Notice

Hewlett-Packard to Agilent Technologies Transition

This documentation supports a product that previously shipped under the Hewlett-Packard company brand name. The brand name has now been changed to Agilent Technologies. The two products are functionally identical, only our name has changed. The document still includes references to Hewlett-Packard products, some of which have been transitioned to Agilent Technologies.
The information contained in this document is subject to change without notice.

Agilent Technologies makes no warranty of any kind with regard to this material, including but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Agilent Technologies shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.
**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
   CLRDSP ............................................... 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
   EN'TE'R' .............................................. 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
LIMFAIL .............................. 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
LIMTSFL ................................. 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
RELHPB ................................. 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
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
Al 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

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 Main Menu Softkeys (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/Recal ................................................................................ 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
# 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” (g-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.

![HP 85620A Mass Memory Module and HP 85700A Memory Card](image)

*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 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**

<table>
<thead>
<tr>
<th>Spectrum Analyzer</th>
<th>HP 85620A Firmware Revision</th>
<th>890720 and later firmware revisions</th>
<th>890720 and later firmware revisions</th>
</tr>
</thead>
<tbody>
<tr>
<td>HP 8560A</td>
<td>Not supported</td>
<td>890720 and later firmware revisions</td>
<td>890720 and later firmware revisions</td>
</tr>
<tr>
<td>HP 8560E</td>
<td>Not supported</td>
<td>941028 and earlier firmware revisions</td>
<td>All revisions of firmware</td>
</tr>
<tr>
<td>HP 8561A</td>
<td>All revisions of firmware</td>
<td>All revisions of firmware</td>
<td>All revisions of firmware</td>
</tr>
<tr>
<td>HP 8561B</td>
<td>Not supported</td>
<td>890720 and later firmware revisions</td>
<td>890720 and later firmware revisions</td>
</tr>
<tr>
<td>HP 8561E</td>
<td>Not supported</td>
<td>941028 and earlier firmware revisions</td>
<td>All revisions of firmware</td>
</tr>
<tr>
<td>HP 8562A</td>
<td>870728 and later firmware revisions</td>
<td>870728 and later firmware revisions</td>
<td>870728 and later firmware revisions</td>
</tr>
<tr>
<td>HP 8562B</td>
<td>870728 and later firmware revisions</td>
<td>870728 and later firmware revisions</td>
<td>870728 and later firmware revisions</td>
</tr>
<tr>
<td>HP 8562E</td>
<td>Not supported</td>
<td>Not supported</td>
<td>All revisions of firmware</td>
</tr>
<tr>
<td>HP 8563A</td>
<td>Not supported</td>
<td>All revisions of firmware</td>
<td>All revisions of firmware</td>
</tr>
<tr>
<td>HP 8563E</td>
<td>Not supported</td>
<td>941028 and earlier firmware revisions</td>
<td>All revisions of firmware</td>
</tr>
<tr>
<td>HP 8564E</td>
<td>Not supported</td>
<td>941028 and earlier firmware revisions</td>
<td>All revisions of firmware</td>
</tr>
<tr>
<td>HP 8565E</td>
<td>Not supported</td>
<td>941028 and earlier firmware revisions</td>
<td>All revisions of firmware</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>Pin Number</th>
<th>Function</th>
<th>Current</th>
<th>“Logic” Mode</th>
<th>“Serial Bit” Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Control A</td>
<td>TTL Out</td>
<td>TTL Output Hi/Lo</td>
<td>TTL Output Hi/Lo</td>
</tr>
<tr>
<td>2</td>
<td>Control B</td>
<td>TTL Out</td>
<td>TTL Output Hi/Lo</td>
<td>TTL Output Hi/Lo</td>
</tr>
<tr>
<td>3</td>
<td>Control C</td>
<td>TTL Out</td>
<td>TTL Output Hi/Lo</td>
<td>Strobe</td>
</tr>
<tr>
<td>4</td>
<td>Control D</td>
<td>TTL Out</td>
<td>TTL Output Hi/Lo</td>
<td>Serial Data</td>
</tr>
<tr>
<td>5</td>
<td>Control I</td>
<td>TTL In</td>
<td>TTL Input Hi/Lo</td>
<td>TTL Input Hi/Lo</td>
</tr>
<tr>
<td>6</td>
<td>Ground</td>
<td></td>
<td>Ground</td>
<td>Ground</td>
</tr>
<tr>
<td>7</td>
<td>-15 Vdc</td>
<td>150 ma</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>+5 Vdc</td>
<td>150 ma</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>+15 Vdc</td>
<td>150 ma</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>t_su</td>
<td>Minimum setup time, data to strobe</td>
<td>250 ns</td>
</tr>
<tr>
<td>t_ho</td>
<td>Minimum hold time, strobe to data</td>
<td>250 ns</td>
</tr>
<tr>
<td>t_w</td>
<td>Minimum pulse width, strobe</td>
<td>500 ns</td>
</tr>
<tr>
<td>t_d</td>
<td>Maximum delay, strobe to data not read</td>
<td>250 ns</td>
</tr>
<tr>
<td>t_latch</td>
<td>Minimum pulse width, latch</td>
<td>150 μs</td>
</tr>
</tbody>
</table>
Memory Card Specifications and Characteristics

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

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

<table>
<thead>
<tr>
<th>Electrical Specifications</th>
<th>General Specifications</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Voltage (supplied by the spectrum analyzer) +5 Vdc (nominal)</td>
<td>Operating Temperature 0°C to +60°C</td>
<td>Card Weight with battery: 20.9 grams without battery: 19.0 grams</td>
</tr>
<tr>
<td>Battery Voltage (RAM back-up) 3 Vdc (nominal)</td>
<td>Storage Temperature −20°C to +60°C</td>
<td>Card Dimensions (Figure 15) Length: 86 mm ±0.2 mm Width: 54 mm ±0.1 mm Thickness: 2.4 mm ±0.15 mm</td>
</tr>
<tr>
<td></td>
<td>Memory Size 32 kilobyte, battery-backed RAM</td>
<td>Battery Type Lithium</td>
</tr>
<tr>
<td></td>
<td>Type CMOS</td>
<td>Commercial Part Number: CR2016</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HP Part Number: 1420-0383</td>
</tr>
</tbody>
</table>
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.
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.

<table>
<thead>
<tr>
<th>Caution</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
</tr>
</tbody>
</table>

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 the 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.

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.

**Figure 1-9.**
HP 85620A Shipping Container Materials
(Option T01 does not include memory card capability.)
<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>HP Part Number</th>
<th>Check Digit</th>
<th>Qty</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Memory card with slip case</td>
<td>85700A</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>(Option TO1 does not include memory card capability.)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Envelope and inner carton</td>
<td>9211-4916</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Bubble-pack bag (with separate card orders)</td>
<td>9222-0784</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>Envelope (with separate card orders)</td>
<td>9222-1219</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>Carton, outer</td>
<td>9211-5570</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>Styrene sheets</td>
<td>9223-0476</td>
<td>7</td>
<td>2</td>
</tr>
</tbody>
</table>

**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\textsuperscript{TM} 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 [P Reset] 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.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>800</td>
<td>U1 EPROM Check Sum Error, hardware failure.</td>
</tr>
<tr>
<td>801</td>
<td>U2 EPROM Check Sum Error, hardware failure.</td>
</tr>
<tr>
<td>802</td>
<td>U3 EPROM Check Sum Error, hardware failure.</td>
</tr>
<tr>
<td>803</td>
<td>U4 EPROM Check Sum Error, hardware failure.</td>
</tr>
<tr>
<td>804</td>
<td>Mass Memory Initialized. RAM data has been erased. Hardware failure or the module battery is intermittent or has failed.</td>
</tr>
<tr>
<td>805</td>
<td>Mass Memory Module Usage Error. The command used was invalid and cannot be executed.</td>
</tr>
<tr>
<td>806</td>
<td>Mass Memory RAM full. Available memory is insufficient for command execution.</td>
</tr>
<tr>
<td>807</td>
<td>Symbol Define Error. There was an attempt to define something illegal, or not of the correct type.</td>
</tr>
</tbody>
</table>
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 TO1 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 TO1 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 TO1 does not include memory card capability.)
854 Memory Card Memory Full. (Option TO1 does not include memory card capability.)
855 Memory Card File Not Found. (Option TO1 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

<table>
<thead>
<tr>
<th>Region</th>
<th>Location</th>
<th>Phone Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Headquarters</strong></td>
<td>Hewlett-Packard Co. 19320 Pruneridge Avenue Cupertino, CA 95014 (800) 752-0900</td>
<td>California, Northern Hewlett-Packard Co. 301 E. Evelyn Mountain View, CA 94041 (415) 694-2000</td>
</tr>
<tr>
<td><strong>Colorado</strong></td>
<td>Hewlett-Packard Co. 24 Inverness Place, East Englewood, CO 80112 (303) 649-5512</td>
<td>California, Southern Hewlett-Packard Co. 1421 South Manhattan Ave. Fullerton, CA 92631 (714) 999-6700</td>
</tr>
<tr>
<td><strong>New Jersey</strong></td>
<td>Hewlett-Packard Co. 150 Green Pond Rd. Rockaway, NJ 07866 (201) 686-6400</td>
<td>Atlanta Annex Hewlett-Packard Co. 2124 Barrett Park Drive Kennesaw, GA 30144 (404) 698-0000</td>
</tr>
<tr>
<td><strong>Texas</strong></td>
<td>Hewlett-Packard Co. 930 E. Campbell Rd. Richardson, TX 75081 (214) 231-6101</td>
<td>Illinois Hewlett-Packard Co. 5201 Tollview Drive Rolling Meadows, IL 60008 (708) 265-9800</td>
</tr>
</tbody>
</table>

#### EUROPEAN FIELD OPERATIONS

<table>
<thead>
<tr>
<th>Region</th>
<th>Location</th>
<th>Phone Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Headquarters</strong></td>
<td>Hewlett-Packard S.A. 160, Route du Nant-d’Avril 12 17 Meyrin 2/Geneva Switzerland (4122) 780.8111</td>
<td>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</td>
</tr>
<tr>
<td><strong>Great Britain</strong></td>
<td>Hewlett-Packard Ltd. Seakdale Road, Winnersh Triangle Wokingham, Berkshire RG41 5DZ England (44 734) 696622</td>
<td>Germany Hewlett-Packard GmbH 61362 Bad Homburg v.d.H Germany (49 6172) 16-0</td>
</tr>
</tbody>
</table>

#### INTERCON FIELD OPERATIONS

<table>
<thead>
<tr>
<th>Region</th>
<th>Location</th>
<th>Phone Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Headquarters</strong></td>
<td>Hewlett-Packard Company 3495 Deer Creek Road Palo Alto, California, USA 94304-1916 (415) 857-5027</td>
<td>Australia Hewlett-Packard Australia Ltd. 31-41 Joseph Street Blackburn, Victoria 3130 (61 3) 896-2895</td>
</tr>
<tr>
<td><strong>China</strong></td>
<td>Hewlett-Packard Company 38 Bei San Huan X1 Road ShuangYuShu Jia Dian District Beijing, china (86 1) 266-6888</td>
<td>Japan Hewlett-Packard Japan, Ltd. 1-27-15 Yabe, Sagamihara Kanagawa 229, Japan (81 427) 59-1311</td>
</tr>
<tr>
<td><strong>Taiwan</strong></td>
<td>Hewlett-Packard Ibiwan 8th Floor, H-P Building 137 Fu Hsing North Road Taipei, Ibiwan (886 2) 712-0404</td>
<td>Canada Hewlett-Packard (Canada) Ltd. 17600 South Service Road Trans-Canada Highway Kirkland, Quebec H9J 2X8 Canada (614) 697-4232</td>
</tr>
<tr>
<td><strong>Singapore</strong></td>
<td>Hewlett-Packard Singapore (Pte.) Ltd. 150 Beach Road #28-00 Gateway West Singapore 0718 (65) 291-9088</td>
<td>Intercon Field Operations</td>
</tr>
</tbody>
</table>
2

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.

Figure 2-1. HP 856208 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.

![Figure 2-2. User Keys Menu with DLP Label Examples](image)

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

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.

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 TO1 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 TRACE 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.
A maximum of one trace per second can be saved.

**Note**

**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 datecode 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.
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.
**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).

---

**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

<table>
<thead>
<tr>
<th>Trigger</th>
<th>Start Time</th>
<th>Stop Time</th>
<th>Autosave Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unspecified</td>
<td>Unspecified</td>
<td>Unspecified</td>
<td>Autosave off.</td>
</tr>
<tr>
<td>Unspecified</td>
<td>Unspecified</td>
<td>Specified</td>
<td>Autosave data now. Save at the end of each sweep. Autosave until specified stop time is reached or memory is full.</td>
</tr>
<tr>
<td>Unspecified</td>
<td>Specified</td>
<td>Unspecified</td>
<td>Start at specified time. Save at the end of each sweep, then until memory is full.</td>
</tr>
<tr>
<td>Specified</td>
<td>Specified</td>
<td>Specified</td>
<td>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.</td>
</tr>
<tr>
<td>Specified</td>
<td>Unspecified</td>
<td>Unspecified</td>
<td>Autosave data when trigger is reached and until memory is full.</td>
</tr>
</tbody>
</table>

**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.

---

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 datecode 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 $+50 \text{ dBm}$. If a lower-limit value is not entered, the lower limit assumes the default value of $-175 \text{ dBm}$. 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 $-10 \text{ dB}$ indicates that $-10 \text{ dB}$ 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.
**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.
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 TO1
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.
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.
Figure 2-11. Edit Line Menu Softkeys and Table Example

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 (BKSP) 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.
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.

DELT A 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 or 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.

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 A
select upper/lower to enter amplitude values that display a limit line above or  
below an active trace. Use the delta (A) 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.

![Diagram of AUTOEXEC Menu](image)

**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.

![Diagram of Edit Autoexec Softkeys](image)

**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. 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.

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.

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.

AI TOEXEC 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 TO1 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.
**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.
Press UTILITY to gain access to the utility functions available with the HP 85620A Mass Memory Module. See Figure 2-20.
TIMEDATE
activates menu selections for setting the time and date. See Figure 2-22 for the softkeys defined below.

**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** 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.
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
Figure 3-1. User Keys
Figure 3-2. Trace Save/Rcl

* default
Figure 3-3. Limit Line

- default

† This softkey label appears only if the spectrum analyzer has printing capability.
Figure 3-4. Autoexec Menu

- default
Figure 3-5. Keydef
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.

![Figure 4-1. Equipment Setup](DF235)
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
```

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, G;IP;CF 300MHZ;SP 10MHZ;G;";
   !Assign the
   !label "ZOOM"
   !to the indicated
   !sequence of commands.

20

30

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, and NOW.

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

![Figure 4-3. Typical DLP Listing](image)
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,0;IP;CF 300MHZ;SP 10MHZ;0;";
20 ! Assigns the label ZOOM to the indicated sequence of commands.
30
40 OUTPUT 718;"KEYDEF 1, ZOOM,%ZOOM%;";
50 ! Assigns the label "ZOOM" 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 value and prepare to output value.
20 ! value and prepare to output value.
30 ENTER 718; "M" ! Store the available user-memory value in the variable "M".
40 ! value in the variable "M".
50 PRINT "MEMORY = "; "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 x 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**

```
IO 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
to 275 and 325 MHz respectively.
50 ! Put a marker on the largest
60 OUTPUT 718;"TS;MKPK HI;"; ! signal on the trace.
70 ! Move the marker amplitude
80 OUTPUT 718;"MOV AMPLITUDE,MKA;"; ! value into "AMPLITUDE".
90 !
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**

```
IO 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 $1 \times 10^6$
- MHz/-dBm/sec ....................... Multiplies entered value by $1 \times 10^6$
- kHz/mV/ms .......................... Multiplies entered value by $1 \times 10^3$
- 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.

```plaintext
10 !
20 ASSIGN OSa TO 718
30 !
40 OUTPUT OSa;"FUNCDEF H_ARMDIST,0;"
50 OUTPUT OSa;"LCLVAR E_NTRY,0;"
60 OUTPUT OSa;"IP;SNGLS;TS;"
70 OUTPUT OSa;"HD;PU;PA100,575;TEXT/ENTER FUNDAMENTAL FREQUENCY/;"
80 OUTPUT OSa;"E_NTRY EP;"
90 OUTPUT OSa;"EM;"
100 OUTPUT OSa;"DIV FA,E_NTRY,2;"
110 OUTPUT OSa;"MPY FB,E_NTRY,3.5;"
120 OUTPUT OSa;"TS;MKPK HI;MKD;MKPK NH;"
130 OUTPUT OSa;"@;"
140 !
150 OUTPUT OSa;"KEYDEF 1,H_ARMDIST,/ THD SETUP/;"
160 !
170 ASSIGN OSa TO *
180 END
```

Programming 4-8
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 MEM 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;";
20 ! Define a variable named POWER and initialized it to zero.
30 OUTPUT 718;"TRDEF SAVE, 601;";
40 ! Define a 601-point trace named SAVE.
50 !
60 OUTPUT 718;"FUNCDEF CHECK,Q;";
70 ! Define a function named "CHECK".
80 OUTPUT 718;"IP;";
90 ! Perform instrument preset.
100 OUTPUT 718;"SNGLS;CF 600MHZ;SP 1GZ;TS;";

110 OUTPUT 718;"MKPK HI;TS;";
120 ! Set to single sweep, set the center frequency to 600 MHz,
130 OUTPUT 718;"IF MA,GT,-60 THEN;";
140 OUTPUT 718;" SP 1KZ;TS;TS;";
145 OUTPUT 718;" MKRL;TS;";
150 OUTPUT 718;" MOV POWER, MKA;";
160 ! Move marker to reference level to the previously defined variable POWER.
170 OUTPUT 718;" MOV SAVE, TRA;";
180 ! Move Trace A into previously defined TRDEF named SAVE.
190 OUTPUT 718;"ENDIF;";
200 ! End the IF statement.
210 END ! End the definition of CHECK.

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.
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, O;"; ! Define a variable named "POWER"
120 OUTPUT 718; "TRDEF SAVE, 601;"; ! and initialize it to zero.
130 OUTPUT 718; "TRDEF SAVE, 601;"; ! Define a 601-point trace, "SAVE".
140 ! Subroutines:
150 OUTPUT 718; "FUNCDEF SPANONE, Q;"; ! Define function SPANONE
160 OUTPUT 718; "FA 10MHZ;FB 12MHZ;"; ! Set the start and stop frequencies
170 OUTPUT 718; "@;";
180 OUTPUT 718; "FUNCDEF SPANTWO, Q;"; ! Define function SPANTWO.
190 OUTPUT 718; "FA 12MHZ;FB 14MHZ;";
200 OUTPUT 718; "@;";
210 OUTPUT 718; "FUNCDEF SPANTHREE, Q;";
220 OUTPUT 718; "FA 14MHZ;FB 16MHZ;";
230 OUTPUT 718; "@;";
240 OUTPUT 718; "FUNCDEF SPANFOUR, Q;";
250 OUTPUT 718; "FA 16MHZ;FB 110MHZ;";
260 OUTPUT 718; "@;";
270 OUTPUT 718; "FUNCDEF CHECK, Q;";
280 OUTPUT 718; "TS;MKPK HI;";
290 OUTPUT 718; "IF MA, GT, -50 THEN";
300 OUTPUT 718; "MKTRACK ON;";
310 OUTPUT 718; "SP 1MHZ;";
320 OUTPUT 718; "MKTRACK OFF;TS;";
330 OUTPUT 718; "MKPK HI;MKCF;TS;";
340 OUTPUT 718; "MOV POWER, MKA;";
350 OUTPUT 718; "MOV SAVE, TRA;";
360 OUTPUT 718; "SAVES 1;";
370 OUTPUT 718; "ENDIF;";
380 OUTPUT 718; "FUNCDEF EXAMPLE, Q;";
390 OUTPUT 718; "IP;SNGLS;MOV SAVE, O;";
400 OUTPUT 718; "REPEAT;";
410 OUTPUT 718; "SPANONE;CHECK;";
420 OUTPUT 718; "SPANTWO;CHECK;";
430 OUTPUT 718; "SPANTHREE;CHECK;";
440 OUTPUT 718; "SPANFOUR;CHECK;";
450 OUTPUT 718; "UNTIL SAVE[O], NE, O;";
460 OUTPUT 718; "MOV TRB, SAVE;";
470 OUTPUT 718; "RCLS 1;BLANK TRA;VIEW TRB;";

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.

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.
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.
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.

greater 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.

character sp ! " # $ % & ( \ ) [ ] ? @ + , - . / : ; 

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 $-1.790 \times 10^{30}$ through $-2.225 \times 10^{-30}$, 0, and $+2.225 \times 10^{-30}$ through $+1.798 \times 10^{30}$.

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
- OFFSET
- MKT
- SS
- VTL
- HNLOCK
- ML
- ST
- LIMIFAIL*

* Firmware datecode 910116 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.
units

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

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Amplitude</th>
<th>Time</th>
<th>Current</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>HZ</td>
<td>DB</td>
<td>S</td>
<td>A</td>
<td>EP1</td>
</tr>
<tr>
<td>KHZ</td>
<td>DBM</td>
<td>SC</td>
<td>MA</td>
<td></td>
</tr>
<tr>
<td>MHZ</td>
<td>DBMV</td>
<td>US</td>
<td>UA</td>
<td></td>
</tr>
<tr>
<td>GHZ</td>
<td>DBUV</td>
<td>MS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KZ</td>
<td>DM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MZ</td>
<td>MV</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GZ</td>
<td>u v</td>
<td>MW</td>
<td>V</td>
<td></td>
</tr>
</tbody>
</table>

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 (_).
Command Syntax Diagrams

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>|<b> 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>::=<b><c> indicates that <a> can be replaced by the series of elements, <b><c> in any statement where <a> occurs.

{}

enclose descriptive comment which refers to the preceding item in the command sequence.
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

<table>
<thead>
<tr>
<th>Function</th>
<th>Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto-function command</td>
<td>AUTOEXEC</td>
<td>Automatic execution</td>
</tr>
<tr>
<td></td>
<td>AUTOFUNC</td>
<td>Automatic function</td>
</tr>
<tr>
<td></td>
<td>AUTOSAVE</td>
<td>Automatic trace save</td>
</tr>
<tr>
<td></td>
<td>CLRSCHED</td>
<td>Clear autoexec/autosave schedule</td>
</tr>
<tr>
<td>Active function command</td>
<td>ACTVFUNC</td>
<td>Active function</td>
</tr>
<tr>
<td>Clock command</td>
<td>SETDATE</td>
<td>Set date</td>
</tr>
<tr>
<td></td>
<td>SETTIME</td>
<td>Set time</td>
</tr>
<tr>
<td></td>
<td>TIMEDATE</td>
<td>Set time and date display on or off</td>
</tr>
<tr>
<td>Graphics operation command</td>
<td>CLRDSP</td>
<td>Clear display</td>
</tr>
<tr>
<td></td>
<td>DATEMODE</td>
<td>Date format display</td>
</tr>
<tr>
<td></td>
<td>DSPLY</td>
<td>Display</td>
</tr>
<tr>
<td></td>
<td>EM</td>
<td>Erase user graphics memory</td>
</tr>
<tr>
<td></td>
<td>OR</td>
<td>Set origin</td>
</tr>
<tr>
<td></td>
<td>PA</td>
<td>Plot absolute</td>
</tr>
<tr>
<td></td>
<td>PD</td>
<td>Pen down</td>
</tr>
<tr>
<td></td>
<td>PR</td>
<td>Plot relative</td>
</tr>
<tr>
<td></td>
<td>PU</td>
<td>Pen up</td>
</tr>
<tr>
<td></td>
<td>TEXT</td>
<td>Screen text</td>
</tr>
<tr>
<td>HP-IB control command</td>
<td>CTRLHPIB</td>
<td>Control HP-IB</td>
</tr>
<tr>
<td></td>
<td>ENTER</td>
<td>Enter from HP-IB</td>
</tr>
<tr>
<td></td>
<td>OUTPUT</td>
<td>Output from HP-IB</td>
</tr>
<tr>
<td></td>
<td>RELHPIB</td>
<td>Release HP-IB control</td>
</tr>
<tr>
<td>Limit-line command</td>
<td>EDITDONE</td>
<td>Limit-line edit done</td>
</tr>
<tr>
<td></td>
<td>EDITLIML</td>
<td>Edit limit line</td>
</tr>
<tr>
<td></td>
<td>LIMD</td>
<td>Limit-line segment delta</td>
</tr>
<tr>
<td></td>
<td>LIMF</td>
<td>Limit-line segment freq</td>
</tr>
<tr>
<td></td>
<td>LIMFAIL</td>
<td>Limit-line test fail status</td>
</tr>
<tr>
<td></td>
<td>LIMIPURGE</td>
<td>Purge limit line</td>
</tr>
<tr>
<td></td>
<td>LIMIREL</td>
<td>Relative limit line</td>
</tr>
<tr>
<td></td>
<td>LIMIRCL</td>
<td>Recall limit line</td>
</tr>
<tr>
<td></td>
<td>LIMISAV</td>
<td>Save limit line</td>
</tr>
<tr>
<td></td>
<td>LIMITEST</td>
<td>Limit-line test mode</td>
</tr>
<tr>
<td></td>
<td>LIML</td>
<td>Lower amplitude limit</td>
</tr>
<tr>
<td></td>
<td>LIMM</td>
<td>Middle amplitude limit</td>
</tr>
<tr>
<td></td>
<td>LIMTFL</td>
<td>Flat limit-line segment</td>
</tr>
<tr>
<td></td>
<td>LIMITSL</td>
<td>Sloped limit-line segment</td>
</tr>
<tr>
<td></td>
<td>LIMU</td>
<td>Upper limit-line value</td>
</tr>
<tr>
<td></td>
<td>SADD</td>
<td>Add limit-line segment</td>
</tr>
<tr>
<td></td>
<td>SDEL</td>
<td>Delete limit-line segment</td>
</tr>
<tr>
<td></td>
<td>SDON</td>
<td>Limit-line segment edit done</td>
</tr>
<tr>
<td></td>
<td>SEDI</td>
<td>Edit limit-line segment</td>
</tr>
<tr>
<td></td>
<td>SENTER</td>
<td>Enter limit-line segment parameters</td>
</tr>
</tbody>
</table>
### Table 4-1. Functional Command Listing (continued)

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

4-24 Programming
### Table 4-1. Functional Command Listing (continued)

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

### Table 4-2. Alphabetic Command Listing

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Description</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABORT</td>
<td>Abort</td>
<td>User program control flow</td>
</tr>
<tr>
<td>ABS</td>
<td>Absolute</td>
<td>Predefined math operation</td>
</tr>
<tr>
<td>ACTVFUNC</td>
<td>Active function</td>
<td>Active function command</td>
</tr>
<tr>
<td>ADD</td>
<td>Addition</td>
<td>Predefined math operation</td>
</tr>
<tr>
<td>ARRAYDEF</td>
<td>Array definition</td>
<td>User definition command</td>
</tr>
<tr>
<td>AUTOEXEC</td>
<td>Automatic execute</td>
<td>Auto-function command</td>
</tr>
<tr>
<td>AUTOFUNC</td>
<td>Automatic function</td>
<td>Auto-function command</td>
</tr>
<tr>
<td>AUTOSAVE</td>
<td>Automatic save trace</td>
<td>Auto-function command</td>
</tr>
<tr>
<td>AVG</td>
<td>Average trace</td>
<td>Predefined math operation</td>
</tr>
<tr>
<td>CARDLOAD</td>
<td>Copy card data to memory</td>
<td>Memory operation command</td>
</tr>
<tr>
<td>CARDSTORE</td>
<td>Copy memory data to card</td>
<td>Memory operation command</td>
</tr>
<tr>
<td>CATALOG</td>
<td>Catalog module memory or card over HP-IB</td>
<td>Memory operation command</td>
</tr>
<tr>
<td>CLRDSP</td>
<td>Clear display</td>
<td>Graphics control command</td>
</tr>
<tr>
<td>CLRSCALL</td>
<td>Clear autoexec/autosave schedule</td>
<td>Auto-function command</td>
</tr>
<tr>
<td>CNTLA</td>
<td>Sets control line A</td>
<td>Aux interface command</td>
</tr>
<tr>
<td>CNTLB</td>
<td>Sets control line B</td>
<td>Aux interface command</td>
</tr>
<tr>
<td>CNTLC</td>
<td>Sets control line C</td>
<td>Aux interface command</td>
</tr>
<tr>
<td>CNTLD</td>
<td>Sets control line D</td>
<td>Aux interface command</td>
</tr>
<tr>
<td>CNTLI</td>
<td>Status of control line I</td>
<td>Aux interface command</td>
</tr>
<tr>
<td>CTRLHPIB</td>
<td>Control HP-IB</td>
<td>HP-IB control command</td>
</tr>
<tr>
<td>DATEMODE</td>
<td>Date mode</td>
<td>Clock control command</td>
</tr>
<tr>
<td>DISPOSE</td>
<td>Dispose</td>
<td>User definition command</td>
</tr>
<tr>
<td>DIV</td>
<td>Divide</td>
<td>Predefined math operation</td>
</tr>
<tr>
<td>DSPLY</td>
<td>Display</td>
<td>Graphics control command</td>
</tr>
<tr>
<td>EDITDONE</td>
<td>Edit done</td>
<td>Limit-line command</td>
</tr>
<tr>
<td>EDITLIML</td>
<td>Edit limit line</td>
<td>Limit-line command</td>
</tr>
<tr>
<td>EM</td>
<td>Erase user display memory</td>
<td>HP-IB control command</td>
</tr>
<tr>
<td>ENTER</td>
<td>Enter</td>
<td>HP-IB control command</td>
</tr>
<tr>
<td>EXP</td>
<td>Exponent</td>
<td>Predefined math operation</td>
</tr>
<tr>
<td>Mnemonic</td>
<td>Description</td>
<td>Function</td>
</tr>
<tr>
<td>------------</td>
<td>---------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>FORMAT</td>
<td>Format</td>
<td>Formats the memory card</td>
</tr>
<tr>
<td>FUNCDEF</td>
<td>Function definition</td>
<td>User definition command</td>
</tr>
<tr>
<td>IF/THEN/ELSE/END</td>
<td>IF</td>
<td>User program control flow</td>
</tr>
<tr>
<td>INT</td>
<td>Integer</td>
<td>Predefined math operation</td>
</tr>
<tr>
<td>KEYCLR</td>
<td>Clear a user key</td>
<td>Module menu command</td>
</tr>
<tr>
<td>KEYDEF</td>
<td>Define a user key</td>
<td>Module menu command</td>
</tr>
<tr>
<td>LCLVAR</td>
<td>Local variable</td>
<td>User definition command</td>
</tr>
<tr>
<td>LIMD</td>
<td>Limit-line delta</td>
<td>Limit-line command</td>
</tr>
<tr>
<td>LIMF</td>
<td>Limit-line frequency</td>
<td>Limit-line command</td>
</tr>
<tr>
<td>LIMFFAIL</td>
<td>Limit-line fail</td>
<td>Limit-line command</td>
</tr>
<tr>
<td>LIMIPURGE</td>
<td>Limit-line purge</td>
<td>Limit-line command</td>
</tr>
<tr>
<td>LIMIRCL</td>
<td>Limit-line recall</td>
<td>Limit-line command</td>
</tr>
<tr>
<td>LIMIREL</td>
<td>Relative limit line</td>
<td>Limit-line command</td>
</tr>
<tr>
<td>LIMSAV</td>
<td>Limit-line save</td>
<td>Limit-line command</td>
</tr>
<tr>
<td>LIMITEST</td>
<td>Limit-line test</td>
<td>Limit-line command</td>
</tr>
<tr>
<td>LIML</td>
<td>lower limit value</td>
<td>Limit-line command</td>
</tr>
<tr>
<td>LIMM</td>
<td>Middle limit value</td>
<td>Limit-line command</td>
</tr>
<tr>
<td>LIMTFL</td>
<td>Flat limit-line segment</td>
<td>Limit-line command</td>
</tr>
<tr>
<td>LIMTSL</td>
<td>Slope limit-line segment</td>
<td>Limit-line command</td>
</tr>
<tr>
<td>LIMU</td>
<td>Upper limit value</td>
<td>Limit-line command</td>
</tr>
<tr>
<td>LOG</td>
<td>Logarithm</td>
<td>Predefined math operation</td>
</tr>
<tr>
<td>MEAN</td>
<td>Trace mean</td>
<td>Predefined math operation</td>
</tr>
<tr>
<td>MEM</td>
<td>Memory available</td>
<td>Memory operation command</td>
</tr>
<tr>
<td>MENU</td>
<td>Menu</td>
<td>Module menu command</td>
</tr>
<tr>
<td>MIN</td>
<td>Minimum</td>
<td>Predefined math operation</td>
</tr>
<tr>
<td>MOD</td>
<td>Modulo</td>
<td>Predefined math operation</td>
</tr>
<tr>
<td>MODRCLT</td>
<td>Recall module trace</td>
<td>Trace operation command</td>
</tr>
<tr>
<td>MODSAVE</td>
<td>Save module trace</td>
<td>Trace operation command</td>
</tr>
<tr>
<td>MOV</td>
<td>Move</td>
<td>Predefined math operation</td>
</tr>
<tr>
<td>MPY</td>
<td>Multiply</td>
<td>Predefined math operation</td>
</tr>
<tr>
<td>MSDEV</td>
<td>Mass storage device</td>
<td>Memory operation command</td>
</tr>
<tr>
<td>MXM</td>
<td>Maximum</td>
<td>predefned math operation</td>
</tr>
</tbody>
</table>
### Table 4-2. Alphabetical Command Listing (continued)

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

Abort Operation

Syntax

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 Os to 718 ! Assign I/O path to address 718.
60 |
70 OUTPUT Os;"FUNCDEF TST,@;"; ! Logical start of the DLP.
80 OUTPUT Os;"VARDEF Y,500;"; ! Create variable and initialize to 500.
90 OUTPUT Os;"VARDEF X,500;"; ! Create variable and initialize to 500.
100 OUTPUT Os;"CLRDSP;"; ! Clear display.
110 OUTPUT Os;"REPEAT;"; ! Begin loop.
120 OUTPUT Os;"IF Y,LT,100;THEN;"; ! Test condition.
130 OUTPUT Os;"PU;PA 100,X;"; ! Move pen.
140 OUTPUT Os;"TEXT%DLP ABORTED%;"; ! Print text.
150 OUTPUT Os;"ABORT;";
160 OUTPUT Os;"ELSE;";
170 OUTPUT Os;"PU;PA 100,X;"; ! Move pen.
180 OUTPUT Os;"DSPLY Y,5,2;"; ! Display ‘Y’.
190 OUTPUT Os;"SUB X,X,40;"; ! Decrement pen pointer.
200 OUTPUT Os;"SUB Y,Y,100;"; ! Decrement variable.
210 OUTPUT Os;"ENDIF;";
220 OUTPUT Os;"UNTIL Y,EQ,-100;"; ! End loop (the abort will occur
230 ! before this condition will be
240 ! satisfied).
250 OUTPUT Os;"@;"; ! Logical end of DLP.
260 |
270 OUTPUT Os;"TST;"; ! Execute DLP.
280 |
290 ASSIGN Os to * ! Close I/O path.
300 END
ABS

Absolute Value

Syntax

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

Active Function

Syntax

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.
ACTVFUNC

**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**

```plaintext
10 ! The following example shows how 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,0;"; ! 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,0;"; ! 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
270 ! example."
280 PRINT !
290 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
```
Description  The values of source 1 and 2 are added and the sum is sent to the destination.
Program Example

! The following example shows the use of the ADD command. The example uses an input signal of 300 MHz, such as the CAL OUTPUT signal, and looks at its harmonics. An input signal is not necessary to the function of this example, but acts as a visual aid.

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td></td>
<td>! Assign I/O path to address 718.</td>
</tr>
<tr>
<td>60</td>
<td>ASSIGN @Sa TO 718</td>
<td>! Assign I/O path to address 718.</td>
</tr>
<tr>
<td>70</td>
<td></td>
<td>! Logical start of the DLP function.</td>
</tr>
<tr>
<td>80</td>
<td>OUTPUT @Sa;&quot;FUNCDEF ADDEX,F;&quot;;</td>
<td>! Logical start of the DLP function.</td>
</tr>
<tr>
<td>90</td>
<td>OUTPUT @Sa;&quot;VARDEF FREQ,300E6;&quot;;</td>
<td>! Create variable and initialize it to 300 MHz.</td>
</tr>
<tr>
<td>100</td>
<td>OUTPUT @Sa;&quot;IP;SP 1MHZ;&quot;;</td>
<td>! Instrument preset; set span.</td>
</tr>
<tr>
<td>110</td>
<td></td>
<td>! Begin loop.</td>
</tr>
<tr>
<td>120</td>
<td>OUTPUT @Sa;&quot;REPEAT;&quot;;</td>
<td>! Begin loop.</td>
</tr>
<tr>
<td>130</td>
<td>OUTPUT @Sa;&quot;MOV CF,FREQ;&quot;;</td>
<td>! Set center frequency.</td>
</tr>
<tr>
<td>140</td>
<td>OUTPUT @Sa;&quot;TS;&quot;;</td>
<td>! Take a sweep to update display.</td>
</tr>
<tr>
<td>150</td>
<td>OUTPUT @Sa;&quot;ADD FREQ,FREQ,300E6;&quot;;</td>
<td>! Increase FREQ by 300 MHz.</td>
</tr>
<tr>
<td>160</td>
<td>OUTPUT @Sa;&quot;UNTIL FREQ,GT,3E9;&quot;;</td>
<td>! End of loop.</td>
</tr>
<tr>
<td>170</td>
<td>OUTPUT @Sa;&quot;Q;&quot;;</td>
<td>! Logical end of DLP function.</td>
</tr>
<tr>
<td>190</td>
<td>OUTPUT @Sa;&quot;ADDEX;&quot;;</td>
<td>! Execute function.</td>
</tr>
<tr>
<td>210</td>
<td>ASSIGN @Sa TO * ! Close I/O path.</td>
<td></td>
</tr>
<tr>
<td>220</td>
<td>END</td>
<td>! Close I/O path.</td>
</tr>
</tbody>
</table>
Array Definition

Syntax

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

Automatic Execution

Syntax

Description

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

Prerequisite Command: AUTOFUNC

Related Commands: AUTOSAVE, CLRSCHED

Program Example

```plaintext
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 999.
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
```

Programming 4-37
**AUTOFUNC**

**Automatic Function**

**Syntax**

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: MMDDYVHHMMS or DDMYYYYHHMMSS (depending on \texttt{datemode} format)
4. Stop-time: MMDDYVHHMMS or DDMYYYYHHMMSS (depending on \texttt{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 QSa TO 718 ! Assign I/O path to address 718.
40 !
50 OUTPUT QSa;"IP;" ; ! Instrument preset.
60 ! Create three items for the AUTOEXEC Schedule.
70 OUTPUT QSa;"AUTOFUNC 1,BOX,010189000000,010589000000 EOS;";
80 OUTPUT QSa;"AUTOFUNC 2,CHK_SIG,123188235500,123188235900 INTVL 000030;";
90 OUTPUT QSa;"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 QSa;"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 QSa;"IP;CLRSCHED ALL;";
220 PRINT "Press \{LCL\} [MODULE] AUTOEXEC MENU to see the cleared schedule."
230 DISP ""
240 !
250 ASSIGN QSa TO *
260 END
AUTOSAVE

Automatically Save Traces

Syntax

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.
**Average**

**Syntax**

\[ \text{DEST}_{\text{new}} = \frac{(\text{RATIO} - 1) \times \text{DEST}_{\text{old}} + \text{SOURCE}}{\text{RATIO}} \]

**Description**

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

**Parameter Range**

- Average Ratio:
  - Minimum: -32,767
  - Maximum: + 32,767
  - 0 is not allowed
**Program Example**

```
10 ! The following example shows the use of the AVG command.
20 !
30 ASSIGN OSa TO 718 ! Assign I/O path to address 718.
40 !
50 OUTPUT OSa;"IP;"; ! Instrument preset.
60 OUTPUT OSa;"SNGLS;CLRW TRA;TS;"; ! Set up the analyzer.
70 OUTPUT OSa;"FA 300MHZ;FB 2GHZ;TS;"; ! /
80 OUTPUT OSa;"VAVG1O;TS;";
90 OUTPUT OSa;"VIEW TRA;VIEW TRB;";
100 OUTPUT OSa;"AVG TRB,TRA,2;"; ! Average trace A and trace B,
110 ! place result in trace B.
120 ASSIGN OSa TO * ! Close I/O path.
130 END
```
CARDLOAD

Copy Data From Memory Card to Module Memory

**Syntax**

```
CARDLOAD filename
```

**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.

! The following example shows the use of the CARDLOAD command. The example creates a DLP (downloadable program) and stores it in memory. It is then copied to the memory card, if available.

<table>
<thead>
<tr>
<th>ASSIGN $Sa TO 718</th>
<th>Assign I/O path to address 718.</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUTPUT $Sa;&quot;FUNCDEF BOX,0;&quot;;</td>
<td>Logical start of function 'BOX'.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;IP&quot;;</td>
<td>Instrument preset.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;CLRDSP&quot;;</td>
<td>Clear the analyzer screen.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PU;PA 300,300;&quot;;</td>
<td>Move pen to starting point.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PD;PR 240,0;&quot;;</td>
<td>Draw rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,-240;&quot;;</td>
<td>/ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,240;&quot;;</td>
<td>\ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PU;PA 300,300;&quot;;</td>
<td>Move pen to starting point.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PD;PR 240,0;&quot;;</td>
<td>Draw rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,-240;&quot;;</td>
<td>/ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,240;&quot;;</td>
<td>\ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PU;PA 300,300;&quot;;</td>
<td>Move pen to starting point.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PD;PR 240,0;&quot;;</td>
<td>Draw rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,-240;&quot;;</td>
<td>/ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,240;&quot;;</td>
<td>\ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PU;PA 300,300;&quot;;</td>
<td>Move pen to starting point.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PD;PR 240,0;&quot;;</td>
<td>Draw rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,-240;&quot;;</td>
<td>/ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,240;&quot;;</td>
<td>\ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PU;PA 300,300;&quot;;</td>
<td>Move pen to starting point.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PD;PR 240,0;&quot;;</td>
<td>Draw rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,-240;&quot;;</td>
<td>/ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,240;&quot;;</td>
<td>\ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PU;PA 300,300;&quot;;</td>
<td>Move pen to starting point.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PD;PR 240,0;&quot;;</td>
<td>Draw rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,-240;&quot;;</td>
<td>/ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,240;&quot;;</td>
<td>\ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PU;PA 300,300;&quot;;</td>
<td>Move pen to starting point.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PD;PR 240,0;&quot;;</td>
<td>Draw rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,-240;&quot;;</td>
<td>/ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,240;&quot;;</td>
<td>\ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PU;PA 300,300;&quot;;</td>
<td>Move pen to starting point.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PD;PR 240,0;&quot;;</td>
<td>Draw rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,-240;&quot;;</td>
<td>/ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,240;&quot;;</td>
<td>\ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PU;PA 300,300;&quot;;</td>
<td>Move pen to starting point.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PD;PR 240,0;&quot;;</td>
<td>Draw rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,-240;&quot;;</td>
<td>/ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,240;&quot;;</td>
<td>\ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PU;PA 300,300;&quot;;</td>
<td>Move pen to starting point.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PD;PR 240,0;&quot;;</td>
<td>Draw rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,-240;&quot;;</td>
<td>/ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,240;&quot;;</td>
<td>\ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PU;PA 300,300;&quot;;</td>
<td>Move pen to starting point.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PD;PR 240,0;&quot;;</td>
<td>Draw rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,-240;&quot;;</td>
<td>/ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,240;&quot;;</td>
<td>\ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PU;PA 300,300;&quot;;</td>
<td>Move pen to starting point.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PD;PR 240,0;&quot;;</td>
<td>Draw rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,-240;&quot;;</td>
<td>/ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,240;&quot;;</td>
<td>\ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PU;PA 300,300;&quot;;</td>
<td>Move pen to starting point.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PD;PR 240,0;&quot;;}</td>
<td>Draw rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,-240;&quot;;</td>
<td>/ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,240;&quot;;</td>
<td>\ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PU;PA 300,300;&quot;;</td>
<td>Move pen to starting point.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PD;PR 240,0;&quot;;</td>
<td>Draw rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,-240;&quot;;</td>
<td>/ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,240;&quot;;</td>
<td>\ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PU;PA 300,300;&quot;;</td>
<td>Move pen to starting point.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PD;PR 240,0;&quot;;</td>
<td>Draw rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,-240;&quot;;</td>
<td>/ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,240;&quot;;</td>
<td>\ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PU;PA 300,300;&quot;;</td>
<td>Move pen to starting point.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PD;PR 240,0;&quot;;</td>
<td>Draw rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,-240;&quot;;</td>
<td>/ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,240;&quot;;</td>
<td>\ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PU;PA 300,300;&quot;;</td>
<td>Move pen to starting point.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PD;PR 240,0;&quot;;</td>
<td>Draw rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,-240;&quot;;</td>
<td>/ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PR 0,240;&quot;;</td>
<td>\ rectangle.</td>
</tr>
<tr>
<td>OUTPUT $Sa;&quot;PU;PA 300,300;&quot;;</td>
<td>Move pen to starting point.</td>
</tr>
</tbody>
</table>
CARDSTORE

Copy Data to a Memory Card

**Syntax**

```
CARDSTORE  \text{delimiter}  \text{character}  \text{delimiter}  \text{filename}
```

**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.
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,0;"; ! 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;"; ! \ Draw
120 OUTPUT $Sa;" PR 0,240;"; ! / rectangle.
130 OUTPUT $Sa;" PR -240,0;"; ! / Draw
140 OUTPUT $Sa;" PR 0,-240;"; ! / rectangle.
150 OUTPUT $Sa;"Q;"; ! 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

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 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 Pile Information Parameters table for more information.

Note

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

Query Response

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

<table>
<thead>
<tr>
<th>File Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>filename.LIM</td>
<td>This file contains a limit line</td>
</tr>
<tr>
<td>filename.VAR</td>
<td>This file is a variable defined using the VARDEF, or TRDEF, or ARRAYDEF command. It is only available when MSDEV is set to MEM</td>
</tr>
<tr>
<td>filename.DLP</td>
<td>If MSDEV is set to MEM this is the downloadable program defined using the FUNCDEF command</td>
</tr>
<tr>
<td>filename.timestamp.TRC</td>
<td>This file is a trace. It can be recalled by way of the front panel or by using filename.timestamp as an argument to MODRECIT when MSDEV is set to MEM</td>
</tr>
<tr>
<td>filename.TRC :title:timestamp</td>
<td>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</td>
</tr>
</tbody>
</table>

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 XXXXXX<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$; data up to <lf-EOI> will go into Bytes_avail$.
80
90
100
110  PRINT Cat$
120  PRINT Bytes_avail$
130  ASSIGN @Sa TO *
140  END
```
CATALOG

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


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

  Dot_loc=Two_dots+Dot_loc
  END IF
  Space_loc=POS(String_parsed$," ")
  WHILE Space_loc<Dot_loc  ! Ignore spaces which are
    Temp$=String_parsed$[Space_loc+1]! part of LIMIT LINE title.
    Space_loc=Space_loc+POS(Temp$," ")
  END WHILE
  String$=String_parsed$[1,Dot_loc-1]
  WHILE NUM(String$[1,1])<32  ! Strip any special characters from
    String$=String$[2]
  END WHILE
  Ent$(Entry_name,Entry)=String$

  SELECT Mass-stg-dev$
    ! Allow for differences in string
    CASE "M"  ! of the Memory Card and Module Memory
      Ent$(Entry_length,Entry)=String_parsed$[Space_loc]
    CASE "C"
      Rev_str$=REV$(String_parsed$)
      Ent$(Entry_length,Entry)=TRIM$(REV$(Rev_str$[1,POS(Rev_str$," ")]))
    END SELECT
    Ent$(Entry_type,Entry)=TRIM$(String_parsed$[Dot_loc+1:4])
  UNTIL Last-match

  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
      ! Variable key is only displayed for a catalog of MEM. Variables
      ! cannot be stored on the card.
  650 ON KEY 8 LABEL "Trace" GOSUB List-traces
  660 DISP "Select list mode."
  670 END LOOP

  No_entries:PRINT "No Catalog Entries !!"
  Exit,cat:ASSIGN QSa TO *
  DISP "PROGRAM ENDED"
  STOP

  List-dlps:Display_cat("DLP",Ent$(*),Entry)
  RETURN
  List_all:Display_cat("ALL",Ent$(*),Entry)
  RETURN
  List_vars:Display_cat("VAR",Ent$(*),Entry)
  RETURN
  List_traces:Display_cat("TRC",Ent$(*),Entry)
  RETURN
  List_limit:Display_cat("LIM",Ent$(*),Entry)
  RETURN
  END

I-------------------------------------------------------------------

I-------------------------------------------------------------------
```

Universal_parse(Delimiter$, In_n_out$, String_parsed$, INTEGER Last-match)

A generic parsing routine.

Universal-parse: !

INTEGER Delim_loc, Len_delim
Len_delim=LEN(Delimiter$)
Delim_loc=POS(In_n_out$, Delimiter$)
IF Delim loc=0 THEN
       String-parsed$=In_n_out$
       Last_match=1
ELSE
       String-parsed$=TRIM$(In_n_out$[1, Delim_loc-1])
       In_n_out$=TRIM$(In_n_out$[Delim_loc+Len_delim])
       Last_match=0
END IF

SUBEND ! <Universal-parse>

Display_cat(Class$, Ent$(*) INTEGER Num, entries)

Displays the previously parsed catalog retrieved from the 85620A+

Display-cat:!

INTEGER Entry_name, Entry_length, Entry_type, Entry
REAL Total_length, Total_entries
!

Entry_name=FNWhich_pos("NAME")
Entry_length=FNWhich_pos("LENGTH")
Entry_type=FNWhich_pos("TYPE")
Total_length=0
Total_entries=0
!

FOR Entry=1 TO Num, entries
   IF Class$=Ent$(Entry_type, Entry) OR Class$="ALL" THEN
      SELECT Ent$(Entry_type, Entry)
      CASE "DLP"
         String$="DLP (Func)"
      CASE "VAR"
         String$="Variable"
      CASE "TRC"
         String$="Trace"
      CASE "LIM"
         String$="Limit Line"
      END SELECT
      PRINT USING Im:STRING, Entry_type, Entry, Entry_name, Entry_length, Entry_type, Entry
      GOSUB Totals
      END IF
   NEXT Entry
!

PRINT "TOTALS. \";VAL$(Total_length);\" bytes."
PRINT ";\";VAL$(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
CLRDSP

Clear Display

Syntax

CLRDSP

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 CLRDSP command.
20 !
30 ASSIGN @Sa TO 718 ! Assign I/O path to address 718.
40 !
50 OUTPUT @Sa;"CLRDSP;"; ! 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$(I)&",1;"; ! Take the LOG and
120 OUTPUT @Sa;"PU;PA 20,YVAL;"; ! multiply by scaling factor of 1.
130 OUTPUT @Sa;"TEXT %The LOG of %;"; ! Position "pen" for text.
140 OUTPUT @Sa;"VARDEF YVAL,700;"; ! Write text.
150 OUTPUT @Sa;"DSPLY"&VAL$(I)&",6,3;"; ! Write a value.
160 OUTPUT @Sa;"TEXT % = %;"; ! End loop.
170 OUTPUT @Sa;"DSPLY LOG_TEN,5,3;"; ! Calculate new "pen" position.
180 OUTPUT @Sa;"SUB YVAL,YVAL,50;"; !
190 NEXT I
200 !
210 ASSIGN @Sa TO * ! Close I/O path.
220 END
Clear Autosave/Autoexec Schedule Buffer

Syntax

```
[CLRSCHED] [integer] [ALL]
```

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,01018900000,01058900000 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 ! Remove one item from the schedule.
140 OUTPUT @Sa;"IP;CLRSCHED 2;"
150 PRINT "Press {LCL} [MODULE] (AUTOEXEC MENU) to see the updated schedule."
160 DISP "Press [CONTINUE] on the computer when ready"
170 PAUSE
180 ! Clear the entire schedule.
190 OUTPUT @Sa;"IP;CLRSCHED ALL;"
200 PRINT "Press {LCL} [MODULE] (AUTOEXEC MENU) to see the cleared schedule."
210 DISP ""
220 !
230 ASSIGN @Sa TO *
240 END
```
CNTLA

Auxiliary Control Line A

Syntax

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

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,Q;";
70 OUTPUT @Sa;" CNTLB 0;"; !SET LINE B TO 0
80 OUTPUT @Sa;" MOV S_CNTLB,0;" !UPDATE CNTLB IMAGE
90 OUTPUT @Sa;"Q;"
100 OUTPUT @Sa;"FUNCDEF AUX_SET_B,Q;";
110 OUTPUT @Sa;" CNTLB 1;"; !SET LINE B TO 1
120 OUTPUT @Sa;" MOVE S_CNTLB,1;" !UPDATE CNTLB IMAGE
130 OUTPUT@Sa;"Q;"
140 OUTPUT@Sa;"AUX_SET_B;" !GET 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**

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

**Description**

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,0;";
70  OUTPUT #Sa;"CNTLC 0;";           !SET LINE C TO 0
80  OUTPUT #Sa;"MOV S_CNTLC,0;";    !UPDATE CNTLC IMAGE
90  OUTPUT #Sa;"S;"                  !SET LINE C TO 1
100 OUTPUT #Sa;"FUNCDEF AUX_SET_C,1;";
110 OUTPUT #Sa;"CNTLC 1;";           !UPDATE CNTLC IMAGE
120 OUTPUT #Sa;"MOVE S_CNTLC,1;";
130 OUTPUT #Sa;"S;"                  !GET LINE C STATUS
140 OUTPUT #Sa;"AUX_SET_C;"         !GET 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

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 OSa TO 718 !Assign I/O path to address 718.
40 !
50 OUTPUT OSa;"VARDEF S_CNTLD,0;" !CNTLD STATE
60 OUTPUT OSa;"FUNCDEF AUX_CLR,D,0;";
70 OUTPUT OSa;"CNTLD 0;"; !SET LINE D TO 0
80 OUTPUT OSa;"MOV S_CNTLD,0;"; !UPDATE CNTLD IMAGE
90 OUTPUT OSa;"0;"
100 OUTPUT OSa;"FUNCDEF AUX_SET,D,0;";
110 OUTPUT OSa;"CNTLD 1;"; !SET LINE D TO 1
120 OUTPUT OSa;"MOVE S_CNTLD,1;"; !UPDATE CNTLD IMAGE
130 OUTPUT OSa;"0;"
140 OUTPUT OSa;"AUX_SET,D;" !SET LINE D
150 OUTPUT OSa;"S_CNTLD?;" !GET LINE D STATUS
160 ENTER OSa;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.
CNTLI

Auxiliary Control Line Input

Syntax

![Diagram]

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,O;";
40 OUTPUT @Sa;" LCLVAR I_VAL,O;";              !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;" DISPLAY I_VAL,2,0;";
90 OUTPUT @Sa;"O;"
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**

```plaintext
CTRLHPIB
```

**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 'Q' 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 QSa TO 718 ! Assign I/O path to address 718.
90 !
100 OUTPUT QSa:"FUNCDEF E,XMPL,E,Q;"; ! Logical start of the DLP.
110 OUTPUT QSa:" CTRLHPIB;"; ! Tells analyzer to take control
120 ! of the bus.
130 OUTPUT QSa:" MKN;"; ! Turn on a marker.
140 OUTPUT QSa:" OUTPUT 1,KC,'THIS IS AN EXAMPLE OF THE ANALYZER
SENDING'.'".
150 OUTPUT QSa:" OUTPUT 1,KC,'DATA TO ANOTHER HPIB DEVICE';";
160 OUTPUT QSa:" OUTPUT 1,KC,'MARKER FREQUENCY = ' ';";
170 OUTPUT QSa:" OUTPUT 1,K,MKF;";
180 OUTPUT QSa:" OUTPUT 1,K,'Hz' '';";
190 OUTPUT QSa:" OUTPUT 1,KC,"";;";
200 OUTPUT QSa:" OUTPUT 1,KC,"";;";
210 OUTPUT QSa:" RELHPIB;"; ! Release HP-IB control.
220 OUTPUT QSa:"Q;"! Logical end of the DLP.
230 |
240 OUTPUT QSa:"E,XMPL,E;"
250 ! Have the analyzer execute the
260 SEND 7;UNL TALK 18 LISTEN 1 DATA ! DLP E,XMPL,E.
270 ! Send to HP-IB select code 7 the
280 ! commands necessary-to allow
290 ! the analyzer (address 18) to
300 ! talk to the printer (address 1)
310 ! while this controller is still
320 ! connected.
330 LOCAL 7 ! Local HP-IB select code 7.
340 WAIT 2 ! Wait for printer to finish.
350 REMOTE 7 ! Toggle the HP-IB REN line.
360 LOCAL QSa ! Local the SA.
370 ASSIGN QSa TO * ! Close I/O path.
380 END
DATEMODE

Date Mode

Syntax

Description Used to set European (DDMMYY) or United States (MMDDYY) date format. This command cannot be queried.
**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 Ele 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>
<thead>
<tr>
<th>File Hierarchy</th>
<th>File Type</th>
<th>Programming Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Trace</td>
<td>DISPOSE_TRACE.961225101522.TR.C</td>
</tr>
<tr>
<td>2.</td>
<td>User-Defined Variable</td>
<td>DISPOSE_D_DATA_VAR</td>
</tr>
<tr>
<td>3.</td>
<td>DLP (Down-Loadable Program)</td>
<td>DISPOSE_F_PRCTAM.DLP</td>
</tr>
<tr>
<td>4.</td>
<td>Limit Line</td>
<td>DISPOSE_B_CDMA.LIM</td>
</tr>
</tbody>
</table>

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.
A *timedate* stamp is included as part of the filename for a trace. If there are multiple traces with the same filename, the *timedate* stamp can be used to identify each trace individually. When the DISPOSE command is used to dispose of a trace, the *timedate* stamp 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 *timedate* stamps 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 QSa TO 718 ! Assign I/O path to address 718.
70 !
80 OUTPUT QSa;"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 QSa;"FUNCDEF TST,O;"; ! Create a DLP function.
150 OUTPUT QSa;"VARDEF NUMBER,999;"; ! Create, initialize a real variable.
160 OUTPUT QSa;"ADD NUMBER,NUMBER,1;"
170 OUTPUT QSa;"Q;";
180 !
190 OUTPUT QSa;"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 QSa;"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 QSa TO * ! Close I/O path.
300 END
**DIV**

**Divide**

**Syntax**

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 QSa TO 718 ! Assign I/O path to address 718.
70 
80 OUTPUT QSa;"FUNCDEF DIVEX,Q;"; ! Logical start of the DLP function.
90 OUTPUT QSa;"TRDEF ARY,300;"; ! Create 300 element trace.
100 OUTPUT QSa;"IP; CF 300MHZ;SP 1MHZ;";

110 OUTPUT QSa;"IP; CF 300MHZ;SP 1MHZ;";

120 ! Instrument preset; set center
130 ! frequency and span.
140 OUTPUT QSa;"CLRW TRA; TS;"; ! Clear trace A and take a sweep.
150 OUTPUT QSa;"VIEW TRA;"; ! View trace A.
160 OUTPUT QSa;"MOV ARY,TRA;"; ! Move first 300 points of trace A
170 OUTPUT QSa;"DIV TRA,ARY,2;"; ! Divide user-defined trace by 2 and
180 OUTPUT QSa;"VIEW TRA;"; ! place result in trace A.
190 OUTPUT QSa;"DIVEX;"; ! View result.
200 OUTPUT QSa;"Q;"; ! Logical end of DLP function.
210 
220 OUTPUT QSa;"DIVEX;"; ! Execute function.
230 
240 ASSIGN QSa TO * ! Close I/O path.
250 

---

**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**

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: CLRDSPL, 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

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

! This program demonstrates the use of the EDITDONE command to generate a limit line:

130  ! Purge current limit line.
140  ! Begin editing limit line.
150  ! Make sure it is an absolute limit line.
160  ! Add segment to limit line.
170  ! Edit the first limit line.
180  ! Set the frequency of the segment.
190  ! Provide the upper amplitude.
200  ! Enter the lower amplitude.
210  ! Set type of segment to flat.
220  ! Enter parameters into table.
230  ! Add, then edit second segment.
240  ! Set the frequency of segment 2.
250  ! Set amplitude data for segment.
260  ! Set segment type to slope.
270  ! Enter parameters into table.
280  ! Enter parameters into table.
290  ! Enter parameters into table.
300  ! Enter parameters into table.
310  ! Enter parameters into table.
320  ! Enter parameters into table.
330  ! Enter parameters into table.
340  ! Enter parameters into table.
350  ! Enter parameters into table.
360  ! Enter parameters into table.
370  ! Enter parameters into table.
380  ! Enter parameters into table.
390  ! Enter parameters into table.
400  ! Enter parameters into table.
410  ! Enter parameters into table.
420  ! Enter parameters into table.
430  ! Enter parameters into table.
440  ! Enter parameters into table.
450  ! Enter parameters into table.
460  ! Enter parameters into table.
470  ! Enter parameters into table.
480  ! Enter parameters into table.
490  ! End of limit-line definition.
500  ! Set up analyzer to display the limit line.
505  END
EDITLIML

Edit Limit Line

Syntax

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 OUTPUT718;"LIMIPURGE;"; ! Purge current limit line.
170 OUTPUT718;"EDITLIML;"; ! Begin editing limit line.
180 OUTPUT718;"LIMIREL OFF;"; ! Make sure it is an absolute limit line.
190 OUTPUT718;"SADD;"; ! Add segment to limit line.
200 OUTPUT718;"SEDI I;"; ! Edit the first limit line.
210 OUTPUT718;"LIMF 250MHZ;"; ! Set the frequency of the segment.
220 OUTPUT718;"LIMU -35DBM;"; ! Provide the upper amplitude.
230 OUTPUT718;"LIML -80DBM;"; ! Enter the lower amplitude.
240 OUTPUT718;"LIMTFL;"; ! Set type of segment to flat.
250 OUTPUT718;"SDON;"; ! Enter parameters into table.
260 OUTPUT718;"SADD;SEDI 2;"; ! Add, then edit second segment.
270 OUTPUT718;"LIMF 290MHZ;"; ! Set the frequency of segment 2.
280 OUTPUT718;"LIMU -35DBM;LIML -80DBM;"; ! Set amplitude data for segment.
290 OUTPUT718;"LIMTSL;"; ! Set segment type to slope.
300 OUTPUT718;"SDON;"; ! Enter parameters into table.
310 OUTPUT718;"SADD;SEDI 3;";
320 OUTPUT718;"LIMF 296.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;";
330 OUTPUT718;"SDON;";
340 OUTPUT718;"SADD;SEDI 4;";
350 OUTPUT718;"LIMF 300MHZ;LIMU -5DBM;LIML -25DBM;LIMTSL;";
360 OUTPUT718;"SDON;";
370 OUTPUT718;"SADD;SEDI 5;";
380 OUTPUT718;"LIMF 303.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;";
390 OUTPUT718;"SDON;";
400 OUTPUT718;"SADD;SEDI 6;";
410 OUTPUT718;"LIMF 304.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;";
420 OUTPUT718;"SDON;";
430 OUTPUT718;"SADD;SEDI 7;";
440 OUTPUT718;"LIMF 310MHZ;LIMU -35DBM;LIML -80DBM;LIMTFL;";
450 OUTPUT718;"SDON;";
460 OUTPUT718;"SADD;SEDI 8;";
470 OUTPUT718;"LIMF 350MHZ;LIMU -35DBM;LIML -80DBM;LIMTFL;";
480 OUTPUT718;"SDON;";
490 OUTPUT718;"EDITDONE;"; ! End of limit-line definition.
500 OUTPUT718;"CF 300MHZ;SP 50MHZ;RB 2MHZ;VB 30KHZ;TS;"; ! Set up analyzer to display
505 ! the limit line.
510 END !

EDITLIML
EM

Erase User Display Memory

Syntax

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 from HP-IB

Syntax

![Syntax Diagram]

**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

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

<scaling factor>::= <variable identifier> | <numeric data field>
<destination> = 10 <source>/<scaling factor>
Program Example

10 ! The following example shows the use of the EXP command. This example calculates the power in milliwatts of the largest signal on the analyzer CRT. Initial settings are sent outside the function to set up the CAL OUTPUT signal on the CRT. These settings act as a visual aid.

60 ASSIGN QSa TO 718 ! Assign I/O path to address 718.
70 !
80 OUTPUT QSa:"IP; CF 300MHZ; SP 100MHZ;"; ! CAL OUTPUT signal to center-screen.

90 !
100 OUTPUT QSa:"FUNCDEFDBM_TO_MW,0;"; ! Logical start of DLP.
110 OUTPUT QSa:" LCLVAR PWR_MW,0;"; ! Define local variable and initialize to 0.

120 OUTPUT QSa:" TS;MKPK HI;"; ! Take sweep; marker to peak;
130 OUTPUT QSa:" EXP PWR_MW,MKA,10;"; ! pwr,mw = 10^(mka/10).
150 OUTPUT QSa:" PU;PA 100,650;PD;"; ! Move to starting position.
160 OUTPUT QSa:" TEXT %POWER (mW)=%;"; ! Write text.
170 OUTPUT QSa:" DSPLY PWR_MW,5,3;"; ! Write results.
180 OUTPUT QSa:"0;"; ! Logical end of DLP.
190 !
200 OUTPUT QSa:"DBM_TO_MW;"; ! Execute DLP.
210 !
220 ASSIGN QSa TO * ! Close I/O path.
230 END
**FUNCDEF**

**Function Definition**

**Syntax**

```
FUNCDEF user-defined function
```

**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,0;"; ! Logical start of the DLP function.
90 OUTPUT @Sa; "VARDEF FREQ,300E6;"; ! Create variable and initialize
100 OUTPUT @Sa; "IP;SP IMHZ;"; ! to 300 MHz.
110
120 OUTPUT @Sa; "REPEAT;"; ! Instrument preset.
130 OUTPUT @Sa; "MOV CF,FREQ;"; ! Begin loop.
140 OUTPUT @Sa; "TS;"; ! Set center frequency.
150 OUTPUT @Sa; "ADD FREQ,FREQ,300E6;"; ! Take a sweep to update display.
160 OUTPUT @Sa; "UNTIL FREQ,GT,3E9;"; ! Increase FRED by 300 MHz.
170 OUTPUT @Sa; "O;"; ! End of loop.
180 ! ! Logical end of DLP function.
190 |
200 OUTPUT @Sa; "INCRFREQ;"; ! Execute function.
210 |
220 ASSIGN @Sa TO * ! Close I/O path.
230 END
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).

**TEEN**  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 QSa TO 718 ! Assign I/O path to address 718.
60 !
70 OUTPUT QSa;"FUNCDEF TST,<"; ! Logical start of the DLP.
80 OUTPUT QSa;"VARDEF Y,500;"; ! Create variable and initialize to 500.
90 OUTPUT QSa;"VARDEF X,500;"; ! Create variable and initialize to 500.
100 OUTPUT QSa;"CLRDSP;"; ! Clear display.
110 OUTPUT QSa;"REPEAT;"; ! Begin loop.
120 OUTPUT QSa;" IF Y,LT,100; THEN;"; ! Test condition.
130 OUTPUT QSa;" PU;PA 100,X;"; ! Move pen.
140 OUTPUT QSa;" TEXT%DLPABORTED%;"; ! Print text.
150 OUTPUT QSa;" ABORT;";
160 OUTPUT QSa;" ELSE;";
170 OUTPUT QSa;"PU;PA 100,X;"; ! Move pen.
180 OUTPUT QSa;" DSPLY Y,5,2;"; ! Display ‘Y’.
190 OUTPUT QSa;" SUB X,X,40;"; ! Decrement pen pointer.
200 OUTPUT QSa;" SUB Y,Y,100;"; ! Decrement variable.
210 OUTPUT QSa;"ENDIF;";
220 OUTPUT QSa;"UNTIL Y,EQ,-100;"; ! End loop (The abort will occur before this condition will be satisfied.
230 !
240 !
250 OUTPUT QSa;"@;" ! Logical end of DLP.
260 !
270 OUTPUT QSa;"TST;" ! Execute DLP.
280 !
290 ASSIGN QSa TO * ! Close I/O path.
300 END
INT

Integer

Syntax

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;";
70  OUTPUT @Sa;"VARDEF INT_NUM,0;";
80  OUTPUT @Sa;"INT INT_NUM,REAL_NUM;";
90  ! Put the integer portion of the
100  OUTPUT @Sa;"REAL_NUM?;";  ! real number into INT_NUM.
110  ENTER @Sa;Real_num  ! Query REAL_NUM.
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
160  ! Print contents of variables
170  ! to computer CRT.
180  ASSIGN @Sa TO *  ! Close I/O path.
190  END
KEYCLR

Clear User Defined Keys

**Syntax**

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**

```
KEYDEF
```

**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.
Program Example

! The following example shows the use of the KEYDEF command.

ASSIGN QSa TO 718 ! Assign I/O path to address 718.

OUTPUT QSa;"KEYDEF 2, BOX, %DRAW BOX%"; ! Define softkey 2 to execute the
! DLP BOX when key is pressed.

OUTPUT QSa;"FUNCDEF BOX, 0"; ! Logical start of function.

OUTPUT QSa;"IP"; ! Instrument preset.

OUTPUT QSa;"CLRDSP"; ! Clear the analyzer screen.

OUTPUT QSa;"PU; PA 300, 300"; ! Move pen to starting point.

OUTPUT QSa;"PD; PR 240, 0"; ! \ Draw

OUTPUT QSa;"PR 0, 240"; ! / Rectangle

OUTPUT QSa;"PR -240, 0"; ! / Rectangle

OUTPUT QSa;"PR 0, -240"; ! / Rectangle

OUTPUT QSa;"Q"; ! Logical end of function.

ASSIGN QSa TO * ! Close I/O path.

END
Local Variable

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.
The following example shows the use of the LCLVAR command. This example calculates the power in milliwatts of the largest signal on the analyzer CRT. Initial settings are sent outside the function to set up the CAL OUTPUT signal on the CRT. These settings act as a visual aid.

```
ASSIGN QSa TO 718 ! Assign I/O path to address 718.
OUTPUT QSa;"IP; CF 300MHZ; SP 100MHZ;"; ! CAL OUTPUT signal to center screen.
OUTPUT QSa;"FUNCDEFDBM_TO_MW,0;"; ! Logical start of DLP.
OUTPUT QSa;"LCLVAR PWR,MW,0;"; ! Define local variable and initialize to 0.
OUTPUT QSa;"TS;MKPK HI;"; ! Take sweep; marker to peak;
OUTPUT QSa;"EXP PWR_MW,MKA,10;"; ! pwr_mw = 10^(mka/10)
OUTPUT QSa;"PU;PA 100,650;PD;"; ! Move to starting position.
OUTPUT QSa;"TEXT %POWER (mW)=%;"; ! Write text.
OUTPUT QSa;"DSPLY PWR_MW,5,3;"; ! Write results.
OUTPUT QSa;"Q;"; ! Logical end of DLP.
OUTPUT QSa;"DBM_TO_MW;"; ! Execute DLP.
ASSIGN QSa TO * ! Close I/O path.
END
```
Limit-Line Delta

Syntax

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 LIMM to define the deviation, both positive and negative, from a middle value.

Parameter
Number Range: -175 dB to 50 dB
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;";
290 OUTPUT 718;"LIMTSL;";      ! Set amplitude data for segment.
300 OUTPUT 718;"SDON;";        ! Set segment type to slope.
310 OUTPUT 718;"SADD;SEDI 3;"; ! Enter parameters into table.
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 OUTPUT 718;"CF 300MHZ;SP 50MHZ;RB 2MHZ;VB 30KHZ;TS;";    ! limit line.
510 END
LIMF

Limit-Line Frequency

Syntax

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.
This program demonstrates the use of the LIMF command to generate a limit line:

- Purge current limit line:
  `OUTPUT 718; "LIMIPURGE;";`
- Begin edit of limit line:
  `OUTPUT 718; "EDITLIML;";`
- Make sure it's an absolute limit line:
  `OUTPUT 718; "LIMIREL OFF;";`
- Add segment to the limit line:
  `OUTPUT 718; "SADD;";`
- Edit the first segment:
  `OUTPUT 718; "LIMF 250MHZ;";`
- Set frequency of the segment:
  `OUTPUT 718; "LIMU -35DBM;";`
- Enter the lower amplitude:
  `OUTPUT 718; "LIML -80DBM;";`
- Set type of segment to flat:
  `OUTPUT 718; "LIMTFL;";`
- Enter parameters into table:
  `OUTPUT 718; "SDON;";`
- Add, then edit second segment:
  `OUTPUT 718; "SADD;SEDI 2;";`
- Set frequency of segment 2:
  `OUTPUT 718; "LIMF 290MHZ;";`
- Set amplitude data for segment:
  `OUTPUT 718; "LIMTSL;";`
- Set segment type to slope:
  `OUTPUT 718; "SDON;";`
- Enter parameters into table:
  `OUTPUT 718; "SADD;SEDI 3;";`
- Add, then edit third segment:
  `OUTPUT 718; "SADD;SEDI 4;";`
- Set frequency of segment 3:
  `OUTPUT 718; "LIMF 300MHZ;";`
- Set amplitude data for segment:
  `OUTPUT 718; "LIMTSL;";`
- Set segment type to slope:
  `OUTPUT 718; "SDON;";`
- Enter parameters into table:
  `OUTPUT 718; "SADD;SEDI 5;";`
- Add, then edit fourth segment:
  `OUTPUT 718; "SADD;SEDI 6;";`
- Set frequency of segment 4:
  `OUTPUT 718; "LIMF 304.5MHZ;";`
- Set amplitude data for segment:
  `OUTPUT 718; "LIMTSL;";`
- Set segment type to slope:
  `OUTPUT 718; "SDON;";`
- Enter parameters into table:
  `OUTPUT 718; "SADD;SEDI 7;";`
- Add, then edit fifth segment:
  `OUTPUT 718; "SADD;SEDI 8;";`
- Set frequency of segment 5:
  `OUTPUT 718; "LIMF 310MHZ;";`
- Set amplitude data for segment:
  `OUTPUT 718; "LIMTFL;";`
- Set segment type to slope:
  `OUTPUT 718; "SDON;";`
- Enter parameters into table:
  `OUTPUT 718; "SADD;SEDI 9;";`
- Add, then edit sixth segment:
  `OUTPUT 718; "SADD;SEDI 10;";`
- Set frequency of segment 6:
  `OUTPUT 718; "LIMF 350MHZ;";`
- Set amplitude data for segment:
  `OUTPUT 718; "LIMTFL;";`
- Set segment type to slope:
  `OUTPUT 718; "SDON;";`
- Enter parameters into table:
  `OUTPUT 718; "SADD;SEDI 11;";`
- Add, then edit seventh segment:
  `OUTPUT 718; "SADD;SEDI 12;";`
- Set frequency of segment 7:
  `OUTPUT 718; "LIMF 354.5MHZ;";`
- Set amplitude data for segment:
  `OUTPUT 718; "LIMTFL;";`
- Set segment type to slope:
  `OUTPUT 718; "SDON;";`
- Enter parameters into table:
  `OUTPUT 718; "SADD;SEDI 13;";`
- Add, then edit eighth segment:
  `OUTPUT 718; "SADD;SEDI 14;";`
- Set frequency of segment 8:
  `OUTPUT 718; "LIMF 400MHZ;";`
- Set amplitude data for segment:
  `OUTPUT 718; "LIMTFL;";`
- Set segment type to slope:
  `OUTPUT 718; "SDON;";`
LIMIFAIL

Limit-Fail Query

Syntax

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

```
LIMIPURGE
```

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

This program demonstrates the use of the LIMIPURGE command to generate a limit line:

```
130 ! Purge current limit line.
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;LIML -80DBM;”; ! 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;”; ! Enter parameters into table.
320 OUTPUT 718; "LIMF 296.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;”; ! Add, then edit second segment.
330 OUTPUT 718; "SDON;”; ! Set segment type to slope.
340 OUTPUT 718; "LIMF 300MHZ;LIMU -5DBM;LIML -25DBM;LIMTSL;”; ! Set segment type to slope.
350 OUTPUT 718; "SADD;SEDI 4;”; ! Set segment type to slope.
360 OUTPUT 718; "SDON;”; ! Set segment type to slope.
370 OUTPUT 718; "LIMF 303.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;”; ! Set segment type to slope.
380 OUTPUT 718; "SDON;”; ! Set segment type to slope.
390 OUTPUT 718; "LIMF 304.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;”; ! Set segment type to slope.
400 OUTPUT 718; "SDON;”; ! Set segment type to slope.
410 OUTPUT 718; "SADD;SEDI 7;”; ! Set segment type to slope.
420 OUTPUT 718; "SADD;SEDI 8;”; ! Set segment type to slope.
430 OUTPUT 718; "LIMF 305MHZ;LIMU -35DBM;LIML -80DBM;LIMTFL;”; ! Set segment type to slope.
440 OUTPUT 718; "SDON;”; ! Set segment type to slope.
450 OUTPUT 718; "SADD;SEDI 9;”; ! Set segment type to slope.
460 OUTPUT 718; "SDON;”; ! Set segment type to slope.
470 OUTPUT 718; "LIMF 350MHZ;LIMU -35DBM;LIML -80DBM;LIMTFL;”; ! Set segment type to slope.
480 OUTPUT 718; "END";
```

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 END`

Programming 4-95
LIMIRCL

Recall Limit Line

Syntax

Explanation of Syntax:
- `LIMIRCL` followed by a delimiter, identifier, and another delimiter

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

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.
This program demonstrates the use of the LIMIREL command to generate a limit line:

- **OUTPUT 718;"LIMIPURGE;"** ! Purge current limit line.
- **OUTPUT 718;"EDITLIML;"** ! Begin edit of limit line.
- **OUTPUT 718;"LIMIREL ON;"** ! Make sure it's a relative limit line.
- **OUTPUT 718;"SADD;"** ! Add segment to the limit line.
- **OUTPUT 718;"SADD;SEDI 1;"** ! Edit the first segment.
- **OUTPUT 718;"LIMF 250MHZ;"** ! Set frequency of the segment.
- **OUTPUT 718;"LIMU -35DBM;"** ! Provide the upper amplitude.
- **OUTPUT 718;"LIML -80DBM;"** ! Enter the lower amplitude.
- **OUTPUT 718;"LIMF 290MHZ;"** ! Set frequency of segment 2.
- **OUTPUT 718;"LIMU -35DBM;LIML -80DBM;LIMTFL;"** ! Set type of segment to flat.
- **OUTPUT 718;"SADD;SEDI 2;"** ! Add, then edit second segment.
- **OUTPUT 718;"LIMF 290MHZ;"** ! Set frequency of segment 2.
- **OUTPUT 718;"LIMU -35DBM;LIML -80DBM;LIMTSL;"** ! Set amplitude data for segment.
- **OUTPUT 718;"SADD;SEDI 3;"** ! Set segment type to slope.
- **OUTPUT 718;"SADD;SEDI 4;"** ! Enter parameters into table.
- **OUTPUT 718;"SADD;SEDI 5;"** ! Set segment type to slope.
- **OUTPUT 718;"SADD;SEDI 6;"** ! Enter parameters into table.
- **OUTPUT 718;"SADD;SEDI 7;"** ! Set segment type to slope.
- **OUTPUT 718;"SADD;SEDI 8;"** ! Set segment type to slope.
- **OUTPUT 718;"CF 300MHZ;SP 50MHZ;RB 2MHZ;VB 30KHZ;TS;"** ! Set up analyzer to display the limit line.

**END**
LIMISAV

Limit-Line Save

Syntax

```
LIMISAV delimiter identifier delimiter
```

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.
LIMTEST

Limit-Line Test

Syntax

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

```
+-------------------+  
| LIML              |  
| number            |  
| DB                |  
| DBM               |  
| LIML              |
```

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
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;"SADD;SEDI 2;";  ! Enter parameters into table.
260  OUTPUT 718;"LIMF 290MHZ;";  ! Add, then edit second segment.
270  OUTPUT 718;"LIMU -5DBM;LIML -80DBM;";  ! Set frequency of segment 2.
280  OUTPUT 718;"LIMTSL;";      ! Set amplitude data for segment.
290  OUTPUT 718;"SADD;SEDI 3;";  ! Set segment type to slope.
300  OUTPUT 718;"SADD;SEDI 4;";  ! Enter parameters into table.
310  OUTPUT 718;"LIMF 306.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;";  ! Enter parameters into table.
320  OUTPUT 718;"SADD;SEDI 5;";  ! Enter parameters into table.
330  OUTPUT 718;"LIMF 303.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;";  ! Enter parameters into table.
340  OUTPUT 718;"SADD;SEDI 6;";  ! Enter parameters into table.
350  OUTPUT 718;"LIMF 304.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;";  ! Enter parameters into table.
360  OUTPUT 718;"SADD;SEDI 7;";  ! Enter parameters into table.
370  OUTPUT 718;"LIMF 310MHZ;LIMU -35DBM;LIML -80DBM;LIMTFL;";  ! Enter parameters into table.
380  OUTPUT 718;"SADD;SEDI 8;";  ! Enter parameters into table.
390  OUTPUT 718;"LIMF 350MHZ;LIMU -35DBM;LIML -80DBM;LIMTFL;";  ! Enter parameters into table.
400  OUTPUT 718;"SADD;SEDI 9;";  ! Enter parameters into table.
410  OUTPUT 718;"LIMF 360MHZ;LIMU -35DBM;LIML -80DBM;LIMTFL;";  ! Enter parameters into table.
420  OUTPUT 718;"SADD;SEDI 10;"; ! Enter parameters into table.
430  OUTPUT 718;"LIMF 400MHZ;LIMU -35DBM;LIML -80DBM;LIMTFL;"; ! Enter parameters into table.
440  OUTPUT 718;"SADD;SEDI 11;"; ! Enter parameters into table.
450  OUTPUT 718;"LIMF 450MHZ;LIMU -35DBM;LIML -80DBM;LIMTFL;"; ! Enter parameters into table.
460  OUTPUT 718;"SADD;SEDI 12;"; ! Enter parameters into table.
470  OUTPUT 718;"LIMF 500MHZ;LIMU -35DBM;LIML -80DBM;LIMTFL;"; ! Enter parameters into table.
480  OUTPUT 718;"SADD;SEDI 13;"; ! Enter parameters into table.
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  END  ! limit line.
LIMM

Middle Limit-Line Value

Syntax

![Diagram]

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
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;"LIMF 250MHZ;"; ! Set frequency of the segment.
250 OUTPUT 718;"LIMF 290MHZ;"; ! Set frequency of segment 2.
260 OUTPUT 718;"SDON;"; ! Enter parameters into table.
270 OUTPUT 718;"SDON;"; ! Enter parameters into table.
280 OUTPUT 718;"SDON;"; ! Set amplitude data for segment.
290 OUTPUT 718;"SADD;SEDI 2;"; ! Add, then edit second segment.
300 OUTPUT 718;"LIMF 296.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTFL;"; ! Set segment type to flat.
310 OUTPUT 718;"SDON;"; ! Enter parameters into table.
320 OUTPUT 718;"SDON;"; ! Enter parameters into table.
330 OUTPUT 718;"SDON;"; ! Enter parameters into table.
340 OUTPUT 718;"SDON;"; ! Enter parameters into table.
350 OUTPUT 718;"SDON;"; ! Enter parameters into table.
360 OUTPUT 718;"SDON;"; ! Enter parameters into table.
370 OUTPUT 718;"SDON;"; ! Enter parameters into table.
380 OUTPUT 718;"SDON;"; ! Enter parameters into table.
390 OUTPUT 718;"SDON;"; ! Enter parameters into table.
400 OUTPUT 718;"SDON;"; ! Enter parameters into table.
410 OUTPUT 718;"SDON;"; ! Enter parameters into table.
420 OUTPUT 718;"SDON;"; ! Enter parameters into table.
430 OUTPUT 718;"SDON;"; ! Enter parameters into table.
440 OUTPUT 718;"SDON;"; ! Enter parameters into table.
450 OUTPUT 718;"SDON;"; ! Enter parameters into table.
460 OUTPUT 718;"SDON;"; ! Enter parameters into table.
470 OUTPUT 718;"SDON;"; ! Enter parameters into table.
480 OUTPUT 718;"SDON;"; ! Enter parameters into table.
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 END

limit line.
LIMTFL

Flat Limit-Line Segment

Syntax

```
LIMTFL
```

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

130 ! This program demonstrates the use of the LIMTFL command
to generate a limit line:
140 !
150 !
160 OUTPUT 718;"LIMIPURGE;"; \( \text{! Purge current limit line.} \)
170 OUTPUT 718;"EDITLIML;"; \( \text{! Begin edit of limit line.} \)
180 OUTPUT 718;"LIMIREL OFF;"; \( \text{! Make sure it's an absolute limit line.} \)
190 OUTPUT 718;"SADD;"; \( \text{! Add segment to the limit line.} \)
200 OUTPUT 718;"SEDI 1;"; \( \text{! Edit the first segment.} \)
210 OUTPUT 718;"LIMF 250MHZ;"; \( \text{! Set frequency of the segment.} \)
220 OUTPUT 718;"LIMU -35DBM;"; \( \text{! Provide the upper amplitude.} \)
230 OUTPUT 718;"LIML -80DBM;"; \( \text{! Enter the lower amplitude.} \)
240 OUTPUT 718;"LIMTFL;"; \( \text{! Set type of segment to flat.} \)
250 OUTPUT 718;"SADD;SEDI 2;"; \( \text{! Add, then edit second segment.} \)
260 OUTPUT 718;"LIMF 290MHZ;"; \( \text{! Set frequency of segment 2.} \)
270 OUTPUT 718;"LIMU -35DBM;LIML -80DBM;"; \( \text{! Set ampliude data for segment.} \)
280 OUTPUT 718;"LIMTSL;"; \( \text{! Set segment type to slope.} \)
290 OUTPUT 718;"SDON;"; \( \text{! Enter parameters into table.} \)
300 OUTPUT 718;"SADD;SEDI 3;"; \( \text{! Enter parameters into table.} \)
310 OUTPUT 718;"LIMF 296.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;"; \( \text{! Set up analyzer to display the} \)
320 OUTPUT 718;"SDON;"; \( \text{! limit line.} \)
330 OUTPUT 718;"SADD;SEDI 4;"; \( \text{! Enter parameters into table.} \)
340 OUTPUT 718;"LIMF 300MHZ;LIMU -5DBM;LIML -25DBM;LIMTSL;"; \( \text{! End of limit-line definition.} \)
350 OUTPUT 718;"LIMF 303.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;"; \( \text{! Set up analyzer to display the} \)
360 OUTPUT 718;"SADD;SEDI 5;"; \( \text{! limit line.} \)
370 OUTPUT 718;"LIMF 304.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;"; \( \text{! Set up analyzer to display the} \)
380 OUTPUT 718;"SADD;SEDI 6;"; \( \text{! limit line.} \)
390 OUTPUT 718;"LIMF 310MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;"; \( \text{! End of limit-line definition.} \)
400 OUTPUT 718;"SADD;SEDI 7;"; \( \text{! Set up analyzer to display the} \)
410 OUTPUT 718;"LIMF 315MHZ;LIMU -5DBM;LIML -80DBM;LIMTFL;"; \( \text{! limit line.} \)
420 OUTPUT 718;"SADD;SEDI 8;"; \( \text{! End of limit-line definition.} \)
430 OUTPUT 718;"LIMF 350MHZ;LIMU -5DBM;LIML -80DBM;LIMTFL;"; \( \text{! Set up analyzer to display the} \)
440 OUTPUT 718;"SADD;SEDI 9;"; \( \text{! limit line.} \)
450 OUTPUT 718;"LIMF 390MHZ;LIMU -5DBM;LIML -80DBM;LIMTFL;"; \( \text{! End of limit-line definition.} \)
460 OUTPUT 718;"LIMF 400MHZ;LIMU -5DBM;LIML -80DBM;LIMTFL;"; \( \text{! End of limit-line definition.} \)
470 OUTPUT 718;"LIMF 420MHZ;LIMU -5DBM;LIML -80DBM;LIMTFL;"; \( \text{! End of limit-line definition.} \)
480 OUTPUT 718;"LIMF 450MHZ;LIMU -5DBM;LIML -80DBM;LIMTFL;"; \( \text{! End of limit-line definition.} \)
490 OUTPUT 718;"LIMF 470MHZ;LIMU -5DBM;LIML -80DBM;LIMTFL;"; \( \text{! End of limit-line definition.} \)
491 OUTPUT 718;"LIMF 490MHZ;LIMU -5DBM;LIML -80DBM;LIMTFL;"; \( \text{! End of limit-line definition.} \)
492 OUTPUT 718;"LIMF 500MHZ;LIMU -5DBM;LIML -80DBM;LIMTFL;"; \( \text{! End of limit-line definition.} \)
493 OUTPUT 718;"LIMF 520MHZ;LIMU -5DBM;LIML -80DBM;LIMTFL;"; \( \text{! End of limit-line definition.} \)
494 OUTPUT 718;"LIMF 540MHZ;LIMU -5DBM;LIML -80DBM;LIMTFL;"; \( \text{! End of limit-line definition.} \)
495 OUTPUT 718;"LIMF 560MHZ;LIMU -5DBM;LIML -80DBM;LIMTFL;"; \( \text{! End of limit-line definition.} \)
496 OUTPUT 718;"LIMF 580MHZ;LIMU -5DBM;LIML -80DBM;LIMTFL;"; \( \text{! End of limit-line definition.} \)
497 OUTPUT 718;"LIMF 600MHZ;LIMU -5DBM;LIML -80DBM;LIMTFL;"; \( \text{! End of limit-line definition.} \)
498 OUTPUT 718;"LIMF 620MHZ;LIMU -5DBM;LIML -80DBM;LIMTFL;"; \( \text{! End of limit-line definition.} \)
499 OUTPUT 718;"LIMF 640MHZ;LIMU -5DBM;LIML -80DBM;LIMTFL;"; \( \text{! End of limit-line definition.} \)
500 END
LIMTSL

Sloped Limit-Line Segment

Syntax

```
LIMTSL
```

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

130 ! This program demonstrates the use of the LIMTSL 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; "SADD;SEDI 2;"; ! Add, then edit second segment.
260 OUTPUT 718; "LIMF 290MHZ;"; ! Set frequency of segment 2.
270 OUTPUT 718; "LIMU -35DBM;LIML -80DBM;"; ! Enter parameters into table.
280 OUTPUT 718; "SADD;SEDI 3;"; ! Set amplitude data for segment.
290 OUTPUT 718; "SADD;SEDI 4;"; ! Set segment type to slope.
300 OUTPUT 718; "LIMF 300MHZ;LIMU -35DBM;LIML -80DBM;LIMTSL;"; ! Enter parameters into table.
310 OUTPUT 718; "LIMF 303.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;"; ! End of limit-line definition.
320 OUTPUT 718; "LIMF 306.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;"; ! Set up analyzer to display the
330 OUTPUT 718; "LIMF 309.5MHZ;LIMU -5DBM;LIML -80DBM;LIMTSL;"; ! limit line.
340 OUTPUT 718; "LIMF 310MHZ;LIMU -35DBM;LIML -80DBM;LIMTFL;"; OUTPUT 718; "SF 300MHZ;SP 50MHZ;RB 2MHZ;VB 30KHZ;TS;"; END
LIMU

Upper Limit-Line Value

Syntax

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
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; "LIMUFL;"; ! 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-25DBM;LIML-5DBM;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 OUTPUT 718; "CF 300MHZ;SP 50MHZ;RB 2MHZ;VB 30KHZ;TS;";
500 END
Description  The logarithm (base 10) of the source is taken, the result is multiplied by a specified scaling factor, then sent to the destination.
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 QSa TO 718                ! Assign I/O path to address 718.
60   
70   OUTPUT QSa;"CLRDSP;";          ! Clears the analyzer display.
80   
90   OUTPUT QSa;"VARDEF LOG_TEN,0;";    ! \ Initialize variables.
100  OUTPUT QSa;"VARDEF YVAL,700;";    ! /
110  
120  FOR I=100 TO 1000 STEP 100      ! Begin loop.
130   OUTPUT QSa;"LOG LOG_TEN,VAL$(I)&",1;";  ! Take the LOG and multiply
140      ! by scaling factor of 1.
150   OUTPUT QSa;"PU;PA 20,YVAL;";       ! Position "pen" for text.
160   OUTPUT QSa;"TEXT %The LOG of%;";    ! Write text.
170   OUTPUT QSa;"DSPLY "&VAL$(I)&",6,3;";    ! Write a value.
180   OUTPUT QSa;"TEXT % = %;";           ! Write text.
190   OUTPUT QSa;"DSPLY LOG_TEN,5,3;";     ! Write a value.
200  OUTPUT QSa;"SUB YVAL,YVAL,50;";   ! Calculate new "pen" position.
210  NEXT I                            ! End loop.
220  
230  ASSIGN QSa TO *                  ! Close I/O path.
240  END

Program Examples
Example #2

The following example shows another use of the LOG command. This example performs the same function as the previous example, but is fully downloadable. Note that only minor changes need to be made to convert a program from computer-dependent to computer-independent! This method is also visibly much faster for the spectrum analyzer to execute.

Assign I/O path to address 718.

Logical start of DLP.

Clears the analyzer display.

\ Initialize variables.

\ Initialize loop counter.

Begin loop.

Take the LOG and multiply by scaling factor of 1.

Position "pen" for text.

Write text.

Write a value.

Write text.

Write a value.

Calculate new "pen" position.

Increment loop counter.

End loop.

Execute the DLP.

Close I/O path.

END
**MEAN**

**Trace Mean**

**Syntax**

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,O;"; ! 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

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
Show the User Key Menu

Syntax

![MENU Diagram]

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 CSa TO 718 ! Assign I/O path to address 718.
70 |
80 OUTPUT CSa;"FUNCDEF DISP_KEYS,Q;"; ! Logical start of DLP.
90 OUTPUT CSa;"KEYDEF1,BOX,%DRAW BOX%;"; ! Create a labeled softkey.
100 OUTPUT CSa;"KEYDEF 2,DBM_TO_MW,%dBm TO mW%;"; ! Create another softkey.
110 OUTPUT CSa;"MENU 1;"; ! Display softkey menu 1.
120 OUTPUT CSa;"Q;"; ! Logical end of DLP.
130 |
140 ASSIGN CSa TO * ! Close I/O path.
150 END
```
Minimum

Syntax

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

! The following example shows a use of the MIN command. The example
! requires an input signal of 300 MHz, such as the CAL OUTPUT.

ASSIGN OSa TO 718  ! Assign I/D path to address 718.

OUTPUT OSa:"IP;";  ! Instrument preset.
OUTPUT OSa:"CF 300MHZ; SP 200KHZ;";  ! Set center frequency and span.

OUTPUT OSa:"CLRW TRA;";  ! Clear-write trace A.
OUTPUT OSa:"CLRW TRB;";  ! Clear-write trace B.
OUTPUT OSa:"TS;";  ! Update the display.
OUTPUT OSa:"VIEW TRA;";  ! View trace A.

FOR Counter=1 TO 10
  OUTPUT OSa;"APB*";  ! A + B --> A
NEXT Counter

OUTPUT OSa;"AMB ON*";  ! A - B --> A

OUTPUT OSa;"MIN TRA,TRA,TRB;";  ! Compare trace A and trace B.
OUTPUT OSa;"BLANK TRB;";  ! Blank trace B to view the results

ASSIGN OSa TO *  ! Close I/O path.
END
Module

Syntax

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.
Program Example

The following example shows the use of the MOD command. The instructions within the FUNCDEF 'Q' delimiters form a structure called a DLP (downloadable program).

ASSIGN OsSa TO 718 ! Assign I/O path to address 718.

OUTPUT OsSa;"LP;"; ! Instrument preset.

OUTPUT OsSa;"FUNCDEF M_OD_EX,Q;"; ! Logical start of the DLP.

OUTPUT OsSa;"VARDEF DIVIDEND,10;"; ! Create variable and initialize to 10.

OUTPUT OsSa;"VARDEF DIVIDOR,6;"; ! Create variable and initialize to 6.

OUTPUT OsSa;"VARDEF RESULT,0;"; ! Create variable.

OUTPUT OsSa;"MOD RESULT,DIVIDEND,DIVISOR;"; ! Perform calculation.

OUTPUT OsSa;"CLRDSP;"; ! Clear display.

OUTPUT OsSa;"PU;PA10,400;"; ! Move pen to starting position.

OUTPUT OsSa;"DSPLY DIVIDEND,4,2;TEXT % MOD %;"; ! Display calculation and results.

OUTPUT OsSa;"DSPLY DIVIDOR,2,1;";

OUTPUT OsSa;"TEXT %=%;DSPLY RESULT,2,0;"; ! / Logical end of the DLP.

OUTPUT OsSa;"O;";

OUTPUT OsSa;"M_OD.EX;"; ! Have the analyzer execute the DLP M_OD.EX.

ASSIGN OsSa TO * ! Close I/O path.

END
MODRCLT

Recall Trace from Module Memory

**Syntax**

```
MODRCLT TRA file name
```

**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 'Q' 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,Q;"; ! Logical start of the DLP.
80 OUTPUT @Sa:"IP;SNGLS;TS;VIEW TRA;"; ! Set up the analyzer.
90 OUTPUT @Sa:"MSDEV MEM;";
100 ! TELL the analyzer where the
110 OUTPUT @Sa:"MODRCLT TRA,%CAL,OUT%;"; ! Recall the trace from memory.
120 OUTPUT @Sa:"Q;";
130 OUTPUT @Sa:"GET_TRC;";
140 ! Execute the DLP.
150 ASSIGN @Sa TO * ! Close I/O path.
160 END
```

Programming 4-121
MODSAVT

Save Trace in Module Memory

**Syntax**

ModSAVT -> TRA

**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 'Q' 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;"FUNCTION SAVE_TRC,Q;";   ! Logical start of the DLP.
80 OUTPUT @Sa;"IP;CF 300MHZ;SP 1MHZ;TS;";  ! Set up the analyzer.
```

4-122 Programming
90  OUTPUT QSa:"MSDEV MEM"; ;! Tell the analyzer where the
100  trace is to be stored.
110  OUTPUT QSa:"MODSAVT TRA,%CAL_OUT%;";  ! Store the trace.
120  OUTPUT QSa:"*";  ! Logical end of the DLP.
130  OUTPUT QSa:"SAVE_TRC";  ! Execute the DLP.
140  !
150  ASSIGN QSa TO *  ! Close I/O path.
160  END
MOV

Move

Syntax

```
MOV
```

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 QSa TO 718 ! Assign I/O path to address 718.
70 !
80 OUTPUT QSa;"FUNCDEF INCRFREQ, Q;"; ! Logical start of the DLP function.
90 OUTPUT QSa;"VARDEF FREQ, 300E6;"; ! Create variable and initialize to
100 MHz.
110 
120 OUTPUT QSa;"IP; SP 1MHZ;"; ! Instrument preset; set span;
130 !
140 OUTPUT QSa;"REPEAT;"; ! Begin loop.
150 OUTPUT QSa;"MOV CF, FREQ;"; ! Set center frequency.
160 OUTPUT QSa;"TS;"; ! Take a sweep to update display.
170 OUTPUT QSa;"ADD FREQ, FREQ, 300E6;"; ! Increase FREQ by 300 MHz.
180 OUTPUT QSa;"UNTIL FREQ, GT, 3E9;"; ! End of loop.
190 OUTPUT QSa;"Q;"; ! Logical end of DLP function.
200 !
210 ASSIGN QSa TO * ! Close I/O path.
220 END
```

4.1 24 Programming
Multiply

Syntax

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.
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
35 ! this example, but acts as a visual aid.
40 !
45 ASSIGN @Sa TO 718 ! Assign I/O path to address 718.
50 !
60 OUTPUT @Sa:"FUNCDEF MPYEX,O;"; ! Logical start of the DLP function.
70 OUTPUT @Sa:"VARDEF FUNDNL,300EG;"; ! Create variable and initialize to
80 ! 300 MHz.
90 OUTPUT @Sa;"VARDEF HARMNC,0;"; ! Create variable and initialize to 0.
100 OUTPUT @Sa;"MOV HARMNC,FUNDNL;"; ! Let HARMNC equal
105 ! FUNDNL.
110 OUTPUT @Sa;"VARDEF HARMNUM,O;"; ! Initialize HARMNUM to 0.
115 OUTPUT @Sa;"IP;SP IMHZ;"; ! Instrument preset; set span.
120 !
125 OUTPUT @Sa;"REPEAT;"; ! Begin loop.
130 OUTPUT @Sa;"ADD HARMNUM,HARMNUM,1;"; ! Increment to next harmonic.
135 OUTPUT @Sa;" MPY HARMNC,FUNDNL,HARMNUM;"; ! Harm freq=fund * harm num.
140 OUTPUT @Sa;" MOV CF,HARMNC;"; ! Set center frequency.
145 OUTPUT @Sa;"TS;"; ! Take a sweep to update display.
150 OUTPUT @Sa;"UNTIL HARMNC,GE,3E9;"; ! End of loop.
155 OUTPUT @Sa;"@;"; ! Logical end of DLP function.
160 !
165 OUTPUT @Sa;"MPYEX;"; ! Execute function.
170 !
175 ASSIGN @Sa TO * ! Close I/O path.
180 !
185 END

4-1 26 Programming
MSDEV

Mass Storage Device

Syntax

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

Query Response

Program Example

! The following example shows the use of the MSDEV command. The
! instructions within the FUNCDEF ‘Q’ delimiters form a structure
! called a DLP (downloadable program).

ASSIGN @Sa TO 718 ! Assign I/O path to address 718.

OUTPUT @Sa;"FUNCDEF SAVE_TRC,0;"; ! Logical start of the DLP.
OUTPUT @Sa;"IP;CF300MHZ;SP 1MHZ;TS;"; ! Set up the analyzer.
OUTPUT @Sa;"MSDEV CARD;"; ! Tell the analyzer where the
OUTPUT @Sa;"MODSAVT TRA,%CAL-OUT%;"; ! Store the trace on the card.
OUTPUT @Sa;"@;"; ! Logical end of the DLP.
OUTPUT @Sa;"SAVE TRC;"; ! Execute the DLP.
ASSIGN @Sa TO* !Close I/O path.
END
Maximum

Syntax

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**

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

Syntax

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
Program Example

! The following example shows the use of the OR command. The example creates and displays two boxes. The function 'BOX' always draws the same box but it appears in a different area of the screen according to the position of the ORigin.

ASSIGN QSa TO 718 ! Assign I/O path to address 718.
OUTPUT QSa;"FUNCDEF DRAW_BOXES,$;"; ! Logical start of function.
OUTPUT QSa;" IP;"; ! Instrument preset.
OUTPUT QSa;" CLRDSP;"; ! Clear the analyzer screen.
OUTPUT QSa;" OR 0,0;"; ! Set origin to lower left corner of graticule.
OUTPUT QSa;" BOX;"; ! Draw box.
OUTPUT QSa;" OR 200,200;"; ! Move origin to a different spot.
OUTPUT QSa;" BOX;"; ! Draw the same box.
OUTPUT QSa;"$;"; ! Note that it appears in a different place on the analyzer.
OUTPUT QSa;":;"; ! Logical end of function.

! Logical start of function 'BOX'.
OUTPUT QSa;"FUNCDEF BOX,0;"; ! Move pen to starting point.
OUTPUT QSa;" PU;PA 200,200;"; ! \ Draw rectangle.
OUTPUT QSa;" PD;PR 240,0;"; ! /
OUTPUT QSa;" PR 0,240;"; ! Logical end of function.
OUTPUT QSa;" PR -240,0;"; ! Execute DLP.
OUTPUT QSa;" PR 0,-240;"; ! Close I/O path.
OUTPUT QSa;"P I;";
OUTPUT QSa;"DRAW-BOXES;";
ASSIGN QSa TO *
END
OUTPUT

Output to HP-IB

Syntax

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**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integers 0 to 30 are for the HPIB interface, excluding the HP-IB address of the spectrum analyzer.</td>
<td></td>
</tr>
<tr>
<td>Integers 98 and 99 are for the Auxiliary interface. *</td>
<td></td>
</tr>
<tr>
<td>98 outputs with the least-significant bit (LSB) of the serial data byte first.</td>
<td></td>
</tr>
<tr>
<td>99 outputs with the most-significant bit (MSB) of the serial data byte first.</td>
<td></td>
</tr>
</tbody>
</table>

**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

! The following example shows how to use the OUTPUT command to have
! the analyzer send data to another HP-IB device. The instructions
! within the FUNCDEF 'Q' delimiters form a structure called a
! DLP (downloadable program).

! NOTE: A printer (HP-IB address 01) needs to be connected.

ASSIGN OsA TO 718 ! Assign I/O path to address 718.

OUTPUT OsA;"FUNCDEF_EXAMPLE,Q;" ! Logical start of the DLP.
OUTPUT OsA;"CTRLHPIB;" ! Tells analyzer to take control
OUTPUT OsA;" MKN;" ! of the bus.
OUTPUT OsA;" THIS IS AN EXAMPLE OF THE ANALYZER
SENDING';";
OUTPUT OsA;" DATA TO ANOTHER HPIB DEVICE';";
OUTPUT OsA;" MARKER FREQUENCY = ' ';";
OUTPUT OsA;" OUTPUT 1,KC,'Hz';";
OUTPUT OsA;" OUTPUT 1,KC,'';";
OUTPUT OsA;" RELHPIB;" ! Release HP-IB control.
OUTPUT OsA;"Q;" ! Logical end of the DLP.

OUTPUT OsA,"E,XAMPLE;". ! Have the analyzer execute the
OUTPUT OsA," Send to HP-IB select code 7 the
SEND 7;UNL TALK 18 LISTEN 1 DATA ! commands necessary to allow
! the analyzer (address 18) to
talk to the printer (address 1)
! while this controller is still
! connected.
LOCAL 7 ! Local HP-IB select code 7.
WAIT 2 ! Wait for printer to finish.
REMOTE 7 ! Toggle the HP-IB REN line.
LOCAL OsA ! Local the SA.
ASSIGN OsA TO * ! Close I/O path.

END
Plot Absolute

Syntax

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

**Syntax**

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

**Description**

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

**Program Example**

```plaintext
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 OUTPUT @Sa:"PU;PA 300,300;"; ! Move pen to starting point.
90 OUTPUT @Sa:"PD;PR 240,0;"; ! \ Draw rectangle.
100 OUTPUT @Sa:"PR 0,240;"; ! / \ Draw rectangle.
110 OUTPUT @Sa:"PR -240,0;"; ! / \ Draw rectangle.
120 OUTPUT @Sa:"PR 0,-240;"; ! / \ Draw rectangle.
130 OUTPUT @Sa:"PR 0,240;"; ! / \ Draw rectangle.
140 !
150 OUTPUT @Sa:"PU;PA 415,305; PD;"; ! Move pen to starting point.
160 FOR I=0 TO 2*PI STEP PI/40
170 X=PROUND(10*COS(I)\3); ! \ Calculate x and y
180 Y=PROUND(10*SIN(I)\3); ! / coordinates.
190 !
200 OUTPUT @Sa;"PR \"&VAL$(X)\"\",\"&VAL$(Y)\"\";"; ! Create plot statement using calculated coordinates.
210 !
220 !
230 NEXT I
240 !
250 ASSIGN @Sa TO * ! Close I/O path.
260 END
```
PDA

Probability Distribution of Amplitude

Syntax

```
PDA
```

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 QSa TO 718 ! Assign I/O path to address 718.
70 !
80 OUTPUT QSa;"FUNCDEF TST,0;"; ! Logical start of DLP.
90 OUTPUT QSa;"IP;SNGLS;"; ! Instrument preset the analyzer.
100 OUTPUT QSa;"VB10KZ;HD;TS;"; ! Set video BW at 10 kHz.
110 OUTPUT QSa;"MOV TRB,0;"; ! Put all zeros into trace B.
120 OUTPUT QSa;"PDA TRB,TRA,.8;"; ! Determine the distribution
130 ! of trace A and put results
140 ! into trace B.
150 OUTPUT QSa;"MPY TRB,TRB,20;"; ! Multiply values in trace B
160 ! by 20.
170 OUTPUT QSa;"VIEW TRB;"; ! View trace B.
180 OUTPUT QSa;"Q;"; ! Logical end of DLP.
190 !
200 OUTPUT QSa;"TST;"; ! Execute the DLP.
210 !
220 LOCAL QSa ! Local analyzer for manual use.
230 ASSIGN QSa TO * ! Close I/O path.
240 END
**PDF**

**Probability Distribution of Frequency**

**Syntax**

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

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_PK5,0;";  ! 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;"Q;";                 ! Logical end of DLP.
170  !
180 OUTPUT @Sa;"NUM_PK5;";               ! 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

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 OUTPUT @Sa;"PU;PA 300,300;";  ! Move pen to starting point.
90 OUTPUT @Sa;"PD;PR 240,0;";   ! Move pen to starting point.
100 OUTPUT @Sa;"PR -240,0;";    ! / rectangle.
110 OUTPUT @Sa;"PR 0,-240;";    ! / rectangle.
120 OUTPUT @Sa;"PR 0,240;";     ! Draw
130 OUTPUT @Sa;"PR 0,-240;";    ! / rectangle.
140 !
150 OUTPUT @Sa;"PU;PA 415,305; PD;";  ! 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     ! Move pen to starting point.
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
220 ! calculated coordinates.
230 NEXT I
240 !
250 ASSIGN @Sa TO *              ! Close I/O path.
260 END
**PU**

**Pen Up**

**Syntax**

![PU symbol]

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

Syntax

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
The following example shows how to use the RELHPIB command to have the analyzer send data to another HP-IB device. The instructions within the FUNCDEF ‘@’ delimiters form a structure called a DLP (downloadable program).

NOTE: A printer (HP-IB address 01) needs to be connected.

ASSIGN QSa TO 718 ! Assign I/O path to address 718.

OUTPUT QSa:"FUNCDEF E,XMPL,E;";  ! Logical start of the DLP.
OUTPUT QSa:"CTRLHPIB;";  ! Tells analyzer to take control of the bus.
OUTPUT QSa:"MKN;";  ! Turn on a marker.
OUTPUT QSa:"OUTPUT 1,KC,'THIS IS AN EXAMPLE OF THE ANALYZER SENDING',".
OUTPUT QSa:"OUTPUT 1,KC,'DATA TO ANOTHER HPIB DEVICE';";
OUTPUT QSa:"OUTPUT 1,KC,'MARKER FREQUENCY = ',";
OUTPUT QSa:"OUTPUT 1,K,MKF;";
OUTPUT QSa:"OUTPUT 1,KC,'Hz';";
OUTPUT QSa:"OUTPUT 1,KC;";
OUTPUT QSa:"RELHPIB;";  ! Release HP-IB control.
OUTPUT QSa:"@;";  ! Logical end of the DLP

OUTPUT QSa:"E,XMPL,E;";  ! Have the analyzer execute the DLP E,XMPL.
SEND 7;UNL TALK 18 LISTEN 1 DATA ! Send to HP-IB select code 7 the commands necessary to allow the analyzer (address 18) to talk to the printer (address 1) while this controller is still connected.
LOCAL 7 ! Local HP-IB select code 7.
WAIT 2 ! Wait for printer to finish.
REMOTE 7 ! Toggle the HP-IB REN line.
LOCAL QSa ! Local the SA.
ASSIGN QSa TO * ! Close I/O path.
END
Repeat/Until

Syntax

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:"LP:";
90 OUTPUT @Sa:"VARDEF OUTER_LP,0";
100 OUTPUT @Sa:"VARDEF TOTAL,0";
110
120 OUTPUT @Sa:"FUNCDEF CNT_LP,Q";
130 OUTPUT @Sa:"VARDEF INNER_LP,0";
140 OUTPUT @Sa:"REPEAT";
150 OUTPUT @Sa:" ADD TOTAL,TOTAL,1";
160 OUTPUT @Sa:" UNTIL INNER_LP,INNER_LP,3";
170 OUTPUT @Sa:" UNTIL OUTER_LP,OUTER_LP,4";
180 OUTPUT @Sa:"*";
190
200 OUTPUT @Sa:" REPEAT";
210 OUTPUT @Sa:" CNT_LP";
220 OUTPUT @Sa:" ADD OUTER_LP,OUTER_LP,1";
230 OUTPUT @Sa:" UNTIL OUTER_LP,OUTER_LP,4";
240
250 OUTPUT @Sa:"CLRDSP";
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 *
340 END
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,"R;";, ! 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,"CLRDSP;";
210 OUTPUT @Sa,"PU;PA10,500;TEXT %INNER LOOP COUNTER=%;";
220 OUTPUT @Sa,"DSPLY INNER_LP,5,2;";
230 OUTPUT @Sa,"PU;PA10,400;TEXT %OUTER LOOP COUNTER=%;";
240 OUTPUT @Sa,"DSPLY OUTER_LP,5,2;";
250 OUTPUT @Sa,"PU;PA10,300;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

```
(RETURN --> (RETURN)
```

**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
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 | ASSIGN @Sa TO 718
150 |
160 |
170 OUTPUT @Sa;"FUNCDEF C_ALLER,Q;";
180 OUTPUT @Sa;"IF CF,LT,1.5E+9;THEN;"
190 OUTPUT @Sa;"ELSE;";
200 OUTPUT @Sa;"RETURN;";
210 OUTPUT @Sa;"ENDIF;"
220 OUTPUT @Sa;"ENDIF;"
230 OUTPUT @Sa;"ENDIF;"
240 OUTPUT @Sa;"ENDIF;"
250 |
260 OUTPUT @Sa;"FUNCDEF C_ALLEE,0;";
270 OUTPUT @Sa;"IF CF,LT,1.5E+9;THEN;"
280 OUTPUT @Sa;"ELSE;";
290 OUTPUT @Sa;"RETURN;";
300 OUTPUT @Sa;"ENDIF;"
310 OUTPUT @Sa;"ENDIF;"
320 OUTPUT @Sa;"ENDIF;"
330 OUTPUT @Sa;"ENDIF;"
340 OUTPUT @Sa;"ENDIF;"
350 |
360 ASSIGN @Sa TO *
370 END
RMS

Trace Root Mean Square Value

Syntax

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 0Sa TO 718 ! Assign an I/O path.
40  |
50  OUTPUT 0Sa:"IP;";  ! Instrument preset.
60  OUTPUT 0Sa:"SNGLS;";  ! Single sweep mode.
70  OUTPUT 0Sa:"TS;RMS TRA,?;";  ! Take sweep; find RMS value.
80  ENTER 0Sa;Rms_value  ! Get RMS value from analyzer.
90  |
100 Rms_value=PROUND(Rms_value,-2) ! Round the value to two decimal places.
110 |
120 PRINT "The RMS value of the trace is ";Rms_value;" measurement units"
130 |
140 ASSIGN 0Sa TO *  ! Close I/O path.
150 END
SADD

Add a Limit-Line Segment

**Syntax**

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; "EDITLIM; "; ! 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; "SEDI 2; "; ! Enter parameters into table.
260 OUTPUT 718; "LIMU -35DBM;LIML -80DBM; "; ! Set frequency of segment 2.
270 OUTPUT 718; "EDITDONE; "; ! End of limit-line definition.
280 OUTPUT 718; "CF 300MHZ;SP 50MHZ;RB 2MHZ;VB 30KHZ;TS; ";

! Set up spectrum analyzer to display the limit line.

 !
SDEL

Delete Limit-Line Segment

Syntax

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

Limit-Line Segment Done

Syntax

Description This command is used to terminate the SEDI command.
**Program Example**

130 ! This program demonstrates the use of the SDON 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;"SADD;"; ! 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;"SADD;"; ! 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 to display the limit line. 492 OUTPUT 718;"CF 300MHZ;SP 50MHZ;RB 2MHZ;VB 30KHZ;TS;"; 500 END
SEDI

Edit Limit-Line Segment

Syntax

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.
This program demonstrates the use of the SEDI command to generate a limit line:

```
OUTPUT 718; "LIMIPURGE;"; ! Purge current limit line.
OUTPUT 718; "EDITLIML;"; ! Begin edit of limit line.
OUTPUT 718; "LIMIREL OFF;"; ! Make sure it's an absolute line.
OUTPUT 718; "SADD;"; ! Add segment to the limit line.
OUTPUT 718; "SEDI 1;"; ! Edit the first segment.
OUTPUT 718; "LIMF 250MHZ;"; ! Set frequency of the segment.
OUTPUT 718; "LIMU -35DBM; LIML -80DBM;"; ! Provide the upper amplitude.
OUTPUT 718; "LIML -80DBM;"; ! Enter the lower amplitude.
OUTPUT 718; "LIMTFL;"; ! Set type of segment to flat.
OUTPUT 718; "SADD;SEDI 2;"; ! Add then edit second segment.
OUTPUT 718; "LIMF 290MHZ;"; ! Set frequency of segment 2.
OUTPUT 718; "LIMU -35DBM; LIML -80DBM;"; ! Set amplitude data for segment.
OUTPUT 718; "LIMTSL;"; ! Set segment type to slope.
OUTPUT 718; "SADD;SEDI 3;"; ! Enter parameters into table.
OUTPUT 718; "SADD;SEDI 4;"; ! Enter parameters into table.
OUTPUT 718; "LIMF 300MHZ; LIMU -35DBM; LIML -80DBM; LIMTSL;"; ! Set up to display the limit line.
OUTPUT 718; "CF 300MHZ; SP 50MHZ; RB 2MHZ; VB 30KHZ; TS;"; ! End of limit-line definition.
```

4-162 Programming
**SENTER**

Segment Enter

**Syntax**

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

\[ \text{SETDATE} \rightarrow \text{integer} \rightarrow \text{?} \]

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$&";";
 ! Set the new date.
210 !
220 !
230 Translate-month: !
240 ! Converts the month from alpha characters to 'numerics'. The numbers
250 ! are left as strings so that they can be concatenated later.
260 !
270 SELECT Mon$ 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"
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

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

```
SHOWMENU
```

Description

This command displays labels in the softkey area on the display. The user specifies the 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.

softkey
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
61 ! 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 QSa TO 718 ! Assign I/O path to address 718.
110 OUTPUT QSa;"FUNCDEF KEYS,$;"; ! Logical start of function 'KEYS'.
120 OUTPUT QSa;"SKYCLR;"; ! Clear softkeys.
130 OUTPUT QSa;"SKYDEF 1,BOX,%DRAW BOX%;"; ! Define softkey 1 to execute
140 ! function BOX. Label key with
150 ! DRAW BOX.
160 OUTPUT QSa;"SHOWMENU;"; ! Display the SKYDEF'ed label.
170 OUTPUT QSa;"$;"; ! Logical end of function.
180 |
190 OUTPUT QSa;"FUNCDEF BOX,@;"; ! Logical start of function 'BOX'.
200 OUTPUT QSa;"IP;"; ! Instrument preset.
210 OUTPUT QSa;"CLRDS;"; ! Clear the analyzer screen.
220 |
230 OUTPUT QSa;"PU;PA 300,300;"; ! Move pen to starting point.
240 OUTPUT QSa;"PD;PR 240,0;"; ! \ Draw
250 OUTPUT QSa;"PR 0,240;"; ! \ Draw
260 OUTPUT QSa;"PR -240,0;"; ! / rectangle.
270 OUTPUT QSa;"PR 0,-240;"; ! /
280 OUTPUT QSa;"$;"; ! Logical end of function.
290 |
300 ASSIGN QSa TO * ! Close I/O path.
310 END
SKYCLR

Clear User Softkeys

Syntax

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.
Program Example

The following example shows the use of the SKYCLR command. The example creates and displays a softkey which can be used to execute a DLP (downloadable program).

Note: Since SHOWMENU is not executable over the HP-IB, the function KEYS needs to be executed from the front panel of the analyzer after it is loaded. Press {LCL} [MODULE] {AUTOEXEC MENU} {EDIT AUTOEXEC} (CHOOSE DLP).

Scroll knob to 'KEYS' and press (EXECUTE NOW).

ASSIGN QSa TO 718  ! Assign I/O path to address 718.
OUTPUT QSa;"FUNCDEF KEYS,$;";  ! Logical start of function 'KEYS'.
OUTPUT QSa;"SKYCLR;";  ! Clear softkeys.
OUTPUT QSa;"SKYDEF BOX,%,DRAW BOX%;";  ! Define softkey 1 to execute function BOX. Label key with DRAW BOX.
OUTPUT QSa;"SHOWMENU;";  ! Display the SKYDEF'ed label.
OUTPUT QSa;"$;";  ! Logical end of function.

OUTPUT QSa;"FUNCDEF BOX,%;";  ! Logical start of function 'BOX'.
OUTPUT QSa;" IP;";  ! Instrument preset.
OUTPUT QSa;" CLRDSP;";  ! Clear the analyzer screen.
OUTPUT QSa;" PU;PA 300,300;";  ! Move pen to starting point.
OUTPUT QSa;" PD;PR 240,0;";  ! \ Draw
OUTPUT QSa;" PR 0,240;";  ! \ Draw rectangle.
OUTPUT QSa;" PR -240,0;";  ! / Draw rectangle.
OUTPUT QSa;" PR 0,-240;";  ! / Draw rectangle.
OUTPUT QSa;" $;";  ! Logical end of function.
ASSIGN QSa TO *  ! Close I/O path.
END
SKYDEF

Define a User Keys Menu Softkey

Syntax

```
SKYDEF i-key number
```

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.
Program Example

! The following example shows the use of the SKYDEF command. The example creates and displays a softkey which can be used to execute a DLP (downloadable program).

! Note: Since SHOWMENU is not executable over HP-IB, the function KEYS needs to be executed from the front panel of the analyzer after it is loaded. Press {LCL} [MODULE] {AUTOEXEC MENU} (EDIT AUTOEXEC) {CHOOSE DLP}.

! Scroll knob to 'KEYS' and press {EXECUTE NOW}.

ASSIGN QSa TO 718 ! Assign I/O path to address 718.
OUTPUT QSa;"FUNCDEF KEYS,$;"; ! Logical start of function 'KEYS'.
OUTPUT QSa;"SKYCLR;"; ! Clear softkeys.
OUTPUT QSa;"SKYDEF1, BOX,%DRAW BOX%;"; ! Define softkey 1 to execute function BOX. Label key with DRAW BOX.
OUTPUT QSa;"SHOWMENU;"; ! Display the SKYDEF'ed label.
OUTPUT QSa;"$;"; ! Logical end of function.

OUTPUT QSa;"FUNCDEF BOX,O;"; ! Logical start of function 'BOX'.
OUTPUT QSa;"IP;"; ! Instrument preset.
OUTPUT QSa;"CLRDSP;"; ! Clear the analyzer screen.

OUTPUT QSa;"PU;PA 300,300;"; ! Move pen to starting point.
OUTPUT QSa;"PD;PR 240,0;"; ! \ Draw
OUTPUT QSa;"PR 0,240;"; ! \ Draw
OUTPUT QSa;"PR -240,0;"; ! / rectangle.
OUTPUT QSa;"PR 0,-240;"; ! /
OUTPUT QSa;" Q;"; ! Logical end of function.

ASSIGN QSa TO * ! Close I/O path.
END
**SMOOTH**

**Smooth Trace**

**Syntax**

```
SMOOTH
```

**Description**
This command smoothes 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.
**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 SMOOTH_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 View mode.
110
120 !
130 OUTPUT @Sa:"SMOOTH TRA,5;"; ! Smooth the trace with a five-point running average.
140
150 !
160 OUTPUT @Sa:"@;"; ! Logical end of the DLP.
170 !
180 OUTPUT @Sa:"S_SMOOTH_EX;"; ! Have the analyzer execute the DLP S_SMOOTH_EX
190
200 ASSIGN @Sa TO * ! Close I/O path.
SQR

Square Root

Syntax

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

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

```
! The following example shows the use of the STDEV function to return
! the standard deviation of the trace to the controller. The example
! uses an input signal of 300 MHz such as the CAL OUTPUT.

INTEGER Mean-value ! Define an INTEGER variable.
ASSIGN QSa TO 718 ! Assign I/O path to address 718.

OUTPUT QSa;"IP;CF 300MHZ;SP 1MHZ;TS;"; ! Set up the analyzer.

OUTPUT QSa;"MEAN TRA,?;"; ! Determine the mean of trace A.
ENTER QSa;Mean_value ! Read value from analyzer.
PRINT "MEAN of trace A in measurement units = ";Mean_value

OUTPUT QSa;"STDEV TRA,?;";
ENTER QSa;Stdev_value
PRINT "STANDARD DEVIATION of trace A in measurement units = ";Stdev_value

ASSIGN QSa TO * ! Close I/O path.
```

END

4-176 Programming
**Description**  Subtract the value of source 1 from source 2 and send the results to the destination.
Program Example

! The following example shows the use of the SUB command. The example
! uses an input signal of 300 MHz, such as the CAL OUTPUT signal, and
! looks at its harmonics. An input signal is not necessary to the
! function of this example, but acts as a visual aid.

ASSIGN QSa TO 718 ! Assign I/O path to address 718.

OUTPUT QSa;"FUNCDEF SUB_EX,Q;"; ! Logical start of the DLP function.
OUTPUT QSa;"VARDEF FREQ,3E9;"; ! Create variable and initialize to 3 GHz.
OUTPUT QSa;"IP;SP 1MHZ;"; ! Instrument preset; set span.

OUTPUT QSa;"REPEAT;"; ! Begin loop.
OUTPUT QSa;"MOV CF,FREQ;"; ! Set center frequency.
OUTPUT QSa;"TS;"; ! Take a sweep to update display.
OUTPUT QSa;"SUB FREQ,FREQ,300E6;";! Decrease FREQ by 300 MHz.
OUTPUT QSa;"UNTIL FREQ,LT,300E6;"; ! End of loop.
OUTPUT QSa;"Q;"; ! Logical end of DLP function.

OUTPUT QSa;"SUB_EX;"; ! Execute function.

ASSIGN QSa TO * ! Close I/O path.
END
SUM

Sum of Trace Amplitudes

**Syntax**

```
SUM [ #, ] TRA, TRB, ...
```

**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 QSa TO 718 ! Assign I/O path to address 718.
50
60 OUTPUT QSa;"IP;"; ! Instrument preset.
70 OUTPUT QSa;"CF 300MZ; SP 1MZ;"; ! \n
80 OUTPUT QSa;";SNGLS;TS;";
90 OUTPUT QSa;"MKPK HI;MKRL;TS;HD;"; ! /
100
110 OUTPUT QSa;"SUM TRA,?;";
120 ENTER QSa;Sum_trace ! Returns sum in measurement units.
130 PRINT "The sum of all the trace points equals ";Sum_trace
140
150 ASSIGN QSa TO * ! Close I/O path.
160 END
```
SUMSQR

Sum of Squared Trace Amplitudes

Syntax

```
Syntax
```

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   ASSIGN @Sa TO 718        ! Assign I/O path to address 718.
20   ! The following example shows the use of the SUMSQR command. The example uses an input signal of 300 MHz such as the CAL OUTPUT.
30   !
40   OUTPUT @Sa;"IP;";        ! Instrument preset.
50   !
60   OUTPUT @Sa;"CF 300MZ; SP 1MZ;";   ! Set up the analyzer.
70   OUTPUT @Sa;"RB 30KHZ;";   !
80   OUTPUT @Sa;"TS;MKPK HI;MKRL;TS;HD;";   !
90   !
100  OUTPUT @Sa;"SUMSQRTTRA,?;";   ! Query the analyzer.
110  ENTER @Sa;Sum_sqr         ! Get the value from the analyzer.
120  PRINT "The Sum of the Squares of the trace points = ";Sum_sqr;
130  PRINT "measurement units."
140  !
150  !
160  ASSIGN @Sa TO *          ! Close I/O path.
170  END
```
The following example shows the use of the REPEAT/UNTIL commands in a FOR/NEXT loop context. This is an immediate-execute DLP (downloadable program). This is another way of nesting REPEATS.

ASSIGN @Sa TO 718  ! Assign I/O path to address 718.

OUTPUT @Sa;"IP;";  ! Instrument preset.
OUTPUT @Sa;"VARDEF OUTER_LP,O;";  ! Create variable and initialize to 0.
OUTPUT @Sa;"VARDEF TOTAL,O;";  ! Create variable and initialize to 0.

REPEAT;  ! Begin outer loop.
  VARDEF INNER_LP,O;  ! Create and initialize inner loop counter.
  REPEAT;  ! Begin inner loop.
    OUTPUT @Sa;"ADD TOTAL,TOTAL,l;";  ! Increment running total.
    OUTPUT @Sa;"ADD INNER_LP,INNER_LP,l;";  ! Increment inner loop counter.
  UNTIL INNER_LP,INNER_LP,EQ,3;  ! End of inner loop.
  ADD OUTER_LP,INNER_LP,1;";  ! Increment outer loop counter.
UNTIL OUTER_LP,EQ,4;";  ! End of outer loop.

OUTPUT @Sa;"CLRDSP;";
OUTPUT @Sa;"PU;PA 10,500;TEXT %INNER LOOP COUNTER=%;";
OUTPUT @Sa;"DSPLY INNER_LP,5,2;";
OUTPUT @Sa;"PU;PA 10,400;TEXT %OUTER LOOP COUNTER=%;";
OUTPUT @Sa;"DSPLY OUTER_LP,5,2;";
OUTPUT @Sa;"PU;PA 10,300;TEXT %TOTAL ITERATIONS=%;";
OUTPUT @Sa;"DSPLY TOTAL,5,2;";

ASSIGN @Sa TO *  ! Close I/O path.
END

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.

Syntax
```
TEXT [delimiter] [character] [delimiter]
```

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,O;";           ! Create variable and initialize to 0.
90 OUTPUT @Sa;"VARDEF TOTAL,O;";             ! Create variable and initialize to 0.
100
110 OUTPUT @Sa;"REPEAT;";                    ! Begin outer loop.
120 OUTPUT @Sa;"VARDEF INNER_LP,O;";         ! 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;"CLRDSP;";
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

Program 4-18
**TIMEDATE**

**Time Date**

**Syntax**

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

**Description**

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 O Sa TO 718 ! Assign I/O path to address 718.
40 OUTPUT O Sa;"IP;"; ! Instrument preset.
50 DISP "Activate 856X Time and Date Display"
60 OUTPUT O 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 O Sa;"TIMEDATE OFF;"
120 |
130 ASSIGN O Sa TO * ! Close I/O path.
140 END
```
TRDEF

Trace Definition

Syntax

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
The following example shows the use of the TRDEF command. The example creates a sine wave in the computer and sends it to trace A of the analyzer. It is then moved to the user-defined trace where it can be manipulated to fit the user's needs.

ASSIGN 0Sa TO 718 ! Assign I/O path to address 718.
DIM Y_axis(1:601) ! Create an array in the computer.
RAD ! Angles measured in radians.
GINIT ! Graphics Init (computer).
GRAPHICS ON ! Graphics On (computer).
WINDOW 1,601,1,601 ! Set up the calculator screen to have the same number of points as the spectrum analyzer.
MOVE 0,SIN(0) ! Move the pen (computer).
X=0 ! Create the array in the computer.
FOR Angle=0 TO 20*PI STEP (2*PI)/60
  X=X+1 ! Increment array pointer.
  Y=SIN(Angle)*120+300 ! 120 gives +/- 2 divs of amplitude.
  300 offsets to mid-screen
  PLOT X,Y ! Draw sine wave on computer screen.
  Y_axis(X)=Y ! Load the array.
NEXT Angle
!
! Set up the analyzer to receive the trace array from the computer.
!
OUTPUT 0Sa;"IP;SNGLS;TS;"; ! Instrument preset. Single sweep so that the trace is not over-written by another sweep.
OUTPUT 0Sa;"VIEW TRA;TDF A;"; ! View trace A. Set to 'A' format.
OUTPUT 0Sa USING ",K,W,601(W);"TRA#A",1202,Y_axis(*)",";"; ! Send the trace.
!
OUTPUT 0Sa;"TRDEF SINE,601;"; ! Create a user-defined trace that is the same size as a default screen trace.
OUTPUT 0Sa;"MOV SINE,TRA;"; ! Copy trace A to the user-defined trace.
OUTPUT 0Sa;"ADD SINE,SINE,60;"; ! Add 60 to each point in SINE.
OUTPUT 0Sa;"MOV TRB,SINE;"; ! Copy the modified SINE trace to trace B.
OUTPUT 0Sa;"VIEW TRA; VIEW TRB;"; ! View both traces.
!
! Note that 60 offsets the sine wave one division from the original trace.
!
ASSIGN 0Sa TO * ! Close I/O path.
END

4-184 Programming
VARDEF

Variable Definition

Syntax

\[
\text{VARDEF} \to \text{variable name} \to \text{user-defined variable} \to \text{number} \to \text{<initial value> ::= <real>}
\]

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

! The following example shows the use of the VARDEF command. The example uses an input signal of 300 MHz, such as the CAL OUTPUT signal and looks at its harmonics. An input signal is not necessary to the function of this example, but acts as a visual aid.

10 ASSIGN @Sa TO 718 ! Assign I/O path to address 718.
70 !
80 OUTPUT @Sa;"FUNCDEF INCRFREQ,0;"; ! Logical start of the DLP function.
90 OUTPUT @Sa;"VARDEF FREQ,300E6;"; ! Create variable and initialize to 300 MHz.
100 OUTPUT @Sa;"IP;SP IMHZ;"; ! 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

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 QSa TO 718 ! Assign I/O path to -address 718.
50 !
60 OUTPUT QSa:"TR:SMRLS;" ! Instrument preset.
70 OUTPUT QSa:"CF 300 MZ;SP 1KHZ;" ! \ Set up analyzer.
80 OUTPUT QSa:"RB 300HZ;TS;" ! /
90 !
100 OUTPUT QSa:"VARIANCE TRA,?;" ! Query the trace variance.
110 ENTER QSa;Variance ! Get the value from the analyzer.
120 PRINT "The variance of trace A is ";Variance
130 !
140 ASSIGN QSa 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.

Figure 5-1. Static-Safe Workstation
Static-Safe Accessories

### Table 5-1. Static-Safe Accessories

<table>
<thead>
<tr>
<th>HP Part Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9300-0797</td>
<td>Set includes: 3M static control mat 0.6 m x 1.2 m (2 ft x 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.)</td>
</tr>
<tr>
<td>9300-0980</td>
<td>Wrist-strap cord 1.5 m (5 ft)</td>
</tr>
<tr>
<td>9300-1383</td>
<td>Wrist-strap, color black, stainless steel, without cord, has four adjustable links and a 7 mm post-type connection.</td>
</tr>
<tr>
<td>9300-1169</td>
<td>ESD heel-strap (reusable 6 to 12 months).</td>
</tr>
</tbody>
</table>

## 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.

![Figure 5-2. A1TP1 Configuration](image)
Replacement Procedures

Introduction
Replacement procedures provide specific disassembly and assembly information. Refer to the appropriate replacement procedures for the assembly you are replacing.
- Al Memory Board Replacement
- Bl Module Battery Replacement
- Memory Card Battery Replacement

Al 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.

Bl Module Battery Replacement

1. Refer to the Al Memory Board Assembly Replacement procedure to remove the Al 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.

Figure 5-3. A1 Board Assembly & Battery Replacement (Serial Prefix <3143A)
Figure 5-4.
AI Board Assembly & Battery Replacement (Serial Prefix ≥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.)

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 ±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 ±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

<table>
<thead>
<tr>
<th>Mfr. Code</th>
<th>Manufacturer Name</th>
<th>Address</th>
<th>ZIP Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1433</td>
<td>ABELEKTRONIK GMBH</td>
<td>SALZBURG, AU</td>
<td>A-501</td>
</tr>
<tr>
<td>K7253</td>
<td>STD/STANTEL</td>
<td>DEVON, EG</td>
<td></td>
</tr>
<tr>
<td>S4013</td>
<td>HITACHI AMERICA LTD</td>
<td>SUNNYVALE, CA, US</td>
<td>94086</td>
</tr>
<tr>
<td>00779</td>
<td>AMP INC</td>
<td>HARRISBURG, PA, US</td>
<td>17111</td>
</tr>
<tr>
<td>08709</td>
<td>EPSON AMERICA</td>
<td>TORRANCE, CA, US</td>
<td>75265</td>
</tr>
<tr>
<td>01295</td>
<td>TEXAS INSTRUMENTS INC</td>
<td>DALLAS, TX, US</td>
<td>90505</td>
</tr>
<tr>
<td>04222</td>
<td>AVX CORP</td>
<td>GREAT NECK, NY, US</td>
<td>11021</td>
</tr>
<tr>
<td>10421</td>
<td>PANASONIC, INC</td>
<td>SECAUCUS, NJ, US</td>
<td>07094</td>
</tr>
<tr>
<td>06383</td>
<td>PANDUIT CORP</td>
<td>TINLEY PARK, IL, US</td>
<td>69477</td>
</tr>
<tr>
<td>06424</td>
<td>SPERRY RAND µ-WAVE ELEC DIV</td>
<td>CLEARWATER, FL, US</td>
<td>33518</td>
</tr>
<tr>
<td>10899</td>
<td>EASTERN AIR DEVICES INC</td>
<td>GREAT NECK, NY, US</td>
<td>11021</td>
</tr>
<tr>
<td>18873</td>
<td>DUPONT E I DE NEMOURS &amp; CO</td>
<td>WILMINGTON, DE, US</td>
<td>19801</td>
</tr>
<tr>
<td>2M627</td>
<td>ROHM CORP</td>
<td>IRVINE, CA, US</td>
<td>92713</td>
</tr>
<tr>
<td>28480</td>
<td>HEWLETT-PACKARD CO CORP HQ</td>
<td>PALO ALTO, CA, US</td>
<td>94304</td>
</tr>
<tr>
<td>34335</td>
<td>ADVANCED MICRO DEVICES INC</td>
<td>SUNNYVALE, CA, US</td>
<td>94086</td>
</tr>
<tr>
<td>71468</td>
<td>ITT CORP</td>
<td>NEW YORK, NY, US</td>
<td>10022</td>
</tr>
<tr>
<td>72794</td>
<td>DZUS FASTENER CO INC</td>
<td>WEST ISLIP, NY, US</td>
<td>11795</td>
</tr>
</tbody>
</table>
Figure 5-6. HP 85620A Parts Identification (Serial Prefix <3143A)
Figure 5-7. HP 85620A Parts Identification (Serial Prefix ≥3143A)
<table>
<thead>
<tr>
<th>Item</th>
<th>BP Part Number</th>
<th>c d</th>
<th>Qty</th>
<th>Description</th>
<th>Mfg Code</th>
<th>Mfg Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0515-1236</td>
<td>9</td>
<td>4</td>
<td>SCREW-MACH M3 X 0.514MM-LG</td>
<td>28480</td>
<td>0515-1236</td>
</tr>
<tr>
<td>2</td>
<td>5021-9304</td>
<td>6</td>
<td>1</td>
<td>COV-BTM TEST</td>
<td>28480</td>
<td>5021-9304</td>
</tr>
<tr>
<td>3</td>
<td>8160-0448</td>
<td>8</td>
<td>1</td>
<td>RFI ROUND STRIP BE-CU SN-PL .062-IN-OD</td>
<td>10899</td>
<td>SS-04</td>
</tr>
<tr>
<td>4</td>
<td>8160-0650</td>
<td>4</td>
<td>1</td>
<td>GASKET-OPTION MODULE CONNECTOR</td>
<td>28480</td>
<td>8160-0650</td>
</tr>
<tr>
<td>5</td>
<td>85620-40002</td>
<td>0</td>
<td>1</td>
<td>BEZEL MEMORY CARD SLOT</td>
<td>28480</td>
<td>85620-40002</td>
</tr>
<tr>
<td>6</td>
<td>85620-60001†</td>
<td>1</td>
<td>1</td>
<td>BD AY-PORT MEM</td>
<td>28480</td>
<td>85620-60001</td>
</tr>
<tr>
<td>7</td>
<td>85620-60006‡</td>
<td>6</td>
<td>1</td>
<td>BD AY-PORT MEM</td>
<td>28480</td>
<td>85620-60006</td>
</tr>
<tr>
<td>8</td>
<td>85620-60008§</td>
<td>8</td>
<td>1</td>
<td>BD AY-PORT MEM</td>
<td>28480</td>
<td>85620-60008</td>
</tr>
<tr>
<td>9</td>
<td>85620-60014</td>
<td></td>
<td>6</td>
<td>1</td>
<td>BD AY-PORT MEM</td>
<td>28480</td>
</tr>
<tr>
<td>7</td>
<td>0515-0894</td>
<td>3</td>
<td>8</td>
<td>SCREW-MACH M2.5 X 0.456MM-LG PAN-HD</td>
<td>28480</td>
<td>0515-0894</td>
</tr>
<tr>
<td>8</td>
<td>1420-0370‡</td>
<td>0</td>
<td>1</td>
<td>BATTERY, 2.8 V 1.0AH</td>
<td>28480</td>
<td>1420-0370</td>
</tr>
<tr>
<td>9</td>
<td>1420-0341‡</td>
<td></td>
<td>1</td>
<td>1</td>
<td>BATTERY, 3 V 1.2AH</td>
<td>28480</td>
</tr>
<tr>
<td>8</td>
<td>1420-0394§</td>
<td>1</td>
<td>1</td>
<td>BATTERY, 3 V 1.OAH</td>
<td>08709</td>
<td>CR2477-1HF</td>
</tr>
<tr>
<td>9</td>
<td>85620-40001#</td>
<td>9</td>
<td>1</td>
<td>BRACE</td>
<td>28480</td>
<td>85620-40001</td>
</tr>
<tr>
<td>9</td>
<td>5041-8980§**</td>
<td>4</td>
<td>1</td>
<td>BRACE</td>
<td>28480</td>
<td>5041-8980</td>
</tr>
<tr>
<td>10</td>
<td>85620-20005†</td>
<td>1</td>
<td>1</td>
<td>COVER-TOP</td>
<td>28480</td>
<td>85620-20005</td>
</tr>
<tr>
<td>10</td>
<td>5021-9303§</td>
<td></td>
<td>5</td>
<td>1</td>
<td>COVER-TOP</td>
<td>28480</td>
</tr>
<tr>
<td>10</td>
<td>5022-0002‡</td>
<td>5</td>
<td>1</td>
<td>COVER-TOP</td>
<td>28480</td>
<td>5022-0002</td>
</tr>
<tr>
<td>11</td>
<td>1390-0750</td>
<td>5</td>
<td>1</td>
<td>FASTENER-1/4-TURN RCPT</td>
<td>72794</td>
<td>A3.5T127B1100-Z</td>
</tr>
<tr>
<td>12</td>
<td>5041-7247</td>
<td>4</td>
<td>1</td>
<td>SPACER</td>
<td>28480</td>
<td>5041-7247</td>
</tr>
<tr>
<td>85620-80003</td>
<td>5</td>
<td>1</td>
<td>LABEL SET</td>
<td>28480</td>
<td>85620-80003</td>
<td></td>
</tr>
</tbody>
</table>

† 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 above
§ Serial prefix 3143A and below
* Serial prefix 3143A
<table>
<thead>
<tr>
<th>HP 85620A Firmware Revision/Date Code</th>
<th>ROM Fart Number*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>U1</td>
</tr>
<tr>
<td>A/890214</td>
<td>85620-80005</td>
</tr>
<tr>
<td>B/890524</td>
<td>85620-80014</td>
</tr>
<tr>
<td>C/910116†</td>
<td>85620-80023</td>
</tr>
<tr>
<td></td>
<td><strong>U7 ROM Fart Number†</strong></td>
</tr>
<tr>
<td>C/910116†</td>
<td>85620-80021</td>
</tr>
</tbody>
</table>

* Serial prefix 3003A and below.
† The HP part number for the current firmware revision label is 85620-80003.
‡ Serial prefix 0143A and above.

See Firmware Note, HP part number 5962-0452, for descriptions of the firmware revisions.
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>
<thead>
<tr>
<th>Board Assembly</th>
<th>Instrument Serial Prefix</th>
<th>Assembly Part Number</th>
<th>CLIP Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass Memory Assembly</td>
<td>2929A and below</td>
<td>85620-6001</td>
<td>85620-90033</td>
</tr>
<tr>
<td>Mass Memory Assembly</td>
<td>3003A</td>
<td>85620-60008</td>
<td>85620-90034</td>
</tr>
<tr>
<td>Mass Memory Assembly</td>
<td>Standard only (3143A)</td>
<td>85620-60006</td>
<td>85620-90035</td>
</tr>
<tr>
<td>Mass Memory Assembly</td>
<td>3526A and above</td>
<td>85620-60018</td>
<td>85620-90038</td>
</tr>
<tr>
<td>Mass Memory Assembly</td>
<td>Option TO1 only 3143A and above</td>
<td>85620-60014</td>
<td>Contact HP (MID)</td>
</tr>
</tbody>
</table>
The information contained in this document is subject to change without notice.

Agilent Technologies makes no warranty of any kind with regard to this material, including but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Agilent Technologies shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.
## Agilent Part Number 85620-60001
### A1 Mass Memory Assembly

<table>
<thead>
<tr>
<th>Reference Designator</th>
<th>Agilent Part Number</th>
<th>Qty</th>
<th>Description</th>
<th>Mfr Code</th>
<th>Mfr Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>BT1</td>
<td>1420-0307</td>
<td>1</td>
<td>BATTERY 7.2V .45A-IR NL-CD FLAT</td>
<td>28480</td>
<td>1420-0307</td>
</tr>
<tr>
<td>C1</td>
<td>0180-0374</td>
<td>1</td>
<td>CAP-FXD 10uF +.10% 20 V TA</td>
<td>28480</td>
<td>0180-0374</td>
</tr>
<tr>
<td>C2</td>
<td>0180-0229</td>
<td>1</td>
<td>CAP-FXD 33uF +.10% 10 V TA</td>
<td>28480</td>
<td>0180-0229</td>
</tr>
<tr>
<td>C3</td>
<td>0160-4832</td>
<td>6</td>
<td>CAP-FXD 0.01µF +.10% 100 V CER X7R</td>
<td>28480</td>
<td>0160-4832</td>
</tr>
<tr>
<td>C4</td>
<td>0160-4832</td>
<td>1</td>
<td>CAP-FXD 0.01µF +.10% 100 V CER X7R</td>
<td>28480</td>
<td>0160-4832</td>
</tr>
<tr>
<td>C5</td>
<td>0160-4832</td>
<td>1</td>
<td>CAP-FXD 0.01µF +.10% 100 V CER X7R</td>
<td>28480</td>
<td>0160-4832</td>
</tr>
<tr>
<td>C6</td>
<td>0160-4832</td>
<td>1</td>
<td>CAP-FXD 0.01µF +.10% 100 V CER X7R</td>
<td>28480</td>
<td>0160-4832</td>
</tr>
<tr>
<td>C7</td>
<td>0160-4832</td>
<td>1</td>
<td>CAP-FXD 0.01µF +.10% 100 V CER X7R</td>
<td>28480</td>
<td>0160-4832</td>
</tr>
<tr>
<td>C8</td>
<td>0160-4832</td>
<td>1</td>
<td>CAP-FXD 0.01µF +.10% 100 V CER X7R</td>
<td>28480</td>
<td>0160-4832</td>
</tr>
<tr>
<td>C9</td>
<td>0160-5008</td>
<td>1</td>
<td>CAP-FXD 0.22µF +.10% 50 V CER X7R</td>
<td>28480</td>
<td>0160-5008</td>
</tr>
<tr>
<td>C10</td>
<td>0160-4814</td>
<td>1</td>
<td>CAP-FXD 1.0µF +.5% 100 V CER C0G</td>
<td>28480</td>
<td>0160-4814</td>
</tr>
<tr>
<td>C11</td>
<td>0160-4835</td>
<td>1</td>
<td>CAP-FXD 0.1µF +.10% 50 V CER X7R</td>
<td>28480</td>
<td>0160-4835</td>
</tr>
<tr>
<td>E1</td>
<td>1251-0639</td>
<td>1</td>
<td>CONNECTOR-SGL CONT SKT .032-IN-BSC-SZ</td>
<td>01075</td>
<td>06-8801</td>
</tr>
<tr>
<td>E2</td>
<td>1251-2229</td>
<td>2</td>
<td>CONNECTOR-SGL CONT SKT .033-IN-BSC-SZ</td>
<td>01380</td>
<td>1331677-3</td>
</tr>
<tr>
<td>E3</td>
<td>1251-2229</td>
<td>2</td>
<td>CONNECTOR-SGL CONT SKT .033-IN-BSC-SZ</td>
<td>01380</td>
<td>1331677-3</td>
</tr>
<tr>
<td>J1</td>
<td>1252-1263</td>
<td>1</td>
<td>CONNECT D-SUBMIN 50-CKT 50 CONT</td>
<td>28480</td>
<td>1252-1263</td>
</tr>
<tr>
<td>J2</td>
<td>1252-2006</td>
<td>1</td>
<td>CONNECT TYPE 0 CONT</td>
<td>28480</td>
<td>1252-2006</td>
</tr>
<tr>
<td>J3</td>
<td>8120-4857</td>
<td>1</td>
<td>CABLE MCNDC T</td>
<td>28480</td>
<td>8120-4857</td>
</tr>
<tr>
<td>R1</td>
<td>0698-8615</td>
<td>1</td>
<td>RESITOR 7.5K +.1% .05W TF TC=0+-100</td>
<td>28480</td>
<td>0698-8615</td>
</tr>
<tr>
<td>R2</td>
<td>0757-0346</td>
<td>2</td>
<td>RESITOR 10 +.1% .125W TF TC=0+-100</td>
<td>28480</td>
<td>0757-0346</td>
</tr>
<tr>
<td>R3</td>
<td>0757-0346</td>
<td>2</td>
<td>RESITOR 10 +.1% .125W TF TC=0+-100</td>
<td>28480</td>
<td>0757-0346</td>
</tr>
<tr>
<td>R4</td>
<td>0698-7252</td>
<td>3</td>
<td>RESITOR 4.6K +.1% .05W TF TC=0+-100</td>
<td>28480</td>
<td>0698-7252</td>
</tr>
<tr>
<td>R5</td>
<td>0757-0465</td>
<td>2</td>
<td>RESITOR 10K +.1% .125W TF TC=0+-100</td>
<td>28480</td>
<td>0757-0465</td>
</tr>
<tr>
<td>R6</td>
<td>0757-0465</td>
<td>2</td>
<td>RESITOR 10K +.1% .125W TF TC=0+-100</td>
<td>28480</td>
<td>0757-0465</td>
</tr>
<tr>
<td>R7</td>
<td>0698-7252</td>
<td>3</td>
<td>RESITOR 4.6K +.1% .05W TF TC=0+-100</td>
<td>28480</td>
<td>0698-7252</td>
</tr>
<tr>
<td>R10</td>
<td>0698-7252</td>
<td>2</td>
<td>RESITOR 4.6K +.1% .05W TF TC=0+-100</td>
<td>28480</td>
<td>0698-7252</td>
</tr>
<tr>
<td>TP1</td>
<td>1251-7819</td>
<td>1</td>
<td>CONNECT TYPE .100-PIN-SPCG-MTG-END</td>
<td>28480</td>
<td>1251-7819</td>
</tr>
<tr>
<td>TP2</td>
<td>0360-2050</td>
<td>2</td>
<td>CONNECTOR-SGL CONT</td>
<td>04055</td>
<td></td>
</tr>
<tr>
<td>TP3</td>
<td>0360-2050</td>
<td>2</td>
<td>CONNECTOR-SGL CONT</td>
<td>04055</td>
<td></td>
</tr>
<tr>
<td>U1</td>
<td>85620-80005</td>
<td>1</td>
<td>EPROM-PRGMD U1</td>
<td>28480</td>
<td>85620-80005</td>
</tr>
<tr>
<td>U2</td>
<td>85620-80007</td>
<td>1</td>
<td>EPROM-PRGMD U2</td>
<td>28480</td>
<td>85620-80007</td>
</tr>
<tr>
<td>U3</td>
<td>85620-80009</td>
<td>1</td>
<td>EPROM-PRGMD U3</td>
<td>28480</td>
<td>85620-80009</td>
</tr>
<tr>
<td>U5</td>
<td>1820-1437</td>
<td>1</td>
<td>IC TTL/LS MONOSTBL CLEAR DUAL</td>
<td>28480</td>
<td>1820-1437</td>
</tr>
<tr>
<td>U6</td>
<td>1820-5384</td>
<td>1</td>
<td>IC-INTERFACE CMOS -888-BIT</td>
<td>28480</td>
<td>1820-5384</td>
</tr>
<tr>
<td>U7</td>
<td>1818-4160</td>
<td>1</td>
<td></td>
<td>28480</td>
<td>1818-4160</td>
</tr>
</tbody>
</table>
## Agilent Part Number 85620-60001

### A1 Mass Memory Assembly (continued)

<table>
<thead>
<tr>
<th>Reference Designator</th>
<th>Agilent Part Number</th>
<th>Qty</th>
<th>Description</th>
<th>Mfr Code</th>
<th>Mfr Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>U8</td>
<td>1826-1858</td>
<td>1</td>
<td>IC PWR MGT-BAT-MGT</td>
<td>28480</td>
<td>1826-1858</td>
</tr>
<tr>
<td>U9</td>
<td>85620-80001</td>
<td>1</td>
<td>PAL-PROGRAMMED U9</td>
<td>28480</td>
<td>85620-80001</td>
</tr>
<tr>
<td>U10</td>
<td>1820-1198</td>
<td>1</td>
<td>IC GATE TTL/LS NAND QUAD 2-INP</td>
<td>28480</td>
<td>1820-1198</td>
</tr>
<tr>
<td>U11</td>
<td>1820-3143</td>
<td>1</td>
<td>IC FF TTL/ALS D-TYPE POS-EDGE-TRIG COM</td>
<td>28480</td>
<td>1820-3143</td>
</tr>
<tr>
<td>U12</td>
<td>1820-2024</td>
<td>1</td>
<td>IC DRV R TTL/LS BUS OCTL</td>
<td>28480</td>
<td>1820-2024</td>
</tr>
<tr>
<td>U13</td>
<td>1810-0205</td>
<td>1</td>
<td>NETWORK-RES 8-SIP 4.7K OHM X 7</td>
<td>28480</td>
<td>1810-0205</td>
</tr>
<tr>
<td>U14</td>
<td>1820-1190</td>
<td>1</td>
<td>IC INV TTL/LS HEX 1-INP</td>
<td>28480</td>
<td>1820-1190</td>
</tr>
<tr>
<td>U15</td>
<td>1820-5497</td>
<td>1</td>
<td>IC-NONVOLATILE MEMORY CONTROLLER</td>
<td>28480</td>
<td>1820-5497</td>
</tr>
<tr>
<td>XU1</td>
<td>1200-0567</td>
<td>1</td>
<td>SOCKET-IC28-CONT DIP DIP-SDLR</td>
<td>00779</td>
<td>641605-1</td>
</tr>
<tr>
<td>XU2</td>
<td>1200-0567</td>
<td>1</td>
<td>SOCKET-IC28-CONT DIP DIP-SDLR</td>
<td>00779</td>
<td>641605-1</td>
</tr>
<tr>
<td>XU3</td>
<td>1200-0567</td>
<td>1</td>
<td>SOCKET-IC28-CONT DIP DIP-SDLR</td>
<td>00779</td>
<td>641605-1</td>
</tr>
</tbody>
</table>

Agilent 85620A ACCESSORIES PARTS LIST

- 0950-1964  1  MEMORY CARD, SRAM  10421  RBCO32E008
- 1420-0583  1  MEMORY CARD BATTERY  08709  CR2016

*Indicates factory-selected value*
Component-Level Information Packet for 85620-60006

Mass Memory Assembly

Manufacturing Part Number: 85620-90035
Printed in USA
November 2000

© Copyright 1990 - 2000 Agilent Technologies, Inc.
The information contained in this document is subject to change without notice.

Agilent Technologies makes no warranty of any kind with regard to this material, including but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Agilent Technologies shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.
### Agilent Part Number 85620-60006
#### A1 Mass Memory Assembly

<table>
<thead>
<tr>
<th>Reference Designator</th>
<th>Agilent Part Number</th>
<th>Qty</th>
<th>Description</th>
<th>Mfr Code</th>
<th>Mfr Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>BT1</td>
<td>1420-0341</td>
<td>1</td>
<td>BATTERY 3V 1.2A-HR LITHIUM POLYCARBON</td>
<td>28480</td>
<td>1420-0341</td>
</tr>
<tr>
<td>C1</td>
<td>0180-4135</td>
<td>1</td>
<td>CAP-FXD 33μF +.10% 10 V TA</td>
<td>28480</td>
<td>0180-4135</td>
</tr>
<tr>
<td>C2</td>
<td>0160-4832</td>
<td>1</td>
<td>CAP-FXD 0.01μF +.10% 100 V CER X7R</td>
<td>28480</td>
<td>0160-4832</td>
</tr>
<tr>
<td>C3</td>
<td>0160-4832</td>
<td>1</td>
<td>CAP-FXD 0.01μF +.10% 100 V CER X7R</td>
<td>28480</td>
<td>0160-4832</td>
</tr>
<tr>
<td>C4</td>
<td>0160-4832</td>
<td>1</td>
<td>CAP-FXD 0.01μF +.10% 100 V CER X7R</td>
<td>28480</td>
<td>0160-4832</td>
</tr>
<tr>
<td>C5</td>
<td>0160-4835</td>
<td>1</td>
<td>CAP-FXD 0.1μF +.10% 50 V CER X7R</td>
<td>28480</td>
<td>0160-4835</td>
</tr>
<tr>
<td>C6</td>
<td>0180-4136</td>
<td>1</td>
<td>CAP-FXD 10μF +.10% 20 V TA</td>
<td>28480</td>
<td>0180-4136</td>
</tr>
<tr>
<td>C7</td>
<td>0160-4832</td>
<td>1</td>
<td>CAP-FXD 0.01μF +.10% 100 V CER X7R</td>
<td>28480</td>
<td>0160-4832</td>
</tr>
<tr>
<td>C8</td>
<td>0160-4832</td>
<td>1</td>
<td>CAP-FXD 0.01μF +.10% 100 V CER X7R</td>
<td>28480</td>
<td>0160-4832</td>
</tr>
<tr>
<td>C9</td>
<td>0160-4832</td>
<td>1</td>
<td>CAP-FXD 0.01μF +.10% 100 V CER X7R</td>
<td>28480</td>
<td>0160-4832</td>
</tr>
<tr>
<td>C10</td>
<td>0180-3817</td>
<td>1</td>
<td>CAP-FXD 100μF +.20% 50 V AL-ELCTLT</td>
<td>28480</td>
<td>0180-3817</td>
</tr>
<tr>
<td>C11</td>
<td>0180-3817</td>
<td>1</td>
<td>CAP-FXD 100μF +.20% 50 V AL-ELCTLT</td>
<td>28480</td>
<td>0180-3817</td>
</tr>
<tr>
<td>C12</td>
<td>0160-4832</td>
<td>1</td>
<td>CAP-FXD 0.01μF +.10% 100 V CER X7R</td>
<td>28480</td>
<td>0160-4832</td>
</tr>
<tr>
<td>C13</td>
<td>0180-4214</td>
<td>1</td>
<td>CAP-FXD 220μF +.10% 10 V TA</td>
<td>28480</td>
<td>0180-4214</td>
</tr>
<tr>
<td>E1</td>
<td>9100-1788</td>
<td>1</td>
<td>CORE-FERRITE CHOKE-WIDE BAND;IMP;&gt;680</td>
<td>28480</td>
<td>9100-1788</td>
</tr>
<tr>
<td>E2</td>
<td>9100-1788</td>
<td>1</td>
<td>CORE-FERRITE CHOKE-WIDE BAND;IMP;&gt;680</td>
<td>28480</td>
<td>9100-1788</td>
</tr>
<tr>
<td>E3</td>
<td>9100-1788</td>
<td>1</td>
<td>CORE-FERRITE CHOKE-WIDE BAND;IMP;&gt;680</td>
<td>28480</td>
<td>9100-1788</td>
</tr>
<tr>
<td>E4</td>
<td>9100-1788</td>
<td>1</td>
<td>CORE-FERRITE CHOKE-WIDE BAND;IMP;&gt;680</td>
<td>28480</td>
<td>9100-1788</td>
</tr>
<tr>
<td>E5</td>
<td>9100-1788</td>
<td>1</td>
<td>CORE-FERRITE CHOKE-WIDE BAND;IMP;&gt;680</td>
<td>28480</td>
<td>9100-1788</td>
</tr>
<tr>
<td>E6</td>
<td>9100-1788</td>
<td>1</td>
<td>CORE-FERRITE CHOKE-WIDE BAND;IMP;&gt;680</td>
<td>28480</td>
<td>9100-1788</td>
</tr>
<tr>
<td>J1</td>
<td>1252-1263</td>
<td>1</td>
<td>CONN-RECT D-SUBMIN 30-CKT 50-CONT</td>
<td>28480</td>
<td>1252-1263</td>
</tr>
<tr>
<td>J2</td>
<td>1252-2006</td>
<td>1</td>
<td>CONN-POST TYPE 0-CONT</td>
<td>28480</td>
<td>1252-2006</td>
</tr>
<tr>
<td>J3</td>
<td>8120-4857</td>
<td>1</td>
<td>CABLE-MCN DCCT</td>
<td>28480</td>
<td>8120-4857</td>
</tr>
<tr>
<td>J4</td>
<td>1252-1327</td>
<td>1</td>
<td>CONN-RECT D-SUBMIN 9-CKT 9-CONT</td>
<td>28480</td>
<td>1252-1327</td>
</tr>
<tr>
<td>MP2</td>
<td>5001-8737</td>
<td>1</td>
<td>CONTACT-GROUND</td>
<td>28480</td>
<td>5001-8737</td>
</tr>
<tr>
<td>MP4</td>
<td>1590-1361</td>
<td>2</td>
<td>THREADED INSERT-NUT M2.5 X 0.45</td>
<td>28480</td>
<td>1590-1361</td>
</tr>
<tr>
<td>MP6</td>
<td>0515-0367</td>
<td>2</td>
<td>SCREW-MACHINE ASSEMBLY M2.5 X 0.45</td>
<td>28480</td>
<td>0515-0367</td>
</tr>
<tr>
<td>R1</td>
<td>0757-0346</td>
<td>1</td>
<td>RESISTOR 10 +.1% .125W TF TC=0+.100</td>
<td>28480</td>
<td>0757-0346</td>
</tr>
<tr>
<td>R2</td>
<td>0757-0465</td>
<td>1</td>
<td>RESISTOR 100K +.1% .125W TF TC=0+.100</td>
<td>28480</td>
<td>0757-0465</td>
</tr>
<tr>
<td>R3</td>
<td>0098-3155</td>
<td>1</td>
<td>RESISTOR 4.6K +.1% .125W TF TC=0+.100</td>
<td>28480</td>
<td>0098-3155</td>
</tr>
<tr>
<td>R4</td>
<td>0757-0346</td>
<td>1</td>
<td>RESISTOR 10 +.1% .125W TF TC=0+.100</td>
<td>28480</td>
<td>0757-0346</td>
</tr>
<tr>
<td>R5</td>
<td>0098-8816</td>
<td>1</td>
<td>RESISTOR 2.15 +.1% .125W TF TC=0+.100</td>
<td>28480</td>
<td>0098-8816</td>
</tr>
<tr>
<td>R6</td>
<td>0757-0465</td>
<td>1</td>
<td>RESISTOR 100K +.1% .125W TF TC=0+.100</td>
<td>28480</td>
<td>0757-0465</td>
</tr>
<tr>
<td>R7</td>
<td>0098-8816</td>
<td>1</td>
<td>RESISTOR 2.15 +.1% .125W TF TC=0+.100</td>
<td>28480</td>
<td>0098-8816</td>
</tr>
<tr>
<td>R8</td>
<td>0098-8816</td>
<td>1</td>
<td>RESISTOR 2.15 +.1% .125W TF TC=0+.100</td>
<td>28480</td>
<td>0098-8816</td>
</tr>
</tbody>
</table>

*Indicates factory-selected value*  

CLIP 85620-90035  

1
## Agilent Part Number 85620-60006
### A1 Mass Memory Assembly (continued)

<table>
<thead>
<tr>
<th>Reference Designator</th>
<th>Agilent Part Number</th>
<th>Qty</th>
<th>Description</th>
<th>Mfr Code</th>
<th>Mfr Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>R9</td>
<td>0757-0442</td>
<td>1</td>
<td>RESISTOR 10K +/-1% .125W TF TC=0+100</td>
<td>28480</td>
<td>0757-0442</td>
</tr>
<tr>
<td>TP1</td>
<td>1251-7819</td>
<td>1</td>
<td>CONN-POST TYPE .100-PIN-SPC-G-MTG-END</td>
<td>28480</td>
<td>1251-7819</td>
</tr>
<tr>
<td>U1</td>
<td>1820-3143</td>
<td>1</td>
<td>IC FF TTL/ALS D-TYPE POS-EDGE-TRIG COM</td>
<td>28480</td>
<td>1820-3143</td>
</tr>
<tr>
<td>U2</td>
<td>1818-4604</td>
<td>1</td>
<td>IC 1M-BIT SRAM 120-NS CMOS</td>
<td>28480</td>
<td>1818-4604</td>
</tr>
<tr>
<td>U3</td>
<td>85620-80020</td>
<td>1</td>
<td>PAL-PRGM U3</td>
<td>28480</td>
<td>85620-80020</td>
</tr>
<tr>
<td>U4</td>
<td>1820-1198</td>
<td>1</td>
<td>IC GATE TTL/LS NAND QUAD 2-INP</td>
<td>28480</td>
<td>1820-1198</td>
</tr>
<tr>
<td>U5</td>
<td>1820-5384</td>
<td>1</td>
<td>IC INTERFACE CMOS 888-BIT</td>
<td>28480</td>
<td>1820-5384</td>
</tr>
<tr>
<td>U6</td>
<td>1810-0205</td>
<td>1</td>
<td>NETWORK-RES 8-SIP 4.7K OHM X 7</td>
<td>28480</td>
<td>1810-0205</td>
</tr>
<tr>
<td>U7</td>
<td>85620-80021</td>
<td>1</td>
<td>EPROM, PROGRAMMED</td>
<td>28480</td>
<td>85620-80021</td>
</tr>
<tr>
<td>U8</td>
<td>1810-0205</td>
<td>1</td>
<td>NETWORK-RES 8-SIP 4.7K OHM X 7</td>
<td>28480</td>
<td>1810-0205</td>
</tr>
<tr>
<td>U9</td>
<td>1820-1199</td>
<td>1</td>
<td>IC INV TTL/LS HEX 1-INP</td>
<td>28480</td>
<td>1820-1199</td>
</tr>
<tr>
<td>U10</td>
<td>1820-2024</td>
<td>1</td>
<td>IC DHVR TTL/LS BUS OCTL</td>
<td>28480</td>
<td>1820-2024</td>
</tr>
<tr>
<td>U11</td>
<td>1826-1878</td>
<td>1</td>
<td>IC PWR MGT-BAT-MGT</td>
<td>28480</td>
<td>1826-1878</td>
</tr>
<tr>
<td>U12</td>
<td>1820-5497</td>
<td>1</td>
<td>IC-NONVOLATILE MEMORY CONTROLLER</td>
<td>28480</td>
<td>1820-5497</td>
</tr>
<tr>
<td>U13</td>
<td>1820-3465</td>
<td>1</td>
<td>IC FF TTL/ALS D-TYPE POS-EDGE-TRIG COM</td>
<td>28480</td>
<td>1820-3465</td>
</tr>
<tr>
<td>U14</td>
<td>1906-0229</td>
<td>1</td>
<td>DIODE-ARRAY 30V 400MA TO-116</td>
<td>28480</td>
<td>1906-0229</td>
</tr>
<tr>
<td>W1</td>
<td>8159-0005</td>
<td>1</td>
<td>RESISTOR 0 CW</td>
<td>28480</td>
<td>8159-0005</td>
</tr>
<tr>
<td>XBT1</td>
<td>1400-1465</td>
<td>1</td>
<td>HOLDER-BAT .625-WD AL-2024-T4</td>
<td>05535</td>
<td>131</td>
</tr>
<tr>
<td>XU7</td>
<td>1200-1314</td>
<td>1</td>
<td>SOCKET-IC-DIP 32-CONT DIP-SLDR</td>
<td>01380</td>
<td>2444018-1</td>
</tr>
</tbody>
</table>

**Agilent 85620A ACCESSORY PARTS LIST**

<table>
<thead>
<tr>
<th>Qty</th>
<th>Description</th>
<th>Mfr Code</th>
<th>Mfr Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>0950-1964</td>
<td>MEMORY CARD, SRAM</td>
<td>10421</td>
<td>RBC0321E00</td>
</tr>
<tr>
<td>1420-0383</td>
<td>MEMORY CARD BATTERY</td>
<td>08709</td>
<td>CR2016</td>
</tr>
</tbody>
</table>

*Indicates factory-selected value*
The information contained in this document is subject to change without notice.

Agilent Technologies makes no warranty of any kind with regard to this material, including but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Agilent Technologies shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.
# Agilent Part Number 85620-60008
## A1 Mass Memory Assembly

<table>
<thead>
<tr>
<th>Reference Designator</th>
<th>Agilent Part Number</th>
<th>Qty</th>
<th>Description</th>
<th>Mfr Code</th>
<th>Mfr Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>BT1</td>
<td>1420-0394</td>
<td>1</td>
<td>BATTERY 3V 1A-4HR LI MANGANESE DIOXIDE</td>
<td>28480</td>
<td>1420-0394</td>
</tr>
<tr>
<td>C1</td>
<td>0180-4136</td>
<td>1</td>
<td>CAP-FXD 10hF +.10% 20 V TA</td>
<td>28480</td>
<td>0180-4136</td>
</tr>
<tr>
<td>C2</td>
<td>0180-4135</td>
<td>1</td>
<td>CAP-FXD 33aF +.10% 10 V TA</td>
<td>28480</td>
<td>0180-4135</td>
</tr>
<tr>
<td>C3</td>
<td>0160-4832</td>
<td>1</td>
<td>CAP-FXD 0.01uF +.10% 100 V CER X 7R</td>
<td>28480</td>
<td>0160-4832</td>
</tr>
<tr>
<td>C4</td>
<td>0160-4832</td>
<td>1</td>
<td>CAP-FXD 0.01uF +.10% 100 V CER X 7R</td>
<td>28480</td>
<td>0160-4832</td>
</tr>
<tr>
<td>C5</td>
<td>0160-4832</td>
<td>1</td>
<td>CAP-FXD 0.01uF +.10% 100 V CER X 7R</td>
<td>28480</td>
<td>0160-4832</td>
</tr>
<tr>
<td>C6</td>
<td>0160-4832</td>
<td>1</td>
<td>CAP-FXD 0.01uF +.10% 100 V CER X 7R</td>
<td>28480</td>
<td>0160-4832</td>
</tr>
<tr>
<td>C7</td>
<td>0160-4832</td>
<td>1</td>
<td>CAP-FXD 0.01uF +.10% 100 V CER X 7R</td>
<td>28480</td>
<td>0160-4832</td>
</tr>
<tr>
<td>C8</td>
<td>0160-4832</td>
<td>1</td>
<td>CAP-FXD 0.01uF +.10% 100 V CER X 7R</td>
<td>28480</td>
<td>0160-4832</td>
</tr>
<tr>
<td>C9</td>
<td>0160-5008</td>
<td>1</td>
<td>CAP-FXD 0.22uF +.10% 50 V CER X7R</td>
<td>28480</td>
<td>0160-5008</td>
</tr>
<tr>
<td>C10</td>
<td>0160-4814</td>
<td>1</td>
<td>CAP-FXD 150pF +.5% 100 V CER C0G</td>
<td>28480</td>
<td>0160-4814</td>
</tr>
<tr>
<td>C11</td>
<td>0160-4835</td>
<td>1</td>
<td>CAP-FXD 0.1uF +.10% 30 V CER X7R</td>
<td>28480</td>
<td>0160-4835</td>
</tr>
<tr>
<td>J1</td>
<td>1252-2103</td>
<td>1</td>
<td>CONN-RECT D-SUBMIN 30-CKT 30-CONT</td>
<td>28480</td>
<td>1252-2103</td>
</tr>
<tr>
<td>J2</td>
<td>1252-2096</td>
<td>1</td>
<td>CONN-POST TYPE 0-CONT</td>
<td>28480</td>
<td>1252-2096</td>
</tr>
<tr>
<td>J3</td>
<td>8120-4857</td>
<td>1</td>
<td>CABLE-MCND CT</td>
<td>28480</td>
<td>8120-4857</td>
</tr>
<tr>
<td>N0ONE0033</td>
<td>0515-1717</td>
<td>1</td>
<td>SCREW-MACHINE ASSEMBLY M2.5 X 0.45</td>
<td>28480</td>
<td>011125</td>
</tr>
<tr>
<td>N0ONE0034</td>
<td>0590-3355</td>
<td>1</td>
<td>THREADED INSERT-NUT M2.5 X 0.45</td>
<td>28480</td>
<td>F-M2.5-1</td>
</tr>
<tr>
<td>N0ONE0035</td>
<td>3050-0890</td>
<td>1</td>
<td>WASHER-FL. MTLG 2.5 2.78-MM-ID 6.35-MM-OD</td>
<td>28480</td>
<td>00691</td>
</tr>
<tr>
<td>N0ONE0036</td>
<td>5022-0008</td>
<td>1</td>
<td>BATTERY SUPPORT</td>
<td>28480</td>
<td>3022-0008</td>
</tr>
<tr>
<td>N0ONE0037</td>
<td>5041-7282</td>
<td>1</td>
<td>B-INSUL TO-220</td>
<td>28480</td>
<td>5041-7282</td>
</tr>
<tr>
<td>N0ONE0038</td>
<td>7121-4611</td>
<td>1</td>
<td>LABEL-INFORMATION .15-IN-WD .06-IN-LG</td>
<td>28480</td>
<td>000002</td>
</tr>
<tr>
<td>R1</td>
<td>0757-0346</td>
<td>1</td>
<td>RESISTOR 10 +1% .125W TF TC=0+100</td>
<td>28480</td>
<td>0757-0346</td>
</tr>
<tr>
<td>R2</td>
<td>0757-0346</td>
<td>1</td>
<td>RESISTOR 10 +1% .125W TF TC=0+100</td>
<td>28480</td>
<td>0757-0346</td>
</tr>
<tr>
<td>R3</td>
<td>0757-0346</td>
<td>1</td>
<td>RESISTOR 10 +1% .125W TF TC=0+100</td>
<td>28480</td>
<td>0757-0346</td>
</tr>
<tr>
<td>R4</td>
<td>0698-7252</td>
<td>1</td>
<td>RESISTOR 4.64K +1% .05W TF TC=0+100</td>
<td>28480</td>
<td>0698-7252</td>
</tr>
<tr>
<td>R5</td>
<td>0757-0465</td>
<td>1</td>
<td>RESISTOR 100K +1% .125W TF TC=0+100</td>
<td>28480</td>
<td>0757-0465</td>
</tr>
<tr>
<td>R7</td>
<td>0757-0465</td>
<td>1</td>
<td>RESISTOR 100K +1% .125W TF TC=0+100</td>
<td>28480</td>
<td>0757-0465</td>
</tr>
<tr>
<td>R9</td>
<td>0698-7252</td>
<td>1</td>
<td>RESISTOR 4.64K +1% .05W TF TC=0+100</td>
<td>28480</td>
<td>0698-7252</td>
</tr>
<tr>
<td>R10</td>
<td>0698-7252</td>
<td>1</td>
<td>RESISTOR 4.64K +1% .05W TF TC=0+100</td>
<td>28480</td>
<td>0698-7252</td>
</tr>
<tr>
<td>TP1</td>
<td>1251-7819</td>
<td>1</td>
<td>CONN-POST TYPE 100-PIN-SPC-FMG-CON</td>
<td>28480</td>
<td>1251-7819</td>
</tr>
<tr>
<td>TP2</td>
<td>0300-1682</td>
<td>1</td>
<td>TERMINAL-STUD SGL-TUR PRESS-MTG</td>
<td>28480</td>
<td>14758-8</td>
</tr>
<tr>
<td>TP3</td>
<td>0300-1682</td>
<td>1</td>
<td>TERMINAL-STUD SGL-TUR PRESS-MTG</td>
<td>28480</td>
<td>14758-8</td>
</tr>
<tr>
<td>U1</td>
<td>85620-80014</td>
<td>1</td>
<td>EPROM-PRG-MEDU1</td>
<td>28480</td>
<td>85620-80014</td>
</tr>
<tr>
<td>U2</td>
<td>85620-80015</td>
<td>1</td>
<td>EPROM-PRG-MEDU2</td>
<td>28480</td>
<td>85620-80015</td>
</tr>
<tr>
<td>U3</td>
<td>85620-80016</td>
<td>1</td>
<td>EPROM-PRG-MEDU3</td>
<td>28480</td>
<td>85620-80016</td>
</tr>
</tbody>
</table>

*Indicates factory-selected value
<table>
<thead>
<tr>
<th>Reference Designator</th>
<th>Agilent Part Number</th>
<th>Qty</th>
<th>Description</th>
<th>Mfr Code</th>
<th>Mfr Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>U5</td>
<td>1820-1437</td>
<td>1</td>
<td>IC MV TTL/LS MONOSTBL CLEAR DUAL</td>
<td>28480</td>
<td>1820-1437</td>
</tr>
<tr>
<td>U6</td>
<td>1820-5384</td>
<td>1</td>
<td>IC INTERFACE CMOS -888-BIT</td>
<td>28480</td>
<td>1820-5384</td>
</tr>
<tr>
<td>U7</td>
<td>1818-4604</td>
<td>1</td>
<td>IC 1M-BIT SRAM 120-NS CMOS</td>
<td>28480</td>
<td>1818-4604</td>
</tr>
<tr>
<td>U8</td>
<td>1826-1858</td>
<td>1</td>
<td>IC PWR MGT-BAT-MGT</td>
<td>28480</td>
<td>1826-1858</td>
</tr>
<tr>
<td>U9</td>
<td>85620-80001</td>
<td>1</td>
<td>PAL, PROGRAMMED</td>
<td>28480</td>
<td>85620-80001</td>
</tr>
<tr>
<td>U10</td>
<td>1820-1198</td>
<td>1</td>
<td>IC GATE TTL/LS NAND QUAD 2-INP</td>
<td>28480</td>
<td>1820-1198</td>
</tr>
<tr>
<td>U11</td>
<td>1820-3143</td>
<td>1</td>
<td>IC FF TTL/ALS D-TYPE POS-EDGE-TRIG COM</td>
<td>28480</td>
<td>1820-3143</td>
</tr>
<tr>
<td>U12</td>
<td>1820-2024</td>
<td>1</td>
<td>IC DRVR TTL/LS BUS OCTL</td>
<td>28480</td>
<td>1820-2024</td>
</tr>
<tr>
<td>U13</td>
<td>1810-0205</td>
<td>1</td>
<td>NETWORK-RES 8-SIP 4.7K OHM X 7</td>
<td>28480</td>
<td>1810-0205</td>
</tr>
<tr>
<td>U14</td>
<td>1820-1199</td>
<td>1</td>
<td>IC INV TTL/LS HEX 1-INP</td>
<td>28480</td>
<td>1820-1199</td>
</tr>
<tr>
<td>U15</td>
<td>1820-5497</td>
<td>1</td>
<td>IC-NONVOLATILE MEMORY CONTROLLER</td>
<td>28480</td>
<td>1820-5497</td>
</tr>
<tr>
<td>XU1</td>
<td>1200-0567</td>
<td>1</td>
<td>SOCKET-IC28-CONT DIP DIP-SLDR</td>
<td>00779</td>
<td>616065-1</td>
</tr>
<tr>
<td>XU2</td>
<td>1200-0567</td>
<td>1</td>
<td>SOCKET-IC28-CONT DIP DIP-SLDR</td>
<td>00779</td>
<td>616065-1</td>
</tr>
<tr>
<td>XU3</td>
<td>1200-0567</td>
<td>1</td>
<td>SOCKET-IC28-CONT DIP DIP-SLDR</td>
<td>00779</td>
<td>616065-1</td>
</tr>
</tbody>
</table>

Agilent 85620A ACCESSORIES PARTS LIST

<table>
<thead>
<tr>
<th>Reference Designator</th>
<th>Agilent Part Number</th>
<th>Qty</th>
<th>Description</th>
<th>Mfr Code</th>
<th>Mfr Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>0950-1964</td>
<td>MEMORY CARD, SRAM</td>
<td>1</td>
<td></td>
<td>10421</td>
<td>RBCO32IE00</td>
</tr>
<tr>
<td>1420-0583</td>
<td>MEMORY CARD BATTERY</td>
<td>1</td>
<td></td>
<td>08709</td>
<td>CR2016</td>
</tr>
</tbody>
</table>

*Indicates factory-selected value*
The information contained in this document is subject to change without notice.

Agilent Technologies makes no warranty of any kind with regard to this material, including but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Agilent Technologies shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.
<table>
<thead>
<tr>
<th>Reference Designator</th>
<th>Agilent Part Number</th>
<th>Qty</th>
<th>Description</th>
<th>Mfr Code</th>
<th>Mfr Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>BT1</td>
<td>1420-0341</td>
<td>1</td>
<td>BATTERY 3V 1.2A-HR LITHIUM POLYCARBON</td>
<td>28480</td>
<td>1420-0341</td>
</tr>
<tr>
<td>C1</td>
<td>0180-4135</td>
<td>1</td>
<td>CAP-FXD 33µF +10% 10 V TA</td>
<td>28480</td>
<td>0180-4135</td>
</tr>
<tr>
<td>C2</td>
<td>0160-4832</td>
<td>1</td>
<td>CAP-FXD 0.01µF +10% 100 V CER X 7R</td>
<td>28480</td>
<td>0160-4832</td>
</tr>
<tr>
<td>C3</td>
<td>0160-4832</td>
<td>1</td>
<td>CAP-FXD 0.01µF +10% 100 V CER X 7R</td>
<td>28480</td>
<td>0160-4832</td>
</tr>
<tr>
<td>C4</td>
<td>0160-4832</td>
<td>1</td>
<td>CAP-FXD 0.01µF +10% 100 V CER X 7R</td>
<td>28480</td>
<td>0160-4832</td>
</tr>
<tr>
<td>C5</td>
<td>0160-4835</td>
<td>1</td>
<td>CAP-FXD 0.1µF +10% 50 V CER X 7R</td>
<td>28480</td>
<td>0160-4835</td>
</tr>
<tr>
<td>C6</td>
<td>0180-4136</td>
<td>1</td>
<td>CAP-FXD 100µF +10% 20 V TA</td>
<td>28480</td>
<td>0180-4136</td>
</tr>
<tr>
<td>C7</td>
<td>0160-4832</td>
<td>1</td>
<td>CAP-FXD 0.01µF +10% 100 V CER X 7R</td>
<td>28480</td>
<td>0160-4832</td>
</tr>
<tr>
<td>C8</td>
<td>0160-4832</td>
<td>1</td>
<td>CAP-FXD 0.01µF +10% 100 V CER X 7R</td>
<td>28480</td>
<td>0160-4832</td>
</tr>
<tr>
<td>C9</td>
<td>0160-4832</td>
<td>1</td>
<td>CAP-FXD 0.01µF +10% 100 V CER X 7R</td>
<td>28480</td>
<td>0160-4832</td>
</tr>
<tr>
<td>C10</td>
<td>0180-3817</td>
<td>1</td>
<td>CAP-FXD 1000µF +20% 50 V AL-ELCTLT</td>
<td>28480</td>
<td>0180-3817</td>
</tr>
<tr>
<td>C11</td>
<td>0180-3817</td>
<td>1</td>
<td>CAP-FXD 1000µF +20% 50 V AL-ELCTLT</td>
<td>28480</td>
<td>0180-3817</td>
</tr>
<tr>
<td>C12</td>
<td>0160-4832</td>
<td>1</td>
<td>CAP-FXD 0.01µF +10% 100 V CER X 7R</td>
<td>28480</td>
<td>0160-4832</td>
</tr>
<tr>
<td>C13</td>
<td>0180-2208</td>
<td>1</td>
<td>CAP-FXD 200µF +10% 10 V TA</td>
<td>28480</td>
<td>0180-2208</td>
</tr>
<tr>
<td>E1</td>
<td>9100-1788</td>
<td>1</td>
<td>CORE-FERRITE CHOKE-WIDEBAND;IMP:5680</td>
<td>28480</td>
<td>9100-1788</td>
</tr>
<tr>
<td>E2</td>
<td>9100-1788</td>
<td>1</td>
<td>CORE-FERRITE CHOKE-WIDEBAND;IMP:5680</td>
<td>28480</td>
<td>9100-1788</td>
</tr>
<tr>
<td>E3</td>
<td>9100-1788</td>
<td>1</td>
<td>CORE-FERRITE CHOKE-WIDEBAND;IMP:5680</td>
<td>28480</td>
<td>9100-1788</td>
</tr>
<tr>
<td>E4</td>
<td>9100-1788</td>
<td>1</td>
<td>CORE-FERRITE CHOKE-WIDEBAND;IMP:5680</td>
<td>28480</td>
<td>9100-1788</td>
</tr>
<tr>
<td>E5</td>
<td>9100-1788</td>
<td>1</td>
<td>CORE-FERRITE CHOKE-WIDEBAND;IMP:5680</td>
<td>28480</td>
<td>9100-1788</td>
</tr>
<tr>
<td>E6</td>
<td>9100-1788</td>
<td>1</td>
<td>CORE-FERRITE CHOKE-WIDEBAND;IMP:5680</td>
<td>28480</td>
<td>9100-1788</td>
</tr>
<tr>
<td>J1</td>
<td>1252-1263</td>
<td>1</td>
<td>CONN-RECT D-SUBMIN 50-CKT 50-CONT</td>
<td>28480</td>
<td>1252-1263</td>
</tr>
<tr>
<td>J2</td>
<td>1252-2906</td>
<td>1</td>
<td>CONN-POST TYPE 0-CONT</td>
<td>28480</td>
<td>1252-2906</td>
</tr>
<tr>
<td>J3</td>
<td>8120-4857</td>
<td>1</td>
<td>CABLE-MCND CT</td>
<td>28480</td>
<td>8120-4857</td>
</tr>
<tr>
<td>J4</td>
<td>1252-1327</td>
<td>1</td>
<td>CONN-RECT D-SUBMIN 9-CKT 9-CONT</td>
<td>28480</td>
<td>1252-1327</td>
</tr>
<tr>
<td>MP3</td>
<td>5001-8737</td>
<td>1</td>
<td>CONTACT-GROUND</td>
<td>28480</td>
<td>5001-8737</td>
</tr>
<tr>
<td>MP4</td>
<td>0590-1361</td>
<td>1</td>
<td>THREADED INSERT-NUT M2.5 X 0.45</td>
<td>03981</td>
<td>KF2-2.5M-ET</td>
</tr>
<tr>
<td>MP5</td>
<td>0590-1361</td>
<td>1</td>
<td>THREADED INSERT-NUT M2.5 X 0.45</td>
<td>03981</td>
<td>KF2-2.5M-ET</td>
</tr>
<tr>
<td>MP6</td>
<td>0515-0937</td>
<td>1</td>
<td>SCREW-MACHINE ASSEMBLY M2.5 X 0.45</td>
<td>05610</td>
<td></td>
</tr>
<tr>
<td>MP7</td>
<td>0515-0367</td>
<td>1</td>
<td>SCREW-MACHINE ASSEMBLY M2.5 X 0.45</td>
<td>05610</td>
<td></td>
</tr>
<tr>
<td>MP8</td>
<td>1400-2112</td>
<td>1</td>
<td>ACCESS .89-WD ABS</td>
<td>05535</td>
<td>1029C</td>
</tr>
<tr>
<td>MP9</td>
<td>0361-0099</td>
<td>1</td>
<td>RIVET-SEMIBUF OTH .123.D1A .188.LG</td>
<td>02531</td>
<td>R-401 2</td>
</tr>
<tr>
<td>MP10</td>
<td>0361-0099</td>
<td>1</td>
<td>RIVET-SEMIBUF OTH .123.D1A .188.LG</td>
<td>02531</td>
<td>R-401 2</td>
</tr>
<tr>
<td>NONE</td>
<td>85671-90007</td>
<td>1</td>
<td>DEMO QR CARD</td>
<td>28480</td>
<td>85671-90007</td>
</tr>
<tr>
<td>NONE</td>
<td>0470-0410</td>
<td>1</td>
<td>COATING DOW CORNING 3140 SIL-RRH-RTV 1P</td>
<td>04514</td>
<td>3140</td>
</tr>
<tr>
<td>R1</td>
<td>0757-0346</td>
<td>1</td>
<td>RESISTOR 10+1% .125W TF TC=0+100</td>
<td>28480</td>
<td>0757-0346</td>
</tr>
</tbody>
</table>

*Indicates factory-selected value
### Agilent Part Number 85620-60018

**Mass Memory (continued)**

<table>
<thead>
<tr>
<th>Reference Designator</th>
<th>Agilent Part Number</th>
<th>Qty</th>
<th>Description</th>
<th>Mfr Code</th>
<th>Mfr Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>R2</td>
<td>0757-0465</td>
<td>1</td>
<td>RESISTOR 100K +1% .125W TF TC=0-100</td>
<td>28-480</td>
<td>0757-0465</td>
</tr>
<tr>
<td>R3</td>
<td>0698-3155</td>
<td>1</td>
<td>RESISTOR 4.64K +1% .125W TF TC=0-100</td>
<td>28-480</td>
<td>0698-3155</td>
</tr>
<tr>
<td>R4</td>
<td>0757-0346</td>
<td>1</td>
<td>RESISTOR 10 +1% .125W TF TC=0+100</td>
<td>28-480</td>
<td>0757-0346</td>
</tr>
<tr>
<td>R5</td>
<td>0698-8816</td>
<td>1</td>
<td>RESISTOR 215 +1% .125W TF TC=0+100</td>
<td>28-480</td>
<td>0698-8816</td>
</tr>
<tr>
<td>R6</td>
<td>0757-0465</td>
<td>1</td>
<td>RESISTOR 100K +1% .125W TF TC=0-100</td>
<td>28-480</td>
<td>0757-0465</td>
</tr>
<tr>
<td>R7</td>
<td>0698-8816</td>
<td>1</td>
<td>RESISTOR 215 +1% .125W TF TC=0+100</td>
<td>28-480</td>
<td>0698-8816</td>
</tr>
<tr>
<td>R8</td>
<td>0698-8816</td>
<td>1</td>
<td>RESISTOR 215 +1% .125W TF TC=0+100</td>
<td>28-480</td>
<td>0698-8816</td>
</tr>
<tr>
<td>R9</td>
<td>0757-0442</td>
<td>1</td>
<td>RESISTOR 10K +1% .125W TF TC=0+100</td>
<td>28-480</td>
<td>0757-0442</td>
</tr>
<tr>
<td>TP1</td>
<td>1251-7819</td>
<td>1</td>
<td>CONN-POST TYPE 100-PIN-SPCG-MTG-END</td>
<td>28-480</td>
<td>1251-7819</td>
</tr>
<tr>
<td>U1</td>
<td>1820-3143</td>
<td>1</td>
<td>IC FF TTL/ALS D-TYPE POS-EDGE-TRIG COM</td>
<td>28-480</td>
<td>1820-3143</td>
</tr>
<tr>
<td>U2</td>
<td>1818-4604</td>
<td>1</td>
<td>IC 1M-BIT SRAM 120-NS CMOS</td>
<td>28-480</td>
<td>1818-4604</td>
</tr>
<tr>
<td>U3</td>
<td>85620-80020</td>
<td>1</td>
<td>PAL-PRGM U3</td>
<td>28-480</td>
<td>85620-80020</td>
</tr>
<tr>
<td>U4</td>
<td>1820-1198</td>
<td>1</td>
<td>IC GATE TTL/LS NAND QUAD 2-INP</td>
<td>28-480</td>
<td>1820-1198</td>
</tr>
<tr>
<td>U5</td>
<td>1820-3384</td>
<td>1</td>
<td>IC-INTERFACE CMOS 888-BIT</td>
<td>28-480</td>
<td>1820-3384</td>
</tr>
<tr>
<td>U6</td>
<td>1810-0205</td>
<td>1</td>
<td>NETWORK-RES 8-SIP 4.7K OHM X 7</td>
<td>28-480</td>
<td>1810-0205</td>
</tr>
<tr>
<td>U7</td>
<td>85620-80021</td>
<td>1</td>
<td>EPROM-PROGRAMMED</td>
<td>28-480</td>
<td>85620-80021</td>
</tr>
<tr>
<td>U8</td>
<td>1810-0205</td>
<td>1</td>
<td>NETWORK-RES 8-SIP 4.7K OHM X 7</td>
<td>28-480</td>
<td>1810-0205</td>
</tr>
<tr>
<td>U9</td>
<td>1820-1199</td>
<td>1</td>
<td>IC INV TTL/LS HEX 1-INP</td>
<td>28-480</td>
<td>1820-1199</td>
</tr>
<tr>
<td>U10</td>
<td>1820-2024</td>
<td>1</td>
<td>IC DRV/TL/L BUS OCTL</td>
<td>28-480</td>
<td>1820-2024</td>
</tr>
<tr>
<td>U11</td>
<td>1826-1858</td>
<td>1</td>
<td>IC PWR MGT-BAT-MGT</td>
<td>28-480</td>
<td>1826-1858</td>
</tr>
<tr>
<td>U12</td>
<td>1820-5497</td>
<td>1</td>
<td>IC-NONVOLATILE MEMORY CONTROLLER</td>
<td>28-480</td>
<td>1820-5497</td>
</tr>
<tr>
<td>U13</td>
<td>1820-3465</td>
<td>1</td>
<td>IC FF TTL/ALS D-TYPE POS-EDGE-TRIG COM</td>
<td>28-480</td>
<td>1820-3465</td>
</tr>
<tr>
<td>U14</td>
<td>1906-0229</td>
<td>1</td>
<td>DIODE-ARRAY 50V 400MA TO-116</td>
<td>28-480</td>
<td>1906-0229</td>
</tr>
<tr>
<td>W1</td>
<td>8150-0005</td>
<td>1</td>
<td>RESISTOR 0 CWM</td>
<td>28-480</td>
<td>8150-0005</td>
</tr>
<tr>
<td>XBT1</td>
<td>1400-2111</td>
<td>1</td>
<td>HOLDER-BAT FOR 1 A CELLS .72-WD</td>
<td>05-335</td>
<td>1029</td>
</tr>
</tbody>
</table>

*Indicates factory-selected value*
NOTES: UNLESS OTHERWISE SPECIFIED

1. DIMENSIONS ARE IN INCHES.
2. INSTALL MP4 AND MP5 FROM FARSIDE PRIOR TO WAVE SOLDER.
3. REMOVE TABS AFTER WAVE SOLDER, TRIM TABS FLUSH TO BOARD.
4. MAXIMUM COMPONENT HEIGHT __7/32__.
5. MAXIMUM LEAD TRIM LENGTH __1/8__.
6. BEND GROUND TAB AND HAND SOLDER AS SHOWN, (FARSIDE).