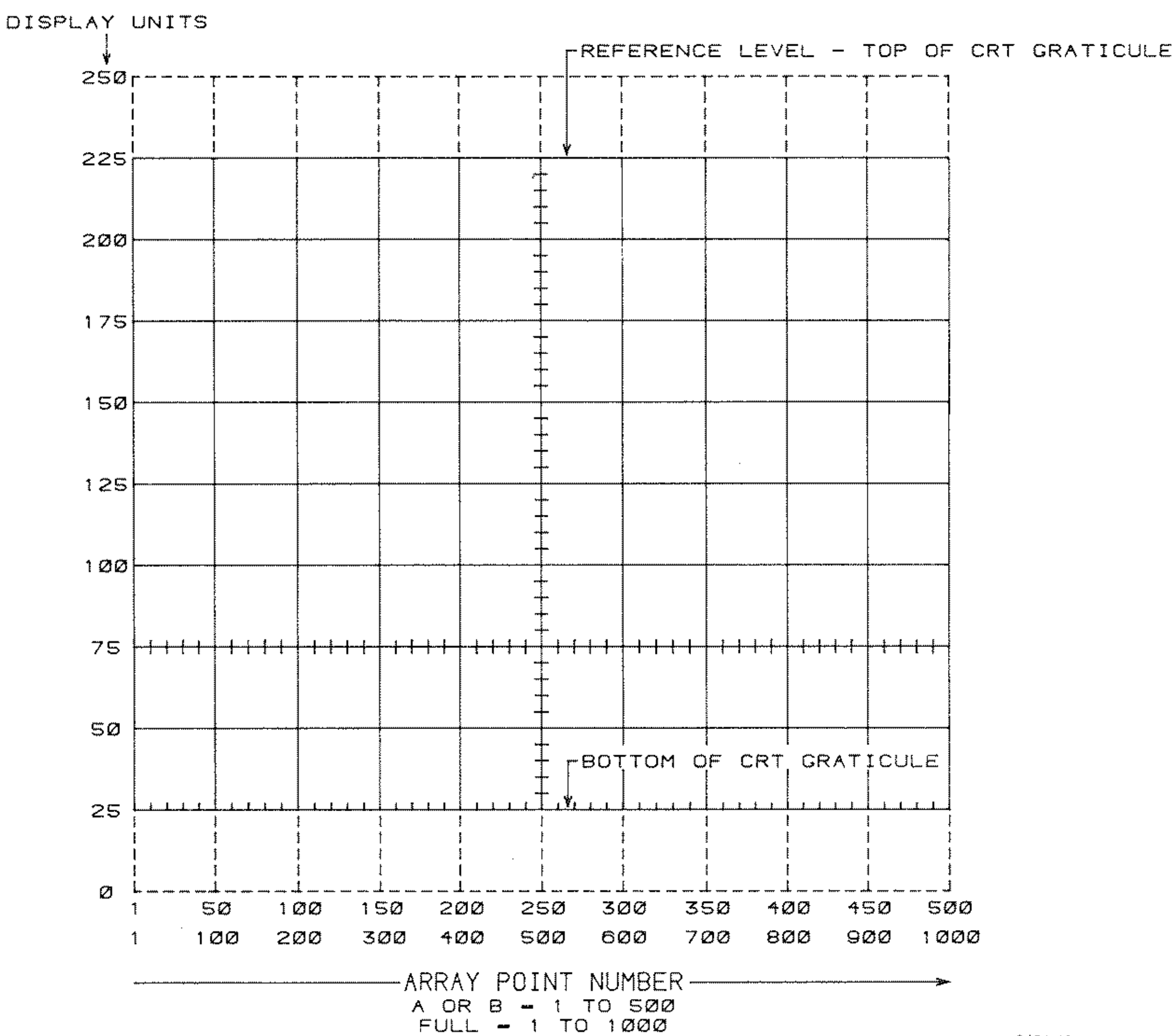
3401-122

Response to CURVE query:



3401-123

Waveform data is related to the display by the following figure:

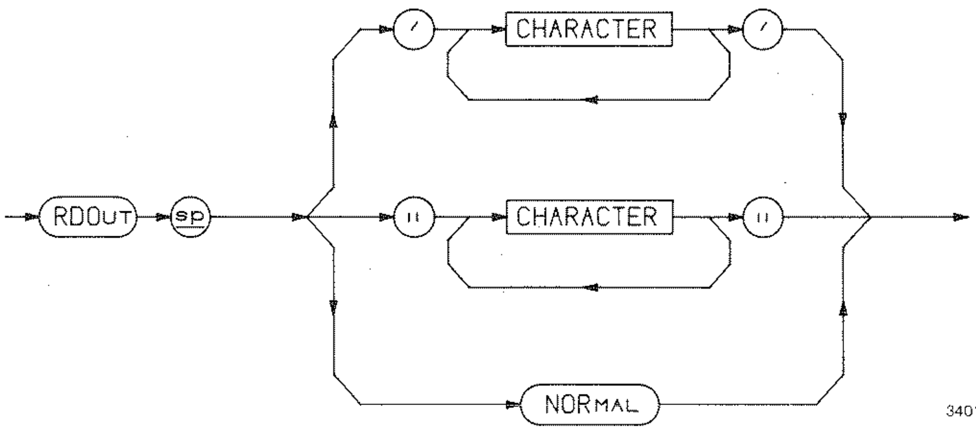


3401-124

CRT READOUT TRANSFERS

Two crt readout queries return the upper row of characters (UPRDO) or lower row (LORDO). A readout command (RDOUT) displays messages on the crt.

RDOUT (readout message) command:



3401-125

CHARACTER: The first 32 ASCII-coded characters are displayed across the bottom of the 496P crt. The previous bottom row of characters is moved to the top and the previous top row discarded. Thus, each new RDOUT command causes the 496P readout to scroll. Lower-case characters are displayed upper-case.

Use a pair of single-quote marks (' ') rather than 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 @A:"RDOUT 'STAND BY FOR IMPORTANT MESSAGE'"
```

NORMAL: Restores normal 496P readout.

Examples:

RDOUT "TEKTRONIX 496P"
RDO 'SET CONTROLS AS DESIRED'
RDOUT NOR

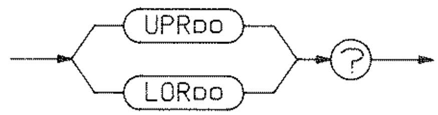
Power-up values: Normal readout.

Interaction: If a crt message sent with a RDOUT command remains after the analyzer is returned to local control, the operator can restore normal readout by changing a control that causes the normal readout to be updated.

REDOUT ON is required to see any crt readout.

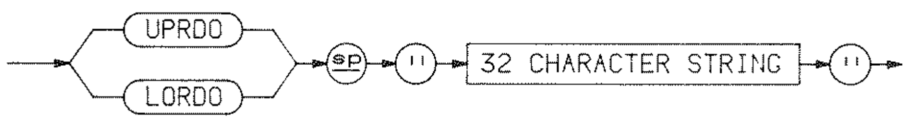
There is no RDOUT query.

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



3401-126

Response to UPRDO and LORDO queries:



3401-127

CHARACTER: ASCII characters in the upper or lower row of normal crt readout. Blanks are transmitted as spaces. Only normal readout is returned by the query, not a message sent to the instrument by RDOUT.

WAVEFORM PROCESSING

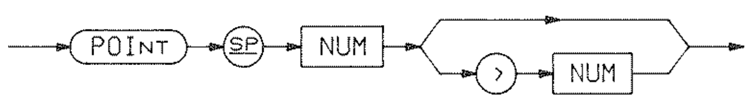
A group of comands allow local processing of spectrum data by the 496P microcomputer. Some of these commands operate on a display data point. This is an ordered pair—an X and a Y value—that corresponds to a point on the 496P display. On command, the 496P microcomputer acquires a display data point from the current digital storage waveform. The point is held in microcomputer memory until another command updates the data point. A query requests that the 496P report the point. Other commands change analyzer settings automatically to zero in on the point.

Commands that 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 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 commands and query work. Two programs at the end of Section 2 show some of these commands in use. For more help in how to use 496P waveform processing, see Section 8.

POINT (display data point) command:



3401-128

First NUM: 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: 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 trace 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.

Examples:

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

Power-up value: 500,225.

Interaction: Sending the SET? response 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 also Display Data Point Commands Interaction later in this section.

POINT (display data point) query:

3401-129

Response to POINT query:

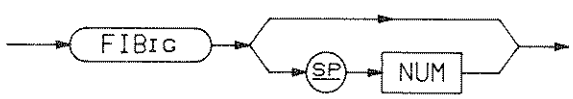
3401-130

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 in either of two cases;

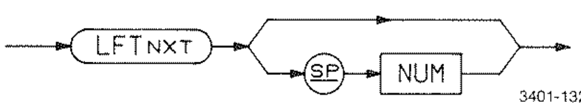
1) if the Y value was set by a POINT command; or

2) if digital storage was updated after the microcomputer acquired the display data point from digital storage.

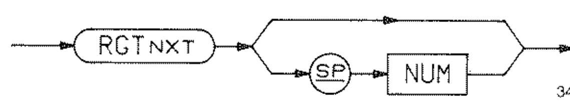
FIBIG, LFTNXT, RGTNXT (signal search) commands:



3401-131



3401-132



3401-133

FIBIG (find big): Acquires the point at the largest signal peak with a value greater than NUM. If a signal peak greater than NUM is not found, the display data point is set to 500,0.

LFTNXT (left-next): Acquires the point at the peak of the next signal to the left of the current point. If a signal peak greater than NUM is not found, the display data point is set to 0,0.

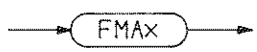
RGTNXT (right-next): Acquires the point at the peak of the next signal to the right of the current point. If a signal peak greater than NUM is not found, the display data point is set to 1001,0.

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.

Examples: FIBIG 100
LFTNXT
RGT

Interaction: See Display Data Point Commands Interaction later in this section.

FMAX (find maximum value) command:



3401-134

This routine acquires the point in digital storage with the largest value for the display data point. Because it looks for the largest value without regard to whether it is the peak of a signal, FMAX runs faster than FIBIG. If more than one point contains the largest value found, the first (left-most) point is acquired.

Interaction: See Display Data Point Commands Interaction later in this section.

FMIN (minimum value) command:

3401-135

This routine acquires the point in digital storage with the smallest value for the display data point. If more than one point contains the smallest value found, the first (left-most) point is acquired.

Display Data Point Commands Interaction

1. The preceding waveform-processing commands operate only on the portion of memory specified by the last WFMPRE or CURVE command: either A or B or full (both A and B). The portion of memory involved is first copied into a buffer; if 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 horizontal scale 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; commands for these two functions can interact if executed as part of the same message. Section 8 explains multiple use of the display data buffer further; what follows here is an explanation of how this impacts the display data point commands.

Display data output may be invalid if CURVE? is followed by a command to update the display data point *and* digital storage is updated during execution of the message (either by repetitive sweeps or by executing a SIGSWP command). Under these conditions, the curve data output (which follows execution of the entire message) will not be the data loaded in the buffer when CURVE? was executed. Instead, it will be the data loaded by the later command to update the display data point (overwriting the data loaded in the buffer when CURVE? was executed). If CURVE? follows, rather than precedes the display data point command, no conflict occurs in the way the commands use the buffer, and the curve data is output as expected.

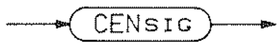
Display data input may be invalid if CURVE is preceded by a command to update the display data point. In this condition, curve data is loaded into the buffer as it is received from the GPIB, but is overwritten when the display data point command executes. Thus, the data loaded into the buffer from digital storage is written right back into digital storage when the following CURVE command is executed, and the curve data sent to the instrument is lost.

There is an exception to the potential interaction described above: when a Y as well as an X value is sent with the POINT command, the microcomputer does not read digital storage data into the buffer.

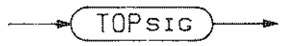
3. VRTDSP LIN interacts with FIBIG, RGTNXT, and LFTNXT because they transform linear data into logarithmic data before executing. This is not apparent unless this transformed data is output over the GPIB or loaded into digital storage because of one of the conditions noted in part 2.

How this interaction arises can be better understood by reading the discussion of Multiple Use of Display Buffer for Waveform Processing and I/O in Section 8.

CENSIG, TOPSIG (center or move signal) commands:



3401-136



3401-137

CENSIG (center signal): TUNES frequency to center the signal represented by the display data point (or as close as possible given the specified span and tuning accuracies).

TOPSIG (top signal): 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 executes, the display data point may no longer match the signal of interest.

SYSTEM COMMANDS AND QUERIES

496P device-dependent message units are provided to set and return parameters of use to the controller in a GPIB system. These commands and queries are described in this section in three groups related to: instrument parameters, message execution, and status and error reporting.

Two queries and a command acquire and initialize instrument parameters:

Message Unit	Function
SET?	Returns values of programmable parameters
INIT	Resets programmable parameters to power-up values
ID?	Returns model number, firmware version, and options.

Two commands control execution of messages:

Message Unit	Function
WAIT	Synchronizes execution of GPIB messages with sweep
REPEAT	Repeats execution of previous units in message

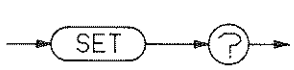
Commands and queries that handle status and error reporting (including serial poll response) are:

Message Unit	Function
EOS,EOS?	Turns on/off, queries end-of-sweep SRQ function
RQS,RQS?	Turns on/off, queries abnormal SRQ function
Status byte	Serial poll response
ERCNT?	Returns number of errors to be reported
ERR?	Returns a code for the current error report

INSTRUMENT PARAMETERS

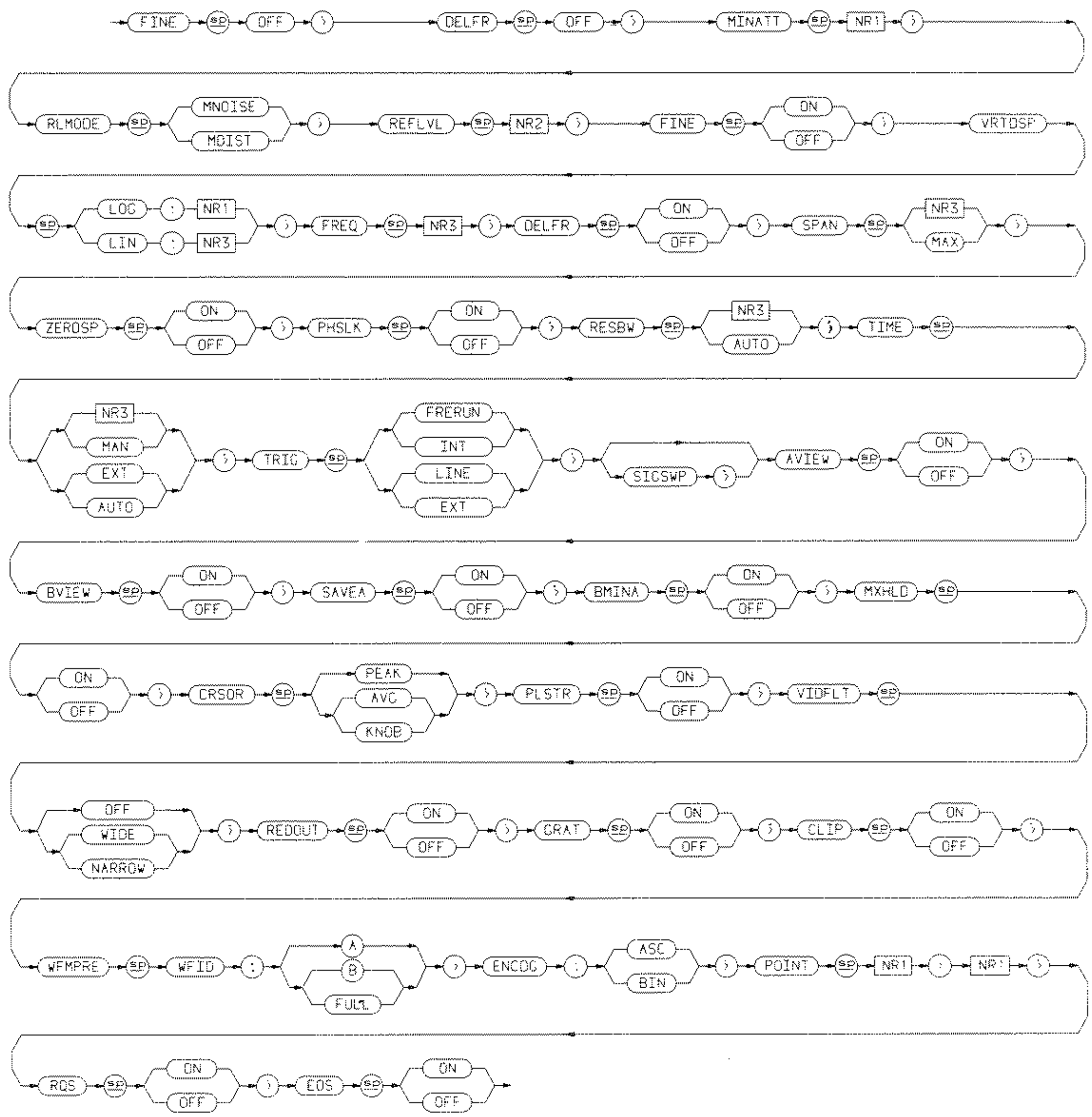
The queries (SET? and ID?) and command (INIT) in this group return settings and identification parameters and initialize settings.

Set (instrument settings) query:



3401-138

Response to SET query:



The instrument returns a string of commands, preceded by FINE OFF. This string can be "learned" for later transfer to the 496P 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 is turned off before the setup begins.

INIT (initialize settings) command:

3401-140

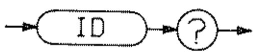
INIT resets the instrument in the same manner as turning the power off, then turning it back on. These functions are reset to:

Function	INIT Value
FREQ	0
DELFR	OFF
FRCAL	OFF
SPAN	MAX
ZEROSP	OFF
RESBW	1 MHZ
ARES	ON
PHSLK	OFF
VIDFLT	OFF
VRTDSP	LOG:10 DB
REFLVL	+30 dBm
FINE	OFF
RLMODE	MDIST
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
CLIP	OFF
EOS	OFF
RQS	ON

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

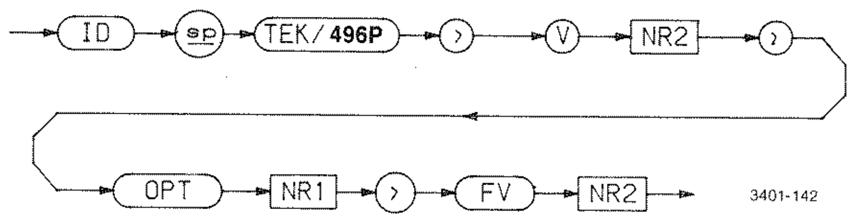
There is no INIT query.

ID (identify) query:



3401-141

Response to ID query:



3401-142

V<NR2>:	Tektronix Codes & Formats Standard version number.
OPT<NR1>:	Each digit represents an installed option.
FV<NR2>:	Instrument firmware version number.

MESSAGE EXECUTION

Two commands (WAIT and REPEAT) affect how the 496P microcomputer executes other message units in the same message.

WAIT (wait to execute) command:

3401-143

The 496P microcomputer waits to execute commands in its input buffer that follow WAIT. While waiting, the microcomputer sets busy in its GPIB status byte and does not input device-dependent messages. The wait is terminated in either of two ways:

1) an end-of-sweep occurs (if armed in single-sweep mode or sweep is in repetitive mode), or

2) DCL or SDC (while listener-addressed) is received.

Case 1 allows the controller to request updated spectrum data and be guaranteed the data has been updated. Here is such a message:

SIGSWP;SIGSWP;WAIT;WFMPRE?CURVE?

The first SIGSWP sets the analyzer to single-sweep mode if previously in repetitive-sweep mode. The next SIGSWP arms the sweep, and WAIT suspends further execution until the sweep completes. The message ends by requesting a waveform preamble and data. (The analyzer should be triggered or set to FRERUN.)

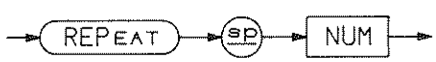
If the sweep is not armed in single-sweep mode when the microcomputer encounters WAIT, the microcomputer continues execution of the message in the buffer without waiting.

Case 2 results in flushing the input and output buffer so any commands following the WAIT are aborted. This also comes up where busy is discusssed under Status Byte.

Interaction: This command delays execution of any portion of a message that follows until the condition(s) just outlined occur(s).

There is no WAIT query.

REPEAT (repeat execution) command:



3401-144

NUM: The number of times the microcomputer is to repeat execution of commands or queries preceding REPEAT.

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

While REPEAT may be one of the commands preceding a REPEAT, the nested REPEAT is only executed on the first pass through the commands preceding the second REPEAT. For example:

RGTNXT;FREQ?REPEAT 10;FREQ 15 MHZ;REPEAT 1

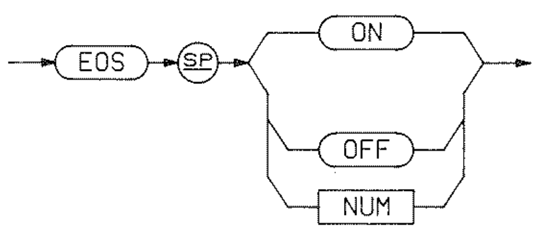
This causes the 496P to output 12 frequency values after it executes the message. The 496P does not output 22 frequency values because it only executes the frequency query once on its second pass through the entire message.

Interaction: A REPEAT loop can only be aborted by DCL and then only if the loop contains a WAIT command. Pressing RESET TO LOCAL does not abort the loop but only causes execution errors to be reported if the loop contains front-panel commands.

STATUS AND ERROR REPORTING

Two commands (EOS and RQS) control SRQs from the 496P. The status byte reports instrument status in a format that implements both IEEE 488 and the Tektronix Codes and Formats Standards. Two queries (ERCNT? and ERR?) provide more detailed information on errors related to abnormal status conditons.

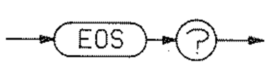
EOS (end-of-sweep) command:



3401-145

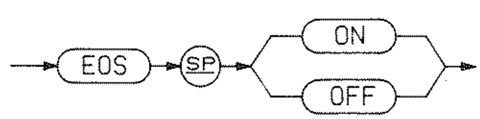
ON:	The analyzer asserts SRQ when a sweep completes.
OFF:	The analyzer does not assert SRQ for the EOS conditon.
NUM:	1 equals ON; 0 equals OFF. Other numbers are rounded to 1 or 0.
Power-up value:	Off.
Interaction:	Always OFF in talk- and listen-only modes.

EOS (end-of-sweep) query:



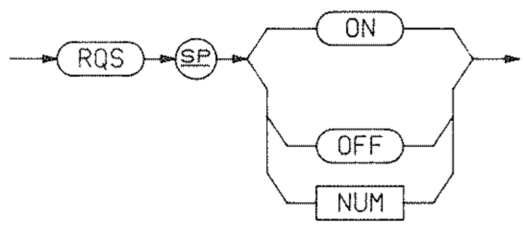
3401-146

Response to EOS query:



3401-147

RQS (request service) command:



3401-148

ON: SRQ is asserted when abnormal status conditions occur.

OFF: SRQ is not asserted (is masked) when abnormal status occurs. RQS OFF does not affect whether SRQ occurs at the end of a sweep—see EOS.

NUM: 1 equals ON; 0 equals OFF. Other numbers are rounded to 1 or 0.

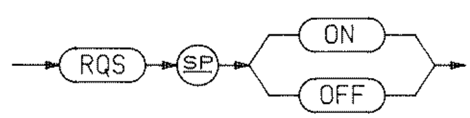
Power-up value: On.

Interaction: Always OFF in talk- and listen-only modes.

RQS (request service) query:

3401-149

Response to RQS query:



3401-150

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: 496P microcomputer busy condition
- Bit 6: Abnornal (1) normal (0) condition
- Bit 7: SRQ asserted (depends on RQS and EOS commands)

Power-on status:	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:	Set when the 496P completes a sweep of the selected spectrum; this indicates digital storage has been updated.
Ordinary operation status:	Exists whenever there is nothing out of the ordinary (no other status condition) to report.
Command error:	Occurs when a message cannot be parsed or recognized.
Execution error:	Results when a message is parsed and is recognized, but cannot be executed such as FREQ [−]1 MHZ.
Internal error:	Indicates the 496P microcomputer has discovered a malfunction that could cause the instrument to operate incorrectly.

Execution error warning:	Results from a command that the 496P executes, but has the potential for error. An example is RESBW 10 KHZ in max span mode. The 496P sets the warning status because the UNCAL indicator is lit.
Internal error warning:	Reports that a non-fatal operating condition has been detected by the 496P microcomputer.
Busy:	Reported whenever the 496P microcomputer is executing a message in its input buffer. This includes the WAIT command; while waiting, the microcomputer reports busy status.

Effect of Busy on Device-Dependent Messages

The microcomputer does not accept further device-dependent messages while busy; 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). Such commands are 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 the following discussion.) 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.

Effect of Busy on Interface Messages

Interface messages (and the rtl message from the RESET TO LOCAL button), however, are processed despite busy status, if the busy status occurs because the microcomputer is executing a WAIT command. If RESET TO LOCAL interrupts execution of WAIT, the microcomputer attempts to execute the remainder of the message after restoring local control and waiting for EOS. Front-panel commands result in error SRQs because these commands conflict with local control.

If the busy status occurs because the microcomputer is executing any device-dependent message other than WAIT, then the response of the 496P to interface messages depends on the manner in which they are handled. Some are handled by the GPIB interface, while others require action by the microcomputer. The former are not affected by busy, while the latter are not executed until the microcomputer is unbusy. The latter generally involve the 496P GPIB address, so are implemented in microcomputer firmware rather than on the interface.

The following considerations apply to interface messages received while the 496P microcomputer is executing a device-dependent message other than WAIT:

1) universal commands LLO, SPE, and SPD are handshaked and acted on by the interface, so are unaffected by busy. 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 listener function, if active. Addresses that do not match those set by the power-up value of the rear-panel switches are handshaked and discarded by the interface. However, 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 (is not busy);

Because the microcomputer gets involved when a current address is received, addressed commands are impacted by busy (except while WAITing). Serial poll is similarly affected if MTA preceded SPE.

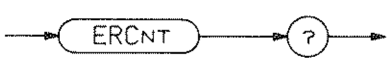
3) GTL is handshaked immediately by the interface. If the 496P is already listener-addressed, the microcomputer executes GTL after it finishes executing any message in its buffer (except WAIT or message units following WAIT);

4) DCL requires microcomputer action, holding up the handshake if the microcomputer is busy. If addressed, the microcomputer treats SDC in the same way. These two device-clear messages are executed as soon as they are accepted;

5) GET also requires action by the microcomputer, so its handshake also occurs only when the microcomputer is not executing a device-dependent message unit other than WAIT. As with DCL and SDC, GET is executed immediately, aborting the current sweep and rearming the sweep;

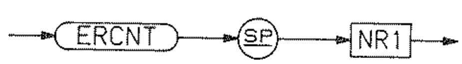
6) parallel polls are handled by the microcomputer, so PPC, PPE, PPD, and PPU must wait for not busy to be executed (assuming the instrument was addressed to start the sequence).

ERCNT (error count) query:



3401-151

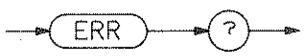
Response to ERCNT query:



3401-152

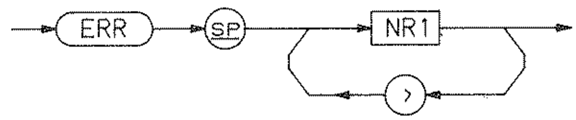
NR1: The number of error codes to be returned for an ERR query.

ERR (error) query:



3401-153

Response to ERR query:



3401-154

Command Errors

The following command errors are defined:

Code	Meaning
0	No error
1	Number error
2	Invalid character in block ISO count
3	EOI in block ISO
4	EOI in block binary
5	Checksum error in block binary
6	Illegal placement of question mark
7	Invalid query
8	Invalid header
9	Invalid end
10	Invalid character argument (not allowed)
11	Invalid number argument " "
12	Invalid string argument " "
13	Invalid binary argument " "
14	Link not allowed
15	Invalid link label
16	Empty link label
17	Invalid character value (not allowed)
18	Invalid number value " "
19	Invalid string value " "
20	Invalid binary value " "
21	Link argument not allowed as link value
22	Character not found
23	Invalid suffix
24	Input buffer overflow

Execution Errors

The following execution errors are defined:

Code	Meaning
26	Output buffer overflow
27	Attempt to execute in local mode
28	FREQ or TUNE beyond range
30	FRCAL out of range
31	SPAN not available
32	RESBW not available
33	Minimum atten (MINATT/MAXPWR) out of range
34	REFLVL out of range
35	VRTDSP out of range (LIN argument)
36	VRTDSP out of range (LOG argument)
37	TIME out of range
38	DEGAUS not allowed in present span/div
40	FIBIG, LFTNXT, or RGTNXT not allowed in present span/div
41	ADDR/DATA argument invalid
42	ADDR not compatible for DATA command
43	CRVID or WFID not valid
44	WFMPRE not compatible with 496P

Execution Warnings

The following execution warnings are defined:

Code	Meaning
50	SPAN defaulted to MAX
51	SPAN defaulted to 0
52	UNCAL light on
53	Multiple use of display buffer

Internal Errors

The following internal errors are defined (and displayed on the crt, as well):

Code	Meaning
57	TUNE carry from lower DAC failed
58	Phaselock failed
59	Lost phaselock
60	Failed to recenter when phaselock turned off or non-phaselock span selected.

Error codes are returned in numerical order. Reading the current codes(s) clears the error response.

HELPS AND HINTS

This section elaborates on some techniques for programming the 496P, using 4050-Series BASIC for examples. We hope the treatment of these topics will speed your progress in putting the 496P to work solving your measurement problems.

NOTE

In the examples throughout this section, the 496P primary address is assumed to be 1. See Section 1 for how to set the GPIB ADDRESS switches.

DATA ACQUISITION

When the 496P is acquiring spectrum data under program control, two programs are running, not just one. One program is running in the controller, and another in the 496P. The key to success is synchronizing 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 496P

Program execution in the controller can be synchronized with 496P execution of messages it receives over the GPIB. The 496P solves the problem by the way it buffers and executes messages.

When the 496P receives a message, it waits until the end of the message to begin execution. While busy executing the message, the 496P does not accept any other device-dependent messages. When it is finished executing a message, the 496P is ready to handshake another message, which it then executes, and so on. You can depend on the 496P to assert NRFD on the GPIB while it is busy, holding up execution of a controller GPIB output statement that sends further insructions to the 496P. For example:

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

Watch the 496P 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 496P to execute the previous FREQ command.

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 496P. Generally, WAIT is placed immediately after a SIGSWP command that arms a sweep so that data is acquired from a full sweep. WAIT delays execution of commands or queries that follow in the same message until the end of the sweep. This means you can direct the 496P to acquire, process, and output data, all in the same message. If the commands or queries you add for processing or output follow the WAIT, you obtain results based on data acquired by the SIGSWP command. For example:

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

Line 110 sets the 496P to single-sweep mode (if the 496P is not already in single-sweep mode). Succeeding SIGSWP commands arm the sweep.

Line 130 illustrates how to use WAIT: it follows SIGSWP and precedes the command and query that ready the 496P to output updated data. The 496P does not handshake out the data in line 140 until it finishes executing the message in line 130.

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 biggest value in digital storage. The Y variable is lost each time through the loop when the 496P receives further input before it can handshake out the second POINT argument.

Fig. 8-1 further illustrates the concept of two programs—one in the controller and one in the 496P—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.

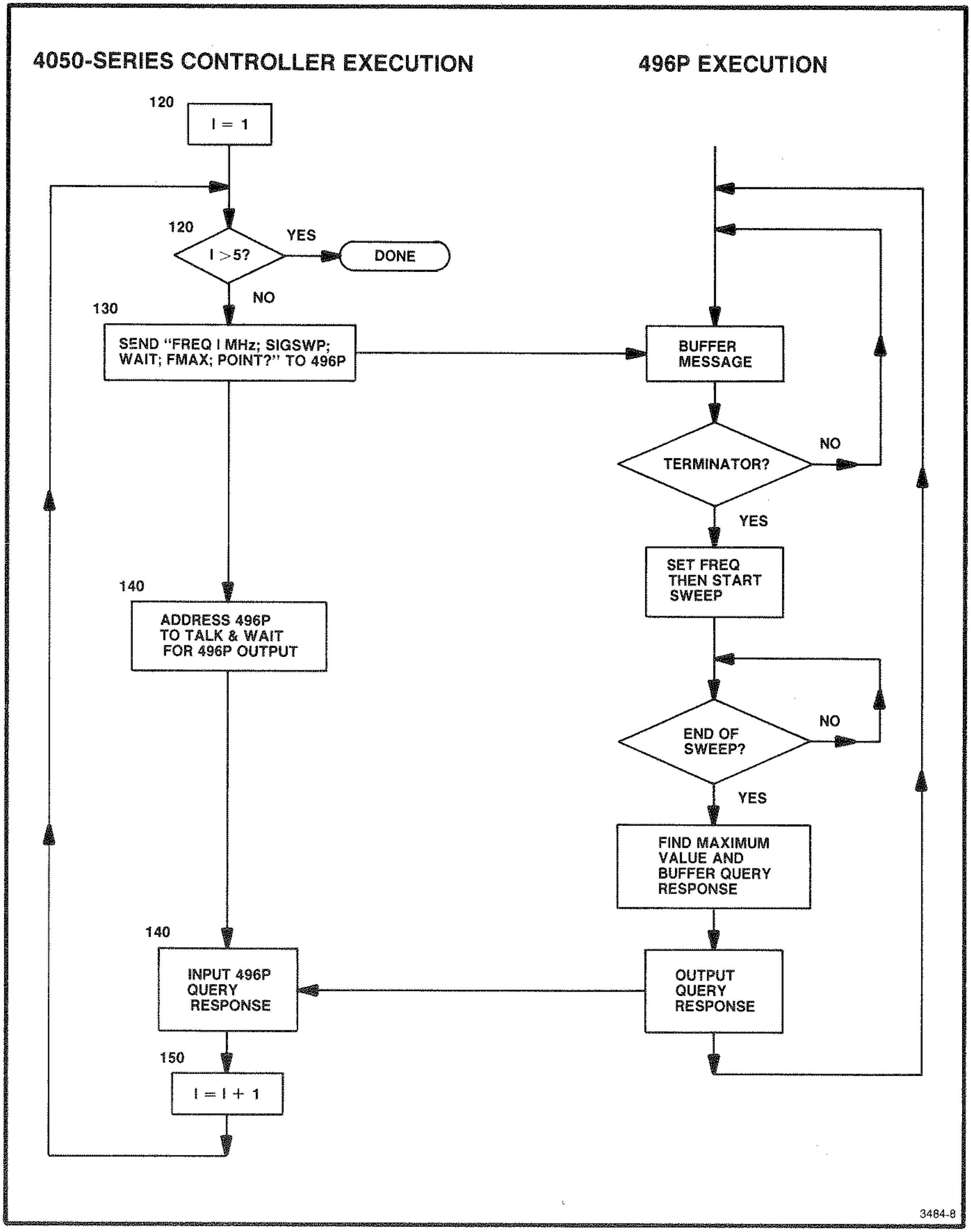


Fig. 8-1. Synchronizing controller and 496P for data acquisition.

Using the End-of-Sweep SRQ

Although the previous method for synchronizing controller/496P 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 496P is made a talker or that must take care of other tasks while the 496P is acquiring data. In such cases, the controller can enable the 496P end-of-sweep SRQ to synchronize data acquisition with the sweep. The following example just shifts the WAIT from the 496P 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 @1:"SIGSWP;EOS ON"
130 FOR I=1 TO 5
140 PRINT @1:"FREQ ";I;"MHZ;SIGSWP"
150 WAIT
160 PRINT @1:"FMAX;POINT?"
170 INPUT @1:P(I)
180 NEXT I


200 POLL A,B;1
210 RETURN
```

With no WAIT command following SIGSWP in line 140, the 496P 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 496P 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 496P 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 @1:"SIGSWP"
120 FOR I=1 TO 5
130 PRINT @2:"FREQ ";I;"MHZ"
140 PRINT @1:"FREQ ";I;"MHZ;SIGSWP;WAIT"
150 INPUT @1:D$
160 PRINT @1:"FMAX;POINT?"
170 INPUT @1:P(I)
180 NEXT I
```

Here we put the WAIT back into the 496P message and let the INPUT statement in line 150 stall the controller while the 496P takes a sweep. D$ serves the purpose of a dummy string; the data is not input until line 170.

BINARY WAVEFORM TRANSFERS

Selecting binary, rather than ASCII-coded decimal, speeds up waveform transfers. Neither the controller nor the 496P 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 496P Binary CURVE Output

The 496P encloses binary waveform data values in some other items in the block binary format. For details, see the syntax diagrams in Section 5. For a 4050-Series routine that handles block binary, try the following:

```
500 REM GET 496P BINARY CURVE OUTPUT
510 DIM W(100)
520 PRINT @37,0:37,255,255
530 PRINT @1:"WFMPRE ENC:BIN;CURVE?"
540 INPUT %1: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 496P response (CURVE CRVID:FULL,) and stops at the block binary delimiter (%), which is discarded.

Line 550 makes the 496P a talker.

Line 560 first inputs the first 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 checksum 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 checksum byte as an error check—the two added together (disregarding the carry) should equal zero.

Line 570 untalks the 496P.

Sending a Binary CURVE to the 496P

The following routine employs end block binary format to transfer a waveform to the 496P. Array W is transferred—if not already created by the routine above, W should be dimensioned to 1000 and filled with data in the range 0 to 255.

```
600 REMARK SEND BINARY CURVE TO 496P
610 WBYTE @33:64,W,-255
620 WBYTE @63:
```

Line 610 sends the 496P 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 496P to act on the message. The CURVE header is omitted; it is not required (but would be accepted if sent). The 496P 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 waveform data output by the 496P are 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 496P binary output program given above. This version transfers whatever portion of memory you have previously specified with a WFMPRE command: A, B, or FULL (power-up default is FULL). The program scales both the X and Y values and stores them in a two-wide array. The program also saves the nscaled binary array so you can transfer it back into 496P digital storage, if you wish.

```
500 REMARK GET AND SCALE 496P BINARY CURVE OUTPUT
510 DELETE W,M
520 PRINT @1:"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
```

Lines 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 496P a talker and inputs the 496P 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 through 580 talk the 496P, input the elements in the block binary format, and untalk the 496P.

Lines 590 through 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
```

Waveform Plotting Made Easy

A simple routine plots array W, the integers output by the 496P. The plot is embellished by labels derived from the waveform preamble obtained by the program above.

```
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"
```

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 through 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; 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 496P crt. If there are data in digital storage from these areas outside the 496P display, the data are 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):

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

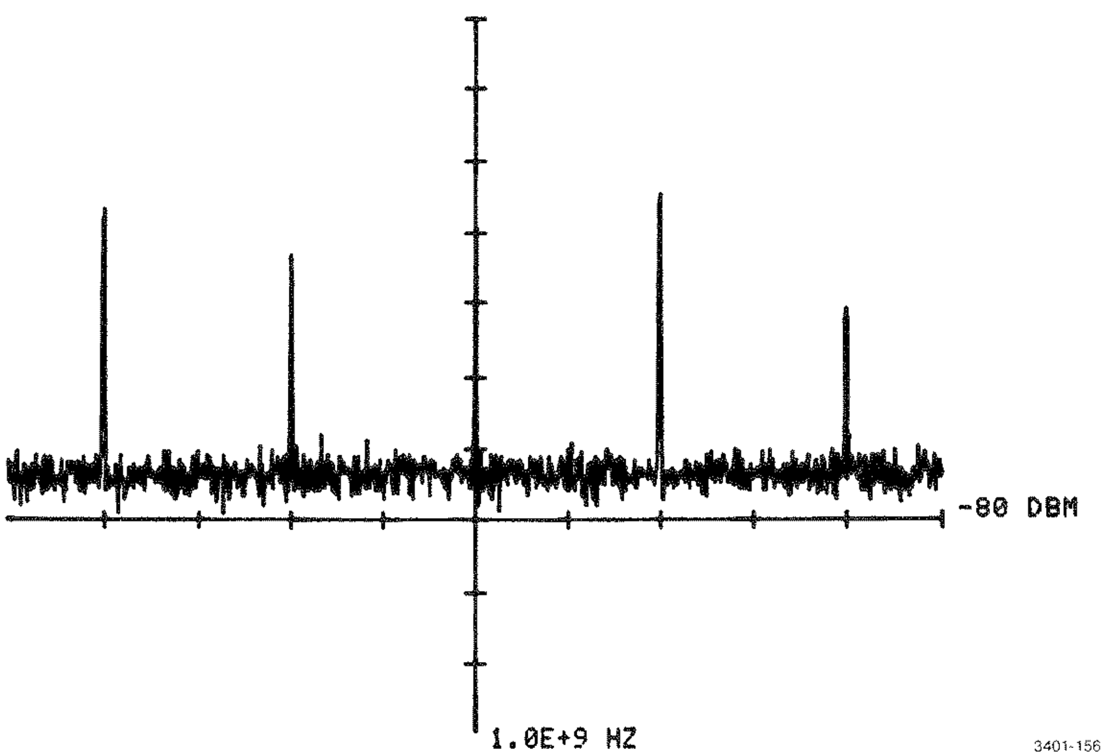


Fig. 8-2. A simple plot of a spectrum acquired from the 496P.

@

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 496P message. (This occurs in Section 5 in connection with CURVE and in Section 6 in connection with the display data point commands). So let's look into this topic further.

There is no conflict in many cases because the 496P buffers the message you send and then executes it in the order you sent it. For example, you can use the 496P as a waveform processor for spectrum data you previously acquired in array A:

```
100 REMARK BUFFER DEMO
110 PRINT @1:"CURVE ";A,";FIBIG;POINT?"
120 INPUT @1:B1,B2
```

In this case, the 496P does what you ask: it loads a waveform into digital storage and returns the point at the peak of the biggest signal.

However, interaction is possible in other cases because there is only one display data buffer used for both display input and output and as workspace for waveform processing. 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 Fig. 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 argument.

Order-Dependent Conflicts

Conflicts in the use of the buffer occur depending on the order in which waveform processing and i/o occurs. The CURVE query and display data point commands, by contrast, 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 496P 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.

Let's try an example combining CURVE output and input in the same message. This is a way to talk to yourself—if you can hold up both ends of the conversation.

```
100 REMARK WRONG WAY
110 DIM B(1000)
120 PRINT @1:"CURVE?CURVE ";A;
130 INPUT @1:B
```

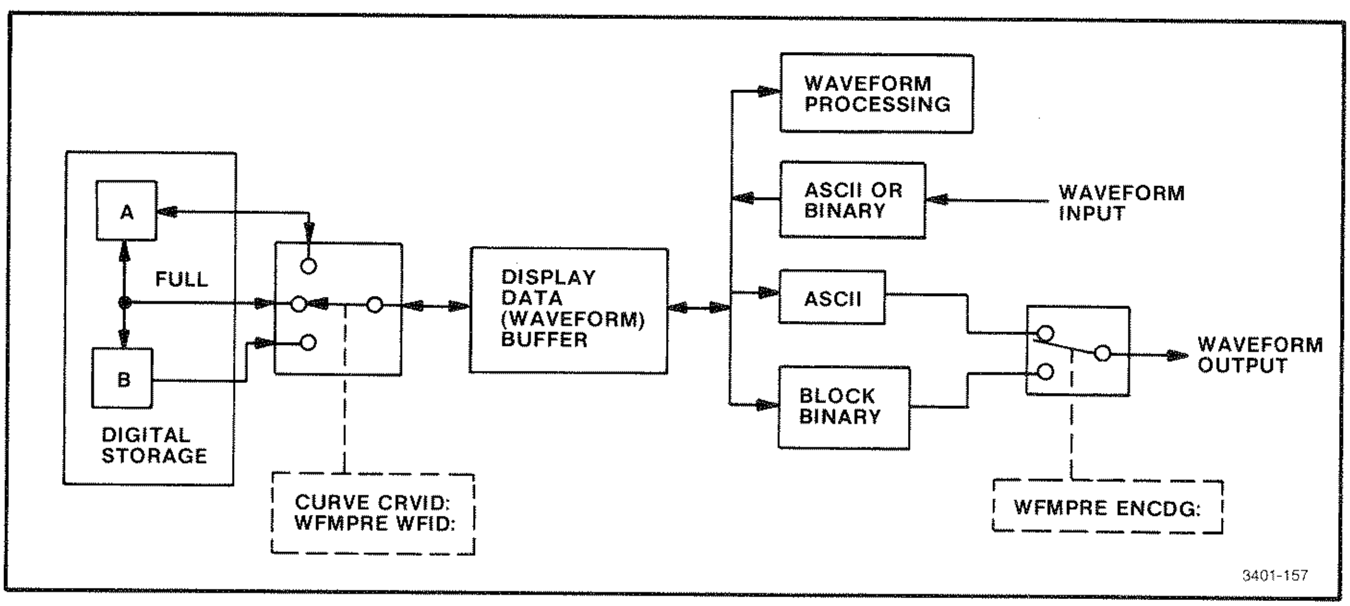


Fig. 8-3. How multiple use of the display data buffer is controlled.

What this program attempts is to obtain a 496P waveform and replace it with a waveform residing in controller array A. But that's not what happens. The 496P does buffer the CURVE data transmitted from array A by line 120 but then the 496P 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 496P 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 are lost and do not replace the original digital storage waveform.

Instead of the above, try:

```
100 REMARK RIGHT WAY
110 DIM B(1000)
120 PRINT @1:"CURVE?"
130 INPUT @1:B
140 PRINT @1:"CURVE ";A;
```

Line 120 requests a curve, which the 496P 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 496P accepts the space (or other format characters), as well as a comma, for a delimiter.

FINDING SIGNALS WITH 496P WAVEFORM PROCESSING

The waveform processing resident in the 496P 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 is intended to help you gain that understanding and suggest how to apply 496P 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 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 such 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 other 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 reading 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 point. The commands that do this, CENSIG and TOPSIG, rely on the span/div and vertical display scale factors when computing how far to change the center frequency and the reference level, respectively. The accuracy of the scale factors, along with the accuracy of TUNE and REFLVL, respectively, 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 factors mentioned above, resolution and noise, 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 use 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 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 a level that is not too high *and* not too low? There is no such level that works in all cases. One 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 approach 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 LFTRNXT 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 RFTNXT.

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 496P loop through a message that uses RGTNXT to search a spectrum is presented in connection with REPEAT, our next topic.

USING REPEAT FOR SIGNAL TRACKING AND SEARCHES

The REPEAT command adds another dimension to 496P messages—the number of times to loop through a string of commands or queries. Several uses are suggested here.

Tracking a Signal

The 496P can track a signal, keeping it centered on the display. To do this, the 496P updates its frequency while executing a message that uses waveform processing to find the peak value after each sweep and center it on the display. The REPEAT command causes the 496P to execute the message repeatedly:

```
100 REMARK TRACK A SIGNAL
110 PRINT @1:"WAIT;FMAX;CENSIG;SIGSWP;REPEAT 10E6"
```

The message in line 110 causes the microcomputer to wait until the sweep completes to process the digital storage waveform, then 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 because the message includes a WAIT command. Even though the microcomputer is busy executing the loop, it recognizes DCL during a wait, 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 in the loop, if any, causing execution error status to be reported.

REPEAT can also be applied to a similar task: keeping a signal centered as span/div 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 @1:"SIGSWP"
120 PRI@1:"CEN,SPA DEC;SIG;WAI;POI 400;RGT 125;REP 5"
```

Line 110 guarantees the analyzer is in single-sweep mode. This routine assumes the signal is already identified by the display data point (as a result of FIBIG, RGTNX, 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 496P can perform a signal search by executing a loop in a single message, buffering the results without controller interaction. The controller can later turn its attention again to the 496P and input the results.

```
100 REMARK SPECTRUM SEARCH
110 PRINT @1:"POINT 0"
120 PRINT @1:"RGTNXT;POINT?REPEAT 20"
130 REMARK****************
140 REMARK INSERT ANY OTHER TASKS
150 REMARK*****************
300 DIM P(20,2)
310 INPUT @1:P
```

This routine works on a waveform in digital storage that is not updated during processing. In line 120, the 496P 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: the buffer can handle 15 responses to the FREQ query, which includes the frequency in scientific notation. But the buffer can handle 43 of the shorter PHSLK query responses (PHSLK ON or PHSLK OFF).

MESSAGES ON THE CRT USING RDOUT

The 496P accepts either single- or double-quote marks to delimit the crt message. With 4050-Series controllers, use single-quotes around the message inside the RDOUT command:

```
100 PRINT @1:"RDOUT 'SET THE PEAK/AVERAGE KNOB'"
```

This tactic is necessary because the 4050-Series controller uses double-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 496P crt:

```
100 PRINT @1:"RDOUT 'PRESS ""RETURN TO LOCAL""'"
```

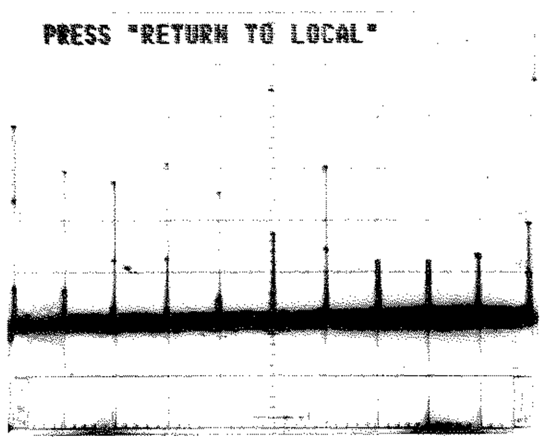
The controller strips off the first in each pair of twin double-quote marks and transmits the second of each pair for display as shown in Fig. 8-4.

The RDOUT message continues to be displayed if the 496P remains under remote control. To demonstrate the above messages by themselves, add the following statement:

```
110 GOTO 110
```

To scroll the RDOUT message to the top of the 496P screen, insert this line:

```
105 PRINT @1:"RDOUT"
```



3401-158

Fig. 8-4. Quote marks can be used in messages on the 496P crt.

WHAT IS THE DIFFERENCE BETWEEN FREQ AND TUNE?

The FREQ command argument is in absolute frequency; the TUNE command argument is in relative frequency. Use FREQ to change center frequency to some value you supply as a command argument; use TUNE to change the center frequency by some offset you supply as a command argument.

While the specified accuracy using FREQ is the same as the frequency readout and applies to the absolute frequency, the accuracy using TUNE is specified in terms of the frequency change. The difference in the specified accuracies is in TUNE's favor—for relative changes in frequency use TUNE rather than asking for the new frequency as an absolute value using FREQ.

The ranges of the FREQ and TUNE commands differ in several respects. The range of the FREQ command covers the entire range of the instrument. The TUNE command is limited to the current frequency range when the 1st LO is being tuned. In addition, in narrow spans TUNE changes the 2nd LO, rather than the 1st LO, so is limited to the range of the 2nd LO. The TUNE range in narrow spans (≤50 kHz) is 3.1 MHz.

The advantage in accuracy obtained with TUNE is especially helpful in narrow spans. If the frequency change is beyond the range of TUNE, you can use FREQ to change the 1st LO frequency, even in narrow spans. This differs from front-panel operation, in that the FREQUENCY knob can only vary the 2nd LO in narrow spans. However, it is easier to locate the signal of interest if you widen the frequency window (SPAN) before sending the new center frequency using FREQ. After the frequency change, you can span back down.

If you choose to leave the analyzer in a narrow span when using the FREQ command and the 1st LO is not phaselocked, you may notice that the analyzer frequency settles slowly. This is caused by a large capacitor that makes the 1st LO driver output very quiet in narrow spans when phaselock is not active. This capacitor is switched out if the analyzer is not in a narrow span or if phaselock is active in a narrow span.

POWER-UP BUS GLITCHES

When the 496P powers up, the GPIB drivers in the interface pulse the GPIB lines. Because this may cause problems, depending on what is occurring on the bus, we recommend that you apply power to the 496P before running the system operating program or that you disconnect the 496P from the bus before applying power.

COMPARING THE STATUS BYTE AND ERR? RESPONSE

The 496P status byte and ERR ? described in Section 7 play complementary roles in GPIB system programming. The status byte is the 496P response to a serial poll. The ERR? response is the 496P answer to a device-dependent query message. The status byte provides information about instrument conditions by category: normal/abnormal, busy, command error, execution error, etc. The ERR ? response details the cause of abnormal status, i.e., what kind of error or warning prompted the 496P to assert SRQ and report abnormal status.

Status bytes are not stacked: the code for the condition that caused the SRQ is not updated, although bit 5 reflects the present instrument state—1 for busy, 0 for not busy. Error codes, however, are accumulated until read and are reported in numerical order. While you can recover only one status byte, you may recover more than one code in the ERR ? response, indicating more than one abnormal condition occurred.

The status byte is cleared by a serial poll of the instrument. Error codes are also cleared by reading them with the ERR query. Reading the status byte does not clear the error codes, and vice versa. DCL and SDC (if addressed) clear both the status byte and error codes.

EXECUTION AND TRANSFER TIMES

The 496P microcomputer system takes typically 10 to 25 milliseconds to execute commands received over the bus. This it the time the 496P 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 milliseconds, 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	
Spans >5 MHz/div	7.7 ms + 0.04 ms/MHz change in 1st LO
Spans ≤5 MHz/div	325 ms + 0.08 ms/MHz change in 1st LO
TUNE	
Narrow spans	7.7 ms + 0.04 ms/kHz change in 2nd LO
Other Spans	Same as FREQ
DEGAUS	800 milliseconds
REFLVL, RLMODE, MINATT, MAXPWR	Add 100 milliseconds if RF attenuator is switched
PHSLK or SPAN commands that operate phaselock function	500 milliseconds
Setup with SET? response	120—3100 milliseconds
INIT	1500—2500 milliseconds
POINT (X argument only)	50 milliseconds
FIBIG	1.5 s
LFTNXT, RFTNXT	(150 milliseconds) * (signal separation in divisions)
FMAX, FMIN	75 milliseconds
CURVE	175 milliseconds
CURVE?	50 milliseconds

Because of the way the 496P handles output, the microcomputer is free after loading an output buffer. The additional time for the transfer is related to the listener for cases where the 496P is faster. For instance, with 4050-Series controllers, the transfer times in Table 8-2 have been observed. Table 8-2 also lists transfer times observed for controller output to the 496P using the CURVE command.

Table 8-2
TRANSFER TIMES

Transfer	4051	4052
SET? response	200 ms	200 ms
CURVE? response:		
Binary (input as numbers)	1.5 s	350 ms
ASCII (input as string)	8.5 s	8.5 s
ASCII (input as numbers)	12 s	9 s
CURVE (binary from number array)	1 s	600 ms
CURVE (as ASCII string)	5.5 s	5 s
CURVE (as ASCII numbers)	14 s	7.5 s

APPENDIX A
IEEE 488 (GPIB) SYSTEM CONCEPTS

INTRODUCTION

The GPIB (General Purpose Interface Bus) is a digital control bus that allows efficient communications between 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 488-1978, "Standard Digital Interface for Programmable Instrumentation." At Tektronix, the IEEE 488 digital interface is commonly known as the General Purpose Interface Bus (GPIB). This part of the manual discusses the basic concepts of the GPIB. For complete specifications, refer to the IEEE 488-1978 standard. The standard is published by the Institute of Electrical and Electronics Engineers, Inc., 345 East 47th Street, New York, New York 10017.

There are four elements of the GPIB: 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. These reactions are described in the programming section of this manual.

MECHANICAL ELEMENTS

The IEEE 488 standard defines the cables and connectors as the mechanical elements of the instrumentation system. Standardizing the connectors and cables ensure 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 Fig. A-1.

The cable atached 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. Interconnecting cable assemblies are 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 counter-bored captive screws.

ELECTRICAL ELEMENTS

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

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

- Logical 0 is a false state, high voltage level ($\geqslant$ +2.0 V), signal line is not asserted.

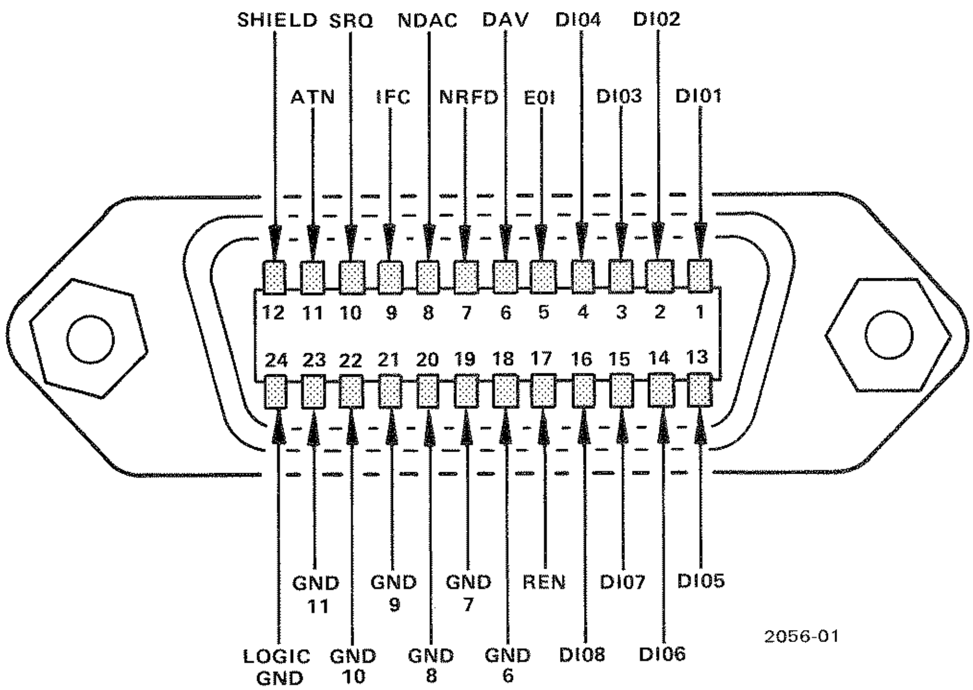


Fig. A-1. Standard IEEE 488 (GPIB) connector.

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

FUNCTIONAL ELEMENTS

The functional elements of the GPIB cover three areas:

- The ten major interface functions 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.
- The specific protocol by which the interface functions send and receive their limited set of messages.
- The logical and timing relationships between allowable states for all interface functions.

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 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 Fig. A-2, along with the nomenclature for the sixteen active signal lines. Only four instruments are shown connected directly to the control bus. However, more than 15 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 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. At least one more than half of the instruments connected directly to the bus must be in the power-on state.

TALKERS, LISTENERS, AND CONTROLLERS

A talker is an instrument that can send messages and data over the bus; a listener is an instrument that can accept messages and data from the bus. An instrument can be a talker only, listener only, or be both a talker and a listener. No instrument can communicate with any other instrument on the bus until it is enabled to do so by the controller-in-change 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 listener whenever the program routine requires it. In addition to designating the current talkers and listeners for a particular communication sequence, the controller is assigned the task of sending special codes and commands (called interface control mesages) to any or all instruments on the bus.

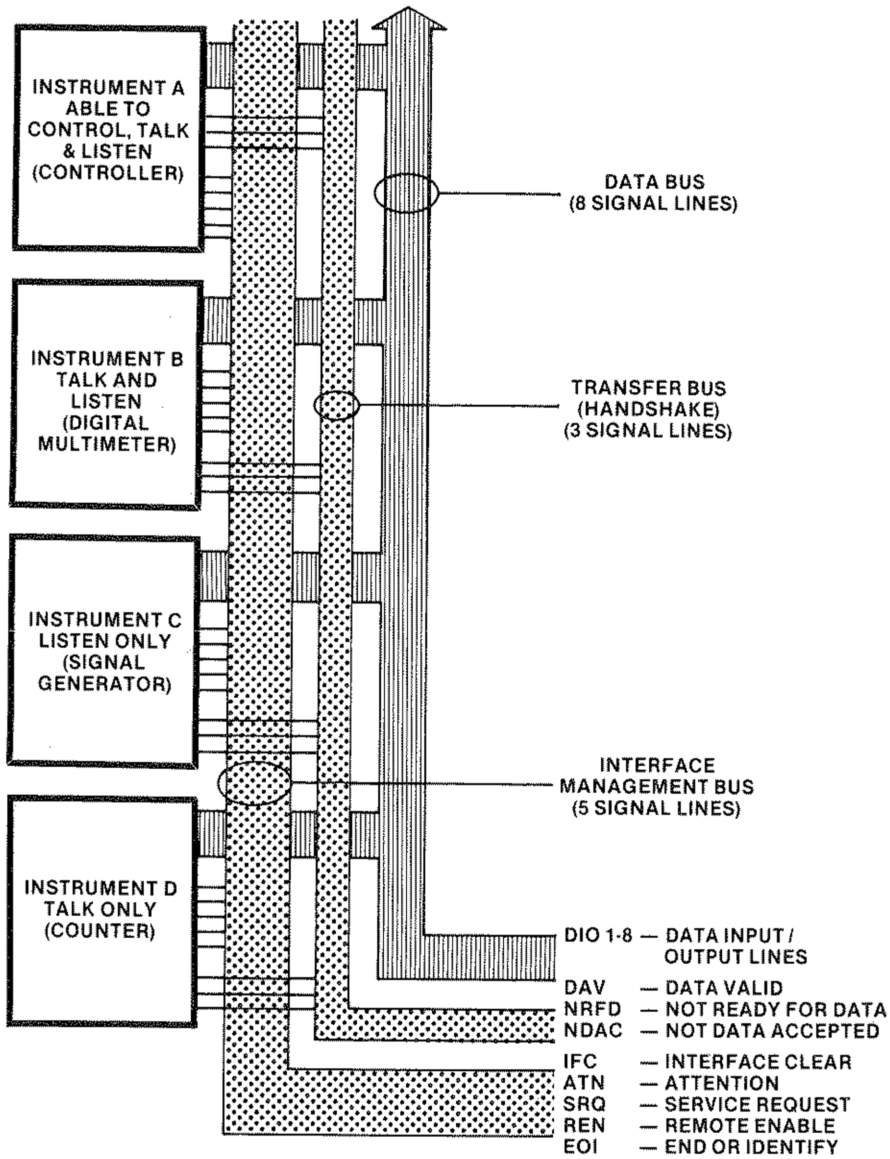


Fig. A-2. A typical GPIB system.

More than one controller may be in a complete operating system. The IEEE 488 standard has provisions for a system controller operating with a controller-in-charge of the bus. The controller-in-charge can take control of the bus only when directed to do so by the system controller. The system controller may, but not necessarily, be the controller-in-charge.

INTERFACE CONTROL MESSAGES

Interface control messages are of two types: multiline messages sent over the data bus and uniline messages. Uniline messages are discussed under GPIB Signal Line Definitions.

The interface control messages sent over the data bus with the ATN line asserted are shown in Fig. A-3. The user can correlate interface message coding to the ISO 7-bit code in Fig. A-3 by relating data bus lines D101 through D107 to bits B1 through B7, respectively.

Interface control messages include the primary talk and listen addresses for instruments on the bus, addressed commands (only instruments previously addressed respond to these commands), universal commands (all instruments, whether they have been addressed or not, respond to these commands), secondary addresses for devices interfaced through their primary instrument, and secondary commands. At present, the IEEE 488 standard classifies only two interface messages as secondary commands, Parallel Poll Enable (PPE) and Parallel Poll Disable (PPD). PPE messages are derived from the characters in the first column under Lower Case letters in Fig. A-3 (decimal coded characters 96 through 111). The standard recommends the use of decimal code 112—lower case letter "p"—for the PPD command. All parallel poll configured instruments respond with status information at the same time when the EOI line is asserted with ATN true.

DEVICE-DEPENDENT MESSAGES

The IEEE 488 standard does not specify coding of device-dependent messages (messages that control the device's internal operating functions). After addressing a talker and the required number of listeners via interface control messages, the controller unasserts the ATN line on the bus. When ATN becomes false (high), any commonly understood 8-bit binary code may be used to represent a device-dependent message.

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

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

The ASCII character string is sent with the ATN line unasserted and 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-dpendent 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.

TYPICAL GPIB EVENTS WHEN TRANSFERRING DATA

Figure A-2 shows the 16 active signal lines on the GPIB to be functionally divided into three component busses: an eight-line data bus, a three-line data transfer control (handshake) bus, and a five-line management bus.

ASCII & IEEE 488 (GPIB) CODE CHART

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

Interface messages are sent with ATN asserted.

KEY

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

Fig. A-3. ASCII & IEEE 488 (GPIB) Code Chart.

The data bus has eight bidirectional signal lines, DI01 through DI08. Information, in the form of data bytes, is transferred over this bus. A handshake timing sequence between an 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 at any one time. A talker cannot place data bytes on the bus faster than any one 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.

After EOI, the controller asserts ATN 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.

Figure A-5 illustrates a typical handshake timing sequence on the three-line data transfer control bus. The ATN line is also shown to illustrate the controller's role in the process.

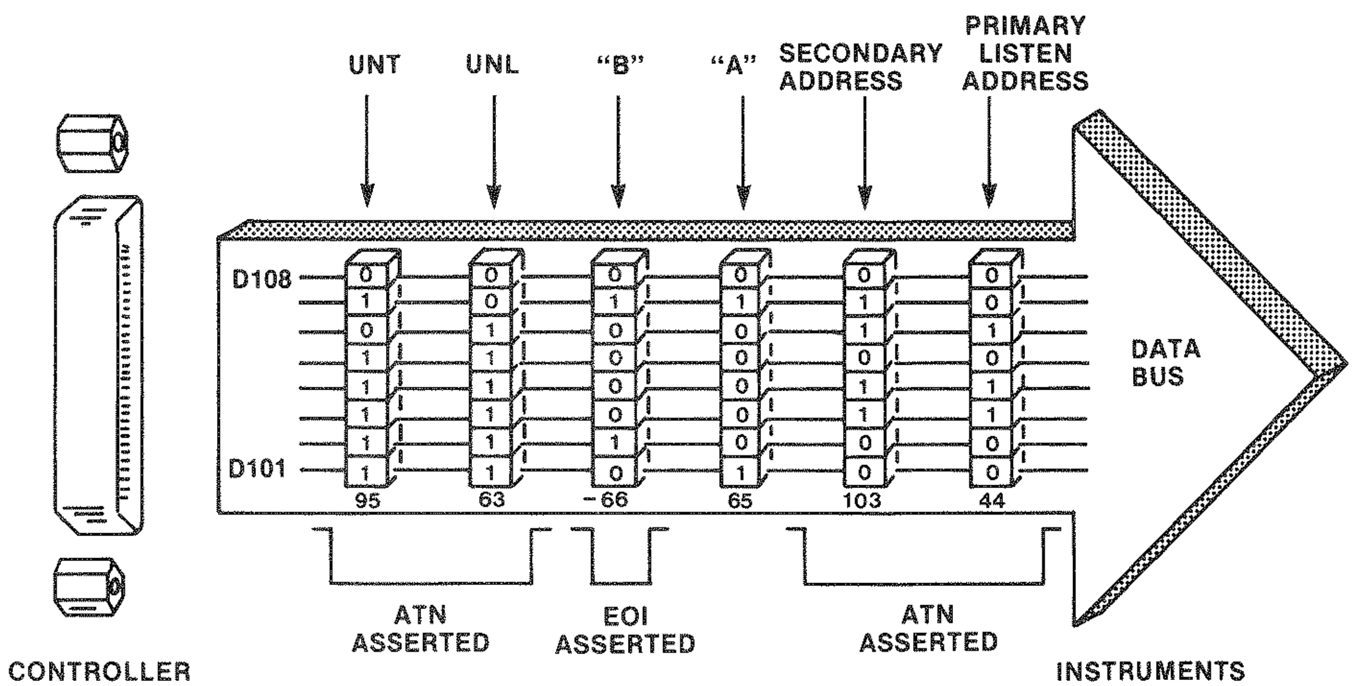


Fig. A-4. An example of data byte traffic on the GPIB.

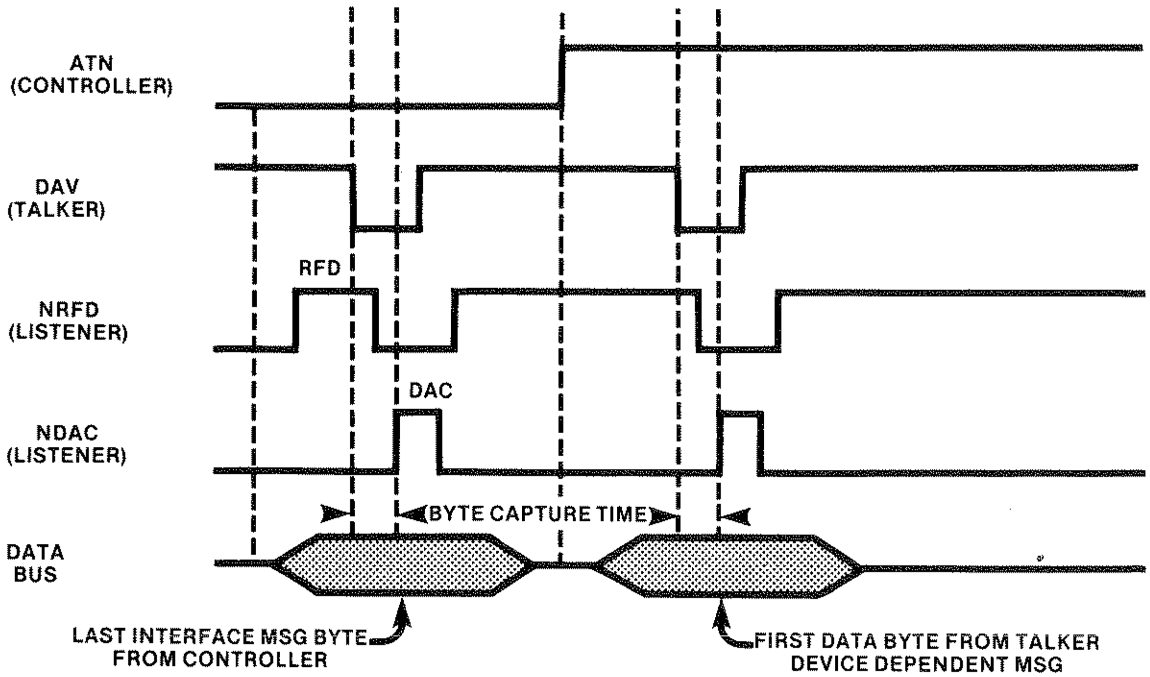


Fig. A-5. A typical handshake timing sequence (idealized). Byte capture time is dependent on the slowest instrument involved in the handshake. RFD means Ready For Data; DAC means Data Accepted.

GPIB SIGNAL LINE DEFINITIONS

Data Transfer Control Bus

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 Fig. A-5. These signal lines are more completely defined as follows:

NRFD (Not Ready For Data). An asserted low true 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.

DAV (Dava Valid). The DAV signal line is asserted low true by the talker after the talker places a data byte on the data bus. When asserted, 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 asseted.

NDAC (Not Data Accepted). Each assigned listener holds the NDAC signal line low true (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

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

Management Bus

The management bus is in a group of five signal lines that are used to control the operation of the IEE 488 (GPIB) Digital Interface. These signal lines are more completely defined as follows:

IFC (Interface Clear). The system controller asserts the IFC signal line for greater than 100 microseconds to place all instruments in a predetermined state. In most instruments the IFC messsage has the same effect as the untalk (UNT) and unlisten (UNL) commands. The system controller is the only instrument on the bus allowed to assert IFC. While IFC is being sent, only the DCL, LLO, PPU, and REN (Remote Enable) universal commands will be recognized.

ATN (Attention). A controller assets the ATN signal line 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, 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. Only the controller-in-charge is allowed to generate the ATN message.

SRQ (Service Request). Any instrument connected to the bus can request the controller's attention by asserting the SRQ line. The controller may respond by asserting ATN and executing a serial poll routine to determine which instrument is requesting service. An instrument requesting service via the SRQ line responds with a device-dependent status byte with bit seven asserted. When the polling routine is completed, program control returns to the main program. When polled, the instrument requesting service unasserts the SRQ line. A controller-in-charge does not have to see the SRQ line asserted to perform a polling routine, it may do so whenever the program requires it.

REN (Remote Enable). The system controller asserts the REN signal line whenever it is required to operate the GPIB under remove program control. Used with other interface control messages, such as LLO or GTL, the REN signal causes an instrument to select between two alternate sources of programming information: Programming data from the bus (Remote) or 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 latest 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 may assume 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 is a discussion of the interface functions and their relationship to interface control messages and commands. The multiline interface control messages discussed herein are found in Fig. A-3 and are sent over the GPIB with the ATN line asserted.

REMOTE-LOCAL FUNCTION (RL)

The RL function provides an instrument with the capability to select between two sources of information input. 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 line, whether or not it is the controller-in-charge.

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, not before. An instrument remains in a remote mode until the REN line is released (false state), 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 command does not return an instrument to the local mode.

All instruments on the bus must recognize when the REN line goes false and go to the local mode within 100 ms. 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.

@

TALKER AND LISTENER FUNCTIONS (T/TE AND L/LE)

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 of sending device-dependent data (or, in case of a controller, the capability to send multiline interface control messages or 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. The second byte is called MSA (My Secondary Address).

Only one instrument in the GPIB system can be in the talker active 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 (up to a maximum of 14) can be programmed for the L (Listener) function by the 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 also requires a two-byte address code. All talker and listener functions myst respond to ATN within 200 ns. They must also respond to IFC or REN false in less than 100 ms.

An instrument may have a "talker only" function, a "listener only" function, or have all functions implemented (T, L, TE, and LE). 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 generally made equal to the binary value of the L bits. The system operator sets these five least significant bits by means of an address switch on each instrument before applying power to the system. The controller's address code may be implemented in software. The controller may listen to bus traffic without actually addressing itself over the bus. No L or LE function is active during the time that ATN is asserted.

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.

SOURCE AND ACCEPTOR HANDSHAKE FUNCTIONS (SH and AH)

Like the T/TE and L/LE functions, the SH and AH functions are totally independent of each other. The SH function guarantees proper transmission of data, while the AH function guarantees proper reception of data. The interlocked handshake sequence between these two functions quarantees asychronous transfer of each data byte. The handshake sequence is performed via the NRFD, DAV, and NDAC signal lines on the bus. See Fig. A-5. Both functions must respond to ATN within 200 nanoseconds.

The SH function must wait for the RFD message plus at least 2 ms 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 microseconds. Faster sttling times are allowed under special conditions and warning notes in the standard. The time it takes for the AH function to accept an interface message byte is dependent on how the designer implemented the function.

DEVICE CLEAR FUNCTION (DC)

The DC function allows a 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 respond. 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 goes to as a result of receiving the DCL or SDC command; it may be, but does not have to be, the power-up default settings. In general, these commands are used only to clear the GPIB interface circuits within an instrument.

DEVICE TRIGGER FUNCTION (DT)

The DT function allows the controller-in-charge to start a basic operation specified for a particular instrument. The basic operation may be different for any group of instrument on the bus. The IEEE 488 standard does not specify an instrument's basic operation it 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 which 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 bsic operation in response to the next GET message. Thus, the basic operating time is the major factor that determines how fast an instrument can be repeatedly "triggered" by commands from the bus.

CONTROLLER (C), SERVICE REQUEST (SR), AND PARALLEL POLL (PP) FUNCTIONS

The Controller function provides the capability to send primary talk and listen addresses, secondary addresses, universal commands, and addressed commands to instruments on the bus. The Controller function also provides the capability to respond to a service request message (SRQ) from an instrument or to conduct a parallel poll routine to determine the status of any or all instruments on the bus that have the Service Request (SR) or Parallel Poll (PP) functions implemented.

If an instrumentation system has more than one controller, only the system controller is allowed to assert the IFC and REN lines at any time during system operation, whether or not it is the controller-in-charge at the time.

ERR? RESPONSE

COMMAND ERRORS

0	No error
1	Number error
2	Invalid character in block ISO count
3	EOI in block ISO
4	EOI in block binary
5	Checksum error in block binary
6	Illegal placement of query
7	Invalid query
8	Invalid header
9	Invalid end
10	Invalid character argument (not allowed)
11	Invalid number argument (not allowed)
12	Invalid string argument (not allowed)
13	Invalid binary argument (not allowed)
14	Link not allowed
15	Invalid link label
16	Empty link label
17	Invalid character value (not allowed)
18	Invalid number value (not allowed)
19	Invalid string value (not allowed)
20	Invalid binary value (not allowed)
21	Link argument not allowed as link value
22	Character not found
23	Invalid suffix
24	Input buffer overflow

EXECUTION ERRORS

26	Output buffer overflow
27	Attempt to execute in local mode
28	FREQ or TUNE beyond range
30	FRCAL out of range
31	SPAN not available
32	RESBW not available
33	Minimum atten (MINATT/MAXPWR) out of range
34	REFLVL out of range
35	VRTDSP out of range (LIN argument)
36	VRTDSP out of range (LOG argument)
37	TIME out of range
38	DEGAUS not allowed in present span/div
40	FIBIG, LFTNXT, or RGTNXT not allowed in present span/div (zero span)
41	ADDR/DATA argument invalid
42	ADDR not compatible for DATA command
43	CRVID or WFID not valid
44	WFMPRE not compatible with 496P

EXECUTION WARNINGS

50	SPAN defaulted to MAX
51	SPAN defaulted to 0
52	UNCAL light on
53	Multiple use of display buffer

INTERNAL ERRORS

57	TUNE carry from lower DAC failed
58	Phase lock failed
59	Lost phase lock
60	Failed to recenter when phase lock turned off

COMMAND AND QUERY DESCRIPTIONS

Message Unit	Page
ARES (auto resolution bandwidth)	4-9
AVIEW (A memory display)	4-26
BMINA (B-A memory display)	4-28
BVIEW (B memory display)	4-26
CENSIG (center signal)	6-5
CLIP (clip baseline)	4-32
CRSOR (peak/average cursor)	4-30
CURVE (display curve)	5-6
DEGAUS (degauss tuning coils)	4-5
DELFR (delta-frequency)	4-4
EOS (end of sweep)	7-6
ERCNT? (error count)	7-10
ERR? (error)	7-11
FIBIG (find biggest signal)	6-2
FINE (fine reference level steps)	4-15
FMAX (find maximum value)	6-3
FMIN (find minimum value)	6-3
FRCAL (frequency cal)	4-5
FREQ (center frequency)	4-3
GRAT (graticule illumination)	4-32
ID? (identify instrument)	7-4
INIT (initialize settings)	7-3
LFTNXT (find left-next signal)	6-2
LORDO? (lower readout)	5-10
MAXPWR (maximum input power)	4-18
MINATT (min RF attenuation)	4-17
MXHLD (max hold)	4-29
PHSLK (phase lock)	4-10
PLSTR (pulse stretcher)	4-19
POINT (display data point)	6-11
RDOUT (readout message)	5-9
REDOUT (readout display)	4-32
REFLVL (reference level)	4-13
REPEAT (repeat execution)	7-5
RESBW (resolution bandwidth)	4-8
RGTNXT (find right-next signal)	6-2
RLMODE (reference level mode)	4-16
RQS (request service)	7-7
SAVEA (save A memory)	4-27
SET? (instrument settings)	7-1
SIGSWP (single sweep)	4-23
SPAN (frequency span/div)	4-7
Status byte (serial poll)	7-8
TIME (time/div)	4-24
TOPSIG (move signal to top)	6-5
TRIG (triggering mode)	4-22
TUNE (increment frequency)	4-4
UPRDO0? (upper readout)	5-11
VIDFLT (video filters)	4-20
VRTDSP (vertical display)	4-13
WAIT (wait to execute)	7-4
WFMPRE (waveform preamble)	5-2
ZEROSP (zero span)	4-10

STATUS BYTE

8 7 6 5 4 3 2 1	Decimal	Condition
0 1 0 X 0 0 0 1	65, 81	Power-on
0 X 0 X 0 0 1 0	2, 18, 66, 82	End of sweep
0 0 0 X 0 0 0 0	0, 16	Ordinary operation
0 X 1 X 0 0 0 1	33, 49, 97, 113	Command error
0 X 1 X 0 0 1 0	34, 50, 98, 114	Execution error
0 X 1 X 0 0 1 1	35, 51, 99, 115	Internal error
0 X 1 X 0 1 0 1	37, 53, 101, 117	Execution error warning
0 X 1 X 0 1 1 0	38, 54, 102, 118	Internal error warning

- Four-bit status code
- 496P microcomputer busy condition
- Abnormal (1)/normal (0) condition
- SRQ is asserted (depends on RQS and EOS commands)

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

INTERFACE MESSAGES

DCL (20) — Clears I/O buffer and status byte
GET (8) — Aborts and then rearms sweep
GTL (1) — Go to local control
IFC (IFC line) — Initializes talker and listener functions
LLO (17) — Local lockout
PPC (5) — Parallel poll configure
PPU (21) — Parallel poll unconfigure
SDC (4) — Same as DCL if listener addressed
SPD (24) — Serial poll disable
SPE (24) — Serial poll enable

DISPLAY DATA I/O

Header	Argument	Function
CURVE	CRVID and integers or block binary	Display data I/O
LORDO?		Upper crt readout
RDOUT	Up to 32 characters or NORM	Crt readout message (no query)
UPRDO?		Lower crt readout
WFMPRE	WFID and ENCDG	Waveform preamble

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

**** **POI**NT? is the only waveform processing query ****

Header	Argument	Function
RGTNXT	Threshold NUM	Find right-next signal
TOPSIG		Change reference level to move signal to top of crt

SYSTEM

Header	Argument	Function
EOS	ON, OFF	End-of-sweep SRQ control
ERCNT?		Error count
ERR?		Errors
ID?		Identify instrument
INIT		Initialize settings
REPEAT		Repeat execution
RQS	ON, OFF	Turn on or off abnormal SRQ mask
SET?		Request instrument settings
WAIT		Wait for end of sweep (no query)

Notes:

Only the first three letters of a mnemonic are required: **ARE** for **ARE**S.

Form a query by adding a question mark to the header: **ARE**S?

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.**

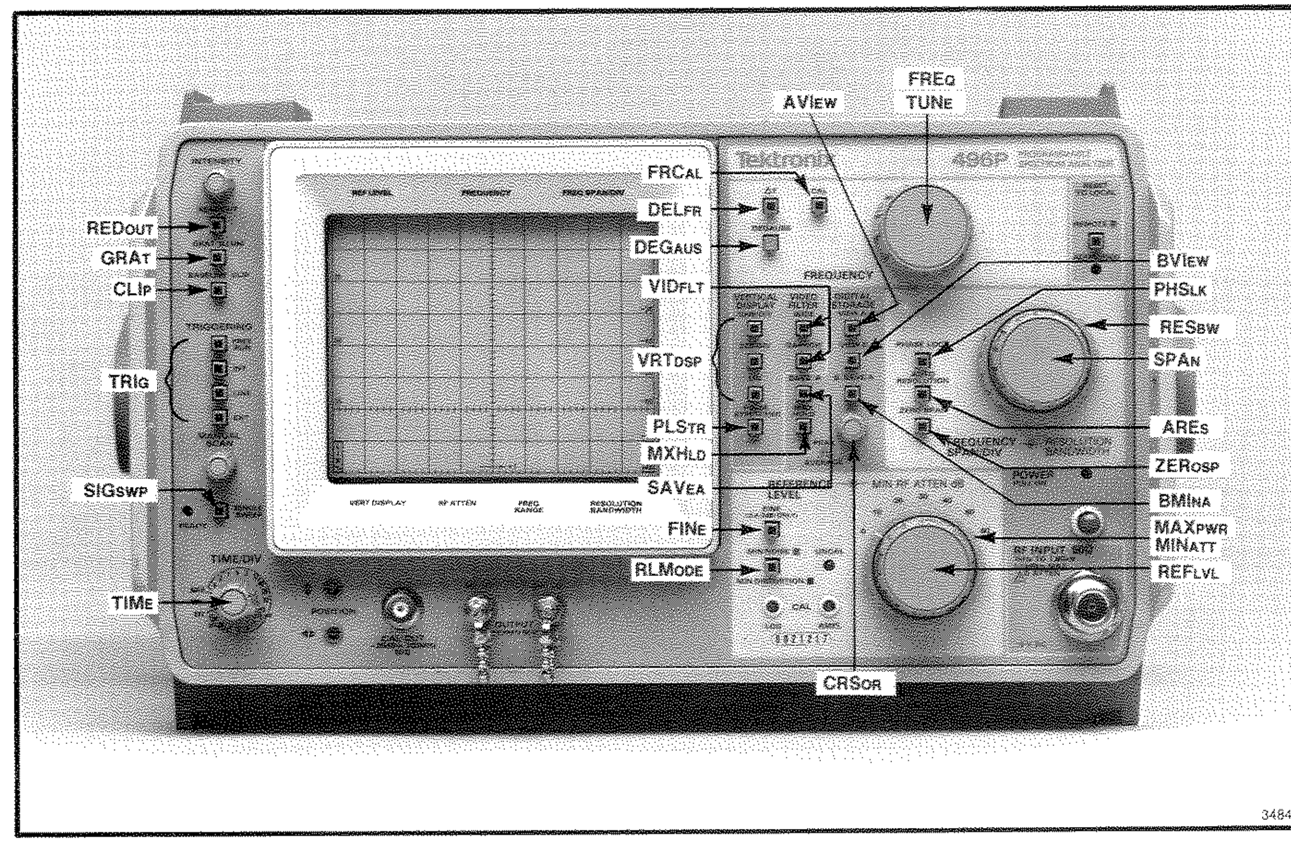


Fig. A-6. 496P Front panel controls.

FRONT PANEL

Header	Argument	Function
ARES	ON, OFF	AUTO RESOLUTION bandwidth
AVIEW	ON, OFF	VIEW A memory
BMINA	ON, OFF	B−SAVE A display
BVIEW	ON, OFF	VIEW B memory
CLIP	ON, OFF	BASELINE CLIP
CRSOR	KNOB, PEAK, AVG	PEAK/AVERAGE cursor
DEGAUS		DEGAUSS (no query)
DELFR	ON, OFF	ΔF
FINE	ON, OFF	FINE ref level steps
FRCAL	NUM	CAL frequency (no query)
FREQ	NUM	FREQUENCY
GRAT	ON, OFF	GRAT ILLUM
MAXPWR	NUM, INC, DEC	Max input power
MXHLD	ON, OFF	MAX HOLD
MINATT	NUM, INC, DEC	MIN RF ATTEN
PHSLK	ON, OFF	PHASE LOCK
PLSTR	ON, OFF	PULSE STRETCHER
REDOUT	ON, OFF	READOUT

Header	Argument	Function
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
SIGSWP		SINGLE SWEEP
SPAN	NUM, INC, DEC, MAX	FREQUENCY SPAN/DIV
TIME	NUM, AUTO, INC, DEC, MAN, EXT	TIME/DIV
TRIG	FRERUN, INT, LINE, EXT, NUM	TRIGGERING
TUNE	NUM	Increment frequency (no query)
VIDFLT	OFF, WIDE, NARROW, NUM	VIDEO FILTER
VRTDSP	LOG: NUM, LIN: NUM, INC, DEC	VERTICAL DISPLAY

MANUAL CHANGE INFORMATION

At Tektronix, we continually strive to keep up with latest electronic developments by adding circuit and component improvements to our instruments as soon as they are developed and tested.

Sometimes, due to printing and shipping requirements, we can't get these changes immediately into printed manuals. Hence, your manual may contain new change information on following pages.

A single change may affect several sections. Since the change information sheets are carried in the manual until all changes are permanently entered, some duplication may occur. If no such change pages appear following this page, your manual is correct as printed.