
Installation, Operation, Programming, and Service Guide

HP 85620A Mass Memory Module



**HP Part No. 85620-90041 Supersedes 85620-90040
Printed in USA October 1996**

©Copyright Hewlett-Packard Company 1992, 1996
All Rights Reserved. Reproduction, adaptation, or translation without
prior written permission is prohibited, except as allowed under the
copyright laws.
1400 Fountaingrove Parkway, Santa Rosa, CA 95403-1799, USA

Certification

Hewlett-Packard Company certifies that this product met its published specifications at the time of shipment from the factory. Hewlett-Packard further certifies that its calibration measurements are traceable to the United States National Institute of Standards and Technology, to the extent allowed by the Institute's calibration facility, and to the calibration facilities of other International Standards Organization members.

Warranty

This Hewlett-Packard instrument product is warranted against defects in material and workmanship for a period of one year from date of shipment. During the warranty period, Hewlett-Packard Company will, at its option, either repair or replace products which prove to be defective.

For warranty service or repair, this product must be returned to a service facility designated by Hewlett-Packard. Buyer shall prepay shipping charges to Hewlett-Packard and Hewlett-Packard shall pay shipping charges to return the product to Buyer. However, Buyer shall pay all shipping charges, duties, and taxes for products returned to Hewlett-Packard from another country.

Hewlett-Packard warrants that its software and firmware designated by Hewlett-Packard for use with an instrument will execute its programming instructions when properly installed on that instrument. Hewlett-Packard does not warrant that the operation of the instrument, or software, or firmware will be uninterrupted or error-free.

LIMITATION OF WARRANTY

The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by Buyer, Buyer-supplied software or interfacing, unauthorized modification or misuse, operation outside of the environmental specifications for the product, or improper site preparation or maintenance.

NO OTHER WARRANTY IS EXPRESSED OR IMPLIED. HEWLETT-PACKARD SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

EXCLUSIVE REMEDIES

THE REMEDIES PROVIDED HEREIN ARE BUYER'S SOLE AND EXCLUSIVE REMEDIES. HEWLETT-PACKARD SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER BASED ON CONTRACT, TORT, OR ANY OTHER LEGAL THEORY.

Assistance

Product maintenance agreements and other customer assistance agreements are available for Hewlett-Packard products.

For any assistance, contact your nearest Hewlett-Packard Sales and Service Office.

Safety Notes

The following safety notes are used throughout this manual. Familiarize yourself with each of the notes and its meaning before operating this instrument.

Caution

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

Warning

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

General Safety Considerations

Warning

Before this instrument is switched on, make sure it has been properly grounded through the protective conductor of the ac power cable to a socket outlet provided with protective earth contact.

Any interruption of the protective (grounding) conductor, inside or outside the instrument, or disconnection of the protective earth terminal can result in personal injury.

Warning

There are many points in the instrument which can, if contacted, cause personal injury. Be extremely careful.

Any adjustments or service procedures that require operation of the instrument with protective covers removed should be performed only by trained service personnel.

Caution

Before this instrument is switched on, make sure its primary power circuitry has been adapted to the voltage of the ac power source.

Failure to set the ac power input to the correct voltage could cause damage to the instrument when the ac power cable is plugged in.

Contents

1. Installation	
Introduction	1-1
HP 85620A Mass Memory Module Description	1-1
Modules Covered by This Manual	1-3
Serial Numbers	1-3
Firmware Revisions	1-3
Specifications and Characteristics	1-5
Auxiliary Interface Connector	1-6
Memory Card Specifications and Characteristics	1-8
Preparation for Use	1-9
Initial Inspection	1-9
Installing the Module	1-9
Installing Memory Cards	1-11
Changing the Memory Card Battery	1-11
Memory Card Demonstration Programs	1-13
Module Operation	1-14
Introduction	1-14
Menu Structure	1-14
Main Menu Softkey Descriptions	1-15
Packaging	1-16
Original Packaging	1-16
Other Packaging	1-17
Error Messages	1-18
Error Codes	1-18
2. Operation	
Introduction	2-1
Module Features	2-2
Using the Mass Memory Module	2-3
Main Menu Keys Descriptions	2-3
USER KEYS Menu	2-4
TRACE SAVE/RCL Menu	2-5
LIMIT LINE Menu	2-13
AUTOEXEC Menu	2-22
KEYDEF Menu	2-26
UTILITY Menu	2-28

3. Softkey Menu	
Introduction	3-1
4. Programming	
Introduction	4-1
Getting Started with DLPs	4-1
What Is a DLP?	4-1
DLP Examples	4-2
Equipment Setup	4-2
Programming Front-Panel Functions	4-3
Programming User-Defined Functions	4-4
Determining Available User Memory	4-6
User-Defined Variable and Trace Declaration	4-7
Displaying Variable and Trace Values	4-8
Using EP to Modify User-Defined Variables (firmware revisions 910116 and later)	4-9
Erasing User-Defined Memory	4-10
Storing DLPs on RAM Cards and in the Module	4-11
Programming Hints	4-12
Command Syntax Diagrams	4-17
ABORT	4-28
ABS	4-30
ACTVFUNC	4-31
ADD	4-33
ARRAYDEF	4-35
AUTOEXEC	4-37
AUTOFUNC	4-38
AUTOSAVE	4-40
AVG	4-41
CARDLOAD	4-43
CARDSTORE	4-45
CATALOG	4-47
CLRDSR	4-53
CLRSCHED	4-54
CNTLA	4-55
CNTLB	4-56
CNTLC	4-57
CNTLD	4-58
CNTLI	4-59
CTRLHPIB	4-60
DATEMODE	4-62
DISPOSE	4-63
DIV	4-66
DSPLY	4-68
EDITDONE	4-70
EDITLIML	4-72
EM	4-74
ENTER	4-75
EXP	4-76
FUNCDEF	4-78
IF/THEN/ELSE/ENDIF	4-80
INT	4-82
KEYCLR	4-84
KEYDEF	4-85
LCLVAR	4-87

LIMD	4-89
LIMF	4-91
LIMIFAIL	4-93
LIMIPURGE	4-94
LIMIRCL	4-96
LIMIREL	4-97
LIMISAV	4-99
LIMITEST	4-100
LIML	4-101
LIMM	4-103
LIMTFL	4-105
LIMTSL	4-107
LIMU	4-109
LOG	4-111
MEAN	4-114
MEM	4-115
MENU	4-116
MIN	4-117
MOD	4-119
MODRCLT	4-121
MODSAVT	4-122
MOV	4-124
MPY	4-125
MSDEV	4-127
MXM	4-128
ONEOS	4-130
OR	4-131
OUTPUT	4-133
PA	4-136
PD	4-138
PDA	4-139
PDF	4-141
PEAKS	4-142
PR	4-144
PU	4-146
RELHPIB	4-148
REPEAT/UNTIL	4-150
RETURN	4-153
RMS	4-155
SADD	4-156
SDEL	4-158
SDON	4-159
SEDI	4-161
SENDER	4-163
SETDATE	4-164
SETTIME	4-166
SHOWMENU	4-167
SKYCLR	4-169
SKYDEF	4-171
SMOOTH	4-173
SQR	4-175
STDEV	4-176
SUB	4-177
SUM	4-179
SUMSQR	4-180

TEXT	4-181
TIMEDATE	4-182
TRDEF	4-183
VARDEF	4-185
VARIANCE	4-187

5. Service

Turning on the Module	5-1
Electrostatic Discharge	5-1
Static-Safe Accessories	5-2
Returning Modules for Service	5-2
Back-Up Battery Voltage Test	5-2
Replacement Procedures	5-3
Introduction	5-3
A1 Memory Board Assembly Replacement	5-3
B1 Module Battery Replacement	5-3
Memory Card Battery Replacement	5-6
Troubleshooting and Replaceable Parts	5-7
Introduction	5-7
Troubleshooting	5-7
Missing Features	5-7
Back-up Battery Voltage Test	5-7
Memory Card Connector	5-7
Error Codes	5-8
Replaceable Parts	5-9
Component-Level Information	

A. Appendixes

Component-Level Information Packet for 85620-60001	
Component-Level Information Packet for 85620-60006	
Component-Level Information Packet for 85620-60008	
Component-Level Information Packet for 85620-60018	

Figures

1-1. HP 85620A Mass Memory Module and HP 85700A Memory Card (Option T01 does not include memory card capability.)	1-1
1-2. HP 85620A Dimensions	1-5
1-3. Command Timing OUTPUT 98-LSB First or OUTPUT 99-MSB First	1-6
1-4. Auxiliary Connector Timing OUTPUT 98/OUTPUT 99	1-7
1-5. Memory Card Dimensions (Option T01 does not include memory card capability.)	1-8
1-6. Installing an HP 85620A Mass Memory Module and Memory Card (Option T01 does not include memory card capability.)	1-10
1-7. Memory Card Battery Replacement (Option T01 does not include memory card capability.)	1-12
1-8. HP 85620A Main Menu Softkeys	1-14
1-9. HP 85620A Shipping Container Materials (Option T01 does not include memory card capability.)	1-16
2-1. HP 85620A Main Menu Softkeys	2-1
2-2. User Keys Menu with DLP Label Examples	2-4
2-3. Trace Save/Recall Menu Softkeys	2-6
2-4. Autosave Trace Menu Softkeys	2-8
2-5. Edit Autosave Menu Softkey	2-8
2-6. Save-Trigger Menu Softkeys	2-9
2-7. Recall Trace Menu Softkeys	2-11
2-8. Limit-line Menu Softkeys	2-14
2-9. Recall Line Menu Softkeys	2-14
2-10. Save Line Menu Softkeys	2-16
2-11. Edit Line Menu Softkeys and Table Example	2-17
2-12. Sample Limit Line Using Figure 2-11 Table Data	2-17
2-13. Enter Parameters Menu Softkeys	2-18
2-14. Segment Type Menu Softkey	2-19
2-15. Autoexec Menu Softkeys	2-22
2-16. Edit Autoexec Softkeys	2-22
2-17. DLP Directory Example	2-23
2-18. Execute Criteria Softkeys	2-24
2-19. Keydef Menu Softkeys	2-26
2-20. Utility Menu Softkeys (MEM selected)	2-28
2-21. Utility Menu Softkeys (CARD selected)	2-28
2-22. Timedate Menu Softkeys	2-29
3-1. User Keys	3-2
3-2. Trace Save/Rcl	3-3
3-3. Limit Line	3-4
3-4. Autoexec Menu	3-5
3-5. Keydef	3-6
3-6. Utility	3-7
4-1. Equipment Setup	4-2

4-2. Preparing the Spectrum Analyzer for Programming . .	4-3
4-3. Typical DLP Listing	4-4
4-4. Program Structure Flowchart	4-15
5-1. Static-Safe Workstation	5-1
5-2. A1TP1 Configuration	5-2
5-3. A1 Board Assembly & Battery Replacement (Serial Prefix <3143A)	5-4
5-4. A1 Board Assembly & Battery Replacement (Serial Prefix ≥3143A)	5-5
5-5. Memory Card Battery Replacement	5-6
5-6. HP 85620A Parts Identification (Serial Prefix <3143A)	5-10
5-7. HP 85620A Parts Identification (Serial Prefix ≥3143A)	5-11
5-8. HP 85620A Block Diagram	5-14

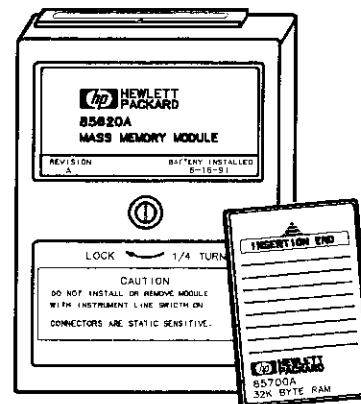
Tables

1-1. HP 85620A and HP 856X Firmware Compatibility . . .	1-4
1-2. HP 85620A Mass Memory Module Specifications and Characteristics	1-5
1-3. Subminiature "D" (9-pin) Connector Pinout	1-6
1-4. Auxiliary Connector Timing Parameters OUTPUT 98/OUTPUT 99	1-7
1-5. HP 85700A RAM Memory Card Specifications and Characteristics (Option T01 does not include memory card capability.)	1-8
1-6. HP 85620A Shipping Container Materials Contents . . .	1-17
1-7. Hewlett-Packard Sales and Service Offices	1-20
2-1. Autosave-Function Settings and Results	2-11
4-1. Functional Command Listing	4-23
4-2. Alphabetical Command Listing	4-25
4-3. Dispose Command File Search Hierarchy and Programming Examples	4-63
5-1. Static-Safe Accessories	5-2
5-2. Manufacturer's Code List	5-9
5-3. HP 85620A Parts Identification	5-12
5-4. Firmware Revision and ROM Part Numbers	5-13
5-5. HP 85620A Mass Memory Module Documented Assemblies	

Installation

Introduction

This chapter describes the HP 85620A Mass Memory Module, provides specifications and characteristics, and illustrates accompanying accessories. The serial numbers covered by this document are also listed here.



CF246

Figure 1-1.
HP 85620A Mass Memory Module and HP 85700A Memory Card
(Option T01 does not include memory card capability.)

HP 85620A Mass Memory Module Description

Note

The spectrum analyzer save-trace registers 5, 6, and 7 are overwritten when the module is connected to the its rear panel OPTION MODULE CONNECTOR. These registers are used for module data and are not available for spectrum analyzer operation. If you attempt to store data to these registers while the module is attached, module operating memory is corrupted.

The mass memory module is an optional memory package that is attached to Hewlett-Packard portable spectrum analyzers. It provides substantially greater user memory than other spectrum analyzer products. This general-purpose memory may be used to store trace data, downloadable programs (DLPs), limit lines, and variables. The module memory is battery-backed. The battery needs to be checked at least annually. Refer to Chapter 5, "Service," for information on how to check the battery voltage.

The module is attached to the OPTION MODULE connector (J3) on the spectrum analyzer rear panel and locked into place with a 1/4-turn fastener. The module interface with the spectrum analyzer is established through this connector.

Caution

You must turn the spectrum analyzer OFF before attaching or removing the mass memory module. Connecting the module while the spectrum analyzer is ON can damage both the spectrum analyzer and the module circuitry.

The connector pins on the module and on the spectrum analyzer are electrostatic discharge (ESD) sensitive. Do not touch the pins on either instrument unless you are adequately protected against ESD.

You may also store data on memory cards that are compatible with the module. Memory cards contain lithium batteries which must be checked at least annually. Refer to Chapter 5, "Service," for how to check the battery. Instructions for installing memory cards and batteries are in "Installing Memory Cards" in this chapter. (Option T01 does not include memory card capability.)

Modules Covered by This Manual

Serial Numbers

The serial-number label is on the rear cover (connector side) of the mass memory module. The first five characters make up the serial number prefix; the last five are the suffix. The only time the serial prefix changes is when there are substantial changes to the module. Suffix numbers are different for each module.

Firmware Revisions

Mass memory module operation is controlled by ROM (read-only memory) firmware. The module firmware version is displayed after pressing the **MODULE** key on the spectrum analyzer. Refer to Table 1-1 for firmware versions of the module that are compatible with the different portable spectrum analyzers

It is possible to get a different set of functions from a given mass memory module, depending on the firmware revision of the host spectrum analyzer. In spectrum analyzers with firmware revision 960401 and later, the firmware that controls the mass memory module actually resides in the host spectrum analyzer and contains a more recent set of features and functionality. If that same mass memory module is installed on a host spectrum analyzer with firmware revision 941028 or earlier, the firmware that resides in the mass memory module (revision A, B, or C) will control the features and functionality of the module.

**Table 1-1.
HP 85620A and HP 856X Firmware Compatibility**

Spectrum Analyzer	HP 85620A Firmware Revision		
	890214 (Rev A)	890524 (Rev B)	910116 (Rev C)
HP 8560A	Not supported	890720 and later firmware revisions	890720 and later firmware revisions
HP 8560E	Not supported	941028 and earlier firmware revisions	All revisions of firmware
HP 8561A	All revisions of firmware	All revisions of firmware	All revisions of firmware
HP 8561B	Not supported	890720 and later firmware revisions	890720 and later firmware revisions
HP 8561E	Not supported	941028 and earlier firmware revisions	All revisions of firmware
HP 8562A	870728 and later firmware revisions	870728 and later firmware revisions	870728 and later firmware revisions
HP 8562B	870728 and later firmware revisions	870728 and later firmware revisions	870728 and later firmware revisions
HP 8562E	Not supported	Not supported	All revisions of firmware
HP 8563A	Not supported	All revisions of firmware	All revisions of firmware
HP 8563E	Not supported	941028 and earlier firmware revisions	All revisions of firmware
HP 8564E	Not supported	941028 and earlier firmware revisions	All revisions of firmware
HP 8565E	Not supported	941028 and earlier firmware revisions	All revisions of firmware
Mass memory module firmware revision 950829 is in spectrum analyzers having instrument firmware revision 960401.			
Mass memory module firmware revision 960830 is in spectrum analyzers having instrument firmware revision 960830.			

Specifications and Characteristics

Specifications describe the warranted HP 85620A Mass Memory Module performance over the indicated temperature range. Characteristics provide useful information in the form of typical, nominal, or approximate values. Table 1-2 lists specifications and characteristics of the module. Refer to Table 1-5 for specifications and characteristics of the RAM memory card and its battery. (Option T01 does not include memory card capability.)

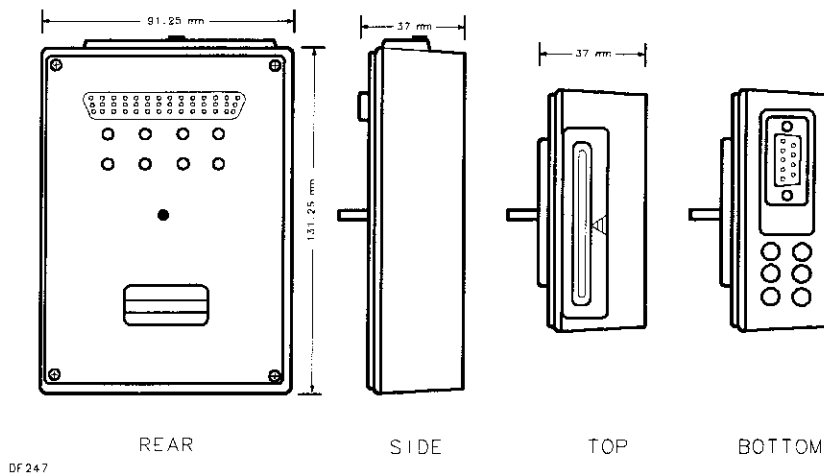


Figure 1-2. HP 85620A Dimensions

**Table 1-2.
HP 85620A Mass Memory Module Specifications and Characteristics**

Electrical Specifications	General Specifications	Characteristics
<p>Operating Power Requirement 5.0 W at +5 Vdc, supplied by the analyzer, not by the battery.</p> <p>Read Only Memory (ROM) 256 kilobytes</p> <p>User Memory 128 kilobytes (battery-backed RAM)</p> <p>Real-Time Clock +50, -120 ppm over temperature</p>	<p>Environmental Military Specification: per MIL-T-28800C, Type III, Class 3</p> <p>Temperature Range Operating -10°C to +55°C</p> <p>Storage -62°C to +85°C</p>	<p>Module Battery Lithium Iodine 2.8 V, 1 A-hour capacity See Table 5-3 for the battery part number.</p> <p>Battery Life Worst Case: 1.0 year Typical: 6.5 years</p> <p>Module Weight Net: 453 g (1 lb) Shipping: 1.59 kg (3.5 lb)</p> <p>Module Dimensions (See Figure 1-2) Height: 131.25 mm (5.25 in) Width: 91.25 mm (3.65 in) Depth: 37 mm (1.48 in)</p>

Auxiliary Interface Connector

The auxiliary interface connector allows the user, by way of the spectrum analyzer, a method of providing minimal power to external devices and to control or receive inputs from external devices. This interface is to provide an alternative to HP-IB in the control of simple switches in “Logic” mode and to control external devices in “Serial Bit” mode. The “Serial Bit” mode is especially useful for Automated Test Equipment (ATE) applications.

The Auxiliary Interface Connector has four output control lines and one input line. In the “Serial Bit” mode one of the outputs is designated as the serial line and another as the strobe. There are actually two serial modes: “OUTPUT 99” sends out the data most significant bit (MSB) first, and “OUTPUT 98” sends out the least significant bit (LSB) first.

As implemented the Auxiliary Interface Connector runs open loop. The serial data is clocked out as fast as the software can toggle the data lines as illustrated in Figure 1-3.

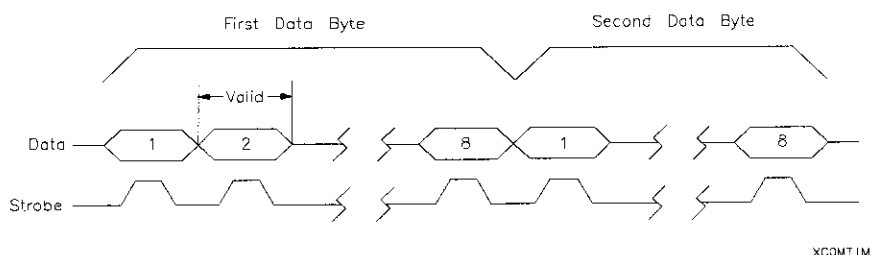


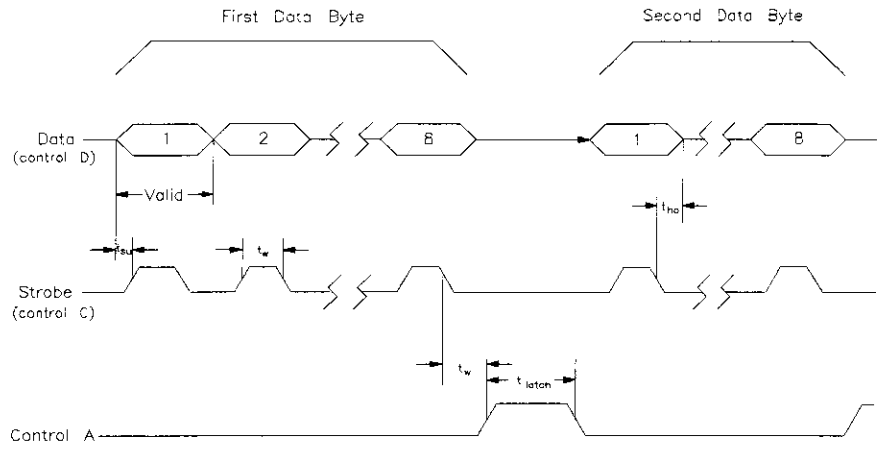
Figure 1-3.
Command Timing
OUTPUT 98-LSB First or OUTPUT 99-MSB First

Table 1-3. Subminiature “D” (9-pin) Connector Pinout

Pin Number	Function	Current	“Logic” Mode	“Serial Bit” Mode
1	Control A		TTL Output Hi/Lo	TTL Output Hi/Lo
2	Control B		TTL Output Hi/Lo	TTL Output Hi/Lo
3	Control C		TTL Output Hi/Lo	Strobe
4	Control D		TTL Output Hi/Lo	Serial Data
5	Control I		TTL Input Hi/Lo	TTL Input Hi/Lo
6	Ground		Ground	Ground
7	-15 Vdc ±5%	150 ma		
8	+5 Vdc ±5%	150 ma		
9	+15 Vdc ±5%	150 ma		

Note

The current drawn by devices using the Auxiliary Interface connector must be limited to 150 ma since this is the limit of the power supply.



XAUXCON

Figure 1-4.
Auxiliary Connector Timing
OUTPUT 98/OUTPUT 99

Table 1-4.
Auxiliary Connector Timing Parameters
OUTPUT 98/OUTPUT 99

Symbol	Parameter	Limit
t_{su}	Minimum setup time, data to strobe	250 ns
t_{ho}	Minimum hold time, strobe to data	250 ns
t_w	Minimum pulse width, strobe	500 ns
t_d	Maximum delay, strobe to data not read	250 ns
t_{latch}	Minimum pulse width, latch	150 μ s

Memory Card Specifications and Characteristics

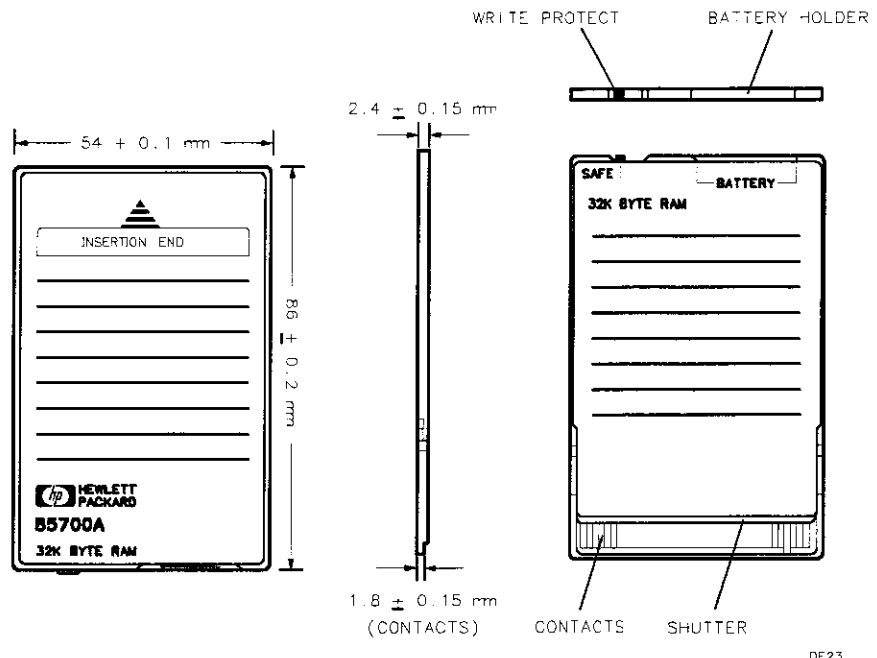


Figure 1-5.
Memory Card Dimensions
 (Option T01 does not include memory card capability.)

Table 1-5.
HP 85700A RAM Memory Card Specifications and Characteristics
 (Option T01 does not include memory card capability.)

Electrical Specifications	General Specifications	Characteristics
Operating Voltage (supplied by the spectrum analyzer) +5 Vdc (nominal)	Operating Temperature 0°C to +60°C	Card Weight with battery: 20.9 grams without battery: 19.0 grams
Battery Voltage (RAM back-up) 3 Vdc (nominal)	Storage Temperature -20°C to +60°C	Card Dimensions (<i>Figure 1-5</i>) Length: 86 mm ±0.2 mm Width: 54 mm ±0.1 mm Thickness: 2.4 mm ±0.15 mm
	Memory Size 32 kilobyte, battery-backed RAM	Battery Type Lithium Commercial Part Number: CR2016 HP Part Number:1420-0383
	Type CMOS	

Preparation for Use

In this section, we cover initial inspection of the module and the shipping container, installing the module, and module operation.

Initial Inspection

Inspect the shipping container upon receipt. Retain it and the cushioning materials for future use. If you need to ship the module to another location or return it to Hewlett-Packard for service, refer to the repackaging and shipping instructions in this section.

If the container or the cushioning materials are damaged, keep them until you have verified that the contents are complete and the module is functioning properly. If the contents are incomplete or the module does not function properly, notify one of the HP Sales and Service Offices listed in Table 1-7. The HP Sales and Service Office will arrange for repair or replacement without waiting for a claim settlement. Also, notify the carrier about container damages, then show the damaged items to the carrier for inspection.

Refer to Figure 1-9 for an illustration of the shipping container, packaging materials, and associated HP part numbers.

Installing the Module

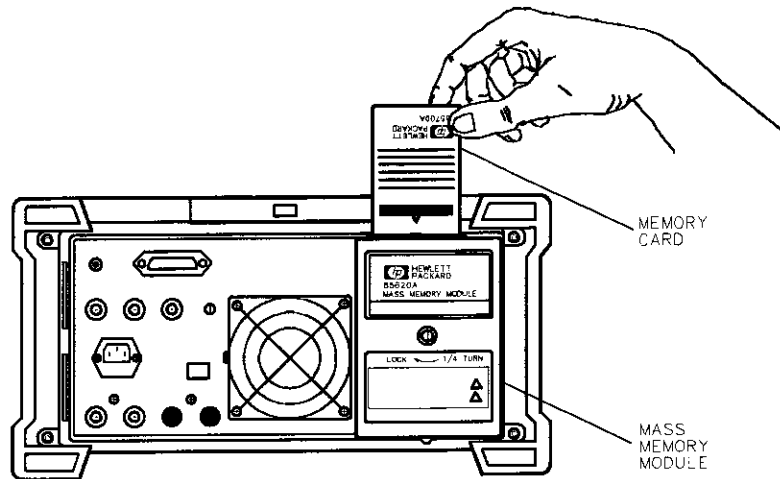
Caution

The portable spectrum analyzer must be turned OFF before the module is connected. Connecting the module while the spectrum analyzer is ON can damage both the spectrum analyzer and the module circuitry.

The connector pins on the module and on the spectrum analyzer are ESD-sensitive. Do not touch the pins on either instrument unless you are adequately protected against ESD.

Refer to the following steps to install your HP 85620A Mass Memory Module properly and safely onto a portable spectrum analyzer.

1. With the spectrum analyzer set to OFF, line the 50-pin connector on the module up with the OPTION MODULE connector (J3) on the rear panel of the analyzer. See Figure 1-6.
2. Press the module into place.
3. Using a flat-blade screwdriver, tighten the 1/4-turn fastener that holds the module in place.



DF 24

Figure 1-6.
Installing an HP 85620A Mass Memory Module and Memory Card
(Option T01 does not include memory card capability.)

Installing Memory Cards

(Option T01 does not include memory card capability.)

Use the following information to ensure that the memory card is inserted correctly. Improper insertion causes error messages to occur, but generally does not damage the card or the module. Care must be taken, however, not to force the card into place. The cards are easy to insert when installed properly.

1. Locate the arrow printed on the card label.
2. Insert the card **with its arrow matching the raised arrow on the bezel around the card-insertion slot**. See Figure 1-6.
3. Press the card into the slot. When correctly inserted, about 4 cm (1.6 in) of the card is exposed above the slot.

Changing the Memory Card Battery

The battery is located beside the card write-protect switch on the end opposite the connector. Refer to Table 1-5 for memory-card battery specifications and characteristics.

Caution

Memory-card data is retained by the battery in the card. You can lose the data when the battery is removed. Replace the battery while the card is installed in a powered-up module. Memory-card data may be backed up in module memory before beginning the battery replacement procedure that follows.

1. Locate the groove along the edge of the battery clip. See Figure 1-7.
2. Gently pry the battery clip out of the card. The battery fits within this clip.
3. Replace the battery, making sure the plus (+) sign on the battery is on the same side as the plus (+) sign on the clip.
4. Insert the battery clip into the memory card, holding the clip as oriented in Figure 1-7. (Face the “open” edge of the clip toward the write-protect switch on the memory card.)

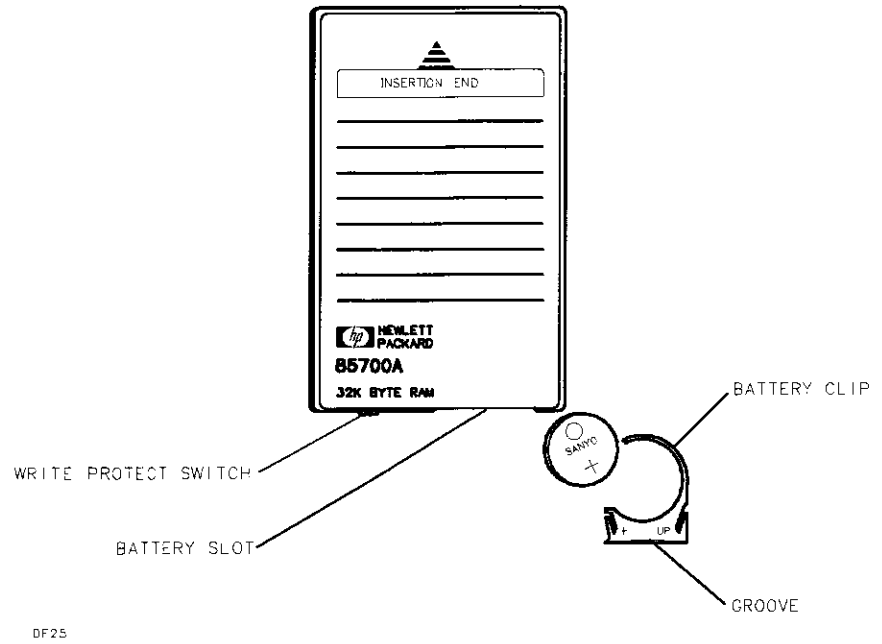


Figure 1-7.
Memory Card Battery Replacement
(Option T01 does not include memory card capability.)

Memory Card Demonstration Programs

To introduce the concept of downloadable programs (DLP) and demonstrate other features of the module, Hewlett-Packard has installed demonstration DLPs onto the HP 85700A 32-kilobyte RAM memory card shipped with each HP 85620 Mass Memory Module.

Caution

To prevent possible damage to the spectrum analyzer or module circuits, the analyzer must be turned off before installing or removing the module.

To load and execute these demonstration DLPs, perform the following steps:

1. Turn the analyzer off and plug the mass memory module into the OPTION MODULE connector on the rear panel of the analyzer. Tighten the locking screw to secure the module.
2. Turn on the analyzer and wait for the alignment to complete. Press the **MODULE** key on the front panel of the instrument. You will now see the module main menu on the display.
3. Press the **UTILITY** softkey.
4. Insert the HP 85700A RAM card into the mass memory module. Be sure the card is oriented properly by matching the arrow on the card to the arrow on the module bezel.
5. Press **CATALOG MEM CARD** softkey to underline **CARD**. Make sure the cursor is located beside **DEMOS** on the display (use the analyzer **RPG** knob or step keys to move the cursor if necessary), then press **COPY TO MEM**. After approximately 10 seconds, the programs are copied into the module. During this time the front-panel keys are inoperable.
6. Press the **MODULE** key, then the **KEYDEF** softkey. Now press **CHOOSE DLP** and position the cursor beside **DEMOS**.
7. Press **EXECUTE NOW** to begin the DLP demonstrations. Use the information on the display to complete the programs.

Module Operation

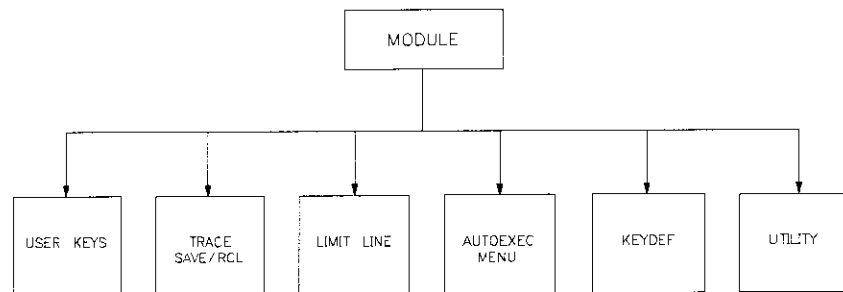
Introduction

This operating information is a brief introduction to the main menu of the HP 85620A Mass Memory Module. For detailed operating information, refer to Chapter 2.

Menu Structure

With the spectrum analyzer in normal operation mode, press the spectrum analyzer **MODULE** key to display the main menu of the HP 85620A Mass Memory Module. See Figure 1-8. The menu is friendly enough for the first-time user, but structured so that an experienced user can get going quickly. Generally, there are no more than three menu levels nested below the main menu level.

To exit the module functions, press any of the front-panel keys on the analyzer.



DF27

Figure 1-8. HP 85620A Main Menu Softkeys

Main Menu Softkey Descriptions

USER KEYS

accesses 10 user-definable softkeys. These keys, when defined using the User Entry Menu, activate DLPs assigned to them. You can also assign DLPs to softkeys remotely.

TRACE SAVE/RCL

enables you to save and recall traces. You can also create trigger criteria (data specifying when an event starts and stops) and automatically store traces.

LIMIT LINE

accesses features for you to create or edit limit lines. Limit lines can be specified by up to 18 points. Each point is composed of a frequency and at least one amplitude value. Trace data can be compared against limit lines.

AUTOEXEC MENU

accesses features for you to execute a DLP automatically. You select the criteria that determines when the DLP starts to run.

KEYDEF

accesses the 10 user-definable softkeys and the DLP Directory so that you can load DLPs onto the module User Keys Menu softkeys.

UTILITY

accesses the module utility functions. You can set the current time and date, catalog the module or memory card contents, copy data between the memory card and the module memory, and delete any memory contents. (Option T01 does not include memory card capability.)

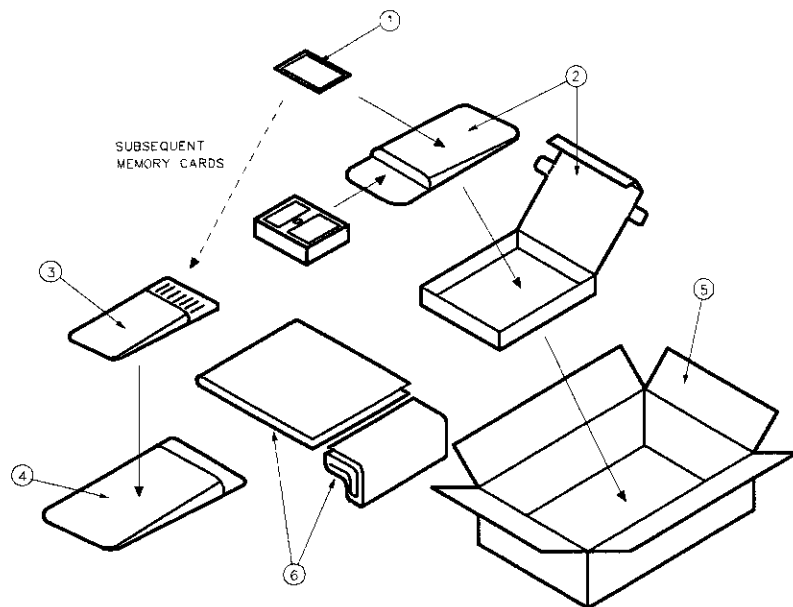
Packaging

Original Packaging

Save the original packaging materials. If the original materials have been discarded, identical materials may be ordered from HP Sales and Service Offices for shipping or transporting purposes. Refer to Figure 1-9 and Table 1-6 for these shipping container materials.

On the outside of the container, write clearly, **FRAGILE, HANDLE WITH CARE**. If the module is being returned to Hewlett-Packard for servicing, include one of the blue repair tags along with the information listed below:

- Type of service required, including a description of the problem.
- Return address, phone number, and person to contact for more information.
- Model number and serial number of the module.
- List of any accessories accompanying the module.



DF 26

Figure 1-9.
HP 85620A Shipping Container Materials
(Option T01 does not include memory card capability.)

Table 1-6. HP 85620A Shipping Container Materials Contents

Item	Description	HP Part Number	Check Digit	Qty
1	Memory card with slip case (Option T01 does not include memory card capability.)	85700A	3	1
2	Envelope and inner carton	9211-4916	3	1
3	Bubble-pack bag (with separate card orders)	9222-0784	8	1
4	Envelope (with separate card orders)	9222-1219	6	1
5	Carton, outer	9211-5570	7	1
6	Styrene sheets	9223-0476	7	2

Other Packaging

Use the steps below if you plan to use materials other than the ones specified as original packaging.

Caution

Do not use packaging materials other than those specified. Improper packaging can damage the module. Never use styrene pellets in any shape as packaging materials. Their cushioning ability may not be adequate enough prevent the module from shifting within the carton. They also generate static electricity and can cause ESD damage to the module.

1. Wrap the module in two or three inches of static-shielding cushioning materials (for example, S.D.-240 Air Cap™ from Sealed Air Corporation, Commerce, CA, 90001).
2. If the module is being returned to Hewlett-Packard, include a blue repair tag with the information listed in "Original Packaging."
3. Place the module in a strong shipping container. Make sure there is enough cushioning material to prevent the module from shifting within the container. Securely seal the shipping container.
4. Print FRAGILE , HANDLE WITH CARE clearly on the shipping container.

Error Messages

Error messages and recovery information are included in this section. If you are unable to recover from an error, contact an HP Sales and Service Office. These offices are listed in Table 1-7.

CHK CARD INSERTION

The memory card is either inserted incorrectly or not inserted at all. Make sure the arrow printed on the card label and the arrow on the module match up. Press the card into place firmly, but do not use excessive force. (Option T01 does not include memory card capability.)

INSUFFICIENT MEMORY

There is not enough user memory available to save the data you are attempting to save. You can either relocate some of the contents in the destination you have selected, or purge some of the contents.

READ ONLY CARD The card inserted in the module is a ROM card. Replace the ROM card with a RAM card and be sure the write-protect switch is *not* set in the SAFE position. (Option T01 does not include memory card capability.)

No applicable entries

You attempted to access data from a memory location that contained no DLPs, limit-lines, or traces.

Error Codes

Error-code numbers for the module range from 800 to 899.

Note

Pressing **MODULE** then **PRESET** on the analyzer, or sending the command IP over HP-IB, will clear module errors reported in the lower left-hand corner of the display. After using the ERR command over HP-IB, an IP command must be used to clear mass memory module errors.

- | | |
|------------|--------------------------------------------------------------------------------------------------------------------------|
| 800 | U1 EPROM Check Sum Error, hardware failure. |
| 801 | U2 EPROM Check Sum Error, hardware failure. |
| 802 | U3 EPROM Check Sum Error, hardware failure. |
| 803 | U4 EPROM Check Sum Error, hardware failure. |
| 804 | Mass Memory Initialized. RAM data has been erased. Hardware failure or the module battery is intermittent or has failed. |
| 805 | Mass Memory Module Usage Error. The command used was invalid and cannot be executed. |
| 806 | Mass Memory RAM full. Available memory is insufficient for command execution. |
| 807 | Symbol Define Error. There was an attempt to define something illegal, or not of the correct type. |

- 808** Symbol Read Error. There was an attempt to read something that was not there.
- 809** Symbol Write Error. There was an attempt to write something that was not defined.
- 810** Symbol Delete Error. There was an attempt to delete something that did not exist.
- 850** Symbol Table Corrupt. The file apparently exists but cannot be accessed.
- 851** Memory Card Not Inserted. The module cannot recognize the memory card. It may be inserted incorrectly or the contacts are damaged. (Option T01 does not include memory card capability.)
- 852** Write to ROM Card or a Write-Protected RAM card. You must use a RAM card with the write-protect switch *not* set to SAFE. (Option T01 does not include memory card capability.)
- 853** Memory Card Operation Error. There was an attempt to do an illegal operation. For example, a write-to attempt was made with the memory card removed, or the filename used for the store operation contained non-alphanumeric characters. (Option T01 does not include memory card capability.)
- 854** Memory Card Memory Full. (Option T01 does not include memory card capability.)
- 855** Memory Card File Not Found. (Option T01 does not include memory card capability.)
- 856** Reserved Word. There was an attempt to use a word in a filename that is reserved for a command.
- 857** Syntax Error.
- 858** Type Error. The defined type does not match the requested type.
- 859** Command Error. Destination invalid or improperly defined.
- 860** Reserved.

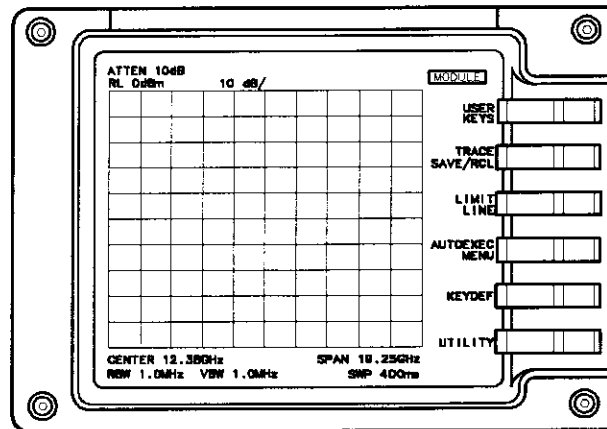
Table 1-7. Hewlett-Packard Sales and Service Offices

US FIELD OPERATIONS		
<p>Headquarters Hewlett-Packard Co. 19320 Pruneridge Avenue Cupertino, CA 95014 (800) 752-0900</p>	<p>California, Northern Hewlett-Packard Co. 301 E. Evelyn Mountain View, CA 94041 (415) 694-2000</p>	<p>California, Southern Hewlett-Packard Co. 1421 South Manhattan Ave. Fullerton, CA 92631 (714) 999-6700</p>
<p>Colorado Hewlett-Packard Co. 24 Inverness Place, East Englewood, CO 80112 (303) 649-5512</p>	<p>Atlanta Annex Hewlett-Packard Co. 2124 Barrett Park Drive Kennesaw, GA 30144 (404) 648-0000</p>	<p>Illinois Hewlett-Packard Co. 5201 Tollview Drive Rolling Meadows, IL 60008 (708) 255-9800</p>
<p>New Jersey Hewlett-Packard Co. 150 Green Pond Rd. Rockaway, NJ 07866 (201) 586-5400</p>	<p>Texas Hewlett-Packard Co. 930 E. Campbell Rd. Richardson, TX 75081 (214) 231-6101</p>	
EUROPEAN FIELD OPERATIONS		
<p>Headquarters Hewlett-Packard S.A. 150, Route du Nant-d'Avril 1217 Meyrin 2/Geneva Switzerland (41 22) 780.8111</p>	<p>France Hewlett-Packard France 1 Avenue Du Canada Zone D'Activite De Courtaboeuf F-91947 Les Ulis Cedex France (33 1) 69 82 60 60</p>	<p>Germany Hewlett-Packard GmbH Hewlett-Packard Strasse 61352 Bad Homburg v.d.H Germany (49 6172) 16-0</p>
<p>Great Britain Hewlett-Packard Ltd. Eskdale Road, Winnersh Triangle Wokingham, Berkshire RG41 5DZ England (44 734) 696622</p>		
INTERCON FIELD OPERATIONS		
<p>Headquarters Hewlett-Packard Company 3495 Deer Creek Road Palo Alto, California, USA 94304-1316 (415) 857-5027</p>	<p>Australia Hewlett-Packard Australia Ltd. 31-41 Joseph Street Blackburn, Victoria 3130 (61 3) 895-2895</p>	<p>Canada Hewlett-Packard (Canada) Ltd. 17500 South Service Road Trans-Canada Highway Kirkland, Quebec H9J 2X8 Canada (514) 697-4232</p>
<p>China China Hewlett-Packard Company 38 Bei San Huan X1 Road Shuang Yu Shu Hai Dian District Beijing, China (86 1) 256-6888</p>	<p>Japan Hewlett-Packard Japan, Ltd. 1-27-15 Yabe, Sagamihara Kanagawa 229, Japan (81 427) 59-1311</p>	<p>Singapore Hewlett-Packard Singapore (Pte.) Ltd. 150 Beach Road #29-00 Gateway West Singapore 0718 (65) 291-9088</p>
<p>Taiwan Hewlett-Packard Taiwan 8th Floor, H-P Building 337 Fu Hsing North Road Taipei, Taiwan (886 2) 712-0404</p>		

Operation

Introduction

This chapter introduces you to the operation of the HP 85620A Mass Memory Module. The module is designed to enhance the memory and capabilities of Hewlett-Packard portable high performance spectrum analyzers. Figure 2-1 illustrates the main menu display which appears after you press the **MODULE** key on the spectrum analyzer front panel. Refer to Chapter 1, "Installation," for information about connecting the module to the spectrum analyzer or inserting a memory card into the module.



DF 28

Figure 2-1. HP 85620A Main Menu Softkeys

Module Features

The module provides the spectrum analyzer with 128 kilobytes of available memory. With this memory you can store and recall trace data, define and store variables, use downloadable programs (called DLPs or personalities), create and store limit lines, or define user softkeys so that you can activate DLPs from the front panel.

Features provided to the portable spectrum analyzer by the mass memory module include:

- DLP execution remotely over HP-IB or using the module softkeys. There are ten softkeys you can assign to activate DLPs you purchase through Hewlett-Packard or create yourself. These softkeys are defined remotely with the KEYDEF command or locally in the KEYDEF menu.
- Trace Save/Recall capability. With the module built-in real-time clock and calendar, you can set up automatic-save conditions to save trace data at specified times and dates, and specified intervals.
- Limit-line generation. You enter the frequency and amplitude parameters to generate a limit line, then activate it with the press of a key. It tracks changes in the spectrum analyzer state and adjusts the displayed line accordingly.
- Automatic execution of DLPs. You can enter up to seven operations or DLPs in the AUTOEXEC/AUTOSAVE SCHEDULE that will execute automatically according to the criteria you define. The location of the operation name in the schedule and the operation start- and stop-times determine its priority for execution. As an example, if a program listed beside the number 1 has an earlier start time than the program listed beside the number 2, program number 1 is executed first.
- Memory card capability (except Option T01). With the mass memory module, you are able to use RAM and ROM memory cards. RAM cards are primarily used for read/write data storage, while the one-time programmable (OTP) ROM cards are used for personality DLPs available from Hewlett-Packard. RAM card memory is maintained by a lithium battery when the card is removed from a powered-up module. Check the battery voltage at least annually to be assured of data retention.

Note

When mass memory module functions are used, the analyzer save-trace registers 5, 6, and 7 are overwritten with module data. Therefore, these registers are not available for spectrum analyzer operations. If you attempt to store data to these registers while the module is active, it corrupts the module operating memory.

Note

To perform a complete instrument preset from the front panel of the spectrum analyzer, you must first press **MODULE**, then **PRESET**. If you do not press **MODULE** first, some module error codes will not clear.

Using the Mass Memory Module

This section provides descriptions of the module menus and operation information to help you learn to use the HP 85620A Mass Memory Module. It begins with the first menu that appears after you press the **MODULE** key on the spectrum analyzer, then moves through the various levels of menu. Generally, there are no more than three menus nested beneath the **MODULE** key. Press **MODULE** to activate the module main menu.

Note

You should not attempt to perform multiple mass-memory module/spectrum analyzer operations simultaneously. Doing so can produce improper or unexpected results. For example, you should not perform signal identification or attempt to download a DLP while an autoexec/autosave operation is in progress.

Main Menu Keys Descriptions

The main menu offers six softkey selections. Press any one of the keys to access their lower-level menus. Refer to each menu section for specific information. A brief description of the main-menu softkeys is provided below.

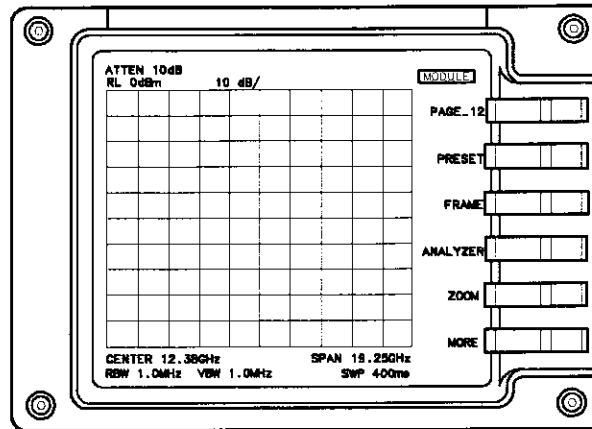
- Press **USER KEYS** to access the module ten user-definable softkeys. After you have loaded a program onto one of these softkeys, you can activate it from this menu by pressing its softkey. There are two menu pages: one appears when you press **USER KEYS** and the other is accessed by pressing **MORE**.
- Press **TRACE SAVE/RCL** to access softkeys for saving or recalling a spectrum analyzer trace. Trace A or trace B may be stored in either the battery-backed RAM of the module or on a memory card. (Option T01 does not include memory card capability). Traces can be saved manually or automatically using the Autosave function.
- Press **LIMIT LINE** to access the softkeys that allow you to create, review, save, recall, or edit limit lines. You can turn an active limit line on or off in this menu as well.
- Press **AUTOEXEC MENU** to access the AUTOEXEC/AUTOSAVE SCHEDULE. Use the softkey of the Autoexec menu to set up automatic-execution conditions for DLPs. You can modify conditions and priority, or eliminate functions (DLPs) that are scheduled to execute, in this menu.
- Press **KEYDEF** to assign a DLP to one of the User Keys menu softkeys. Display the softkey labels by pressing **USER KEYS**.
- Press **UTILITY** to access the various Utility menu features of the module. A few of these are the time/date settings, cataloging the module or card memory, and deleting DLPs, limit lines, variables, or traces from memory.

USER KEYS Menu

Press **USER KEYS** to access five user-definable softkeys and a sixth one labeled **MORE**; press **MORE** to access five additional user-definable softkeys and a sixth one labeled **PREV MENU**. Press **PREV MENU** to return to the first set of user-definable softkeys.

There are 10 user-definable softkeys that can be labeled through the Keydef menu. Softkeys that have not been assigned a program name display the default label **EMPTY**. DLPs can be written so that they redefine softkeys for specified functions within the program.

Once you have loaded a program onto one of the User Keys menu softkeys, you can activate its operation simply by pressing its softkey. Labeling the softkeys is defined in the Keydef menu section of this chapter.



DF 29

Figure 2-2. User Keys Menu with DLP Label Examples

TRACE SAVE/RCL Menu

TRACE SAVE/RCL activates the Trace Save/Recall functions menu. See Figure 2-3. In this menu you can access the features that allow you to manually or automatically save spectrum analyzer traces, or recall traces from memory. If you save a trace without specifying a unique name, it is given the default name "TR" and a time/date stamp when traces are saved on a memory card, or "TRACE" and a time/date stamp when saved to module memory. To create a trace title, you can use the spectrum analyzer screen-title function or the remote command, TITLE. Press the **DISPLAY** key, then **MORE** to access the **SCREEN TITLE** softkey. Memory cards are LIF formatted (Logical Interchange Format). LIF entries are from 1 to 10 ASCII characters long. The module reserves one of the 10 for file encoding purposes, such as .LMT, .TRC, or .DLP. The user, therefore provides 1 to 9 of the ASCII characters for memory card entries. (Refer to the LIF Document, HP part number 5955-2676.)

The information saved by the save-trace operation includes trace data, spectrum analyzer state, the trace name, if one is created, and the time/date that the trace was stored. The date and time format is YYMMDDHHMMSS.

Note

Traces stored on memory cards using a mass-memory module with firmware datecode 910116 or later cannot be read into a module with an earlier firmware datecode (for example, 890524). Traces stored on cards using modules with firmware datecode 890524 or earlier can be read into a module with firmware datacode 910116 or later.

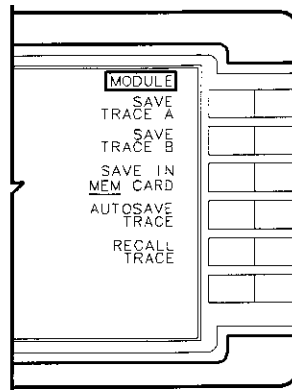
If available memory is insufficient, the save-trace operation is aborted. The message **INSUFFICIENT MEMORY** appears momentarily in the active function block of the CRT.

The first step in saving traces is to name the trace, if you choose to, and select where you want the trace saved. Choose module or card memory by underlining your preference with **SAVE IN MEM CARD**. (Option T01 does not include memory card capability.) Manually save trace A or B by selecting **SAVE TRACE A** or **SAVE TRACE B**. These last two softkeys are immediate-execute function keys. Immediate-execute means that pressing the softkey immediately activates its function.

If you want to save traces automatically, select **AUTOSAVE TRACE**, determine whether to save trace A or B by underlining your preference with the **AUTOSAVE TRA TRB** softkey. Move the cursor to the position of priority you desire in the **AUTOEXEC/AUTOSAVE SCHEDULE** (placing the trace at position one makes it the first priority), then press **EDIT AUTOSAVE**. In this menu, enter start- and stop-time/date values with the spectrum analyzer data keys and set up save-triggering conditions by pressing **SAVE TRIGGER**. Refer to the softkeys described in the following pages for more information.

Note

A maximum of one trace per second can be saved.



DF210

Figure 2-3. Trace Save/Recall Menu Softkeys

SAVE TRACE A

highlights and immediately stores trace A and the instrument state in the memory location you selected using the **SAVE IN MEM CARD** softkey. (Option T01 does not include memory card capability.)

SAVE TRACE B

highlights and immediately stores trace B and the instrument state in the memory location you selected using the **SAVE IN MEM CARD** softkey. (Option T01 does not include memory card capability.)

SAVE IN MEM CARD

toggles between memory locations. The default setting is MEM, which stores the trace in the internal, battery-backed RAM of the module. Choose CARD to save the trace on a memory card. If CARD is selected and the write-protect switch is set to SAFE, an error message appears on the display. If the card is inserted incorrectly, or not at all, the message CHECK CARD INSERTION is displayed in the active function block of the spectrum analyzer display.

Note

The memory card must be inserted correctly. Match the black arrow on the memory-card label with the raised arrow on the module bezel, then insert the card into the slot. Also be aware of whether the write-protect switch on the RAM card is switched to the write position or to SAFE.

AUTOSAVE TRACE

displays the AUTOEXEC/AUTOSAVE SCHEDULE and accesses another menu from which you can set up conditions that save a trace automatically.

Note

The default name "TRACE" is assigned to traces saved in module memory without unique names. When saving to a card, the default name "TR" and a time/date stamp is assigned to the saved trace unless you create and assign a unique name using the spectrum analyzer **SCREEN TITLE** function key. Memory cards are LIF formatted (Logical Interchange Format). LIF entries are from 1 to 10 ASCII

characters long. The module reserves one of the 10 for file encoding purposes, such as .LMT, .TRC, or .DLP. The user, therefore, provides 1 to 9 of the ASCII characters for memory card entries. (Refer to the LIF Document, HP part number 5955-2676.) In either case, the time and date are saved along with the trace data. Return to the Trace Save/RCL menu by pressing **(MODULE)**, **TRACE SAVE/RCL**.

Note

Traces stored on memory cards using a mass-memory module with firmware datecode 910116 or later cannot be read into a module with an earlier firmware datecode (for example, 890524). Traces stored on cards using modules with firmware datecode 890524 or earlier can be read into a module with firmware datacode 910116 or later.

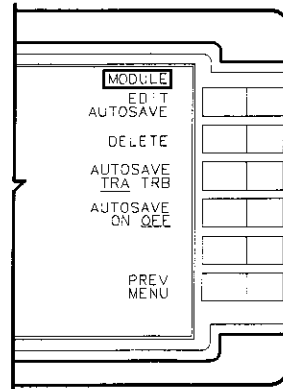
The following steps may be used to set up the Autosave operation.

1. Determine where you want the trace saved, either the module memory or card memory. Press **SAVE IN MEM CARD** until your preference is underlined.
2. Press **AUTOSAVE TRACE**. Move the cursor, using the RPG knob or STEP keys, to one of the seven **FUNCTION NAME** positions in the schedule. This position helps determine the priority of the trace-saving operation in relation to other operations scheduled here.
3. Press **AUTOSAVE TRA TRB** until your preference is underlined.
4. Press **EDIT AUTOSAVE** and define the criteria to save traces automatically. Refer to Table 2-1 in this chapter for Autosave function settings and results information. The label **AUTOSAVE** is loaded into the **AUTOEXEC/AUTOSAVE SCHEDULE** in the position you selected.
5. Complete the sequence by pressing **PREV MENU**, then **AUTOSAVE ON OFF** to ON. This activates the Autosave function.

The trace is stored by its assigned name and according to your start/stop time settings and trigger conditions. Recall stored traces by using the **RECALL TRACE** softkey. Autosave Trace menu softkeys are illustrated in Figure 2-4.

Note

If Autosave is set to ON and there are no DLPs currently running, then you select **EDIT AUTOSAVE**, any currently running Autosave operation is suspended until you have completed your edits and exited the Edit Autosave menu. **(PRESET)** sets Autosave to OFF.

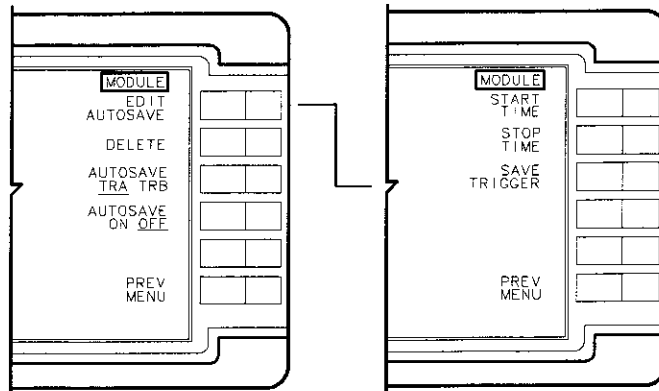


DF211

Figure 2-4. Autosave Trace Menu Softkeys

EDIT AUTOSAVE

displays the AUTOEXEC/AUTOSAVE SCHEDULE and the start/stop times and dates and trigger selections that you can set up for Autosave operations. Press either the start- or stop-time softkey to set up these conditions. Use the spectrum analyzer data keys to enter time values in 24-hour clock mode. The **SAVE TRIGGER** softkey operation is described below.



DF212

Figure 2-5. Edit Autosave Menu Softkey

START TIME

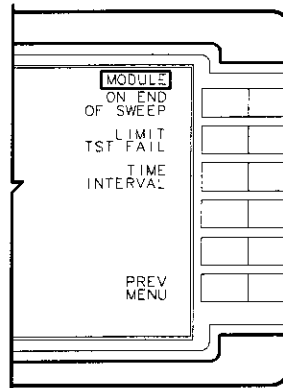
allows entry of the start-time value which determines the time, as well as the date, that the Autosave function begins operation. Use the spectrum analyzer data keys and terminate the entry with any units key. After being entered, the time and date appear beside START TIME in the AUTOEXEC/AUTOSAVE SCHEDULE. The default value is negative infinity.

STOP TIME

allows entry of the stop-time value which determines the time, as well as the date, that the Autosave function ends. Use the spectrum analyzer data keys and terminate the entry with any units key. After being entered, the time and date appear beside STOP TIME in the AUTOEXEC/AUTOSAVE SCHEDULE. If you do not enter a stop-time, or you enter a stop-time that is earlier than the start-time, the stop-time entry is ignored, and when Autosave is active, the function stops only when the memory location is full. The default value is positive infinity.

SAVE TRIGGER

accesses a menu in which conditions that automatically save a trace may be selected. Select the conditions you want using the softkeys illustrated in Figure 2-6 and described below. The default setting is ON EOS (on end of sweep).



DF 213

Figure 2-6. Save-Trigger Menu Softkeys

ON END OF SWEEP

saves the trace at the end of the next sweep to occur after the start-time you provided. ON EOS appears next to TRIGGER in the schedule display.

LIMIT TST FAIL

saves the active trace at the end of the sweep any time the trace exceeds the parameters of an active limit line. LIMIT TEST appears next to TRIGGER in the schedule.

Note

The default name "TRACE" is assigned to traces saved in module memory without unique names. When saving to a card, the default name "TR" and a time/date stamp is assigned to the saved trace unless you create and assign a unique name using the spectrum analyzer

SCREEN TITLE function key. Trace data is saved any time the trace exceeds the active limit-line parameters.

Note

If a file has been stored into memory using a 16-character title, the first 9 characters must be unique to avoid writing over an existing file. If a file name is longer than 9 characters or if lower-case letters are used, the file name will be converted, if possible, to an LIF compatible file name using only the first 9 characters.

TIME INTERVAL

displays **INTERVAL** (HHMMSS) next to **TRIGGER** in the schedule. Use the spectrum analyzer data keys to enter the time interval you want the Autosave function to use. Trace data is stored in the memory location you selected after the end of a sweep is reached and at the first interval specified. The last time trace data is saved is when the stop-time, minus the time interval, is reached.

PREV MENU

returns you to the Edit Autosave menu softkeys in Figure 2-6. Press **PREV MENU** again to return to the Autosave Trace menu.

DELETE

deletes the function name you have highlighted by locating the cursor beside its number in the **AUTOEXEC/AUTOSAVE SCHEDULE**.

AUTOSAVE TRA TRB

toggles between saving trace A or trace B. Press the softkey until your preference is underlined. The default setting is TRA.

AUTOSAVE ON OFF

toggles to set the Autosave function to on or off. **AUTOSAVE OFF** deactivates all Autosave functions. Before turning the Autosave function on, you must determine which memory location trace data is to be stored in, either module memory or on the memory card. If stop/start times or trigger conditions have not been specified, **AUTOSAVE ON** will set default conditions and begin storing traces. Refer to Table 2-1 which lists the results of setting start/stop times or trigger conditions. Autosave remains activated until the stop-time is reached or until the memory location is full. When memory is full, Autosave stops and **INSUFFICIENT MEMORY** is displayed on the spectrum analyzer screen.

PREV MENU

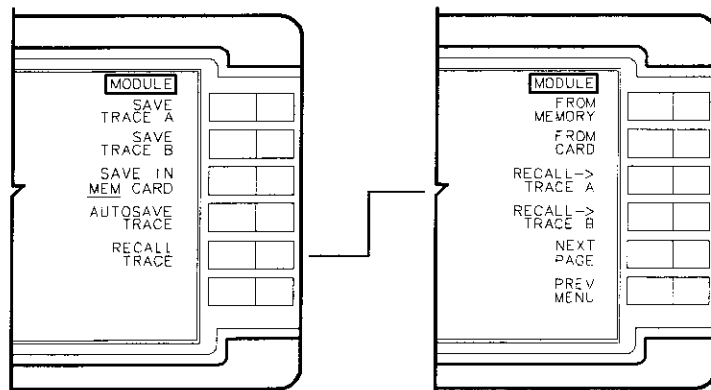
displays the main menu of **TRACE SAVE/RCL**.

Table 2-1. Autosave-Function Settings and Results

Trigger	Start Time	Stop Time	Autosave Results
Unspecified	Unspecified	Unspecified	Autosave off.
Unspecified	Unspecified	Specified	Autosave data now. Save at the end of each sweep. Autosave until specified stop time is reached or memory is full.
Unspecified	Specified	Unspecified	Start at specified time. Save at the end of each sweep, then until memory is full.
Specified	Specified	Specified	Auto save when start time is reached. Save when the specified trigger condition is met, then until memory is used up or the specified stop time is reached.
Specified	Unspecified	Unspecified	Autosave data when trigger is reached and until memory is full.

RECALL TRACE

accesses the TRACE/ARRAY DIRECTORY. From here you can retrieve a saved trace and state for review. Select the memory location, either the default module memory or card memory, from which to recall a stored trace. The default setting is **FROM MEMORY**. If no trace is available from the location you selected, the message **No applicable entries** is displayed. Display the stored trace and state in trace A or trace B, according to your **RECALL—TRACE** selection. Traces are displayed with their associated state in the spectrum analyzer view mode.



DF214

Figure 2-7. Recall Trace Menu Softkeys

FROM MEMORY

displays the TRACE/ARRAY DIRECTORY and all traces stored in the battery-backed RAM of the HP 85620A Mass Memory Module. The date and time that the trace was saved is displayed in the lower box of the directory when you highlight the trace in the directory with the cursor. Use either the RPG knob or STEP keys on the spectrum analyzer to highlight your selection.

FROM CARD

displays the MEMORY CARD DIRECTORY and all traces stored on the card currently inserted into the module. Traces saved without created names are labeled TR and an encoded date code. If the card is improperly inserted, missing, or empty, the message **No applicable entries** appears. Highlight your trace selection using either the RPG knob or STEP keys. (Option T01 does not include memory card capability.)

Note

Traces stored on memory cards using a mass-memory module with firmware datecode 910116 or later cannot be read into a module with an earlier firmware datecode (for example, 890524). Traces stored on cards using modules with firmware datecode 890524 or earlier can be read into a module with firmware datacode 910116 or later.

RECALL→ TRACE A

immediately recalls the trace and its associated spectrum analyzer state from the specified location. The state is changed to that of the recalled trace and trace A is placed in the view mode.

RECALL→ TRACE B

immediately recalls the trace and its associated spectrum analyzer state from the specified location. The state is changed to that of the recalled trace and trace B is placed in the view mode.

NEXT PAGE

displays the subsequent page(s) of the directory. There can be up to 54 entries per page of directory.

PREV MENU

displays the **TRACE SAVE/RCL** main menu.

LIMIT LINE Menu

A limit line is a display line specified by a set of coordinate points, each point consists of a frequency component, an amplitude value, and, optionally, a second amplitude value. The amplitude values represent either upper/lower or middle/delta amplitude limits for the corresponding frequency component. Limit lines work only when the spectrum analyzer vertical scaling is in the log mode, not the linear mode. One of the advantages of using the HP 85620A Mass Memory Module limit-line feature is that, coupled with its Autosave function, trace data can be stored automatically any time a signal fails a specified limit.

A set of limit-line coordinates makes up a segment. These segments are entered into a limit-line table. There can be up to 18 segments per limit line. You may enter amplitude values in one of the two following formats.

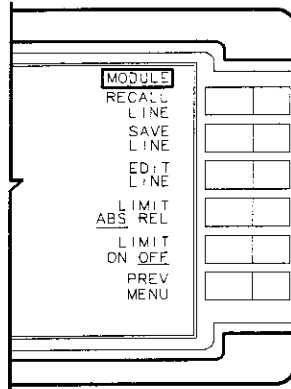
- Upper/Lower: input upper and/or lower amplitude values. If an upper-limit value is not entered, the upper limit assumes a default value of +50 dBm. If a lower-limit value is not entered, the lower limit assumes the default value of -175 dBm. If the lower-limit value is greater than the upper-limit value, the two values are set equal to the value entered last.
- The middle/delta format requires the input of a middle amplitude value. You may also specify a deviation (positive and negative values) from either side of this value. If no deviation is entered, the deviation defaults to zero.

The limit line is displayed according to the spectrum analyzer settings (instrument state). If the spectrum analyzer is in single-sweep mode and you change the settings, the limit line is not updated until another sweep is initiated.

The two types of limit lines that you can generate are relative and absolute limit lines. Relative limit lines consist of frequency components referenced to a center frequency, and amplitude components relative to the reference level. A frequency component of 0 Hz corresponds to the current center frequency of the signal. An amplitude component of -10 dB indicates that -10 dB is added to the reference level value to obtain the amplitude of the given component (reference level offset included).

Absolute limit lines contain only absolute amplitude and frequency values. The amplitude and frequency offsets of the spectrum analyzer (FOFFSET and ROFFSET) do, however, affect the absolute displayed amplitude and frequency. This then influences the actual location of the limit line on the screen.

Limit lines are generated from a set of coordinates, interpolated between any two points the user has entered in the limit-line table for a given segment. With a relative limit line, the line is mapped to the screen based on the spectrum analyzer span. Absolute limit-line displays are influenced by the span, center frequency, and reference level. Figure 2-8 illustrates the Limit-line menu softkeys.

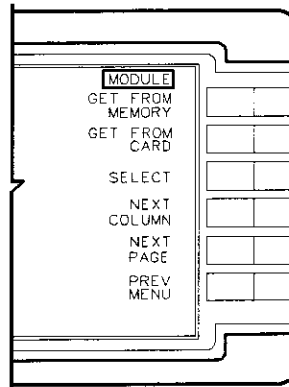


d126a

Figure 2-8. Limit-line Menu Softkeys

RECALL LINE

activates the menu from which you can recall a limit line. The first two softkeys specify the memory location from which to recall a limit line. The RAM memory of the module is the default location. If no limit lines exist in your selected location, the message **No applicable entries** is displayed on the CRT. If there are several limit lines, notice in the upper right-hand corner of the directory the message **PAGE 1 of ?**. Press the **NEXT PAGE** softkey to review subsequent pages of the directory. The Recall Line menu softkeys are illustrated in Figure 2-9 and explained after the figure.



d127a

Figure 2-9. Recall Line Menu Softkeys

GET FROM MEMORY

displays the limit lines stored in the battery-backed RAM of the module. Use the RPG knob or STEP keys to move the cursor to the limit line you want to recall. Its full name is displayed in the lower box of the directory. Press **SELECT** to activate the limit line.

GET FROM CARD

displays limit lines stored on a memory card. If the card is empty, improperly installed, or not installed at all, the message **No applicable entries** is displayed on the CRT. Press **SELECT** to activate the limit line. (Option T01 does not include memory card capability.)

Note

Limit lines stored on memory cards using a mass-memory module with firmware datecode 910116 or later cannot be read into a module with an earlier firmware datecode (for example, 890524). Limit lines stored on cards using modules with firmware datecode 890524 or earlier can be read into a module with firmware datecode 910116 or later.

SELECT

selects the limit line that you have highlighted. Highlight the limit-line title by using the STEP keys or RPG knob to move the cursor to the title.

NEXT PAGE

displays the subsequent page(s) of the directory. There can be up to 54 items per page of menu.

PREV MENU

displays the main menu for limit lines.

SAVE LINE

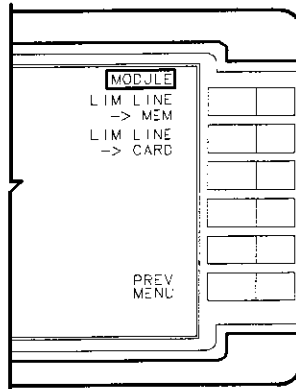
displays the Save-Line menu softkeys. You select where to store a limit line. See Figure 2-10. Limit lines are stored by their titles. The program generates and attaches the default title "LIMIT_LINE" if you do not create one. Remotely create or edit a title using the **TITLE** command. Select **DISPLAY**, **MORE**, then **SCREEN TITLE**, and enter your title.

Note

If remote mode is active and another limit line is stored without specifying a unique title, the program automatically gives the new limit line the same default title and overwrites the old limit line without warning. If you are saving a line under the default title using the front-panel keys, you are asked whether the old line should be overwritten. Titles you specify are overwritten if they are saved under the same name.

Note

If a file has been stored into memory using a 16-character title, the first 9 characters must be unique to avoid writing over an existing file. If a file name is longer than 9 characters or if lower-case letters are used, the file name will be converted, if possible, to an LIF compatible file name using only the first 9 characters.



DF 217

Figure 2-10. Save Line Menu Softkeys

LIM LINE →MEM

highlights as the program stores the table of limit-line data in the battery-backed RAM of the module.

LIM LINE →CARD

highlights as the program stores the table of limit-line data on a RAM card, if one is installed and not set to safe. (Option T01 does not include memory card capability.)

Note

Limit lines stored on memory cards using a mass-memory module with firmware datecode 910116 or later cannot be read into a module with an earlier firmware datecode (for example, 890524). Limit lines stored on cards using modules with firmware datecode 890524 or earlier can be read into a module with firmware datecode 910116 or later.

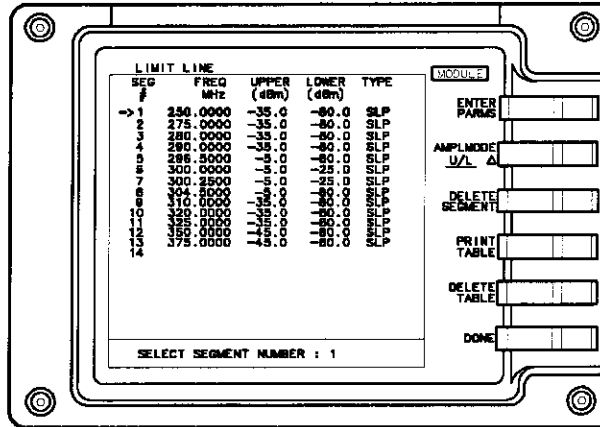
PREV MENU

returns you to the previous menu displaying the **SAVE LINE** softkey.

EDIT LINE

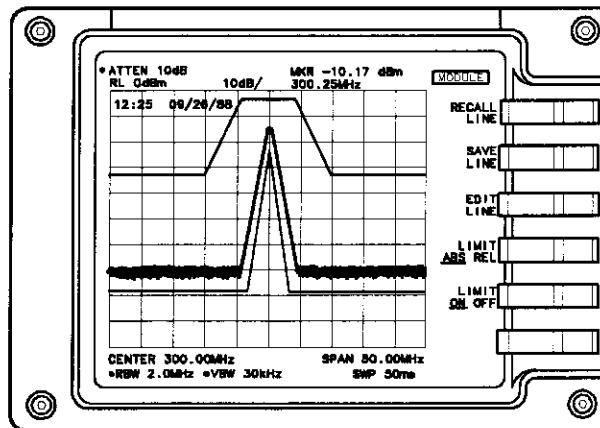
displays the table of limit-line segments and the menu of softkeys illustrated in Figure 2-11. The softkeys are explained below. Refer to Figure 2-12 for the screen display of the limit line. You can create up to 18 limit-line segments per table. Determine whether you want a relative or absolute limit line then begin creating or editing limit lines using the following steps:

1. Choose either upper/lower or middle/delta amplitude parameters.
2. Enter the parameter data into the table for each segment.
3. Select the segment type.
4. Name the limit line using the Display menu keys.
5. Return to the module Limit-line menu to save the limit line. The limit-line information you enter into the table is volatile; therefore, be sure to save the information with the **SAVE LINE** softkey.



DF218

Figure 2-11. Edit Line Menu Softkeys and Table Example

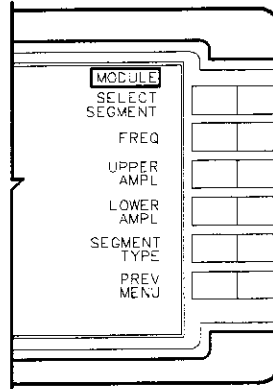


DF219

Figure 2-12. Sample Limit Line Using Figure 2-11 Table Data

ENTER PARMS

displays the softkeys you use to edit or create limit lines. To create a line, press **SELECT SEGMENT** followed by one of the other softkeys. Enter the parameter value using the RPG knob, STEP, or data keys. Use the spectrum analyzer **(BK SP)** key to correct errors. Press an appropriate units key to complete the data entry. Select the remaining parameters and enter their values by pressing the appropriate softkey and data keys. When all segments are defined, press **PREV MENU** and **DONE**, then, title the line using the **(DISPLAY)** function keys. Press **SAVE LINE** to store the limit-line parameters in a table. If you are editing the parameters of an existing limit line, you can select which segment you want to activate for editing. Press **SELECT SEGMENT**, then using the keypad or the RPG select the segment you want to edit. If using the keypad, follow the number with any units key. The cursor moves to the segment you have specified. Figure 2-13 illustrates the softkeys described below.



DF220

Figure 2-13. Enter Parameters Menu Softkeys

SELECT SEGMENT

allows you to choose a segment number. You may either create new parameters for another segment or edit the parameters of an existing segment. Locate the cursor at the segment number by using the RPG knob or step keys, or enter a numeric key corresponding to the segment number you want, followed by any units key. The cursor moves to the segment you have specified.

Note

The number of characters accepted for frequency values is to the nearest 100 Hz over the full range of spectrum analyzer frequencies. The number of characters accepted for amplitude values is to the nearest tenth of a dB over the full range of spectrum analyzer amplitudes.

FREQ

highlights and allows you to enter up to 12 characters (10 plus the decimal point and sign) for a frequency value using the data keys, RPG knob, or STEP keys. Terminate the data entry by pressing a units key, then any other parameter softkey to continue creating or editing a limit-line segment.

MID or UPPER AMPL

highlights and allows you to enter up to six characters (plus or minus sign, four digits, and a decimal point) for the upper- or mid-amplitude value using the data keys, RPG knob, or STEP keys. Terminate the data entry by pressing an appropriate units terminator key, then any other parameter softkey to continue creating or editing a limit-line segment.

DELTA or LOWER AMPL

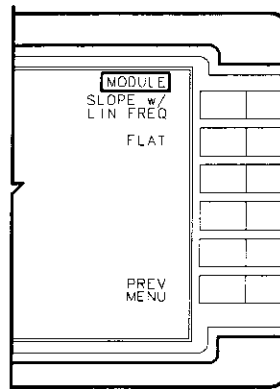
highlights and allows you to enter up to six characters (plus or minus sign, four digits, and a decimal point) for the lower or delta amplitude value using the data keys, RPG knob, or STEP keys. Terminate the data entry by pressing the appropriate units key, then any other parameter softkey to continue creating or editing a limit-line segment.

SEGMENT TYPE

accesses the **SLOPE w/LIN FREQ** (default setting) or **FLAT** segment type softkeys and **PREV MENU** softkey. See Figure 2-14. Choose a line type, or return to the previous menu.

Note

Limit-line data is sorted in frequency order in the limit-line table. The sorting occurs after you have pressed the **PREV MENU** key. If two data points are at the same frequency, they are sorted by the order in which they were entered.



DF 221

Figure 2-14. Segment Type Menu Softkey

SLOPE w/ LIN FREQ

draws lines by connecting all frequency/amplitude pairs in the order determined by the sorting operation previously explained. The data is displayed with a linear slope in frequency and a logarithmic slope in amplitude.

FLAT

draws a flat line, connecting frequency/amplitude pairs point to point until the line encounters another frequency whose amplitude is different, then the line is drawn vertically to the next amplitude value. The limit line is drawn without the logarithmic slope in amplitude.

PREV MENU

returns you to the previous menu displaying the **SEGMENT TYPE** softkey.

PREV MENU

returns you to the previous menu displaying the **ENTER PARAMS** softkey. Pressing this key also performs the sorting of the limit-line segments in frequency order in the limit-line table.

AMPL MODE U/L Δ

select upper/lower to enter amplitude values that display a limit line above or below an active trace. Use the delta (Δ) amplitude mode to create a dual limit line with an upper limit of MID plus DELTA, and a lower limit of MID minus DELTA. The delta should be positive.

DELETE SEGMENT

deletes the limit-line segment highlighted with the cursor. When the table is empty (no limit-line segments defined), only 1 appears under SEG #.

PRINT TABLE

This softkey label exists only if your spectrum analyzer has printer capability. Pressing this key prints the currently selected limit-line table on an HP-IB printer. The output format is tabular. Pressing the **PRINT** softkey prints the table only; softkey labels, graticule, or other characters and symbols are not printed. If the printer is disconnected or off, the print function must be repeated after you connect or turn the printer on.

Note

The limit-line table cannot be printed while the instrument is operating with Trace A in VIEW or BLANK mode. Pressing the **PRINT TABLE** key under these conditions will have no effect until the Module menu is exited, at which time the current trace and display will be printed.

DELETE TABLE

displays **CONFIRM DELETE** **CANCEL DELETE** softkeys. Select **CANCEL DELETE** if you do not want the contents of the table deleted. Select **CONFIRM DELETE** if you wish to delete all table contents.

DONE

restores the menu containing the **EDIT LINE** softkey.

LIMITS ABS REL

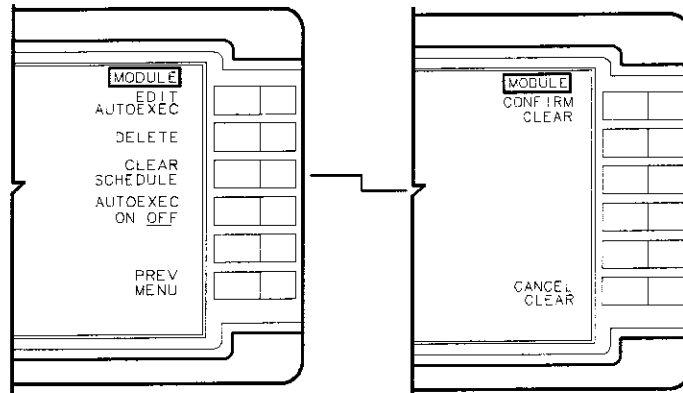
uses the current limit line as a reference for absolute frequency and amplitude values when the default **ABS** mode is activated. The **REL** setting causes the current limit-line value to be relative to the displayed center frequency and reference-level amplitude values.

LIMITS ON OFF

allows you to turn the currently selected limit-line function on or off. With limit lines set to ON, a limit-line test is made at the end of each sweep. If the active trace falls outside any point of the upper or lower limit, the message **LIMIT FAILED** is displayed on the screen. If limits are set to OFF, the active trace is not tested. Press the **MODULE** key on the spectrum analyzer to return to the module main menu.

AUTOEXEC Menu

Accessing this menu displays the AUTOEXEC/AUTOSAVE SCHEDULE. From here, you can define the criteria to execute DLPs automatically. Up to seven DLPs may be scheduled. These may be interleaved with the Autosave operations. The schedule can be revised in this menu by using the RPG knob or STEP keys and other softkeys for editing described below.

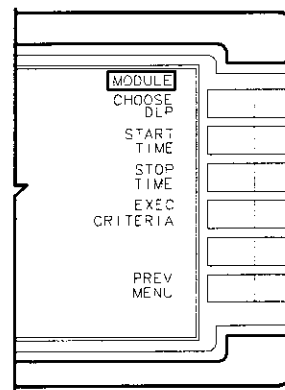


DF 222

Figure 2-15. Autoexec Menu Softkeys

EDIT AUTOEXEC

uses RPG knob or STEP keys to move cursor to the position, based on numeric hierarchy, in which you want the DLP scheduled. The softkeys of the menu illustrated in Figure 2-16 are described below.



DF 223

Figure 2-16. Edit Autoexec Softkeys

CHOOSE DLP

displays the DLP directory and suspends any running DLPs until you exit this menu. See Figure 2-17 for an example of the directory. If no DLPs are stored in memory, the message `No applicable entries` is displayed. Otherwise, use the `RPG` knob or `STEP` keys to move the cursor to the entry you wish to choose. Notice that there is a `NEXT PAGE` softkey. If your DLP is not listed in the first page of entries, check the subsequent pages. DLPs stored only on memory cards are not listed in the directory. (Option T01 does not include memory card capability.) They must be stored in the module RAM before they appear in the directory. Do this through the Utility menu. Press `UTILITY` and `CATALOG MEM CARD` until card is underlined. Move the cursor to the DLP you want to copy, press `COPY TO MEMORY`. Now return to the Autoexec menu, press `EDIT AUTOEXEC`, then `CHOOSE DLP`. With the cursor located by your selection, press one of the Choose DLP menu softkeys listed below.

Note

DLPs stored on memory cards using a mass-memory module with firmware datecode 910116 or later cannot be read into a module with an earlier firmware datecode (for example, 890524). DLPs stored on cards using modules with firmware datecode 890524 or earlier can be read into a module with firmware datecode 910116 or later.

SELECT

loads the DLP into one of the seven positions of the schedule. The position the DLP occurs in the schedule determines its priority in execution.

EXECUTE NOW

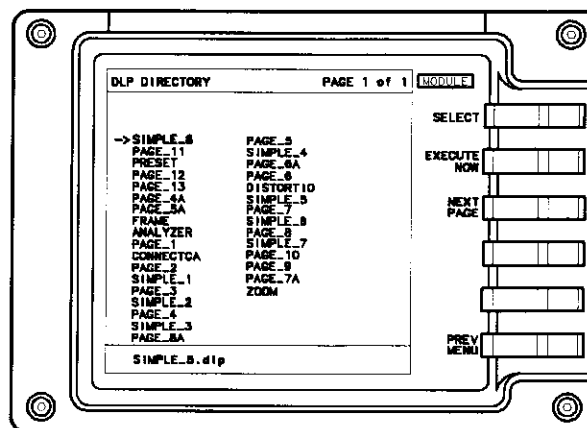
initiates the selected DLP immediately.

NEXT PAGE

accesses the subsequent page of the directory.

PREV MENU

returns you to the preceding menu.



DF 236

Figure 2-17. DLP Directory Example

START TIME

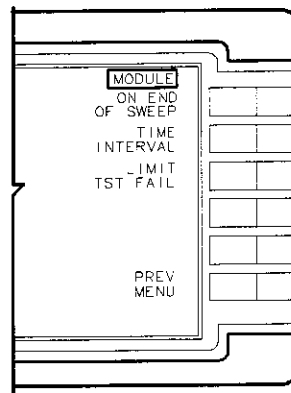
allows you to enter the date and time you want the Autoexecute function to begin. Enter the date and time in the format shown in the lower box of the schedule display. Any DLPs which were running when you selected **EDIT AUTOEXEC** are suspended during the editing process.

STOP TIME

allows you to enter the date and time you want the Autoexecute function to end. Enter the time and date in the format shown in the lower box of the schedule display. If the stop time is less than the start-time, the stop-time is left undefined.

EXEC CRITERIA

accesses the menu for you to define the criteria, or conditions, that will activate the Autoexec function. Execute criteria defaults to ON EOS (on end of sweep). Select one of the softkeys illustrated in Figure 2-18.



DF 225

Figure 2-18. Execute Criteria Softkeys

ON END OF SWEEP

executes the selected DLP at the end of every normal sweep. This is the default criteria if no other is selected. If the DLP operation is incomplete at the end of the sweep, the DLP completes before more data is evaluated.

TIME INTERVAL

displays HHMMSS in the lower box of the AUTOEXEC/AUTOSAVE SCHEDULE display. Use the data keys to enter a time interval in its displayed format. The program executes the selected DLP at the specified intervals. (If you enter an interval value of zero, the module defaults to the ON EOS setting.)

LIMIT TST FAIL

causes the module to execute a DLP when an active limit line is exceeded. The first step in setting up a limit-test fail function is to have an active signal with all the spectrum analyzer settings determined. Next, build or recall a limit line, select a start/stop time, then press **LIMIT TST FAIL**.

PREV MENU

returns you to the previous menu displaying the **EXEC CRITERIA** softkey. Press **PREV MENU** again to return to the menu displaying the **EDIT AUTOEXEC** softkey.

DELETE

deletes the function name you have selected using the RPG knob or STEP keys.

CLEAR SCHEDULE

displays the **CONFIRM CLEAR** **CANCEL CLEAR** softkeys. Select **CONFIRM CLEAR** if you wish to remove all entries from the AUTOEXEC/AUTOSAVE SCHEDULE. Otherwise, press **CANCEL CLEAR** to return to the previous menu.

AUTOEXEC ON OFF

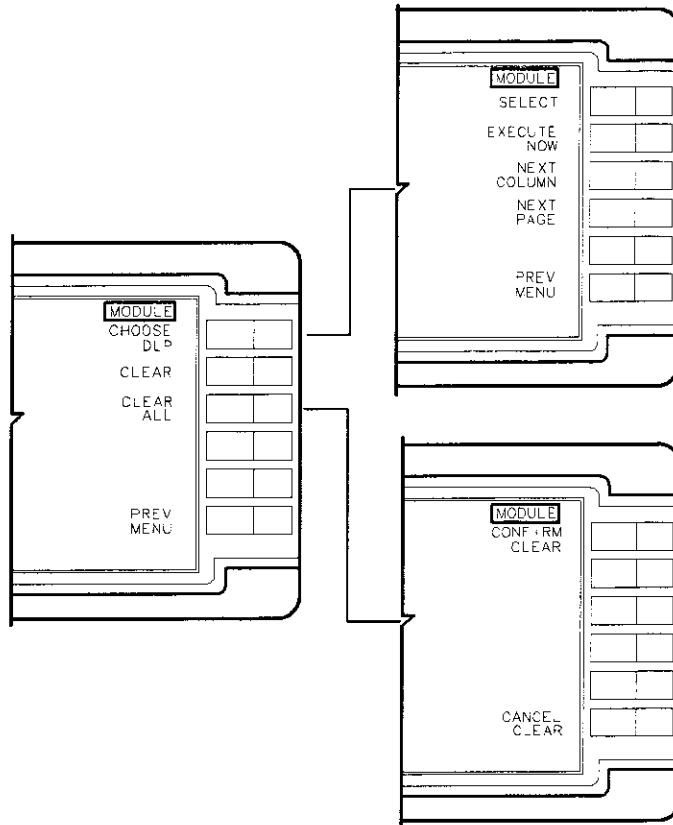
turns the Autoexec schedule on or off. While you are in this menu, no DLPs are allowed to execute. If AUTOEXEC is set to ON, DLP execution priority is determined by its location in the schedule. If a DLP is activated as soon as you turn Autoexec on, you will have to press **PRESET** on the spectrum analyzer to interrupt the DLP and regain control of any module or spectrum analyzer front panel keys.

PREV MENU

returns you to the previous menu displaying the **AUTOEXEC MENU** softkey, if there are no DLPs running. If a DLP is activated as soon as you turn Autoexec on, you will have to press **PRESET** on the spectrum analyzer to interrupt the DLP and regain control of any module or spectrum analyzer front panel keys. **PRESET** turns the Autoexec schedule off.

KEYDEF Menu

Press **KEYDEF** to display the *User Key Definitions* table. The 10 user-definable softkeys that can be labeled to activate DLPs stored in the module are listed here. Use the RPG knob or STEP keys to place the cursor beside one of the 10 user-definable softkeys in this menu. The DLP name you select from the module memory is loaded onto this softkey. To load a DLP stored on the memory card onto a user softkey, you must first copy it into the module memory. (Option T01 does not include memory card capability.) Do this remotely with the **CARDLOAD** command or locally using the Utility menu softkeys. Figure 2-19 illustrates the softkeys described below.



of 280

Figure 2-19. Keydef Menu Softkeys

CHOOSE DLP

displays the DLP directory. If no DLPs are stored in memory, the message **No applicable entries** is displayed. Otherwise, use the **RPG** knob or **STEP** keys to move the cursor to the entry you wish to choose. Notice that there is a **NEXT PAGE** softkey. If your DLP is not listed in the first page of entries, check the subsequent page. Either run it by pressing **EXECUTE NOW** to start the operation, or press **SELECT** to load it onto the selected key.

Note

DLPs stored on memory cards using a mass-memory module with firmware datecode 910116 or later cannot be read into a module with an earlier firmware datecode (for example, 890524). DLPs stored on cards using modules with firmware datecode 890524 or earlier can be read into a module with firmware datecode 910116 or later.

SELECT

assigns the DLP you have highlighted to a User Key menu key. Use the **RPG** knob or **STEP** keys to move to the DLP you want to assign to a key.

EXECUTE NOW

executes the highlighted DLP immediately.

NEXT PAGE

accesses the subsequent page of the directory.

PREV MENU

returns you to the previous menu displaying the **CHOOSE DLP** softkey.

CLEAR

clears the DLP name from the user-key you have highlighted. Move the cursor to the user-key number with the **RPG** knob or **STEP** keys.

CLEAR ALL

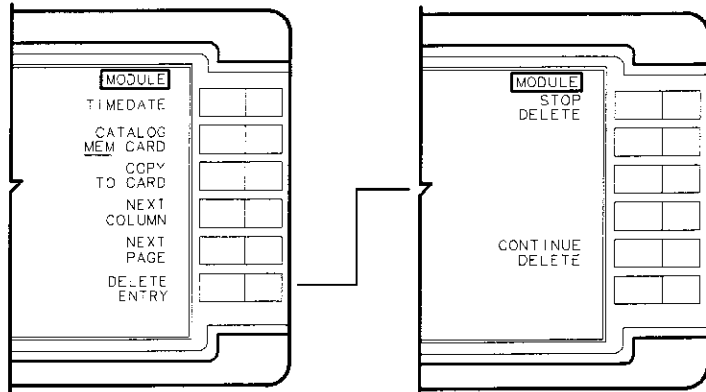
accesses the **CONFIRM CLEAR CANCEL CLEAR** softkeys. Press **CONFIRM CLEAR** to clear all the DLPs from the user keys. Otherwise, press **CANCEL CLEAR** to return to the previous menu displaying the **CLEAR ALL** softkey.

PREV MENU

returns you to the previous menu displaying the module main menu.

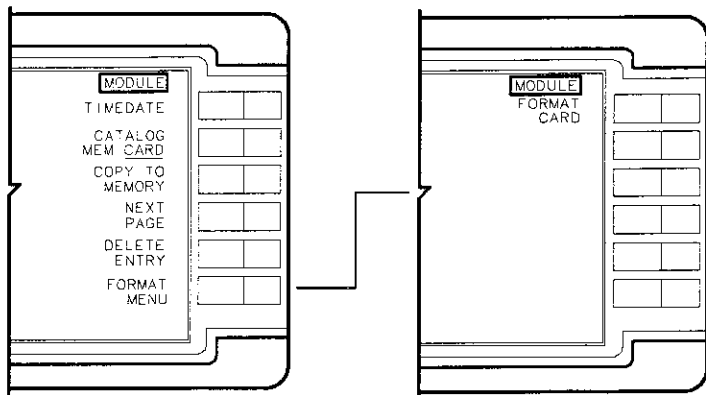
UTILITY Menu

Press **UTILITY** to gain access to the utility functions available with the HP 85620A Mass Memory Module. See Figure 2-20.



d129a

Figure 2-20. Utility Menu Softkeys (MEM selected)

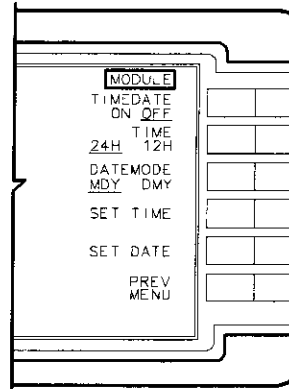


d1210a

Figure 2-21. Utility Menu Softkeys (CARD selected)

TIMEDATE

activates menu selections for setting the time and date. See Figure 2-22 for the softkeys defined below.



DF 226

Figure 2-22. Timedate Menu Softkeys

TIMEDATE ON OFF

displays the time and date in the active function block of the spectrum analyzer display when set to ON. The default setting is OFF. The date display format is a function of the **DATEMODE** softkey. The time display can be formatted for a 24-hour or 12-hour clock. The clock is updated every minute. Seconds are not displayed.

TIME 24H 12H

toggles the time display to either 24-hour (International) mode or 12-hour (AM/PM) mode. Press the **TIME** softkey until your preference is underlined. The default format is the 24-hour mode.

DATEMODE MDY DMY

toggles the format of the date displayed on the CRT. Select MDY (month, day, year) or DMY (day, month, year) by pressing the **DATEMODE MDY DMY** softkey until your preference is underlined. The default setting is MDY.

SET TIME

sets the real-time clock to the time you specify. Enter the 24-hour time in the HHMMSS format (the time display format is a function of the **TIME 24H 12H** softkey). Valid hour (HH) values are from 00 to 23. Valid minute (MM) and second (SS) values are from 00 to 59.

SET DATE

initializes the real-time clock to the date you specify. Enter the date in the MMDDYY or DDMMYY format, depending on the date format you have selected. The default format is MDY. Valid data entry for the month is MM (values 01 through 12), DD (values 01 through 31), and YY (values 00 through 99).

PREV MENU

returns you to the previous menu containing the **TIMEDATE** softkey.

CATALOG MEM CARD

toggles between the catalogs of the contents of the module battery-backed memory or contents of the memory card, if one is installed. If CARD is selected, the card format menu key is made available by pressing **FORMAT MENU** on later firmware revisions, or by pressing the blank softkey on earlier firmware revisions. (Option T01 does not include memory card capability.) If the card is missing, empty, or incorrectly installed the message **No applicable entries** is displayed in the **MEMORY CARD DIRECTORY**. Memory cards are LIF formatted (Logical Interchange Format). LIF entries are from 1 to 10 ASCII characters long. The module reserves one of the 10 characters for file encoding purposes, such as .LMT, .TRC, or .DLP. The user, therefore provides 1 to 9 ASCII characters for titling memory card entries. (Refer to the LIF Document, HP part number 5955-2676.) The default setting is MEM.

COPY TO MEMORY CARD

toggling the **CATALOG MEM CARD** key automatically toggles the copy key between memory and card to allow you to copy any of the directory entries to module memory or to a card. (Option T01 does not include memory card capability.) The card must be properly installed and not write-protected. Use the **RPG** knob or **STEP** keys to move the cursor to your selection. Press the **COPY TO MEMORY CARD** softkey to activate the copy function.

Note

DLPs, traces, and limit lines stored on memory cards using a mass-memory module with firmware datecode 910116 or later cannot be read into a module with an earlier firmware datecode (for example 890524). DLPs, traces, and limit lines stored on cards using modules with firmware datecode 890524 or earlier can be read into a module with firmware datecode 910116 or later.

NEXT PAGE

accesses any subsequent page(s) of the directory.

DELETE ENTRY

press this softkey after using the **RPG** knob or **STEP** keys to move the cursor to the entry you wish to delete. The **STOP DELETE** and **CONTINUE DELETE** softkeys are then displayed.

STOP DELETE

returns you to the directory.

CONTINUE DELETE

immediately deletes the highlighted entry.

FORMAT MENU (firmware revision 950829 and later)

accesses the card formatting key.

FORMAT CARD

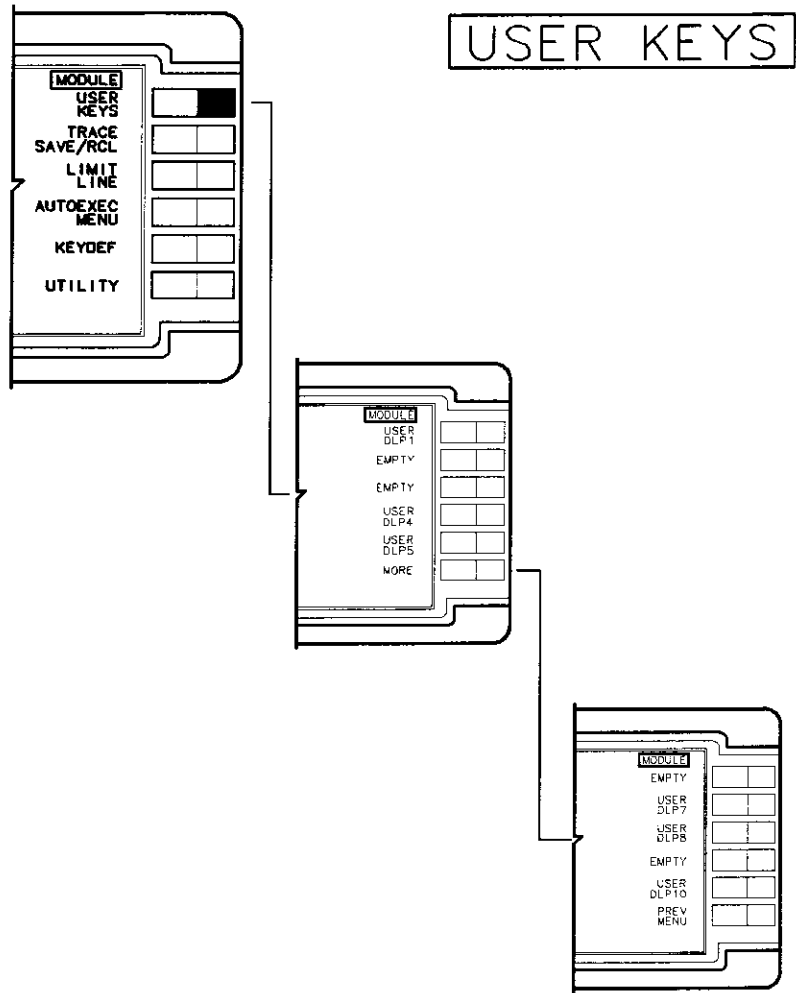
requests you to enter the number of directory entries. The number of entries **MUST** be a multiple of 8 (for example, 8, 16, 24, 32, 40, and so on). Optimum memory utilization is obtained when one directory per kilobyte of memory is selected. For example, enter 32 for a 32 kilobyte RAM card, or 128 for a 128 kilobyte RAM card.

Softkey Menu

Introduction

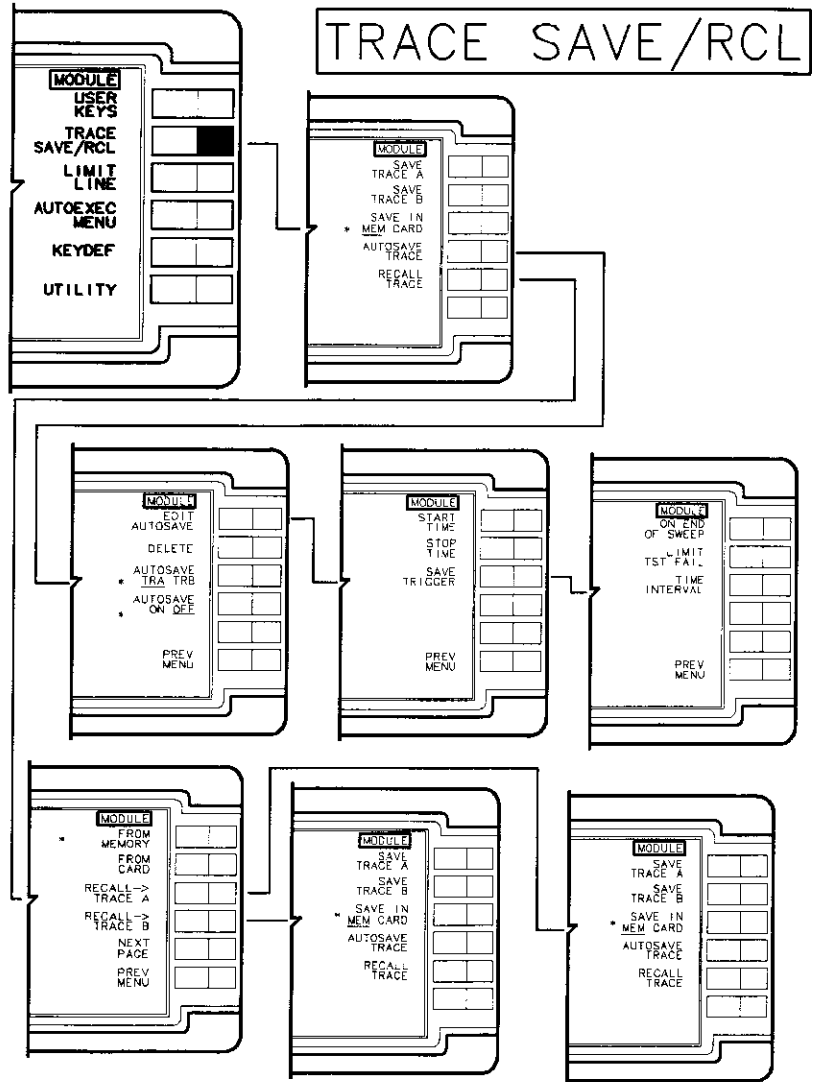
The menus of the HP 85620A Mass Memory Module are graphically represented in this chapter. The Main Menu keys are labeled in the upper right-hand corner of each grouping. Main Menu keys are listed below:

- USER KEYS
- TRACE SAVE/RCL
- LIMIT LINE
- AUTOEXEC MENU
- KEYDEF
- UTILITY



DF 229

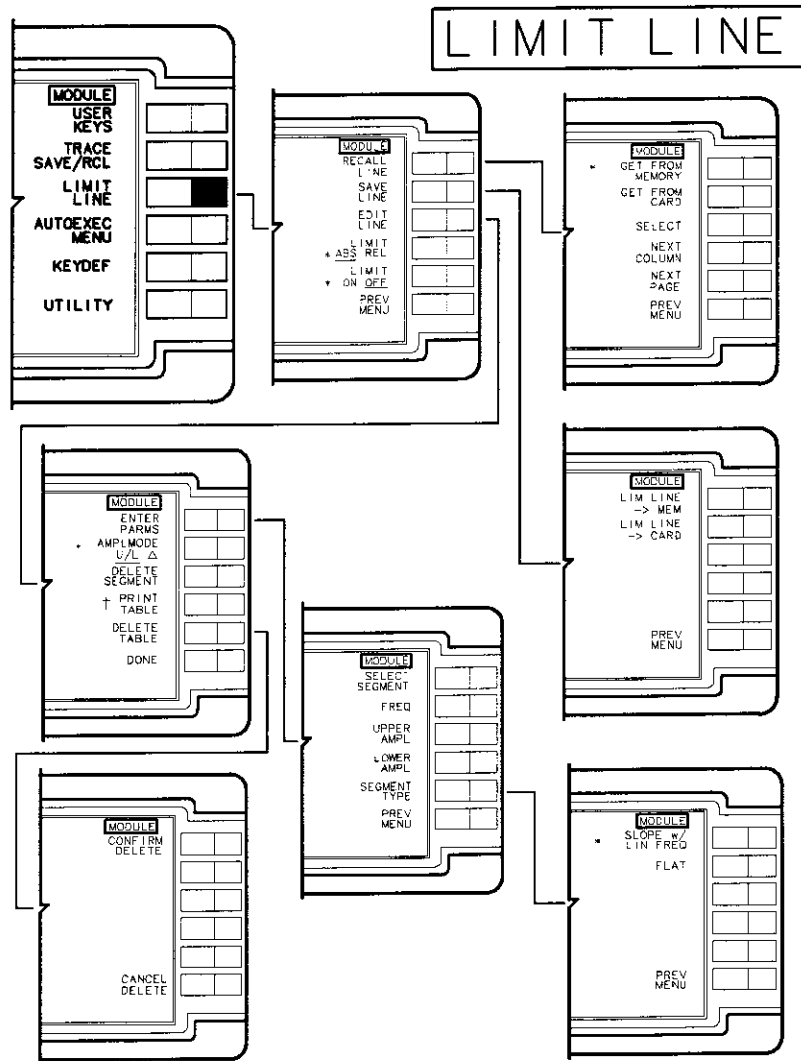
Figure 3-1. User Keys



nf22c

Figure 3-2. Trace Save/Rcl

* default

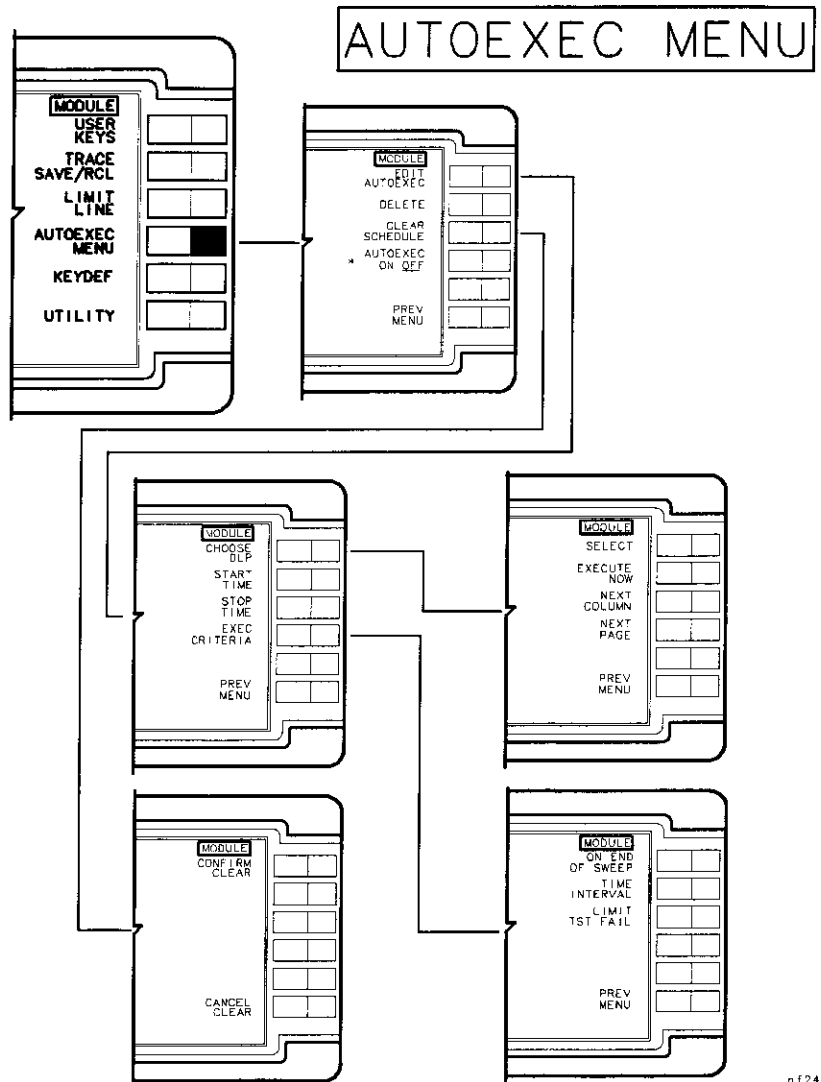


nf236

Figure 3-3. Limit Line

* default

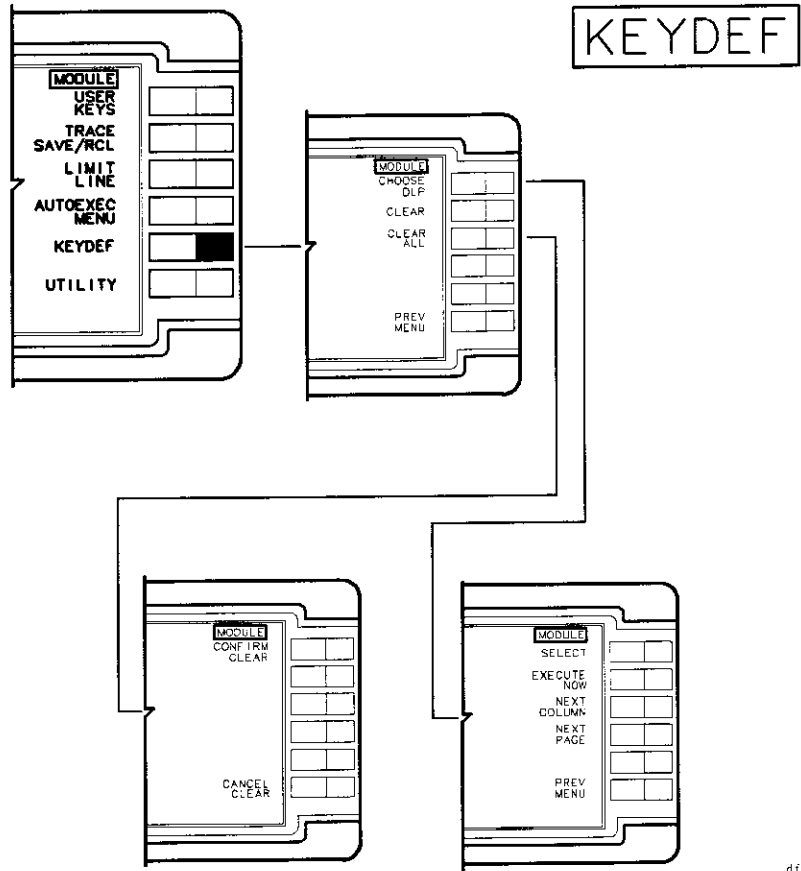
† This softkey label appears only if the spectrum analyzer has printing capability.



nf24a

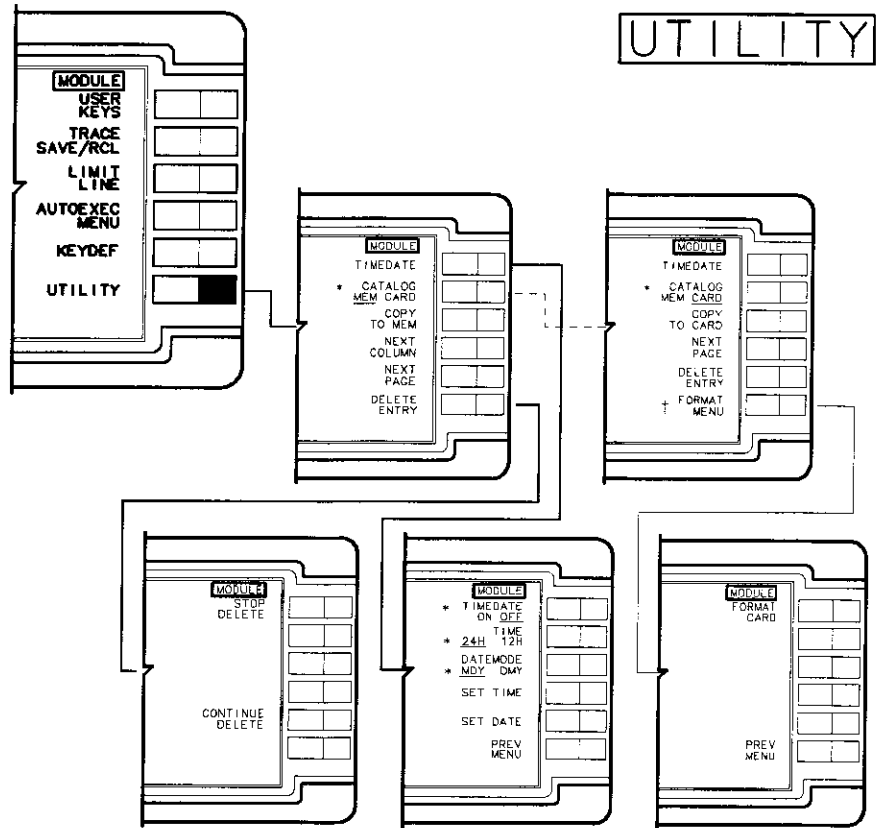
Figure 3-4. Autoexec Menu

* default



df24a

Figure 3-5. Keydef



n125a

Figure 3-6. Utility

- * Alternate functions appear when CARD is underlined in CATALOG MEM CARD.
- † This softkey is absent for mass memory modules having firmware revision 910116 and earlier.

Note

FORMAT MENU does not appear (blank softkey) for firmware revisions 910116 and earlier, but the FORMAT MENU functionality is still available by pressing the blank softkey.

Programming

Introduction

This chapter contains programming information for the HP 85620A Mass Memory Module. The section that follows provides some fundamental information about creating your own DLPs (downloadable programs). Also included are descriptions of syntax diagram terms along with syntax diagrams and some program examples.

Getting Started with DLPs

This section is an introduction to the concept of DLPs. It includes front-panel operation and information on how to program and download programs into the mass memory module from a controller or from a memory card. You will find out how to correctly declare functions, variables, and traces, then allocate these variables in spectrum analyzer memory. After these are declared and allocated, the module allows you to execute DLPs without an external controller. Special features such as limit lines and automatic execution (autoexec), are also discussed. This section assumes that you are familiar with HP 9000 Series 200/300 Controllers.

What Is a DLP?

A DLP is a single command or a sequence of commands used to perform a customer-specified operation. The user can define DLPs made up of several functions, variables, and traces, download them into the module RAM memory as one DLP, or define each command as a DLP. In the HP 85620A, there are one hundred twenty-eight kilobytes of RAM available for user DLPs. This memory is called the user-defined memory. Almost any instruction that can be executed over HP-IB can be executed in a DLP. In addition, DLPs have the ability to control other instruments over HP-IB. It is possible to design application programs and store them in user-defined memory as DLPs. These programs can then be executed without an external controller.

DLPs remain in memory even when the spectrum analyzer power is turned off. They are stored in the battery-backed RAM of the mass memory module and can be used repeatedly, whenever needed.

DLP Examples

Several programming examples are included in this section to illustrate the operation of DLPs both remotely and via user-defined softkeys. Examples showing commands unique to DLPs are also given. Refer to the list of required equipment and the equipment setup information to set your system up and begin running the program examples.

Required Equipment

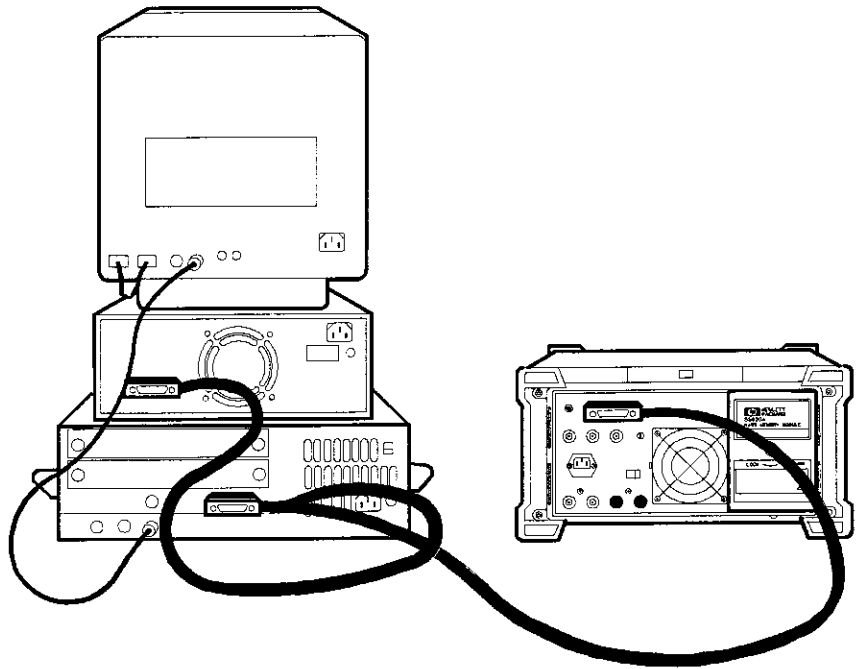
- HP 85620A Mass Memory Module
- HP 8560A/E, or HP 8561A/B/E, or HP 8562A/B/E, or HP 8563A/E Spectrum Analyzer
- HP Series 200/300 Controller
- 10833 A/B/C/D HP-IB Cable Assembly
- HP BASIC 2.0 (or greater)

Optional Equipment

- Any HP-IB Compatible Printer such as an HP 2225D ThinkJet

Equipment Setup

Connect the CAL OUTPUT to the RF INPUT of the spectrum analyzer. Set up the equipment as shown in Figure 4-1.



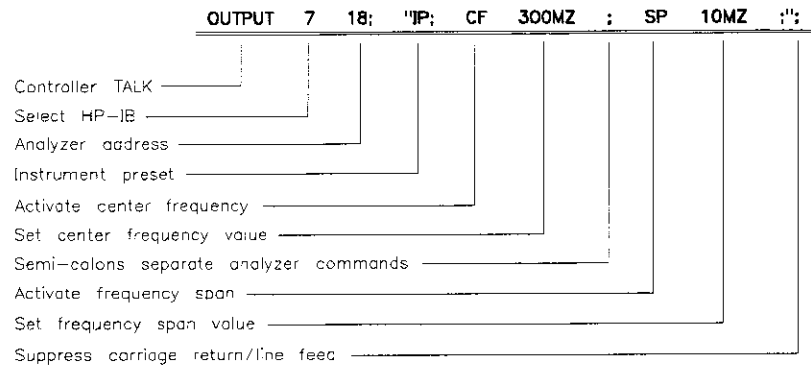
DF 235

Figure 4-1. Equipment Setup

Programming Front-Panel Functions

To prepare the spectrum analyzer for programming, use an external controller to enter the following literal example of the diagrammed command. It will preset the spectrum analyzer, set center frequency to 300 MHz, and span to 10 MHz.

Example 4-1 10 OUTPUT 718;"IP;CF 300MHZ;SP 10MHZ;";
 20 END



DF245

Figure 4-2. Preparing the Spectrum Analyzer for Programming

Executing the above program initiates the sequence of operations described above. If the CAL OUTPUT is connected to the RF INPUT, the 300 MHz calibrator signal should be displayed. The last function activated, SPAN, appears with its current value on the spectrum analyzer display.

It is important to note that the sequence of operations executed above may also be entered manually, from the front panel, to yield the same result. In fact, a manual sequence of keystrokes is usually developed first, then used as a basis for executing the same procedure under program control. This simple technique is recommended as a tool in developing software for automatic spectrum analyzers.

The semicolon (;) at the end of command lines serves to suppress the carriage-return/line-feed, and therefore conserves user memory.

Programming User-Defined Functions

Example 4-2 demonstrates how to make a DLP from the commands used in Example 4-1.

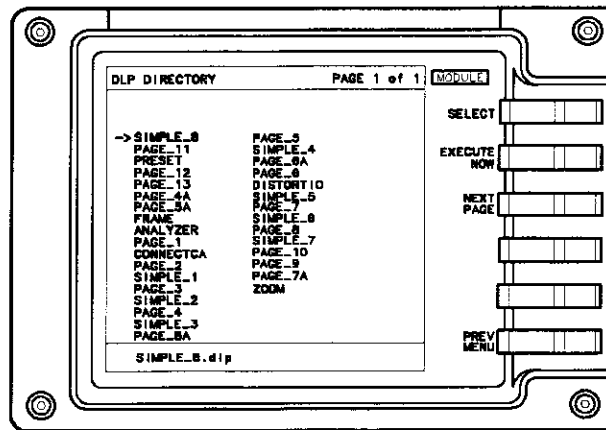
Example 4-2

```
10 OUTPUT 718;"FUNCDEF ZOOM, @;IP;CF 300MHZ;SP 10MHZ;@";
                                     !Assign the
                                     !label "ZOOM"
20                                     !to the indicated
30                                     !sequence of commands.
40 END
```

The FUNCDEF command specifies a user-defined function. The “at” sign (@) is used as a delimiter for the user-defined function “ZOOM”. All characters between the “at” signs are part of the “ZOOM” routine.

Refer to the “Remote Programming” section in this chapter for further discussion of programming delimiters. Downloading the program in Example 4-2 from an external controller to the spectrum analyzer stores the routine in the mass memory module internal battery-backed RAM. The routine can now be executed without an external controller from the spectrum analyzer front panel. Press LCL, (MODULE), AUTOEXEC MENU, EDIT AUTOEXEC, and CHOOSE DLP. Locate the cursor at the DLP you want to execute, press EXECUTE NOW.

A typical listing of DLPs stored in a mass memory module is shown in Figure 4-3.



DF 236

Figure 4-3. Typical DLP Listing

Programming User-Defined Functions

For frequently used DLPs, it may be more convenient to load the program onto one of the user softkeys in the User Keys menu. Any DLP can occupy one of these ten user-definable softkeys. The DLP, once assigned to a softkey, can be executed independently without a controller by pressing the appropriate softkey in the User Keys menu. To assign a DLP to a softkey, the KEYDEF command must be invoked. The following program stores the DLP named "ZOOM" under softkey number 1 in the User Keys menu. Notice that the percent sign is used to delimit the title "ZOOM".

Example 4-3

```
10 OUTPUT 718;"FUNCDEF ZOOM,@;IP;CF 300MHZ;SP 10MHZ;@";
                                ! Assigns the label
20                                ! ZOOM to the indicated
30                                ! sequence of commands.
40 OUTPUT 718;"KEYDEF 1, ZOOM,%ZOOM%";    ! Assigns the label "ZOOM"
50                                ! to user key 1.
60 END
```

To execute this DLP, press **USER KEYS**. "ZOOM" should be labeled on softkey number 1. Press the **ZOOM** softkey. Labels for user-defined softkeys can be from 1 to 16 characters long. You can manually assign DLPs to a User Keys softkey from the Keydef menu found in the main menu of the module. Refer to the Operation Chapter in this manual for details about the Keydef menu.

Note

The user-defined softkeys do not need to be labeled with the same label used to define the DLPs they will activate.

Determining Available User Memory

Since user memory is limited, it may sometimes be valuable to determine the amount of memory a DLP requires. Doing this is a simple four-step process that requires the use of the MEM command.

1. Enter the command OUTPUT 718;"DISPOSE ZOOM;"", then run Example 4-4 before the DLP is downloaded to verify available memory in the mass memory module.

Example 4-4

```
10 OUTPUT 718;"MEM?";           ! Determine available user-memory
20                               ! value and prepare to output value.
30 ENTER 718;M                  ! Store the available user-memory
40                               ! value in the variable "M".
50 PRINT "MEMORY = ";M;"BYTES" ! Print the user-memory value.
60 END
```

2. Run Example 4-3 to download the entire DLP and allocate the required memory.
3. Run the program in step 1 again to determine the new value of available memory in the mass memory module.
4. Calculate the user memory required for the DLP by subtracting the value of memory found in step 1 from the value found in step 3.

Note

The mass memory module has 128-kilobytes of memory. This corresponds to 131,072 bytes (1024 bytes/kilobyte × 128 kilobytes).

User-Defined Variable and Trace Declaration

To store a single value in the mass memory module RAM, a variable must be defined and set to an initial value. This allocates space in internal RAM for the variable name and value. The following example shows how to declare a variable and how the amplitude of a signal can be stored in a variable.

Example 4-5

```

10 OUTPUT 718;"VARDEF AMPLITUDE, 0;"; ! Define variable, "AMPLITUDE
20                                     ! and initialize its value to zero.
30 OUTPUT 718;"IP;SNGLS;";           ! Instrument preset, single sweep.
40 OUTPUT 718;"FA 275MHZ;FB 325MHZ;"; ! Set the start and stop frequencies
50                                     ! to 275 and 325 MHz respectively.
60 OUTPUT 718;"TS;MKPK HI;";         ! Put a marker on the largest
70                                     ! signal on the trace.
80 OUTPUT 718;"MOV AMPLITUDE,MKA;";   ! Move the marker amplitude
90                                     ! value into "AMPLITUDE".
100 END

```

The programming example above alters the variable "AMPLITUDE" that was defined using the VARDEF command. Other commands like MOV that can be used to alter variable values are the math commands ADD, DIV, and MPY. Refer to these commands in the "Syntax Diagrams, Descriptions, and Program Examples" section of this chapter.

To store a trace in the module RAM, define a trace array within a DLP using the TRDEF command. TRDEF allocates a specified amount of memory to a specified trace name.

Example 4-6

```

10 OUTPUT 718;"TRDEF EXAMPLETRACE,601;"; ! Define a 601-point array
20                                     ! labeled "EXAMPLETRACE".
30 OUTPUT 718;"MOV EXAMPLETRACE,0;";    ! Initialize trace values to zero.
40 END

```

The programming commands for a user-defined trace, function, or variable name may consist of 1 to 16 capital letters.

It is important to realize that VARDEF and TRDEF are global variables. This means that variables and traces retain their values until redefined, disposed of, or altered by MOV or math commands. Each time the variable or trace is altered, the new value writes over the old value in memory. Potential problems can be avoided by defining variables and traces at the beginning of a program. If variables and traces are defined at the beginning of a program, then the MEM? query returns the correct value for available memory after the DLP is downloaded by an external controller.

Displaying Variable and Trace Values

The value of a variable in user memory can be displayed on the spectrum analyzer screen using the DSPLY command. The DSPLY command requires that two numbers be specified after the variable name. The first number indicates the total field width, or the total amount of numbers to be displayed. The second number specifies the resolution, or the amount of numbers to be displayed after the decimal point. Those numbers should be separated by a comma.

The variable may also be displayed at a particular location on the spectrum analyzer screen by specifying the pen location. Refer to the "Graphics Operation Commands" in Table 4-1 for command mnemonics that perform this operation. Then locate the actual command alphabetically listed in this chapter for a description.

User-defined traces that are stored in memory can be displayed by using the MOV command. The user-defined trace must first be moved into either trace A or B, then it can be displayed. The following example demonstrates how to move the trace EXAMPLETRACE (previously defined in Example 4-6) so it can be displayed in trace A.

```
OUTPUT 718;"MOV TRA, EXAMPLETRACE;"
```


Using EP to Modify User-Defined Variables (firmware revisions 910116 and later)

The secondary keyword EP can be used within a DLP to modify values of user-defined variables (e.g. variables defined using LCLVAR or VARDEF). This keyword can be used in a DLP to pause the program, allowing the user to input a value for the variable, and then continue. This is a simple alternative to using the ACTVFUNC command and can be very helpful in debugging a DLP.

When EP is used with a user-defined variable and the data entry keys are used, the terminators have the following effect on entered values:

GHz/+ dBm/dB	Multiplies entered value by 1×10^9
MHz/-dBm/sec	Multiplies entered value by 1×10^6
kHz/mV/ms	Multiplies entered value by 1×10^3
Hz/uV/us/ENTER	Multiplies entered value by 1

These multipliers are convenient when entering frequency values, but care should be taken when entering amplitude values, especially in dB.

For proper operation, the display annotation should be turned on and at least one trace must be in either CLEAR/WRITE or MAX HOLD. When using EP, display a prompt in the first line of the active function block, as shown in the example below. A question mark will appear on the next line, and will be overwritten as data is entered. If a prompt is not displayed, the first line of the most current active function will appear with a question mark on the next line.

Program Example:

The following example could be used to setup for a total harmonic distortion measurement. The user will enter the fundamental frequency of the input signal into the variable E_NTRY. Line 70 prompts the user to enter the frequency. The HD command clears the active function block before writing the prompt text.

```

10  !
20  ASSIGN @Sa TO 718
30  !
40  OUTPUT @Sa;"FUNCDEF H_ARMDIST,@;";
50  OUTPUT @Sa;"LCLVAR E_NTRY,0;";
60  OUTPUT @Sa;"IP;SNGLS;TS;";
70  OUTPUT @Sa;"HD;PU;PA 100,575;TEXT/ENTER FUNDAMENTAL FREQUENCY/;";
80  OUTPUT @Sa;"E_NTRY EP;";
90  OUTPUT @Sa;"EM;";
100 OUTPUT @Sa;"DIV FA,E_NTRY,2;";
110 OUTPUT @Sa;"MPY FB,E_NTRY,3.5;";
120 OUTPUT @Sa;"TS;MKPK HI;MKD;MKPK NH;";
130 OUTPUT @Sa;"@";
140 !
150 OUTPUT @Sa;"KEYDEF 1,H_ARMDIST,/      THD      SETUP/;";
160 !
170 ASSIGN @Sa TO *
180 END

```

Erasing User-Defined Memory

Use the DISPOSE command with a controller to remove previously stored contents (variables, traces, DLPs and limit lines) from the mass memory module RAM. You may remove any or all of the contents from memory. Refer to the DISPOSE command in this chapter for additional information. The following examples may be used to erase user-defined memory.

The following command line removes an individual variable called "ZOOM." Replace "ZOOM" with other variables, functions, and traces to remove them from memory.

```
OUTPUT 718;"DISPOSE ZOOM;"
```

Remove all contents from user-defined memory. This includes DLPs, traces, limit lines, and variables:

```
OUTPUT 718;"DISPOSE ALL;"
```

Any or all of user-defined memory contents can be removed using the softkeys of the module. Remove individual user-defined memory contents by pressing **MODULE**, **UTILITY**. Press **CATALOG MEM CARD** until your memory location preference is underlined, then move the cursor to the entry you wish to delete. Press **DELETE ENTRY**, then **CONTINUE DELETE**.

Clear a user-defined softkey by pressing **MODULE**, **KEYDEF**, move the cursor to the key label you wish to clear, then press **CLEAR**.

Storing DLPs on RAM Cards and in the Module

Downloading DLPs From Memory Cards to the Module

To download a DLP from a memory card, correctly insert the memory card into the module as described in Chapter 1, "Installation." Press **MODULE**, **UTILITY**, then press **CATALOG MEM CARD** until **CARD** is underlined. Use the RPG knob or **STEP** keys to move the cursor to the DLP you want to copy. Press **COPY TO MEMORY**.

Storing DLPs on Memory Cards

To copy a DLP from the module RAM to a memory card, first install a RAM card into the module. Be sure the write-protect switch is not set to **SAFE** (write-protected). Refer to Chapter 1, "Installation." Press **MODULE**, **UTILITY**, then press **CATALOG MEM CARD** until **MEM** is underlined. Use the RPG knob or **STEP** keys to move the cursor to the DLP you want to copy. Press **COPY TO CARD**.

Note

DLPs stored on memory cards using a mass-memory module with firmware datecode 910116 or later cannot be read into a module with an earlier firmware datecode (for example, 890524). DLPs stored on cards using modules with firmware datecode 890524 or earlier can be read into a module with firmware datecode 910116 or later.

Programming Hints

Making a program easy to read also makes it easier to debug and document. Here are a few simple rules to follow which help make any program, including a DLP, more readable.

1. Write short program lines.
2. Use standard indent format for looping, branching, and subroutines.
3. Use descriptive variable names and labels.
4. Document program lines clearly.

In addition to the general readability rules above, here are some other procedures which apply specifically to DLPs to make them more readable and less prone to error.

1. Define all variables with VARDEF, traces with TRDEF, and arrays with ARRAYDEF at the beginning of the program *NOT* within a FUNCDEF. Use LCLVAR to define local variables within DLPs. For more information about LCLVAR, refer to its description in the Syntax Diagram, Descriptions, and Program Example section.
2. Use unique names (unique from command language) for all user-defined function, variable, trace, and array labels.
3. Use semicolons between commands. IEEE Standard 728 recommends this use of semicolons to avoid possible misinterpretation by the spectrum analyzer.

If rule number 2 is not followed, an error results and the definition process is aborted. Commands are reserved names and must not be used.

The program in Example 4-7 below demonstrates the concepts described above. It checks to see if there are any signals on the spectrum analyzer screen that are above -60 dBm. If there are, the spectrum analyzer zooms in on the signal to a 100 kHz span and saves that trace in the module battery-backed memory.

The “@” symbols, appearing in lines 60 and 170, delimit the function definition. All commands appearing between delimiters are assigned to the function label, CHECK.

Example 4-7

```

10 OUTPUT 718;"VARDEF POWER,0;";           ! Define a variable named POWER
20                                           ! and initialized it to zero.
30 OUTPUT 718;"TRDEF SAVE, 601;";         ! Define a 601-point trace
40                                           ! named SAVE.
50 !
60 OUTPUT 718;"FUNCDEF CHECK,@;";         ! Define a function named "CHECK".
70 OUTPUT 718;"IP;";                       ! Perform instrument preset.
80 OUTPUT 718;"SNGLS;CF 600MHZ;SP 1GZ;TS;";
90                                           ! Set to single sweep, set the
100                                          ! center frequency to 600 MHz,
110 OUTPUT 718;"MKPK HI;TS";              ! Place the marker on the highest
120                                          ! peak, then take a sweep.
130 OUTPUT 718;"IF MA,GT,-60 THEN;";       ! If a peak is greater than -60 dBm,
140 OUTPUT 718;"  SP 1KZ;TS;TS;";         ! then set span to 100 kHz.
145 OUTPUT 718;"  MKRL;TS;";              ! Move marker to reference level
150 OUTPUT 718;"  MOV POWER, MKA;";       ! Move the marker amplitude
160                                          ! to the previously defined variable
                                          ! POWER.
170 OUTPUT 718;"  MOV SAVE, TRA;";         ! Move Trace A into previously defined
180                                          ! TRDEF named SAVE
190 OUTPUT 718;"ENDIF;";                   ! End the IF statement.
200 OUTPUT 718;"@";                       ! End the definition of CHECK.
210 END

```

Modularity The preceding example, 4-7, is a DLP formatted in a modular style. This style offers four distinct advantages:

- easy to read
- easy to change
- easy to debug
- easy to document

Before beginning Example 4-8, manually set the spectrum analyzer to the correct span. To automate the operation completely, we can add command lines to set the span automatically. The following DLP steps through four predefined spans to find a signal higher than -60 dBm. If no signal is found in the first span, it steps to the next higher span. When a signal is found, the DLP zooms in on the signal, stores the signal, and records its amplitude. If a signal is found in any of the four spans, the DLP halts execution and displays the last signal found.

Programming Hints

Example 4-8

```
10      ! File name: EXAMPLE
20      ! Date: 9/1/88 Author: Jane Doe
30      ! Description of the program: This program checks for signals above
40      ! -50 dBm in the following frequency
50      ! spans: 10 - 12 MHz, 12 - 14 MHz,
60      ! 14 - 16 MHz, and 16 - 110 MHz. If a
70      ! signal is found, it "autozooms" to
80      ! 1 MHz span, records the signal
90      ! level, and displays the highest frequency
100     ! signal found in trace B.
110 OUTPUT 718;"VARDEF POWER,0;";           ! Define a variable named "POWER"
120                                           ! and initialize it to zero.
130 OUTPUT 718;"TRDEF SAVE, 601;";         ! Define a 601-point trace, "SAVE".
140           ! Subroutines:
150 OUTPUT 718;"FUNCDEF SPANONE,@;";       ! Define function SPANONE
170 OUTPUT 718;"FA 10MHZ;FB 12MHZ;";     ! Set the start and stop frequencies
180 OUTPUT 718;"@";
190 !
200 OUTPUT 718;"FUNCDEF SPANTWO,@;";      ! Define function SPANTWO.
210 OUTPUT 718;"FA 12MHZ;FB 14MHZ;";
220 OUTPUT 718;"@";
230 OUTPUT 718;"FUNCDEF SPANTHREE,@;";
240 OUTPUT 718;"FA 14MHZ;FB 16MHZ;";
250 OUTPUT 718;"@";
260 OUTPUT 718;"FUNCDEF SPANFOUR,@;";
270 OUTPUT 718;"FA 16MHZ;FB 110MHZ;";
280 OUTPUT 718;"@";
290 OUTPUT 718;"FUNCDEF CHECK,@;";        ! Define function named CHECK.
300 OUTPUT 718;"TS;MKPK HI;";            ! Place a marker on the highest signal.
310 OUTPUT 718;"IF MA,GT,-50 THEN";      ! If there is a signal higher
320 OUTPUT 718;" MKTRACK ON;";          ! than -50 dBm, zoom to 1 MHz
330 OUTPUT 718;" SP 1MHZ;";             ! span, center it and bring it
340 OUTPUT 718;" MKTRACK OFF;TS;";      ! to the reference level.
350 OUTPUT 718;" MKPK HI;MKCF;TS;";     ! Store it in a 601-point
360 OUTPUT 718;" MKRL;TS;";             ! trace previously defined as
370 OUTPUT 718;" MOV POWER, MKA;";      ! having the label, "SAVE".
380 OUTPUT 718;" MOV SAVE, TRA;";       ! Save the control settings
390 OUTPUT 718;" SAVES 1;";             ! in register 1.
400 OUTPUT 718;"ENDIF;";               ! End the IF statement.
410 OUTPUT 718;"@";                    ! End the definition of "CHECK".
420                                     ! Main Program
430 OUTPUT 718;"FUNCDEF EXAMPLE,@;";     ! Label the main program EXAMPLE.
440 OUTPUT 718;"IP;SNGLS;MOV SAVE,0;";   ! Place the analyzer in single-sweep
445                                     ! and set all values in "SAME" to zero.
450 OUTPUT 718;" REPEAT";
460 OUTPUT 718;" SPANONE;CHECK;";
470 OUTPUT 718;" SPANTWO;CHECK;";       ! Check each span for a
480 OUTPUT 718;" SPANTHREE;CHECK;";     ! signal greater than -50 dBm.
490 OUTPUT 718;" SPANFOUR;CHECK;";     ! Repeat sequence until
500 OUTPUT 718;" UNTIL SAVE[0],NE,0;";  ! a non-zero value is found in
510 OUTPUT 718;" MOV TRB,SAVE;";        ! "SAVE". It then displays the located
520 OUTPUT 718;"RCLS 1;BLANK TRA;VIEW TRB;";
                                           ! signal in trace B, and
```

```

530 OUTPUT 718;"@";
540                                     ! recalls the analyzer settings
550                                     ! that existed when the signal
550                                     ! was found. "EXAMPLE" is assigned
560 OUTPUT 718;"KEYDEF 2,EXAMPLE,%EXAMPLE%";
                                     ! to User Keys menu softkey 2 so the
570 END                                 ! program may be executed from
580                                     ! the front panel.

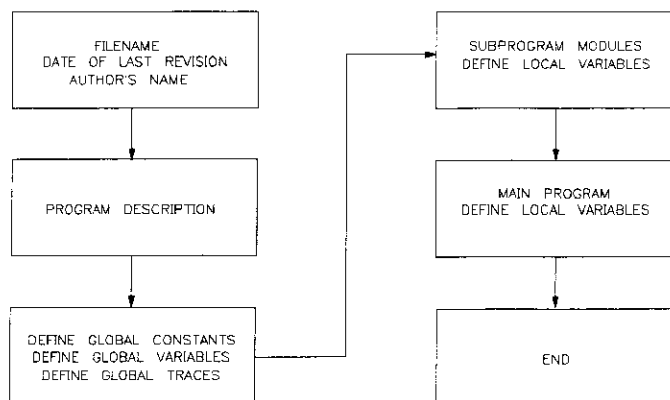
```

Notice that four subroutines have been added (SPANONE, SPANTWO, and so on). Each subroutine sets the spectrum analyzer to a different frequency range. The lines from 150 through 410 now become a subprogram. Each of the five subroutines is called from the main program, Example. Line 560 enables access to the DLP stored in a **USER KEYS** softkey labeled **EXAMPLE**.

The DLP in Example 4-8 uses descriptive labels and flows in a logical fashion, making it readable. In addition, it is easy to modify. For example, if the application requires the stop frequency of the last span to extend to 4 GHz, simply change FB 110MHZ in SPANFOUR to FB 4GHZ.

Program Structure

It is important for the DLP to follow a logical, structured order. The program structure illustrated in Figure 4-4 is highly recommended for making all downloadable programs easy to read and easy to debug.



DF237

Figure 4-4. Program Structure Flowchart

Looping and Branching

Looping and branching are the main types of program-flow control. The REPEAT/UNTIL command is used for program looping and the IF/THEN/ELSE/ENDIF statement is used for program branching.

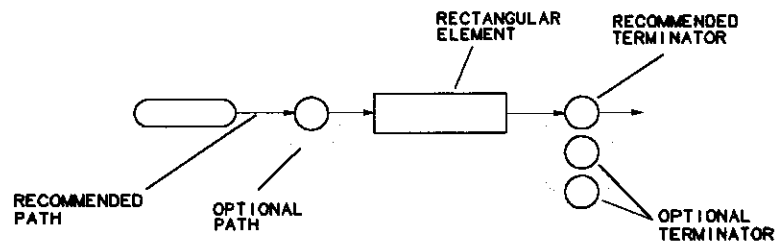
Programming Hints

Debugging

More often than not, new programs require debugging. In DLPs, bugs may appear in any of the following ways:

- As an error message displayed on the spectrum analyzer screen.
- The DLP does the unexpected. For example, it halts execution, or enters an infinite loop, or starts executing before its start-execution command occurs.
- As an unexpected or out-of-range result or value is obtained.

Command Syntax Diagrams



DF 238

Syntax Format

Syntax Diagram Notation

circles or ovals	enclose literals which must be entered exactly as shown (except for SP, ASCII code 32, which creates a space). Literals enclosed by ovals are printed in bold, capital letters in this manual. Literals enclosed by circles include the items listed below:
CR	carriage return (ASCII code 13)
LF	line-feed (ASCII code 10).
SP	space (ASCII code 32), used to separate parameters.
,	comma (ASCII code 44), used to separate parameters.
;	semi-colon (ASCII code 59), used to separate and terminate commands. LF, CR, SP, and comma (,) are allowed, but not recommended.
solid lines	represent the recommended command path. Each line can be followed in only one direction, as indicated by an arrow at the end of each line. Any combination of items generated by following the lines in the indicated direction is syntactically correct.
dotted lines	represent optional paths. These paths are not recommended.
curved intersections	clarify the flow of command path direction.
rectangles	contain a parameter used in the command sequence. A description of each parameter is provided with its command description.

Syntax Diagram Parameters

Parameters or elements contained in rectangular boxes of syntax diagrams may be any of the following:

analyzer command

any spectrum analyzer command.

array element any point of a user-defined array that describes the array being accessed. When an array of a greater length is operated upon and stored in an array of

Command Syntax Diagrams

lesser length, the array is truncated to fit. When an array of a lesser length is operated upon and stored in a trace of greater length, the last array element is extended for operations with the greater length.

array range any segment or point of a user-defined array that specifies the array limits.

character Sp ! " " # \$ % & [\] ? () + , _ / 0 1 2 3 4 5 6 7 8 9 : ;
A B C D E F G H I J K L M N O P Q R S T U V W X Y
Z a b c d e f g h i j k l m n o p q r s t u v w x y z

compatible function any spectrum-analyzer command in this chapter that performs an action on another function and contains predefined function in the syntax diagram. Refer to predefined function in this section. Some of the compatible functions are listed below:

AVG EXP
DIV IF/THEN/ELSE
DSPLY KEYDEF
ENTER

data byte 8-bit byte, containing numeric or character data.

identifier ASCII string composed of from 1 to 16 characters. Alpha character strings require an underscore to separate the alpha characters from subsequent numeric characters that are embedded into the identifier. As an example ZOOM_2 is an acceptable identifier, but ZOOM2 or ZOOM 2 is not.

delimiter matching characters marking the beginning and end of character strings, user-defined functions, or pre-defined functions. They include these characters:

! " \$ % & ' / \ : @ ~

digit 0 1 2 3 4 5 6 7 8 9

DLP a user-defined function.

LF with EOI line feed with end-or-identify function. ASCII code 10 (LF) is sent via HP-IB, then the end-or-identify control line on HP-IB sets to indicate the transmission is complete.

measurement units apply to TRA and TRB and range from 0 to 600 points.

number integer or real numerical data. Integers are numbers having no fractional part. *Integer range* is $-32,768$ through $+32,767$.
 Real numbers include integers and all other numbers. The range for *reals* is $-1.790E^{+308}$ through $-2.225E^{-308}$, 0, and $+2.225E^{-308}$ through $+1.798E^{+308}$.
 The *byte range* is 0 through 255.

predefined function math functions that return a value. The following list contains some predefined function commands. They are predefined functions when they do not end with a question mark (?). If they end with a question mark, they are queries.

MEAN	STDEV
PEAKS	SUM
PWRBW	SUMSQR
RMS	VARIANCE

predefined variable functions that include variable data, usually numeric data. Some of these are included in the following list:

AT	IDFREQ	ML	STB
CF	LG	RB	TH
CNV	MKA	RBR	TRA
DL	MKN	RL	TRB
FA	MKPT	ROFFSET	VB
FB	MKPX	SP	VBR
FOFFSET	MKT	SS	VTL
HNLOCK	ML	ST	LIMIFAIL*

* Firmware datecode 9I0116 and later.

trace element any point (element) of a user-defined trace element that identifies the trace being accessed. When a trace of a greater length is operated upon and stored in a trace of lesser length, the trace is truncated to fit. When a trace of a lesser length is operated upon and stored in a trace of greater length, the last trace element is extended for operations with the greater length.

trace range any segment or point of a user-defined trace that specifies trace limits. When a trace of a greater length is operated upon and stored in a trace of lesser length, the trace is truncated to fit. When a trace of a lesser length is operated upon and stored in a trace of greater length, the last trace element is extended for operations with the greater length.

Command Syntax Diagrams

units frequency, amplitude, time, and current units. These are listed below.

Frequency	Amplitude	Time	Current	Entry
HZ	DB	S	A	EP ¹
KHZ	DBM	SC	MA	
MHZ	DBMV	US	UA	
GHZ	DBUV	MS		
KZ	DM			
MZ	MV			
GZ	UV			
	MW			
	V			

¹ For firmware datecode 910116 and later, the value of a user-defined variable may be modified using EP. Refer to "Using EP to Modify User-Defined Variables (firmware revision 910116 and later)" in this chapter.

user-defined function

user-defined label from 1 to 16 characters long declared in FUNCDEF statement. Choice of characters are A to Z, 0 through 9, and underscore (-).

user-defined array

user-defined array label from 1 to 16 characters long declared in ARRAYDEF statement. Choice of characters are A to Z, 0 through 9, and underscore (-). The range is from 2 to 2047 elements.

user-defined trace

user-defined trace or array label from 1 to 16 characters long declared in TRDEF statement. Choice of characters are A to Z, 0 through 9, and underscore (-). The maximum number of elements possible is about 65,000. The actual number is dependent on available user memory. M format supported for user-defined traces only.

user-defined variable

user-defined variable label from 1 to 16 characters long declared in VARDEF or LCLVAR statement. Choice of characters are A to Z, 0 through 9, and underscore (-).

Textual Notation

Text used in syntax diagrams is defined below.

Bold Type	is used to represent literals which must be entered in the command exactly as shown.
CAPITAL LETTERS	are used to represent literals which must be entered in the command exactly as shown.
< >	enclose command parameters or elements of the language being defined. These elements are described in the above section titled "Parameters in Rectangular Boxes."
[]	indicate that whatever occurs within the brackets is an optional entry.
	means "or." You may choose only one of the elements from a list. As an example, <a> means a or b, but not both.
()	clarify which elements may be chosen.
_	(underscore) represents a space which must be placed where indicated.
::=	means "is defined as . . ." As an example, <a>::=<c> indicates that <a> can be replaced by the series of elements, <c> in any statement where <a> occurs.
{ }	enclose descriptive comment which refers to the preceding item in the command sequence.

Command Syntax Diagrams

Textual Notation Conventions

<A-block data field>	#A<high-byte><low-byte><data byte><data byte &END>
<A-block data format>	#A<high-byte><low-byte><data byte><data byte &END>
<analyzer command>	any spectrum analyzer command
<block data field>	<A-block data field> <M-block data field> > (B, I, and P block-data fields are not supported)
<CR>	{13} (ASCII carriage return)
<destination>	<trace label array label> <variable identifier> TRA TRB
<EOI>	end or identify
<integer>	integer number
<key number>	integer 1–10 defined in KEYDEF statement
<length>	two 8-bit bytes specifying the length of the identifier
<LF>	{10} (ASCII line feed)
<numeric data field>	<real>
<numeric data format>	<real><CR><LF><EOI><trace label> <variable identifier> <numeric data field> TRA TRB)
<real>	positive or negative real number
<source 1 or 2>	<trace label array label> <variable identifier> <numeric data field> TRA TRB predefined variable
<string data field>	<string delimiter><ASCII character><string delimiter>
<string delimiter>	<!> <"> <\$> <%> <&> <'>- </> <\> <:> <@>
<trace destination>	<trace label> TRA TRB
<trace label>	1 to 16 ASCII characters defined in TRDEF statement
<trace source>	<trace label> TRA TRB
<variable identifier>	1 to 16 ASCII characters defined in the VARDEF statement, LCLVAR, or ARRAYDEF.

Table 4-1. Functional Command Listing

Function	Mnemonic	Description
Auto-function command	AUTOEXEC	Automatic execution
	AUTOFUNC	Automatic function
	AUTOSAVE	Automatic trace save
	CLRSCHED	Clear autoexec/ autosave schedule
Active function command	ACTVFUNC	Active function
Clock command	SETDATE	Set date
	SETTIME	Set time
	TIMEDATE	Set time and date display on or off
Graphics operation command	CLRDISP	Clear display
	DATEMODE	Date format display
	DSPLY	Display
	EM	Erase user graphics memory
	OR	Set origin
	PA	Plot absolute
	PD	Pen down
	PR	Plot relative
	PU	Pen up
	TEXT	Screen text
HP-IB control command	CTRLHPIB	Control HP-IB
	ENTER	Enter from HP-IB
	OUTPUT	Output from HP-IB
	RELHPIB	Release HP-IB control
Limit-line command	EDITDONE	Limit-line edit done
	EDITLIML	Edit limit line
	LIMD	Limit-line segment delta
	LIMF	Limit-line segment freq
	LIMIFAIL	Limit-line test fail status
	LIMIPURGE	Purge limit line
	LIMIREL	Relative limit line
	LIMIRCL	Recall limit line
	LIMISAV	Save limit line
	LIMITEST	Limit-line test mode
	LIML	Lower amplitude limit
	LIMM	Middle amplitude limit
	LIMTFL	Flat limit-line segment
	LIMTSL	Sloped limit-line segment
	LIMU	Upper limit-line value
	SADD	Add limit-line segment
	SDEL	Delete limit-line segment
	SDON	Limit-line segment edit done
	SEDI	Edit limit-line segment
	SENDER	Enter limit-line segment parameters

Command Syntax Diagrams

Table 4-1. Functional Command Listing (continued)

Function	Mnemonic	Description
Memory Operation Command	CARDLOAD	Copy card data to module
	CARDSTORE	Copy module data to card
	CATALOG	Catalog module memory or card over HP-IB
	FORMAT	Format the memory card
	MSDEV	Mass storage device
Module-menu command	KEYCLR	Clear user softkey labels
	KEYDEF	Define user softkey label
	MENU	Show user softkey menu
	SHOWMENU	Show all user softkeys
	SKYDEF	Define user softkey
	SKYCLR	Clear user softkey
Predefine math operation	ABS	Absolute
	ADD	Addition
	AVG	Trace average
	DIV	Division
	EXP	Exponent
	INT	Integer
	LOG	Logarithm
	MIN	Minimum
	MOD	Modulo
	MOV	Move
	MPY	Multiply
	MXM	Maximum
	SQR	Square root
	SUB	Subtract
Trace Operation	MEAN	Trace mean
	MODRCLT	Recall trace from module or card
	MODSAVT	Save trace in module memory or card
	PDA	Probability distribution of amplitude
	PDF	Probability distribution of frequency
	PEAKS	Trace peaks
	RMS	Trace root-mean square
	SMOOTH	Smooth trace
	STDEV	Standard deviation of trace amplitude
	SUM	Sum of trace amplitude
	SUMSQR	Sum of squared trace amplitude
	VARIANCE	Variance
User definition command	ARRAYDEF	Array definition
	DISPOSE	Dispose
	FUNCDEF	Function definition
	LCLVAR	Local variable definition
	MEM	Memory available
	ONEOS	On end of sweep
	TRDEF	Trace definition
	VARDEF	Variable definition

Table 4-1. Functional Command Listing (continued)

Function	Mnemonic	Description
User program control flow	ABORT	Abort
	IF/THEN/ELSE/END IF	If/then/else/end if
	REPEAT/UNTIL	repeat/until
	RETURN	Return from function
Auxiliary interface	CNTLA	Aux control line A
	CNTLB	Aux control line B
	CNTLC	Aux control line C
	CNTLD	Aux control line D
	CNTLI	Aux control line I

Table 4-2. Alphabetical Command Listing

Mnemonic	Description	Function
-A-		
ABORT	Abort	User program control flow
ABS	Absolute	Predefined math operation
ACTVFUNC	Active function	Active function command
ADD	Addition	Predefined math operation
ARRAYDEF	Array definition	User definition command
AUTOEXEC	Automatic execute	Auto-function command
AUTOFUNC	Automatic function	Auto-function command
AUTOSAVE	Automatic save trace	Auto-function command
AVG	Average trace	Predefined math operation
-C-		
CARDLOAD	Copy card data to memory	Memory operation command
CARDSTORE	Copy memory data to card	Memory operation command
CATALOG	Catalog module memory or card over HP-IB	Memory operation command
CLRDISP	Clear display	Graphics control command
CLRSCHED	Clear autoexec/ autosave schedule	Auto-function command
CNTLA	Sets control line A	Aux interface command
CNTLB	Sets control line B	Aux interface command
CNTLC	Sets control line C	Aux interface command
CNTLD	Sets control line D	Aux interface command
CNTLI	Status of control line I	Aux interface command
CTRLHPIB	Control HP-IB	HP-IB control command
-D-		
DATEMODE	Date mode	Clock control command
DISPOSE	Dispose	User definition command
DIV	Divide	Predefined math operation
DSPLY	Display	Graphics control command
-E-		
EDITDONE	Edit done	Limit-line command
EDITLIML	Edit limit line	Limit-line command
EM	Erase user display memory	HP-IB control command
ENTER	Enter	HP-IB control command
EXP	Exponent	Predefined math operation

Command Syntax Diagrams

Table 4-2. Alphabetical Command Listing (continued)

Mnemonic	Description	Function
-F-		
FORMAT	Format	Formats the memory card
FUNCDEF	Function definition	User definition command
-I-		
IF/THEN/ELSE/END IF	If/then/else/end if	User program control flow
INT	Integer	Predefined math operation
-K-		
KEYCLR	Clear a user key	Module menu command
KEYDEF	Define a user key	Module menu command
-L-		
LCLVAR	Local variable	User definition command
LIMD	Limit-line delta	Limit-line command
LIMF	Limit-line frequency	Limit-line command
LIMIFAIL	Limit-line fail	Limit-line command
LIMIPURGE	Limit-line purge	Limit-line command
LIMIRCL	Limit-line recall	Limit-line command
LIMIREL	Relative limit line	Limit-line command
LIMISAV	Limit-line save	Limit-line command
LIMITEST	Limit-line test	Limit-line command
LIML	lower limit value	Limit-line command
LIMM	Middle limit value	Limit-line command
LIMTFL	Flat limit-line segment	Limit-line command
LIMTSL	Slope limit-line segment	Limit-line command
LIMU	Upper limit value	Limit-line command
LOG	Logarithm	Predefined math operation
-M-		
MEAN	Trace mean	Predefined math operation
MEM	Memory available	Memory operation command
MENU	Menu	Module menu command
MIN	Minimum	Predefined math operation
MOD	Modulo	Predefined math operation
MODRCLT	Recall module trace	Trace operation command
MODSAVT	Save module trace	Trace operation command
MOV	Move	Predefined math operation
MPY	Multiply	Predefined math operation
MSDEV	Mass storage device	Memory operation command
MXM	Maximum	predefined math operation

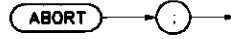
Table 4-2. Alphabetical Command Listing (continued)

Mnemonic	Description	Function
-O-		
ONEOS	On end of sweep	User definition command
OR	Set Origin	Graphics control command
OUTPUT	Output to HP-IB	HP-IB control command
-P-		
PA	Plot absolute	Graphics control command
PD	Pen Down	Graphics control command
PDA	Probability distribution of amplitude	Trace operation command
PDF	Probability distribution of frequency	Trace operation command
PEAKS	Trace peaks	Trace operation command
PR	Plot relative	Graphics control command
PU	Pen up	Graphics control command
-R-		
RELHPIB	Release HP-IB	HP-IB control command
REPEAT/ UNTIL	Repeat sequence until	User program control Flow
RETURN	Return from function	User program control Flow
RMS	Trace root-mean-square root value	Trace operation command
-S-		
SADD	Add line segment	Limit-line command
SDEL	Delete line segment	Limit-line command
SDON	Line segment done	Limit-line command
SEDI	Edit line segment	Limit-line command
SENDER	Enter line segment	Limit-line command
SETDATE	Set date	Clock control command
SETTIME	Set time	Clock control command
SHOWMENU	Show menu	Menu command
SKYCLR	Clear user softkey	Menu command
SKYDEF	Define user softkey	Menu command
SMOOTH	Smooth trace	Trace operation command
SQR	Square root	Predefined math operation
STDEV	Standard deviation of Trace amplitude	Trace operation command Memory operation command
SUB	Subtract	Predefined math operation
SUM	Sum of trace amplitude	Trace operation command
SUMSQR	Sum of squared trace amplitude	Trace operation command
-T-		
TEXT	Text	Graphics control command
TIMEDATE	Time and date mode	Clock control command
TRDEF	Trace definition	User definition command
-V-		
VARDEF	Variable definition	User definition command
VARIANCE	Variance	Trace operation command

ABORT

Abort Operation

Syntax



xf21a

Description If a user defined function is executing, nested to any level, it is aborted. Control is then returned to the user-input level. If the nested program is initiated with a front-panel key, the control returns to the front-panel operation level. If the nested program is initiated with an HP-IB command, control is returned to the HP-IB level.

Related Commands: FUNCDEF, REPEAT/UNTIL, IF/THEN, AUTOFUNC

Program Example

```

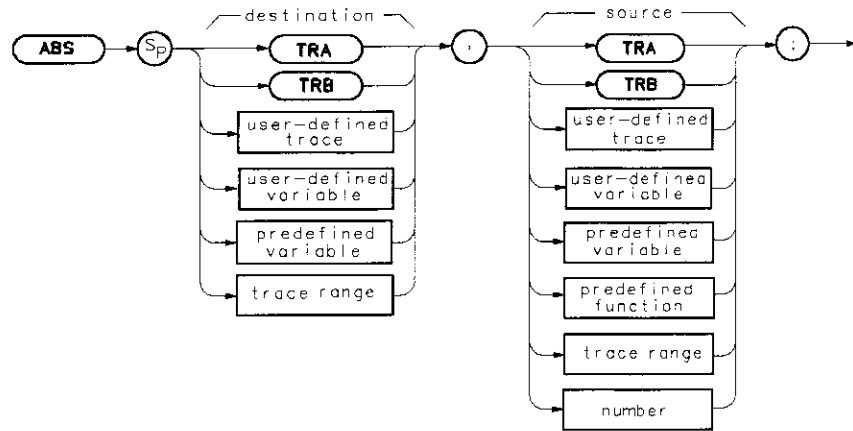
10  ! The following example shows the use of the ABORT command.  The
20  ! instructions within the FUNCDEF '@' delimiters form a structure
30  ! called a DLP (downloadable program).
40  !
50  ASSIGN @Sa TO 718      ! Assign I/O path to address 718.
60  !
70  OUTPUT @Sa;"FUNCDEF TST,@;"      ! Logical start of the DLP.
80  OUTPUT @Sa;"VARDEF Y,500;"      ! Create variable and initialize to 500.
90  OUTPUT @Sa;"VARDEF X,500;"      ! Create variable and initialize to 500.
100 OUTPUT @Sa;"CLRDISP;"           ! Clear display.
110 OUTPUT @Sa;"REPEAT;"           ! Begin loop.
120 OUTPUT @Sa;" IF Y,LT,100; THEN;"
                                   ! Test condition.
130 OUTPUT @Sa;"  PU;PA 100,X;"     ! Move pen.
140 OUTPUT @Sa;"  TEXT%DLP ABORTED%;"
                                   ! Print text.
150 OUTPUT @Sa;"  ABORT;"
160 OUTPUT @Sa;"  ELSE;"
170 OUTPUT @Sa;"  PU;PA 100,X;"     ! Move pen.
180 OUTPUT @Sa;"  DSPLY Y,5,2;"     ! Display 'Y'.
190 OUTPUT @Sa;"  SUB X,X,40;"      ! Decrement pen pointer.
200 OUTPUT @Sa;"  SUB Y,Y,100;"     ! Decrement variable.
210 OUTPUT @Sa;"  ENDIF;"
220 OUTPUT @Sa;"UNTIL Y,EQ,-100;"   ! End loop (the abort will occur
230                                   ! before this condition will be
240                                   ! satisfied).
250 OUTPUT @Sa;"@";                ! Logical end of DLP.
260 !
270 OUTPUT @Sa;"TST;"              ! Execute DLP.
280 !
290 ASSIGN @Sa TO *                ! Close I/O path.
300 END

```

ABS

Absolute Value

Syntax



xf22o

Description The absolute value of the source is stored in the destination.

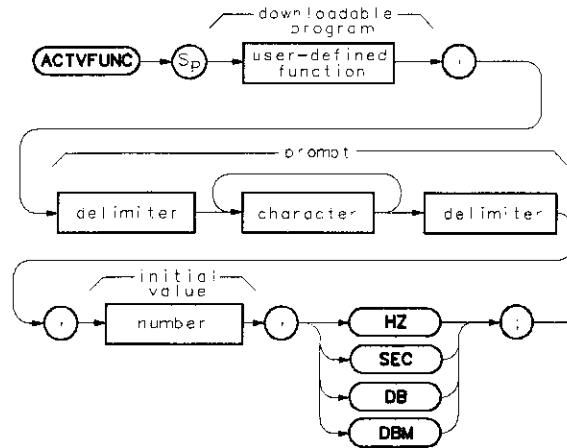
Program Example

```
10  ! The following example shows the use of the ABS command.
50  !
60  ASSIGN @Sa TO 718          ! Assign I/O path to address 718.
70  !
80  OUTPUT @Sa;"IP;";         ! Instrument preset.
81  OUTPUT @Sa;"CF 300MHZ;SP 1MHZ;"; ! Set center frequency and span.
90  OUTPUT @Sa;"VARDEF ABSLT_VAL,0;"; ! Create variable, initialize to 0.
91  OUTPUT @Sa;"TS;MKPK HI;";   ! Marker to signal peak.
100 OUTPUT @Sa;"ABS ABSLT_VAL,MKA;"; ! Put the absolute value of the
101                                     ! marker amplitude into ABSLT_VAL.
103 OUTPUT @Sa;"ABSLT_VAL?;";   ! Query ABSLT_VAL.
104 ENTER @Sa;Abs_val          ! Get the value from the analyzer.
105 !
107 PRINT "The absolute value of the marker amplitude =";Abs_val
190 !
330 ASSIGN @Sa TO * ! Close I/O path.
340 END
```

ACTVFUNC

Active Function

Syntax



x123a

Description

This command makes a user-defined function operate like an active function. Active functions have the following characteristics:

- Their current value is displayed in the active-function block.
- Their values may be modified using front panel keys.
- Their values are expressed as variables for operations with other spectrum analyzer functions.

The ACTVFUNC command operates on any user-defined function, which must in turn operate on the predefined variable, ACTVAL. Use the following procedure to use the ACTVFUNC command:

1. Use FUNCDEF to define a user-defined function that operates on the predefined variable, ACTVAL.
2. Incorporate the ACTVFUNC command into the definition of a second user-defined function (also defined by FUNCDEF).
 - a. Be sure to follow ACTVFUNC with the name of the user-defined function that operates on ACTVAL. Refer to the syntax diagram.
 - b. Make ACTVFUNC the last command in the user-defined function definition.
3. Execute the user-defined function that executes ACTVFUNC.
4. Enter a number with the data keys, then press a hertz, seconds, or decibel terminate key.

When ACTVFUNC is executed, the spectrum analyzer displays the text specified by ACTVFUNC, then waits for data entry from the front panel data keys.

Note

The command line causes the prompt-response input to be multiplied by the corresponding multiplier of the terminate key. As an example, if the initial value is 10 Hz and the GHz units key is used to terminate the prompt response, 10 GHz is displayed in the active-function block.

Restrictions: Subject to available memory. The following commands consume memory: ONEOS, ARRAYDEF, TRDEF, KEYDEF, LCLVAR, VARDEF REPEAT/UNTIL, and IF/THEN.

Program Example

```

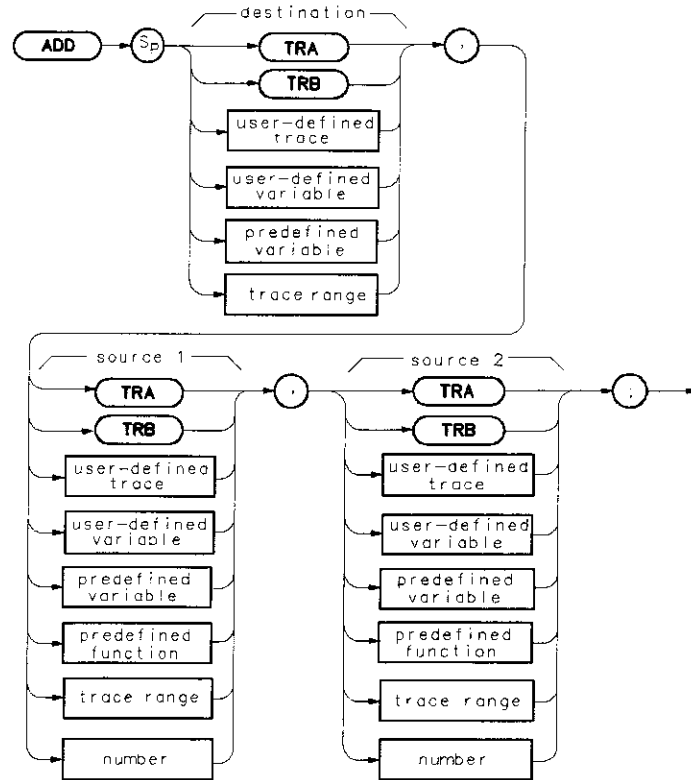
10      ! The following example shows the use of the ACTVFUNC command. The
20      ! example uses an input signal of 300 MHz such as the CAL OUTPUT and
30      ! may be set (by the user input) to look at its harmonics. An input
40      ! signal is not necessary to the function of this example, but acts as
50      ! a visual aid.
60      !
70      ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
80      !
90      OUTPUT @Sa;"FUNCDEF TST,@;"      ! Logical start of the DLP.
100     OUTPUT @Sa;"VARDEF FREQ,0;"      ! Create variable, initialize to 0.
110     OUTPUT @Sa;"SP 5MHZ;"           ! Set span.
120     OUTPUT @Sa;"REPEAT;"            ! Begin loop.
130     OUTPUT @Sa;"  ADD FREQ,FREQ,ACTVAL;" ! Increment freq by number entered.
140     OUTPUT @Sa;"  MOV CF,FREQ;TS;"   ! Set new center freq; take sweep.
150     OUTPUT @Sa;"UNTIL FREQ,GT,2.9E9;" ! Loop until frequency is 2.9 GHz.
160     OUTPUT @Sa;"@";                 ! Logical end of DLP.
170     !
180     OUTPUT @Sa;"FUNCDEF SETUP,@;"    ! Logical start of the DLP.
190     OUTPUT @Sa;"IP;SNGLS;"          ! Instrument preset; single sweep.
200     OUTPUT @Sa;"ACTVFUNC TST,%ENTER FREQ%,300E6,HZ;"
                                         ! Set up the analyzer to
210                                         ! wait for front-panel input.
220     OUTPUT @Sa;"@";                 ! Logical end of DLP.
230     !
240     OUTPUT @Sa;"KEYDEF 1,SETUP,%DEMO ACTVFUNC%";
                                         ! Create a softkey to
250                                         ! initiate the example.
260     PRINT "Press {LCL} [MODULE] {USER KEY}, then {ACTVFUNC} to run the
example."
270     PRINT
280     PRINT "Then enter a starting frequency such as 300 MHz. When the units"
290     PRINT "key has been pressed, the example will begin to increment the"
300     PRINT "Center Frequency from the entered frequency to 2.9 GHz in steps"
310     PRINT "equal to the entered value."
320     !
330     ASSIGN @Sa TO * ! Close I/O path.
340     END

```


ADD

Addition

Syntax



x124c

Description The values of source 1 and 2 are added and the sum is sent to the destination.

ADD

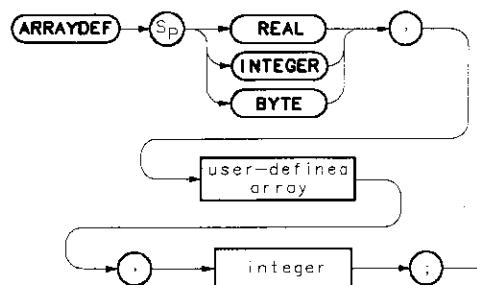
Program Example

```
10  ! The following example shows the use of the ADD command.  The example
20  ! uses an input signal of 300 MHz, such as the CAL OUTPUT signal, and
30  ! looks at its harmonics.  An input signal is not necessary to the
40  ! function of this example, but acts as a visual aid.
50  !
60  ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
70  !
80  OUTPUT @Sa;"FUNCDEF ADDEX,@";    ! Logical start of the DLP function.
90  OUTPUT @Sa;"VARDEF FREQ,300E6";  ! Create variable and initialize it to
                                     ! 300 MHz.
100 OUTPUT @Sa;"IP;SP 1MHZ;";        ! Instrument preset; set span.
110 !
120 OUTPUT @Sa;"REPEAT;";           ! Begin loop.
130 OUTPUT @Sa;"  MOV CF,FREQ;";    ! Set center frequency.
140 OUTPUT @Sa;"  TS;";             ! Take a sweep to update display.
150 OUTPUT @Sa;"  ADD FREQ,FREQ,300E6;";
                                     ! Increase FREQ by 300 MHz.
160 OUTPUT @Sa;"UNTIL FREQ,GT,3E9;"; ! End of loop.
170 OUTPUT @Sa;"@";                 ! Logical end of DLP function.
180 !
190 OUTPUT @Sa;"ADDEX;";            ! Execute function.
200 !
210 ASSIGN @Sa TO * ! Close I/O path.
220 END
```

ARRAYDEF

Array Definition

Syntax



x125a

Description

The ARRAYDEF command allows you to create user-defined arrays of the type REAL, INTEGER, or BYTE. The number of elements in the array is limited only by the amount of free user memory in the module. In array operations, arrays of different types (such as REAL, INTEGER, or BYTE) cannot be operated on unless they are accessed element by element. For example a REAL array defined by the command in line 10 below, then executed with line 20, is an invalid combination.

```

10 OUTPUT 718;"ARRAYDEF REAL,TST,601;"
20 OUTPUT 718;"MOV TRA,TST;"

```

This is because TST is a REAL array and TRA is an integer array. A valid command combination to access an array element by element is as follows:

```

10 OUTPUT 718;"ARRAYDEF REAL,TST,601;"
20 OUTPUT 718;"MOV TRA[1],TST[1];"

```

Restrictions: Subject to available memory. The following commands consume memory: ONEOS, ARRAYDEF, TRDEF, KEYDEF, LCLVAR, VARDEF REPEAT/UNTIL, and IF/THEN.

Query Response

For array data transfer to and from an external controller, only M format is supported. A, B, I, and P block-data field formats are not currently supported.

ARRAYDEF

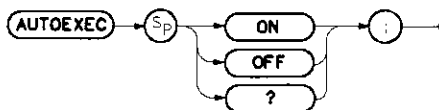
Program Example

```
10 ! The following example shows the use of the ARRAYDEF command. The
20 ! example uses an input signal of 300 MHz such as the CAL OUTPUT and
30 ! looks at its harmonics.
40 !
50 ASSIGN @Sa TO 718 ! Assign I/O path to address 718.
60 !
70 OUTPUT @Sa;"ARRAYDEF REAL,FREQS,10;"; ! Create a 10-element REAL array.
80 OUTPUT @Sa;"ARRAYDEF REAL,AMPLS,10;"; ! Create another REAL array.
90 OUTPUT @Sa;"VARDEF X,1;VARDEF Y,600;"; ! Create, initialize two variables.
100 OUTPUT @Sa;"IP;FA 100MHZ;"; ! Preset; set start frequency.
110 OUTPUT @Sa;"FB 2900MHZ;TS;"; ! Set stop frequency; take sweep.
120 OUTPUT @Sa;"MKPK HI;"; ! Marker to peak of highest signal.
130 !
140 OUTPUT @Sa;"REPEAT;"; ! Begin loop.
150 OUTPUT @Sa;" MOV FREQS[X],MKF;"; ! Put marker frequency into Xth
160 ! element of FREQS array.
170 OUTPUT @Sa;" MOV AMPLS[X],MKA;"; ! Put marker amplitude into Xth
180 ! element of AMPLS array.
190 OUTPUT @Sa;" MKPK NH;"; ! Marker to next highest peak.
200 OUTPUT @Sa;" ADD X,X,1;"; ! Increment array pointer.
210 OUTPUT @Sa;"UNTIL X,GT,10;"; ! Loop until array pointer > 10.
220 !
230 OUTPUT @Sa;"CLRDSP;"; ! Clear analyzer display.
240 OUTPUT @Sa;"PA 110,650;"; ! Move pen to starting point.
250 OUTPUT @Sa;"TEXT %FREQUENCY(Hz) AMPLITUDE(dBm)%;";
! Write titles.
260 OUTPUT @Sa;"MOV X,1;"; ! Initialize array pointer.
270 OUTPUT @Sa;"PA 100,Y;"; ! Pen to first column.
280 !
290 OUTPUT @Sa;"REPEAT;"; ! Begin loop.
300 OUTPUT @Sa;" DSPLY FREQS[X],10,3;"; ! Display Xth frequency.
310 OUTPUT @Sa;" TEXT % %"; ! Space over to next column.
320 OUTPUT @Sa;" DSPLY AMPLS[X],10,3;"; ! Display Xth amplitude.
330 OUTPUT @Sa;" SUB Y,Y,30;"; ! Decrement pen location.
340 OUTPUT @Sa;" PA 100,Y;"; ! Move pen to new location
350 OUTPUT @Sa;" ADD X,X,1;"; ! Increment array pointer.
360 OUTPUT @Sa;"UNTIL X,GT,10;"; ! Loop until array pointer > 10.
370 !
380 ASSIGN @Sa TO * ! Close I/O path.
390 END
```

AUTOEXEC

Automatic Execution

Syntax



x125a

Description Turns off or on the automatic function as defined with the AUTOFUNC command.

Prerequisite Command: AUTOFUNC

Related Commands: AUTOSAVE, CLRSCHEM

Program Example

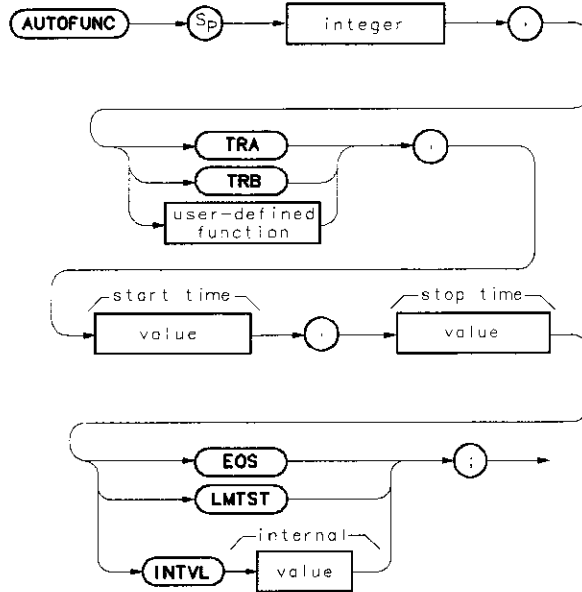
```

10    ! The following example shows the use of the AUTOEXEC command.
20    !
30    ASSIGN @Sa TO 718           ! Assign I/O path to address 718.
40    !
50    Autoexec_status=999       ! Initialize status variable to
60                                ! ELSE case value.
70    OUTPUT @Sa;"AUTOEXEC ON;"; ! Turn AUTOEXEC function on.
80    OUTPUT @Sa;"AUTOEXEC?;";  ! Query its status.
90    ENTER @Sa;Autoexec_status ! Get the status from the analyzer.
100   GOSUB Check_status
110   !
120   OUTPUT @Sa;"AUTOEXEC OFF;"; ! Turn AUTOEXEC function off.
130   OUTPUT @Sa;"AUTOEXEC?;";  ! Query its status.
140   ENTER @Sa;Autoexec_status ! Get the status from the analyzer.
150   GOSUB Check_status
160   STOP
170   !
180   Check_status:              ! Subroutine to display the status of
190                                ! the AUTOEXEC function.
200   SELECT Autoexec_status
210   CASE 0
220     PRINT "The AUTOEXEC function is OFF."
230   CASE 1
240     PRINT "The AUTOEXEC function is ON."
250   CASE ELSE
260     ! This condition does not exist, however it is a good programming
270     ! practice to include an ELSE case for unexpected situations.
280     PRINT "The AUTOEXEC function returned an unknown status value."
290   END SELECT
300   RETURN                      ! From Check_status subroutine.
310   END
  
```

AUTOFUNC

Automatic Function

Syntax



x127a

Description

This command specifies the operation for automatic execution. Use the AUTOEXEC or AUTOSAVE function to turn execution on or off. When using LMTST as a trigger, the start and stop times must be valid dates (for example, 000000000000 or 999999999999 are not valid.) The specific operation may be one of the following:

- Store trace A data.
- Store trace B data.
- Execute a user-defined function.

when one of the following conditions occur:

- A specified time period elapses.
- The end of the sweep occurs.
- Current trace data exceeds limit-line values.

- Parameters**
1. There can be from one to seven processes scheduled.
 2. TRA | TRB | DLP (TRA, TRB for autosaving traces; DLP::= name of user-defined function.)
 3. Start-time format: MMDDYYHHMMSS or DDMMYYHHMMSS (depending on datemode format)
 4. Stop-time: MMDDYYHHMMSS or DDMMYYHHMMSS (depending on datemode format)
 5. **EOS** | **LMTST** | **INTVL** selects criteria to perform autosave or to execute a DLP.
 - a. **EOS** occurs at the end of a sweep.
 - b. **LMTST** occurs at the end of a sweep after a limit test failure.
 - c. **INTVL** occurs after the end of a sweep, when the first designated time interval is reached. The format for an interval entry is HHMMSS.
 6. **INTVL** occurs at the interval time if INTVL is selected.

Restrictions: Subject to available memory. The following commands consume memory: ONEOS, ARRAYDEF, TRDEF, KEYDEF, LCLVAR, VARDEF REPEAT/UNTIL, and IF/THEN.

Prerequisite Commands: FUNCDEF, CONTS, TITLE, SLRW, and limit-line commands.

Related Commands: CLRDSP, TITLE, ABORT, RETURN, PAUSE, ERASE, DISPOSE.

Program Example

```

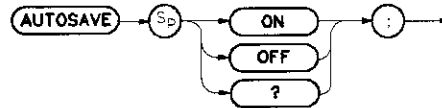
10  ! The following example shows the use of the AUTOFUNC command.
20  !
30  ASSIGN @Sa TO 718      ! Assign I/O path to address 718.
40  !
50  OUTPUT @Sa;"IP;";      ! Instrument preset.
60  ! Create three items for the AUTOEXEC Schedule.
70  OUTPUT @Sa;"AUTOFUNC 1 ,BOX,010189000000,010589000000 EOS;";
80  OUTPUT @Sa;"AUTOFUNC 2 ,CHK_SIG,123188235500,123188235900 INTVL 000030;";
90  OUTPUT @Sa;"AUTOFUNC 3 TRA,122488090000,122488125900 LMTST;";
100 PRINT "Press {LCL} [MODULE] AUTOEXEC MENU to see the new schedule."
110 DISP "Press [CONTINUE] on the computer when ready"
120 PAUSE
130 !
140 ! Remove one item from the schedule.
150 OUTPUT @Sa;"IP;CLRSCHED 2;";
160 PRINT "Press {LCL} [MODULE] {AUTOEXEC MENU} to see the updated schedule."
170 DISP "Press [CONTINUE] on the computer when ready"
180 PAUSE
190 !
200 ! Clear the entire schedule.
210 OUTPUT @Sa;"IP;CLRSCHED ALL;";
220 PRINT "Press {LCL} [MODULE] {AUTOEXEC MENU} to see the cleared schedule."
230 DISP ""
240 !
250 ASSIGN @Sa TO *
260 END

```

AUTOSAVE

Automatically Save Traces

Syntax



x128a

Description

When set to ON, this command activates the automatic saving of traces. The data to be automatically saved must be identified in the AUTOFUNC command.

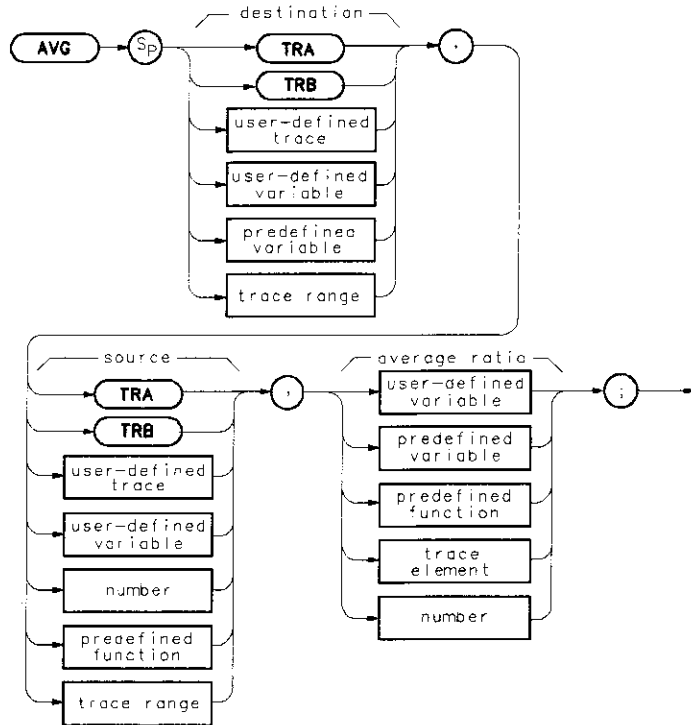
Prerequisite Commands: AUTOFUNC and TITLE.

Restrictions: A maximum of one trace per second and subject to available memory. The following commands consume memory: ONEOS, ARRAYDEF, TRDEF, KEYDEF, LCLVAR, VARDEF REPEAT/UNTIL, and IF/THEN.

AVG

Average

Syntax



xf29a

Description The data in the source and destination are averaged, then stored in the destination. The following averaging algorithm is used:

$$DEST_{new} = ((RATIO - 1) \times DEST_{old} + SOURCE) / RATIO$$

Parameter Range Average Ratio:
 Minimum: -32,767
 0 is not allowed
 Maximum: +32,767

AVG

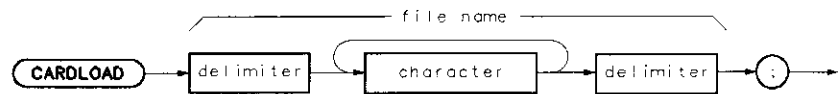
Program Example

```
10    ! The following example shows the use of the AVG command.
20    !
30    ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
40    !
50    OUTPUT @Sa;"IP;";                ! Instrument preset.
60    OUTPUT @Sa;"SNGLS;CLRW TRA;TS;"; ! \ Set up the analyzer.
70    OUTPUT @Sa;"FA 300MHZ;FB 2GHZ;TS;"; ! /
80    OUTPUT @Sa;"VAVG 10;TS;";
90    OUTPUT @Sa;"VIEW TRA;VIEW TRB;"; !
100   OUTPUT @Sa;"AVG TRB,TRA,2;";    ! Average trace A and trace B,
110                                       ! place result in trace B.
120   ASSIGN @Sa TO * ! Close I/O path.
130   END
```

CARDLOAD

Copy Data From Memory Card to Module Memory

Syntax



xf210a

Description

Copy the specified data from the memory card to the module battery-backed memory. The filename label (identifier) should follow this format: 1 to 9 ASCII characters followed by a period, then followed by one of these three-letter suffixes: DLP or LMT. Valid ASCII characters that may be selected are: A through Z, 0 through 9, and _ (underscore).

A limit line may be created with the extension .lim appended to the filename. The extension LMT must be used when using the CARDLOAD and CARDSTORE commands with mass memory module firmware revisions before 950829. For later firmware, either extension is allowed.

The following statement copies the DLP "ABCDEFGHI" from the memory card to the module:

```
CARDLOAD %ABCDEFGHI.DLP%;
```

CARDLOAD

Program Example

Note

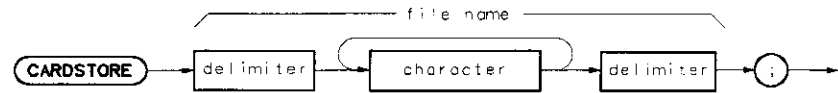
DLPs, traces, and limit lines stored on memory cards using a mass-memory module with firmware datecode 910116 or later cannot be read into a module with an earlier firmware datecode (for example 890524). DLPs, traces, and limit lines stored on cards using modules with firmware datecode 890524 or earlier can be read into a module with firmware datecode 910116 or later.

```
10  ! The following example shows the use of the CARDLOAD command. The
20  ! example creates a DLP (downloadable program) and stores it in memory.
30  ! It is then copied to the memory card, if available.
40  !
50  ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
60  !
70  OUTPUT @Sa;"FUNCDEF BOX,@;";     ! Logical start of function 'BOX'.
80  OUTPUT @Sa;" IP;";              ! Instrument preset.
90  OUTPUT @Sa;" CLRDSP;";          ! Clear the analyzer screen.
100 OUTPUT @Sa;" PU;PA 300,300;";    ! Move pen to starting point.
110 OUTPUT @Sa;" PD;PR 240,0;";     ! \
120 OUTPUT @Sa;" PR 0,240;";        !  \ Draw
130 OUTPUT @Sa;" PR -240,0;";       !  / rectangle.
140 OUTPUT @Sa;" PR 0,-240;";      ! /
150 OUTPUT @Sa;"@";                ! Logical end of function.
160 !
170 OUTPUT @Sa;"CARDSTORE %BOX.DLP%"; ! Copy DLP to memory card.
180 !
190 ! The DLP now appears on the catalog of the memory card available in
200 ! the module utility menu.
210 !
220 OUTPUT @Sa;"DISPOSE BOX;";      ! Remove the DLP 'BOX' from
230                                ! the module memory. It is
240                                ! still on the card!!
250 !
260 OUTPUT @Sa;"CARDLOAD %BOX.DLP%"; ! Copy DLP from card to module
270                                ! memory. Now it's in both places.
280 !
290 ASSIGN @Sa TO * ! Close I/O path.
300 END
```

CARDSTORE

Copy Data to a Memory Card

Syntax



xf211a

Description

Copy the specified data from the module memory to a RAM card. The filename should follow this format: 1 to 9 ASCII characters followed by a period, then followed by one of these three-letter suffixes: DLP or LMT. Valid ASCII characters that may be selected are: A through Z (upper-case only), 0 through 9, and _ (underscore). RAM cards use the Logical Interchange Format (LIF) and therefore accept only 9 of the 16 characters used to title data.

Note

The first 9 characters must be unique to avoid writing over an existing file. If a file name is longer than 9 characters or if lower-case letters are used, the file name will be converted, if possible, to an LIF compatible file name using only the first 9 characters.

A limit line may be created with the extension .lim appended to the filename. The extension .LMT must be used when using the CARDLOAD and CARDSTORE commands with mass memory module firmware revisions before 950829. For later firmware, either extension is allowed.

Note

DLPs, traces, and limit lines stored on memory cards using a mass-memory module with firmware datecode 910116 or later cannot be read into a module with an earlier firmware datecode (for example 890524). DLPs, traces, and limit lines stored on cards using modules with firmware datecode 890524 or earlier can be read into a module with firmware datecode 910116 or later.

CARDSTORE

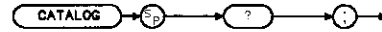
Program Example

```
10 ! The following example shows the use of the CARDSTORE command. The
20 ! example creates a DLP (downloadable program) and stores it in memory.
30 ! It is then copied to the memory card, if available.
40 !
50 ASSIGN @Sa TO 718 ! Assign I/O path to address 718.
60 !
70 OUTPUT @Sa;"FUNCDEF BOX,@;" ! Logical start of function 'BOX'.
80 OUTPUT @Sa;" IP;" ! Instrument preset.
90 OUTPUT @Sa;" CLRDSP;" ! Clear the analyzer screen.
100 OUTPUT @Sa;" PU;PA 300,300;" ! Move pen to starting point.
110 OUTPUT @Sa;" PD;PR 240,0;" ! \
120 OUTPUT @Sa;" PR 0,240;" ! \ Draw
130 OUTPUT @Sa;" PR -240,0;" ! / rectangle.
140 OUTPUT @Sa;" PR 0,-240;" ! /
150 OUTPUT @Sa;"@";" ! Logical end of function.
160 !
170 OUTPUT @Sa;"CARDSTORE %BOX.DLP%;" ! Copy DLP to memory card.
180 ASSIGN @Sa TO * ! Close I/O path.
190 END
```

CATALOG

Directory Listing over HPIB

Syntax



XCAT

Description

The CATALOG command returns a directory listing of the current mass storage device over the HPIB interface. The mass storage device should be specified using MSDEV prior to executing the CATALOG command.

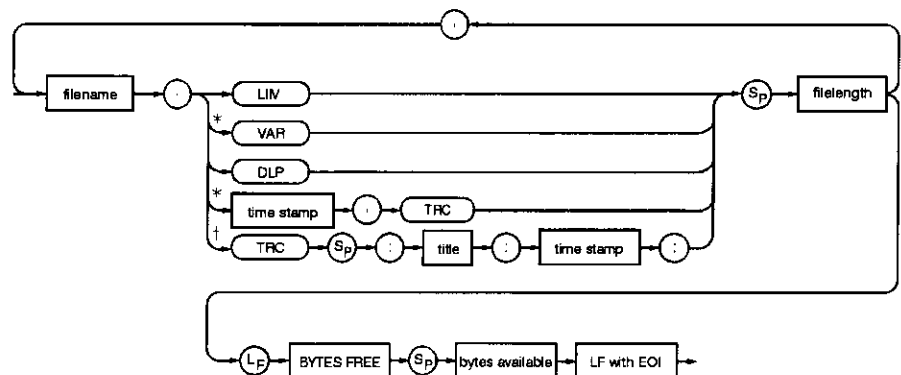
Related command: MSDEV

The listing is in the form of one string terminated by a linefeed with EOI. The string is divided into two substrings by a linefeed character. The first substring contains the actual listing and the second substring contains the amount of free memory in bytes. Refer to the Query Response figure and File Information Parameters table for more information.

Note

The CATALOG command is only available in firmware revision datecodes of 910116 and later.

Query Response



nf21a

Where:

- filename up to 16 characters (up to 9 if MSDEV is CARD)
- filelength up to 6 characters
- timestamp 12 characters
- title up to 32 characters

* These paths only available when MSDEV is MEM.

† This path only available when MSDEV is CARD.

CATALOG

File Information Parameters

File Information	Description
filename.LIM	This file contains a limit line
filename.VAR	This file is a variable defined using the VARDEF , or TRDEF , or ARRAYDEF command. It is only available when MSDEV is set to MEM.
filename.DLP	If MSDEV is set to MEM this is the downloadable program defined using the FUNCDEF command.
filename.timestamp.TRC	This file is a trace. It can be recalled by way of the front panel or by using filename.timestamp as an argument to MODRCLT when MSDEV is set to MEM
filename.TRC :title:timestamp:	This trace file is stored in the memory card. When the file is copied to the module memory, it will be stored under filename title with the corresponding timestamp.

Program Example 1 The following example simply queries the module memory for its catalog and prints the string returned from the module. Example 2 shows how to query the catalog and parse the output string.

```
10  ASSIGN @Sa TO 718
20  DIM Cat$[10000]          ! Dimension string arbitrarily large.
30  DIM Bytes_avail$[20]    ! Must accommodate "BYTES FREE XXXXX<lf>".
40  OUTPUT @Sa;"MSDEV MEM;" ! Specify MSDEV to be cataloged.
60  OUTPUT @Sa;"CATALOG?;" ! Query module for directory listing.
70  ENTER @Sa USING "#,K";Cat$,Bytes_avail$
                                ! Data up to the first <lf>
                                ! will go into Cat$; data up
80                                ! to <lf-EOI> will go into
90                                ! Bytes_avail$.
100
110 PRINT Cat$
120 PRINT Bytes_avail$
130 ASSIGN @Sa TO *
140 END
```


The following is a sample output of the mass memory module memory after the DEMO DLPs have been loaded:

```
ANALYZER.DLP 966,CONNECTCAL.DLP 157,DEMOS.DLP 34,DISTORTION_1.DLP 691,DISTORTION
_2.DLP 78,DISTORTION_3.DLP 1458,FRAME.DLP 42,PAGE_1.DLP 804,PAGE_10.DLP 961,PAGE
_11.DLP 891,PAGE_12.DLP 848,PAGE_13.DLP 958,PAGE_13A.DLP 86,PAGE_2.DLP 945,PAGE_
3.DLP 882,PAGE_4.DLP 1012,PAGE_4A.DLP 884,PAGE_5.DLP 999,PAGE_5A.DLP 802,PAGE_6
.DLP 828,PAGE_6A.DLP 697,PAGE_7.DLP 563,PAGE_7A.DLP 561,PAGE_8.DLP 856,PAGE_8A.DLP
857,PAGE_9.DLP 1007,PRESET.DLP 21,SIGMON_1.DLP 1056,SIGMON_2.DLP 73,SIGMON_3.DLP
1145,SIMPLE_1.DLP 284,SIMPLE_2.DLP 285,SIMPLE_3.DLP 279,SIMPLE_4.DLP 287,SIMPLE_
5.DLP 244,SIMPLE_6.DLP 589,SIMPLE_7.DLP 699,SIMPLE_8.DLP 1134
BYTES FREE 103886
```

Program Example 2 The following example parses the output of the CATALOG command. The program displays softkeys enabling the user to catalog the entire mass storage device or only certain file types, such as DLPs, limit lines, or traces. Variables can be cataloged only if the mass storage device is MEM; variables cannot be stored to the memory card.

This example is written for HP BASIC 2.1 or greater and runs on an HP Series 200/300 Controller.

```
10 !The following example shows the use of the CATALOG command. The example
20 !catalogs the current mass storage device (CARD or MEM) and parses the
30 !returned string into a holding array. Softkeys are then enabled to
40 !allow the user to select which part of the catalog to display.
50 !
60 INTEGER Last_match,Entry_name,Entry_length,Entry_type,Entry
70 INTEGER Space_loc,Dot_loc,Two_dots
80 DIM String_parsed$[75],String$[75],Rev_str$[75],Bytes_avail$[20]
90 DIM Mass_stg_dev$[1],Temp$[75]
100 DIM Ent$(1:3,1:500)[75],Cat$[5000]
110 !
120 ASSIGN @Sa TO 718 ! Spectrum analyzer HP-IB address
130 DISP "Loading user defined memory."
140 OUTPUT @Sa;"MSDEV?;"
150 ENTER @Sa;Mass_stg_dev$
160 IF Mass_stg_dev$="M" THEN PRINT "Mass Storage Device is MEMORY"
170 IF Mass_stg_dev$="C" THEN PRINT "Mass Storage Device is CARD"
180 !
190 OUTPUT @Sa;"CATALOG?;"
200 ENTER @Sa USING "#,K";Cat$,Bytes_avail$
210 PRINT Bytes_avail$;" (of available user memory)"
220 !
230 Entry=0
240 Entry_name=FNWhich_pos("NAME")
250 Entry_length=FNWhich_pos("LENGTH")
260 Entry_type=FNWhich_pos("TYPE")
270 REPEAT
280 Entry=Entry+1
290 CALL Universal_parse(", ",Cat$,String_parsed$,Last_match)
300 IF LEN(String_parsed$)=0 THEN No_entries
310 Dot_loc=POS(String_parsed$,".")
320 Two_dots=POS(String_parsed$[Dot_loc+1],".")
330 IF Two_dots THEN ! (it must be a trace in MEM)
```

CATALOG

```

340     Dot_loc=Two_dots+Dot_loc
350     END IF
360     Space_loc=POS(String_parsed$," ")
370     WHILE Space_loc<Dot_loc      ! Ignore spaces which are
380         Temp$=String_parsed$[Space_loc+1] ! part of LIMIT LINE title.
390         Space_loc=Space_loc+POS(Temp$," ")
400     END WHILE
410     !
420     String$=String_parsed$[1,Dot_loc-1]
430     WHILE NUM(String$[1,1])<32      ! Strip any special characters from
440         String$=String$[2]          ! system-generated dlp names.
450     END WHILE
460     Ent$(Entry_name,Entry)=String$
470     !
480     SELECT Mass_stg_dev$            !Allow for differences in string
490     CASE "M"                        !of the Memory Card and Module Memory
500         Ent$(Entry_length,Entry)=String_parsed$[Space_loc]
510     CASE "C"
520         Rev_str$=REV$(String_parsed$)
530         Ent$(Entry_length,Entry)=TRIM$(REV$(Rev_str$[1,POS(Rev_str$," ")]))
540     END SELECT
550     !
560     Ent$(Entry_type,Entry)=TRIM$(String_parsed$[Dot_loc+1;4])
570 UNTIL Last_match
580     !
590 LOOP                                ! Display softkeys until Exit key
                                        ! is pressed.

600     ON KEY 1 LABEL "DLP" GOSUB List_dlps
610     ON KEY 2 LABEL "All" GOSUB List_all
620     ON KEY 4 LABEL "Exit" RECOVER Exit_cat
630     ON KEY 3 LABEL "Lim Line" GOSUB List_limit
640     IF Mass_stg_dev$="M" THEN ON KEY 6 LABEL "Variable" GOSUB List_vars
650     !Variable key is only displayed for a catalog of MEM. Variables
660     !cannot be stored on the card.
670     ON KEY 8 LABEL "Trace" GOSUB List_traces
680     DISP "Select list mode."
690 END LOOP
700 No_entries:PRINT "No Catalog Entries !!"
710 Exit_cat:ASSIGN @Sa TO *
720 DISP "PROGRAM ENDED"
730 STOP
740 !-----!
750 List_dlps:Display_cat("DLP",Ent$(*),Entry)
760 RETURN
770 List_all:Display_cat("ALL",Ent$(*),Entry)
780 RETURN
790 List_vars:Display_cat("VAR",Ent$(*),Entry)
800 RETURN
810 List_traces:Display_cat("TRC",Ent$(*),Entry)
820 RETURN
830 List_limit:Display_cat("LIM",Ent$(*),Entry)
840 RETURN
850 END
860
!-----!

```

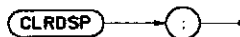
```

870 SUB Universal_parse(Delimiter$,In_n_out$,String_parsed$,INTEGER Last_match)
880 !A generic parsing routine.
890 Universal_parse: !
900 INTEGER Delim_loc,Len_delim
910 Len_delim=LEN(Delimiter$)
920 Delim_loc=POS(In_n_out$,Delimiter$)
930 IF Delim_loc=0 THEN
940 String_parsed$=In_n_out$
950 Last_match=1
960 ELSE
970 String_parsed$=TRIM$(In_n_out$[1,Delim_loc-1])
980 In_n_out$=TRIM$(In_n_out$[Delim_loc+Len_delim])
990 Last_match=0
1000 END IF
1010 SUBEND ! <Universal_parse>
1020 !-----
1030 SUB Display_cat(Class$,Ent$(*),INTEGER Num_entries)
1040 !Displays the previously parsed catalog retrieved from the 85620A+.
1050 Display_cat:!
1060 INTEGER Entry_name,Entry_length,Entry_type,Entry
1070 REAL Total_length,Total_entries
1080 !
1090 Entry_name=FNWhich_pos("NAME")
1100 Entry_length=FNWhich_pos("LENGTH")
1110 Entry_type=FNWhich_pos("TYPE")
1120 Total_length=0
1130 Total_entries=0
1140 !
1150 Im:IMAGE 12A,2X,20A,2X,8D
1160 Im2:IMAGE 12A,2X,20A,2X,15A
1170 DISP
1180 OUTPUT KBD;CHR$(255)&"K"; ! Clear screen
1190 PRINT USING Im2;" Type "," Name "," Byte total"
1200 PRINT USING Im2;"-----","-----","-----"
1210 !
1220 FOR Entry=1 TO Num_entries
1230 IF Class$=Ent$(Entry_type,Entry) OR Class$="ALL" THEN
1240 SELECT Ent$(Entry_type,Entry)
1250 CASE "DLP"
1260 String$="DLP (Func) "
1270 CASE "VAR"
1280 String$="Variable "
1290 CASE "TRC"
1300 String$="Trace "
1310 CASE "LIM"
1320 String$="Limit Line "
1330 END SELECT
1340 PRINT USING Im;String$,Ent$(Entry_name,Entry),
VAL(Ent$(Entry_length,Entry))
1350 GOSUB Totals
1360 END IF
1370 NEXT Entry
1380 PRINT
1390 PRINT "TOTALS: ";VAL$(Total_length);" bytes."
1400 PRINT " ";VAL$(Total_entries);" entries."

```

CATALOG

```
1410 SUBEXIT
1420 !
1430 Totals: !
1440 Total_entries=Total_entries+1
1450 Total_length=Total_length+VAL(Ent$(Entry_length,Entry))
1460 RETURN
1470 !
1480 SUBEND
1490 !-----!
1500 DEF FNWhich_pos(Class$)
1510 !Determines where information is placed in the array. Allows
1520 !for re-arranging of the array.
1530 Which_pos: !
1540 SELECT Class$
1550 CASE "NAME"
1560 RETURN 1
1570 CASE "LENGTH"
1580 RETURN 2
1590 CASE "TYPE"
1600 RETURN 3
1610 CASE ELSE
1620 RETURN 0
1630 END SELECT
1640 FNEND
```

CLR DSP
Clear Display**Syntax**

x1212a

Description This command erases spectrum-analyzer display annotation.

Related Commands: DSPLY, OR, TEXT, PA, PR, PD, and PU

Program Example

```

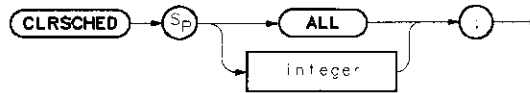
10  ! The following example shows the use of the CLR DSP command.
20  !
30  ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
40  !
50  OUTPUT @Sa;"CLR DSP;";          ! Clears the analyzer display.
60  !
70  OUTPUT @Sa;"VARDEF LOG_TEN,0;";  ! \ Initialize variables.
80  OUTPUT @Sa;"VARDEF YVAL,700;";  ! /
90  !
100 FOR I=100 TO 1000 STEP 100      ! Begin loop.
110     OUTPUT @Sa;"LOG LOG_TEN,"&VAL$(I)&",1;";
                                     !Take the LOG and
120     ! multiply by scaling factor of 1.
130     OUTPUT @Sa;"PU;PA 20,YVAL;";  ! Position "pen" for text.
140     OUTPUT @Sa;"TEXT %The LOG of %;"; ! Write text.
150     OUTPUT @Sa;"DSPLY "&VAL$(I)&",6,3;";
                                     ! Write a value.
160     OUTPUT @Sa;"TEXT % = %;";     ! Write text.
170     OUTPUT @Sa;"DSPLY LOG_TEN,5,3;"; ! Write a value.
180     OUTPUT @Sa;"SUB YVAL,YVAL,50;"; ! Calculate new "pen" position.
190 NEXT I                            ! End loop.
200 !
210 ASSIGN @Sa TO *                    ! Close I/O path.
220 END

```

CLRSCHED

Clear Autosave/Autoexec Schedule Buffer

Syntax



x*213a

Description Clears the Autosave/Autoexec Schedule of all or individually specified contents. Clear all schedule contents with the ALL command or selectively by specifying which number (from one to seven) to clear.

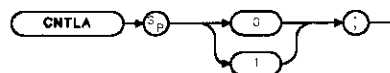
Program Example

```
10 ! The following example shows the use of the CLRSCHED command.
20 !
30 ASSIGN @Sa TO 718 ! Assign I/O path to address 718.
40 !
50 OUTPUT @Sa;"IP;"; ! Instrument preset.
60 ! Create three items for the AUTOEXEC Schedule.
70 OUTPUT @Sa;"AUTOFUNC 1 ,BOX,010189000000,010589000000 EOS;";
80 OUTPUT @Sa;"AUTOFUNC 2 ,CHK_SIG,123188235500,123188235900 INTVL 000030;";
90 OUTPUT @Sa;"AUTOFUNC 3 TRA,122488090000,122488125900 LMTST;";
100 PRINT "Press {LCL} [MODULE] {AUTOEXEC MENU} to see the new schedule."
110 DISP "Press [CONTINUE] on the computer when ready"
120 PAUSE
130 !
140 ! Remove one item from the schedule.
150 OUTPUT @Sa;"IP;CLRSCHED 2;";
160 PRINT "Press {LCL} [MODULE] {AUTOEXEC MENU} to see the updated schedule."
170 DISP "Press [CONTINUE] on the computer when ready"
180 PAUSE
190 !
200 ! Clear the entire schedule.
210 OUTPUT @Sa;"IP;CLRSCHED ALL;";
220 PRINT "Press {LCL} [MODULE] {AUTOEXEC MENU} to see the cleared schedule."
230 DISP ""
240 !
250 ASSIGN @Sa TO *
260 END
```

CNTLA

Auxiliary Control Line A

Syntax



XCNTLA

Description The CNTLA command* sets control line A of the auxiliary interface high or low.

Related commands: CNTLB*, CNTLC*, CNTLD*, CNTLI*, and OUTPUT*

Refer to "Specifications and Characteristics" in Chapter 1, Installation, for a detailed description of the auxiliary interface.

Preset State On

Program Example

```

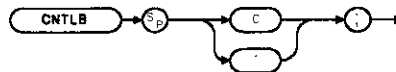
10  !The following example shows a use of the CNTLA command.
20  !
30  ASSIGN @Sa TO 718                !Assign I/O path to address 718.
40  !
50  OUTPUT @Sa;"VARDEF S_CNTLA,0;"   !CNTLA STATE
60  OUTPUT @Sa;"FUNCDEF AUX_CLR_A,@;"
70  OUTPUT @Sa;" CNTLA 0;"           !SET LINE A TO 0
80  OUTPUT @Sa;" MOV S_CNTLA,0;"     !UPDATE CNTLA IMAGE
90  OUTPUT @Sa;"@"
100 OUTPUT @Sa;"FUNCDEF AUX_SET_A,@;"
110 OUTPUT @Sa;" CNTLA 1;"           !SET LINE A TO 1
120 OUTPUT @Sa;" MOVE S_CNTLA,1;"    !UPDATE CNTLA IMAGE
130 OUTPUT @Sa;"@"
140 OUTPUT @Sa;"AUX_CLR_A;"          !CLEAR LINE A
150 OUTPUT @Sa;"S_CNTLA?;"          !GET LINE A STATUS
160 ENTER @Sa;A$
170 END
  
```

* These auxiliary interface connector commands can only be accessed remotely by way of the HPIB and DLPs and are only available with 85620A with serial prefix 3143A or higher and firmware 910116 and later date codes.

CNTLB

Auxiliary Control Line B

Syntax



XCNTLB

Description The CNTLB command* sets control line B of the auxiliary interface high or low.

Related commands: CNTLA*, CNTLC*, CNTLD*, CNTLI*, and OUTPUT*

Refer to "Specifications and Characteristics" in Chapter 1, Installation, for a detailed description of the auxiliary interface.

Preset State On

Program Example

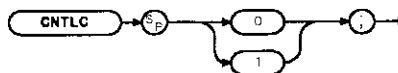
```
10    !The following example shows a use of the CNTLB command.
20    !
30    ASSIGN @Sa TO 718                !Assign I/O path to address 718.
40    !
50    OUTPUT @Sa;"VARDEF S_CNTLB,0;"    !CNTLB STATE
60    OUTPUT @Sa;"FUNCDEF AUX_CLR_B,@;"
70    OUTPUT @Sa;" CNTLB 0;"           !SET LINE B TO 0
80    OUTPUT @Sa;" MOV S_CNTLB,0;"     !UPDATE CNTLB IMAGE
90    OUTPUT @Sa;"@"
100   OUTPUT @Sa;"FUNCDEF AUX_SET_B,@;"
110   OUTPUT @Sa;" CNTLB 1;"           !SET LINE B TO 1
120   OUTPUT @Sa;" MOVE S_CNTLB,1;"    !UPDATE CNTLB IMAGE
130   OUTPUT @Sa;"@"
140   OUTPUT @Sa;"AUX_SET_B;"          !SET LINE B
150   OUTPUT @Sa;"S_CNTLB?;"          !GET LINE B STATUS
160   ENTER @Sa;A$
170   END
```

* These auxiliary interface connector commands can only be accessed remotely by way of the HPIB and DLPs and are only available with 85620A with serial prefix 3143A or higher and firmware 910116 and later date codes.

CNTLC

Auxiliary Control Line C

Syntax



XCNTLC

Description The CNTLC command* sets control line C of the auxiliary interface high or low.

Related commands: CNTLA*, CNTLB*, CNTLD*, CNTLI*, and OUTPUT*

Refer to "Specifications and Characteristics" in Chapter 1, Installation, for a detailed description of the auxiliary interface.

Preset State On

Program Example

```

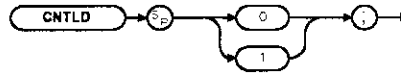
10  !The following example shows a use of the CNTLC command.
20  !
30  ASSIGN @Sa TO 718                !Assign I/O path to address 718.
40  !
50  OUTPUT @Sa;"VARDEF S_CNTLC,0;"    !CNTLC STATE
60  OUTPUT @Sa;"FUNCDEF AUX_CLR_C,@;"
70  OUTPUT @Sa;" CNTLC 0;"           !SET LINE C TO 0
80  OUTPUT @Sa;" MOV S_CNTLC,0;"     !UPDATE CNTLC IMAGE
90  OUTPUT @Sa;"@"
100 OUTPUT @Sa;"FUNCDEF AUX_SET_C,@;"
110 OUTPUT @Sa;" CNTLC 1;"           !SET LINE C TO 1
120 OUTPUT @Sa;" MOVE S_CNTLC,1;"    !UPDATE CNTLC IMAGE
130 OUTPUT @Sa;"@"
140 OUTPUT @Sa;"AUX_SET_C;"          !SET LINE C
150 OUTPUT @Sa;"S_CNTLC?;"          !GET LINE C STATUS
160 ENTER @Sa;A$
170 END
  
```

* These auxiliary interface connector commands can only be accessed remotely by way of the HPIB and DLPs and are only available with 85620A with serial prefix 3143A or higher and firmware 910116 and later date codes.

CNTLD

Auxiliary Control Line D

Syntax



XCN*LD

Description The CNTLD command* sets control line D of the auxiliary interface high or low.

Related commands: CNTLA*, CNTLB*, CNTLC*, CNTLI*, and OUTPUT*

Refer to "Specifications and Characteristics" in Chapter 1, Installation, for a detailed description of the auxiliary interface.

Preset State On

Program Example

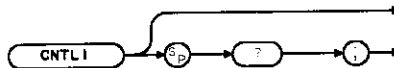
```
10 !The following example shows a use of the CNTLD command.
20 !
30 ASSIGN @Sa TO 718 !Assign I/O path to address 718.
40 !
50 OUTPUT @Sa;"VARDEF S_CNTLD,0;" !CNTLD STATE
60 OUTPUT @Sa;"FUNCDEF AUX_CLR_D,@;"
70 OUTPUT @Sa;" CNTLD 0;"; !SET LINE D TO 0
80 OUTPUT @Sa;" MOV S_CNTLD,0;"; !UPDATE CNTLD IMAGE
90 OUTPUT @Sa;"@";
100 OUTPUT @Sa;"FUNCDEF AUX_SET_D,@;"
110 OUTPUT @Sa;" CNTLD 1;"; !SET LINE D TO 1
120 OUTPUT @Sa;" MOVE S_CNTLD,1;"; !UPDATE CNTLD IMAGE
130 OUTPUT @Sa;"@";
140 OUTPUT @Sa;"AUX_SET_D;" !SET LINE D
150 OUTPUT @Sa;"S_CNTLD?;" !GET LINE D STATUS
160 ENTER @Sa;A$
170 END
```

* These auxiliary interface connector commands can only be accessed remotely by way of the GPIB and DLPs and are only available with 85620A with serial prefix 3143A or higher and firmware 910116 and later date codes.

CNTLI

Auxiliary Control Line Input

Syntax



XCNTLI

Description The CNTLI command* is a predefined variable used to read the control line I status of the auxiliary interface.

Related commands: CNTLA*, CNTLB*, CNTLC*, CNTLD*, OUTPUT*

Refer to "Specifications and Characteristics" in Chapter 1, Installation, for a detailed description of the auxiliary interface.

Program Example

```

10  !The following example shows a use of the CNTLI command.
20  ASSIGN @Sa TO 718                !Assign I/O path to address 718.
30  OUTPUT @Sa;FUNCDEF AUX_GET_CTRLI,@;";
40  OUTPUT @Sa;" LCLVAR I_VAL,0;";    !CONTROL VALUE TEMP
50  OUTPUT @Sa;" MOV I_VAL,CNTLI;";  !GET CONTROL I VALUE
60  OUTPUT @Sa;" EM; PU;PA 110,576;"; !DISPLAY VALUE ON SCREEN
70  OUTPUT @Sa;" TEXT 'CNTLI: ';";
80  OUTPUT @Sa;" DISPLY I_VAL,2,0;";
90  OUTPUT @Sa;"@";
100 OUTPUT @Sa;"AUX_GET_CTRLI;";
110 END

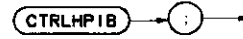
```

* These auxiliary interface connector commands can only be accessed remotely by way of the HPIB and DLPS and are only available with 85620A with serial prefix 3143A or higher and firmware 910116 and later date codes.

CTRLHPIB

Control HP-IB

Syntax



xf214g

Description

This command takes control of the HP-IB. If a controller is active and detected on the bus, then the command is not executed and an error results. If none is detected, the spectrum analyzer assumes control by asserting the remote-enable (REN) line. This command should precede the related commands RELHPIB, ENTER, and OUTPUT.

Program Example

```

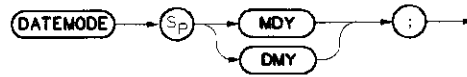
10      ! The following example shows how to use the CTRLHPIB command to have
20      ! the analyzer send data to another HP-IB device. The instructions
30      ! within the FUNCDEF '@' delimiters form a structure called a
40      ! DLP (downloadable program).
50      !
60      ! NOTE: A printer (HP-IB address 01) needs to be connected.
70      !
80      ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
90      !
100     OUTPUT @Sa;"FUNCDEF E_XMPLE,@;"; ! Logical start of the DLP.
110     OUTPUT @Sa;"  CTRLHPIB;";       ! Tells analyzer to take control
120                                     ! of the bus.
130     OUTPUT @Sa;"  MKN;";           ! Turn on a marker.
140     OUTPUT @Sa;"  OUTPUT 1,KC,'THIS IS AN EXAMPLE OF THE ANALYZER
SENDING';";
150     OUTPUT @Sa;"  OUTPUT 1,KC,'DATA TO ANOTHER HPIB DEVICE';";
160     OUTPUT @Sa;"  OUTPUT 1,KC,",";";
170     OUTPUT @Sa;"  OUTPUT 1,K,'MARKER FREQUENCY = '";";
180     OUTPUT @Sa;"  OUTPUT 1,K,MKF;";
190     OUTPUT @Sa;"  OUTPUT 1,KC,' Hz';";
200     OUTPUT @Sa;"  OUTPUT 1,KC,",";";
210     OUTPUT @Sa;"  RELHPIB;";       ! Release HP-IB control.
220     OUTPUT @Sa;"@";               ! Logical end of the DLP.
230     !
240     OUTPUT @Sa;"E_XMPLE;"         ! Have the analyzer execute the
250                                     ! DLP E_XMPLE.
260     SEND 7;UNL TALK 18 LISTEN 1 DATA ! Send to HP-IB select code 7 the
270                                     ! commands necessary to allow
280                                     ! the analyzer (address 18) to
290                                     ! talk to the printer (address 1)
300                                     ! while this controller is still
310                                     ! connected.
320     LOCAL 7                        ! Local HP-IB select code 7.
330     WAIT 2                          ! Wait for printer to finish.
340     REMOTE 7                        ! Toggle the HP-IB REN line.
350     LOCAL @Sa                      ! Local the SA.
360     !
370     ASSIGN @Sa TO *                ! Close I/O path.
380     END

```

DATEMODE

Date Mode

Syntax



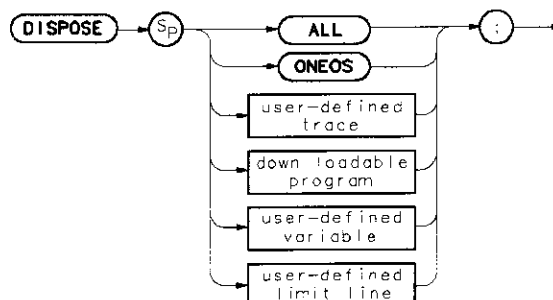
x1215a

Description Used to set European (DDMMYY) or United States (MMDDYY) date format. This command cannot be queried.

DISPOSE

Dispose

Syntax



x*216a

Description Dispose allows the user to free module RAM previously allocated for user-defined functions. These functions include traces, user-defined variables, DLPs, and limit-lines. The DISPOSE command will search for the first occurrence of the indicated file name and delete it. The command has a file search hierarchy; it searches for different types of files in the order shown in Table 4-3. This table also shows examples of using the DISPOSE command for each file type.

Table 4-3.
Dispose Command File Search Hierarchy and Programming Examples

File Hierarchy	File Type	Programming Example
1.	Trace	DISPOSE TRACE.961225101522.TRC
2.	User-Defined Variable	DISPOSE D_DATA.VAR
3.	DLP (Down-Loadable Program)	DISPOSE F_PRCTAM.DLP
4.	Limit Line	DISPOSE B_CDMA.LIM

File extensions such as .TRC, .VAR, and .DLP shown in the programming examples in Table 4-3 are not required, as long as the file name is unique. However, using extensions help reduce the possibility that an incorrect file type is deleted. The DISPOSE command chooses the first filename match in its hierarchy and ignores the extension if the extension is omitted in the DISPOSE command.

DISPOSE

A timestamp is included as part of the filename for a trace. If there are multiple traces with the same filename, the timestamp can be used to identify each trace individually. When the DISPOSE command is used to dispose of a trace, the timestamp must be included with the trace filename or it will dispose of the first trace that it finds with that filename. The CATALOG command can be used to obtain timestamps for traces.

The dispose command file search was limited to DLP files in earlier revisions of firmware.

The DISPOSE ALL command deletes all existing files and makes available the maximum amount of memory.

Program Example

```

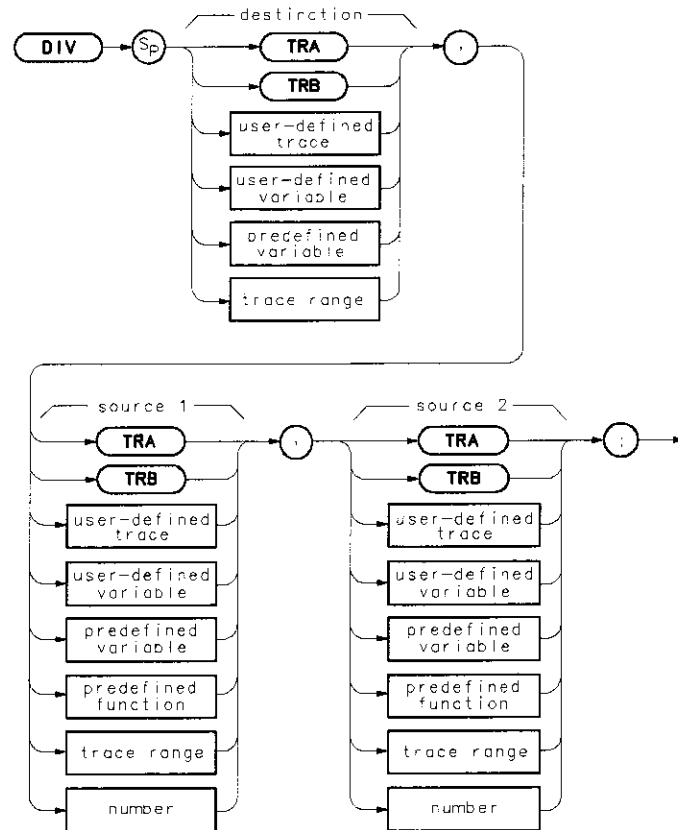
10  ! The following example shows the use of the DISPOSE command. First it
20  ! clears memory. Then it creates a DLP and a variable. Finally, it deletes
30  ! the variable. Since this example disposes EVERYTHING currently downloaded
40  ! into the memory module, it should be used with caution.
50  !
60  ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
70  !
80  OUTPUT @Sa;"IP;DISPOSE ALL;";
90  PRINT "DISPOSE ALL executed. Press {LCL} [MODULE] {UTILITY} to"
100 PRINT "observe that there are no applicable entries."
110 DISP "Press [CONTINUE] on computer when ready"
120 PAUSE
130 !
140 OUTPUT @Sa;"FUNCDEF TST,@;";      ! Create a DLP function.
150 OUTPUT @Sa;"VARDEF NUMBER,999;";  ! Create, initialize a real variable.
160 OUTPUT @Sa;"ADD NUMBER,NUMBER,1;";
170 OUTPUT @Sa;"@";
180 !
190 OUTPUT @Sa;"IP;TST;";             ! Execute the DLP creating.
200 PRINT "Press {LCL} [MODULE] {UTILITY} to observe that there are"
210 PRINT "now two entries."
220 DISP "Press [CONTINUE] on computer when ready"
230 PAUSE
240 !
250 OUTPUT @Sa;"IP;DISPOSE NUMBER;";
260 PRINT "Press {LCL} [MODULE] {UTILITY} to observe that there is"
270 PRINT "now ONE entry. The variable NUMBER has been disposed."
280 !
290 ASSIGN @Sa TO *                  ! Close I/O path.
300 END

```

DIV

Divide

Syntax



x12:17a

Description

Divides source 1 by source 2, then stores the results in the destination. A divide-by-zero attempt generates an error, attaches the sign of the value contained in source 1 to the maximum value, and sends it to the destination.

Program Example

```

10  ! The following example shows the use of the DIV command.  The example
20  ! uses an input signal of 300MHz, such as the CAL OUTPUT.
30  ! An input signal is not necessary to the function of
40  ! this example, but acts as a visual aid.
50  !
60  ASSIGN @Sa TO 718          ! Assign I/O path to address 718.
70  !
80  OUTPUT @Sa;"FUNCDEF DIVEX,@"; ! Logical start of the DLP function.
100 OUTPUT @Sa;"TRDEF ARY,300";  ! Create 300 element trace.
110 OUTPUT @Sa;"IP; CF 300MHZ;SP 1MHZ;";
                                ! Instrument preset; set center
120                                ! frequency and span.
130 OUTPUT @Sa;"CLRW TRA; TS;";  ! Clear trace A and take a sweep.
140 OUTPUT @Sa;"VIEW TRA;";      ! View trace A.
150 OUTPUT @Sa;"MOV ARY,TRA;";   ! Move first 300 points of trace A
160                                ! into user-defined trace.
170 OUTPUT @Sa;"DIV TRA,ARY,2;"; ! Divide user-defined trace by 2 and
180                                ! place result in trace A.
190 OUTPUT @Sa;"VIEW TRA;";      ! View result.
200 OUTPUT @Sa;"@";             ! Logical end of DLP function.
210 !
220 OUTPUT @Sa;"DIVEX;";         ! Execute function.
230 !
240 ASSIGN @Sa TO *            ! Close I/O path.
250 END

```

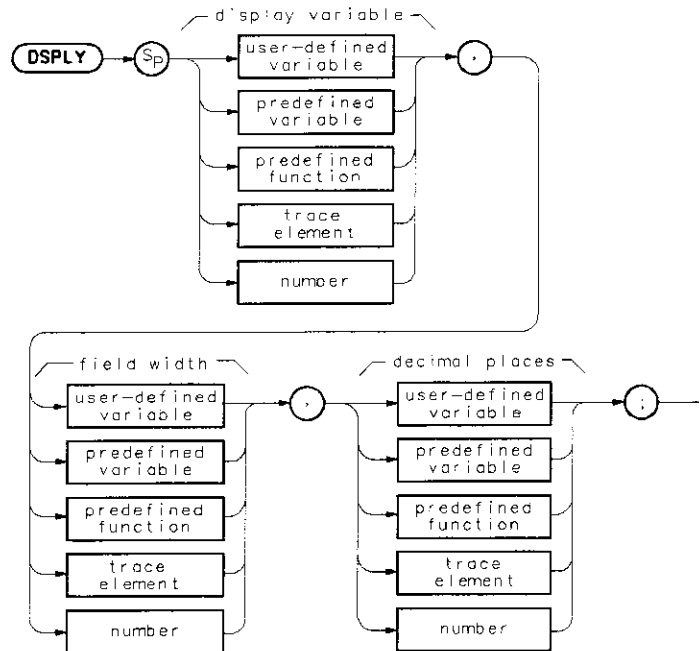
Note

Since in this example the 300-point user-defined trace called ARY is smaller than the 600-point trace A, the last value in the user-defined trace is used as an operand on the remaining elements of trace A.

DSPLY

Display Variable

Syntax



x'218a

Description Displays the value of the variable at the current position of the graphics pen. Refer to the OR, PA, PR, PU, or PD commands. The variable is displayed according to the field-width specifier and the decimal places assigned to the current position. Field width is made up of all digits, including the sign and decimal point. If the variable value is too large to fit into the field-width and decimal-places specification, exponential notation is used.

Note For the OR command to function properly in conjunction with the DSPLY command, you must use PU or PA after OR to position the graphics pen correctly.

Prerequisite Command: LCLVAR or VARDEF when using a user-defined variable, user-defined trace or user-defined array.

Related Commands: CLRDSP, PA, and PR

Field Width: 1 to 16

Decimal-Place Range: Minimum = 0. Maximum = if Field Width is >3, then maximum is Field Width - 3, otherwise it is 0.

Program Example

```

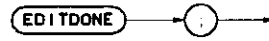
10  ! The following example shows the use of the DSPLY command. The example
20  ! uses values from HPBASIC and concatenates them into the statements as
30  ! they are sent to the spectrum analyzer.
40  !
50  ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
60  !
70  OUTPUT @Sa;"IP;CLRDSP;";         ! Clears the analyzer display.
80  !
90  OUTPUT @Sa;"VARDEF LOG_TEN,0;";  ! \ Initialize variables.
100 OUTPUT @Sa;"VARDEF YVAL,700";    ! /
110 !
120 FOR I=100 TO 1000 STEP 100       ! Begin loop.
130     OUTPUT @Sa;"LOG LOG_TEN,"&VAL$(I)&","1;";
                                         ! Take the LOG and multiply
140                                         ! by scaling factor of 1.
150     OUTPUT @Sa;"PU;PA 20,YVAL;";  ! Position "pen" for text.
160     OUTPUT @Sa;"TEXT %The LOG of %;"; ! Write text.
170     OUTPUT @Sa;"DSPLY "&VAL$(I)&","6,3;";
                                         ! Write a value.
180     OUTPUT @Sa;"TEXT % = %;";      ! Write text.
190     OUTPUT @Sa;"DSPLY LOG_TEN,5,3;"; ! Write a value.
200     OUTPUT @Sa;"SUB YVAL,YVAL,50;"; ! Calculate new "pen" position.
210 NEXT I                            ! End loop.
220 !
230 ASSIGN @Sa TO *                    ! Close I/O path.
240 END

```

EDITDONE

Limit-Line Edit Done

Syntax



x1219a

Description This command is used at the completion of limit-line editing with the EDITLIML command.

Program Example

```

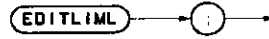
130 ! This program demonstrates the use of the EDITDONE command
140 ! to generate a limit line:
150 !
160 OUTPUT 718;"LIMIPURGE;"; ! Purge current limit line.
170 OUTPUT 718;"EDITLIML;"; ! Begin editing limit line.
180 OUTPUT 718;"LIMIREL OFF;"; ! Make sure it is an absolute limit line.
190 OUTPUT 718;"SADD;"; ! Add segment to limit line.
200 OUTPUT 718;"SEDI 1;"; ! Edit the first limit line.
210 OUTPUT 718;"LIMF 250MHZ;"; ! Set the frequency of the segment.
220 OUTPUT 718;"LIMU -35DBM;"; ! Provide the upper amplitude.
230 OUTPUT 718;"LIML -80DBM;"; ! Enter the lower amplitude.
240 OUTPUT 718;"LIMTFL;"; ! Set type of segment to flat.
250 OUTPUT 718;"SDON;"; ! Enter parameters into table.
260 OUTPUT 718;"SADD;SEDI 2;"; ! Add, then edit second segment.
270 OUTPUT 718;"LIMF 290MHZ;"; ! Set the frequency of segment 2.
280 OUTPUT 718;"LIMU -35DBM;LIML -80DBM;";
! Set amplitude data for segment.
290 OUTPUT 718;"LIMTSL;"; ! Set segment type to slope.
300 OUTPUT 718;"SDON;"; ! Enter parameters into table.
310 OUTPUT 718;"SADD;SEDI 3;";
320 OUTPUT 718;"LIMF 296.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;";
330 OUTPUT 718;"SDON;";
340 OUTPUT 718;"SADD;SEDI 4;";
350 OUTPUT 718;"LIMF 300MHZ;LIMU -5DBM;LIML -25DBM;LIMTSL;";
360 OUTPUT 718;"SDON;";
370 OUTPUT 718;"SADD;SEDI 5;";
380 OUTPUT 718;"LIMF 303.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;";
390 OUTPUT 718;"SDON;";
400 OUTPUT 718;"SADD;SEDI 6;";
410 OUTPUT 718;"LIMF 304.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;";
420 OUTPUT 718;"SDON;";
430 OUTPUT 718;"SADD;SEDI 7;";
440 OUTPUT 718;"LIMF 310MHZ;LIMU -35DBM;LIML -80DBM;LIMTFL;";
450 OUTPUT 718;"SDON;";
460 OUTPUT 718;"SADD;SEDI 8;";
470 OUTPUT 718;"LIMF 350MHZ;LIMU -35DBM;LIML -80DBM;LIMTFL;";
480 OUTPUT 718;"SDON;";
490 OUTPUT 718;"EDITDONE;"; ! End of limit-line definition.
500 OUTPUT 718;"CF 300MHZ;SP 50MHZ;RB 2MHZ;VB 30KHZ;TS;";
! Set up analyzer to display
505 ! the limit line.
510 END

```

EDITLIML

Edit Limit Line

Syntax



xr220a

Description

This command turns off the currently active limit line, then places you in limit-line edit mode. Use this command with the commands SEDI and SADD to call up a limit-line segment for editing. The editing of each segment is terminated with SDON. EDITLIML is terminated with the EDITDONE command.

Related commands are SENTER and SEDI.

Program Example

```

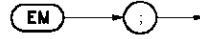
130  ! This program demonstrates the use of the EDITLIML command
140  ! to generate a limit line:
150  !
160  OUTPUT 718;"LIMIPURGE;";          ! Purge current limit line.
170  OUTPUT 718;"EDITLIML;";          ! Begin editing limit line.
180  OUTPUT 718;"LIMIREL OFF;";       ! Make sure it is an absolute limit line.
190  OUTPUT 718;"SADD;";              ! Add segment to limit line.
200  OUTPUT 718;"SEDI 1;";            ! Edit the first limit line.
210  OUTPUT 718;"LIMF 250MHZ;";       ! Set the frequency of the segment.
220  OUTPUT 718;"LIMU -35DBM;";       ! Provide the upper amplitude.
230  OUTPUT 718;"LIML -80DBM;";       ! Enter the lower amplitude.
240  OUTPUT 718;"LIMTFL;";            ! Set type of segment to flat.
250  OUTPUT 718;"SDON;";              ! Enter parameters into table.
260  OUTPUT 718;"SADD;SEDI 2;";       ! Add, then edit second segment.
270  OUTPUT 718;"LIMF 290MHZ;";       ! Set the frequency of segment 2.
280  OUTPUT 718;"LIMU -35DBM;LIML-80DBM;";
                                         ! Set amplitude data for segment.
290  OUTPUT 718;"LIMTSL;";            ! Set segment type to slope.
300  OUTPUT 718;"SDON;";              ! Enter parameters into table.
310  OUTPUT 718;"SADD;SEDI 3;";
320  OUTPUT 718;"LIMF 296.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;";
330  OUTPUT 718;"SDON;";
340  OUTPUT 718;"SADD;SEDI 4;";
350  OUTPUT 718;"LIMF 300MHZ;LIMU -5DBM;LIML -25DBM;LIMTSL;";
360  OUTPUT 718;"SDON;";
370  OUTPUT 718;"SADD;SEDI 5;";
380  OUTPUT 718;"LIMF 303.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;";
390  OUTPUT 718;"SDON;";
400  OUTPUT 718;"SADD;SEDI 6;";
410  OUTPUT 718;"LIMF 304.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;";
420  OUTPUT 718;"SDON;";
430  OUTPUT 718;"SADD;SEDI 7;";
440  OUTPUT 718;"LIMF 310MHZ;LIMU -35DBM;LIML -80DBM;LIMTFL;";
450  OUTPUT 718;"SDON;";
460  OUTPUT 718;"SADD;SEDI 8;";
470  OUTPUT 718;"LIMF 350MHZ;LIMU -35DBM;LIML -80DBM;LIMTFL;";
480  OUTPUT 718;"SDON;";
490  OUTPUT 718;"EDITDONE;";          ! End of limit-line definition.
500  OUTPUT 718;"CF 300MHZ;SP 50MHZ;RB 2MHZ;VB 30KHZ;TS;";
                                         ! Set up analyzer to display
505                                     ! the limit line.
510  END      !

```

EM

Erase User Display Memory

Syntax



xf221a

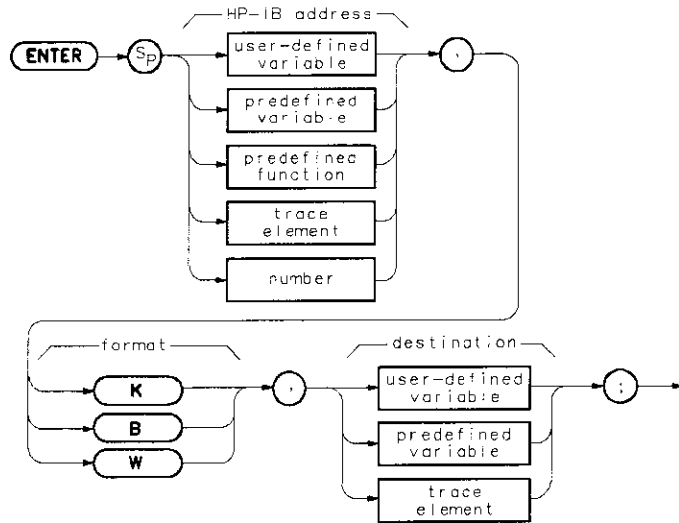
Description

This command can be used to clear the display of user-generated graphics. EM does not erase spectrum analyzer display annotation. Refer to the CLRDSP command as well.

ENTER

Enter from HP-IB

Syntax



x f222a

Description

This command allows a DLP, or user-defined function, to enter data over HP-IB. If the spectrum analyzer is not the controller, the command aborts. The controller that is active and detected on the bus has control. Use the CTRLHPIB command to gain control of the HP-IB, then RELHPIB after the ENTER command to relinquish HP-IB control. The data entered is formatted as specified by the format field of the syntax diagram. This format may be one of the following:

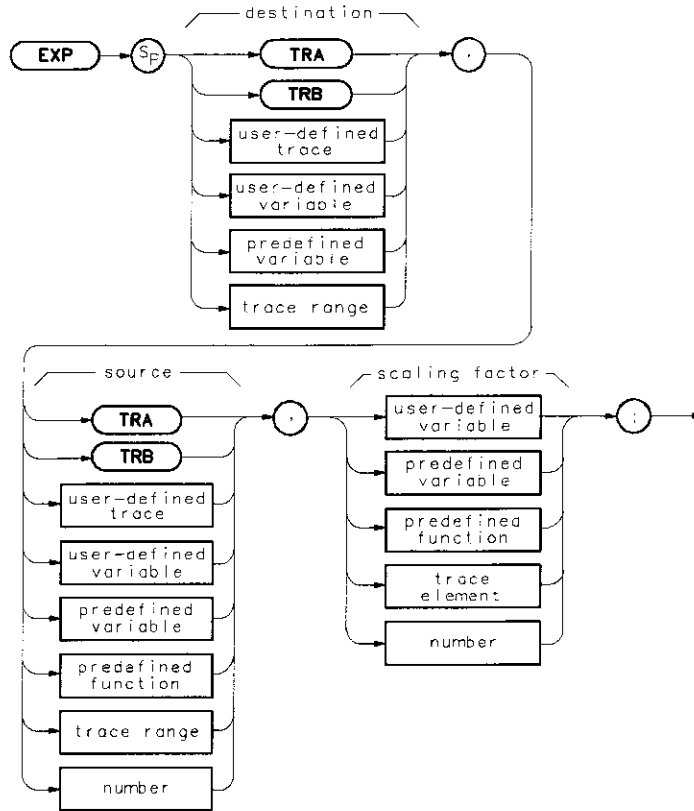
- K, a free field (ASCII real-number format)
- B, one binary byte
- W, one binary word (two bytes)

Prerequisite Commands: VARDEF and LCLVAR when using user-defined variable, user-defined trace, or user-defined array.

EXP

Exponent

Syntax



x.f223a

Description The source is divided by the scaling factor and the result is raised to a power of 10, then stored in the destination.

Parameters

$\langle \text{scaling factor} \rangle ::= \langle \text{variable identifier} \rangle | \langle \text{numeric data field} \rangle$
 $\langle \text{destination} \rangle = 10 \langle \text{source} \rangle / \langle \text{scaling factor} \rangle$

Program Example

```

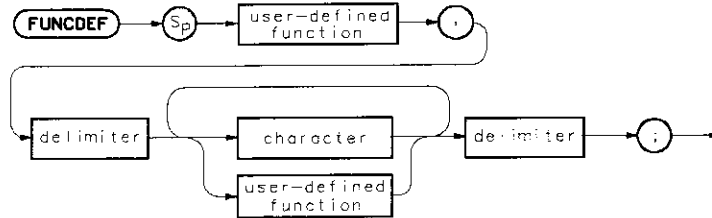
10    ! The following example shows the use of the EXP command.  This example
20    ! calculates the power in milliwatts of the largest signal on the analyzer
30    ! CRT.  Initial settings are sent outside the function to set up the
40    ! CAL OUTPUT signal on the CRT.  These settings act as a visual aid.
50    !
60    ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
70    !
80    OUTPUT @Sa;"IP; CF 300MHZ; SP 100MHZ;";
                                         ! CAL OUTPUT signal to center-screen.
90    !
100   OUTPUT @Sa;"FUNCDEFDBM_TO_MW,@;"; ! Logical start of DLP.
110   OUTPUT @Sa;"  LCLVAR PWR_MW,0;";
                                         ! Define local variable and
                                         ! initialize to 0.
120   OUTPUT @Sa;"  TS;MKPK HI;";      ! Take sweep; marker to peak;
130   OUTPUT @Sa;"  EXP PWR_MW,MKA,10;"; ! pwr_mw = 10^(mka/10).
150   OUTPUT @Sa;"  PU;PA 100,650;PD;"; ! Move to starting position.
160   OUTPUT @Sa;"  TEXT %POWER (mW)=%;"; ! Write text.
170   OUTPUT @Sa;"  DSPLY PWR_MW,5,3;"; ! Write results.
180   OUTPUT @Sa;"@";                  ! Logical end of DLP.
190   !
200   OUTPUT @Sa;"DBM_TO_MW;";         ! Execute DLP.
210   !
220   ASSIGN @Sa TO *                  ! Close I/O path.
230   END

```

FUNCDEF

Function Definition

Syntax



xf224a

Description

This command allows you to define a program which is identified as a user-defined function or a DLP. If the user-defined function label is the same as a command mnemonic, an error results and the command is ignored.

This command requires user memory in the module to execute. Memory is allocated by executing this function and becomes free user memory with the DISPOSE command.

Restrictions: Subject to available memory. The following commands consume memory: ONEOS, ARRAYDEF, TRDEF, KEYDEF, LCLVAR, VARDEF REPEAT/UNTIL, and IF/THEN.

Related Commands: ABORT, RETURN, ERASE, and DISPOSE

Program Example

```

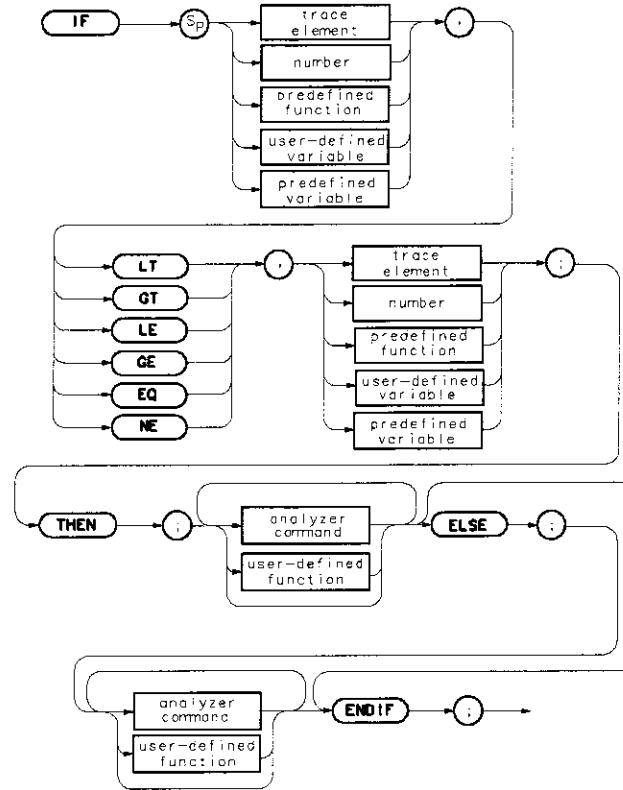
10  ! The following example shows the use of the FUNCDEF command.  The example
20  ! uses an input signal of 300 MHz, such as the CAL OUTPUT, and looks at
30  ! its harmonics.  An input signal is not necessary to the function of
40  ! this example, but acts as a visual aid.
50  !
60  ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
70  !
80  OUTPUT @Sa;"FUNCDEF INCRFREQ,@"; ! Logical start of the DLP function.
90  OUTPUT @Sa;"VARDEF FREQ,300E6";  ! Create variable and initialize
                                     ! to 300 MHz.
100 OUTPUT @Sa;"IP;SP 1MHZ;";       ! Instrument preset.
110 !
120 OUTPUT @Sa;"REPEAT;";           ! Begin loop.
130 OUTPUT @Sa;"  MOV CF,FREQ;";    ! Set center frequency.
140 OUTPUT @Sa;"  TS;";             ! Take a sweep to update display.
150 OUTPUT @Sa;"  ADD FREQ,FREQ,300E6;";
                                     ! Increase FREQ by 300 MHz.
160 OUTPUT @Sa;"UNTIL FREQ,GT,3E9;"; ! End of loop.
170 OUTPUT @Sa;"@";                 ! Logical end of DLP function.
180 !
190 !
200 OUTPUT @Sa;"  INCRFREQ;";       ! Execute function.
210 !
220 ASSIGN @Sa TO *                 ! Close I/O path.
230 END

```

IF/THEN/ELSE/ENDIF

If/Then/Else/Endif

Syntax



v1225a

Description

The IF/THEN/ELSE/ENDIF statement combination allows the comparison of two operands to a condition. If the condition is true, the command list following THEN is executed. If false, commands following either the next ELSE or ENDIF statements are executed. Valid conditions are less than (LT), greater than (GT), less than or equal to (LE), greater than or equal to (GE), equal (EQ), and not equal (NE).

- THEN** The THEN command is treated as a no-operation function, but is required for user program flow purposes.
- ELSE** The ELSE command delimits the alternate condition of an IF command.
- ENDIF** The ENDIF command delimits the end of a conditional command sequence.

Program Example

```

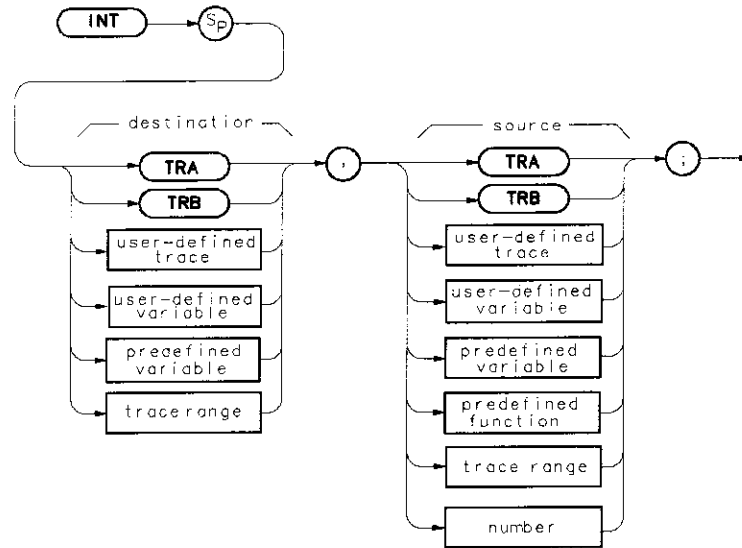
10  !The following example shows the use of the IF/THEN/ELSE/ENDIF command.
20  !The instructions within the FUNCDEF '@' delimiters form a structure
30  !called a DLP (downloadable program).
40  !
50  ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
60  !
70  OUTPUT @Sa;"FUNCDEF TST,@;"      ! Logical start of the DLP.
80  OUTPUT @Sa;"VARDEF Y,500;"       ! Create variable and initialize to 500.
90  OUTPUT @Sa;"VARDEF X,500;"       ! Create variable and initialize to 500.
100 OUTPUT @Sa;"CLRDISP;"           ! Clear display.
110 OUTPUT @Sa;"REPEAT;"            ! Begin loop.
120 OUTPUT @Sa;" IF Y,LT,100; THEN;" ! Test condition.
                                     ! Test condition.
130 OUTPUT @Sa;"  PU;PA 100,X;"      ! Move pen.
140 OUTPUT @Sa;"  TEXT%DLP ABORTED%;" ! Print text.
                                     ! Print text.
150 OUTPUT @Sa;"  ABORT;"
160 OUTPUT @Sa;" ELSE;"
170 OUTPUT @Sa;"  PU;PA 100,X;"      ! Move pen.
180 OUTPUT @Sa;"  DSPLY Y,5,2;"      ! Display 'Y'.
190 OUTPUT @Sa;"  SUB X,X,40;"       ! Decrement pen pointer.
200 OUTPUT @Sa;"  SUB Y,Y,100;"      ! Decrement variable.
210 OUTPUT @Sa;" ENDIF;"
220 OUTPUT @Sa;"UNTIL Y,EQ,-100;"    ! End loop (The abort will occur
230                                     ! before this condition will be
240                                     ! satisfied.
250 OUTPUT @Sa;"@"                  ! Logical end of DLP.
260 !
270 OUTPUT @Sa;"TST;"                ! Execute DLP.
280 !
290 ASSIGN @Sa TO *                  ! Close I/O path.
300 END

```

INT

Integer

Syntax



v f226a

Description Stores in the destination the greatest integer number which is less than or equal to the real number in the source. Since traces and trace ranges consist of integers only, using them as source data in the INT command yields the same result as using them as source data with the MOV command.

Program Example

```

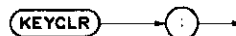
10  ! The following example shows the use of the INT command. The real
20  ! value is truncated by the INT command.
30  !
40  ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
50  !
60  OUTPUT @Sa;"VARDEF REAL_NUM,3.14;";
                                   ! Create and initialize a real variable.
70  OUTPUT @Sa;"VARDEF INT_NUM,0;";
                                   ! Create and initialize a real variable.
80  OUTPUT @Sa;"INT INT_NUM,REAL_NUM;";
                                   ! Put the integer portion of the
90                                   ! real number into INT_NUM.
100 OUTPUT @Sa;"REAL_NUM?";        ! Query REAL_NUM.
110 ENTER @Sa;Real_num              ! Get contents of variable.
120 OUTPUT @Sa;"INT_NUM?";        ! Query INT_NUM.
130 ENTER @Sa;Int_num              ! Get contents of variable.
140 !
150 PRINT "REAL=";Real_num,"INTEGER=";Int_num
                                   ! Print contents of variables
                                   ! to computer CRT.
160
170 !
180 ASSIGN @Sa TO *                ! Close I/O path.
190 END

```

KEYCLR

Clear User Defined Keys

Syntax



x1227a

Description Use this command to clear all the softkeys in the module User Keys menu.

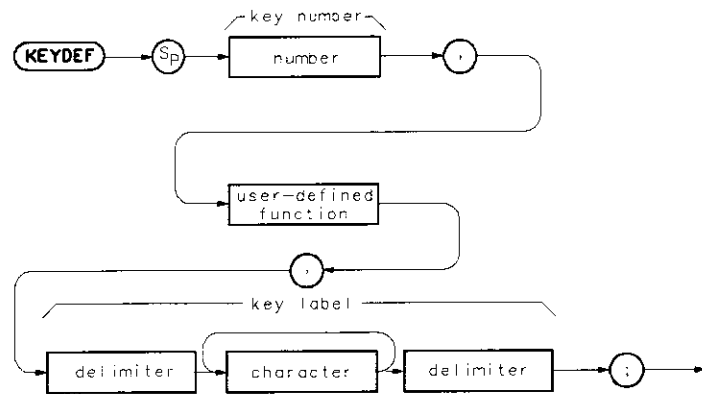
Program Example

```
10 ! The following example shows the use of the KEYCLR command.
20 !
30 ASSIGN @Sa TO 718 ! Assign I/O path to address 718.
40 !
50 OUTPUT @Sa;"KEYCLR;"; ! Clear all user-defined keys.
60 ASSIGN @Sa TO * ! Close I/O path.
70 END
```

KEYDEF

User Defined Key Definition

Syntax



x1228a

Description KEYDEF assigns a DLP (function label) to a USER KEY (key number) and labels the key with the function label. Pressing the softkey executes the function or DLP. There are 10 softkeys available to label. Softkey labels may be up to two lines long containing eight characters in each line.

Parameters Minimum: 1
Maximum: 10

Query Response The query returns the function or softkey name.

KEYDEF

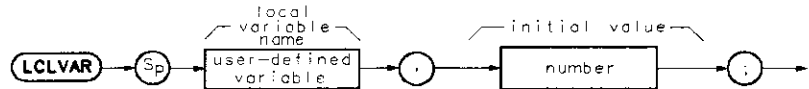
Program Example

```
10    ! The following example shows the use of the KEYDEF command.
20    !
30    ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
40    OUTPUT @Sa;"KEYDEF 2,BOX,%DRAW BOX%";
                                         ! Define softkey 2 to execute the
                                         ! DLP BOX when key is pressed.
50                                         ! Give the key a label 'DRAW BOX'.
60                                         !
70    !
80    OUTPUT @Sa;"FUNCDEF BOX,@";      ! Logical start of function.
90    OUTPUT @Sa;" IP;";               ! Instrument preset.
100   OUTPUT @Sa;" CLRDSP;";          ! Clear the analyzer screen.
110   !
120   OUTPUT @Sa;" PU;PA 300,300;";
                                         ! Move pen to starting point.
130   OUTPUT @Sa;" PD;PR 240,0;";      ! \
140   OUTPUT @Sa;" PR 0,240;";         ! \ Draw
150   OUTPUT @Sa;" PR -240,0;";       ! / Rectangle
160   OUTPUT @Sa;" PR 0,-240;";       ! /
170   OUTPUT @Sa;"@";                 ! Logical end of function.
180   !
190   ASSIGN @Sa TO *                  ! Close I/O path.
200   END
```

LCLVAR

Local Variable

Syntax



xf228a

Description A local variable is a variable that is defined within a FUNCDEF. It is recognized only when the FUNCDEF which defined it is running. Local variables have no meaning in a program outside the FUNCDEF that defined it, but are recognized by any FUNCDEF that is called up by the defining FUNCDEF.

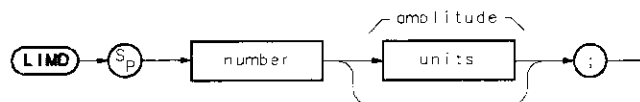
Note In Mass Memory Modules with firmware datecode 910116 and later, the value of a user-defined variable can be modified using the secondary keyword EP. Refer to “Using EP to Modify User-Defined Variables (firmware revision 910116 and later)” in this chapter.

Memory is allocated for the local-variable operation only as long as its defining FUNCDEF is running. Memory used by the local variable is freed at the completion of the defining FUNCDEF.

LCLVAR

Program Example

```
10  ! The following example shows the use of the LCLVAR command. This example
20  ! calculates the power in milliwatts of the largest signal on the analyzer
30  ! CRT. Initial settings are sent outside the function to set up the
40  ! CAL OUTPUT signal on the CRT. These settings act as a visual aid.
50  !
60  ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
70  !
80  OUTPUT @Sa;"IP; CF 300MHZ; SP 100MHZ;";
                                   ! CAL OUTPUT signal to center screen.
90  !
100 OUTPUT @Sa;"FUNCDEFDBM_TO_MW,@;"; ! Logical start of DLP.
110 OUTPUT @Sa;" LCLVAR PWR_MW,0;";
                                   ! Define local variable and
                                   ! initialize to 0.
120 OUTPUT @Sa;" TS;MKPK HI;";      ! Take sweep; marker to peak;
130 OUTPUT @Sa;" EXP PWR_MW,MKA,10;"; ! pwr_mw = 10^(mka/10)
150 OUTPUT @Sa;" PU;PA 100,650;PD;"; ! Move to starting position.
160 OUTPUT @Sa;" TEXT %POWER (mW)=%;"; ! Write text.
170 OUTPUT @Sa;" DSPLY PWR_MW,5,3;"; ! Write results.
180 OUTPUT @Sa;"@;";                ! Logical end of DLP.
190 !
200 OUTPUT @Sa;"DBM_TO_MW;";        ! Execute DLP.
210 !
220 ASSIGN @Sa TO *                  ! Close I/O path.
230 END
```

LIMD**Limit-Line Delta****Syntax**

x1230o

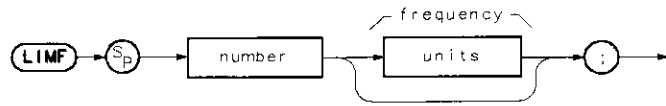
Description Use this command to enter the delta value for the amplitude of a limit-line segment. Related commands are EDITLIML and SEDI. This command is used along with LMM to define the deviation, both positive and negative, from a middle value.

Parameter Number Range: -175 dB to 50 dB

LIMD

Program Example

```
120 ! This program demonstrates the use of the LIMD command
130 ! to generate a limit line:
150 !
160 OUTPUT 718;"LIMIPURGE;";      ! Purge current limit line.
170 OUTPUT 718;"EDITLIML;";      ! Begin edit of limit line.
180 OUTPUT 718;"LIMIREL OFF;";    ! Make sure it's an absolute limit line.
190 OUTPUT 718;"SADD;";          ! Add segment to the limit line.
200 OUTPUT 718;"SEDI 1;";        ! Edit the first segment.
210 OUTPUT 718;"LIMF 250MHZ;";    ! Set frequency of the segment.
220 OUTPUT 718;"LIMU -35DBM;";    ! Provide the upper amplitude.
230 OUTPUT 718;"LIML -80DBM;";   ! Enter the lower amplitude.
240 OUTPUT 718;"LIMTFL;";        ! Set type of segment to flat.
250 OUTPUT 718;"SDON;";          ! Enter parameters into table.
260 OUTPUT 718;"SADD;SEDI 2;";   ! Add, then edit second segment.
270 OUTPUT 718;"LIMF 290MHZ;";   ! Set frequency of segment 2.
280 OUTPUT 718;"LIMU -35DBM;LIML-80DBM;";
                                   ! Set amplitude data for segment.
290 OUTPUT 718;"LIMTSL;";        ! Set segment type to slope.
300 OUTPUT 718;"SDON;";          ! Enter parameters into table.
310 OUTPUT 718;"SADD;SEDI 3;";
320 OUTPUT 718;"LIMF 296.5MHZ;LIMU -5DBM;LIML-80DBM;LIMTSL;";
330 OUTPUT 718;"SDON;";
340 OUTPUT 718;"SADD;SEDI 4;";
350 OUTPUT 718;"LIMF 300MHZ;LIMM -15DBM;LIMD 10DBM;LIMTSL;";
360 OUTPUT 718;"SDON;";
370 OUTPUT 718;"SADD;SEDI 5;";
380 OUTPUT 718;"LIMF 303.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;";
390 OUTPUT 718;"SDON;";
400 OUTPUT 718;"SADD;SEDI 6;";
410 OUTPUT 718;"LIMF 304.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;";
420 OUTPUT 718;"SDON;";
430 OUTPUT 718;"SADD;SEDI 7;";
440 OUTPUT 718;"LIMF 310MHZ;LIMU -35DBM;LIML -80DBM;LIMTFL;";
450 OUTPUT 718;"SDON;";
460 OUTPUT 718;"SADD;SEDI 8;";
470 OUTPUT 718;"LIMF 350MHZ;LIMU -35DBM;LIML -80DBM;LIMTFL;";
480 OUTPUT 718;"SDON;";
490 OUTPUT 718;"EDITDONE;";      ! End of limit-line definition.
500 OUTPUT 718;"CF 300MHZ;SP 50MHZ;RB 2MHZ;VB 30KHZ;TS;";
                                   ! Set up analyzer to display the
505                                     ! limit line.
510 END
```

LIMF**Limit-Line Frequency****Syntax**

xf231a

Description Use this command to enter a frequency value for a limit-line segment. This command is used along with the SEDI command while editing a table of limit-line segments.

LIMF

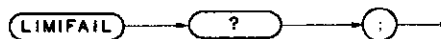
Program Example

```
120 ! This program demonstrates the use of the LIMF command
130 ! to generate a limit line:
150 !
160 OUTPUT 718;"LIMIPURGE;";          ! Purge current limit line.
170 OUTPUT 718;"EDITLIML;";          ! Begin edit of limit line.
180 OUTPUT 718;"LIMIREL OFF;";       ! Make sure it's an absolute limit line.
190 OUTPUT 718;"SADD;";              ! Add segment to the limit line.
200 OUTPUT 718;"SEDI 1;";            ! Edit the first segment.
210 OUTPUT 718;"LIMF 250MHZ;";       ! Set frequency of the segment.
220 OUTPUT 718;"LIMU -35DBM;";       ! Provide the upper amplitude.
230 OUTPUT 718;"LIML -80DBM;";       ! Enter the lower amplitude.
240 OUTPUT 718;"LIMTFL;";           ! Set type of segment to flat.
250 OUTPUT 718;"SDON;";             ! Enter parameters into table.
260 OUTPUT 718;"SADD;SEDI 2;";       ! Add, then edit second segment.
270 OUTPUT 718;"LIMF 290MHZ;";       ! Set frequency of segment 2.
280 OUTPUT 718;"LIMU -35DBM;LIML-80DBM;";
                                     ! Set amplitude data for segment.
290 OUTPUT 718;"LIMTSL;";           ! Set segment type to slope.
300 OUTPUT 718;"SDON;";             ! Enter parameters into table.
310 OUTPUT 718;"SADD;SEDI 3;";
320 OUTPUT 718;"LIMF 296.5MHZ;LIMU -5DBM;LIML-80DBM;LIMTSL;";
330 OUTPUT 718;"SDON;";
340 OUTPUT 718;"SADD;SEDI 4;";
350 OUTPUT 718;"LIMF 300MHZ;LIMU -5DBM;LIML-25DBM;LIMTSL;";
360 OUTPUT 718;"SDON;";
370 OUTPUT 718;"SADD;SEDI 5;";
380 OUTPUT 718;"LIMF 303.5MHZ;LIMU -5DBM;LIML-80DBM;LIMTSL;";
390 OUTPUT 718;"SDON;";
400 OUTPUT 718;"SADD;SEDI 6;";
410 OUTPUT 718;"LIMF 304.5MHZ;LIMU -5DBM;LIML-80DBM;LIMTSL;";
420 OUTPUT 718;"SDON;";
430 OUTPUT 718;"SADD;SEDI 7;";
440 OUTPUT 718;"LIMF 310MHZ;LIMU -35DBM;LIML-80DBM;LIMTFL;";
450 OUTPUT 718;"SDON;";
460 OUTPUT 718;"SADD;SEDI 8;";
470 OUTPUT 718;"LIMF 350MHZ;LIMU -35DBM;LIML-80DBM;LIMTFL;";
480 OUTPUT 718;"SDON;";
490 OUTPUT 718;"EDITDONE;";         ! End of limit-line definition.
500 OUTPUT 718;"CF 300MHZ;SP 50MHZ;RB 2MHZ;VB 30KHZ;TS;";
                                     ! Set up analyzer to display the
505                                     ! limit line.
510 END
```

LIMIFAIL

Limit-Fail Query

Syntax



xf232e

Description This command is a query which returns a zero when a limit-line test passes. If an active trace fails the lower-amplitude parameter, the query response is 1; an upper-amplitude failure query response is 2. If an active trace fails both upper- and lower-limits, the query response is 3.

LIMIPURGE

Purge Limit Line

Syntax



x1233a

Description

Deletes the current limit line, but does not remove any limit-line tables saved in the module RAM. Use the DISPOSE command to remove limit-line tables from module memory.

Program Example

```

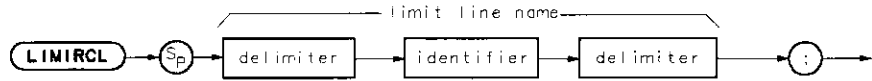
130  ! This program demonstrates the use of the LIMIPURGE command
140  ! to generate a limit line:
150  !
160  OUTPUT 718;"LIMIPURGE;";          ! Purge current limit line.
170  OUTPUT 718;"EDITLIML;";          ! Begin edit of limit line.
180  OUTPUT 718;"LIMIREL OFF;";        ! Make sure it's an absolute limit line.
190  OUTPUT 718;"SADD;";               ! Add segment to the limit line.
200  OUTPUT 718;"SEDI 1;";             ! Edit the first segment.
210  OUTPUT 718;"LIMF 250MHZ;";        ! Set frequency of the segment.
220  OUTPUT 718;"LIMU -35DBM;";        ! Provide the upper amplitude.
230  OUTPUT 718;"LIML -80DBM;";        ! Enter the lower amplitude.
240  OUTPUT 718;"LIMTFL;";             ! Set type of segment to flat.
250  OUTPUT 718;"SDON;";               ! Enter parameters into table.
260  OUTPUT 718;"SADD;SEDI 2;";        ! Add, then edit second segment.
270  OUTPUT 718;"LIMF 290MHZ;";        ! Set frequency of segment 2.
280  OUTPUT 718;"LIMU -35DBM;LIML-80DBM;";
                                         ! Set amplitude data for segment.
290  OUTPUT 718;"LIMTSL;";             ! Set segment type to slope.
300  OUTPUT 718;"SDON;";               ! Enter parameters into table.
310  OUTPUT 718;"SADD;SEDI 3;";
320  OUTPUT 718;"LIMF 296.5MHZ;LIMU -5DBM;LIML-80DBM;LIMTSL;";
330  OUTPUT 718;"SDON;";
340  OUTPUT 718;"LIMTFL;";
340  OUTPUT 718;"SADD;SEDI 4;"; 1
350  OUTPUT 718;"LIMF 300MHZ;LIMU -5DBM;LIML-25DBM;LIMTSL;";
360  OUTPUT 718;"SDON;";
370  OUTPUT 718;"SADD;SEDI 5;";
380  OUTPUT 718;"LIMF 303.5MHZ;LIMU -5DBM;LIML-80DBM;LIMTSL;";
390  OUTPUT 718;"SDON;";
400  OUTPUT 718;"SADD;SEDI 6;";
410  OUTPUT 718;"LIMF 304.5MHZ;LIMU -5DBM;LIML-80DBM;LIMTSL;";
420  OUTPUT 718;"SDON;";
430  OUTPUT 718;"SADD;SEDI 7;";
440  OUTPUT 718;"LIMF 310MHZ;LIMU -35DBM;LIML-80DBM;LIMTFL;";
450  OUTPUT 718;"SDON;";
460  OUTPUT 718;"SADD;SEDI 8;";
470  OUTPUT 718;"LIMF 350MHZ;LIMU -35DBM;LIML-80DBM;LIMTFL;";
480  OUTPUT 718;"SDON;";
490  OUTPUT 718;"EDITDONE;";          ! End of limit-line definition.
500  OUTPUT 718;"CF 300MHZ;SP 50MHZ;RB 2MHZ;VB 30KHZ;TS;";
                                         ! Set up analyzer to display the
505                                         ! limit line.
510  END

```

LIMIRCL

Recall Limit Line

Syntax



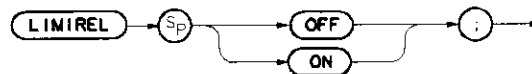
xf234b

Description

Note

Limit lines stored on memory cards using a mass-memory module with firmware datecode 910116 or later cannot be read into a module with an earlier firmware datecode (for example, 890524). Limit lines stored on cards using modules with firmware datecode 890524 or earlier can be read into a module with firmware datecode 910116 or later.

Recalls a limit-line set from the limit-line table in the module user memory. The table is stored in user memory with the LIMISAVE command. The command displays a limit line which is recalled by the name assigned to it. A limit line may be saved and given a name using the remote command LIMISAV, or entered from the front panel with the screen-title function. To recall a limit line from a memory card, use the command CARDLOAD first to copy the limit line to the module memory. Use the LIMITEST ON command to display the line.

LIMIREL**Relative Limit line****Syntax**

x f 235a

Description When set to ON, this command used with the EDITLIML command creates a relative limit line. The default setting is OFF, which makes the limit line amplitude and frequency parameters absolute.

LIMIREL

Program Example

```
130 ! This program demonstrates the use of the LIMIREL command
140 ! to generate a limit line:
150 !
160 OUTPUT 718;"LIMIPURGE;"; ! Purge current limit line.
170 OUTPUT 718;"EDITLIML;"; ! Begin edit of limit line.
180 OUTPUT 718;"LIMIREL ON;"; ! Make sure it's a relative limit line.
190 OUTPUT 718;"SADD;"; ! Add segment to the limit line.
200 OUTPUT 718;"SEDI 1;"; ! Edit the first segment.
210 OUTPUT 718;"LIMF 250MHZ;"; ! Set frequency of the segment.
220 OUTPUT 718;"LIMU -35DBM;"; ! Provide the upper amplitude.
230 OUTPUT 718;"LIML -80DBM;"; ! Enter the lower amplitude.
240 OUTPUT 718;"LIMTFL;"; ! Set type of segment to flat.
250 OUTPUT 718;"SDON;"; ! Enter parameters into table.
260 OUTPUT 718;"SADD;SEDI 2;"; ! Add, then edit second segment.
270 OUTPUT 718;"LIMF 290MHZ;"; ! Set frequency of segment 2.
280 OUTPUT 718;"LIMU -35DBM;LIML-80DBM;";
! Set amplitude data for segment.
290 OUTPUT 718;"LIMTSL;"; ! Set segment type to slope.
300 OUTPUT 718;"SDON;"; ! Enter parameters into table.
310 OUTPUT 718;"SADD;SEDI 3;";
320 OUTPUT 718;"LIMF 296.5MHZ;LIMU -5DBM;LIML-80DBM;LIMTSL;";
330 OUTPUT 718;"SDON;";
340 OUTPUT 718;"SADD;SEDI 4;";
350 OUTPUT 718;"LIMF 300MHZ;LIMU -5DBM;LIML-25DBM;LIMTSL;";
360 OUTPUT 718;"SDON;";
370 OUTPUT 718;"SADD;SEDI 5;";
380 OUTPUT 718;"LIMF 303.5MHZ;LIMU -5DBM;LIML-80DBM;LIMTSL;";
390 OUTPUT 718;"SDON;";
400 OUTPUT 718;"SADD;SEDI 6;";
410 OUTPUT 718;"LIMF 304.5MHZ;LIMU -5DBM;LIML-80DBM;LIMTSL;";
420 OUTPUT 718;"SDON;";
430 OUTPUT 718;"SADD;SEDI 7;";
440 OUTPUT 718;"LIMF 310MHZ;LIMU -35DBM;LIML-80DBM;LIMTFL;";
450 OUTPUT 718;"SDON;";
460 OUTPUT 718;"SADD;SEDI 8;";
470 OUTPUT 718;"LIMF 350MHZ;LIMU -35DBM;LIML-80DBM;LIMTFL;";
480 OUTPUT 718;"SDON;";
490 OUTPUT 718;"EDITDONE;"; ! End of limit-line definition.
500 OUTPUT 718;"CF 300MHZ;SP 50MHZ;RB 2MHZ;VB 30KHZ;TS;";
! Set up analyzer to display the
505 ! limit line.
510 END
```

LIMISAV

Limit-Line Save

Syntax



xf236c

Description

Note

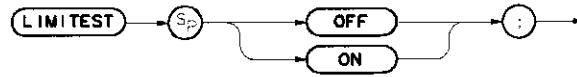
Limit lines stored on memory cards using a mass-memory module with firmware datecode 910116 or later cannot be read into a module with an earlier firmware datecode (for example, 890524). Limit lines stored on cards using modules with firmware datecode 890524 or earlier can be read into a module with firmware datecode 910116 or later.

This command saves the active limit line to module memory under the name assigned to it. Any previously existing limit line having the same name is overwritten with the new limit-line table data. Refer also to the LIMIRCL command. To save a limit line to the memory card, execute CARDSTORE after the LIMISAV command.

LIMITEST

Limit-Line Test

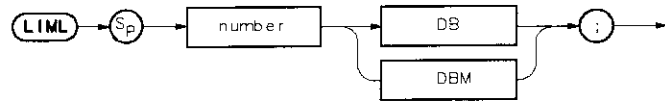
Syntax



x1237a

Description

This command activates the limit-line test function, which compares the trace data in the current sweep with the limits set up in the limit table of the active limit line. The results of the current active trace compared with the active limit line can be read using the LIMIFAIL command. When this command is set to ON, the active limit-line test limits are displayed on-screen along with a LIMIT FAILED message if the trace data fails.

LIML**Lower Limit-Line Value****Syntax**

x1236a

Description This command is used within the SEDI command to assign the lower-limit amplitude value to a limit-line segment. This command used with the LIMU command creates upper and lower limit-line amplitude parameters.

Parameter Number range: -175 dB to 50 dB

LIML

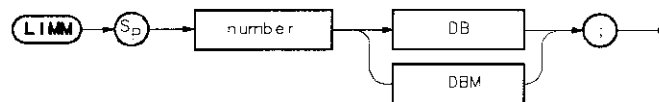
Program Example

```
130 ! This program demonstrates the use of the LIML command
140 ! to generate a limit line:
150 !
160 OUTPUT 718;"LIMIPURGE;"; ! Purge current limit line.
170 OUTPUT 718;"EDITLIML;"; ! Begin edit of limit line.
180 OUTPUT 718;"LIMIREL OFF;"; ! Make sure it's an absolute limit line.
190 OUTPUT 718;"SADD;"; ! Add segment to the limit line.
200 OUTPUT 718;"SEDI 1;"; ! Edit the first segment.
210 OUTPUT 718;"LIMF 250MHZ;"; ! Set frequency of the segment.
220 OUTPUT 718;"LIMU -35DBM;"; ! Provide the upper amplitude.
230 OUTPUT 718;"LIML -80DBM;"; ! Enter the lower amplitude.
240 OUTPUT 718;"LIMTFL;"; ! Set type of segment to flat.
250 OUTPUT 718;"SDON;"; ! Enter parameters into table.
260 OUTPUT 718;"SADD;SEDI 2;"; ! Add, then edit second segment.
270 OUTPUT 718;"LIMF 290MHZ;"; ! Set frequency of segment 2.
280 OUTPUT 718;"LIMU -35DBM;LIML -80DBM;";
! Set amplitude data for segment.
290 OUTPUT 718;"LIMTSL;"; ! Set segment type to slope.
300 OUTPUT 718;"SDON;"; ! Enter parameters into table.
310 OUTPUT 718;"SADD;SEDI 3;";
320 OUTPUT 718;"LIMF 296.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;";
330 OUTPUT 718;"SDON;";
340 OUTPUT 718;"SADD;SEDI 4;";
350 OUTPUT 718;"LIMF 300MHZ;LIMU -5DBM;LIML -25DBM;LIMTSL;";
360 OUTPUT 718;"SDON;";
370 OUTPUT 718;"SADD;SEDI 5;";
380 OUTPUT 718;"LIMF 303.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;";
390 OUTPUT 718;"SDON;";
400 OUTPUT 718;"SADD;SEDI 6;";
410 OUTPUT 718;"LIMF 304.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;";
420 OUTPUT 718;"SDON;";
430 OUTPUT 718;"SADD;SEDI 7;";
440 OUTPUT 718;"LIMF 310MHZ;LIMU -35DBM;LIML -80DBM;LIMTFL;";
450 OUTPUT 718;"SDON;";
460 OUTPUT 718;"SADD;SEDI 8;";
470 OUTPUT 718;"LIMF 350MHZ;LIMU -35DBM;LIML -80DBM;LIMTFL;";
480 OUTPUT 718;"SDON;";
490 OUTPUT 718;"EDITDONE;"; ! End of limit-line definition.
500 OUTPUT 718;"CF 300MHZ;SP 50MHZ;RB 2MHZ;VB 30KHZ;TS;";
! Set up analyzer to display the
! limit line.
505
510 END
```

LIMM

Middle Limit-Line Value

Syntax



xf239a

Description This command is used within the SEDI command to assign a middle amplitude value to a limit line. This command used with the LIMD command create middle and delta limit-line amplitude parameters.

Parameter Number range: -175 dB to 50 dB

LIMM

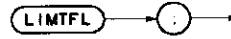
Program Example

```
120 ! This program demonstrates the use of the LIMM command
130 ! to generate a limit line:
150 !
160 OUTPUT 718;"LIMIPURGE;";      ! Purge current limit line.
170 OUTPUT 718;"EDITLIML;";      ! Begin edit of limit line.
180 OUTPUT 718;"LIMIREL OFF;";    ! Make sure it's an absolute limit line.
190 OUTPUT 718;"SADD;";          ! Add segment to the limit line.
200 OUTPUT 718;"SEDI 1;";        ! Edit the first segment.
210 OUTPUT 718;"LIMF 250MHZ;";    ! Set frequency of the segment.
220 OUTPUT 718;"LIMU -35DBM;";    ! Provide the upper amplitude.
230 OUTPUT 718;"LIML -80DBM;";    ! Enter the lower amplitude.
240 OUTPUT 718;"LIMTFL;";        ! Set type of segment to flat.
250 OUTPUT 718;"SDON;";          ! Enter parameters into table.
260 OUTPUT 718;"SADD;SEDI 2;";    ! Add, then edit second segment.
270 OUTPUT 718;"LIMF 290MHZ;";    ! Set frequency of segment 2.
280 OUTPUT 718;"LIMU -35DBM;LIML-80DBM;";
                                   ! Set amplitude data for segment.
290 OUTPUT 718;"LIMTSL;";        ! Set segment type to slope.
300 OUTPUT 718;"SDON;";          ! Enter parameters into table.
310 OUTPUT 718;"SADD;SEDI 3;";
320 OUTPUT 718;"LIMF 296.5MHZ;LIMU -5DBM;LIML-80DBM;LIMTSL;";
330 OUTPUT 718;"SDON;";
340 OUTPUT 718;"SADD;SEDI 4;";
350 OUTPUT 718;"LIMF 300MHZ;LIMM -15DBM;LIMD 10DBM;LIMTSL;";
360 OUTPUT 718;"SDON;";
370 OUTPUT 718;"SADD;SEDI 5;";
380 OUTPUT 718;"LIMF 303.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;";
390 OUTPUT 718;"SDON;";
400 OUTPUT 718;"SADD;SEDI 6;";
410 OUTPUT 718;"LIMF 304.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;";
420 OUTPUT 718;"SDON;";
430 OUTPUT 718;"SADD;SEDI 7;";
440 OUTPUT 718;"LIMF 310MHZ;LIMU -35DBM;LIML -80DBM;LIMTFL;";
450 OUTPUT 718;"SDON;";
460 OUTPUT 718;"SADD;SEDI 8;";
470 OUTPUT 718;"LIMF 350MHZ;LIMU -35DBM;LIML -80DBM;LIMTFL;";
480 OUTPUT 718;"SDON;";
490 OUTPUT 718;"EDITDONE;";      ! End of limit-line definition.
500 OUTPUT 718;"CF 300MHZ;SP 50MHZ;RB 2MHZ;VB 30KHZ;TS;";
                                   ! Set up analyzer to display the
505                                   ! limit line.
510 END
```

LIMTFL

Flat Limit-Line Segment

Syntax



x 1240a

Description This command is used within the SEDI command to make the selected limit-line segment flat.

LIMTFL

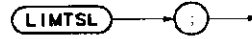
Program Example

```
130 ! This program demonstrates the use of the LIMTFL command
140 ! to generate a limit line:
150 !
160 OUTPUT 718;"LIMIPURGE;";          ! Purge current limit line.
170 OUTPUT 718;"EDITLIML;";          ! Begin edit of limit line.
180 OUTPUT 718;"LIMIREL OFF;";       ! Make sure it's an absolute limit line.
190 OUTPUT 718;"SADD;";              ! Add segment to the limit line.
200 OUTPUT 718;"SEDI 1;";            ! Edit the first segment.
210 OUTPUT 718;"LIMF 250MHZ;";       ! Set frequency of the segment.
220 OUTPUT 718;"LIMU -35DBM;";       ! Provide the upper amplitude.
230 OUTPUT 718;"LIML -80DBM;";       ! Enter the lower amplitude.
240 OUTPUT 718;"LIMTFL;";            ! Set type of segment to flat.
250 OUTPUT 718;"SDON;";              ! Enter parameters into table.
260 OUTPUT 718;"SADD;SEDI 2;";       ! Add, then edit second segment.
270 OUTPUT 718;"LIMF 290MHZ;";       ! Set frequency of segment 2.
280 OUTPUT 718;"LIMU-35DBM;LIML-80DBM;";
                                     ! Set amplitude data for segment.
290 OUTPUT 718;"LIMTSL;";            ! Set segment type to slope.
300 OUTPUT 718;"SDON;";              ! Enter parameters into table.
310 OUTPUT 718;"SADD;SEDI 3;";
320 OUTPUT 718;"LIMF 296.5MHZ;LIMU-5DBM;LIML-80DBM;LIMTSL;";
330 OUTPUT 718;"SDON;";
340 OUTPUT 718;"SADD;SEDI 4;";
350 OUTPUT 718;"LIMF 300MHZ;LIMU-5DBM;LIML-25DBM;LIMTSL;";
360 OUTPUT 718;"SDON;";
370 OUTPUT 718;"SADD;SEDI 5;";
380 OUTPUT 718;"LIMF 303.5MHZ;LIMU-5DBM;LIML-80DBM;LIMTSL;";
390 OUTPUT 718;"SDON;";
400 OUTPUT 718;"SADD;SEDI 6;";
410 OUTPUT 718;"LIMF 304.5MHZ;LIMU-5DBM;LIML-80DBM;LIMTSL;";
420 OUTPUT 718;"SDON;";
430 OUTPUT 718;"SADD;SEDI 7;";
440 OUTPUT 718;"LIMF 310MHZ;LIMU-35DBM;LIML-80DBM;LIMTFL;";
450 OUTPUT 718;"SDON;";
460 OUTPUT 718;"SADD;SEDI 8;";
470 OUTPUT 718;"LIMF 350MHZ;LIMU-35DBM;LIML-80DBM;LIMTFL;";
480 OUTPUT 718;"SDON;";
490 OUTPUT 718;"EDITDONE;";          ! End of limit-line definition.
491 !
492                                     ! Set up analyzer to display the
493                                     ! limit line.
494 OUTPUT 718;"CF 300MHZ;SP 50MHZ;RB 2MHZ;VB 30KHZ;TS;";
500 END
```

LIMTSL

Sloped Limit-Line Segment

Syntax



x12410

Description This command is used within the SEDI command to create a limit line with a sloped line segment.

LIMITSL

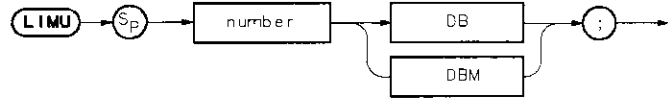
Program Example

```
130 ! This program demonstrates the use of the LIMITSL command
140 ! to generate a limit line:
150 !
160 OUTPUT 718;"LIMIPURGE;"; ! Purge current limit line.
170 OUTPUT 718;"EDITLIML;"; ! Begin edit of limit line.
180 OUTPUT 718;"LIMIREL OFF;"; ! Make sure it's an absolute limit line.
190 OUTPUT 718;"SADD;"; ! Add segment to the limit line.
200 OUTPUT 718;"SEDI 1;"; ! Edit the first segment.
210 OUTPUT 718;"LIMF 250MHZ;"; ! Set frequency of the segment.
220 OUTPUT 718;"LIMU -35DBM;"; ! Provide the upper amplitude.
230 OUTPUT 718;"LIML -80DBM;"; ! Enter the lower amplitude.
240 OUTPUT 718;"LIMTFL;"; ! Set type of segment to flat.
250 OUTPUT 718;"SDON;"; ! Enter parameters into table.
260 OUTPUT 718;"SADD;SEDI 2;"; ! Add, then edit second segment.
270 OUTPUT 718;"LIMF 290MHZ;"; ! Set frequency of segment 2.
280 OUTPUT 718;"LIMU-35DBM;LIML-80DBM;";
! Set amplitude data for segment.
290 OUTPUT 718;"LIMITSL;"; ! Set segment type to slope.
300 OUTPUT 718;"SDON;"; ! Enter parameters into table.
310 OUTPUT 718;"SADD;SEDI 3;";
320 OUTPUT 718;"LIMF 296.5MHZ;LIMU-5DBM;LIML-80DBM;LIMITSL;";
330 OUTPUT 718;"SDON;";
340 OUTPUT 718;"SADD;SEDI 4;";
350 OUTPUT 718;"LIMF 300MHZ;LIMU-5DBM;LIML-25DBM;LIMITSL;";
360 OUTPUT 718;"SDON;";
370 OUTPUT 718;"SADD;SEDI 5;";
380 OUTPUT 718;"LIMF 303.5MHZ;LIMU-5DBM;LIML-80DBM;LIMITSL;";
390 OUTPUT 718;"SDON;";
400 OUTPUT 718;"SADD;SEDI 6;";
410 OUTPUT 718;"LIMF 304.5MHZ;LIMU-5DBM;LIML-80DBM;LIMITSL;";
420 OUTPUT 718;"SDON;";
430 OUTPUT 718;"SADD;SEDI 7;";
440 OUTPUT 718;"LIMF 310MHZ;LIMU-35DBM;LIML-80DBM;LIMTFL;";
450 OUTPUT 718;"SDON;";
460 OUTPUT 718;"SADD;SEDI 8;";
470 OUTPUT 718;"LIMF 350MHZ;LIMU-35DBM;LIML-80DBM;LIMTFL;";
480 OUTPUT 718;"SDON;";
490 OUTPUT 718;"EDITDONE;"; ! End of limit-line definition.
491 !
492 ! Set up analyzer to display the
493 ! limit line.
494 OUTPUT 718;"CF 300MHZ;SP 50MHZ;RB 2MHZ;VB 30KHZ;TS;";
500 END
```

LIMU

Upper Limit-Line Value

Syntax



x1242a

Description This command is used within the SEDI command to assign the upper-amplitude value to a limit-line segment. This command combined with the LIML command may be used to create upper and lower limit-line amplitude parameters.

Parameters Number Range: -175 dB to 50 dB

LIMU

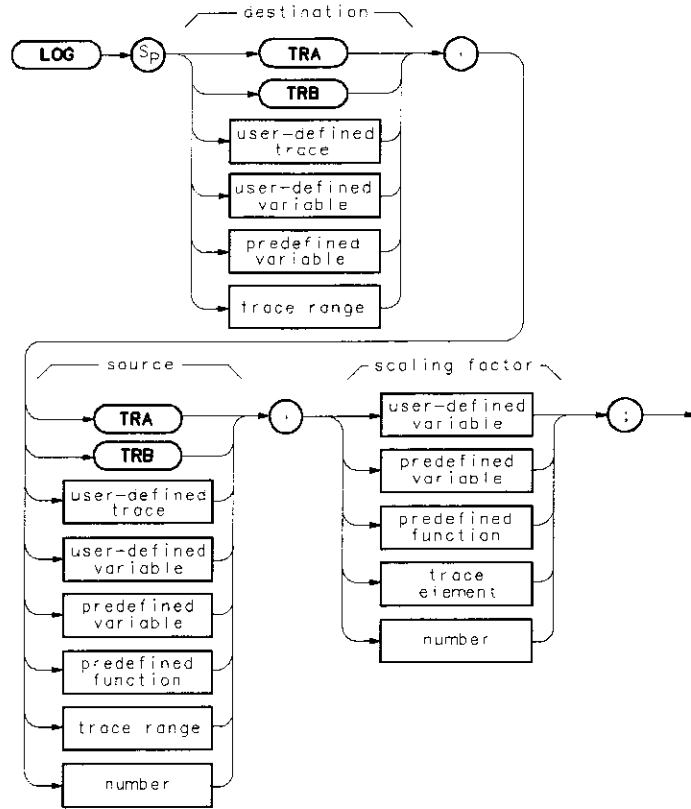
Program Example

```
130 ! This program demonstrates the use of the LIMU command
140 ! to generate a limit line:
150 !
160 OUTPUT 718;"LIMIPURGE;"; ! Purge current limit line.
170 OUTPUT 718;"EDITLIML;"; ! Begin edit of limit line.
180 OUTPUT 718;"LIMIREL OFF;"; ! Make sure it's an absolute limit line.
190 OUTPUT 718;"SADD;"; ! Add segment to the limit line.
200 OUTPUT 718;"SEDI 1;"; ! Edit the first segment.
210 OUTPUT 718;"LIMF 250MHZ;"; ! Set frequency of the segment.
220 OUTPUT 718;"LIMU -35DBM;"; ! Provide the upper amplitude.
230 OUTPUT 718;"LIML -80DBM;"; ! Enter the lower amplitude.
240 OUTPUT 718;"LIMTFL;"; ! Set type of segment to flat.
250 OUTPUT 718;"SDON;"; ! Enter parameters into table.
260 OUTPUT 718;"SADD;SEDI 2;"; ! Add, then edit second segment.
270 OUTPUT 718;"LIMF 290MHZ;"; ! Set frequency of segment 2.
280 OUTPUT 718;"LIMU-35DBM;LIML-80DBM;";
! Set amplitude data for segment.
290 OUTPUT 718;"LIMTSL;"; ! Set segment type to slope.
300 OUTPUT 718;"SDON;"; ! Enter parameters into table.
310 OUTPUT 718;"SADD;SEDI 3;";
320 OUTPUT 718;"LIMF 296.5MHZ;LIMU-5DBM;LIML-80DBM;LIMTSL;";
330 OUTPUT 718;"SDON;";
340 OUTPUT 718;"SADD;SEDI 4;";
350 OUTPUT 718;"LIMF 300MHZ;LIMU-5DBM;LIML-25DBM;LIMTSL;";
360 OUTPUT 718;"SDON;";
370 OUTPUT 718;"SADD;SEDI 5;";
380 OUTPUT 718;"LIMF 303.5MHZ;LIMU-5DBM;LIML-80DBM;LIMTSL;";
390 OUTPUT 718;"SDON;";
400 OUTPUT 718;"SADD;SEDI 6;";
410 OUTPUT 718;"LIMF 304.5MHZ;LIMU-5DBM;LIML-80DBM;LIMTSL;";
420 OUTPUT 718;"SDON;";
430 OUTPUT 718;"SADD;SEDI 7;";
440 OUTPUT 718;"LIMF 310MHZ;LIMU-35DBM;LIML-80DBM;LIMTFL;";
450 OUTPUT 718;"SDON;";
460 OUTPUT 718;"SADD;SEDI 8;";
470 OUTPUT 718;"LIMF 350MHZ;LIMU-35DBM;LIML-80DBM;LIMTFL;";
480 OUTPUT 718;"SDON;";
490 OUTPUT 718;"EDITDONE;"; ! End of limit-line definition.
491 !
! Set up analyzer to display the
492 ! limit line.
493
494 OUTPUT 718;"CF 300MHZ;SP 50MHZ;RB 2MHZ;VB 30KHZ;TS;";
500 END
```

LOG

Logarithm

Syntax



x1243a

Description The logarithm (base 10) of the source is taken, the result is multiplied by a specified scaling factor, then sent to the destination.

LOG

Program Examples

Example #1

```
10  ! The following example shows the use of the LOG command.  The example
20  ! uses values from HP BASIC and concatenates them into the statements as
30  ! they are sent to the spectrum analyzer.
40  !
50  ASSIGN @Sa TO 718          ! Assign I/O path to address 718.
60  !
70  OUTPUT @Sa;"CLRDSP;";      ! Clears the analyzer display.
80  !
90  OUTPUT @Sa;"VARDEF LOG_TEN,0;"; ! \ Initialize variables.
100 OUTPUT @Sa;"VARDEF YVAL,700;"; ! /
110 !
120 FOR I=100 TO 1000 STEP 100 ! Begin loop.
130     OUTPUT @Sa;"LOG LOG_TEN,"&VAL$(I)&",1;";
                                     ! Take the LOG and multiply
                                     ! by scaling factor of 1.
140                                     !
150     OUTPUT @Sa;"PU;PA 20,YVAL;"; ! Position "pen" for text.
160     OUTPUT @Sa;"TEXT %The LOG of %;";
                                     ! Write text.
170     OUTPUT @Sa;"DSPLY "&VAL$(I)&",6,3;";
                                     ! Write a value.
180     OUTPUT @Sa;"TEXT % = %;";      ! Write text.
190     OUTPUT @Sa;"DSPLY LOG_TEN,5,3;";
                                     ! Write a value.
200     OUTPUT @Sa;"SUB YVAL,YVAL,50;"; ! Calculate new "pen" position.
210 NEXT I                          ! End loop.
220 !
230 ASSIGN @Sa TO *                  ! Close I/O path.
240 END
```


Example #2

```

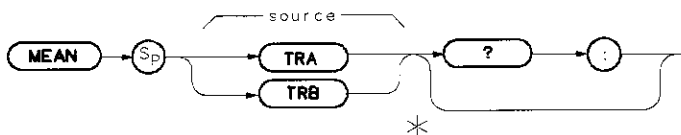
10  !The following example shows another use of the LOG command. This example
20  !performs the same function as the previous example, but is fully
30  !downloadable. Note that only minor changes need to be made to convert
40  !a program from computer-dependent to computer-independent! This method
50  !is also visibly much faster for the spectrum analyzer to execute.
60  !
70  ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
80  !
90  OUTPUT @Sa;"FUNCDEF LOG_EX,@;";
                                     ! Logical start of DLP.
100 OUTPUT @Sa;"CLRDISP;";           ! Clears the analyzer display.
110 !
120 OUTPUT @Sa;"VARDEF LOG_TEN,0;";   ! \ Initialize variables.
130 OUTPUT @Sa;"VARDEF YVAL,700;";   ! /
140 OUTPUT @Sa;"VARDEF LP_CTR,100;"; ! Initialize loop counter.
150 !
160 OUTPUT @Sa;"REPEAT;";            ! Begin loop.
170   OUTPUT @Sa;"LOG LOG_TEN,LP_CTR,1;";
                                     ! Take the LOG and multiply
175                                     ! by scaling factor of 1.
190   OUTPUT @Sa;"PU;PA 20,YVAL;";   ! Position "pen" for text.
200   OUTPUT @Sa;"TEXT %The LOG of %;"; ! Write text.
210   OUTPUT @Sa;"DSPLY LP_CTR,6,3;"; ! Write a value.
220   OUTPUT @Sa;"TEXT % = %;";      ! Write text.
230   OUTPUT @Sa;"DSPLY LOG_TEN,5,3;"; ! Write a value.
240   OUTPUT @Sa;"SUB YVAL,YVAL,50;"; ! Calculate new "pen" position.
250   OUTPUT @Sa;"ADD LP_CTR,LP_CTR,100;";
                                     ! Increment loop counter.
260 OUTPUT @Sa;"UNTIL LP_CTR,GT,1000;"; ! End loop.
270 OUTPUT @Sa;"@";
280 !
290 OUTPUT @Sa;"LOG_EX;";            ! Execute the DLP.
300 ASSIGN @Sa TO *                  ! Close I/O path.
310 END

```

MEAN

Trace Mean

Syntax



xf244a

Note

This bypass command path is only legal if you use MEAN as a predefined function. It must reside within a compatible-function operation.

Description

Returns the mean value of a trace in measurement units. This single value must be used as the <source> of another function. In the first example below, MEAN TRA is the source for the MOV command.

Program Examples

Example #1

```
10 ! The following example shows how to use the trace MEAN function
20 ! to return the mean value to the controller using an internal variable.
30 !
40 INTEGER Mean_value ! Define an INTEGER variable.
50 ASSIGN @Sa TO 718 ! Assign I/O path to address 718.
60 OUTPUT @Sa;"VARDEF M_VALUE,0;"; ! Define an analyzer variable.
70 OUTPUT @Sa;"MOV M_VALUE,MEAN TRA;"; ! Determine the mean of trace A.
80 OUTPUT @Sa;"M_VALUE?;"; ! Return contents of variable.
90 ENTER @Sa;Mean_value ! Read value from analyzer.
100 PRINT "MEAN of trace A in measurement units = ";Mean_value
110 ASSIGN @Sa TO * ! Close I/O path.
120 END
```

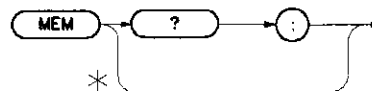
Example #2

```
10 ! The following example shows how to use the trace MEAN function to return
20 ! the mean value of the specified trace to the controller.
30 !
40 INTEGER Mean_value ! Define an INTEGER variable.
50 ASSIGN @Sa TO 718 ! Assign I/O path to address 718.
60 OUTPUT @Sa;"MEAN TRA,?;"; ! Determine the mean of trace A.
70 ENTER @Sa;Mean_value ! Read value from analyzer.
80 PRINT "MEAN of trace A in measurement units = ";Mean_value
90 ASSIGN @Sa TO * ! Close I/O path.
100 !
110 END
```

MEM

Memory Available

Syntax



x f245a

Note

This bypass command path is only legal if you use MEM as a predefined function. It must reside within a compatible-function operation.

Description

Use this command to query the amount of unused user memory. This command cannot be used within a DLP (or a FUNCDEF). The quantity is dependent on the following conditions:

- number and length of DLPs stored
- number of traces stored
- number of limit lines stored
- number and length of variables stored

The following commands use available memory to store data:

- LIMISAV (save a limit line)
- ARRAYDEF (define an array)
- TRDEF (define a trace)
- VARDEF (define a variable)
- FUNCDEF (define a function)
- ONEOS (on-end-of-sweep)
- REPEAT/UNTIL (looping construct sent as remote HP-IB command)
- IF/THEN/ELSE/ENDIF (conditional construct sent as remote HP-IB command)
- MODSAVT (save trace data in module memory)

Related Commands: DISPOSE, FUNCDEF, VARDEF, LCLVAR, ARRAYDEF, TRDEF, REPEAT/UNTIL, IF/THEN, and ONEOS

Program Example

```

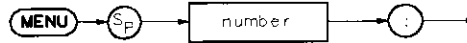
10    ! The following example shows the use of the MEM command.
20    !
30    ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
40    !
50    OUTPUT @Sa;"IP;";                ! Instrument preset.
60    !
70    OUTPUT @Sa;"MEM?;";              ! Query available memory.
80    ENTER @Sa;A_mem                  ! Enter value.
90    PRINT "AVAILABLE ANALYZER MEMORY = ";A_mem;" BYTES"
100   !
110   ASSIGN @Sa TO *                  ! Close I/O path.
120   END

```

MENU

Show the User Key Menu

Syntax



x1246a

Description This command can be used in a DLP only to display the module User Keys menu you specify. Specify zero to display the User Keys menu with all labels removed, 1 to display the first level of user keys (1 through 5), and 2 to display the second level of user keys (6 through 10). This command is used only within a DLP and has no meaning if the spectrum analyzer is in remote mode. It must be used in a user-defined function, or DLP. If used from an external controller, an error results.

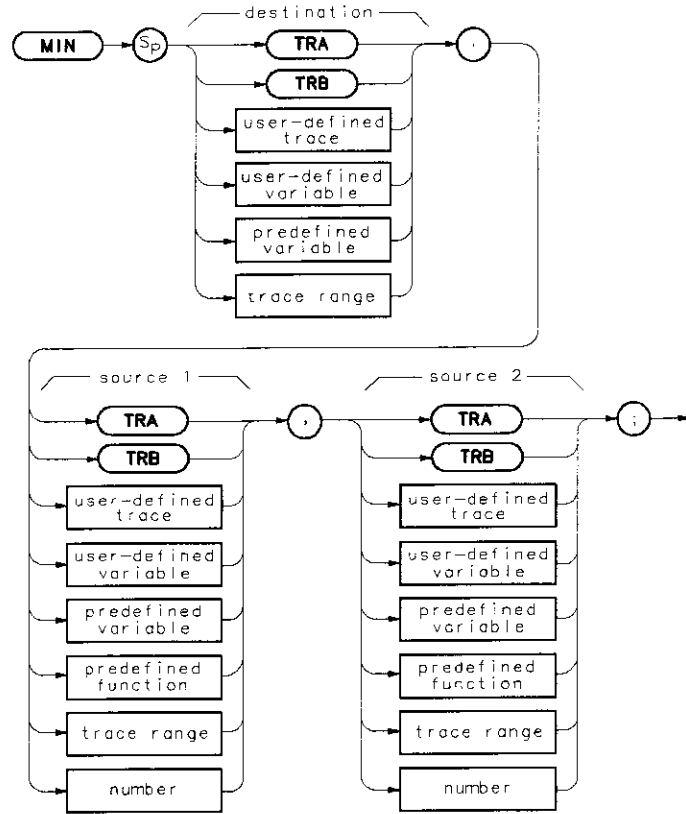
Program Example

```
10 ! The following example shows the use of the MENU command. The
20 ! example is a function that loads two user softkeys. By executing
30 ! this function (from the front panel), softkeys will be displayed
40 ! which can then be used to execute other previously loaded DLPs.
50 !
60 ASSIGN @Sa TO 718 ! Assign I/O path to address 718.
70 !
80 OUTPUT @Sa;"FUNCDEF DISP_KEYS,@;"; ! Logical start of DLP.
90 OUTPUT @Sa;"KEYDEF 1,BOX,%DRAW BOX%;"; ! Create a labeled softkey.
100 OUTPUT @Sa;"KEYDEF 2,DBM_TO_MW,%dBm TO mW%;";
!Create another softkey.
110 OUTPUT @Sa;"MENU 1;"; ! Display softkey menu 1.
120 OUTPUT @Sa;"@"; ! Logical end of DLP.
130 !
140 ASSIGN @Sa TO * ! Close I/O path.
150 END
```

MIN

Minimum

Syntax



x1247a

Description

Compares source 2 with source 1, point by point, and sends the lesser value of each comparison to the destination.

MIN

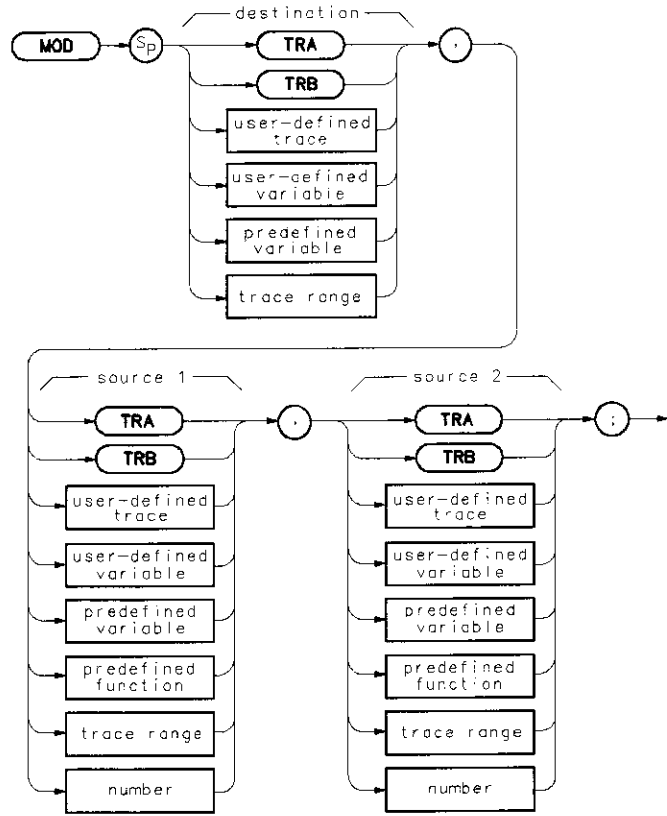
Program Example

```
10 ! The following example shows a use of the MIN command. The example
20 ! requires an input signal of 300 MHz, such as the CAL OUTPUT.
30 !
40 ASSIGN @Sa TO 718 ! Assign I/O path to address 718.
50 !
60 OUTPUT @Sa;"IP;"; ! Instrument preset.
70 OUTPUT @Sa;"CF 300MHZ; SP 200KHZ;";
! Set center frequency and span.
80 OUTPUT @Sa;"CLRW TRA;"; ! Clear-write trace A.
90 OUTPUT @Sa;"CLRW TRB;"; ! Clear-write trace B.
100 OUTPUT @Sa;"TS;"; ! Update the display.
110 OUTPUT @Sa;"VIEW TRA;"; ! View trace A.
120 !
130 FOR Counter=1 TO 10
140     OUTPUT @Sa;"APB;"; ! A + B --> A
150 NEXT Counter
160 OUTPUT @Sa;"AMB ON;"; ! A - B --> A
170 !
180 OUTPUT @Sa;"MIN TRA,TRA,TRB;"; ! Compare trace A and trace B.
190 ! Put the minimum point in trace A.
200 OUTPUT @Sa;"BLANK TRB;"; ! Blank trace B to view the results
210 !
220 ASSIGN @Sa TO * ! Close I/O path.
230 END
```

MOD

Modulo

Syntax



x1248a

Description

Divides source 1 by source 2 and sends the remainder to the destination. If source 2 is zero, an error occurs, and source 1 becomes the result.

MOD

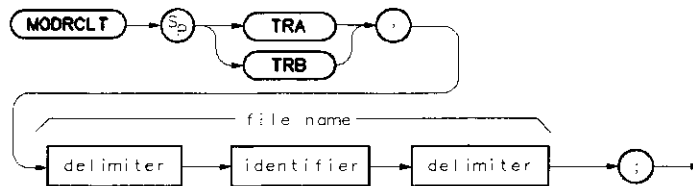
Program Example

```
10      ! The following example shows the use of the MOD command.  The
20      ! instructions within the FUNCDEF '@' delimiters form a structure
30      ! called a DLP (downloadable program).
40      !
50      ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
60      !
70      OUTPUT @Sa;"IP;";                ! Instrument preset.
80      OUTPUT @Sa;"FUNCDEF M_OD_EX,@;"; ! Logical start of the DLP.
90      OUTPUT @Sa;"VARDEF DIVIDEND,10;"; ! Create variable and initialize to 10.
100     OUTPUT @Sa;"VARDEF DIVISOR,6;";  ! Create variable and initialize to 6.
110     OUTPUT @Sa;"VARDEF RESULT,0;";    ! Create variable.
120     !
130     OUTPUT @Sa;"MOD RESULT,DIVIDEND,DIVISOR;";
                                           ! Perform calculation.
140     OUTPUT @Sa;"CLRDSP;";            ! Clear display.
150     OUTPUT @Sa;"PU;PA 10,400;";      ! Move pen to starting position.
160     OUTPUT @Sa;"DSPLY DIVIDEND,4,2;TEXT % MOD %;"; ! \ Display calculation
170     OUTPUT @Sa;"DSPLY DIVISOR,2,1;"; ! and results.
180     OUTPUT @Sa;"TEXT % =%;DSPLY RESULT,2,0;"; ! /
190     OUTPUT @Sa;"@";                  ! Logical end of the DLP.
200     !
210     OUTPUT @Sa;"M_OD_EX;";           ! Have the analyzer execute the
220     ! DLP M_OD_EX.
230     !
240     ASSIGN @Sa TO *                  ! Close I/O path.
250     END
```


MODRCLT

Recall Trace from Module Memory

Syntax



xf249a

Description Recall a specified trace from the source specified by the MSDEV command, to the TRA or TRB of the instrument. This command specifies that the recall occur from a memory module or memory card as specified by the MSDEV command.

Note *For firmware datecode 890524 and earlier:*

If more than one trace is labeled with the same name, MODRCLT cannot distinguish between them.

Note Traces stored on memory cards using a mass-memory module with firmware datecode 910116 or later cannot be read into a module with an earlier firmware datecode (for example, 890524). Traces stored on cards using modules with firmware datecode 890524 or earlier can be read into a module with firmware datecode 910116 or later.

Prerequisite Command: MSDEV

Related Commands: MODSAVET and DISPOSE

Program Example

```

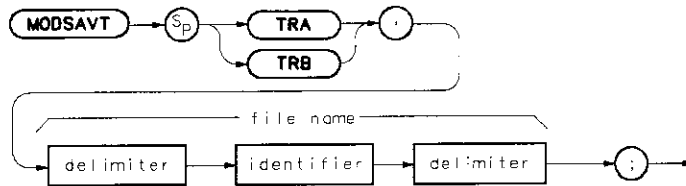
10  ! The following example shows the use of the MODRCLT command. The
20  ! instructions within the FUNCDEF '@' delimiters form a structure
30  ! called a DLP (downloadable program).
40  !
50  ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
60  !
70  OUTPUT @Sa;"FUNCDEF GET_TRC,@;"; ! Logical start of the DLP.
80  OUTPUT @Sa;"IP;SNGLS;TS;VIEW TRA;"; ! Set up the analyzer.
90  OUTPUT @Sa;"MSDEV MEM;";        ! Tell the analyzer where the
100 !                                ! trace is stored.
110 OUTPUT @Sa;"MODRCLT TRA,%CAL_OUT%;"; ! Recall the trace from memory.
120 OUTPUT @Sa;"@";                ! Logical end of the DLP.
130 OUTPUT @Sa;"GET_TRC;";        ! Execute the DLP.
140 !
150 ASSIGN @Sa TO *                ! Close I/O path.
160 END

```

MODSAVT

Save Trace in Module Memory

Syntax



xf250a

Description

This command saves a trace in module memory by its created name or by the default name, *trace*. The spectrum analyzer state and the time and date the trace was saved is also stored. The trace is stored in the destination specified with the MSDEV command.

Note

For firmware datecode 890524 and earlier: It is recommended that unique names be assigned to each trace, since the MODRCLT command does not distinguish traces by the time and date stamp. In addition, when traces are saved on the card the title is modified. If no name is given, the title becomes TR plus the last five digits of the time and date stamp. If a name is given, the title becomes the first four characters of the name plus the last five digits of the time and date stamp.

Note

Traces stored on memory cards using a mass-memory module with firmware datecode 910116 or later cannot be read into a module with an earlier firmware datecode (for example, 890524). Traces stored on cards using modules with firmware datecode 890524 or earlier can be read into a module with firmware datecode 910116 or later.

Restrictions: Subject to available memory. The following commands consume memory: ONEOS, ARRAYDEF, TRDEF, KEYDEF, LCLVAR, VARDEF REPEAT/UNTIL, and IF/THEN.

Prerequisite Command: MSDEV

Related Commands: MODRCLT and DISPOSE

Program Example

```
10  ! The following example shows the use of the MODSAVT command. The
20  ! instructions within the FUNCDEF '@' delimiters form a structure
30  ! called a DLP (downloadable program).
40  !
50  ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
60  !
70  OUTPUT @Sa;"FUNCDEF SAVE_TRC,@;";    ! Logical start of the DLP.
80  OUTPUT @Sa;"IP;CF 300MHZ;SP 1MHZ;TS;"; ! Set up the analyzer.
```

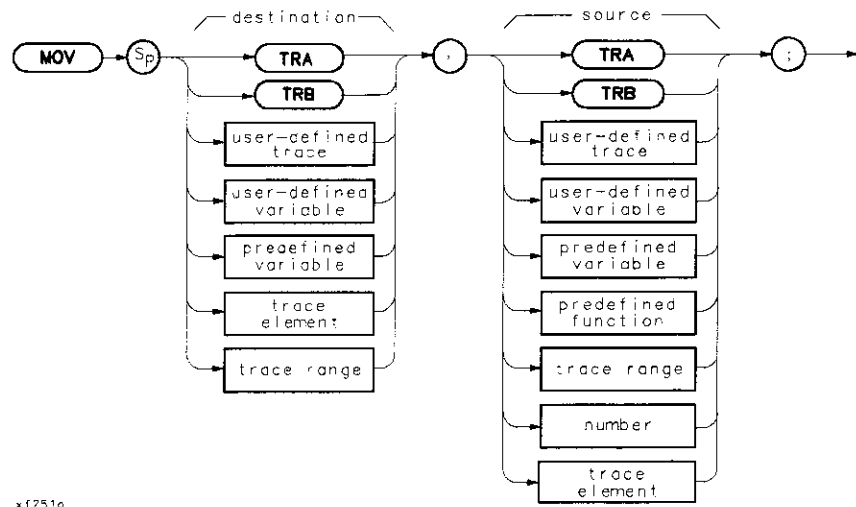
MODSAVT

```
90  OUTPUT @Sa;"MSDEV MEM;";           ! Tell the analyzer where the
100                                     ! trace is to be stored.
110  OUTPUT @Sa;"MODSAVT TRA,%CAL_OUT%"; ! Store the trace.
120  OUTPUT @Sa;"@";                   ! Logical end of the DLP.
130  OUTPUT @Sa;"SAVE_TRC;";           ! Execute the DLP.
140  !
150  ASSIGN @Sa TO *                   ! Close I/O path.
160  END
```

MOV

Move

Syntax



Description Store the source contents in the destination.

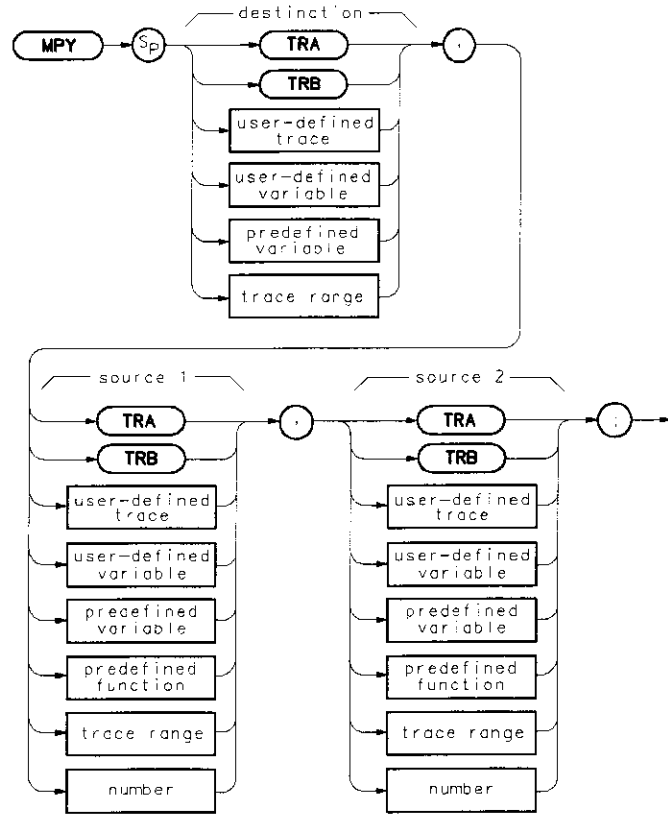
Program Example

```
10 ! The following example shows the use of the MOV command. The example
20 ! uses an input signal of 300 MHz, such as the CAL OUTPUT, and looks at
30 ! its harmonics. An input signal is not necessary to the function of
40 ! this example, but acts as a visual aid.
50 !
60 ASSIGN @Sa TO 718 ! Assign I/O path to address 718.
70 !
80 OUTPUT @Sa;"FUNCDEF INCRFREQ,@;" ! Logical start of the DLP function.
90 OUTPUT @Sa;"VARDEF FREQ,300E6;" ! Create variable and initialize to
300 MHz.
100 OUTPUT @Sa;"IP;SP 1MHZ;" ! Instrument preset; set span;
110 !
120 OUTPUT @Sa;"REPEAT;" ! Begin loop.
130 OUTPUT @Sa;" MOV CF,FREQ;" ! Set center frequency.
140 OUTPUT @Sa;" TS;" ! Take a sweep to update display.
150 OUTPUT @Sa;" ADD FREQ,FREQ,300E6;" ! Increase FREQ by 300 MHz.
160 OUTPUT @Sa;"UNTIL FREQ,GT,3E9;" ! End of loop.
170 OUTPUT @Sa;"@"; ! Logical end of DLP function.
180 !
190 OUTPUT @Sa;" INCRFREQ;" ! Execute function.
200 ! 210 ASSIGN @Sa TO * ! Close I/O path.
220 END
```

MPY

Multiply

Syntax



x1252a

Description Multiplies source 1 by source 2, point by point, and sends the result to the destination. In case of destination overflow, an error is reported and the result is limited to its maximum legal value with a proper sign.

MPY

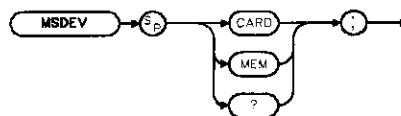
Program Example

```
10  ! The following example shows the use of the MPY command.  The example
20  ! uses an input signal of 300MHz, such as the CAL OUTPUT) and looks at
30  ! its harmonics.  An input signal is not necessary to the function of
40  ! this example, but acts as a visual aid.
50  !
60  ASSIGN @Sa TO 718          ! Assign I/O path to address 718.
70  !
80  OUTPUT @Sa;"FUNCDEF MPYEX,@";      ! Logical start of the DLP function.
90  OUTPUT @Sa;"VARDEF FUNDMNTL,300E6";
                                     ! Create variable and initialize to
                                     ! 300 MHz.
100 OUTPUT @Sa;"VARDEF HARMNC,0";      ! Create variable and initialize to 0.
105 OUTPUT @Sa;"MOV HARMNC,FUNDMNTL";  ! Let HARMNC equal
                                     ! FUNDMNTL.
110 OUTPUT @Sa;"VARDEF HARMNUM,0";     ! Initialize HARMNUM to 0.
120 OUTPUT @Sa;"IP;SP 1MHZ";          ! Instrument preset; set span.
130  !
140 OUTPUT @Sa;"REPEAT";              ! Begin loop.
150 OUTPUT @Sa;"  ADD HARMNUM,HARMNUM,1";
                                     ! Increment to next harmonic.
160 OUTPUT @Sa;"  MPY HARMNC,FUNDMNTL,HARMNUM";
                                     ! Harm freq=fund * harm num.
170 OUTPUT @Sa;"  MOV CF,HARMNC";     ! Set center frequency.
180 OUTPUT @Sa;"  TS";                ! Take a sweep to update display.
190 OUTPUT @Sa;"UNTIL HARMNC,GE,3E9"; ! End of loop.
200 OUTPUT @Sa;"@";                  ! Logical end of DLP function.
210  !
220 OUTPUT @Sa;"MPYEX";              ! Execute function.
230  !
240 ASSIGN @Sa TO *                  ! Close I/O path.
250  END
```

MSDEV

Mass Storage Device

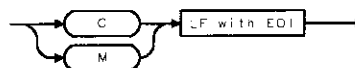
Syntax



x1253a

Description Establishes the data storage and access device as either the module memory or the memory card.

Query Response



QMSDEV

Program Example

```

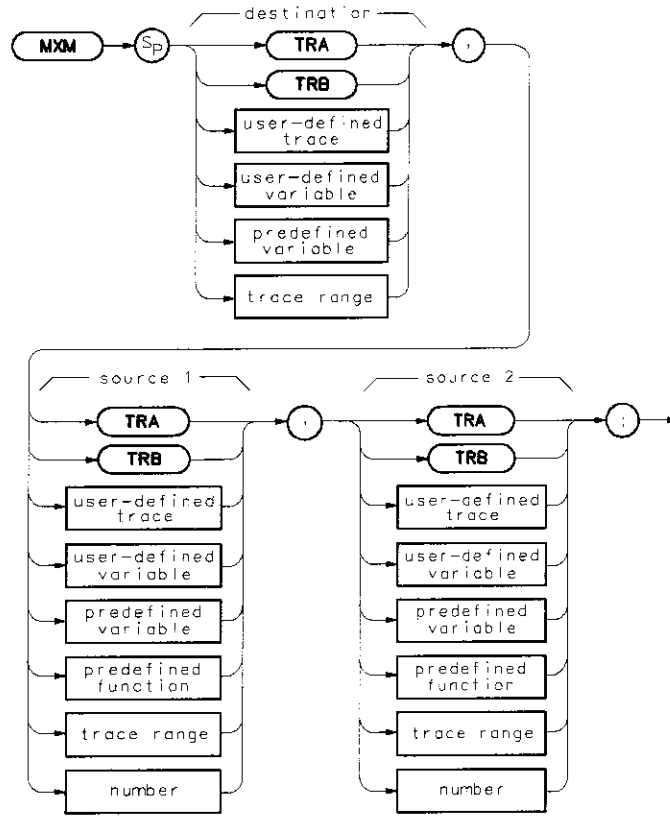
10  ! The following example shows the use of the MSDEV command. The
20  ! instructions within the FUNCDEF '@' delimiters form a structure
30  ! called a DLP (downloadable program).
40  !
50  ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
60  !
70  OUTPUT @Sa;"FUNCDEF SAVE_TRC,@;"; ! Logical start of the DLP.
80  OUTPUT @Sa;"IP;CF300MHZ;SP 1MHZ;TS;"; ! Set up the analyzer.
90  OUTPUT @Sa;"MSDEV CARD;";        ! Tell the analyzer where the
100 !
110 OUTPUT @Sa;"MODSAVT TRA,%CAL_OUT%;"; ! Store the trace on the card.
120 OUTPUT @Sa;"@";                  ! Logical end of the DLP.
130 OUTPUT @Sa;"SAVE_TRC;";         ! Execute the DLP.
140 !
150 ASSIGN @Sa TO*                   !Close I/O path.
160 END

```

MXM

Maximum

Syntax



xf254a

Description Compares source 2 with source 1, point by point, and sends the greater value of each comparison to the destination.

Program Example

```

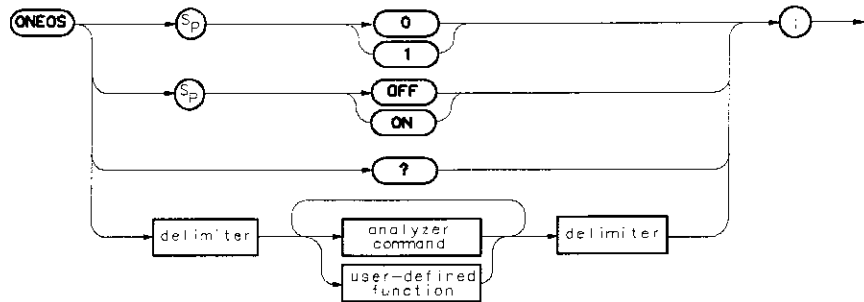
10      ! The following example shows a use of the MXM command.  The example
20      ! requires an input signal of 300 MHz, such as the CAL OUTPUT signal.
30      !
40      ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
50      !
60      OUTPUT @Sa;"IP;";                ! Instrument preset.
70      OUTPUT @Sa;"CF 300MHZ; SP 200KHZ;";
                                           ! Set center frequency and span.
80      OUTPUT @Sa;"CLRW TRA;";          ! Clear-write trace A.
90      OUTPUT @Sa;"CLRW TRB;";          ! Clear-write trace B.
100     OUTPUT @Sa;"TS;";                ! Update the display.
110     OUTPUT @Sa;"VIEW TRA;";          ! View trace A.
120     !
130     FOR Counter=1 TO 10
140         OUTPUT @Sa;"APB;";            ! A + B --> A
150     NEXT Counter
160     OUTPUT @Sa;"AMB ON;";            ! A - B --> A
170     !
180     OUTPUT @Sa;"MXM TRA,TRA,TRB;";    ! Compare trace A and trace B.
190     ! Put the maximum point in trace A.
200     OUTPUT @Sa;"BLANK TRB;";        ! Blank trace B to view the results.
210     ASSIGN @Sa TO *                  ! Close I/O path.
220     END

```

ONEOS

On End of Sweep

Syntax



Description ONEOS is a predefined function name. The contents of ONEOS are executed at the end of a sweep, after completing trace processing and all other end-of-sweep functions.

User memory is required to execute the ONEOS command. The ONEOS command allocates memory which can be freed with the DISPOSE command.

Restrictions: Subject to available memory. The following commands consume memory: ONEOS, ARRAYDEF, TRDEF, KEYDEF, LCLVAR, VARDEF REPEAT/UNTIL, and IF/THEN.

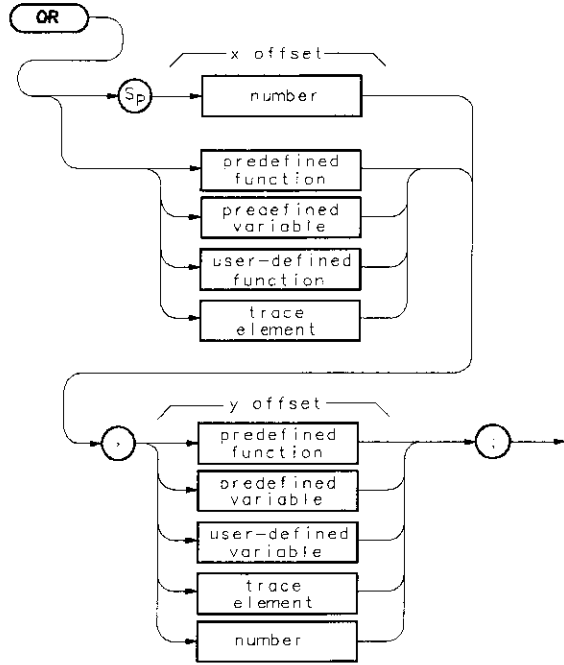
Program Example

```
10 ! The following example shows the use of the ONEOS command. The
20 ! example uses an input signal of 300 MHz such as the CAL OUTPUT.
30 !
40 ASSIGN @Sa TO 718 ! Assign I/O path to address 718.
50 !
60 OUTPUT @Sa;"IP;ST 1SC;TS;"; ! Instrument preset; set sweep
70 ! time; take a sweep.
80 OUTPUT @Sa;"ONEOS $CF 300MHZ;SP 1MHZ;$";
! At the end of the following
! sweep, set frequency and span.
90
100 OUTPUT @Sa;"TS;ONEOS OFF;"; ! Take sweep; turn ONEOS off.
110 !
120 ! Note: If the ONEOS OFF command is not given, the analyzer will
130 ! continue to set the frequency and span at the end of each sweep.
140 !
150 ASSIGN @Sa TO * ! Close I/O path.
160 END
```

OR

Set Origin of Graphics Pen

Syntax



x1256a

Description Use this command to set the origin of the graphics pen as determined by the values of the x- and y-coordinate offsets.

Note For the OR command to function properly in conjunction with the DSPLY command, you must use PU or PA after OR to position the graphics pen correctly.

Prerequisite Command: VARDEF

Related Commands: PA and PR

Preset State: OR 0,0

OR

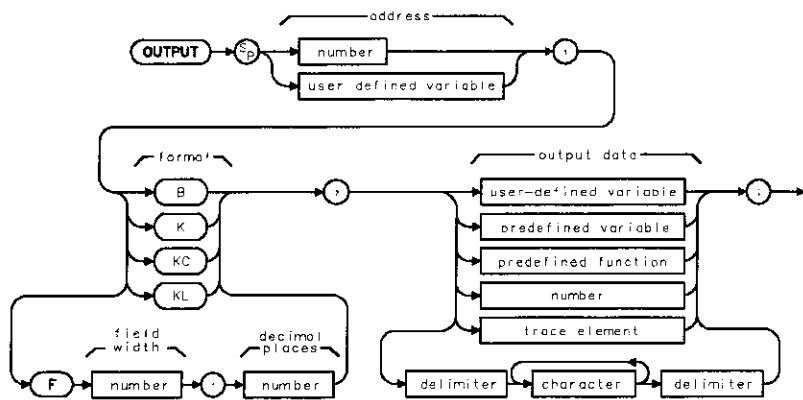
Program Example

```
10    ! The following example shows the use of the OR command. The example
20    ! creates and displays two boxes. The function 'BOX' always draws the
30    ! same box but it appears in a different area of the screen according
40    ! to the position of the ORigin.
50    !
60    ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
70    OUTPUT @Sa;"FUNCDEF DRAW_BOXES,$"; ! Logical start of function.
80    OUTPUT @Sa;" IP;";                ! Instrument preset.
90    OUTPUT @Sa;" CLRDSP;";            ! Clear the analyzer screen.
100   OUTPUT @Sa;" OR 0,0;";            ! Set origin to lower left corner
110   !                                  ! of graticule.
120   OUTPUT @Sa;" BOX;";                ! Draw box.
130   OUTPUT @Sa;" OR 200,200;";        ! Move origin to a different spot.
140   OUTPUT @Sa;" BOX;";                ! Draw the same box.
150   !                                  ! Note that it appears in a
160   !                                  ! different place on the analyzer.
170   OUTPUT @Sa;"$";                    ! Logical end of function.
180   !
190   !
200   OUTPUT @Sa;"FUNCDEF BOX,@;";       ! Logical start of function 'BOX'.
210   OUTPUT @Sa;" PU;PA 200,200;";      ! Move pen to starting point.
220   OUTPUT @Sa;" PD;PR 240,0;";        ! \
230   OUTPUT @Sa;" PR 0,240;";            ! \ Draw
240   OUTPUT @Sa;" PR -240,0;";          ! / rectangle.
250   OUTPUT @Sa;" PR 0,-240;";          ! /
260   OUTPUT @Sa;"@";                    ! Logical end of function.
270   !
280   OUTPUT @Sa;"DRAW_BOXES;";          ! Execute DLP.
290   ASSIGN @Sa TO *                    ! Close I/O path.
300   END
```

OUTPUT

Output to HP-IB

Syntax



x1257a

Description

This command sends output data in the format appropriate to the addressed device. The data may be from a function definition or a DLP. The command assumes the CTRLHPIB command has been executed to place the spectrum analyzer in controller mode. If a controller is detected on HP-IB, the command is aborted. The output format is determined by the format field specifications.

Related commands: RELHPIB, ENTER, and CTRLHPIB

OUTPUT

Parameters	Address	Integers 0 to 30 are for the GPIB interface, excluding the HP-IB address of the spectrum analyzer. Integers 98 and 99 are for the Auxiliary interface.* 98 outputs with the least-significant bit (LSB) of the serial data byte first. 99 outputs with the most-significant bit (MSB) of the serial data byte first.
	Format	B Output a single 8-bit byte to the specified interface. F Output a number in ASCII format, using the specified field width and precision, with LF and END terminators. K Output in ASCII format, without a terminator, to a free field. KC Output in ASCII format, with CR and LF terminators, to a free field. KL Output in ASCII format, with LF and END terminators, to a free field.

Related Commands: CTRLHPIB, RELHPIB, and ENTER

* These auxiliary interface connector commands can only be accessed remotely using DLPs, and are only available with 85620A with serial prefix 3143A or higher and firmware 910116 and later date codes.

Program Example

```

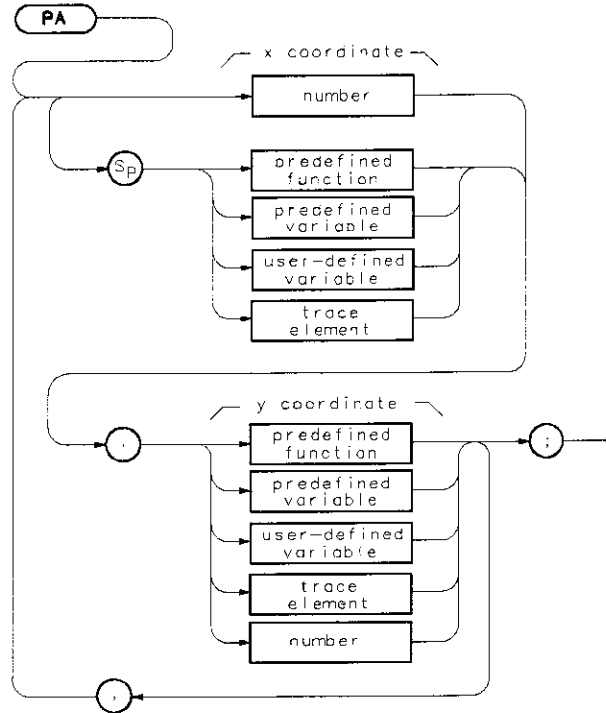
10      ! The following example shows how to use the OUTPUT command to have
20      ! the analyzer send data to another HP-IB device.  The instructions
30      ! within the FUNCDEF '@' delimiters form a structure called a
40      ! DLP (downloadable program).
50      !
60      ! NOTE: A printer (HP-IB address 01) needs to be connected.
70      !
80      ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
90      !
100     OUTPUT @Sa;"FUNCDEF E_XMPLE,@;";    ! Logical start of the DLP.
110     OUTPUT @Sa;"  CTRLHPIB;";          ! Tells analyzer to take control
120                                           ! of the bus.
130     OUTPUT @Sa;"  MKN;";                ! Turn on a marker.
140     OUTPUT @Sa;"  OUTPUT 1,KC,'THIS IS AN EXAMPLE OF THE ANALYZER
SENDING';";
150     OUTPUT @Sa;"  OUTPUT 1,KC,'DATA TO ANOTHER HPIB DEVICE';";
160     OUTPUT @Sa;"  OUTPUT 1,KC,";";
170     OUTPUT @Sa;"  OUTPUT 1,K,'MARKER FREQUENCY = ';";
180     OUTPUT @Sa;"  OUTPUT 1,K,MKF;";
190     OUTPUT @Sa;"  OUTPUT 1,KC,' Hz';";
200     OUTPUT @Sa;"  OUTPUT 1,KC,";";
210     OUTPUT @Sa;"  RELHPIB;";            ! Release HP-IB control.
220     OUTPUT @Sa;"@";                    ! Logical end of the DLP.
230     !
240     OUTPUT @Sa;"E_XMPLE;";              ! Have the analyzer execute the
250                                           ! DLP E_XMPLE.
260     SEND 7;UNL TALK 18 LISTEN 1 DATA  ! Send to HP-IB select code 7 the
270                                           ! commands necessary to allow
280                                           ! the analyzer (address 18) to
290                                           ! talk to the printer (address 1)
300                                           ! while this controller is still
310                                           ! connected.
320     LOCAL 7                            ! Local HP-IB select code 7.
330     WAIT 2                              ! Wait for printer to finish.
340     REMOTE 7                            ! Toggle the HP-IB REN line.
350     LOCAL @Sa                          ! Local the SA.
360     !
370     ASSIGN @Sa TO *                    ! Close I/O path.
380     END

```

PA

Plot Absolute

Syntax



x1258a

Description This command moves the pen from its current position to the position specified by the coordinates with respect to the origin (refer to the OR command). A line is drawn if the pen is down. Refer to PD (pen down) or PU (pen up) commands. The x and y coordinates are in measurement units.

Note With the origin at (0,0), the lower left-hand corner of the spectrum analyzer graticule is at (100,100) and the upper right-hand corner is at (700,700).

Parameters The following parameters apply with OR set to 0,0.
X Maximum = 710 Y Maximum = 710 X Minimum = 90 Y Minimum = 25

Related Commands: OR, PU, PD, and DSPLY

Program Example

```

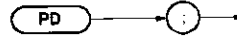
10  ! The following example shows the use of the PA, PR, PU, and PD commands.
20  ! The figures drawn are an ellipse within a rectangle.
30  !
40  ASSIGN @Sa TO 718           ! Assign I/O path to address 718.
50  OUTPUT @Sa;"IP;";          ! Instrument preset.
60  OUTPUT @Sa;"CLRDSP;";      ! Clear the analyzer screen.
70  !
80  ! Draw a rectangle with fixed points.
90  OUTPUT @Sa;"PU;PA 300,300;"; ! Move pen to starting point.
100 OUTPUT @Sa;"PD;PR 240,0;";  ! \
110 OUTPUT @Sa;"PR 0,240;";     ! \ Draw
120 OUTPUT @Sa;"PR -240,0;";    ! / rectangle.
130 OUTPUT @Sa;"PR 0,-240;";    ! /
140 !
150 ! Draw an ellipse with calculated points.
160 OUTPUT @Sa;"PU;PA 415,305; PD;"; ! Move pen to starting point.
170 FOR I=0 TO 2*PI STEP PI/40
180   X=PROUND(10*COS(I),-3)     ! \ Calculate x and y
190   Y=PROUND(10*SIN(I),-3)    ! / coordinates.
200   !
210   OUTPUT @Sa;"PR "&VAL$(X)&","&VAL$(Y)&";";
                                   ! Create plot statement using
                                   ! calculated coordinates.
220
230 NEXT I
240 !
250 ASSIGN @Sa TO *             ! Close I/O path.
260 END

```

PD

Pen Down

Syntax



x1259a

Description This command causes the pen to draw a line when the PA or PR commands are activated.

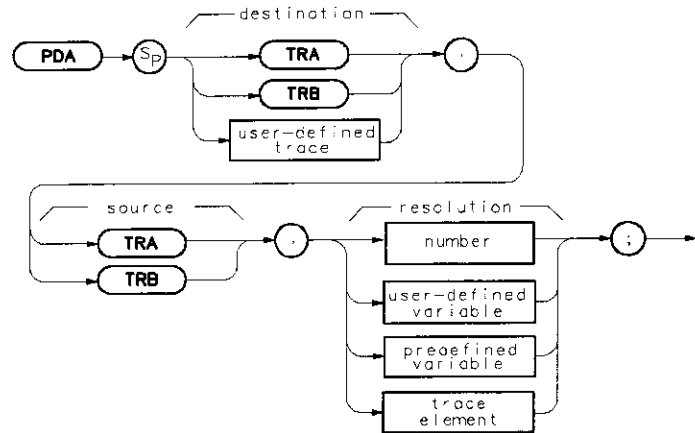
Program Example

```
10 ! The following example shows the use of the PA, PR, PU, and PD commands.
20 ! The figures drawn are an ellipse within a rectangle.
30 !
40 ASSIGN @Sa T0 718 ! Assign I/O path to address 718.
50 OUTPUT @Sa;"IP;"; ! Instrument preset.
60 OUTPUT @Sa;"CLRDISP;"; ! Clear the analyzer screen.
70 !
80 ! Draw a rectangle with fixed points.
90 OUTPUT @Sa;"PU;PA 300,300;"; ! Move pen to starting point.
100 OUTPUT @Sa;"PD;PR 240,0;"; ! \
110 OUTPUT @Sa;"PR 0,240;"; ! \ Draw
120 OUTPUT @Sa;"PR -240,0;"; ! / rectangle.
130 OUTPUT @Sa;"PR 0,-240;"; ! /
140 !
150 ! Draw an ellipse with calculated points.
160 OUTPUT @Sa;"PU;PA 415,305; PD;"; ! Move pen to starting point.
170 FOR I=0 TO 2*PI STEP PI/40
180 X=PROUND(10*COS(I),-3) ! \ Calculate x and y
190 Y=PROUND(10*SIN(I),-3) ! / coordinates.
200 !
210 OUTPUT @Sa;"PR "&VAL$(X)&","&VAL$(Y)&";"; ! Create plot statement using
! calculated coordinates.
220
230 NEXT I
240 !
250 ASSIGN @Sa T0 * ! Close I/O path.
260 END
```

PDA

Probability Distribution of Amplitude

Syntax



Description

Replace the destination trace with the amplitude distribution function of the source trace. The source trace data is taken point by point. The value for bottom-screen is subtracted from the value of each data point, then the difference is divided by the resolution value (which is rounded to an integer). If the result falls within the bucket range of the destination trace (0 to 600), the corresponding destination trace element is increased by one. The function is complete after all source points are dealt with.

Resolution Range: Limited by source amplitude.

PDA

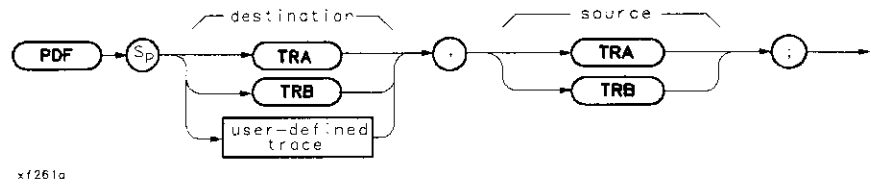
Program Example

```
10      ! The following example shows how to use the PDA command to determine
20      ! the probability distribution of the amplitude values in trace A.
30      ! The results are placed in trace B and expanded by a factor of 20
40      ! to make the results more visible.
50      !
60      ASSIGN @Sa TO 718          ! Assign I/O path to address 718.
70      !
80      OUTPUT @Sa;"FUNCDEF TST,@"; ! Logical start of DLP.
90      OUTPUT @Sa;"IP;SNGLS;";    ! Instrument preset the analyzer.
100     OUTPUT @Sa;"VB10KZ;HD;TS;"; ! Set video BW at 10 kHz.
110     OUTPUT @Sa;"MOV TRB,0;";    ! Put all zeros into trace B.
120     OUTPUT @Sa;"PDA TRB,TRA,.8;"; ! Determine the distribution
130     ! of trace A and put results
140     ! into trace B.
150     OUTPUT @Sa;"MPY TRB,TRB,20;"; ! Multiply values in trace B
160     ! by 20.
170     OUTPUT @Sa;"VIEW TRB;";    ! View trace B.
180     OUTPUT @Sa;"@";           ! Logical end of DLP.
190     !
200     OUTPUT @Sa;"TST;";        ! Execute the DLP.
210     !
220     LOCAL @Sa                 ! Local analyzer for manual use.
230     ASSIGN @Sa TO *          ! Close I/O path.
240     END
```

PDF

Probability Distribution of Frequency

Syntax



Description

Replace the destination trace with the frequency distribution function of the source trace. Source trace elements falling above the TH command threshold level increase the corresponding destination trace elements amplitude by one display unit. The default threshold value is nine major divisions below the reference level.

Program Example

```

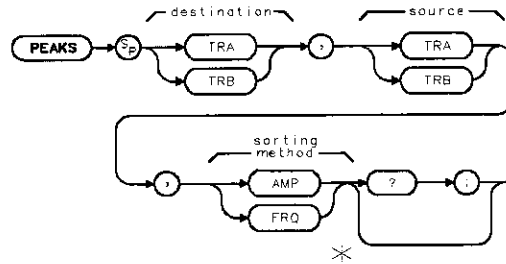
10 OUTPUT 718;"TRDEF SAMPLE,50;";
20 OUTPUT 718;"PDF SAMPLE,TRA;";
30 END

```

PEAKS

Trace Peaks

Syntax



x1262a

Note

This bypass command path is only legal if you use PEAKS as a predefined function. It must reside within a compatible-function operation.

Description

The PEAKS command sorts signal peaks by frequency or amplitude, stores in the destination trace the horizontal position of each peak in position units, then computes the number of peaks found. The value set by MKPT is the minimum amplitude level from which a peak on a trace can be detected. When sorting by frequency, PEAKS first computes the horizontal position of all peaks in the sort trace. These positions are consecutively loaded into the destination trace, the lowest horizontal position value occupying the first element. Thus, the destination trace amplitude values from left to right correspond to signal frequencies from low to high.

When sorting by amplitude, PEAKS first computes the amplitude of all peaks in the source trace in measurement units, and sorts these values from high to low. The horizontal positions of the peaks are then loaded into the destination trace, with the highest value occupying the first element. Thus, the destination trace amplitude values from left to right correspond to the horizontal positions of the source trace amplitudes sorted from high to low.

PEAKS must be used as either a query or as a source in another command function. Form a query by ending the PEAKS statement with a question mark (?). When used as a query, PEAKS returns the number of peaks found.

Use PEAKS as a source by incorporating the PEAKS statement into any command having "predefined function" in its syntax diagram. When PEAKS is used as a source, the number of peaks found is used for operation by the command that contains PEAKS.

Query Response The PEAKS command outputs the number of signal peaks found.

Program Example

```

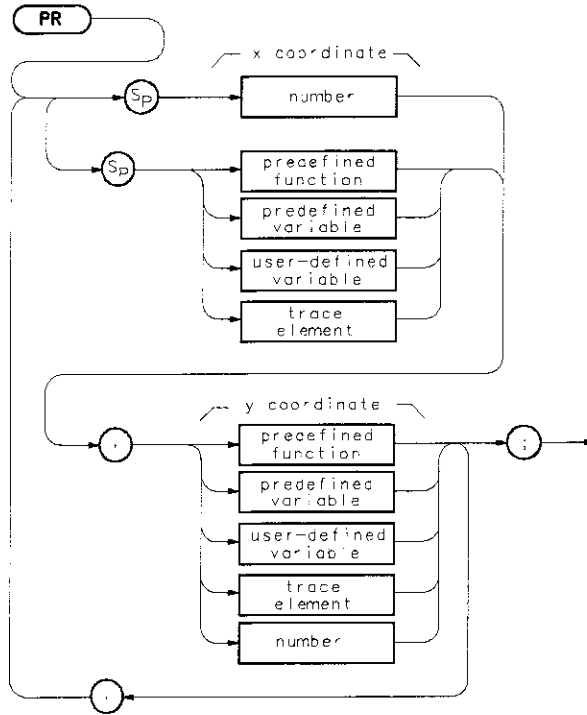
10  ! The following example shows the use of the PEAKS command. The example
20  ! uses an input signal of 300 MHz such as the CAL OUTPUT and looks at its
30  ! harmonics.
40  !
50  ASSIGN @Sa TO 718                ! Assign I/O/ path to address 718.
60  !
70  OUTPUT @Sa;"FUNCDEF NUM_PKS,@;"; ! Logical start of DLP.
80  OUTPUT @Sa;"VARDEF NUM_PEAKS,0;"; ! Create and initialize variable.
90  OUTPUT @Sa;"FA 250MHZ;FB 2900MHZ;"; ! Set start and stop frequencies.
100 OUTPUT @Sa;"MKPT -60DB;MKPX 3DB;TS;"; ! Set the threshold and marker-
110                                     ! peak excursion. Take a sweep.
120 OUTPUT @Sa;"MOV NUM_PEAKS,PEAKS TRB,TRA,AMP;";
                                     ! Sort signals by amplitude
130                                     ! and place results in trace B.
140                                     ! Trace B now contains position
150                                     ! information.
160 OUTPUT @Sa;"@";                  ! Logical end of DLP.
170 !
180 OUTPUT @Sa;"NUM_PKS;";           ! Execute DLP.
190 !
200 OUTPUT @Sa;"NUM_PEAKS?;";        ! How many peaks?
210 ENTER @Sa;Number_of_peaks        ! Get returned value.
220 PRINT "The number of signals above -60 dBm is ",Number_of_peaks
230 !
240 ASSIGN @Sa TO *                  ! Close I/O path.
250 END

```

PR

Plot Relative

Syntax



x126.3a

Description This command moves the pen from its current location to a position determined by adding the new x and y coordinates to the current x , y position.

Prerequisite Commands: VARDEF when using user-defined variable

Related Commands: OR PU, and PD

Program Example

```

10      ! The following example shows the use of the PA, PR, PU, and PD commands.
20      ! The figures drawn are an ellipse within a rectangle.
30      !
40      ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
50      OUTPUT @Sa;"IP;";                ! Instrument preset.
60      OUTPUT @Sa;"CLRDSP;";           ! Clear the analyzer screen.
70      !
80      ! Draw a rectangle with fixed points.
90      OUTPUT @Sa;"PU;PA 300,300;";    ! Move pen to starting point.
100     OUTPUT @Sa;"PD;PR 240,0;";      ! \
110     OUTPUT @Sa;"PR 0,240;";         ! \ Draw
120     OUTPUT @Sa;"PR -240,0;";        ! / rectangle.
130     OUTPUT @Sa;"PR 0,-240;";       ! /
140     !
150     ! Draw an ellipse with calculated points.
160     OUTPUT @Sa;"PU;PA 415,305; PD;"; ! Move pen to starting point.

170     FOR I=0 TO 2*PI STEP PI/40
180         X=PROUND(10*COS(I),-3)      ! \ Calculate x and y
190         Y=PROUND(10*SIN(I),-3)      ! / coordinates.
200     !
210         OUTPUT @Sa;"PR "&VAL$(X)&","&VAL$(Y)&";";
                                           ! Create plot statement using
                                           ! calculated coordinates.
220
230     NEXT I
240     !
250     ASSIGN @Sa TO *                ! Close I/O path.
260     END

```

PU

Pen Up

Syntax



x1254c

Description This command causes the pen to stop drawing at the location specified in the last plot statement.

Related Commands: PA, PR, and DSPLY

Program Example

```

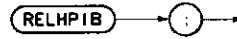
10      ! The following example shows the use of the PA, PR, PU, and PD commands.
20      ! The figures drawn are an ellipse within a rectangle.
30      !
40      ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
50      OUTPUT @Sa;"IP;";                ! Instrument preset.
60      OUTPUT @Sa;"CLRDSF;";           ! Clear the analyzer screen.
70      !
80      ! Draw a rectangle with fixed points.
90      OUTPUT @Sa;"PU;PA 300,300;";    ! Move pen to starting point.
100     OUTPUT @Sa;"PD;PR 240,0;";      ! \
110     OUTPUT @Sa;"PR 0,240;";         ! \ Draw
120     OUTPUT @Sa;"PR -240,0;";        ! / rectangle.
130     OUTPUT @Sa;"PR 0,-240;";       ! /
140     !
150     ! Draw an ellipse with calculated points.
160     OUTPUT @Sa;"PU;PA 415,305; PD;"; ! Move pen to starting point.
170     FOR I=0 TO 2*PI STEP PI/40
180         X=PROUND(10*COS(I),-3)      ! \ Calculate x and y
190         Y=PROUND(10*SIN(I),-3)      ! / coordinates.
200         !
210         OUTPUT @Sa;"PR "&VAL$(X)&","&VAL$(Y)&";";
220         ! Create plot statement using
230         ! calculated coordinates.
240     NEXT I
250     !
260     ASSIGN @Sa TO *                ! Close I/O path.
260     END

```

RELHPIB

Release HP-IB

Syntax



x1285a

Description

This command releases HP-IB control taken by the CTRLHPIB command. Refer also to the commands ENTER, OUTPUT, and CTRLHPIB.

Restrictions: Subject to available memory. The following commands consume memory: ONEOS, ARRAYDEF, TRDEF, KEYDEF, LCLVAR, VARDEF REPEAT/UNTIL, and IF/THEN.

Prerequisite Commands: FUNCDEF when using user-defined function. VARDEF or LCLVAR when using user-defined variable, user-defined array, or user-defined trace.

Related Commands: ABORT and RETURN

Program Example

```

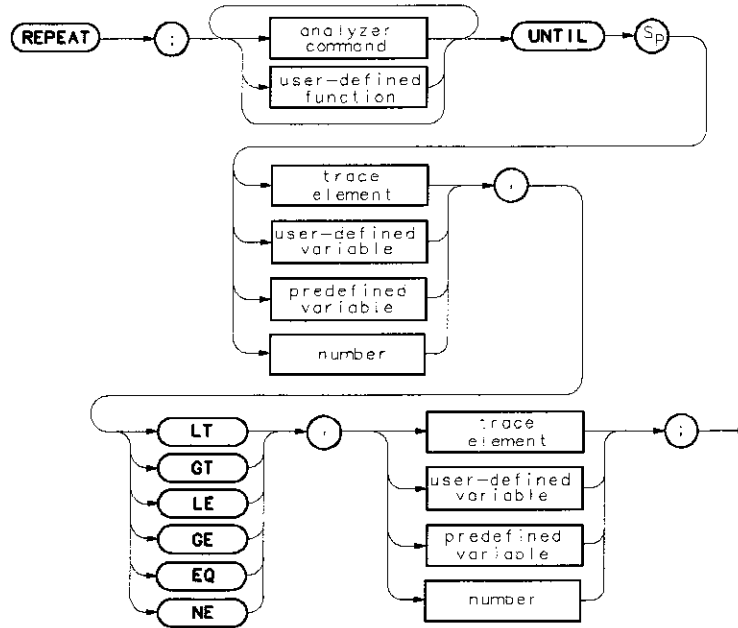
10      ! The following example shows how to use the RELHPIB command to have
20      ! the analyzer send data to another HP-IB device. The instructions
30      ! within the FUNCDEF '@' delimiters form a structure called a
40      ! DLP (downloadable program).
50      !
60      ! NOTE: A printer (HP-IB address 01) needs to be connected.
70      !
80      ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
90      !
100     OUTPUT @Sa;"FUNCDEF E_XMPLE,@;";
                                           ! Logical start of the DLP.
110     OUTPUT @Sa;"  CTRLHPIB;";        ! Tells analyzer to take control
120                                           ! of the bus.
130     OUTPUT @Sa;"  MKN;";            ! Turn on a marker.
140     OUTPUT @Sa;"  OUTPUT 1,KC,'THIS IS AN EXAMPLE OF THE ANALYZER
SENDING';";
150     OUTPUT @Sa;"  OUTPUT 1,KC,'DATA TO ANOTHER HPIB DEVICE';";
160     OUTPUT @Sa;"  OUTPUT 1,KC,";";
170     OUTPUT @Sa;"  OUTPUT 1,K,'MARKER FREQUENCY = '";
180     OUTPUT @Sa;"  OUTPUT 1,K,MKF;";
190     OUTPUT @Sa;"  OUTPUT 1,KC,' Hz';";
200     OUTPUT @Sa;"  OUTPUT 1,KC,";";
210     OUTPUT @Sa;"  RELHPIB;";        ! Release HP-IB control.
220     OUTPUT @Sa;"@";                ! Logical end of the DLP
230     !.
240     OUTPUT @Sa;"E_XMPLE;";        ! Have the analyzer execute the
250                                           ! DLP E_XMPLE.
260     SEND 7;UNL TALK 18 LISTEN 1 DATA
                                           ! Send to HP-IB select code 7 the
270                                           ! commands necessary to allow
280                                           ! the analyzer (address 18) to
290                                           ! talk to the printer (address 1)
300                                           ! while this controller is still
310                                           ! connected.
320     LOCAL 7                        ! Local HP-IB select code 7.
330     WAIT 2                          ! Wait for printer to finish.
340     REMOTE 7                         ! Toggle the HP-IB REN line.
350     LOCAL @Sa                       ! Local the SA.
360     !
370     ASSIGN @Sa TO *                 ! Close I/O path.
380     END

```

REPEAT/UNTIL

Repeat/Until

Syntax



xf266a

Description

Repeat The REPEAT command determines the starting point of the looping process. UNTIL determines the end of the REPEAT/UNTIL command loop based on the comparison of two variables. Valid conditions are less than (LT), greater than (GT), less than or equal to (LE), greater than or equal to (GE), equal (EQ), and not equal (NE). The commands within the repeat/until loop are executed until the comparison result is true.

This command requires user memory while executing as a remote command. When the command is complete, memory is returned as available user memory.

Program Example #1

```

10  ! The following example shows the use of the REPEAT/UNTIL commands
20  ! in a FOR/NEXT loop context.  A previously downloaded function may
30  ! be executed repeatedly from within an immediate execute download.
40  ! This is one way of nesting REPEATs.
50  !
60  ASSIGN @Sa TO 718          ! Assign I/O path to address 718.
70  !
80  OUTPUT @Sa;"IP;";          ! Instrument preset.
90  OUTPUT @Sa;"VARDEF OUTER_LP,0;"; ! Create variable and initialize to 0.
100 OUTPUT @Sa;"VARDEF TOTAL,0;"; ! Create variable and initialize to 0.
110 !
120 OUTPUT @Sa;"FUNCDEF CNT_LP,@;"; ! Logical start of the DLP function.
130 OUTPUT @Sa;" VARDEF INNER_LP,0;"; ! Create and initialize inner loop counter.
140 OUTPUT @Sa;" REPEAT;";      ! Begin inner loop.
150 OUTPUT @Sa;"   ADD TOTAL,TOTAL,1;"; ! Increment running total.
160 OUTPUT @Sa;"   ADD INNER_LP,INNER_LP,1;";
                                   ! Increment inner loop counter.
170 OUTPUT @Sa;" UNTIL INNER_LP,EQ,3;"; ! End of inner loop.
180 OUTPUT @Sa;"@";
190 !
200 OUTPUT @Sa;"REPEAT;";      ! Begin outer loop.
210 OUTPUT @Sa;" CNT_LP;";     ! Execute function.
220 OUTPUT @Sa;" ADD OUTER_LP,OUTER_LP,1;";
                                   ! Increment outer loop counter.
230 OUTPUT @Sa;"UNTIL OUTER_LP,EQ,4;"; ! End of outer loop.
240 !
250 OUTPUT @Sa;"CLR DSP;";
260 OUTPUT @Sa;"PU;PA 10,500;TEXT %INNER LOOP COUNTER = %;";
270 OUTPUT @Sa;"DSPLY INNER_LP,5,2;";
280 OUTPUT @Sa;"PU;PA 10,400;TEXT %OUTER LOOP COUNTER = %;";
290 OUTPUT @Sa;"DSPLY OUTER_LP,5,2;";
300 OUTPUT @Sa;"PU;PA 10,300;TEXT %TOTAL ITERATIONS = %;";
310 OUTPUT @Sa;"DSPLY TOTAL,6,2;";
320 !
330 ASSIGN @Sa TO *          ! Close I/O path.
340 END

```

REPEAT/UNTIL

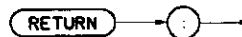
Program Example #2

```
10  ! The following example shows the use of the REPEAT/UNTIL commands
20  ! in a FOR/NEXT loop context.  This is an immediate-execute DLP
30  ! (downloadable program).  This is another way of nesting REPEATs.
40  !
50  ASSIGN @Sa TO 718          ! Assign I/O path to address 718.
60  !
70  OUTPUT @Sa;"IP;";         ! Instrument preset.
80  OUTPUT @Sa;"VARDEF OUTER_LP,0;"; ! Create variable and initialize to 0.
90  OUTPUT @Sa;"VARDEF TOTAL,0;"; ! Create variable and initialize to 0.
100 !
110 OUTPUT @Sa;"REPEAT;";     ! Begin outer loop.
120 OUTPUT @Sa;"  VARDEF INNER_LP,0;"; ! Create and initialize inner loop counter.
130 OUTPUT @Sa;"  REPEAT;";   ! Begin inner loop.
140 OUTPUT @Sa;"    ADD TOTAL,TOTAL,1;"; ! Increment running total
150 OUTPUT @Sa;"    ADD INNER_LP,INNER_LP,1;";
                                ! Increment inner loop counter.
160 OUTPUT @Sa;"  UNTIL INNER_LP,EQ,3;"; ! End of inner loop.
170 OUTPUT @Sa;"    ADD OUTER_LP,OUTER_LP,1;";
                                ! Increment outer loop counter.
180 OUTPUT @Sa;"UNTIL OUTER_LP,EQ,4;"; ! End of outer loop.
190 !
200 OUTPUT @Sa;"CLR DSP;";
210 OUTPUT @Sa;"PU;PA 10,500;TEXT %INNER LOOP COUNTER=%;";
220 OUTPUT @Sa;"DSPLY INNER_LP,5,2;";
230 OUTPUT @Sa;"PU;PA 10,400;TEXT %OUTER LOOP COUNTER=%;";
240 OUTPUT @Sa;"DSPLY OUTER_LP,5,2;";
250 OUTPUT @Sa;"PU;PA 10,300;TEXT %TOTAL ITERATIONS=%;";
260 OUTPUT @Sa;"DSPLY TOTAL,5,2;";
270 !
280 ASSIGN @Sa TO *          ! Close I/O path.
290 END
```

RETURN

Return from Function

Syntax



x1287a

Description

Use this command to return from a user-defined function before its completion. The program returns to the point from which the function was called.

Related Commands: IF/THEN and FUNCDEF

RETURN

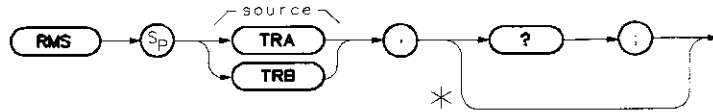
Program Example

```
10  ! Program to test the RETURN command for use in DLPs. Two FUNCDEFs
20  ! are defined, C_ALLER and C_ALLEE. C_ALLER sets the spectrum analyzer
30  ! up in single sweep mode with a 1 second sweeptime, so instrument
40  ! state changes can be easily observed. C_ALLER first sets the CF to
50  ! 1 GHz and then calls C_ALLEE. C_ALLEE tests the CF value, and if it
60  ! is less than 1.5 GHz, it changes the span to 300 MHz and takes
70  ! another sweep. C_ALLEE then changes the span to 100 MHz, takes
80  ! another sweep, and returns to C_ALLER.
90  !
100 ! C_ALLER then changes the CF to 2 GHz, takes a sweep, and again calls
110 ! C_ALLEE. Since the CF is now greater than 1.5 GHz, C_ALLEE returns
120 ! immediately to C_ALLER changes the span to 200 MHz and takes a sweep.
130 !-----
140 !
150 ASSIGN @Sa TO 718
160 !
170 OUTPUT @Sa;"FUNCDEF C_ALLER,@";
180 OUTPUT @Sa;"IP;SNGLS;ST 1SC;";
190 OUTPUT @Sa;"CF 1GZ;TS;SP 500MZ;TS;";
200 OUTPUT @Sa;"C_ALLEE;";
210 OUTPUT @Sa;"CF 2 GZ;TS;";
220 OUTPUT @Sa;"C_ALLEE;";
230 OUTPUT @Sa;"SP 200MZ;TS;";
240 OUTPUT @Sa;"@";
250 !
260 OUTPUT @Sa;"FUNCDEF C_ALLEE,@";
270 OUTPUT @Sa;"IF CF,LT,1.5E+9;THEN;";
280 OUTPUT @Sa;"SP 300MZ;TS;";
290 OUTPUT @Sa;"ELSE;";
300 OUTPUT @Sa;"RETURN;";
310 OUTPUT @Sa;"SP 1GZ;TS;";
320 OUTPUT @Sa;"ENDIF;";
330 OUTPUT @Sa;"SP 100MZ;TS;";
340 OUTPUT @Sa;"@";
350 !
360 ASSIGN @Sa TO *
370 END
```

RMS

Trace Root Mean Square Value

Syntax



xf268a

Note

This bypass command path is only legal if you use RMS as a predefined function. It must reside within a compatible-function operation.

Description

The root mean square (rms) value of a trace is determined and returned in measurement units.

Restrictions: Subject to available memory. The following commands consume memory: ONEOS, ARRAYDEF, TRDEF, KEYDEF, LCLVAR, VARDEF REPEAT/UNTIL, and IF/THEN.

Program Example

```

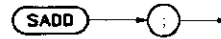
10  ! The following example shows the use of the RMS command.
20  !
30  ASSIGN @Sa TO 718          ! Assign an I/O path.
40  !
50  OUTPUT @Sa;"IP;";         ! Instrument preset.
60  OUTPUT @Sa;"SNGLS;";     ! Single sweep mode.
70  OUTPUT @Sa;"TS;RMS TRA,?;"; ! Take sweep; find RMS value.
80  ENTER @Sa;Rms_value      ! Get RMS value from analyzer.
90  !
100 Rms_value=PROUND(Rms_value,-2) ! Round the value to two decimal
110                               ! places.
120 PRINT "The RMS value of the trace is ";Rms_value;" measurement units"
130 !
140 ASSIGN @Sa TO *          ! Close I/O path.
150 END

```

SADD

Add a Limit-Line Segment

Syntax



x f269a

Description This command is used to add a limit-line segment to the current limit line.

Related commands include: LIMU, LIML, LImm, LIMD, LIMIFL, LIMISL, LIMISP, SDEL, SEDI

Program Example

```

130  ! This program demonstrates the use of the SADD command
140  ! to generate a limit line:
150  !
160  OUTPUT 718;"LIMIPURGE;";          ! Purge current limit line.
170  OUTPUT 718;"EDITLIML;";          ! Begin edit of limit line.
180  OUTPUT 718;"LIMIREL OFF;";       ! Make sure it's an absolute line.
190  OUTPUT 718;"SADD;";              ! Add segment to the limit line.
200  OUTPUT 718;"SEDI 1;";            ! Edit the first segment.
210  OUTPUT 718;"LIMF 250MHZ;";       ! Set frequency of the segment.
220  OUTPUT 718;"LIMU -35DBM;";       ! Provide the upper amplitude.
230  OUTPUT 718;"LIML -80DBM;";       ! Enter the lower amplitude.
240  OUTPUT 718;"LIMTFL;";            ! Set type of segment to flat.
250  OUTPUT 718;"SDON;";              ! Enter parameters into table.
260  OUTPUT 718;"SADD;SEDI 2;";       ! Add then edit second segment.
270  OUTPUT 718;"LIMF 290MHZ;";       ! Set frequency of segment 2.
280  OUTPUT 718;"LIMU-35DBM;LIML-80DBM;";
                                       ! Set amplitude data for segment.
290  OUTPUT 718;"LIMTSL;";            ! Set segment type to slope.
300  OUTPUT 718;"SDON;";              ! Enter parameters into table.
310  OUTPUT 718;"SADD;SEDI 3;";
320  OUTPUT 718;"LIMF 296.5MHZ;LIMU-5DBM;LIML-80DBM;LIMTSL;";
330  OUTPUT 718;"SDON;";
340  OUTPUT 718;"SADD;SEDI 4;";
350  OUTPUT 718;"LIMF 300MHZ;LIMU-5DBM;LIML-25DBM;LIMTSL;";
360  OUTPUT 718;"SDON;";
370  OUTPUT 718;"SADD;SEDI 5;";
380  OUTPUT 718;"LIMF 303.5MHZ;LIMU-5DBM;LIML-80DBM;LIMTSL;";
390  OUTPUT 718;"SDON;";
400  OUTPUT 718;"SADD;SEDI 6;";
410  OUTPUT 718;"LIMF 304.5MHZ;LIMU-5DBM;LIML-80DBM;LIMTSL;";
420  OUTPUT 718;"SDON;";
430  OUTPUT 718;"SADD;SEDI 7;";
440  OUTPUT 718;"LIMF 310MHZ;LIMU-35DBM;LIML-80DBM;LIMTFL;";
450  OUTPUT 718;"SDON;";
460  OUTPUT 718;"SADD;SEDI 8;";
470  OUTPUT 718;"LIMF 350MHZ;LIMU-35DBM;LIML-80DBM;LIMTFL;";
480  OUTPUT 718;"SDON;";
490  OUTPUT 718;"EDITDONE;";          ! End of limit-line definition.
491  !
492  ! Set up spectrum analyzer to display the limit line.
493  !
494  OUTPUT 718;"CF 300MHZ;SP 50MHZ;RB 2MHZ;VB 30KHZ;TS;";
500  END

```

SDEL

Delete Limit-Line Segment

Syntax



xf270a

Description This command deletes the limit-line segment specified with the SEDI command.

SDON

Limit-Line Segment Done

Syntax



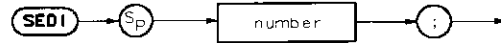
x1271a

Description This command is used to terminate the SEDI command.

SDON

Program Example

```
130 ! This program demonstrates the use of the SDON command
140 ! to generate a limit line:
150 !
160 OUTPUT 718;"LIMPURGE;";          ! Purge current limit line.
170 OUTPUT 718;"EDITLIML;";        ! Begin edit of limit line.
180 OUTPUT 718;"LIMIREL OFF;";     ! Make sure it's an absolute line.
190 OUTPUT 718;"SADD;";           ! Add segment to the limit line.
200 OUTPUT 718;"SEDI 1;";         ! Edit the first segment.
210 OUTPUT 718;"LIMF 250MHZ;";    ! Set frequency of the segment.
220 OUTPUT 718;"LIMU -35DBM;";    ! Provide the upper amplitude.
230 OUTPUT 718;"LIML -80DBM;";    ! Enter the lower amplitude.
240 OUTPUT 718;"LIMTFL;";        ! Set type of segment to flat.
250 OUTPUT 718;"SDON;";          ! Enter parameters into table.
260 OUTPUT 718;"SADD;SEDI 2;";    ! Add then edit second segment.
270 OUTPUT 718;"LIMF 290MHZ;";    ! Set frequency of segment 2.
280 OUTPUT 718;"LIMU-35DBM;LIML-80DBM;";
                                   ! Set amplitude data for segment.
290 OUTPUT 718;"LIMTSL;";        ! Set segment type to slope.
300 OUTPUT 718;"SDON;";          ! Enter parameters into table.
310 OUTPUT 718;"SADD;SEDI 3;";
320 OUTPUT 718;"LIMF 296.5MHZ;LIMU-5DBM;LIML-80DBM;LIMTSL;";
330 OUTPUT 718;"SDON;";
340 OUTPUT 718;"SADD;SEDI 4;";
350 OUTPUT 718;"LIMF 300MHZ;LIMU-5DBM;LIML-25DBM;LIMTSL;";
360 OUTPUT 718;"SDON;";
370 OUTPUT 718;"SADD;SEDI 5;";
380 OUTPUT 718;"LIMF 303.5MHZ;LIMU-5DBM;LIML-80DBM;LIMTSL;";
390 OUTPUT 718;"SDON;";
400 OUTPUT 718;"SADD;SEDI 6;";
410 OUTPUT 718;"LIMF 304.5MHZ;LIMU-5DBM;LIML-80DBM;LIMTSL;";
420 OUTPUT 718;"SDON;";
430 OUTPUT 718;"SADD;SEDI 7;";
440 OUTPUT 718;"LIMF 310MHZ;LIMU-35DBM;LIML-80DBM;LIMTFL;";
450 OUTPUT 718;"SDON;";
460 OUTPUT 718;"SADD;SEDI 8;";
470 OUTPUT 718;"LIMF 350MHZ;LIMU-35DBM;LIML-80DBM;LIMTFL;";
480 OUTPUT 718;"SDON;";
490 OUTPUT 718;"EDITDONE;";      ! End of limit-line definition.
491 !
492                               ! Set up to display the limit line.
493 !
494 OUTPUT 718;"CF 300MHZ;SP 50MHZ;RB 2MHZ;VB 30KHZ;TS;";
500 END
```

SEDI**Edit Limit-Line Segment****Syntax**

x'272o

Description Activates the limit-line segment you identify by its segment number in the limit-line table.

Related commands: LIMU, LIML, LIMM, LIMD, SDEL, LIMTFL, and LIMTSL to modify the limit-line parameters.

SEDI

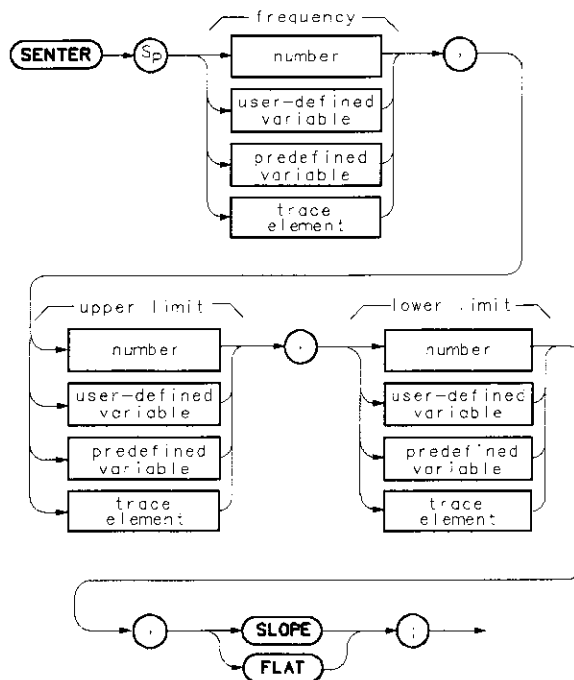
Program Example

```
130 ! This program demonstrates the use of the SEDI command
140 ! to generate a limit line:
150 !
160 OUTPUT 718;"LIMPURGE;";           ! Purge current limit line.
170 OUTPUT 718;"EDITLIML;";          ! Begin edit of limit line.
180 OUTPUT 718;"LIMIREL OFF;";       ! Make sure it's an absolute line.
190 OUTPUT 718;"SADD;";              ! Add segment to the limit line.
200 OUTPUT 718;"SEDI 1;";            ! Edit the first segment.
210 OUTPUT 718;"LIMF 250MHZ;";       ! Set frequency of the segment.
220 OUTPUT 718;"LIMU -35DBM;";       ! Provide the upper amplitude.
230 OUTPUT 718;"LIML -80DBM;";       ! Enter the lower amplitude.
240 OUTPUT 718;"LIMTFL;";            ! Set type of segment to flat.
250 OUTPUT 718;"SDON;";              ! Enter parameters into table.
260 OUTPUT 718;"SADD;SEDI 2;";       ! Add then edit second segment.
270 OUTPUT 718;"LIMF 290MHZ;";       ! Set frequency of segment 2.
280 OUTPUT 718;"LIMU-35DBM;LIML-80DBM;";
                                     ! Set amplitude data for segment.
290 OUTPUT 718;"LIMTSL;";            ! Set segment type to slope.
300 OUTPUT 718;"SDON;";              ! Enter parameters into table.
310 OUTPUT 718;"SADD;SEDI 3;";
320 OUTPUT 718;"LIMF 296.5MHZ;LIMU-5DBM;LIML-80DBM;LIMTSL;";
330 OUTPUT 718;"SDON;";
340 OUTPUT 718;"SADD;SEDI 4;";
350 OUTPUT 718;"LIMF 300MHZ;LIMU-5DBM;LIML-25DBM;LIMTSL;";
360 OUTPUT 718;"SDON;";
370 OUTPUT 718;"SADD;SEDI 5;";
380 OUTPUT 718;"LIMF 303.5MHZ;LIMU-5DBM;LIML-80DBM;LIMTSL;";
390 OUTPUT 718;"SDON;";
400 OUTPUT 718;"SADD;SEDI 6;";
410 OUTPUT 718;"LIMF 304.5MHZ;LIMU-5DBM;LIML-80DBM;LIMTSL;";
420 OUTPUT 718;"SDON;";
430 OUTPUT 718;"SADD;SEDI 7;";
440 OUTPUT 718;"LIMF 310MHZ;LIMU-35DBM;LIML-80DBM;LIMTFL;";
450 OUTPUT 718;"SDON;";
460 OUTPUT 718;"SADD;SEDI 8;";
470 OUTPUT 718;"LIMF 350MHZ;LIMU-35DBM;LIML-80DBM;LIMTFL;";
480 OUTPUT 718;"SDON;";
490 OUTPUT 718;"EDITDONE;";          ! End of limit-line definition.
491 !
492                                     ! Set up to display the limit line.
493 !
494 OUTPUT 718;"CF 300MHZ;SP 50MHZ;RB 2MHZ;VB 30KHZ;TS;";
500 END
```

SENDER

Segment Enter

Syntax



x1273a

Description This command can be used to create a complete limit-line segment within a DLP. Define which limit-line segment to create using the SEDI command.

Related command: SADD.

SETDATE

Set Date

Syntax



x12746

Description Use this command to set the date.

Parameter Where integer = MMDDYY

Program Example

```
10 ! The following examples show the use of the SETDATE command<M>.
20 ! The date can be specified directly or generated from another
30 ! source such as the HP BASIC TIMEDATE command.
40 !
50 OUTPUT 718;"TIMEDATE ON;"; ! Turn on the time and date display
60 ! so that date changes can be observed.
70 DISP "Setting date to 9/12/88"
80 OUTPUT 718;"SETDATE 091288;"; ! Set the date to 9/12/88 directly.
90 !
100 WAIT 5 ! Note the date on the display.
110 !
120 DISP "Setting date to Series 200/300 Internal clock"
130 Day_mon_year$=DATE$(TIMEDATE) ! \
140 Day$=Day_mon_year$[1;2] ! \
150 IF VAL(Day$) <10 THEN Day$[1;1]="0" ! Fill in leading "0".
160 Mon$=Day_mon_year$[4;3] ! Convert date from TIMEDATE.
170 Year$=Day_mon_year$[10;2] ! / format to spectrum analyzer
180 GOSUB Translate_month ! / format.
190 !
200 OUTPUT 718;"SETDATE "&Month$&Day$&Year$&";";
! Set the new date.
210 STOP
220 !
230 Translate_month: !
240 ! Converts the month from alpha characters to 'numerics'. The numbers
250 ! are left as strings so that they can be concatenated later.
260 !
270 SELECT Mon$
280 CASE "Jan"
290 Month$="01"
300 CASE "Feb"
310 Month$="02"
320 CASE "Mar"
330 Month$="03"
340 CASE "Apr"
350 Month$="04"
360 CASE "May"
```

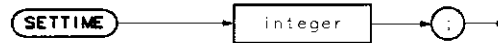
SETDATE

```
370     Month$="05"  
380 CASE "Jun"  
390     Month$="06"  
400 CASE "Jul"  
410     Month$="07"  
420 CASE "Aug"  
430     Month$="08"  
440 CASE "Sep"  
450     Month$="09"  
460 CASE "Oct"  
470     Month$="10"  
480 CASE "Nov"  
490     Month$="11"  
500 CASE "Dec"  
510     Month$="12"  
520 END SELECT  
530 RETURN  
540 END
```

SETTIME

Set Time

Syntax



x'275a

Description Use this command to set the time. (Enter the 24-hour time in this format: HHMMSS.)

Parameter Where integer = HHMMSS

Program Example

```
10 ! The following example shows the use of the SETTIME command. The clock
20 ! is set using 24-hour notation. When time is displayed on the spectrum
30 ! analyzer screen, it appears in the current time mode, that is, the
40 ! 12- or 24-hour mode.
50 !
60 OUTPUT 718;"TIMEDATE ON;" ! Turn on the time and date display to
70 ! observe the results of the next command.
80 OUTPUT 718;"SETTIME 183000;" ! The clock is set using 24-hour notation.
90 !
100 END
```

SHOWMENU

Show the Menu

Syntax



x f276 a

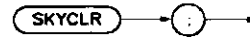
Description

This command displays labels in the softkey area on the display. The user specifies the softkey labels using the SKYCLR and SKYDEF commands. This command has no meaning if the spectrum analyzer is in remote mode. It must be used in a user-defined function, or DLP. If used from an external controller, an error results.

SHOWMENU

Program Example

```
10      ! The following example shows the use of the SHOWMENU command. The
20      ! example creates and displays a softkey which can be used to execute
30      ! a DLP (downloadable program).
40      !
50      ! Note: Since SHOWMENU is not executable over HP-IB, the function
60      ! KEYS needs to be executed from the front panel of the analyzer
        ! after being loaded. Press
70      ! {LCL} [MODULE] {AUTOEXEC MENU} {EDIT AUTOEXEC} {CHOOSE DLP}.
80      ! Scroll knob to 'KEYS' and press {EXECUTE NOW}.
90      !
100     ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
110     OUTPUT @Sa;"FUNCDEF KEYS,$";      ! Logical start of function 'KEYS'.
120     OUTPUT @Sa;"SKYCLR;";            ! Clear softkeys.
130     OUTPUT @Sa;"SKYDEF 1,BOX,%DRAW BOX%";
140                                     ! Define softkey 1 to execute
150                                     ! function BOX. Label key with
160     OUTPUT @Sa;"SHOWMENU;";          ! Display the SKYDEF'ed label.
170     OUTPUT @Sa;"$";                  ! Logical end of function.
180     !
190     OUTPUT @Sa;"FUNCDEF BOX,@;";      ! Logical start of function 'BOX'.
200     OUTPUT @Sa;" IP;";                ! Instrument preset.
210     OUTPUT @Sa;" CLRDSP;";           ! Clear the analyzer screen.
220     !
230     OUTPUT @Sa;" PU;PA 300,300;";     ! Move pen to starting point.
240     OUTPUT @Sa;" PD;PR 240,0;";      ! \
250     OUTPUT @Sa;" PR 0,240;";         ! \ Draw
260     OUTPUT @Sa;" PR -240,0;";        ! / rectangle.
270     OUTPUT @Sa;" PR 0,-240;";       ! /
280     OUTPUT @Sa;"@";                  ! Logical end of function.
290     !
300     ASSIGN @Sa TO *                    ! Close I/O path.
310     END
```

SKYCLR**Clear User Softkeys****Syntax**

x1277a

Description Use this command to clear all user-definable softkeys set up in DLPs. The command SKYDEF can then be used to define other DLP-related softkey labels.

Related commands: SKYDEF and SHOWMENU.

SKYCLR

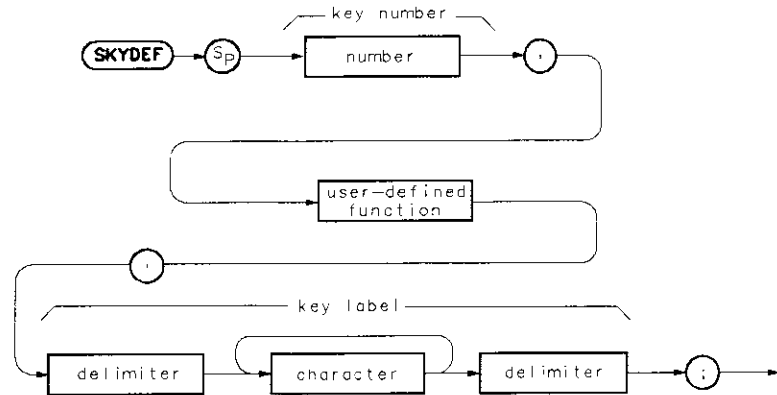
Program Example

```
10   ! The following example shows the use of the SKYCLR command.  The
20   ! example creates and displays a softkey which can be used to execute
30   ! a DLP (downloadable program).
40   !
50   ! Note: Since SHOWMENU is not executable over the HP-IB, the function
60   ! KEYS needs to be executed from the front panel of the analyzer
      after it is loaded.  Press
70   ! {LCL} [MODULE] {AUTOEXEC MENU} {EDIT AUTOEXEC} {CHOOSE DLP}.
80   ! Scroll knob to 'KEYS' and press {EXECUTE NOW}.
90   !
100  ASSIGN @Sa TO 718           ! Assign I/O path to address 718.
110  OUTPUT @Sa;"FUNCDEF KEYS,$"; ! Logical start of function 'KEYS'.
120  OUTPUT @Sa;"SKYCLR;";       ! Clear softkeys.
130  OUTPUT @Sa;"SKYDEF 1,BOX,%DRAW BOX%;";
      ! Define softkey 1 to execute
140      ! function BOX.  Label key with
150      ! DRAW BOX.
160  OUTPUT @Sa;"SHOWMENU;";     ! Display the SKYDEF'ed label.
170  OUTPUT @Sa;"$";             ! Logical end of function.
180  !
190  OUTPUT @Sa;"FUNCDEF BOX,@;"; ! Logical start of function 'BOX'.
200  OUTPUT @Sa;" IP;";          ! Instrument preset.
210  OUTPUT @Sa;" CLRDSP;";     ! Clear the analyzer screen.
220  !
230  OUTPUT @Sa;" PU;PA 300,300;"; ! Move pen to starting point.
240  OUTPUT @Sa;" PD;PR 240,0;"; ! \
250  OUTPUT @Sa;" PR 0,240;";    ! \ Draw
260  OUTPUT @Sa;" PR -240,0;";   ! / rectangle.
270  OUTPUT @Sa;" PR 0,-240;";  ! /
280  OUTPUT @Sa;"@";            ! Logical end of function.
290  !
300  ASSIGN @Sa TO *           ! Close I/O path.
310  END
```

SKYDEF

Define a User Keys Menu Softkey

Syntax



x12780

Description In a DLP, use this command to attach a program to a softkey. SKYDEF must be followed in the DLP with SHOWMENU to display the softkeys for the DLP operation. This then keeps the softkey labels within the DLP; they are activated only when the DLP is operating.

Related commands: SKYCLR and SHOWMENU.

SKYDEF

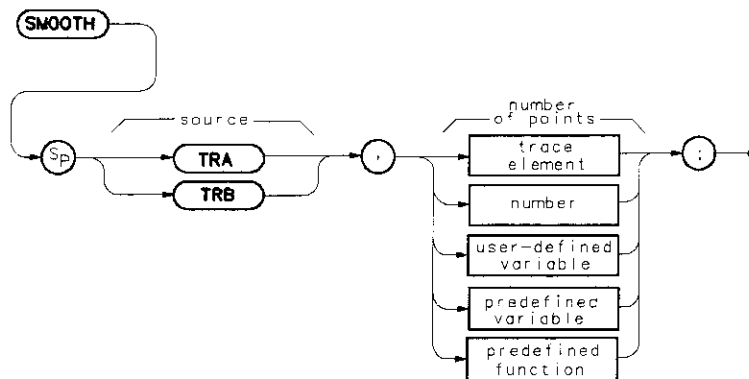
Program Example

```
10      ! The following example shows the use of the SKYDEF command.  The
20      ! example creates and displays a softkey which can be used to execute
30      ! a DLP (downloadable program).
40      !
50      ! Note: Since SHOWMENU is not executable over HP-IB, the function
60      ! KEYS needs to be executed from the front panel of the analyzer
        after it is loaded. Press
70      ! {LCL} [MODULE] {AUTOEXEC MENU} {EDIT AUTOEXEC} {CHOOSE DLP}.
80      ! Scroll knob to 'KEYS' and press {EXECUTE NOW}.
90      !
100     ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
110     OUTPUT @Sa;"FUNCDEF KEYS,$";    ! Logical start of function 'KEYS'.
120     OUTPUT @Sa;"SKYCLR;";          ! Clear softkeys.
130     OUTPUT @Sa;"SKYDEF 1,BOX,%DRAW BOX%";
                                        ! Define softkey 1 to execute
140                                         ! function BOX. Label key with
150                                         ! DRAW BOX.
160     OUTPUT @Sa;"SHOWMENU;";        ! Display the SKYDEF'ed label.
170     OUTPUT @Sa;"$";                ! Logical end of function.
180     !
190     OUTPUT @Sa;"FUNCDEF BOX,@;";    ! Logical start of function 'BOX'.
200     OUTPUT @Sa;" IP;";             ! Instrument preset.
210     OUTPUT @Sa;" CLRDSP;";        ! Clear the analyzer screen.
220     !
230     OUTPUT @Sa;" PU;PA 300,300;";  ! Move pen to starting point.
240     OUTPUT @Sa;" PD;PR 240,0;";    ! \
250     OUTPUT @Sa;" PR 0,240;";       ! \ Draw
260     OUTPUT @Sa;" PR -240,0;";      ! / rectangle.
270     OUTPUT @Sa;" PR 0,-240;";     ! /
280     OUTPUT @Sa;"@";                ! Logical end of function.
290     !
300     ASSIGN @Sa TO *                 ! Close I/O path.
310     END
```

SMOOTH

Smooth Trace

Syntax



x1279a

Description

This command smooths the trace according to the number of points specified as the running average. Point values are replaced with the average value (in measurement units) of a given quantity of points centered about it. As the number of points increases, smoothing increases, but resolution decreases.

This function provides spatial video-averaging as compared with time-based video-averaging from an HP 856X VAVG command. Using averaging for the point value, any high-frequency noise or signals are attenuated without affecting the corresponding low-frequency signals. The end result is similar to reducing the video bandwidth, but without changing the sweep time. The frequency resolution, however, is reduced.

Parameters

number of points = an odd integer from 3 to 601. If an even number is specified, it will be converted to an odd integer having the next greater value.

SMOOTH

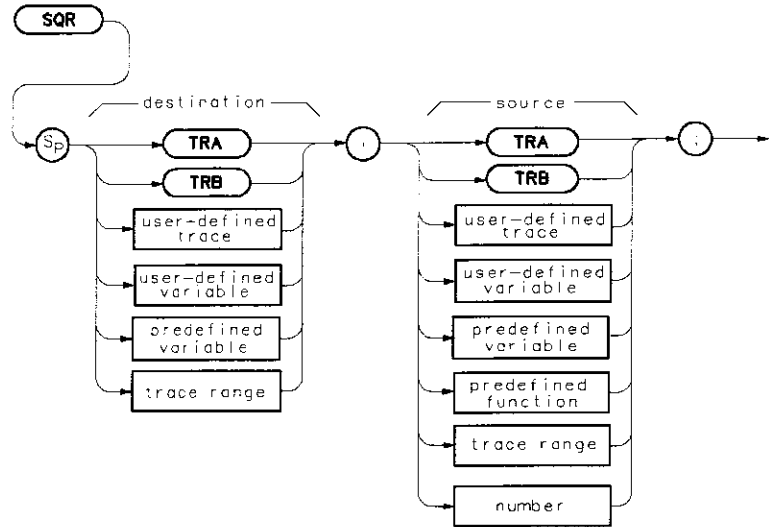
Program Example

```
10    ! The following example shows the use of the SMOOTH command. The
20    ! instructions within the FUNCDEF '@' delimiters form a structure
30    ! called a DLP (downloadable program).
40    !
50    ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
60    !
70    OUTPUT @Sa;"FUNCDEF S_MOOTH_EX,@"; ! Logical start of the DLP.
80    OUTPUT @Sa;"IP;";                ! Instrument preset.
90    !
100   OUTPUT @Sa;"TS; VIEW TRA;";      ! Take a sweep and put trace into
110                                       ! View mode.
120   !
130   OUTPUT @Sa;"SMOOTH TRA,5;";      ! Smooth the trace with a five-point
140                                       ! running average.
150   !
160   OUTPUT @Sa;"@";                  ! Logical end of the DLP.
170   !
180   OUTPUT @Sa;"S_MOOTH_EX;";        ! Have the analyzer execute the
190                                       ! DLP S_MOOTH_EX
200   ASSIGN @Sa TO *                  ! Close I/O path.
210   END
```

SQR

Square Root

Syntax



x1280a

Description Computes the square root of the source and sends the result to the destination. If the source is negative, an error message is generated, and the absolute value of the source is returned.

Program Example

```

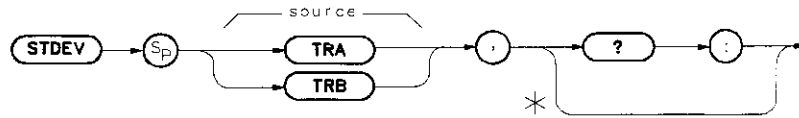
10      ! The following example shows the use of the SQR command.
20      !
30      ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
40      !
50      OUTPUT @Sa;"IP;CLRDSP;";        ! Instrument preset;
60      ! Clear CRT for text display.
70      OUTPUT @Sa;"VARDEF NUMBER,123;"; ! \ Create variable and
80      OUTPUT @Sa;"VARDEF ROOT,0;";    ! / initialize.
90      OUTPUT @Sa;"SQR ROOT,NUMBER;";  ! Take the square root.
100     !
110     OUTPUT @Sa;"PU;PA 10,500;";     ! Move "pen" to starting point.
120     OUTPUT @Sa;"TEXT %The Square Root of %;";
                                           ! Write text.
130     OUTPUT @Sa;"DSPLY NUMBER,7,5;"; ! Write value.
140     OUTPUT @Sa;"TEXT % = %;";       ! Write text.
150     OUTPUT @Sa;"DSPLY ROOT,7,5;";   ! Write value.
160     !
170     ASSIGN @Sa TO *                  ! Close I/O path.
180     END

```

STDEV

Standard Deviation of Trace Amplitudes

Syntax



xf289a

Note

This bypass command path is only legal if you use STDEV as a predefined function. It must reside within a compatible-function operation.

Description

Return the standard deviation of the trace amplitude in measurement units.

Prerequisite Commands: TS when using TRA, TRB, or trace range for source.

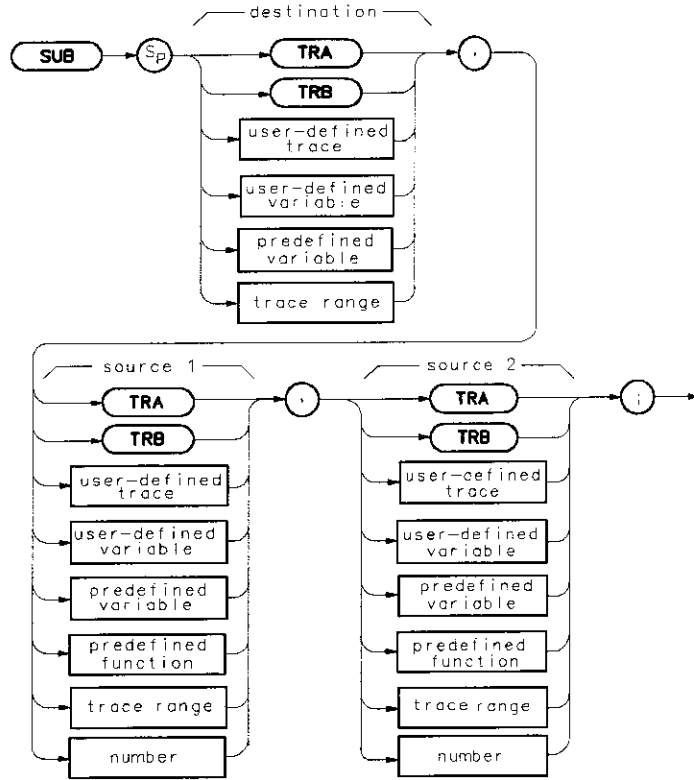
Program Example

```
10 ! The following example shows the use of the STDEV function to return
20 ! the standard deviation of the trace to the controller. The example
30 ! uses an input signal of 300 MHz such as the CAL OUTPUT.
40 !
50 INTEGER Mean_value ! Define an INTEGER variable.
60 ASSIGN @Sa TO 718 ! Assign I/O path to address 718.
70 !
80 OUTPUT @Sa;"IP;CF 300MHZ;SP 1MHZ;TS;";
90 OUTPUT @Sa;"MEAN TRA,?"; ! Set up the analyzer.
100 ENTER @Sa;Mean_value ! Determine the mean of trace A.
110 ! Read value from analyzer.
110 PRINT "MEAN of trace A in measurement units = ";Mean_value
120 OUTPUT @Sa;"STDEV TRA,?";
130 ENTER @Sa;Stdev_value
140 PRINT "STANDARD DEVIATION of trace A in measurement units = ";Stdev_value
150 !
160 ASSIGN @Sa TO * ! Close I/O path.
170 !
180 END
```


SUB

Subtract

Syntax



x1281a

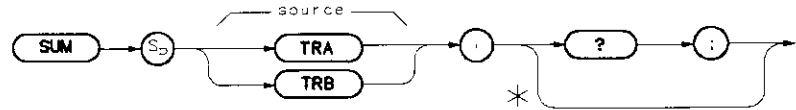
Description

Subtract the value of source 1 from source 2 and send the results to the destination.

SUB

Program Example

```
10   ! The following example shows the use of the SUB command.  The example
20   ! uses an input signal of 300 MHz, such as the CAL OUTPUT signal, and
30   ! looks at its harmonics. An input signal is not necessary to the
40   ! function of this example, but acts as a visual aid.
50   !
60   ASSIGN @Sa TO 718           ! Assign I/O path to address 718.
70   !
80   OUTPUT @Sa;"FUNCDEF SUB_EX,@";   ! Logical start of the DLP function.
90   OUTPUT @Sa;"VARDEF FREQ,3E9";    ! Create variable and initialize to 3 GHz.
100  OUTPUT @Sa;"IP;SP 1MHZ";        ! Instrument preset; set span.
110  !
120  OUTPUT @Sa;"REPEAT;";           ! Begin loop.
130  OUTPUT @Sa;"  MOV CF,FREQ;";    ! Set center frequency.
140  OUTPUT @Sa;"  TS;";            ! Take a sweep to update display.
150  OUTPUT @Sa;"  SUB FREQ,FREQ,300E6;"; ! Decrease FREQ by 300 MHz.
160  OUTPUT @Sa;"UNTIL FREQ,LT,300E6;"; ! End of loop.
170  OUTPUT @Sa;"@";               ! Logical end of DLP function.
180  !
190  OUTPUT @Sa;"SUB_EX;";          ! Execute function.
200  !
210  ASSIGN @Sa TO *              ! Close I/O path.
220  END
```

SUM
Sum of Trace Amplitudes**Syntax**

xf282a

Note

This bypass command path is only legal if you use SUM as a predefined function. It must reside within a compatible-function operation.

Description Compute the sum of the given trace.

Program Example

```

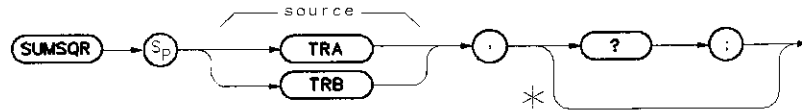
10      ! The following example shows the use of the SUM command.  The example
20      ! uses an input signal of 300 MHz such as the CAL OUTPUT.
30      !
40      ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
50      !
60      OUTPUT @Sa;"IP;";                ! Instrument preset.
70      OUTPUT @Sa;"CF 300MZ; SP 1MZ;";  ! \
80      OUTPUT @Sa;"SNGLS;TS;";         ! Set up analyzer.
90      OUTPUT @Sa;"MKPK HI;MKRL;TS;HD;"; ! /
100     !
110     OUTPUT @Sa;"SUM TRA,?;";         ! Returns sum in measurement units.
120     ENTER @Sa;Sum_trace              ! Bring result into computer.
130     PRINT "The sum of all the trace points equals ";Sum_trace
140     !
150     ASSIGN @Sa TO *                  ! Close I/O path.
160     END

```

SUMSQR

Sum of Squared Trace Amplitudes

Syntax



xf283e

Note

This bypass command path is only legal if you use SUMSQR as a predefined function. It must reside within a compatible-function operation.

Description Squares the amplitude of each trace element, and returns the sum of the squares to the controller in measurement units.

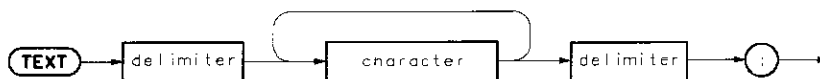
Program Example

```
10 ! The following example shows the use of the SUMSQR command. The
20 ! example uses an input signal of 300 MHz such as the CAL OUTPUT.
30 !
40 ASSIGN @Sa TO 718 ! Assign I/O path to address 718.
50 !
60 OUTPUT @Sa;"IP;"; ! Instrument preset.
70 OUTPUT @Sa;"CF 300MZ; SP 1MZ;"; ! \
80 OUTPUT @Sa;"RB 30KHZ;"; ! Set up the analyzer.
90 OUTPUT @Sa;"TS;MKPK HI;MKRL;TS;HD;";! /
100 !
110 OUTPUT @Sa;"SUMSQR TRA,?;"; ! Query the analyzer.
120 ENTER @Sa;Sum_sqr ! Get the value from the analyzer.
130 PRINT "The Sum of the Squares of the trace points = ";Sum_sqr;
140 PRINT "measurement units."
150 !
160 ASSIGN @Sa TO * ! Close I/O path.
170 END
```

TEXT

Text

Syntax



xf284a

Description This command is used to display the user-defined text on the spectrum analyzer screen at the current graphics pen location. Related commands are PU, PA, and PD.

Program Example

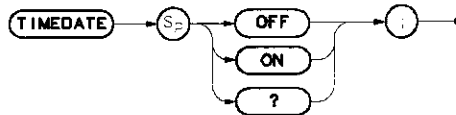
```

10 ! The following example shows the use of the REPEAT/UNTIL commands
20 ! in a FOR/NEXT loop context. This is an immediate-execute DLP
30 ! (downloadable program). This is another way of nesting REPEATs.
40 !
50 ASSIGN @Sa TO 718 ! Assign I/O path to address 718.
60 !
70 OUTPUT @Sa;"IP;"; ! Instrument preset.
80 OUTPUT @Sa;"VARDEF OUTER_LP,0;"; ! Create variable and initialize to 0.
90 OUTPUT @Sa;"VARDEF TOTAL,0;"; ! Create variable and initialize to 0.
100 !
110 OUTPUT @Sa;"REPEAT;"; ! Begin outer loop.
120 OUTPUT @Sa;" VARDEF INNER_LP,0;"; ! Create and initialize inner loop counter.
130 OUTPUT @Sa;" REPEAT;"; ! Begin inner loop.
140 OUTPUT @Sa;" ADD TOTAL,TOTAL,1;"; ! Increment running total.
150 OUTPUT @Sa;" ADD INNER_LP,INNER_LP,1;";
! Increment inner loop counter.
160 OUTPUT @Sa;" UNTIL INNER_LP,EQ,3;"; ! End of inner loop.
170 OUTPUT @Sa;" ADD OUTER_LP,OUTER_LP,1;";
! Increment outer loop counter.
180 OUTPUT @Sa;"UNTIL OUTER_LP,EQ,4;"; ! End of outer loop.
190 !
200 OUTPUT @Sa;"CLR DSP;";
210 OUTPUT @Sa;"PU;PA 10,500;TEXT %INNER LOOP COUNTER=%;";
220 OUTPUT @Sa;"DSPLY INNER_LP,5,2;";
230 OUTPUT @Sa;"PU;PA 10,400;TEXT %OUTER LOOP COUNTER=%;";
240 OUTPUT @Sa;"DSPLY OUTER_LP,5,2;";
250 OUTPUT @Sa;"PU;PA 10,300;TEXT %TOTAL ITERATIONS=%;";
260 OUTPUT @Sa;"DSPLY TOTAL,5,2;";
270 !
280 ASSIGN @Sa TO * ! Close I/O path.
290 END
  
```

TIMEDATE

Time Date

Syntax



x1285a

Description Use this command to turn the time and date display on or off. The default setting is OFF.

Note The time/date display must be turned off in order to display HP-IB errors properly on the spectrum analyzer display.

Query Response The time/date setting of the module is returned in the MMDDYY format.

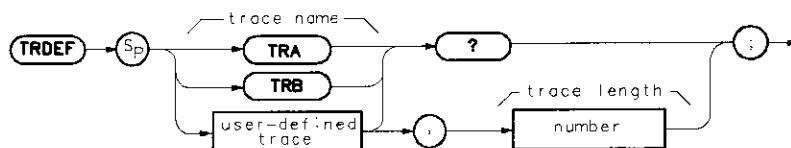
Program Example

```
10 ! The following example shows the use of the TIMEDATE command.
20 !
30 ASSIGN @Sa TO 718 ! Assign I/O path to address 718.
40 OUTPUT @Sa;"IP;"; ! Instrument preset.
50 DISP "Activate 856X Time and Date Display"
60 OUTPUT @Sa;"TIMEDATE ON;";
70 !
80 WAIT 5 ! Note that time and date are displayed.
90 !
100 DISP "De-activate 8562 Time and Date Display"
110 OUTPUT @Sa;"TIMEDATE OFF;";
120 !
130 ASSIGN @Sa TO * ! Close I/O path.
140 END
```

TRDEF

Trace Definition

Syntax



xf286a

Description Defines the name and length of a user-defined trace. Any number of points from 0 to 65,000 may be used for a trace length. Changing the length of predefined traces results in an error. The length of a predefined trace can be queried.

User memory is required to execute the TRDEF command.

Parameters <trace length>::= integer from 1 to 65,000
(there are 601 points in a predefined trace)

Restrictions: Subject to available memory. The following commands consume memory: ONEOS, ARRAYDEF, TRDEF, KEYDEF, LCLVAR, VARDEF REPEAT/UNTIL, and IF/THEN.

Related Command: DISPOSE

Querying a User-Defined Trace

For user-defined trace data transfer to and from an external controller, only M format is supported. A, B, I, and P block-data field formats are not currently supported.

User-Defined Trace Query Response

Response Syntax Notation: <trace length>

Parameters Minimum Range: 1
Maximum Range: 65000

TRDEF

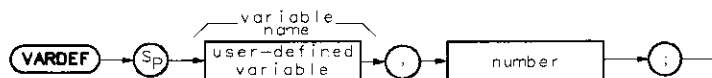
Program Example

```
10   ! The following example shows the use of the TRDEF command. The example
11   ! creates a sine wave in the computer and sends it to trace A of the
12   ! analyzer. It is then moved to the user-defined trace where it can be
13   ! manipulated to fit the user's needs.
14   !
15   !
16   ! Assign I/O path to address 718.
17   ! Create an array in the computer.
18   ! Angles measured in radians.
19   ! Graphics Init (computer).
20   ! Graphics On (computer).
21   ! Set up the calculator screen to
22   ! have the same number of points as
23   ! the spectrum analyzer.
24   ! Move the pen (computer).
25   ! Create the array in the computer.
26   ! Increment array pointer.
27   ! 120 gives +/- 2 divs of amplitude.
28   ! 300 offsets to mid-screen
29   ! Draw sine wave on computer screen.
30   ! Load the array.
31   ! Set up the analyzer to receive the trace array from the computer.
32   ! Instrument preset. Single sweep
33   ! so that the trace is not over-
34   ! written by another sweep.
35   ! View trace A. Set to 'A' format.
36   ! Send the trace.
37   ! Create a user-defined trace that
38   ! is the same size as a default
39   ! screen trace.
40   ! Copy trace A to the user-defined
41   ! trace.
42   ! Add 60 to each point in SINE.
43   ! Copy the modified SINE trace to
44   ! trace B.
45   ! View both traces.
46   ! Note that 60 offsets the sine wave one division from the original trace.
47   !
48   ! Close I/O path.
49   END
```


VARDEF

Variable Definition

Syntax



*F287a

Description Defines the name of a user-defined variable and assigns its initial value. If a command mnemonic and the label of the user-defined variable match, an error results.

Note In Mass Memory Modules with firmware datecode 910116 and later, the value of a user-defined variable can be modified using the secondary keyword EP. Refer to “Using EP to Modify User-Defined Variables (firmware revision 910116 and later)” in this chapter.

User memory is required to execute the VARDEF command. After using VARDEF to define a label, other commands may be used as described below.

Parameters <initial value> ::= <real>

Restrictions: Subject to available memory. The following commands consume memory: ONEOS, ARRAYDEF, TRDEF, KEYDEF, LCLVAR, VARDEF REPEAT/UNTIL, and IF/THEN.

Prerequisite Commands: VARDEF when using user-defined variable.

Related Command: DISPOSE

VARDEF

Program Example

```
10  ! The following example shows the use of the VARDEF command.  The
20  ! example uses an input signal of 300 MHz, such as the CAL OUTPUT signal
30  ! and looks at its harmonics.  An input signal is not necessary to the
40  ! function of this example, but acts as a visual aid.
50  !
60  ASSIGN @Sa TO 718          ! Assign I/O path to address 718.
70  !
80  OUTPUT @Sa;"FUNCDEF INCRFREQ,@;" ! Logical start of the DLP function.
90  OUTPUT @Sa;"VARDEF FREQ,300E6;" ! Create variable and initialize to
   ! 300 MHz.
100 OUTPUT @Sa;"IP;SP 1MHZ;" ; ! Instrument preset.
110 !
120 OUTPUT @Sa;"REPEAT;" ; ! Begin loop.
130 OUTPUT @Sa;"  MOV CF,FREQ;" ; ! Set center frequency.
140 OUTPUT @Sa;"  TS;" ; ! Take a sweep to update display.
150 OUTPUT @Sa;"  ADD FREQ,FREQ,300E6;" ;
   ! Increase FREQ by 300 MHz.
160 OUTPUT @Sa;"UNTIL FREQ,GT,3E9;" ; ! End of loop.
170 OUTPUT @Sa;"@" ; ! Logical end of DLP function.
180 !
190 !
200 OUTPUT @Sa;"  INCRFREQ;" ; ! Execute function.
210 !
220 ASSIGN @Sa TO *          ! Close I/O path.
230 END
```

Query Value of Current Variable

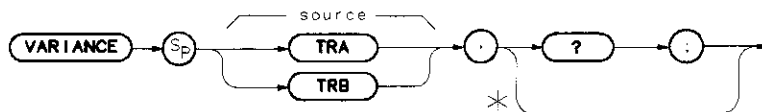
Variable value may be queried by executing the command as shown:

```
10 OUTPUT 718;"FREQ?;"
20 OUTPUT 718;FREQUENCY
```

VARIANCE

Variance

Syntax



xf285a

Note

This bypass command path is only legal if you use VARIANCE as a predefined function. It must reside within a compatible-function operation.

Description

The VARIANCE command returns the amplitude variations of the specified trace to the controller in measurement units.

Prerequisite Commands: TS when using TRA, TRB, or trace range for source.

Program Example

```

10      ! The following example shows the use of the VARIANCE command.
20      ! The example uses an input signal of 300 MHz such as the CAL OUTPUT.
30      !
40      ASSIGN @Sa TO 718                ! Assign I/O path to address 718.
50      !
60      OUTPUT @Sa;"IP;SNGLS;"          ! Instrument preset.
70      OUTPUT @Sa;"CF 300 MZ;SP 1KHZ;" ! \ Set up analyzer.
80      OUTPUT @Sa;"RB 300HZ; TS;"      ! /
90      !
100     OUTPUT @Sa;"VARIANCE TRA,?;"    ! Query the trace variance.
110     ENTER @Sa;Variance              ! Get the value from the analyzer.
120     PRINT "The variance of trace A is ";Variance
130     !
140     ASSIGN @Sa TO *                  ! Close I/O path.
150     END

```

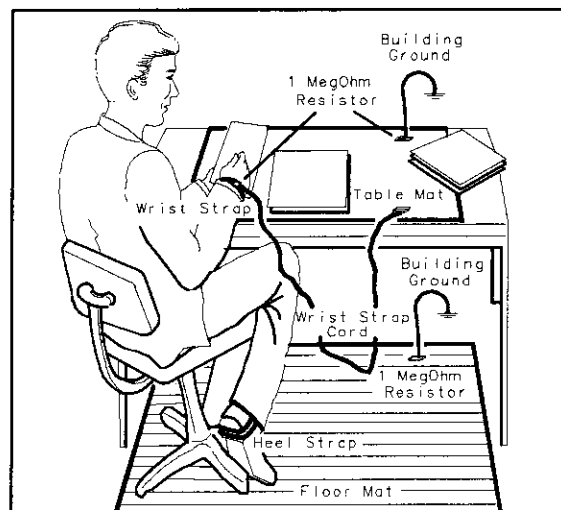

Service

Turning on the Module

The HP 85620A Mass Memory Module requires no specific performance verification tests nor adjustment procedures. If the module should fail to power up at turn-on, turn the spectrum analyzer off, then verify that the module is installed correctly. Refer to Chapter 1, "Installation," for module connection information. If the connection appears to be correct, contact one of the Hewlett-Packard Sales and Service Offices listed in Chapter 1, Table 1-7.

Electrostatic Discharge

Electrostatic discharge (ESD) can damage or destroy electronic components. All work performed on assemblies containing electronic components should be done **ONLY** at a static-safe workstation. See Figure 5-1. Static-safe accessories may be ordered from any Hewlett-Packard Sales and Service Office listed in Chapter 1, Table 1-7. Refer to Table 5-1 for a list of the part numbers for these accessories.



FORMAT46/DF239

Figure 5-1. Static-Safe Workstation

Static-Safe Accessories

Table 5-1. Static-Safe Accessories

HP Part Number	Description
9300-0797	Set includes: 3M static control mat 0.6 m × 1.2 m (2 ft × 4 ft) and 4.6 cm (15 ft) ground wire. (The wrist-strap and wrist-strap cord are not included. They must be ordered separately.)
9300-0980	Wrist-strap cord 1.5 m (5 ft)
9300-1383	Wrist-strap, color black, stainless steel, without cord, has four adjustable links and a 7 mm post-type connection.
9300-1169	ESD heel-strap (reusable 6 to 12 months).

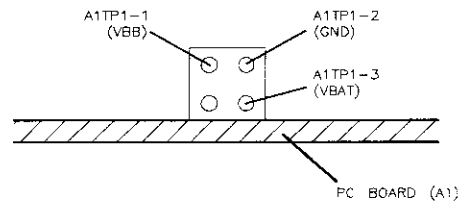
Returning Modules for Service

If you are returning a module to Hewlett-Packard for servicing, fill in and attach one of the blue service tags located at the end of this chapter. Please be as specific as possible about the nature of the problem. If you have recorded any error messages that appeared on the screen or have any other specific data on the performance of the module, please send a copy of this information along with the unit.

Back-Up Battery Voltage Test

Test the module back-up battery by connecting a DVM between A1TP1-3 and A1TP1-2 (ground). The dc voltage measurement should be about 2.8 Vdc ±0.5 Vdc. See Figure 5-2.

Test the memory card battery by connecting a DVM between A1TP1-1 and A1TP1-2 (ground). The dc voltage measurement should be about 2.7 Vdc ±0.5 Vdc. (Option T01 does not include memory card capability.) See Figure 5-2.



d1216a

Figure 5-2. A1TP1 Configuration

Replacement Procedures

Introduction Replacement procedures provide specific disassembly and assembly information. Refer to the appropriate replacement procedures for the assembly you are replacing.

- A1 Memory Board Replacement
- B1 Module Battery Replacement
- Memory Card Battery Replacement

A1 Memory Board Assembly Replacement

Caution

- The module and board assembly are static sensitive. Be sure to perform any disassembly at a static-safe workstation as illustrated in Figure 5-1.
 - Store or transport these items **ONLY** in static-shielding containers.
 - Personnel should be grounded with a resistor-isolated wrist strap before touching any connector pins or before removing any assembly from the module.
 - Be sure that all instruments are properly earth-grounded to prevent build up of static charge.
-

1. Remove the four module-assembly screws (1). See Figure 5-3 or Figure 5-4.
2. Open the module from the left-hand side (2) with the front-panel label facing you.
3. With the module laying flat, remove the brace (3) and the eight board-assembly screws (4). Lift the board assembly out of the module.
4. Replace the board assembly into the module and secure it with the eight board-assembly screws (4). Torque each one to three inch-pounds.
5. Replace the brace (3), then close the module halves.
6. Replace the four module screws (1) and torque each one to six inch-pounds.

B1 Module Battery Replacement

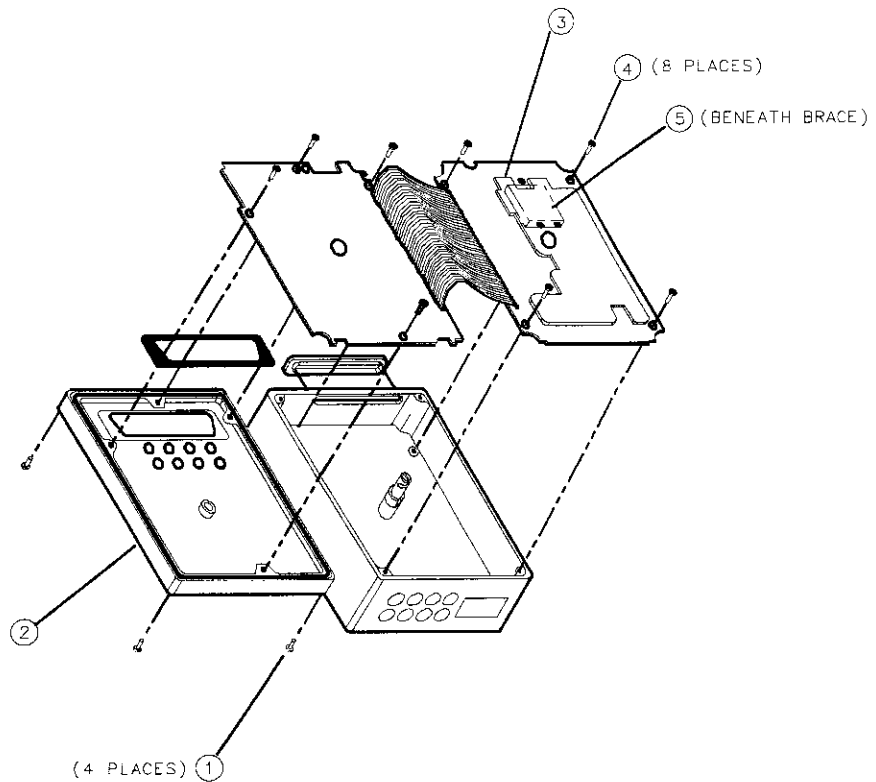
1. Refer to the A1 Memory Board Assembly Replacement procedure to remove the A1 assembly.
2. Carefully desolder the battery (5).* See Figure 5-3 or Figure 5-4.

Warning

The battery case becomes very hot during desoldering. Use care when handling it.

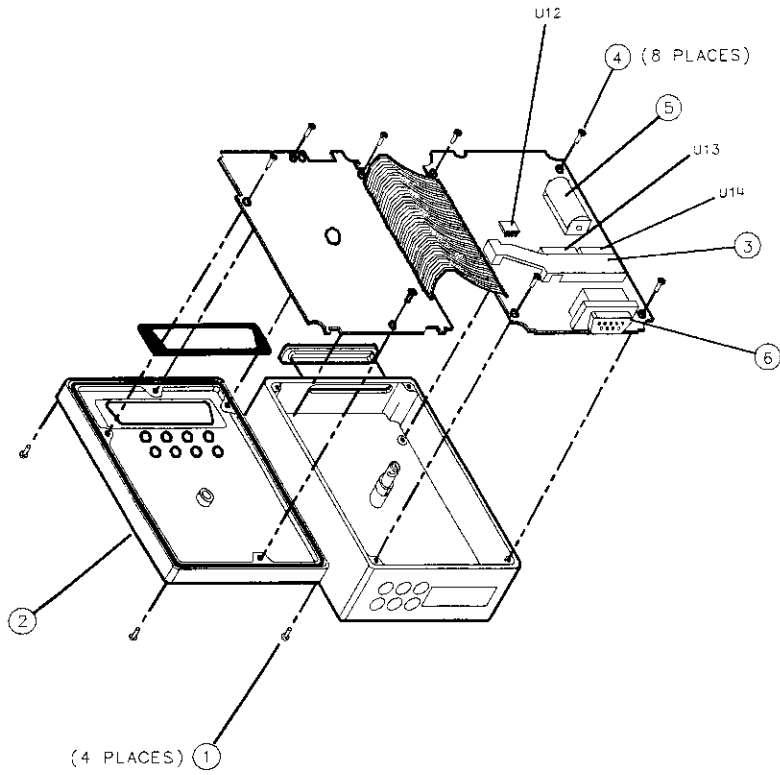
3. Replace the battery and solder it into place.*
4. Replace the board assembly into the module and secure it with the eight board-assembly screws (4). Torque each one to three inch-pounds.
5. Replace the brace (3), then close the module halves.
6. Replace the four module screws (1) and torque each one to six inch-pounds.

* For modules with serial prefix 3143A and above, the battery mounts in a clip and does not need to be soldered in place.



df240

Figure 5-3.
A1 Board Assembly & Battery Replacement (Serial Prefix <3143A)



df213a

Figure 5-4.
A1 Board Assembly & Battery Replacement (Serial Prefix \geq 3143A)

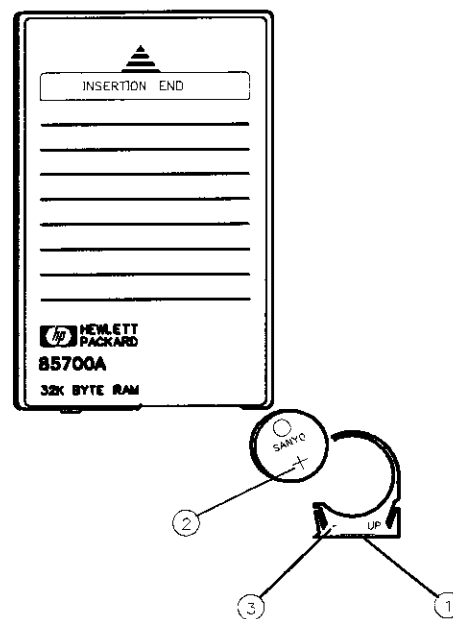
Memory Card Battery Replacement

The battery is located beside the card write-protect switch on the end opposite the connector. Table 1-5 in Chapter 1 contains battery specifications and characteristics. (Option T01 does not include memory card capability.)

Caution

Unless you replace the battery with the card installed and the module powered up by the spectrum analyzer, you lose memory-card data when the battery is removed. Back up memory-card data on some other medium before beginning the battery replacement procedure that follows.

1. Locate the groove (1) along the edge of the battery clip. See Figure 5-5.
2. Gently pry the battery clip out of the card. The battery fits within this clip.
3. Replace the battery, HP part number 1420-0383, making sure the plus (+) sign (2) on the battery is on the same side as the plus (+) sign (3) on the clip.
4. Insert the battery clip into the memory card, holding the clip as oriented in Figure 5-5. (Face the "open" edge of the clip toward the write-protect switch.)



DF 241

Figure 5-5. Memory Card Battery Replacement

Troubleshooting and Replaceable Parts

Introduction The HP 85620A Mass Memory Module is supported to the component level. Refer to the “Troubleshooting” section when troubleshooting the module. Table 5-3 contains the list of components in the module. Refer to Figure 5-6 or Figure 5-7 for the illustration of replaceable hardware. Figure 5-8 is the block diagram. Component level information is at the end of this chapter.

Troubleshooting

Missing Features

It is possible to get a different set of features from the HP 85620A if you connect it to the rear panel of a different spectrum analyzer, depending on the firmware revision of the host spectrum analyzer. In spectrum analyzers with firmware revision 960401 and later, the firmware that controls the mass memory module actually resides in the host spectrum analyzer and contains a more recent set of features and functionality. If the same mass memory module is installed on a host spectrum analyzer with firmware revision 941028 or earlier, the firmware that resides in the mass memory module (revision A, B, or C) will control the features and functionality of the module.

Verify the mass memory firmware revision by pressing **MODULE**. The firmware revision that controls the features and functionality of the module is displayed. Firmware revision 910116 or earlier will be displayed when the host spectrum analyzer firmware revision is 941028 or earlier. Firmware revision 950829 or later will be displayed when the host spectrum analyzer firmware revision is 960401 or later.

If the HP 85620A mass memory module does not exhibit the set of features and functionality that you expect it to have, check its firmware revision as indicated above. New features and functionality that were added in firmware revisions 950829 and later require the use of a host spectrum analyzer with firmware revision 960401 or later.

Back-up Battery Voltage Test

Test the module back-up battery at least annually by connecting a DVM between A1TP1-3 and A1TP1-2 (ground). The dc voltage measurement should be about 2.8 Vdc \pm 0.5 Vdc. See Figure 5-2.

Test the memory card battery at least annually by connecting a DVM between A1TP1-1 and A1TP1-2 (ground). The dc voltage measurement should be about 2.7 Vdc \pm 0.5 Vdc. See Figure 5-2.

Memory Card Connector

If there appear to be intermittent problems and other failures related to using the memory card, a possible cause is memory card connector wear. Connector life can be shortened by frequent memory card insertion. In applications where the memory card is inserted more than twice daily, we recommend that you change the connector on the A1 board assembly at least every 5 years. Replacement part

numbers for the connector and the board assembly are included in the component level information packets.

Error Codes

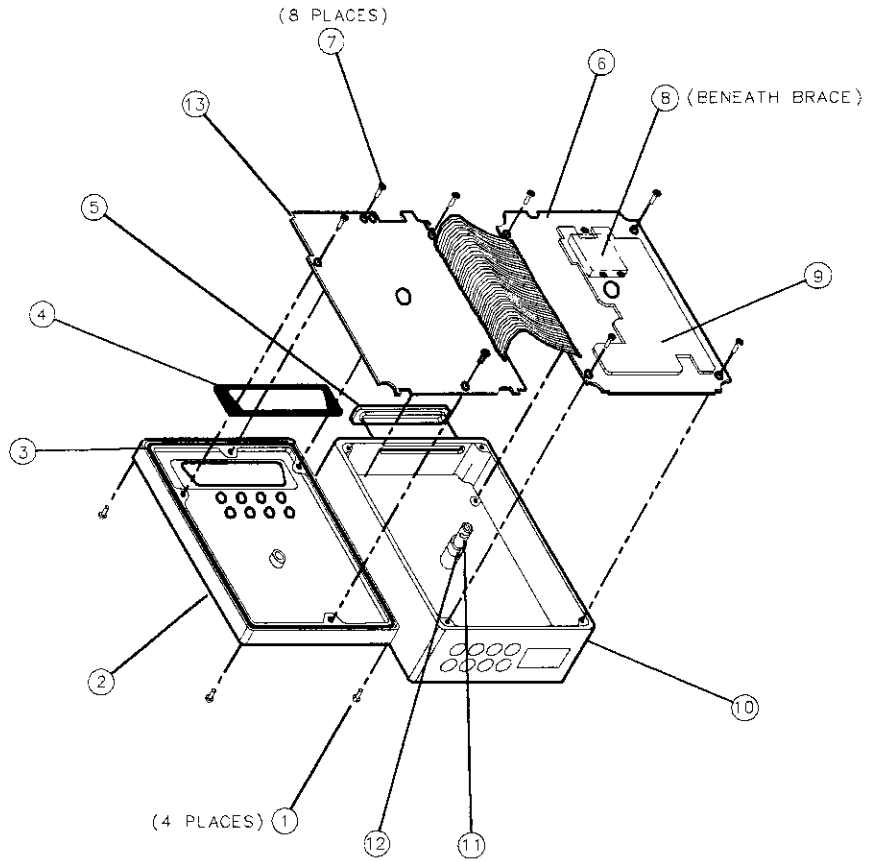
Refer to “Error Codes” in Chapter 1 for an explanation of error-code numbers 800 to 899, which are reserved for the mass memory module and its memory card.

Replaceable Parts

Table 5-3 lists the mechanical parts illustrated in the parts identification drawing, Figure 5-6 or Figure 5-7, depending on the serial prefix of the mass memory module. The component-level replaceable parts, component location diagram, and schematic for each board assembly are contained in individual packets for each board assembly. Refer to component-level information at the end of this chapter.

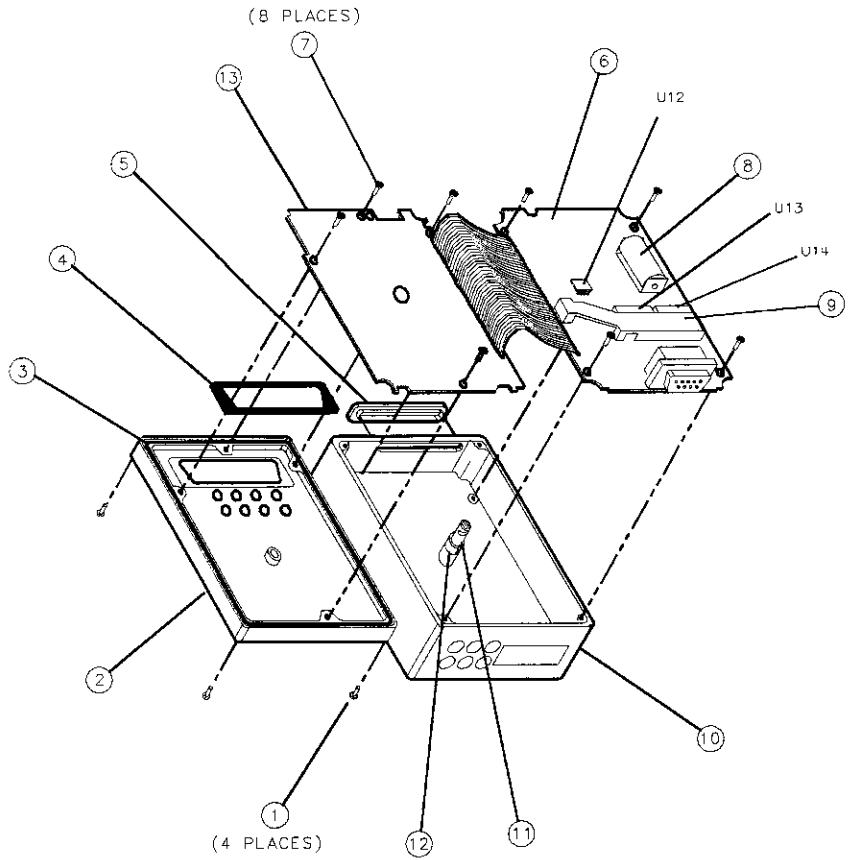
Table 5-2. Manufacturer's Code List

Mfr. Code	Manufacturer Name	Address	ZIP Code
C1433	ABELEKTRONIK GMBH	SALZBURG, AU	A-501
K7253	STD/STANTEL	DEVON, EG	
S4013	HITACHI AMERICA LTD	SUNNYVALE, CA, US	94086
00779	AMP INC	HARRISBURG, PA, US	17111
08709	EPSON AMERICA	TORRANCE, CA, US	75265
01295	TEXAS INSTRUMENTS INC	DALLAS, TX, US	90505
04222	AVX CORP	GREAT NECK, NY, US	11021
10421	PANASONIC, INC	SECAUCUS, NJ, US	07094
06383	PANDUIT CORP	TINLEY PARK, IL, US	60477
06424	SPERRY RAND μ - WAVE ELEK DIV	CLEARWATER, FL, US	33518
10899	EASTERN AIR DEVICES INC	GREAT NECK, NY, US	11021
18873	DUPONT E I DE NEMOURS & CO	WILMINGTON, DE, US	19801
2M627	ROHM CORP	IRVINE, CA, US	92713
28480	HEWLETT-PACKARD CO CORP HQ	PALO ALTO, CA, US	94304
34335	ADVANCED MICRO DEVICES INC	SUNNYVALE, CA, US	94086
71468	ITT CORP	NEW YORK, NY, US	10022
72794	DZUS FASTENER CO INC	WEST ISLIP, NY, US	11795



df242

Figure 5-6. HP 85620A Parts Identification (Serial Prefix <3143A)



d1215a

Figure 5-7. HP 85620A Parts Identification (Serial Prefix \geq 3143A)

Table 5-3. HP 85620A Parts Identification

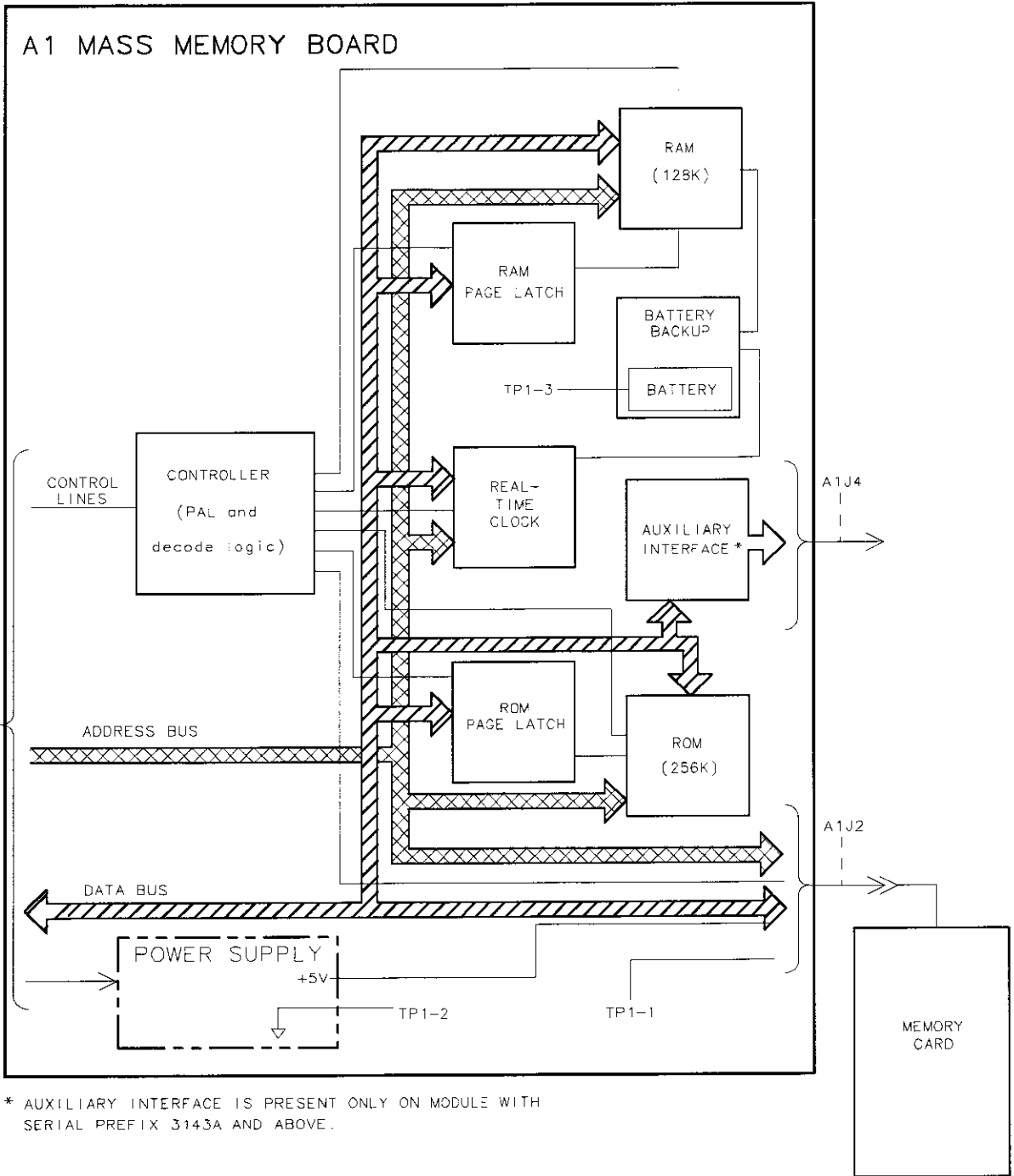
Item	HP Part Number	CD	Qty	Description	Mfg Code	Mfg Part Number
1	0515-1236	9	4	SCREW-MACH M3 X 0.5 14MM-LG	28480	0515-1236
2	5021-9304	6	1	COV-BTM TEST	28480	5021-9304
3	8160-0448	8	1	RFI ROUND STRIP BE-CU SN-PL .062-IN-OD	10899	SS-04
4	8160-0650	4	1	GASKET-OPTION MODULE CONNECTOR	28480	8160-0650
5	85620-40002*	0	1	BEZEL MEMORY CARD SLOT	28480	85620-40002
6	85620-60001†	1	1	BD AY-PORT MEM	28480	85620-60001
6	85620-60006‡	6	1	BD AY-PORT MEM	28480	85620-60006
6	85620-60008§	8	1	BD AY-PORT MEM	28480	85620-60008
6	85620-60014	6	1	BD AY-PORT MEM	28480	85620-60014
7	0515-0894	3	8	SCREW-MACH M2.5 X 0.45 6MM-LG PAN-HD	28480	0515-0894
8	1420-0370†	0	1	BATTERY, 2.8 V 1.0AH	28480	1420-0370
8	1420-0341‡		1	BATTERY, 3 V 1.2AH	28480	1420-0341
8	1420-0394§		1	BATTERY, 3 V 1.0AH	08709	CR2477-1HF
9	85620-40001#	9	1	BRACE	28480	85620-40001
9	5041-8980§**	4	1	BRACE	28480	5041-8980
10	85620-20005†	1	1	COVER-TOP	28480	85620-20005
10	5021-9303§	5	1	COVER-TOP	28480	5021-9303
10	5022-0002‡	5	1	COVER-TOP	28480	5022-0002
11	1390-0750	5	1	FASTENER-1/4-TURN RCPT	72794	A3.5T127B1100-Z
12	5041-7247	4	1	SPACER	28480	5041-7247
	85620-80003	5	1	LABEL SET	28480	85620-80003

* Option TO1 does not include memory card capability.
† Serial prefix 2929A and below
‡ Standard Serial prefix 3143A and above
§ Serial prefix 3003A
|| Option TO1 Serial prefix 3143A and above
Serial prefix 3003A and below
** Serial prefix 3143A

Table 5-4. Firmware Revision and ROM Part Numbers

HP 85620A Firmware Revision/Date Code	ROM Part Number*		
	U1	U2	U3
A/890214	85620-80005	85620-80007	85620-80009
B/890524	85620-80014	85620-80015	85620-80016
C/910116†	85620-80023	85620-80024	85620-80025
	U7 ROM Part Number‡		
C/910116†	85620-80021		
* Serial prefix 3003A and below. † The HP part number for the current firmware revision label is 85620-80003. ‡ Serial prefix 3143A and above.			

See Firmware Note, HP part number 5962-0452, for descriptions of the firmware revisions.



DF243

Figure 5-8. HP 85620A Block Diagram

Component-Level Information

Component-Level information is available for selected instrument assemblies.

Table 5-5 lists board assembly part numbers and where they are used.

Note

Drawings may not be available for recently introduced assemblies.

Table 5-5. HP 85620A Mass Memory Module Documented Assemblies

Board Assembly	Instrument Serial Prefix	Assembly Part Number	CLIP Part Number
Mass Memory Assembly	2929A and below	85620-60001	85620-90033
Mass Memory Assembly	3003A	85620-60008	85620-90034
Mass Memory Assembly	Standard only (3143A)	85620-60006	85620-90035
Mass Memory Assembly	3526A and above	85620-60018	85620-90038
Mass Memory Assembly	Option T01 only 3143A and above	85620-60014	Contact HP (MID)

