User's Guide

EDFA Test Personality for HP 71450B/1B/2B

HP Part No. 70952-90001
Edition 1
Printed in USA June 1995

1400 Fountaingrove Parkway,
Santa Rosa, CA
95403-1799, USA

Safety Notes

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

CAUTION

The *caution* note 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.

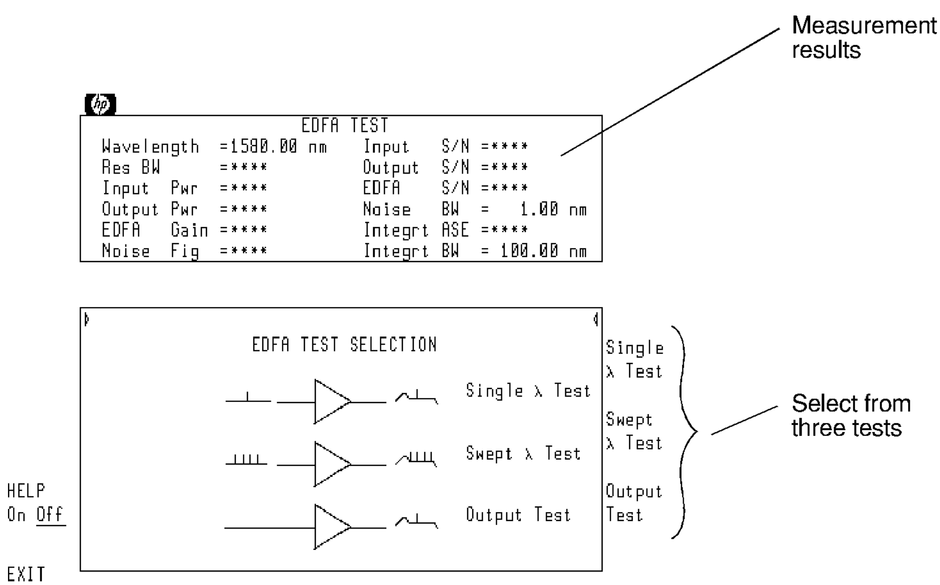
WARNING

The *warning* note 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.

EDFA Analysis with the EDFA Test Personality

The EDFA test personality performs measurements on erbium-doped fiber amplifiers. It uses the interpolation source-subtraction method and measures the following parameters:

- Source wavelength.
- Gain versus wavelength.
- Noise figure versus wavelength.
- Input and output power versus wavelength.
- Input, output, and EDFA signal-to-noise ratios versus wavelength.
- Input, output, and EDFA noise versus wavelength.
- Integrated amplified spontaneous emission (ASE) over wavelength.

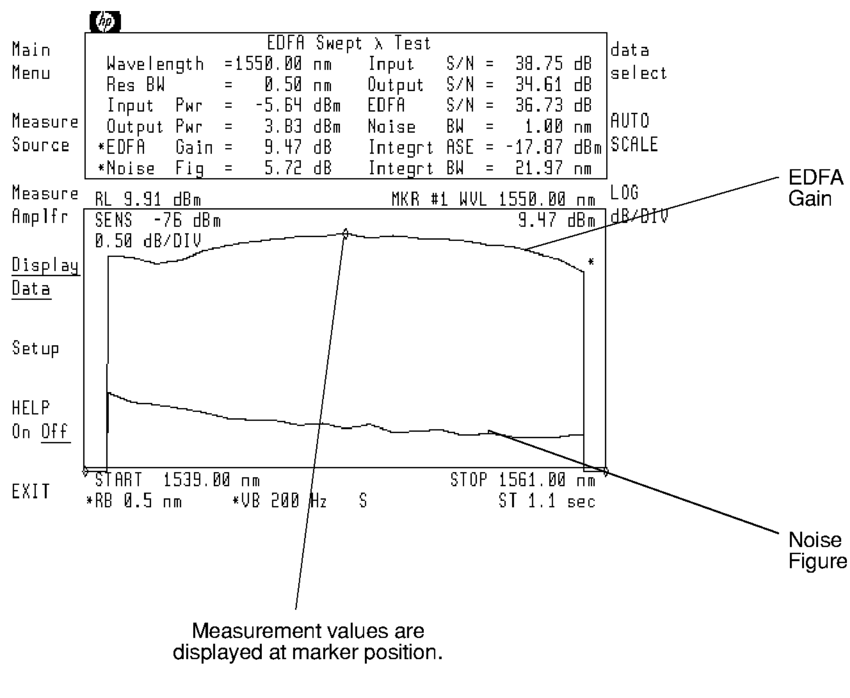


The personality's Main Menu.

The personality is installed on HP 71452B and HP 71450B/1B Option 051 optical spectrum analyzers. It can also be loaded on HP 71450A/1A optical spectrum analyzers. If not already installed, install the program as described in Chapter 1. Once installed, you can start the program by pressing (USER) and then the EDFA softkey.

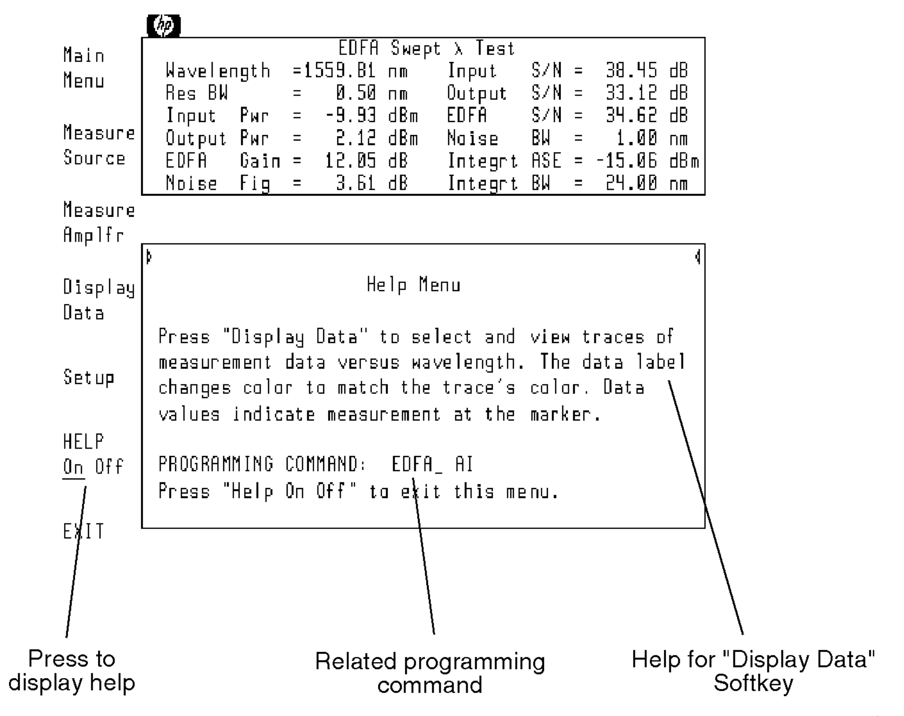
Swept measurement results are displayed graphically

If a tunable laser source is available, swept wavelength testing can be performed. HP 8168B and 8168C tunable laser sources are recommended. The modulation output of the tunable laser is used to trigger the sweeps on the optical spectrum analyzer. As shown in the following figure, the personality graphically displays swept measurement results.



Traces of Gain and Noise Figure versus wavelength.

On-line help is available A help menu is always available by pressing the `HELP On Off` softkey. Once this menu is displayed, pressing any other softkey displays information about that softkey. The information includes an equivalent programming command if available. To remove the help menu, simply press `HELP On Off` so that `Off` is underlined.



Help Menu

Ensuring the greatest accuracy

To ensure the greatest possible accuracy, observe the following:

- Maintain a resolution bandwidth $\geq$ 0.5 nm for best amplitude accuracy.
- Perform a user calibration of the optical spectrum analyzer as described in Chapter 2.
- Optical isolator reduces reflections.

  Inserting isolators at the input and output of the erbium-doped fiber amplifier reduces the effect of external reflections on amplifier performance. Many amplifiers have built-in isolators.

- Clean connections are required for best accuracy. When connecting fiber-optic cables, do the following steps:
  1. Clean the connectors as described in Chapter 1 of the *HP 71450B/1B/2B Optical Spectrum Analyzers User's Guide*. Dry connections are recommended.
  2. Align connectors so that the fiber end does not touch the outside of the mating connector. Do not rub the fiber end against any other surfaces.
  3. After the ferrule is properly seated inside the connector, use one hand to keep it straight and finger-tighten it with the other hand. Refer to the manufacturer's data sheet for torque recommendations. Overtightening or undertightening connectors can result in misalignment and nonrepeatable connections.

- Fusion splices reduce measurement uncertainty.

  The largest source of measurement uncertainty in EDFA measurements is repeatability of fiber-optic connections. Fusion splices reduce typical connector uncertainties of ±0.25 dB to a ±0.05 dB splice uncertainty.

Control program using softkeys or programming commands

The EDFA test personality can be controlled either via softkey menus or remote programming commands. Softkeys are the seven buttons located on each side of the screen. The functions of softkeys change according to the menus displayed on the screen. Generally, left-side softkeys access major menus. Refer to Chapter 3 for a definition of the programming commands.

Use the following three front-panel keys on the HP 70004A display to select the available softkey menus:

- Press (USER) to view EDFA test personality menus. The personality must first be installed as described in Chapter 1.
- Press (MENU) to view optical spectrum analyzer menus.
- Press (DISPLAY) to view HP 70004A display menus.

Key Conventions

The following key conventions are used in this guide:

(Front-panel key)	Text shown like this represents a key physically located on the spectrum analyzer.
`Softkey`	Text shown like this represents a softkey. (The softkeys are located next to the softkey labels, and the softkey labels are the annotation on the right or left side of the spectrum analyzer display.)
`Screen Text`	Text printed in this typeface indicates text displayed on the instrument's screen.

Regulatory Information

The specifications and characteristics chapter of the *HP 71450B/1B/2B Optical Spectrum Analyzers Reference* contains regulatory information.

Hewlett-Packard Software ProductLicense Agreement

Important

Please carefully read this License Agreement before opening the media envelope or operating the equipment. Rights in the software are offered only on the condition that the Customer agrees to all terms and conditions of the License Agreement. Opening the media envelope or operating the equipment indicates your acceptance of these terms and conditions. If you do not agree to the License Agreement, you may return the unopened package for a full refund.

In return for payment of the applicable fee, Hewlett-Packard grants the Customer a license in the software, until terminated, subject to the following:

Use

- Customer may use the software on one spectrum-analyzer instrument.
- Customer may not reverse assemble or decompile the software.

Copies and Adaptations

- Customer may make copies or adaptations of the software:
  - For archival purposes, or
  - When copying or adaptation is an essential step in the use of the software with a computer so long as the copies and adaptations are used in no other manner.
- Customer has no other rights to copy unless they acquire an appropriate license to reproduce which is available from Hewlett-Packard for some software.
- Customer agrees that no warranty, free installation, or free training is provided by Hewlett-Packard for any copies or adaptations made by Customer.
- All copies and adaptations of the software must bear the copyright notices(s) contained in or on the original.

Ownership

- Customer agrees that they do not have any title or ownership of the software, other than ownership of the physical media.
- Customer acknowledges and agrees that the software is copyrighted and protected under the copyright laws.
- Customer acknowledges and agrees that the software may have been developed by a third party software supplier named in the copyright notice(s) included with the software, who shall be authorized to hold the Customer responsible for any copyright infringement or violation of this License Agreement.

Transfer of Rights in Software

- Customer may transfer rights in the software to a third party only as part of the transfer of all their rights and only if Customer obtains the prior agreement of the third party to be bound by the terms of this License Agreement.
- Upon such a transfer, Customer agrees that their rights in the software are terminated and that they will either destroy their copies and adaptations or deliver them to the third party.
- Transfer to a U.S. government department or agency or to a prime or lower tier contractor in connection with a U.S. government contract shall be made only upon their prior written agreement to terms required by Hewlett-Packard.

Sublicensing and Distribution

- Customer may not sublicense the software or distribute copies or adaptations of the software to the public in physical media or by telecommunication without the prior written consent of Hewlett-Packard.

Termination

- Hewlett-Packard may terminate this software license for failure to comply with any of these terms provided Hewlett-Packard has requested Customer to cure the failure and Customer has failed to do so within thirty (30) days of such notice.

Updates and Upgrades

- Customer agrees that the software does not include future updates and upgrades which may be available for HP under a separate support agreement.

Export

- Customer agrees not to export or re-export the software or any copy or adaptation in violation of the U.S. Export Administration regulations or other applicable regulations.

Limited Warranty

Software

Hewlett-Packard warrants for a period of 1 year from the date of purchase that the software product will execute its programming instructions when properly installed on the spectrum-analyzer instrument indicated on this package. Hewlett-Packard does not warrant that the operation of the software will be uninterrupted or error free. In the event that this software product fails to execute its programming instructions during the warranty period, customer's remedy shall be to return the measurement card ("media") to Hewlett-Packard for replacement. Should Hewlett-Packard be unable to replace the media within a reasonable amount of time, Customer's alternate remedy shall be a refund of the purchase price upon return of the product and all copies.

Media

Hewlett-Packard warrants the media upon which this product is recorded to be free from defects in materials and workmanship under normal use for a period of 1 year from the date of purchase. In the event any media prove to be defective during the warranty period, Customer's remedy shall be to return the media to Hewlett-Packard for replacement. Should Hewlett-Packard be unable to replace the media within a reasonable amount of time, Customer's alternate remedy shall be a refund of the purchase price upon return of the product and all copies.

Notice of Warranty Claims

Customer must notify Hewlett-Packard in writing of any warranty claim not later than thirty (30) days after the expiration of the warranty period.

Limitation of Warranty

Hewlett-Packard makes no other express warranty, whether written or oral, with respect to this product. Any implied warranty of merchantability or fitness is limited to the 1 year duration of this written warranty.

This warranty gives specific legal rights, and Customer may also have other rights which vary from state to state, or province to province.

Exclusive Remedies

The remedies provided above are Customer's sole and exclusive remedies. In no event shall Hewlett-Packard be liable for any direct, indirect, special, incidental, or consequential damages (including lost profit) whether based on warranty, contract, tort, or any other legal theory.

Warranty Service

Warranty service may be obtained from the nearest Hewlett-Packard sales office or other location indicated in the owner's manual or service booklet.

Contents

4. Reference

Index

1

Installing the Personality

Installing the Personality

If you have an HP 71452B optical spectrum analyzer, the EDFA Test personality has already been installed at the factory. Use the procedures in this chapter to install the personality into HP 71450A/0B/1A/1B instruments and to reinstall the program if it is erased from memory. Copies of the program are included in the following forms:

- Memory card.
- 3.5-inch diskette. *(HP-LIF format)*

Do you have an HP 70004A display?

YES Install the personality from the memory card. Refer to "To install from the memory card" in this chapter.

NO Install the personality directly from the an external HP-IB disk drive. Refer to "To install from a 3.5-inch dikette drive" in this chapter.

Battery Power

Once installed, the program is stored in battery powered memory. If the internal battery loses power, the program is erased from memory. With normal use, the internal battery lasts for several years. To replace the internal battery, return the optical spectrum analyzer to a Hewlett Packard service center.

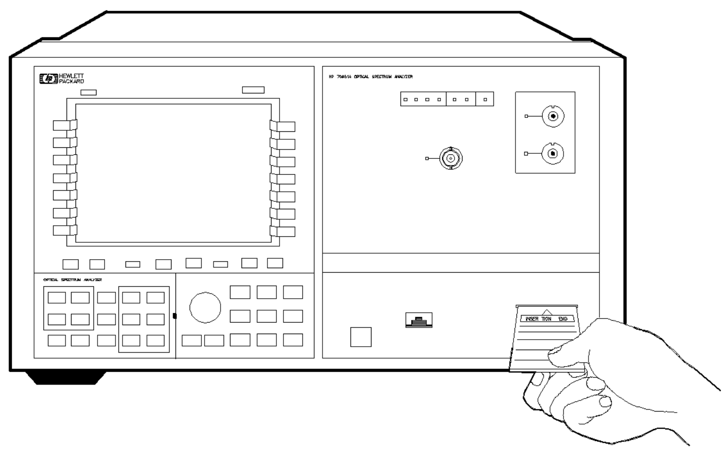
Installing on HP 71450A/1A optical spectrum analyzers

The EDFA test personality can be installed in HP 71450A/1A optical spectrum analyzers. However there are some memory constraints to consider. The optical spectrum analyzer should have 1 MB of memory. This ensures that enough memory is available to simultaneously store additional downloadable programs, personalities, and files. Optical spectrum analyzers with serial number prefix 3246A and earlier that were ordered without option 512 need to increase the memory to 1 MB. Contact Hewlett Packard for information on upgrading these instruments. To determine how much memory your optical spectrum analyzer has, refer to "If a memory error is displayed" in this chapter.

Chapter Contents

To install from the memory card

1. Press (USER). If the softkey EDFA appears, the personality is already installed. Continue with Chapter 2 to learn how to perform measurements.
2. Locate the memory card containing the program.
3. Locate the arrow printed on one end of the card.
4. Insert the card into the HP 70004A display's front-panel card slot. Match the card's arrow with the arrow printed above the card slot.



pkb1

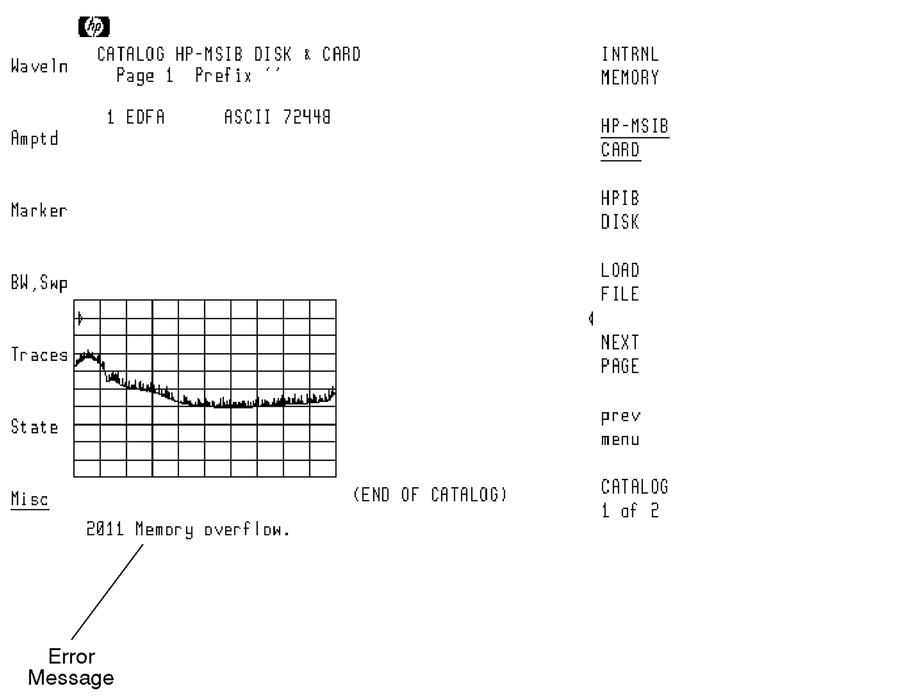
5. Press (DISPLAY) and then the left-side Mass Storage softkey.
6. Press msi and then MEMORY CARD .
7. Press (MENU) and then the left-side Misc softkey.
8. Press MORE 1 of 3 and then catalog & MSI .
9. Press HP-MSIB CARD to display all files contained on the memory card.

If `2053 Storage device error` is displayed, either the card is missing or the card's HP-MSIB address is listed incorrectly. The HP-MSIB address for the card is the same address as the display's HP-IB address and is normally set to 4. If the address is not correct, enter the correct address using the numeric keypad.

10. Press the `LOAD FILE` softkey.
11. Use the numeric keypad to enter the `EDFA` file number. (Normally, this means pressing (1) and then `ENTER`.)

    The front-panel LED next to the card slot lights indicating that the file is being copied into the (USER) menu.

12. If the error message shown in the following figure is displayed, there is not enough free memory in the optical spectrum analyzer to load the program. Refer to "If a memory error is displayed" in this chapter.



Error message displayed.

To install from a 3.5-inch diskette drive

When copying programs directly from an external diskette drive, the drive must be a 3.5 inch, CS80-compatible drive, such as an HP 9122. The diskette containing the EDFA advanced-measurement program uses the LIF format.

1. Press USER. If the softkey `EDFA` appears, the EDFA test personality is already installed. Continue with Chapter 2 to learn how to perform measurements.
2. Connect a 3.5-inch diskette drive to the MMS mainframe that contains the optical spectrum analyzer module. Use an HP-IB cable.
3. Insert the diskette containing the program into the diskette drive.
4. Press DISPLAY and then the left-side `Mass Storage` softkey.
5. Press `msi` and then `HPIB DISK`. Ignore any message such as `Unable to read device.`
6. Press MENU and then the left-side `Misc` softkey.
7. Press `MORE 1 of 3` and then `catalog & MSI`.
8. Press `HPIB DISK`, and use the numeric keypad to enter the diskette drive's address.

   New addresses are entered in the following form: *A.UV*

   where:

   A is a digit from 0 to 7, representing the drive's HP-IB address.

   U is a digit from 0 to 9, representing the unit number. The unit number is typically 0 or 1 and refers to an individual disk drive slot. The default value is 0.

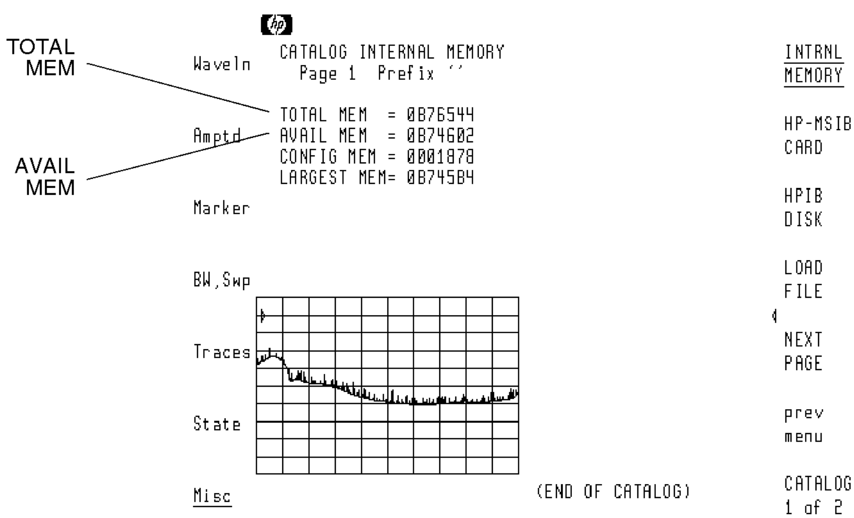
   V is a digit from 0 to 9, representing the volume number. Volume numbers are used for hard disk drives. So, for reading diskettes, the volume number should be 0. The default value is 0.

   For example, entering *3.00* indicates an address of 3, a unit number of 0 and a volume number of 0. This accesses a diskette in the left drive of an external diskette drive at HP-IB address 3.

9. Press the ENTER softkey.
10. Press the LOAD FILE softkey.
11. Use the numeric keypad to enter the EDFA file number. (Normally, this means pressing 1 and then ENTER.)
12. If the error message 2011 Memory overflow is displayed, there is not enough free memory in the optical spectrum analyzer to load the program. Refer to "If a memory error is displayed" in this chapter.

If a memory error is displayed

1. Press (MENU) and then the left-side **Misc** softkey.
2. Press **MORE 1 of 3** and then **catalog & MSI**.
3. Press **INTRNL MEMORY**. The following figure shows a typical display. Notice the `TOTAL MEM` and `AVAIL MEM` listings. You must increase `AVAIL MEM` (available memory) by deleting existing files to free additional memory. Or, if you have an HP 71450A/1A optical spectrum analyer, and `TOTAL MEM` is approximately 131000 bytes, contact Hewlett Packard.



smallmem

Optical Spectrum Analyzer Memory Configuration

2

Performing Measurements

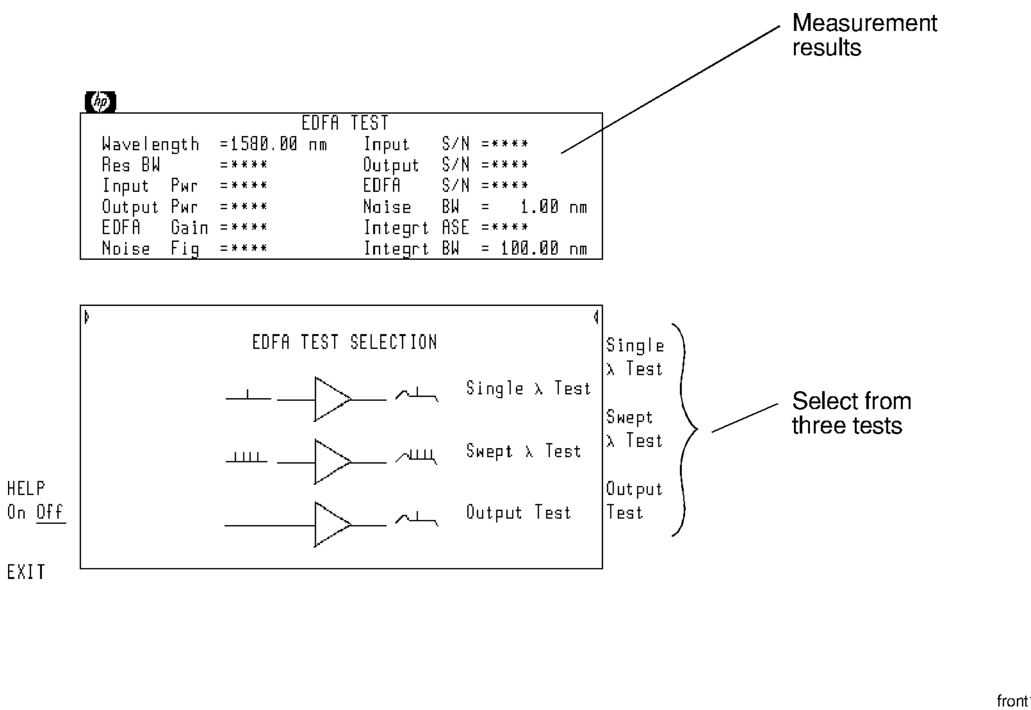
Performing Measurements

In this chapter, you'll learn how to characterize erbium-doped fiber amplifiers. The EDFA test personality provides three separate testing modes:

Single λ Test — measures EDFA noise figure and gain at a single wavelength.

Swept λ Test — measures EDFA noise figure and gain over a wavelength range using a tunable laser source.

Output Test — measures signal, noise, and signal-to-noise ratio at the output of an EDFA or in a system. Use Output Test in situations when total system performance is required.



Select the tests from the Main Menu.

To get the most from your measurements, be sure to read "Configuring the Program" in this chapter; it describes customizing the measurements via the Setup menu.

Chapter Contents

Calibrating the Optical Spectrum Analyzer

Performing a user calibration on the optical spectrum analyzer ensures maximum wavelength and amplitude accuracy for your EDFA measurements. User calibrations require a stable (amplitude and wavelength) single-frequency laser within the 600 to 1700 nm range. You can access the Calibration menu from the optical spectrum analyzer's `Amptd` menu. (Press (MENU) and then `Amptd`.)

The optical spectrum analyzer's maximum calibration adjustment is about 2 nm in wavelength. If a larger adjustment is attempted, error 2023, `Illegal Cal signal` is displayed.

Guard against changes due to polarization

Because the optical spectrum analyzer is slightly polarization sensitive, this calibration should be performed by persons knowledgeable on the effects of polarization on optical power measurements. During the calibration, the light source's output power is first measured with a power meter. Then, the fiber-optic cable is disconnected from the power meter and connected to the optical spectrum analyzer. Because moving fiber-optic cables changes polarization, the measured value of the output power may vary.

Use a wavelength within the amplifier's range

For optimum results, perform the calibration at a wavelength that is within the range of the amplifier you are testing.

During a calibration, the optical spectrum analyzer defaults expect a signal within the following limits:

Power . . . . -5 dBm ± 5 dB
Wavelength . . . . 1300 ± 2 nm

You must enter the wavelength and amplitude of your calibration source if it is different from these values.

To calibrate the optical spectrum analyzer

Ensuring amplitude accuracy

During this procedure, avoid moving the fiber-optic cables whenever possible. Moving fiber-optic cables changes the polarization of the light which affects power measurements.

The following procedure requires a power meter and a polarization controller. The HP 11896A is the suggested polarization controller. For HP 71450A/1A optical spectrum analyzers perform the steps listed in "To calibrate an HP 71450A/1A" in this chapter.

Measure the source wavelength

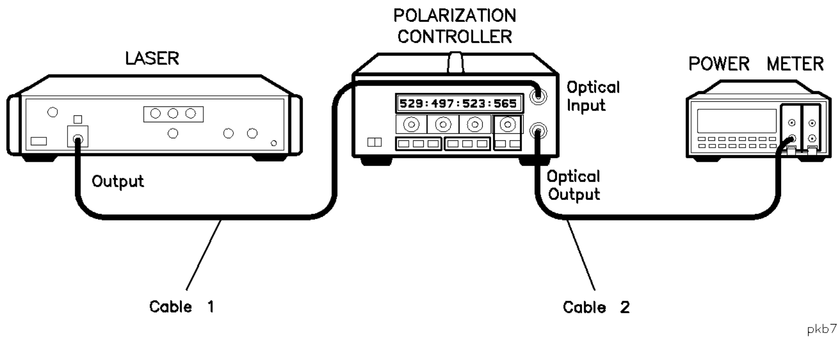
1. Measure the wavelength of a precision single-mode laser using a wavelength meter. Enter the wavelength in the space provided below.

   The laser must have a wavelength between 600 to 1700 nm. Because the optical spectrum analyzer's monochromator is air filled, measure the wavelength as in air.

   wavelength:_______________nm

Measure the average power

2. Connect the laser, polarization controller, and power meter as shown in the following figure.



For accurate measurements

Do not disconnect cable 1 or cable 2 from the laser or polarization controller during the calibration procedure. Maintaining this connection ensures the greatest measurement accuracy.

3. Adjust the polarization controller to achieve the maximum power reading on the power meter. Record the power level on the following line:

   maximum power:_______________dBm

4. Adjust the polarization controller to achieve a minimum power reading on the power meter. Record the power level on the following line:

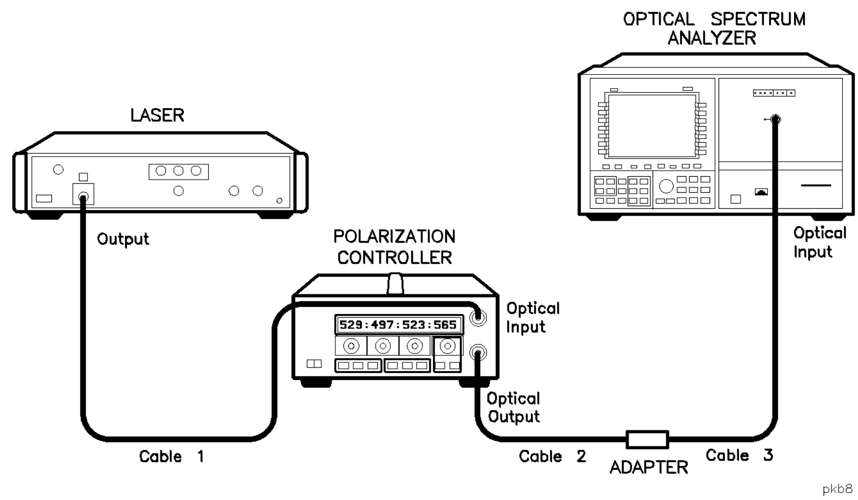
   minimum power:_______________dBm

5. Calculate the average of the two power readings recorded in steps 3 and 4 using the following equation. Record the result below:

$$average\ power = \frac{minimum\ power + maximum\ power}{2}$$

average power:_______________dBm

6. In the test setup, replace the power meter with the optical spectrum analyzer as shown in the following figure. Be sure to use two cables and an adapter to connect the optical spectrum analyzer to the polarization controller.
7. Turn the laser on.



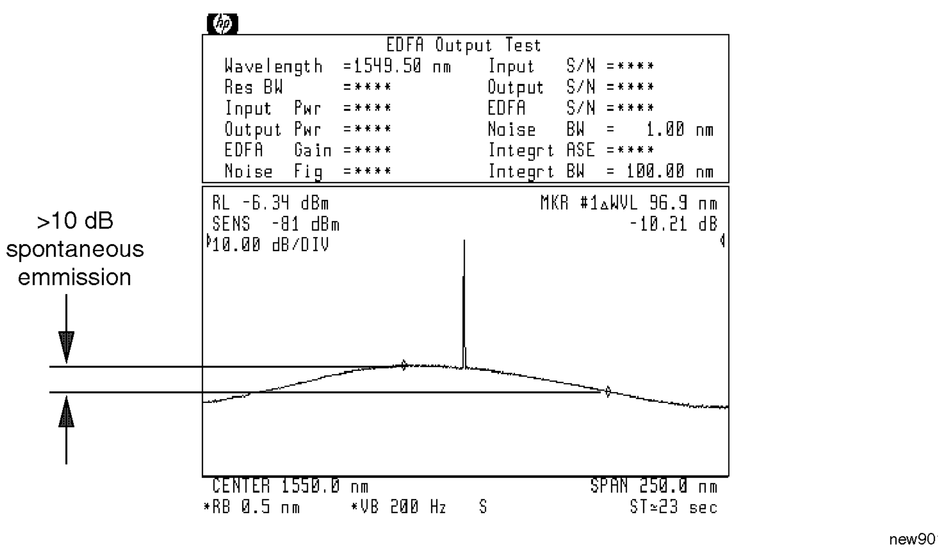
For accurate measurements

Do not disconnect cable 3 from the optical spectrum analyzer during or after the calibration procedure. Maintaining this connection ensures the greatest measurement accuracy, since your optical spectrum analyzer will be calibrated at the free end of cable 3. For fusion splice measurements, the adapter is replaced with a fiber splice.

Determine correction factor for source spontaneous emission

8. Press INSTR PRESET.
9. Press AUTO MEAS to display the laser's response.
10. Press AUTO ALIGN to align the optical spectrum analyzer.
11. Press USER and then `EDFA` to start the EDFA test personality.
12. Press `Output Test`.
13. Locate the peak of the spontaneous emission. Adjust the wavelength span so that, on either side of this peak, 10 dB of spontaneous emmision is visible.

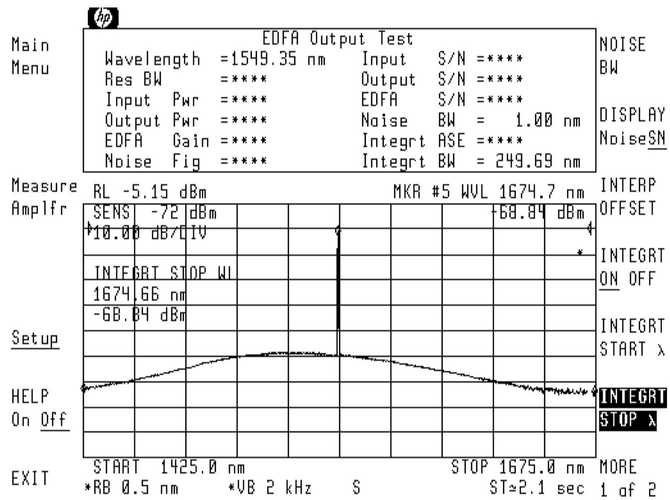
    Use the SPAN key along with the front-panel knob to change the span.



Span adjusted for proper spontaneous emmission levels.

14. Press `Setup`, and then `INTEGRT START λ`.
15. Use the front-panel knob to move the left-integration marker to the left edge of the display.

16. Press `INTEGRT STOP λ`, and use the front-panel knob to move the right-integration marker to the right edge of the display.



17. Press the left-side `Measure Amplr` softkey.

18. When the measurement is complete, record the displayed `Output Pwr` and `Integrt ASE` measurements on the following lines:

`Output Pwr`:______________dBm
`Integrt ASE`:______________dBm

19. Subtract the `Integrt ASE` from the `Output Pwr` to determine the ratio of the two power levels. Record the result on the following line:

$$\Delta\ power\ (dB) = Output\ Pwr\ (dBm) - Integrt\ ASE\ (dBm)$$

Δ power:______________dB

20. Use the following equation to calculate the correction factor for the broadband power meter to the narrow band optical spectrum analyzer measurement:

$$factor = 10\ log\left(1 - \frac{1}{10^{\frac{x}{10}} + 1}\right)$$

where `x` is the Δ power recorded in step 19.

source spontaneous emission correction factor:______________dB

Total the correction factors

21. Power meters generally are calibrated with an open beam of light. By simply using a fiber adapter, they will measure the optical power in the beam emerging from the fiber end. This is lower than the power in the fiber by 3.6%. If this is the case, enter a correction factor of +0.16 dB. If your power meter has already accounted for this factor, enter 0 dB.

correction factor:______________dB

22. Add the values from steps 20 and 21. Be sure to keep track of the sign of each number.

total correction factor:______________dB

Calculate corrected power level

23. Add the value recorded in step 22 to the average power recorded in step 5. Record the result on the following line.

corrected power level:______________dBm

Measure the average power on OSA

24. Press (PEAK SEARCH) and then (TO CENTER).
25. Press (SPAN), and enter a wavelength span of 1 nm.
26. Press (PEAK SEARCH) and then (TO CENTER).
27. Press (SPAN), and enter a wavelength span of 0 nm.
28. Press (MENU).
29. Press the left-side `Amptd` softkey.
30. Press `LOG dB/DIV`, (1), and `dB` to select a 1 dB logarithmic amplitude scale.
31. Press the left-side `BW,Swp` softkey.
32. Press `SWPTIME AutoMan`, (1), (0), and then `s`.
33. Adjust the polarization controller to "peak" the trace displayed on the optical spectrum analyzer.

    This sets the polarization for a maximum power reading.
34. Press (PEAK SEARCH), and record the power level indicated by the marker:

maximum power:______________dBm

35. Press (Δ).
36. Adjust the polarization controller to "dip" the trace displayed on the optical spectrum analyzer to a minimum power value.
37. Record the absolute value of the Δ marker's power ratio on the following line:

Δ power: ______________ dBm

38. Use the values recorded in steps 34 and 37 to calculate the average displayed power as shown in the following equation. Record the result below:

$$average\ power = maximum\ power - \frac{\Delta\ power}{2}$$

average power: ______________

39. Press the left-side `Amptd` softkey.
40. Press `MORE 1 of 4`, `MORE 2 of 4`, and then `A METER On Off` so that `On` is underlined.
41. Adjust the polarization controller to achieve a displayed Amplitude Meter power level equal to the power level recorded in step 38.

Ensuring amplitude accuracy

It is very important not to move the fiber-optic cables during the remaining steps of this procedure.

Enter the calibration power and wavelength

42. Press the left-side `Waveln` softkey. Then, press `MORE 1 of 2`, `cal menu`, `cal setup`, and then `POWER FOR CAL`.
43. Use the numeric keypad to enter the corrected power level calculated in step 23.
44. Press `WAVELEN FOR CAL`. Enter the wavelength recorded in step 1. Then, press `prev menu` and then `CAL ALL`.

To calibrate an HP 71450A/1A

Ensuring amplitude accuracy

During this procedure, avoid moving the fiber-optic cables whenever possible. Moving fiber-optic cables changes the polarization of the light which affects power measurements.

The following procedure requires a power meter and a polarization controller. The HP 11896A is the suggested polarization controller.

The optical spectrum analyzer's ROM version is used to select which calibration procedure should be performed on the instrument.

Determine the ROM version

1. Press the left-side `Misc` softkey to enter the Miscellaneous menu.
2. Press the `MORE 1 of 3` and then `service` softkeys.
3. Press the `ROM VERSION` softkey. The version of the ROM is displayed in the upper right corner of the display.
   - If the ROM version is `B.05.00` or above, do not perform this procedure. Instead, perform the steps in "To calibrate the optical spectrum analyzer" in this chapter.
   - If the ROM version is `B.04.04` or below, continue with this procedure.

Measure the source wavelength

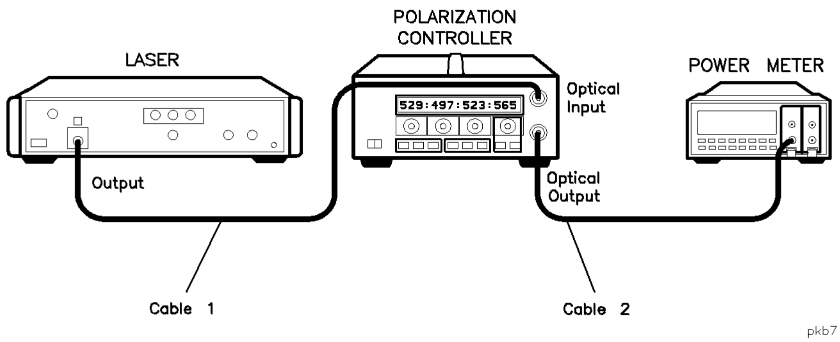
4. Measure the wavelength of a precision single-mode laser using a wavelength meter. Enter the wavelength in the space provided below.

   The laser must have a wavelength between 600 to 1700 nm. Because the optical spectrum analyzer's monochromator is air filled, measure the wavelength as in air.

   wavelength:______________nm

Measure the average power

5. Connect the laser, polarization controller, and power meter as shown in the following figure.



For accurate measurements

Do not disconnect cable 1 or cable 2 from the laser or polarization controller during the calibration procedure. Maintaining this connection ensures the greatest measurement accuracy.

6. Adjust the polarization controller to achieve the maximum power reading on the power meter. Record the power level on the following line:

maximum power:______________dBm

7. Adjust the polarization controller to achieve a minimum power reading on the power meter. Record the power level on the following line:

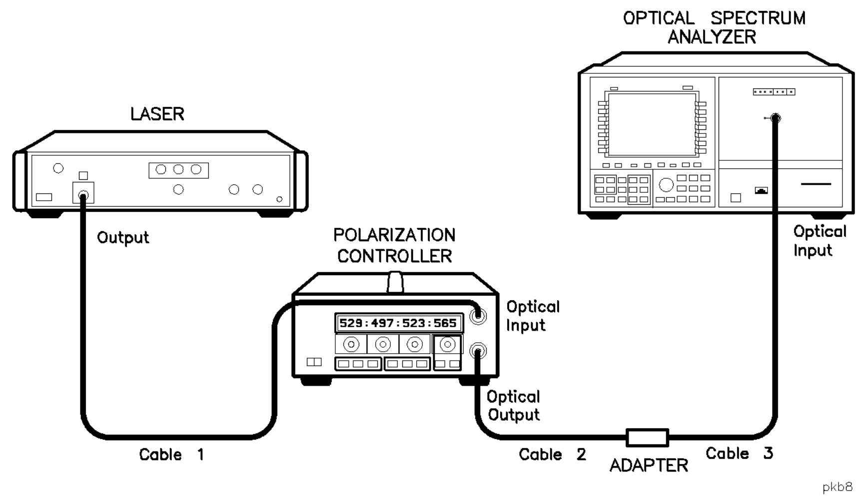
minimum power:______________dBm

8. Calculate the average of the two power readings recorded in steps 6 and 7 using the following equation. Record the result below:

$$average\ power = \frac{minimum\ power + maximum\ power}{2}$$

average power:________________dBm

9. In the test setup, replace the power meter with the optical spectrum analyzer as shown in the following figure. Be sure to use two cables and an adapter to connect the optical spectrum analyzer to the polarization controller.
10. Turn the laser on.



For accurate measurements

Do not disconnect cable 3 from the optical spectrum analyzer during or after the calibration procedure. Maintaining this connection ensures the greatest measurement accuracy, since your optical spectrum analyzer will be calibrated at the free end of cable 3. For fusion splice measurements, the adapter is replaced with a fiber splice.

Determine correction factor for 0.2 nm RBW

11. Press AUTO MEAS to display the laser's response.
12. Press AUTO ALIGN to align the optical spectrum analyzer.
13. Press RES BW, and enter a 0.2 nm bandwidth.
14. Record the source amplitude on the following line:

    power (0.2 nm RBW):______________dBm

15. Press RES BW, and enter a 0.5 nm bandwidth.
16. Record the source amplitude on the following line:

    power (0.5 nm RBW):______________dBm

17. Subtract the reading in step 16 from the reading in step 14 and record the result on the following line:

$$correction\ factor = POWER_{step\ 14} - POWER_{step\ 16}$$

0.2 nm RBW correction factor:______________dB

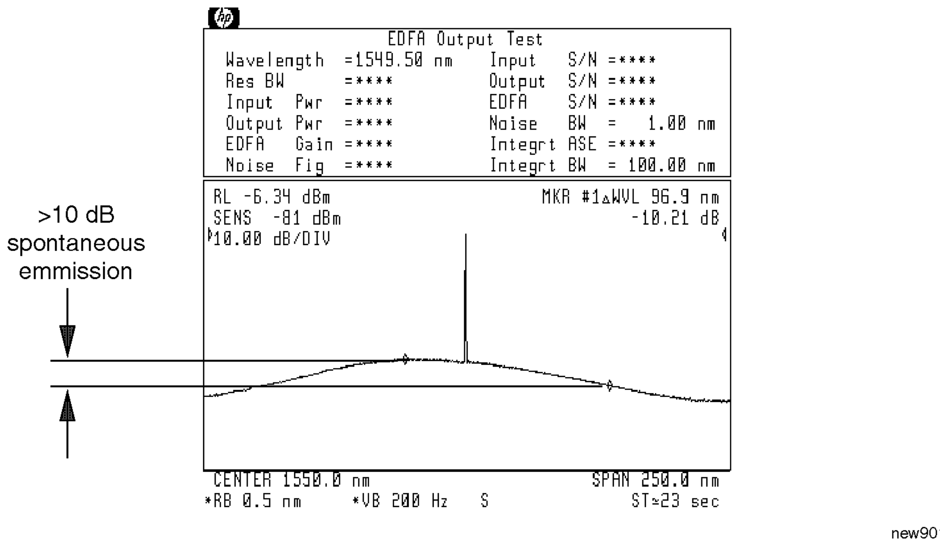
NOTE

This number should be between 0 dB and −0.2 dB.

Determine correction factor for source spontaneous emission

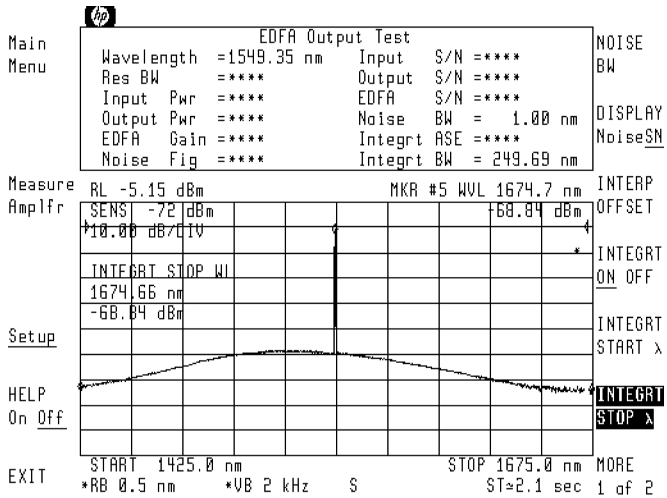
18. Press INSTR PRESET.
19. Press USER and then `EDFA` to start the EDFA test personality.
20. Press `Output Test`.
21. Locate the peak of the spontaneous emission. Adjust the wavelength span so that, on either side of this peak, 10 dB of spontaneous emmision is visible.

    Use the SPAN key along with the front-panel knob to change the span.



Span adjusted for proper spontaneous emmission levels.

22. Press `Setup`, and then `INTEGRT START λ`.
23. Use the front-panel knob to move the left-integration marker to the left edge of the display.
24. Press `INTEGRT STOP λ`, and use the front-panel knob to move the right-integration marker to the right edge of the display.



25. Press `Measure Amplr`.

26. When the measurement is completed, record the displayed **Output Pwr** and **Integrt ASE** measurements on the following lines:

Output Pwr:______________dBm
Integrt ASE:______________dBm

27. Subtract the **Integrt ASE** from the **Output Pwr** to determine the ratio of the two power levels. Record the result on the following line:

$$\Delta\, power\ (dB) = Output\ Pwr\ (dBm) - Integrt\ ASE\ (dBm)$$

Δ power:______________dB

28. Use the following equation to calculate the correction factor for the broadband power meter to the narrow band optical spectrum analyzer measurement:

$$factor = 10\ log\left(1 - \frac{1}{10^{\frac{x}{10}} + 1}\right)$$

where **x** is the Δ power recorded in step 27.

source spontaneous emission correction factor:______________dB

Total the correction factors

29. Power meters generally are calibrated with an open beam of light. By simply using a fiber adapter, they will measure the optical power in the beam emerging from the fiber end. This is lower than the power in the fiber by 3.6%. If this is the case, enter a correction factor of +0.16 dB. If your power meter has already accounted for this factor, enter 0 dB.

    correction factor:_______________dB

30. Add the values from steps 17, 28, and 29. Be sure to keep track of the sign of each number.

    total correction factor:_______________dB

Calculate corrected power level

31. Add the value recorded in step 30 to the average power recorded in step 8. Record the result on the following line.

    corrected power level:_______________dBm

Measure the average power on OSA

32. Press (PEAK SEARCH) and then (TO CENTER).
33. Press (SPAN), and enter a wavelength span of 1 nm.
34. Press (PEAK SEARCH) and then (TO CENTER).
35. Press (SPAN), and enter a wavelength span of 0 nm.
36. Press (MENU).
37. Press the left-side `Amptd` softkey.
38. Press `LOG dB/DIV`, (1), and `dB` to select a 1 dB logarithmic amplitude scale.
39. Press the left-side `BW,Swp` softkey.
40. Press `SWPTIME AutoMan`, (1), (0), and then `s`.
41. Adjust the polarization controller to "peak" the trace displayed on the optical spectrum analyzer.

    This sets the polarization for a maximum power reading.

42. Press (PEAK SEARCH), and record the power level indicated by the marker:

    maximum power:_______________dBm

43. Press (Δ).

44. Adjust the polarization controller to "dip" the trace displayed on the optical spectrum analyzer to a minimum power value.

45. Record the absolute value of the Δ marker's power ratio on the following line:

Δ power: ______________ dBm

46. Use the values recorded in steps 42 and 45 to calculate the average displayed power as shown in the following equation. Record the result below:

$$average\ power = maximum\ power - \frac{\Delta\ power}{2}$$

average power: ______________

47. Press the left-side `Amptd` softkey.

48. Press `MORE 1 of 4`, `MORE 2 of 4`, and then `A METER On Off` so that `On` is underlined.

49. Adjust the polarization controller to achieve a displayed Amplitude Meter power level equal to the power level recorded in step 46.

Ensuring amplitude accuracy

It is very important not to move the fiber-optic cables during the remaining steps of this procedure.

Enter the calibration power and wavelength

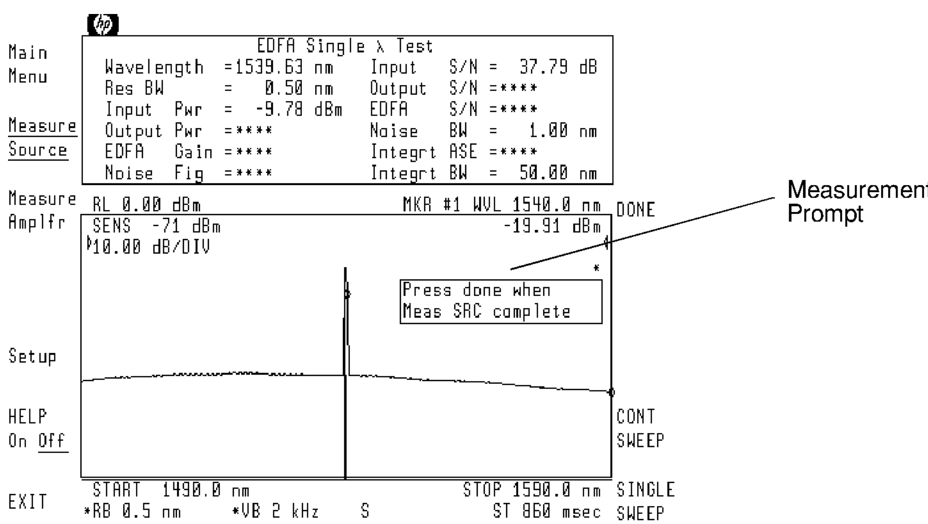
50. Press the left-side `Waveln` softkey. Then, press `MORE 1 of 2`, `cal menu`, `cal setup`, and then `POWER FOR CAL`.

51. Use the numeric keypad to enter the corrected power level calculated in step 31.

52. Press `WAVELEN FOR CAL`. Enter the wavelength recorded in step 4. Then, press `prev menu` and then `CAL ALL`.

Prompts guide you through the measurements

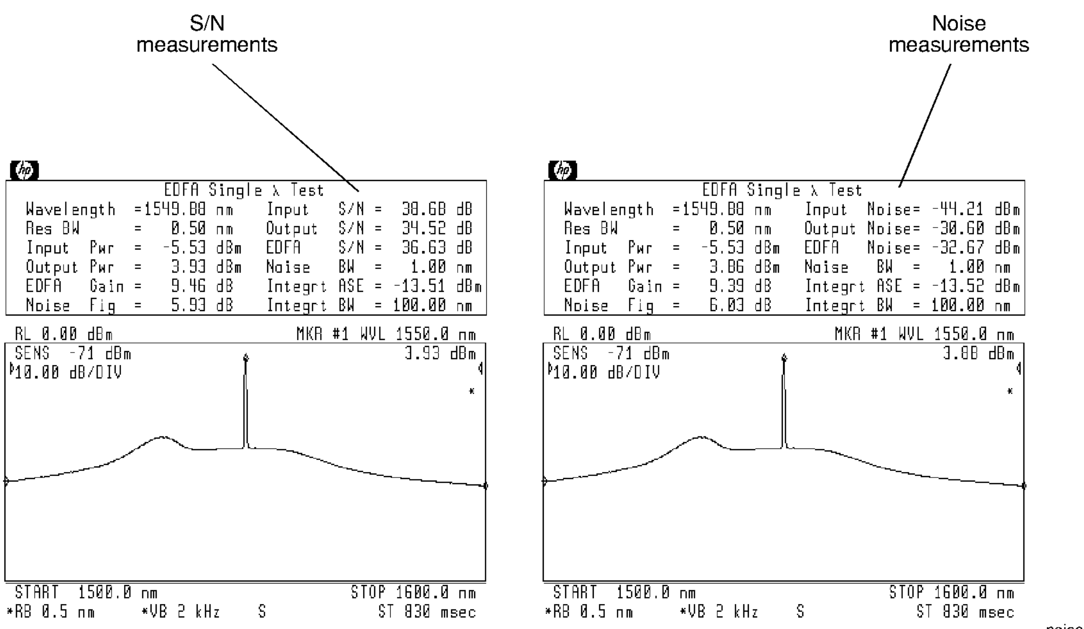
Each testing mode displays messages that guide you through the correct measurement sequence. Generally, the source's output is measured, and then the amplifier's output is measured. (The exception is the Output Test which does not measure the source.) You can disable these prompts at any time by pressing `PROMPT On Off` so that `Off` is underlined. If a prompt message obscures the view of the trace, remove the prompt by pressing (HOLD).



prompt

Display noise values

Normally, the display shows `Input S/N`, `Output S/N`, and `EDFA S/N` measurements. Using the Setup menu's `DISPLAY NoiseSN` softkey, you can change these values to show `Input Noise`, `Output Noise`, and `EDFA Noise`.



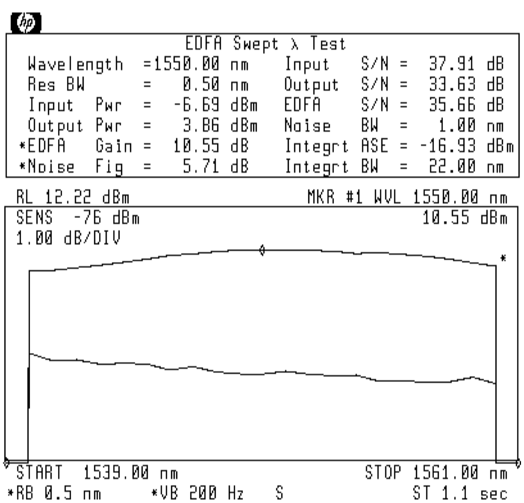
Swept wavelength testing with a tunable source

Swept wavelength testing should be performed using an HP 8168B/C tunable laser source. The optical spectrum analyzer's sweep is triggered from the tunable laser. Pressing HP8168 DWELL displays the amount of time required between wavelength steps on the tunable laser for accurate measurements. Enter this value into the HP 8168B/C. The following cable can be ordered from Hewlett-Packard to connect the HP 8168B/C trigger signal:

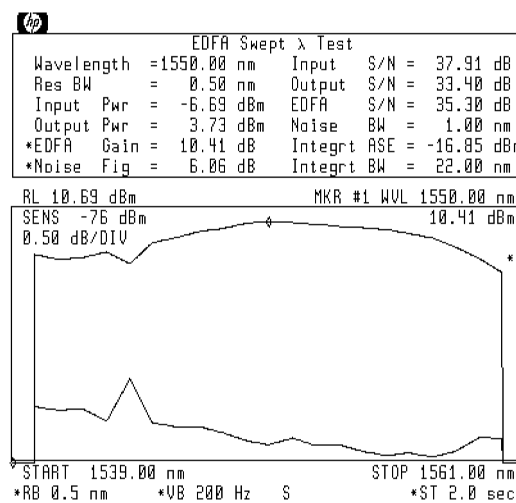
BNC (m) to SMB (f), 122cm (48 in.) . . . . p/n 85680-60093

Avoid inaccurate measurements

Swept wavelength trace results should look similar to the traces displayed in the figure on the left. However, if non-contact fiber optic connections are used, the fiber optic connections are faulty, or the laser is unstable, the displayed traces may have a jagged response as shown in the figure on the right. This jagged response is caused by etalons, or other problems that cause amplitude errors in the signal or noise values.



Accurate display of gain and noise figure.



Jagged response caused by an etalon effect.

To perform a single wavelength test

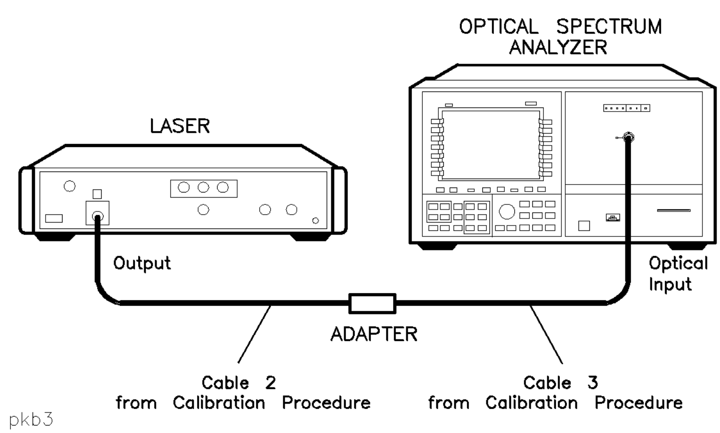
The following measurements are made in this test:

- Wavelength
- EDFA gain
- Noise figure
- Input power
- Output power
- Input S/N *or* input noise
- Output S/N *or* output noise
- EDFA S/N *or* EDFA noise
- Integrated ASE

Procedure

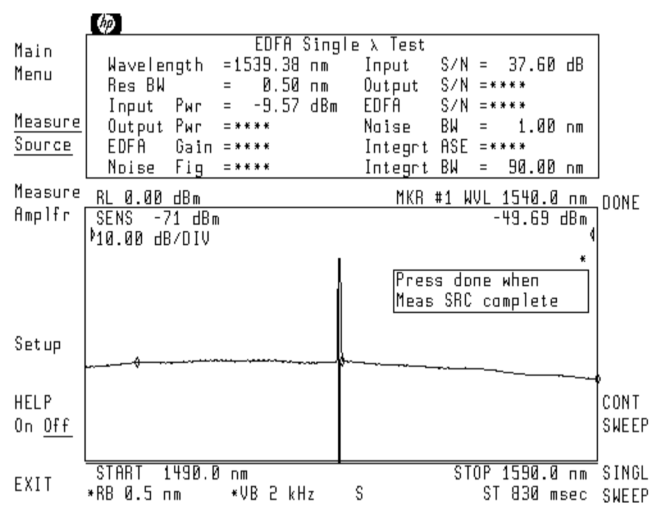
1. Connect the output of the laser to the optical spectrum analyzer as shown in the following figure. The adapter allows you to use the same cables that were used in the calibration procedure. Do not disconnect these cables from the laser or optical spectrum analyzer during the test.

   On HP 71450B/2B optical spectrum analyzers, the input connector is labeled `OPTICAL INPUT`. On HP 71451B optical spectrum analyzers, the input connector is labeled `MONOCHROMATOR INPUT`.



2. Press (USER) and then EDFA to start the EDFA test personality.
3. Press Single λ Test .
4. Perform the following steps if you need to change the tuning range of the optical spectrum analyzer:
   a. Press (START), and enter the starting wavelength.
   b. Press (STOP), and enter the ending wavelength.
   c. Press (HOLD).
5. Press Measure Source .

   The EDFA test personality measures the signal and noise levels at the end of the sweep. Pressing SINGLE SWEEP stops the sweep; each subsequent press of SINGLE SWEEP starts a new sweep. Return to continuous sweeps by pressing CONT SWEEP so that CONT is selected.

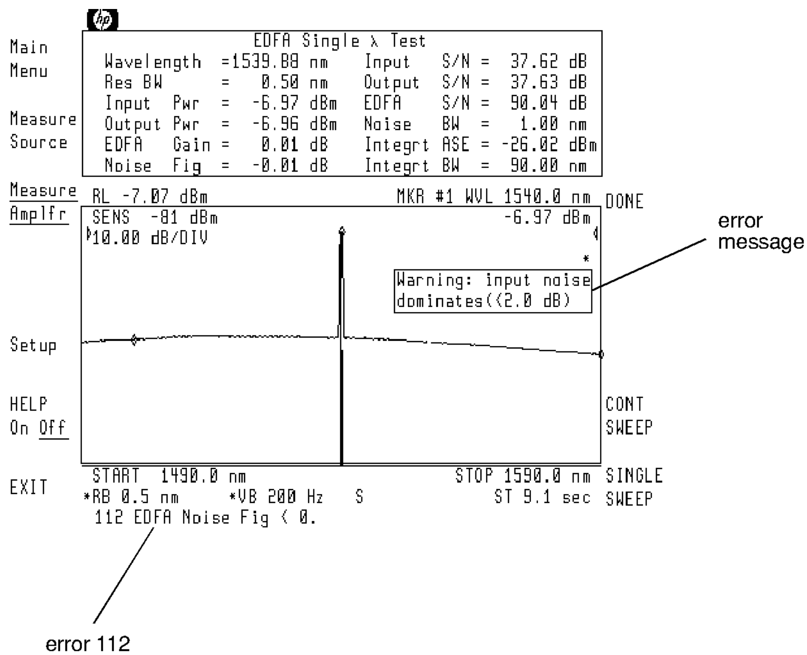


Measurement of the laser source.

6. Allow at least one complete sweep to occur. Then, press DONE .

Error 112 displayed

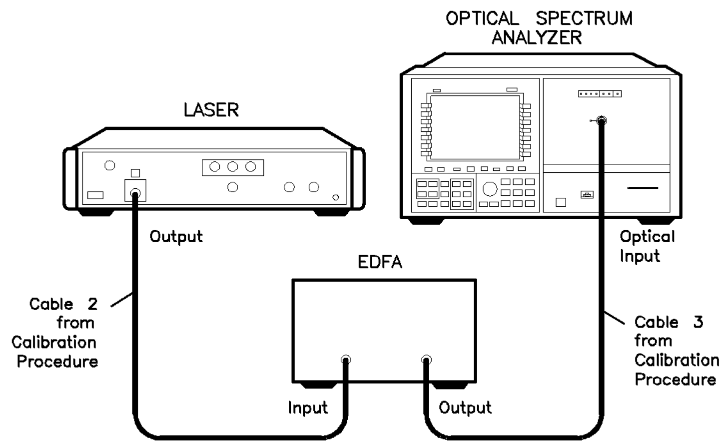
The error messages shown in this figure occur whenever the noise figure calculation results in a negative value. This condition can exist, for example, if you forget to insert the EDFA amplifier before pressing `Measure Amplfr`.



In addition to error 112, this figure shows an error message that appears when the output noise level is dominated by the noise of the source and not the EDFA. The amount of noise contributed by the EDFA is determined by subtracting the input noise, multiplied by the EDFA gain, from the output noise. When the input noise dominates (this message appears when the difference is less than 2 dB), the difference will be small, and the uncertainty potentially large. The error for this part of the measurement is labeled “source spontaneous emission subtraction” in “Measurement Uncertainty” in Chapter 4.

For best accuracy, the input signal-to-noise ratio should be as large as possible. When using a laser with an adjustable bias level for power control, it is best to set the bias for a high power level and use an optical attenuator to achieve the desired source power level.

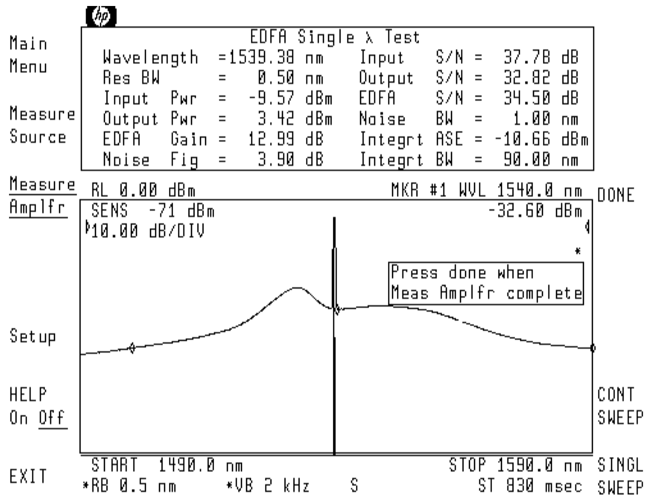
7. Remove the adapter, and connect the EDFA as shown in the following figure.



pkb4

8. Press `Measure Amplfr` . After the first complete sweep occurs, the display lists all the measurement results at the top of the screen. See the figure on the following page.
9. When you are finished with measuring the amplifier, press `DONE` .

Running a Test



Measurement results displayed at top of screen.

To perform a swept wavelength test

The following procedure is meant to be used with an HP 8168B/C tunable laser source. Other sources can be used as explained in the procedure. In this example, the following settings are used:

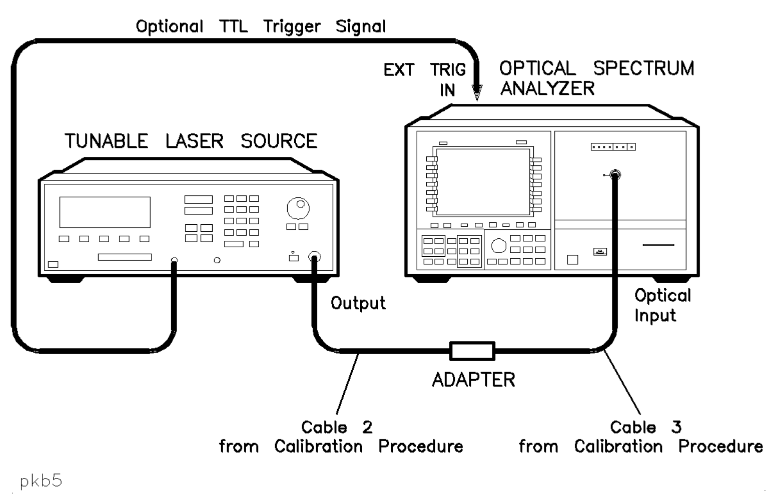
Start wavelength .... 1540 nm
Stop wavelength .... 1560 nm
Step size .... 1 nm
Laser output power .... 200 μw

These settings are for example only. You can use any valid range.

Procedure

1. Connect the output of the HP 8168B/C tunable laser source to the optical spectrum analyzer's input as shown in the following figure. The adapter allows you to use the same cables that were used in the calibration procedure. Do not disconnect these cables from the laser or optical spectrum analyzer.

   On HP 71450B/2B optical spectrum analyzers, the input connector is labeled `OPTICAL INPUT`. On HP 71451B optical spectrum analyzers, the input connector is labeled `MONOCHROMATOR INPUT`.



2. Connect the laser's **Modulation Output** signal to the rear-panel **EXT TRIG IN** connector on the HP 70950B/2B or HP 70951B optical spectrum analyzer module.

   External triggering requires a TTL-compatible signal with a minimum of 0V and a maximum of +5V. For more information on triggering the HP 70950B/1B/2B modules, refer to "Triggering Sweeps" in Chapter 2 of the *HP 71450B/1B/2B Optical Spectrum Analyzers User's Guide*.

3. Press USER and then EDFA to start the EDFA test personality.

Select the test

4. Press Swept λ Test .

5. On the HP 8168B/C, press λ-Sweep , and enter the followng settings:

   Start .... 1540 nm
   Stop .... 1560 nm
   Step .... 1 nm

6. On the optical spectrum analyzer, enter the start and stop wavelength settings:

   START .... 1538 nm
   STOP .... 1562 nm

Setting the wavelength range is critical

The wavelength range on the optical spectrum analyzer is set slightly wider than the range on the tunable laser. The optical spectrum analyzer's range must be increased on both sides by at least the interpolation offset value. (The default interpolation offset value is 1 nm. This value can be changed using the INTERP OFFSET softkey located in the Setup menu.) The interpolation offset is used to calculate noise values.

7. Press HOLD.

8. Press Setup and then MORE 1 of 2 .

9. Press TRIGGER EXT INT so that EXT is underlined.

   This selects external triggering. If you are using a different source that does not provide a trigger signal, leave TRIGGER EXT INT set to INT.

10. Press HP8168 DWELL.

11. On the HP 8168B/C, enter the dwell time listed on the optical spectrum analyzer's display.

    The displayed dwell time is the minimum time required for the optical spectrum analyzer to process the measurement data. If you change any settings that change sweep time or trace length, you should press this key again to check if the dwell time has changed.

Measure the source

12. Press Measure Source.

13. On the HP 8168B/C, press Auto to begin sweeping the source.

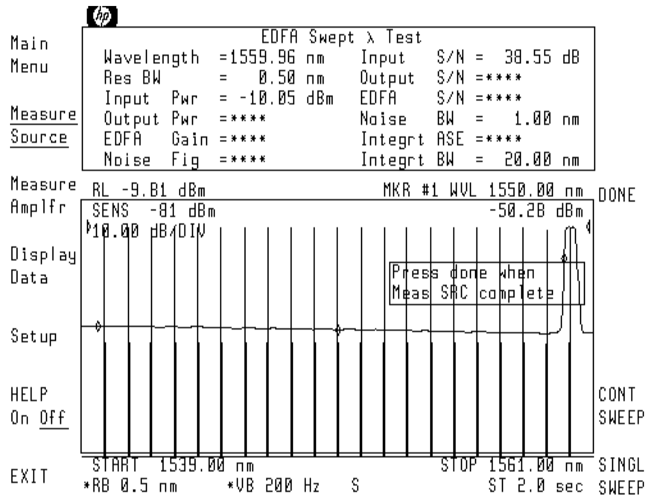
    If a different laser is used that does not provide a trigger signal, press SINGLE SWEEP on the optical spectrum analyzer. Then, repeat the following steps for each wavelength to be measured:

    a. Manually tune the laser to the desired wavelength.

    b. Press SINGLE SWEEP on the optical spectrum analyzer.

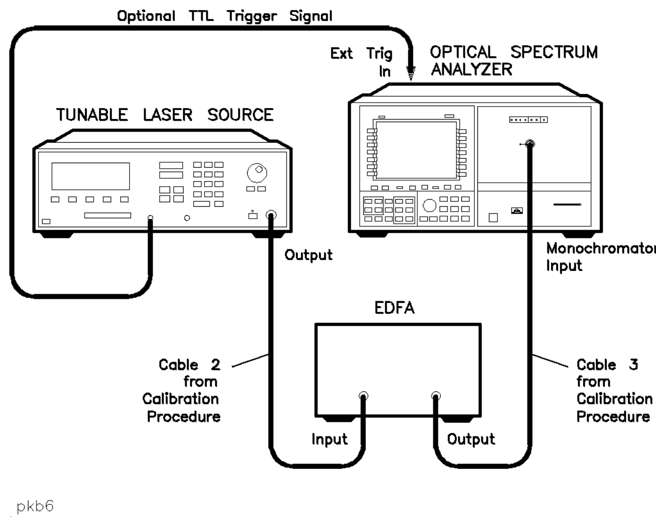
Restarting the test

You can restart testing at any time by resetting the laser and pressing Measure Source. This clears any previously measured data points.



Source measured at each wavelength.

14. Press DONE after the sweep has completed and all the wavelengths are captured.
15. Remove the adapter, and connect the EDFA as shown in the following figure.



Measure the amplifier

16. Press `Measure Amplfr`.

17. On the HP 8168B/C, press `Auto` to begin sweeping the source.

    If a different laser is used that does not provide a trigger signal, press `SINGLE SWEEP` on the optical spectrum analyzer. Then, repeat the following steps for each wavelength to be measured:

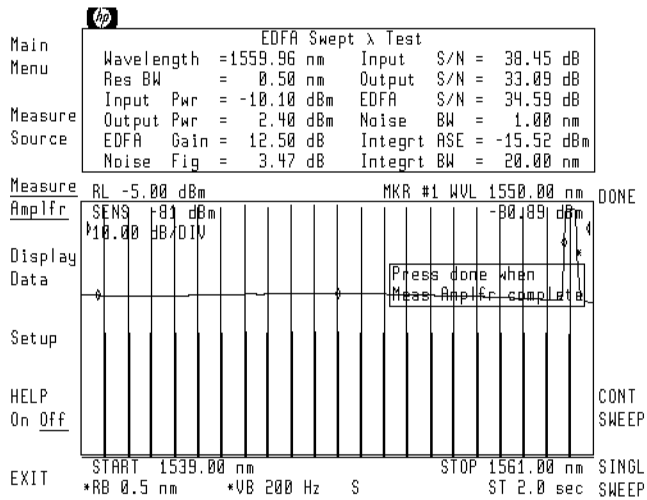
    a. Manually tune the laser to the desired wavelength.

    b. Press `SINGLE SWEEP` on the optical spectrum analyzer.

Restarting the test

You can restart testing at any time by resetting the laser and pressing `Measure Amplfr`. This clears any previously measured amplifier data points. Multiple amplifiers can be tested using the same source measurement data.

Running a Test

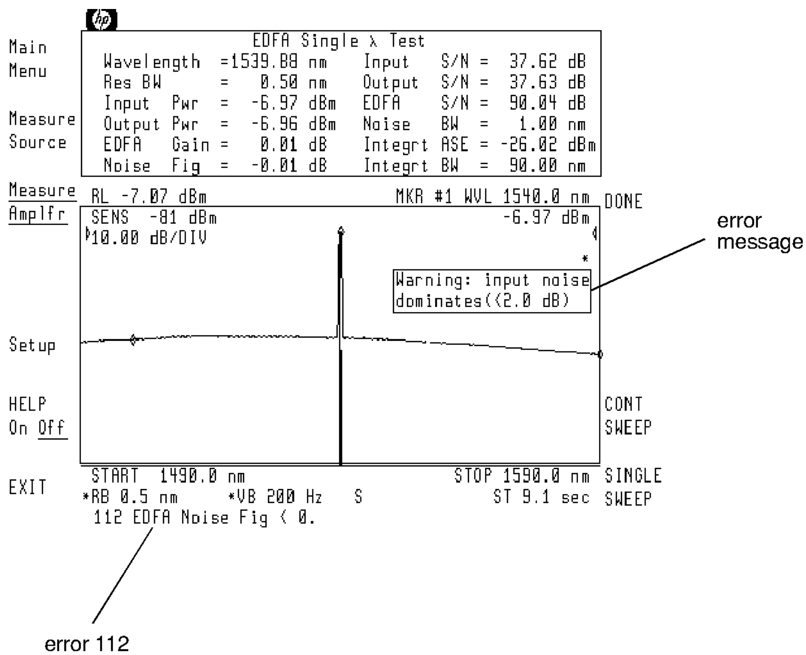


Amplifier output measured at each wavelength.

18. Press DONE after the sweep has completed and all the wavelengths are captured.

Error 112 displayed

The error messages shown in this figure occur whenever the noise figure calculation results in a negative value. This condition can exist, for example, if you forget to insert the EDFA amplifier before pressing `Measure Amplfr` .

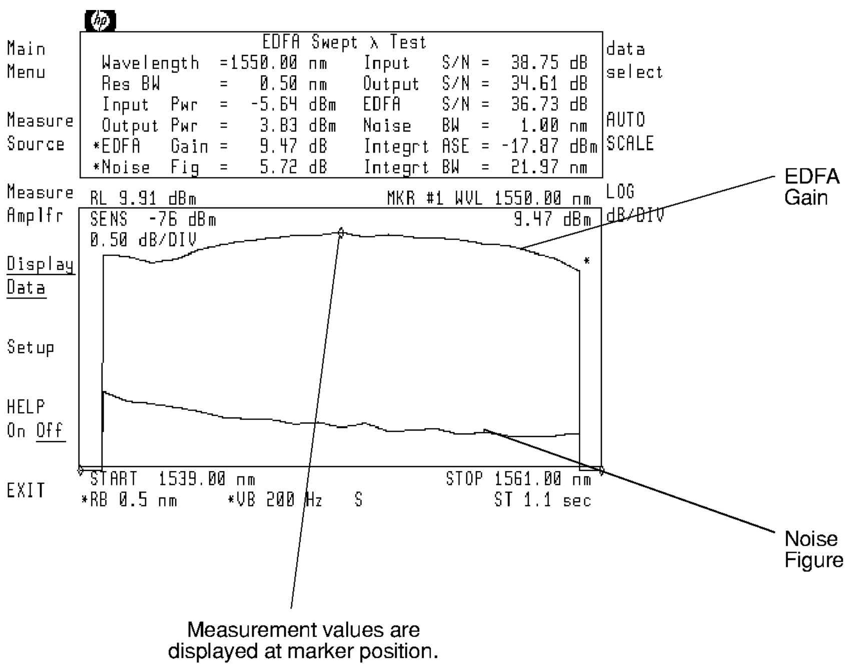


In addition to error 112, this figure shows an error message that appears when the output noise level is dominated by the noise of the source and not the EDFA. The amount of noise contributed by the EDFA is determined by subtracting the input noise, multiplied by the EDFA gain, from the output noise. When the input noise dominates (this message appears when the difference is less than 2 dB), the difference will be small, and the uncertainty potentially large. The error for this part of the measurement is labeled "source spontaneous emission subtraction" in "Measurement Uncertainty" in Chapter 4.

For best accuracy, the input signal-to-noise ratio should be as large as possible. When using a laser with an adjustable bias level for power control, it is best to set the bias for a high power level and use an optical attenuator to achieve the desired source power level.

Display the measurement results

19. Press `Display Data` to view traces of measurement data.
20. Press `data select`.
21. Select from one of the displayed softkeys to view a trace of measurement data versus wavelength. For this example, press `GAIN and NF`.
22. Press `AUTO SCALE` to automatically scale the traces.



sweep3

Color matches trace and measurement

Notice that the measurement values change color to match the color of the displayed trace.

23. If you want to manually scale the display, use LOG dB/DIV . Use the front-panel knob, step keys, or numeric keypad to enter a new value.
24. Press (HOLD) to blank the display of the prompt.
25. Press (NORMAL ON/OFF), and turn the front-panel knob to move the marker. The display shows the data measured at the marker wavelength.

To perform an output test

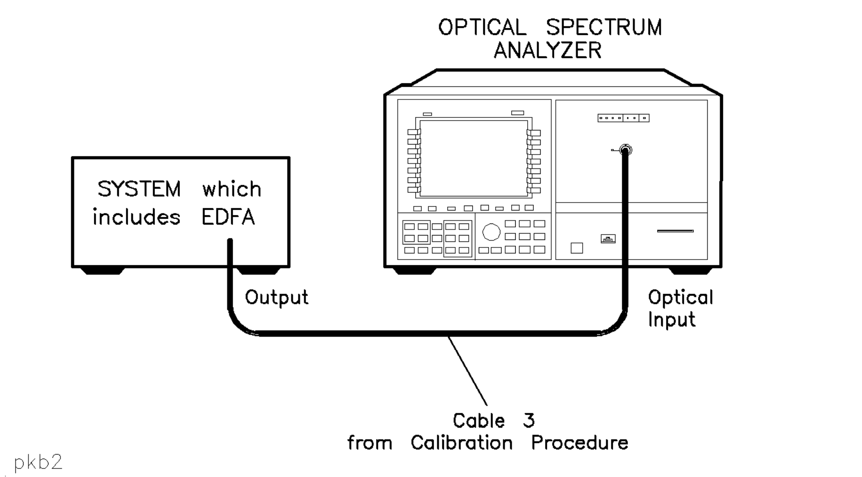
During the output test, the source is not characterized. The following measurements are made in this test:

- Wavelength
- Output power
- Output S/N *or* output noise
- Integrated ASE

Procedure

1. Connect the system with the EDFA to the optical spectrum analyzer as shown in the following figure.

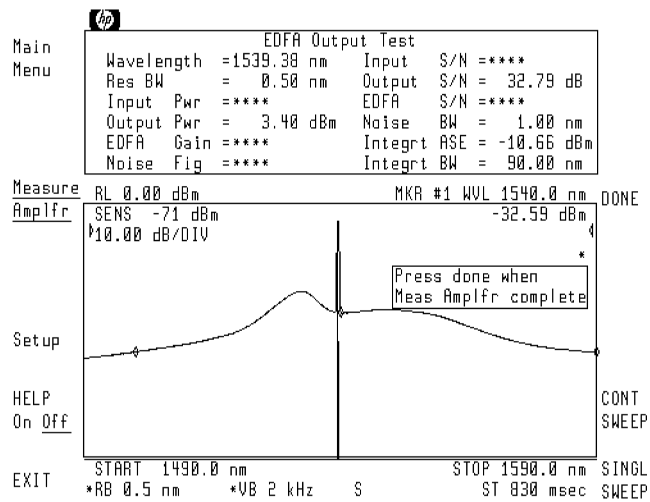
   On HP 71450B/2B optical spectrum analyzers, the input connector is labeled `OPTICAL INPUT`. On HP 71451B optical spectrum analyzers, the input connector is labeled `MONOCHROMATOR INPUT`.



2. Press (USER) and then `EDFA` to start the EDFA test personality.
3. Press `Output Test`.

4. Press `Measure Amplfr` . After the first complete sweep occurs, the display lists all the measurement results at the top of the screen.

   The EDFA test personality measures the signal and noise levels at the end of the sweep. Pressing `SINGLE SWEEP` stops the sweep; each subsequent press of `SINGLE SWEEP` starts a new sweep. Return to continuous sweeps by pressing `CONT SWEEP` so that `CONT` is selected.



Measurements of the output signal.

5. When you are finished with measuring the system, press `DONE` .

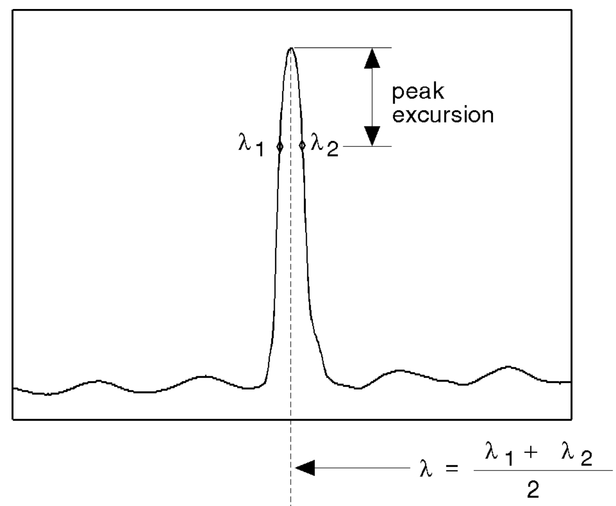
Configuring the Program

This section explains some of the features available with the Setup menu. The Setup menu allows you to configure the following parameters:

- Define the calculation for signal wavelength.
- Define the calculation for noise value at the signal wavelength.
- Adjust the noise integration window.
- Change the noise bandwidth.
- Enter resolution bandwidth correction factors.

The wavelength value is calculated

The signal's wavelength value is calculated as the center of the signal. You can designate which amplitude points are included in determining the wavelength by changing the peak-excursion value. As shown in the following figure, signal wavelength is the average of the wavelength values at the peak excursion amplitude offset.



wavelen

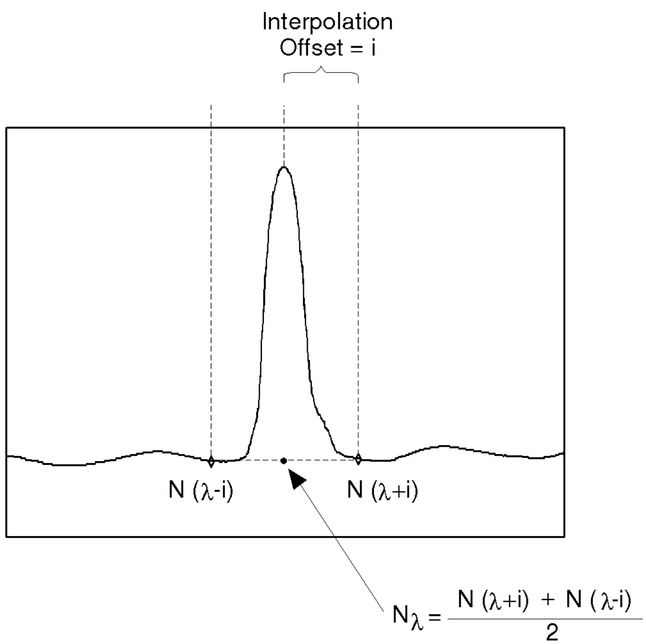
where:

λ_1 is the wavelength to the left side of the signal peak that is equal in amplitude to the signal peak minus the peak excursion value.

λ_2 is the wavelength to the right side of the signal peak that is equal in amplitude to the signal peak minus the peak excursion value.

The noise value at λ is interpolated

The noise value at the signal wavelength is interpolated from noise values that are measured on either side of the peak. A straight-line (average) interpolation is used. The default interpolation offset is 1 nm. Use the INTERP OFFSET softkey to change the interpolation offset. For an accurate noise interpolation, make sure that the interpolation points do not occur at a peak in the displayed response.

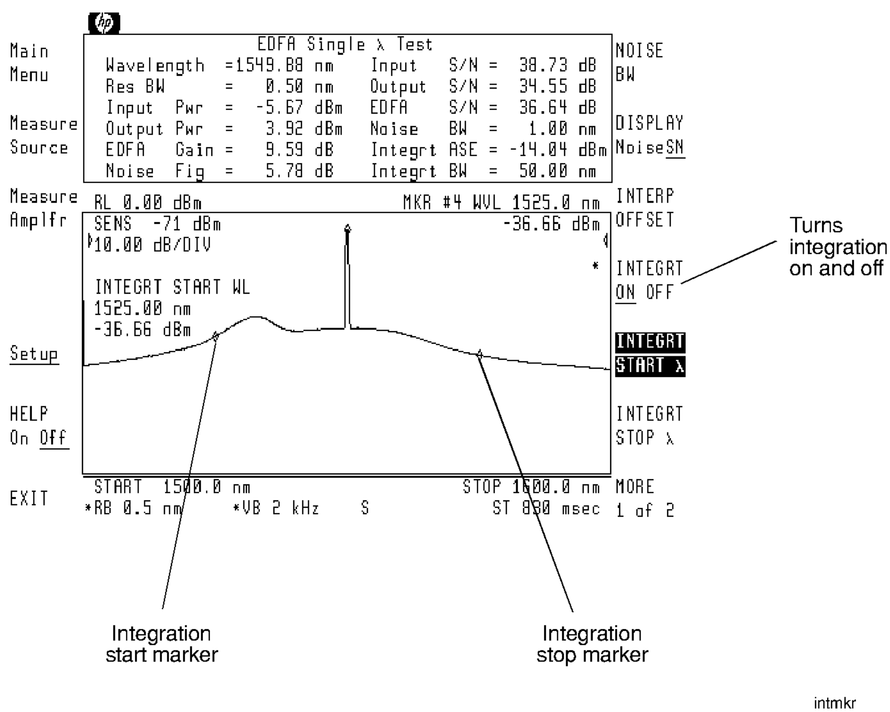


interp

Integration range is adjustable

The Integrgt ASE measurement is the result of integrating the noise between the start and stop integration wavelengths. These two wavelengths are identified by trace markers as shown in the following figure. The default integration points are set to the optical spectrum analyzer's start and stop sweep settings. You can change the integration window using the INTEGRT START λ and INTEGRT STOP λ softkeys. This is shown in the following figure.

The Setup menu's INTEGRT On Off softkey allows you to turn integration on or off. If integration is turned off, asterisks are displayed in place of measurement values. If integration is turned off during a Swept λ Test, the program sets the Integrgt ASE value to −100 dBm at each wavelength. If you display the ASE trace, it will also be at −100 dBm.



You can change the noise bandwidth

The noise bandwidth (`Noise BW`) affects the following measurement results:

`Input S/N` ratio
`Output S/N` ratio
`EDFA S/N` ratio
`Input Noise`
`Output Noise`
`EDFA Noise`

Refer to the equations in Chapter 4 to learn how the `Noise BW` value relates to each of these measurements. Use the `NOISE BW` softkey to change the normalized bandwidth value. The default value is 1 nm.

Enter a factor to correct the resolution bandwidth

The `Res BW` value shown in the measurement table gives the optical spectrum analyzer's true resolution bandwidth. It is a function of wavelength and may differ slightly from the normal screen annotation value. The final step in calculating this value is applying a resolution-bandwidth correction factor:

$$Res\ BW = (Res\ BW)(correction\ factor)$$

For example, for a nominal resolution bandwidth of 0.50 nm, that is measured to be 0.48 nm, the appropriate correction factor to enter is 0.96.

$$0.96 = \frac{0.48}{0.50}$$

The correction factor corrects for variations in the optical spectrum analyzer's slitwheel. (The slitwheel determines bandwidth, and is mounted in the monochromator.) Use the `RES BW CORRECT` softkey to enter resolution-bandwidth correction factors. The default value is 1 which corresponds to 100% (no correction).

To display noise values

1. From the Main Menu, press `Output Test`, `Single λ Test`, or `Swept λ Test`.
2. Press `Setup`.
3. Press `DISPLAY NoiseSN` so that `NOISE` is underlined.

   This replaces the display of `Input S/N`, `Output S/N`, and `EDFA S/N` with the values of `Input Noise`, `Output Noise`, and `EDFA Noise`.

To change the amplitude scale

1. From the Main Menu, press `Output Test`, `Single λ Test`, or `Swept λ Test`.
2. Press `Setup` and then `MORE 1 of 2`.
3. Press `LOG dB/DIV`.
4. Use the knob, step keys, or numeric keypad to enter the desired amplitude scale.

To redefine the calculations for wavelength

1. From the Main Menu, press `Output Test`, `Single λ Test`, or `Swept λ Test`.
2. Press `Setup` and then `MORE 1 of 2`.
3. Press `PEAK EXCURSN`.
4. Use the knob, step keys, or numeric keypad to enter the peak excursion value that is used to calculate the signal wavelength.

To redefine the calculations for noise at peak wavelength

1. From the Main Menu, press `Output Test`, `Single λ Test`, or `Swept λ Test`.
2. Press `Setup` and then `INTERP OFFSET`.
3. Use the knob, step keys, or numeric keypad to enter the interpolation offset value. Make sure that the interpolation offset values do not occur at a peak in the displayed response.

To change the noise bandwidth

1. From the Main Menu, press `Output Test`, `Single λ Test`, or `Swept λ Test`.
2. Press `Setup`.
3. Press `NOISE BW`, and use the knob, step keys, or numeric keypad to enter the noise bandwidth.

   The instrument will now normalize all noise and signal-to-noise measurements to this bandwidth.

To enter a resolution bandwidth correction factor

1. From the Main Menu, press `Output Test`, `Single λ Test`, or `Swept λ Test`.
2. Press `Setup` and then `MORE 1 of 2`.
3. Press `RES BW CORRECT`, and use the knob, step keys, or numeric keypad to enter the resolution bandwidth correction factor.

   Enter this factor as a decimal number. The default value is 1.

To change the ASE integration window

1. From the Main Menu, press `Output Test`, `Single λ Test`, or `Swept λ Test`.
2. Press `Setup`.
3. Press `INTEGRT START λ`, and use the knob, step keys, or numeric keypad to enter the integration's start wavelength.
4. Press `INTEGRT STOP λ`, and use the knob, step keys, or numeric keypad to enter the integration's end wavelength.
5. Press `INTEGRT On Off` so that `On` is underlined.

To select the trigger source

1. From the Main Menu, press `Output Test`, `Single λ Test`, or `Swept λ Test`.
2. Press `Setup` and then `MORE 1 of 2`.
3. If you're using an external trigger signal, press `TRIGGER EXT INT` so that `EXT` is underlined.
4. If you're using internal triggering, press `TRIGGER EXT INT` so that `INT` is underlined.

3

Programming

Programming

This chapter documents the programming commands for the EDFA test personality. EDFA commands can be called from programs in the same manner as any optical spectrum analyzer command. For information on building and running programs, refer to the *HP 71450B/1B/2B Optical Spectrum Analyzers Programmer's Guide*.

Chapter Contents

Introduction

Finding a command is easy

There are two ways of locating an EDFA remote programming command. First, you can use the HELP On Off softkey to display a Help Menu that lists an equivalent programming command. The second method of locating a command is to refer to the command listings in this chapter.

Some optical spectrum analyzer commands are used

The following table lists EDFA functions that should be controlled using optical spectrum analyzer commands. These functions have no equivalent EDFA programming command. Refer to the *HP 71450B/1B/2B Optical Spectrum Analyzers Programmer's Guide* for information on these commands.

Optical Spectrum Analyzer Commands

EDFA Softkey	Definition	OSA Command
CONT SWEEP	Selects continuous sweep mode.	CONTS
SINGLE SWEEP	Selects single sweep mode.	SNGLS
LOG dB/DIV	Changes logarithmic amplitude scale.	LG
PEAK EXCURSN	Sets marker peak-excursion value.	MKPX

Send commands as ASCII strings

EDFA commands are sent to the optical spectrum analyzer as ASCII strings. The method used depends on the programming language and environment. Using an HP Vectra computer with the HP-IB Interface and Command Library (and programming in *C*), send a command as follows:

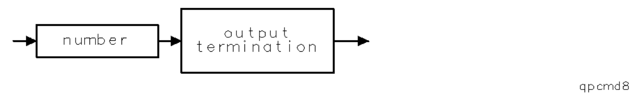
```
iooutputs(723L, "EDFA_ AA", 8);
```

Using an HP 9000 Series 300 technical computer with the HP-BASIC language, the same command would be sent as follows:

```
OUTPUT 723;"EDFA_ AA;"
```

Query responses

Some EDFA commands can be issued as a query. A query causes data to be returned to the computer from the optical spectrum analyzer. The data is returned as an ASCII string. For example, the EDFA_ YA? query might return the string `1.550000E-006` representing 1550 nanometers. The following syntax diagram shows the form of a query response.

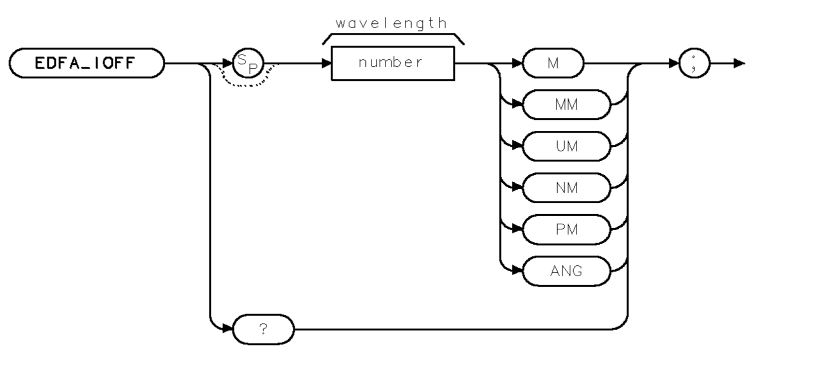


Querying traces in Swept λ Tests results in a string of ASCII numbers separated by commas. You must allocate enough memory to hold all the data. For the default trace length of 800 measurement points, you should allocate at least 6500 bytes.

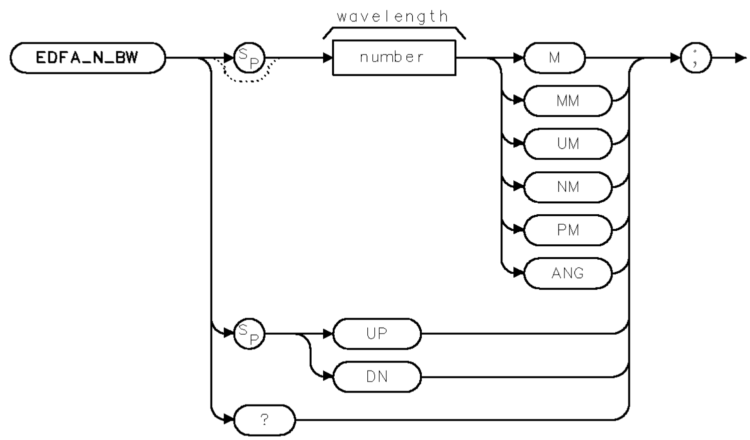
Some commands are more complex

Most of the remote commands are simple and can be output as listed in the next section. However some of the commands are more complex and are described here using syntax diagrams. For a description on how to read syntax diagrams, refer to the end of this section.

The EDFA_IOFF command changes the interpolation offset. This is identical to using the INTERP OFFSET softkey.

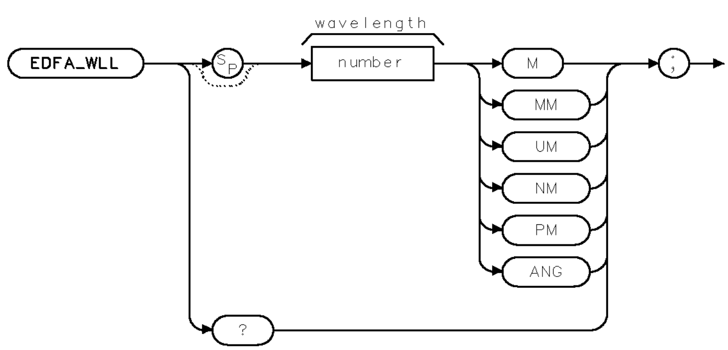


The EDFA_N_BW command changes the noise bandwidth. This is identical to using the NOISE BW softkey.

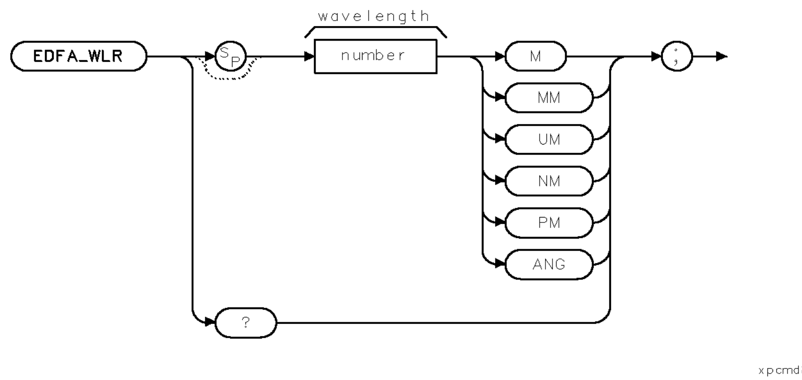


xpcmd6

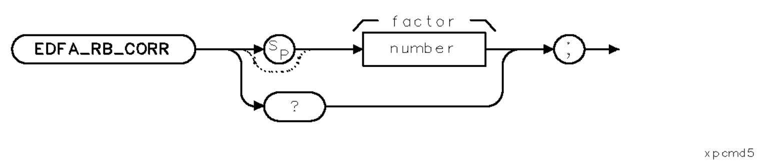
The EDFA_WLL command sets the left integration wavelength. The EDFA_WLR command sets the right integration wavelength. These commands are identical to using INTEGRT START λ and INTEGRT STOP λ softkeys.



xpcmd7

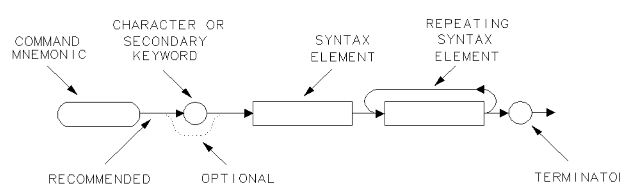


The EDFA_RB_CORR command enters a resolution bandwidth correction factor. This is identical to using `Res BW CORRECT` softkey.



How to read syntax diagrams

Syntax diagrams represent commands pictorially as shown in the following figure.



- Characters enclosed by an oval are commands and their arguments and must be entered *exactly as shown*.

- Characters enclosed by a circle are separators or terminators and must be entered as shown.
- Syntax-diagram elements are connected by solid and dotted lines. Any combination of elements generated by following the lines in the proper direction is correct syntactically. *Solid-line paths are recommended*.

In the syntax diagrams, numbers and units are expressed in the forms shown in the following table.

Syntax Diagram Elements

Syntax Component	Definition/Range
number	Expressed as integer, decimal, or in exponential (E) form. Real Number Range: $\pm$ 1.797,693,134,862,315 $\times$ 10^{308}, including 0. Up to 15 significant figures allowed. Numbers may be as small as $\pm$ 2.225,073,858,507,202 $\times$ 10^{-308} Integer Number Range: −32,768 through +32,767
output termination	Line feed (L_F) with end-or-identify (EOI) condition. ASCII code 10 (line feed) is sent via HP-IB, with the end-or-identify control line on HP-IB set to indicate the end of the transmission.
units	Represent standard scientific units. Amplitude Units: DB, DBM, MW, NW, PW, UW, W Current Units: A, MA, UA Frequency Units: HZ, KHZ, MHZ, GHZ, KZ, MZ, GZ Time Units: S, MS, US, SC Wavelength Units: ANG, KM, NM, UM, PM

Commands by Functional Group

Commands by Functional Group

Softkey/Function	Programming Command	Definition
Starting the Personality		
EDFA	EDFA_	Starts EDFA test personality.
Amplitude Scale:		
AUTO SCALE	EDFA_ B	Automatically scales the displayed data.
LOG dB/DIV	LG	Refer to OSA programmer's guide.
Prompt Control:		
PROMPT Off	EDFA_ JB	Turns prompts off.
PROMPT On	EDFA_ JA	Turns prompts on.
Measurement Parameters:		
HP8168 DWELL	EDFA_LDWELL?	Queries laser dwell for swept λ testing.
Integrt BW	EDFA_ YL?	Queries Integrt BW value.
INTEGRT On	EDFA_ NA	Turns integration on.
INTEGRT Off	EDFA_ NB	Turns integration off.
INTEGRT START λ	EDFA_WLL[1]	Sets and queries the left integration wavelength value.
INTEGRT STOP λ	EDFA_WLR[1]	Sets and queries the right integration wavelength value.
Interpolated noise	EDFA_IAMP?	Queries interpolated noise value (default = −100 dBm.)
INTERP OFFSET	EDFA_IOFF[1]	Sets and queries the interpolation offset.
Noise BW	EDFA_ YJ?	Queries Noise BW value.
NOISE BW	EDFA_N_BW[1]	Enters and queries Noise bandwidth value.
Res BW	EDFA_ YB?	Queries Res BW value.
PEAK EXCURSN	MKPX	Refer to OSA programmer's guide.
Res BW CORRECT	EDFA_RB_CORR[1]	Enters and queries resolution bandwidth correction values.
Menu Control:		
data select	EDFA_ AJ	Displays "data select" menu.
Display Data	EDFA_ AI	Displays "Display Data" menu.
DISPLAY NoiseSN	EDFA_ VA	Displays "Noise" values.
DISPLAY NoiseSN	EDFA_ VX	Displays "S/N" values.
DONE	EDFA_ D	Stops the acquisition of measurement data.

1 Refer to the introduction in this chapter for proper command syntax.

Commands by Functional Group (continued)

Softkey/Function	Programming Command	Definition
Menu Control *continued*:		
EXIT	EDFA_ Q	Quits the EDFA test personality.
Main Menu	EDFA_ AA	Displays the "Main" menu.
Measure Amplfr	EDFA_ AF	Starts amplifier measurements.
Measure Source	EDFA_ AC	Starts source measurements.
MORE 1 of 2	EDFA_ F	Selects the next page of softkeys.
MORE 2 of 2	EDFA_ F	Selects the next page of softkeys.
Output Test	EDFA_ GA	Selects Output Test.
Setup	EDFA_ AB	Displays the "Setup" menu.
Single λ Test	EDFA_ GB	Selects Single λ Test.
Swept λ Test	EDFA_ GC	Selects Swept λ Test.
Query Data at Marker:		
EDFA Gain	EDFA_ YF?	Queries EDFA Gain value.
EDFA S/N	EDFA_ YI?	Queries EDFA S/N value.
EDFA Noise	EDFA_ YO?	Queries EDFA Noise value.
Input Pwr	EDFA_ YC?	Queries Input Pwr value.
Input S/N	EDFA_ YG?	Queries Input S/N value.
Input Noise	EDFA_ YM?	Queries Input Noise value.
Integrt ASE	EDFA_ YK?	Queries Integrt ASE value.
Noise Fig	EDFA_ YE?	Queries Noise Figure value.
Output Pwr	EDFA_ YD?	Queries Output Pwr value.
Output S/N	EDFA_ YH?	Queries Output S/N value.
Output Noise	EDFA_ YN?	Queries Output Noise value.
Wavelength	EDFA_ YA?	Queries Wavelength value.
Query Traces (Swept λ Test):		
EDFA Gain	EDFA_GAIN?	Queries trace of EDFA Gain.
EDFA Noise	EDFA_ CR;TRA?	Queries trace of EDFA Noise.
EDFA S/N	EDFA_SN?	Queries trace of EDFA S/N.
Input Noise	EDFA_ CO;TRA?	Queries trace of Input Noise.
Input Noise	IN_NOISE?	Queries trace of noise measured in true noise bandwidth (OSA bandwidth).
Input Pwr	IN_PWR?	Queries trace of Input Pwr.

Commands by Functional Group (continued)

Softkey/Function	Programming Command	Definition
Query Traces (Swept λ Test) *continued*:		
Input S/N	IN_SN?	Queries trace of Input S/N.
Integrt ASE	INTEGRT_ASE?	Queries trace of Integrt ASE.
Noise Figure	EDFA_NF?	Queries trace of Noise Figure.
Output Noise	EDFA_ CQ;TRA?	Queries trace of Output Noise.
Output Noise	OUT_NOISE?	Queries trace of noise measured in true noise bandwidth (OSA bandwidth).
Output Pwr	OUT_PWR?	Queries trace of Output Pwr.
Output S/N	OUT_SN?	Queries trace of Output S/N.
Trace Displaying (Swept λ Test):		
EDFA NOISE	EDFA_ CR	Displays trace of EDFA noise.
EDFA S/N	EDFA_ CJ	Displays trace of EDFA S/N.
EVERY NOISE	EDFA_ CN	Displays all noise traces.
EVERY S/N	EDFA_ CG	Displays all S/N traces.
GAIN	EDFA_ CB	Displays trace of gain.
GAIN and NF	EDFA_ CA	Displays traces of gain and noise figure.
INPUT NOISE	EDFA_ CO	Displays trace of input noise.
INPUT POWER	EDFA_ CD	Displays trace of input power.
INPUT S/N	EDFA_ CH	Displays trace of input S/N.
INTEGRT ASE[1]	EDFA_ CM	Displays trace of integrated ASE.
NF	EDFA_ CC	Displays trace of noise figure.
OUTPUT POWER	EDFA_ CE	Displays trace of output power.
OUTPUT NOISE	EDFA_ CQ	Displays trace of output noise.
OUTPUT S/N	EDFA_ CI	Displays trace of output S/N.
Sweep Control:		
CONT SWEEP	CONTS	Refer to OSA programmer's guide.
SINGLE SWEEP	SNGLS	Refer to OSA programmer's guide.
Trigger Control:		
TRIGGER EXT	EDFA_ LE	Selects external triggering.
TRIGGER INT	EDFA_ LI	Selects internal triggering.

1 If the INTEGRT On Off softkey is set to Off, all ASE measurement values are automatically set to −100 dBm.

Softkeys versus Commands

Softkeys versus Commands

Softkey	Equivalent Programming Command	Definition
AUTO SCALE	EDFA_ B	Automatically scales the displayed data.
CONT SWEEP	CONTS	Refer to OSA programmer's guide.
data select	EDFA_ AJ	Displays "data select" menu.
Display Data	EDFA_ AI	Displays "Display Data" menu.
DISPLAY NoiseSN	EDFA_ VA, EDFA_ VX	Displays noise (VA) or S/N (VX) values.
DONE	EDFA_ D	Stops the acquisition of measurement data.
EDFA	EDFA_	Starts EDFA test personality.
EDFA NOISE	EDFA_ CR	Displays trace of EDFA noise in swept λ testing.
EDFA S/N	EDFA_ CJ	Displays trace of EDFA S/N ratio in swept λ testing.
EVERY S/N	EDFA_ CG	Displays all S/N traces in swept λ testing.
EVERY NOISE	EDFA_ CN	Displays all noise traces in Swept λ testing.
EXIT	EDFA_ Q	Quits the EDFA test personality.
GAIN	EDFA_ CB	Displays trace of gain in swept λ testing.
GAIN and NF	EDFA_ CA	Displays traces of gain and noise figure in swept λ testing.
HELP On Off	—	—
HP8168 DWELL	EDFA_LDWELL?	Queries laser dwell for swept λ testing.
INPUT POWER	EDFA_ CD	Displays trace of input power in Swept λ testing.
INPUT NOISE	EDFA_ CO	Displays trace of input noise in Swept λ testing.
INPUT S/N	EDFA_ CH	Displays trace of input S/N in swept λ testing.
INTEGRT ASE	EDFA_ CM	Displays trace of integrated ASE in swept λ testing.
INTEGRT On Off	EDFA_ NA, EDFA_ NB	Turns integration on (NA) or off (NB).
INTEGRT START λ	EDFA_WLL[1]	Sets the left integration wavelength.
INTEGRT STOP λ	EDFA_WLR[1]	Sets the right integration wavelength.
INTERP OFFSET	EDFA_IOFF[1]	Enters an interpolation offset.

1 Refer to the introduction in this chapter for proper command syntax.

Softkeys versus Commands (continued)

Softkey	Equivalent Programming Command	Definition
LOG dB/DIV	LG	Refer to OSA programmer's guide.
Main Menu	EDFA_ AA	Displays the "Main" menu.
Measure Amplfr	EDFA_ AF	Starts amplifier measurements.
Measure Source	EDFA_ AC	Starts source measurements.
MORE 1 of 2	EDFA_ F	Selects the next page of softkeys.
MORE 2 of 2	EDFA_ F	Selects the next page of softkeys.
NF	EDFA_ CC	Displays trace of noise figure in swept λ testing.
NOISE BW	EDFA_N_BW[1]	Enters a S/N bandwidth value.
OUTPUT POWER	EDFA_ CE	Displays trace of output power in swept λ testing.
OUTPUT NOISE	EDFA_ CQ	Displays trace of output noise in Swept λ testing.
OUTPUT S/N	EDFA_ CI	Displays trace of output S/N in Swept λ testing.
Output Test	EDFA_ GA	Selects Output Test.
PEAK EXCURSN	MKPX	Refer to OSA programmer's guide.
PROMPT On Off	EDFA_ JA, EDFA_ JB	Turns prompts on (JA) or off (JB).
Res BW CORRECT	EDFA_RB_CORR[1]	Enters a resolution bandwidth correction value.
Setup	EDFA_ AB	Displays the "Setup" menu.
Single λ Test	EDFA_ GB	Selects Single λ Test.
SINGLE SWEEP	SNGLS	Refer to OSA programmer's guide.
Swept λ Test	EDFA_ GC	Selects Swept λ Test.
TRIGGER EXT INT	EDFA_ LI, EDFA_ LE	Selects internal (LI) or external (LE) triggering.

1 Refer to the introduction in this chapter for proper command syntax.

Commands in Alphabetical Order

Programming Command	Definition	Softkey
CONTS	Refer to OSA programmer's guide.	CONT SWEEP
EDFA_	Starts EDFA test personality.	EDFA
EDFA_ AA	Displays the "Main" menu.	Main Menu
EDFA_ AB	Displays the "Setup" menu.	Setup
EDFA_ AC	Starts source measurements.	Measure Source
EDFA_ AF	Starts amplifier measurements.	Measure Amplfr
EDFA_ AI	Displays "Display Data" menu.	Display Data
EDFA_ AJ	Displays "data select" menu.	data select
EDFA_ B	Automatically scales the displayed data.	AUTO SCALE
EDFA_ CA	Displays traces of gain and noise figure in swept λ testing.	GAIN and NF
EDFA_ CB	Displays trace of gain in swept λ testing.	GAIN
EDFA_ CC	Displays trace of noise figure in swept λ testing.	NF
EDFA_ CD	Displays trace of input power in swept λ testing.	INPUT POWER
EDFA_ CE	Displays trace of output power in swept λ testing.	OUTPUT POWER
EDFA_ CG	Displays all S/N traces in swept λ testing.	EVERY S/N
EDFA_ CH	Displays trace of input S/N in swept λ testing.	INPUT S/N
EDFA_ CI	Displays trace of output S/N in swept λ testing.	OUTPUT S/N
EDFA_ CJ	Displays EDFA S/N bandwidth.	EDFA S/N
EDFA_ CM	Displays trace of integrated ASE in swept λ testing.	INTEGRT ASE
EDFA_ CN	Displays traces of **Every Noise**.	
EDFA_ CO	Displays trace of **Input Noise**.	
EDFA_ CO;TRA?	Queries trace of **Input Noise**.	
EDFA_ CQ	Displays trace of **Output Noise**.	
EDFA_ CQ;TRA?	Queries trace of **Output Noise**.	
EDFA_ CR	Displays trace of **EDFA Noise**.	
EDFA_ CR;TRA?	Queries trace of **EDFA Noise**.	
EDFA_ D	Stops the acquisition of measurement data.	DONE

Commands in Alphabetical Order (continued)

Programming Command	Definition	Softkey
EDFA_ F	Selects the next page of softkeys.	MORE 1 of 2
EDFA_ GA	Selects Output Test.	Output Test
EDFA_ GB	Selects Single λ Test.	Single λ Test
EDFA_ GC	Selects Swept λ Test.	Swept λ Test
EDFA_ JA	Turns prompts on.	PROMPT On
EDFA_ JB	Turns prompts off.	PROMPT Off
EDFA_ LE	Selects external triggering.	TRIGGER EXT INT
EDFA_ LI	Selects internal triggering.	TRIGGER EXT INT
EDFA_ NA	Turns integration on.	INTEGRT On Off
EDFA_ NB	Turns integration on.	INTEGRT On Off
EDFA_ Q	Quits the EDFA test personality.	EXIT
EDFA_ VA	Toggles "DISPLAY NoiseSN" softkey to noise.	DISPLAY NoiseSN
EDFA_ VX	Toggles "DISPLAY NoiseSN" softkey to S/N.	DISPLAY NoiseSN
EDFA_ YA?	Queries **Wavelength** value at the marker.	
EDFA_ YB?	Queries **Res BW** value at the marker.	
EDFA_ YC?	Queries **Input Pwr** value at the marker.	
EDFA_ YD?	Queries **Output Pwr** value at the marker.	
EDFA_ YE?	Queries **Noise Figure** value at the marker.	
EDFA_ YF?	Queries **EDFA Gain** value at the marker.	
EDFA_ YG?	Queries **Input S/N** value at the marker.	
EDFA_ YH?	Queries **Output S/N** value at the marker.	
EDFA_ YI?	Queries **EDFA S/N** value at the marker.	
EDFA_ YJ?	Queries **Noise BW** value at the marker.	
EDFA_ YK?	Queries **Integrt ASE** value at the marker.	
EDFA_ YL?	Queries **Integrt BW** value at the marker.	
EDFA_ YM?	Queries **Input Noise** value at the marker.	
EDFA_ YN?	Queries **Output Noise** value at the marker.	
EDFA_ YO?	Queries **EDFA Noise** value at the marker.	
EDFA_GAIN?	Queries trace of **EDFA Gain** in Swept λ Testing.	
EDFA_IAMP?	Queries interpolated noise value.	

Commands in Alphabetical Order (continued)

Programming Command	Definition	Softkey
EDFA_IOFF[1]	Enters an interpolation offset.	INTERP OFFSET
EDFA_IOFF?	Queries the interpolation offset.	
EDFA_LDWELL?	Queries laser dwell for Swept λ testing.	HP8168 DWELL
EDFA_N_BW[1]	Enters a noise bandwidth value.	NOISE BW
EDFA_N_BW?	Queries S/N bandwidth value.	
EDFA_NF?	Queries trace of `Noise Figure` in Swept λ Testing.	
EDFA_RB_CORR[1]	Enters a resolution bandwidth correction value.	Res BW CORRECT
EDFA_RB_CORR?	Queries resolution bandwidth correction value.	
EDFA_SN?	Queries trace of `EDFA S/N` in Swept λ Testing.	
EDFA_WLL[1]	Sets the left integration wavelength.	INTEGRT START λ
EDFA_WLL?	Queries the left integration wavelength value.	
EDFA_WLR[1]	Sets the right integration wavelength.	INTEGRT STOP λ
EDFA_WLR?	Queries the right integration wavelength value.	
IN_NOISE?	Queries trace of noise measured in true noise bandwidth (OSA bandwidth).	
IN_PWR?	Queries trace of `Input Pwr` in Swept λ Testing.	
IN_SN?	Queries trace of `Input S/N` in Swept λ Testing.	
INTEGRT_ASE?	Queries trace of `Integrt ASE` in Swept λ Testing.	
LG	Refer to OSA programmer's guide.	LOG dB/DIV
MKPX	Refer to OSA programmer's guide.	PEAK EXCURSN
OUT_NOISE?	Queries trace of noise measured in true noise bandwidth (OSA bandwidth).	
OUT_PWR?	Queries trace of `Output Pwr` in Swept λ Testing.	
OUT_SN?	Queries trace of `Output S/N` in Swept λ Testing.	
SNGLS	Refer to OSA programmer's guide.	SINGLE SWEEP

1 Refer to the introduction in this chapter for proper command syntax.

Measurement Examples

The example programs documented in this section control the EDFA test personality from an HP Vectra or other IBM compatible personal computer. The programs, `SINGLEWL.C` and `SWEPTWL.C`, are written in *C* and demonstrate the basics of controlling the EDFA test personality. They are not designed to show optimum measurement techniques.

Required equipment

In order to run these programs, you'll need the following equipment:

- A C compiler that is compatible with the HP 82335B.
- HP 82335B Interface and Command Library (includes HP-IB card).
- HP 8168B/C tunable laser source. *(for SWEPTWL.C)*

Compiling and linking

To create an executable file, you must do the following:

1. Install the HP-IB 82335B Interface and Command Library.

   Place the `clhpib.lib` file in your compiler's library directory. Two header files `chpib.h` and `cfunc.h` should be in the same directory as the compiler's header files.

2. Create a source file that is identical to the example listing found on the following pages.
3. Create a makefile that lists your source file and `clhpib.lib`.

   This links the HP 82335B Interface and Command Library.

4. Connect all the equipment with HP-IB cables, turn on the line power, and install the EDFA test personality.
5. Compile and run the program.

Manifest constants

The following manifest constants, defined at the top of each program, determine the interface select code and instrument addresses. The select code is determined by switches on the HP-IB card. If these values have been changed, you will need to change the following constants. Do not modify any other constants.

ISC (interface select code) 7
OSA (optical spectrum analyzer's address) 23
TLS (tuned laser source's address) 24

Automating Single λ Testing

SINGLEWL.C performs multiple Single λ Tests. Measurement results are returned to the computer and saved in a file. The following figure shows a file created during the testing of two amplifiers.

```
EDFA SINGLE WAVELENGTH TEST RESULTS
--------------------------------------------

EDFA serial number: 324
        Wavelength:         1550.00 nm
        Res BW:               0.50 nm
        Input Power:         -9.78 dBm
        Output Power:         3.74 dBm
        EDFA Gain:           13.52 dB
        Noise Figure:         3.73 dB
        Input S/N:          -100.00 dB
        Output S/N:          32.97 dB
        EDFA S/N:            34.54 dB
        Noise BW:             1.00 nm
        Integrated ASE:     -16.76 dBm
        Integrated BW:       20.00 nm

EDFA serial number: 325
        Wavelength:         1550.00 nm
        Res BW:               0.50 nm
        Input Power:         -9.83 dBm
        Output Power:         3.22 dBm
        EDFA Gain:           13.05 dB
        Noise Figure:         3.65 dB
        Input S/N:          -100.00 dB
        Output S/N:          33.13 dB
        EDFA S/N:            35.47 dB
        Noise BW:             1.00 nm
        Integrated ASE:     -15.18 dBm
        Integrated BW:       20.00 nm
```

Example file created by SINGLEWL.C

You supply the name of the file via a command-line argument to the program. For example, if you wanted to save the results in a file called `results.doc`, start the program with the following command:

```
singlewl c:\results.doc
```

The program uses the following test algorithm:

1. Open file specified in command-line argument.
2. Prompt user for OSA wavelength range.
3. Initialize bus and optical spectrum analyzer.
4. Start EDFA test personality.
5. Enter a `while` loop
   a. Prompt user for EDFA's serial number.
   b. Test for loop exit condition.
   c. Perform EDFA source measurement.
   d. Perform EDFA amplifier measurement.
   e. Display measurement results on computer.
   f. Save measurement results in file.
6. Close file.
7. Exit program.

The data file is an ASCII file that can be read by any word processor. The global `commands` string is used to create output strings and store input strings. The test's wavelength range is entered via the function `Enter_OSA_Range()`.

The `Measure()` function controls the source and amplifier tests. The optical spectrum analyzer is left in continuous sweep mode until the actual measurement is taken. Then, single sweep is selected. A `DONE?` query pauses the program until the optical spectrum analyzer completes the measurement.

The function `Save_Results_In_File()` queries the EDFA test personality for the measurement data, displays the data on the computer's screen, and saves the data into the file. The returned data is converted to a `double` value so it can be easily formatted for display.

Listing: SINGLEWL.C

```
/*******************************************************
                          SINGLEWL.C
7-9-93 KGB
*******************************************************/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <process.h>
#include <ctype.h>
#include <conio.h> /* used for kbhit() */


/* ADDITIONAL HEADER FILES
The following header files come with the HP 82335A/B/I HP-IB
Interface and Command Library
*/
#include <chpib.h> /* HPIB cmd library constant declarations */
#include <cfunc.h> /* HPIB cmd library function prototypes */


/* MANIFEST CONSTANTS
*/
#define ISC 7L
#define OSA (ISC * 100L + 23L)
#define SOURCE 1
#define AMPLIFIER 2
#define START 3
#define STOP 4
#define ARRAY_SIZE 200

/* FUNCTION PROTOTYPES
*/
void Initialize(void);
int Enter_OSA_Range(int);
int Enter_Serial_Number(void);
void Measure(int);
void Save_Results_In_File(void);
void Send_Command(long, char *);
char *Get_Query(long, char *);
void Error_Handle(int, char *);

/* GLOBAL VARIABLES
*/
double Start_WL, Stop_WL;
char commands[ARRAY_SIZE + 1];
```

```
FILE *fpw;


/***************** main() *******************
This is the main function of the program.
The program starts here.
*******************************************/
void
main(int argc, char **argv)
{
        if (argc < 2) {
                system("cls"); /* clear the computer's screen */
                printf("\n\n%s\n\n%s\n\n",
                        "Incorrect usage! You must enter name of file to save data.",
                        "For example: singlewl c:\\data.txt");
                exit(1);
        }
        if ((fpw = fopen(argv[1], "wt")) == NULL) {
                fclose(fpw);
                exit(0);
        }
        if (!Enter_OSA_Range(START)) {
                fclose(fpw);
                exit(0);
        }
        if (!Enter_OSA_Range(STOP)) {
                fclose(fpw);
                exit(0);
        }
        Initialize();
        while (1) {
                if (!Enter_Serial_Number())
                        break;
                Measure(SOURCE);
                Measure(AMPLIFIER);
                Save_Results_In_File();
        }
        fclose(fpw);
        exit(0);
}

/***************** Enter_OSA_Range() *********
This function enters the OSA wavelength range.
*******************************************/
int Enter_OSA_Range(int end)
{
        system("cls"); /* clear the computer's screen */
        printf("\nEDFA Single Wavelength Test\n\n");
```

```
        printf("\n\nPress ‘q’ at any time to exit program.");
        switch (end) {
        case START:
                Error_Handle(IOLLOCKOUT(ISC), "IOLLOCKOUT"); /* locks out front panels */
                printf("\n\nEnter an OSA [START] wavelength in nanometers: ");
                while (1) {
                        gets(commands);
                        strlwr(commands);
                        if (strcmp(commands, "q") == 0)  /* ‘q’ entered, exit program */
                                return 0;
                        Start_WL = atof(commands);
                        if (Start_WL >= 600.00 && Start_WL <= 1700.00)
                                break;
                        printf("\nBad value, try again. Or, enter ‘q’ to exit: ");
                }
                break;
        case STOP:
                printf("\n\nEnter an OSA [STOP] wavelength in nanometers: ");
                while (1) {
                        gets(commands);
                        strlwr(commands);
                        if (strcmp(commands, "q") == 0)  /* ‘q’ entered, exit program */
                                return 0;
                        Stop_WL = atof(commands);
                        if (Stop_WL >= 600.00 && Stop_WL <= 1700.00)
                                break;
                        printf("\nBad value, try again. Or, enter ‘q’ to exit: ");
                }
                break;
        }
        return 1;
}

/****************** Initialize() *************
This function sets the interface and starts
the EDFA personality.
**********************************************/
void Initialize()
{
        system("cls"); /* clear the computer’s screen */
        printf("\n\t\t\tEDFA Single Wavelength Test\n\n");

        Error_Handle(IORESET(ISC), "IORESET");
        Error_Handle(IOTIMEOUT(ISC, 10.0), "IOTIMEOUT #1");
        Error_Handle(IOCLEAR(ISC), "IOCLEAR");
        Error_Handle(IOLLOCKOUT(ISC), "IOLLOCKOUT");

        /* INITIALIZE OPTICAL SPECTRUM ANALYZER
        */
```

```
        printf("\nInitializing HP 71450B/1B/2B.");
        Send_Command(OSA, "IP;"); /* OSA instrument preset */

        /* START EDFA PERSONALITY
        */
        printf("\nStarting the EDFA Personality.");
        Send_Command(OSA, "EDFA_;");  /* load EDFA personality */
        printf("\nWaiting for EDFA to load . . .");
        Get_Query(OSA, "DONE?;");  /* pause until EDFA loaded */
        printf("\nTurning EDFA prompts off.");
        Send_Command(OSA, "EDFA_ JB;");  /* turn EDFA prompt messages off */
        printf("\nSelecting Single Wavelength Testing.");
        Send_Command(OSA, "EDFA_ GB;");  /* select single wavelength test */
        printf("\nWaiting for EDFA to finish tasks . . .");
        Get_Query(OSA, "DONE?;");   /* pause until test is selected */

        /* SET OPTICAL SPECTRUM ANALYZER RANGE
        */
        printf("\nSetting HP 71450B/1B/2B measurement range.");
        sprintf(commands, "STARTWL %.3fNM;STOPWL %.3fNM;SENS -65DBM;",
                Start_WL, Stop_WL);
        Send_Command(OSA, commands);

        /* print file header */
        fprintf(fpw, "%s\n%s\n\n",
                "EDFA SINGLE WAVELENGTH TEST RESULTS",
                "-------------------------------------------");
        return;
}

/****** Enter_Serial_Number() *********
This function prints the EDFA's id number
in the file. Also this function controls
exit from measurement loop.
*********************************************/
int Enter_Serial_Number()
{
        system("cls"); /* clear the computer's screen */
        printf("\n\n%s\n%s",
                "Enter the serial number of the EDFA",
                "Or, press 'q' to exit program: ");

        while(kbhit()) /* clear keyboard buffer */
                getch();
        gets(commands);
        strlwr(commands);
        if (strcmp(commands, "q") == 0)  /* 'q' entered, exit program */
                return 0;
        fprintf(fpw, "EDFA serial number: %s\n", commands); /* print serial number to file */
```

```
        return 1;
}

/****************** Measure() *********
This function performs the source and
amplifier measurements.
**********************************************/
void Measure(int device)
{
        system("cls"); /* clear the computer's screen */
        if (device == SOURCE) {
                printf("\n\n%s\n\n%s\n\n%s",
                        "1. Connect the source to the optical spectrum analyzer.",
                        "2. Turn the source on.",
                        "3. Press any key to continue . . .");
                while(kbhit()) /* clear keyboard buffer */
                        getch();
                getch();       /* wait for key press */
                Send_Command(OSA, "EDFA_ AC;"); /* select source measurement */
        }
        else if (device == AMPLIFIER) {
                printf("\n\n%s\n\n%s\n\n%s\n\n%s",
                        "1. Disconnect the source from the optical spectrum analyzer.",
                        "2. Connect the source to the amplifier input.",
                        "3. Connect amplifier output to optical spectrum analyzer.",
                        "4. Press any key to continue . . .");
                while(kbhit()) /* clear keyboard buffer */
                        getch();
                getch();       /* wait for key press */
                Send_Command(OSA, "EDFA_ AF;"); /* select amplifier measurement */
        }
        system("cls"); /* clear the computer's screen */
        printf("\n\n%s",
                "Performing measurement. Please wait . . . ");
        Send_Command(OSA, "SNGLS;");  /* put OSA in single sweep */
        Send_Command(OSA, "TS;");     /* sweep OSA */
        Get_Query(OSA, "DONE?;");     /* wait for OSA sweep to finish */
        Send_Command(OSA, "CONTS;");  /* put OSA in continuous sweep */
        Send_Command(OSA, "EDFA_ D;");  /* stop measurement */
        return;
}

/************ Save_Results_In_File() *********
This function queries the test results and:
   1. Displays the results on the PC.
   2. Saves the results to the file.
**********************************************/
void Save_Results_In_File()
{
```

```
double value;
system("cls"); /* clear the computer's screen */
printf("\n\t\t\t%s\n\n\t%s\n",       /* write table head to display */
        "EDFA Measurement Results",
        "Parameter           Value");

Get_Query(OSA, "EDFA_ YA?;");
value = atof(commands);
value *= 1000000000;
printf("\tWavelength:          %6.2f nm\n", value);
fprintf(fpw, "\tWavelength:         %6.2f nm\n", value);

Get_Query(OSA, "EDFA_ YB?;");
value = atof(commands);
value *= 1000000000;
printf("\tRes BW:              %6.2f nm\n", value);
fprintf(fpw, "\tRes BW:             %6.2f nm\n", value);

Get_Query(OSA, "EDFA_ YC?;");
value = atof(commands);
printf("\tInput Power:         %6.2f dBm\n", value);
fprintf(fpw, "\tInput Power:        %6.2f dBm\n", value);

Get_Query(OSA, "EDFA_ YD?;");
value = atof(commands);
printf("\tOutput Power:        %6.2f dBm\n", value);
fprintf(fpw, "\tOutput Power:       %6.2f dBm\n", value);

Get_Query(OSA, "EDFA_ YF?;");
value = atof(commands);
printf("\tEDFA Gain:           %6.2f dB\n", value);
fprintf(fpw, "\tEDFA Gain:          %6.2f dB\n", value);

Get_Query(OSA, "EDFA_ YE?;");
value = atof(commands);
printf("\tNoise Figure:        %6.2f dB\n", value);
fprintf(fpw, "\tNoise Figure:       %6.2f dB\n", value);

Get_Query(OSA, "EDFA_ YG?;");
value = atof(commands);
printf("\tInput S/N:           %6.2f dB\n", value);
fprintf(fpw, "\tInput S/N:          %6.2f dB\n", value);

Get_Query(OSA, "EDFA_ YH?;");
value = atof(commands);
printf("\tOutput S/N:          %6.2f dB\n", value);
fprintf(fpw, "\tOutput S/N:         %6.2f dB\n", value);

Get_Query(OSA, "EDFA_ YI?;");
```

```
        value = atof(commands);
        printf("\tEDFA S/N:           %6.2f dB\n", value);
        fprintf(fpw, "\tEDFA S/N:           %6.2f dB\n", value);

        Get_Query(OSA, "EDFA_ YJ?;");
        value = atof(commands);
        value *= 1000000000;
        printf("\tNoise BW:           %6.2f nm\n", value);
        fprintf(fpw, "\tNoise BW:           %6.2f nm\n", value);

        Get_Query(OSA, "EDFA_ YK?;");
        value = atof(commands);
        printf("\tIntegrated ASE:     %6.2f dBm\n", value);
        fprintf(fpw, "\tIntegrated ASE:     %6.2f dBm\n", value);

        Get_Query(OSA, "EDFA_ YL?;");
        value = atof(commands);
        value *= 1000000000;
        printf("\tIntegrated BW:      %6.2f nm\n\n", value);
        fprintf(fpw, "\tIntegrated BW:      %6.2f nm\n\n", value);

        Error_Handle(IOLOCAL(ISC), "IOLOCAL");  /* put OSA in "local" mode */

        while(kbhit())
                getch();
        printf("Press any key to continue . . .");
        getch();
        return;
}

/************* Send_Command() ****************
This function sends HP-IB commands to OSA or TLS.
*********************************************/
void Send_Command(long inst, char *string)
{
        Error_Handle(IOOUTPUTS(inst, string, strlen(string)), "IOOUTPUTS");
        return;
}

/************* Get_Query() *******************
This function gets queries from OSA.
The data is placed in the global commands string.
The I/O timeout is increased to allow instrument
to process commands before returning query data.
*********************************************/
char *Get_Query(long inst, char *string)
{
        int length = ARRAY_SIZE;
        Error_Handle(IOTIMEOUT(ISC, 40.0), "IOTIMEOUT #1");
```

```
        Send_Command(inst, string);
        Error_Handle(IOENTERB(inst, commands, &length, 1), "IOENTERB #1");
        commands[length] = '\0'; /* append NULL byte */
        Error_Handle(IOTIMEOUT(ISC, 5.0), "IOTIMEOUT #2");
        return commands;
}


/******** Error_Handle(int, char *) **********
This function prints HP-IB errors on the PC's
display.
**********************************************/
void Error_Handle(int error, char *routine)
{
        if (error != NOERR) {
                printf("HPIB error in call to %s: %d, %s\n",
                        routine, error, errstr(error));
                fclose(fpw);
                exit(1);
        }
        return;
}
```

Automating Swept λ Testing

`SWEPTWL.C` performs an automated Swept λ Test using an HP 8168B/C tuned laser source. The measurement results are displayed on the computer's screen. You can modify this program so that it also saves the data to a file. Simply copy the appropriate code from the example listing of SINGLEWL.C.

The computer controls laser settings such as wavelength, step size, and power. Because the wavelength stepping of the laser and optical spectrum analyzer are remotely controlled, external triggering is not used. The program uses the following test algorithm:

1. Initialize bus and instruments.
2. Start EDFA test personality.
3. Find laser's wavelength range.
4. Enter a `while` loop
   a. Prompt user for measurement range in nanometers.
   b. Prompt user for measurement step size in nanometers.
   c. Find laser's power range for desired wavelength range.
   d. Prompt user for measurement power level.
   e. Perform EDFA source measurement.
   f. Perform EDFA amplifier measurement.
   g. Display gain and noise figure traces on OSA.
   h. Display measurement results on computer.
5. Exit program.

The program queries the HP 8168B/C for the wavelength and power settings. These values are saved in the global `laser` structure. Since the data is returned as ASCII-decimal strings, the program must first convert these values to `double` values.

The HP 8168B/C's maximum power level varies across its wavelength range. After determining the wavelength range and step size, the `Get_TLS_PW_Range()` function steps the laser through the wavelength range measuring the maximum power at each step. The smallest maximum power found determines the maximum power level that the user can select.

The `Measurement()` function measures the source and amplifier at each wavelength. A `for` loop steps the laser from the start to stop wavelengths. At each wavelength, the `*OPC` query pauses the program until the laser settles. A `DONE?` query pauses the program until the optical spectrum analyzer completes the measurement. The `Display_GainNF()` function displays traces of gain and noise figure on the optical spectrum analyzer's screen. The `Show_Data()` function queries all measurements from the optical spectrum analyzer and displays the results on the computer's screen.

Listing: SWEPTWL.C

```
/*******************************************************
                         SWEPTWL.C
7-9-93 KGB
*******************************************************/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <process.h>
#include <ctype.h>
#include <conio.h> /* used for kbhit() */

/* ADDITIONAL HEADER FILES
The following header files come with the HP 82335A/B/I HP-IB
Interface and Command Library
*/
#include <chpib.h> /* HPIB cmd library constant declarations */
#include <cfunc.h> /* HPIB cmd library function prototypes */


/* MANIFEST CONSTANTS
*/
#define ISC 7L
#define OSA (ISC * 100L + 23L)
#define TLS (ISC * 100L + 24L)

#define SOURCE 1
#define AMPLIFIER 2
#define WAIT 3
#define START 4
#define STOP 5
#define STEP 6
#define POWER 7
#define ARRAY_SIZE 200  /* size of array for placing commands and queries */

/* FUNCTION PROTOTYPES
*/
void Initialize(void);
void Get_TLS_WL_Range(void);
void Get_TLS_PW_Range(void);
void Set_OSA_Range(void);
int Enter_Wavelength(int);
int Enter_Power(void);
int Measure(int);
void Display_GainNF(void);
void Show_Data(void);
```

```
void Show_Message(char);
void Error_Handle(int, char *);
void Send_Command(long, char *);
char *Get_Query(long, char *);
void Quit(void);

/* The following prototype is defined in chpib.h */
char *errstr (int);


/* GLOBAL VARIABLES
*/
char commands[ARRAY_SIZE + 1];
double Start_WL, Stop_WL, Step_Size, Power;

typedef struct {
        double min_wl;
        double max_wl;
        double min_power;
        double max_power;
} LASER;

LASER laser;

/****************** main() ********************
This is the main function of the program.
The program starts here.
**********************************************/
void
main()
{
        Initialize();
        Get_TLS_WL_Range(); /* query laser for maximum wavelength range */
        while (1) {
                if(!Enter_Wavelength(START))
                        break;
                if(!Enter_Wavelength(STOP))
                        break;
                if(!Enter_Wavelength(STEP))
                        break;
                Get_TLS_PW_Range(); /* query laser for power range */
                if(!Enter_Power())  /* set laser power */
                        break;
                Set_OSA_Range();
                if (!Measure(SOURCE))
                        break;
                if (!Measure(AMPLIFIER))
                        break;
                Display_GainNF();
```

```
                Show_Data();
        }
        Quit();
        exit(0);
}

/***************** Initialize() *************
This function sets the interface and tuned laser
source. It also starts the EDFA personality.
**********************************************/
void Initialize()
{
        system("cls"); /* clear the computer's screen */
        printf("\nEDFA Swept Wavelength Test\n\n");

        /* RESET THE BUS
        */
        Error_Handle(IORESET(ISC), "IORESET");
        Error_Handle(IOTIMEOUT(ISC, 10.0), "IOTIMEOUT #1");
        Error_Handle(IOCLEAR(ISC), "IOCLEAR");
        Error_Handle(IOLLOCKOUT(ISC), "IOLLOCKOUT");

        /* INITIALIZE TUNED LASER SOURCE
        */
        printf("\n\n\n\nInitializing HP 8168B/C.");
        Send_Command(TLS, "*RST;"); /* reset the TLS */
        Send_Command(TLS, ":output off;");  /* turn TLS output off */

        /* INITIALIZE OPTICAL SPECTRUM ANALYZER
        */
        printf("\nInitializing HP 71450B/1B/2B.");
        Send_Command(OSA, "IP;"); /* OSA instrument preset */

        /* START EDFA PERSONALITY
        */
        printf("\nStarting the EDFA Personality.");
        Send_Command(OSA, "EDFA_;");  /* load EDFA personality */
        printf("\nWaiting for EDFA to load . . .");
        Get_Query(OSA, "DONE?;");  /* pause until EDFA loaded */
        printf("\nTurning EDFA prompts off.");
        Send_Command(OSA, "EDFA_ JB;");  /* turn EDFA prompt messages off */
        printf("\nSelecting Swept Wavelength Testing.");
        Send_Command(OSA, "EDFA_ GC;");  /* select swept wavelength test */
        printf("\nWaiting for EDFA to finish tasks . . .");
        Get_Query(OSA, "DONE?;");   /* pause until test is selected */
        return;
}

/*************** Get_TLS_WL_Range() *********
```

```
This function queries the TLS wavelength range.
*********************************************/
void Get_TLS_WL_Range()
{
        printf("\nFinding TLS's minimum wavelength.");
        Send_Command(TLS, ":sour:wave min;");  /* set to minimum value */
        Get_Query(TLS, ":sour:wave?;");        /* query minimum wavelength */
        laser.min_wl = atof(commands);         /* convert to float */
        laser.min_wl *= 1000000000;    /* multiply to convert to nanometers */

        printf("\nFinding TLS's maximum wavelength.");
        Send_Command(TLS, ":sour:wave max;");  /* set to minimum value */
        Get_Query(TLS, ":sour:wave?;");        /* query maximum wavelength */
        laser.max_wl = atof(commands);         /* convert to float */
        laser.max_wl *= 1000000000;    /* multiply to convert to nanometers */
        return;
}

/****************** Enter_Wavelength() *********
This function enters the measurement range.
*********************************************/
int Enter_Wavelength(int end)
{
        switch (end) {
        case START:
                Error_Handle(IOLLOCKOUT(ISC), "IOLLOCKOUT"); /* locks out front panels */
                Show_Message(START);
                while (1) {
                        gets(commands);
                        strlwr(commands);
                        if (strcmp(commands, "q") == 0)  /* 'q' entered, exit program */
                                return 0;
                        Start_WL = atof(commands);
                        if (Start_WL >= laser.min_wl && Start_WL <= laser.max_wl)
                                break;
                        printf("\nBad value, try again. Or, enter 'q' to exit: ");
                }
                break;
        case STOP:
                Show_Message(STOP);
                while (1) {
                        gets(commands);
                        strlwr(commands);
                        if (strcmp(commands, "q") == 0)  /* 'q' entered, exit program */
                                return 0;
                        Stop_WL = atof(commands);
                        if (Stop_WL >= laser.min_wl && Stop_WL <= laser.max_wl)
                                break;
                        printf("\nBad value, try again. Or, enter 'q' to exit: ");
```

```
                }
                break;
        case STEP:
                Show_Message(STEP);
                while (1) {
                        gets(commands);
                        strlwr(commands);
                        if (strcmp(commands, "q") == 0)  /* 'q' entered, exit program */
                                return 0;
                        Step_Size = atof(commands);
                        if (Step_Size >= 0.1 && Step_Size <= 100.0)
                                break;
                        printf("\nBad value, try again. Or, enter 'q' to exit: ");
                }
                break;
        }
        return 1;
}

/**************** Get_TLS_PW_Range() *********
This function queries the TLS min and max power at
each stepped wavelength.

The minimum power fount is the greatest minimum power
that can be set across the whole wavelength range.
The maximum power found is the smallest maximum power
that can be set across the whole wavelength range.
**********************************************/
void Get_TLS_PW_Range()
{
        int dec, sign, precision = 3;
        double min, max, i;
        char *p_buffer;

        laser.min_power = 0.0; /* initialize arbitrarily large */
        laser.max_power = 1.0; /* initialize arbitrarily small */

        printf("\nFinding TLS's power range for selected wavelength range.");
        /* step through each wavelength included in measurement */
        for (i = Start_WL; i <= Stop_WL; i += Step_Size) {
                p_buffer = fcvt(i, precision, &dec, &sign); /* convert i to a string */
                sprintf(commands, ":sour:wave %spm;", p_buffer);
                Send_Command(TLS, commands);  /* set TLS wavelength */
                Get_Query(TLS, "*OPC?;");     /* wait for TLS to step to wavelength */
                Get_Query(TLS, ":pow min;:pow?;");  /* query minimum power */
                min = atof(commands);               /* convert to float */
                Get_Query(TLS, ":pow max;:pow?;");  /* query maximum power */
                max = atof(commands);               /* convert to float */
                if (min > laser.min_power)
```

```
                        laser.min_power = min;          /* save if greater than */
                if (max < laser.max_power)
                        laser.max_power = max;          /* save is smaller than */
        }
        laser.min_power *= 1000000;     /* convert to microwatts */
        laser.max_power *= 1000000;
        return;
}

/***************** Enter_Power() *********
This function asks user to set the power level.
Between the minimum and maximum values found by
the Get_TLS_PW_Range() function.
**********************************************/
int Enter_Power()
{
        int ch, dec, sign, precision = 0;
        char *p_buffer;

        Show_Message(POWER);
        while (1) {
                gets(commands);  /* get user's choice */
                strlwr(commands);
                if (strcmp(commands, "q") == 0)  /* 'q' entered, exit program */
                        return 0;
                Power = atof(commands);  /* convert to a float */

                if (Power >= laser.min_power && Power <= laser.max_power)
                        break;                          /* break loop if power level valid */
                printf("\nBad value, try again. Or, enter 'q' to exit: ");
        }
        p_buffer = fcvt(Power, precision, &dec, &sign); /* convert power to a string */
        sprintf(commands, ":pow %sUW;", p_buffer);
        Send_Command(TLS, commands);                    /* set TLS power level */
        return 1;
}

/****************** Set_OSA_Range() **********
This function sets the OSA's wavelength and sensitivity.
**********************************************/
void Set_OSA_Range()
{
        printf("\nSetting HP 71450B/1B/2B measurement range.");
        sprintf(commands, "STARTWL %.3fNM;STOPWL %.3fNM;SENS -65DBM;",
                Start_WL - 2, Stop_WL + 2);
        Send_Command(OSA, commands);
        return;
}
```

```
/****************** Measure() *********
This function sets the EDFA personality to its source
or amplifier measurement. Then the TLS is stepped
from its start to step wavelength.
*********************************************/
int Measure(int device)
{
        int ch, dec, sign, precision = 3;
        double i;
        char *p_buffer;

        switch (device) {
        case SOURCE:
                Show_Message(SOURCE);
                Send_Command(OSA, "EDFA_ AC;"); /* select source measurement */
                break;
        case AMPLIFIER:
                Show_Message(AMPLIFIER);
                Send_Command(OSA, "EDFA_ AF;"); /* select amplifier measurement */
                break;
        }
        Show_Message(WAIT);
        Send_Command(TLS, ":output on;"); /* turn laser output on */
        Send_Command(OSA, "SNGLS;");

        /* step from start to stop wavelength */
        for (i = Start_WL; i <= Stop_WL; i += Step_Size) {
                if (kbhit()) {  /* test if user wants to abort program */
                        ch = getch();
                        if (ch == 'q' || ch == 'Q') {   /* exit program */
                                Send_Command(TLS, ":output off;"); /* turn laser output on */
                                Send_Command(OSA, "EDFA_ D;");  /* stop measurement */
                                Send_Command(OSA, "CONTS;");
                                return 0;
                        }
                        else while(kbhit()) /* else clear keyboard buffer */
                                        getch();
                }
                p_buffer = fcvt(i, precision, &dec, &sign); /* convert i to a string */
                sprintf(commands, ":sour:wave %spm;", p_buffer);
                Send_Command(TLS, commands);  /* set TLS wavelength */
                Get_Query(TLS, "*OPC?;");     /* wait for TLS to step to new wavelength */
                Send_Command(OSA, "TS;");     /* sweep OSA */
                Get_Query(OSA, "DONE?;");     /* wait for OSA sweep to finish */
        }
        Send_Command(TLS, ":output off;");  /* turn laser output off */
        Send_Command(OSA, "EDFA_ D;");  /* stop measurement */
        Send_Command(OSA, "CONTS;");
        return 1;
```

```
}

/*************** Display_GainNF() ************
This function displays the traces of Gain and
Noise Figure on the optical spectrum analyzer.
*********************************************/
void Display_GainNF()
{
        system("cls"); /* clear the computer's screen */
        printf("\n\n\n\nDisplaying the Gain and Noise Figure on OSA.");

        /* display and autoscale gain and noise figure traces */
        Send_Command(OSA, "EDFA_ AI;"); /* select "Display Data" menu */
        Send_Command(OSA, "EDFA_ AJ;"); /* select "data select" menu */
        Send_Command(OSA, "EDFA_ CA;"); /* display gain & noise figure traces */
        Send_Command(OSA, "EDFA_ B;");  /* autoscale the traces */
        Get_Query(OSA, "DONE?;");  /* wait for traces to be displayed */
        return;
}

/****************** Show_Data() **************
This function queries the EDFA personality for
all the displayed measurements and settings.
*********************************************/
void Show_Data()
{
        double value;
        system("cls");
        printf("\n\t\t\t%s\n\n\t%s\t%s\n\n",
                "EDFA Measurement Results",
                "Parameter",
                "    Value");

        Get_Query(OSA, "EDFA_ YA?;");
        value = atof(commands);
        value *= 1000000000;
        printf("\tWavelength:         %6.2f nm\n", value);
        Get_Query(OSA, "EDFA_ YB?;");
        value = atof(commands);
        value *= 1000000000;
        printf("\tRes BW:             %6.2f nm\n", value);
        Get_Query(OSA, "EDFA_ YC?;");
        value = atof(commands);
        printf("\tInput Power:        %6.2f dBm\n", value);
        Get_Query(OSA, "EDFA_ YD?;");
        value = atof(commands);
        printf("\tOutput Power:       %6.2f dBm\n", value);
        Get_Query(OSA, "EDFA_ YF?;");
        value = atof(commands);
```

```
        printf("\tEDFA Gain:            %6.2f dB\n", value);
        Get_Query(OSA, "EDFA_ YE?;");
        value = atof(commands);
        printf("\tNoise Figure:         %6.2f dB\n", value);
        Get_Query(OSA, "EDFA_ YG?;");
        value = atof(commands);
        printf("\tInput S/N:            %6.2f dB\n", value);
        Get_Query(OSA, "EDFA_ YH?;");
        value = atof(commands);
        printf("\tOutput S/N:           %6.2f dB\n", value);
        Get_Query(OSA, "EDFA_ YI?;");
        value = atof(commands);
        printf("\tEDFA S/N:             %6.2f dB\n", value);
        Get_Query(OSA, "EDFA_ YJ?;");
        value = atof(commands);
        value *= 1000000000;
        printf("\tNoise BW:             %6.2f nm\n", value);
        Get_Query(OSA, "EDFA_ YK?;");
        value = atof(commands);
        printf("\tIntegrated ASE:       %6.2f dBm\n", value);
        Get_Query(OSA, "EDFA_ YL?;");
        value = atof(commands);
        value *= 1000000000;
        printf("\tIntegrated BW:        %6.2f nm\n\n", value);

        Error_Handle(IOLOCAL(ISC), "IOLOCAL");  /* put OSA & TLS in "local" mode */

        while(kbhit())
                getch();
        printf("Press any key to continue . . .");
        getch();
        return;
}

/*************** Show_Message() **************
This function displays setup prompts on the PC's
screen.
**********************************************/
void Show_Message(char message)
{
        system("cls");
        switch (message) {
        case SOURCE:
                printf("\n\n\n\n%s\n\n%s",
                "1. Connect the HP 8168B/C output to the OSA.",
                "2. Press any key to continue.");
                while(kbhit())
                        getch();
                getch();
```

```
        break;
case AMPLIFIER:
        printf("\n\n\n\n%s\n\n%s\n\n%s\n\n%s",
        "1. Disconnect the source from the OSA.",
        "2. Connect the source to the amplifier's input.",
        "3. Connect the amplifier's output to the OSA.",
        "4. Press any key to continue.");
        while(kbhit())
                getch();
        getch();
        break;
case WAIT:
        printf("\n\n\n\n\nPlease wait for the measurement to complete.\n\n");
        printf("\t%s%.3f%s\n\t%s%.3f%s\n\t%s%.3f%s\n\t%s%.0f%s\n",
                "Start wavelength:  ", Start_WL, " nm",
                "Stop wavelength:   ", Stop_WL, " nm",
                "Step size:         ", Step_Size, " nm",
                "Power:             ", Power, " uw");
        printf("\n\nTo abort program, press 'q'.");
        break;
case START:
        printf("\n\nPress 'q' at any time to exit program.");
        printf("\n\n%s\n%.3f and %.3f %s",
                "Enter a starting wavelength between ",
                laser.min_wl,
                laser.max_wl,
                "nanometers: ");
        break;
case STOP:
        printf("\n\nPress 'q' at any time to exit program.");
        printf("\n\n%s\n%.3f and %.3f %s",
                "Enter an ending wavelength between ",
                laser.min_wl,
                laser.max_wl,
                "nanometers: ");
        break;
case STEP:
        printf("\n\nPress 'q' at any time to exit program.");
        printf("\n\nEnter an wavelength step size in nanometers: ");
        break;
case POWER:
        printf("\n\nPress 'q' at any time to exit program.");
        printf("\n\nFor the %.3f to %.3f nanometer wavelength range,\n",
                Start_WL,
                Stop_WL);
        printf("%s %.0f UW and %.0f UW %s\n\n",
                "enter a laser power between ",
                laser.min_power,
                laser.max_power,
```

```
                    "in microwatts: ");
                break;
        }
        return;
}

/************* Send_Command() ****************
This function sends HP-IB commands to OSA or TLS.
**********************************************/
void Send_Command(long inst, char *string)
{
        Error_Handle(IOOUTPUTS(inst, string, strlen(string)), "IOOUTPUTS");
        return;
}

/************* Get_Query() ******************
This function gets queries from OSA or TLS.
The data is placed in the global commands string.
The I/O timeout is increased to allow instrument
to process commands before returning query data.
**********************************************/
char *Get_Query(long inst, char *string)
{
        int length = ARRAY_SIZE;
        Error_Handle(IOTIMEOUT(ISC, 40.0), "IOTIMEOUT #1");
        Send_Command(inst, string);
        Error_Handle(IOENTERB(inst, commands, &length, 1), "IOENTERB #1");
        commands[length] = '\0'; /* append NULL byte */
        Error_Handle(IOTIMEOUT(ISC, 5.0), "IOTIMEOUT #2");
        return commands;
}

/*************** Quit() **********
This function changes settings and locals instruments.
**********************************************/
void Quit()
{
        system("cls"); /* clear the computer's screen */
        Send_Command(OSA, "EDFA_ Q;");   /* exit EDFA personality */
        Send_Command(OSA, "CONTS;");     /* select continuous OSA sweeps */
        Error_Handle(IOLOCAL(ISC), "IOLOCAL");  /* put OSA & TLS in "local" mode */
}

/******** Error_Handle(int, char *) **********
This function prints HP-IB errors on the PC's
display.
**********************************************/
void Error_Handle(int error, char *routine)
{
```

```
        if (error != NOERR) {
                printf("HPIB error in call to %s: %d, %s\n",
                        routine, error, errstr(error));
                exit(1);
        }
        return;
}
```

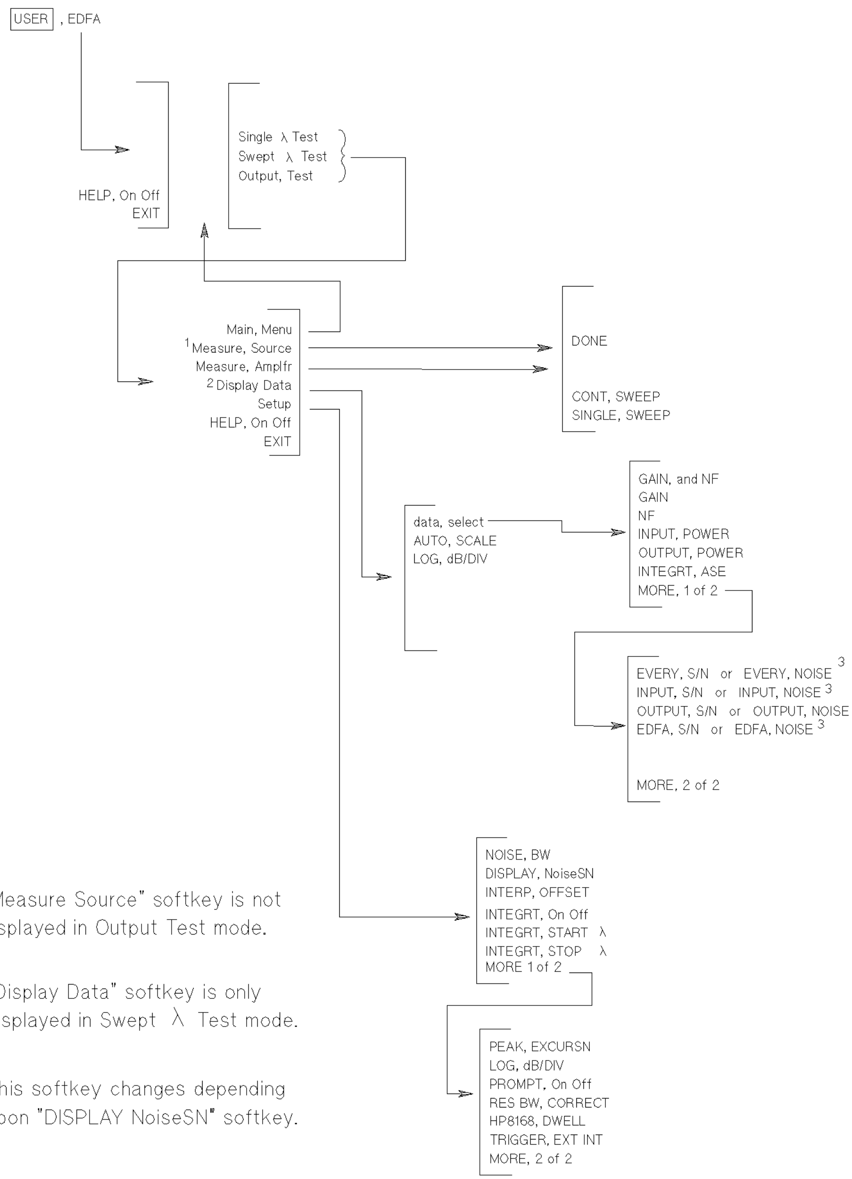
4

Reference

Reference

Chapter Contents

Menu Map



1. "Measure Source" softkey is not displayed in Output Test mode.

2. "Display Data" softkey is only displayed in Swept λ Test mode.

3. This softkey changes depending upon "DISPLAY NoiseSN" softkey.

Error Messages

This section defines error messages for the EDFA test personality. Error messages not defined in this section originate with the HP 70950B/1B/2B module and are documented in the *HP 71450B/1B/2B Optical Spectrum Analyzers Reference*.

Error messages can result from incorrect operating procedures, illegal programming commands, or hardware failures. Normally, the optical spectrum analyzer removes error messages from the screen as soon as the error conditions are corrected. If you have a computer, error messages can be retrieved via HP-IB by executing the `ERR?` command. Refer to the *HP 71450B/1B/2B Optical Spectrum Analyzers Programmer's Guide* for information on the `ERR?` command.

error 111 `EDFA Memory Error:` The program has run out of free memory. To free memory, perform the following:

- Erase objects of the following types from memory: saved traces, limit lines, user menus, and downloadable programs. Refer to the chapter on managing memory in the *HP 71450B/1B/2B Optical Spectrum Analyzers User's Guide* to learn how to erase these objects.

error 112 `EDFA Noise Figure < 0 Error:` Invalid data has resulted in a noise figure calculation less than zero. Valid noise figure results are always positive values.

error 116 `EDFA Command Error:` An unrecognized argument has been used with the programming command. Refer to Chapter 3 for the correct use of the command.

Measurement Calculations

The following calculations describe each measurement result that is shown at the top of the display. A special `typeface` is used in this chapter to indicate these displayed quantities. For example, when the wavelength measurement is referenced, it is printed as **`Wavelength`**.

EDFA Gain

The **`EDFA Gain`** is defined by the following equation:

$$EDFA\ Gain = \frac{P_{out}}{P_{in}}$$

where:

P_{out} is the calculated **`Output Pwr`** at **`Wavelength`**.

P_{in} is the calculated **`Input Pwr`** at **`Wavelength`**.

`EDFA Gain` is not measured during the Output Test.

EDFA Noise

The EDFA amplifier's noise, **EDFA Noise**, is performed at **Wavelength** and is defined by the following equation:

$$EDFA\ Noise = (N_{out} - N_{in}G)\left(\frac{Noise\ BW}{Res\ BW}\right)$$

where:

N_{in} is average input noise power at **Wavelength**. It is calculated using the interpolation offset value. Refer to **Interpolation Offset** in this section. N_{in} is not measured during the Output Test.

N_{out} is average output noise power at **Wavelength**. It is calculated using the interpolation offset value. Refer to **Interpolation Offset** in this section. However, during a Output Test if integration is turned on (see the Setup menus's `INTEGRT On Off` softkey), N_{out} is the integrated noise value.

G is **EDFA Gain** as defined in this chapter.

Res BW is the optical spectrum analyer's true resolution bandwidth. It is a function of wavelength.

Noise BW is entered using the `NOISE BW` softkey. The default value is 1 nm.

Input S/N is not measured during the Output Test.

EDFA S/N

The EDFA amplifier's signal-to-noise ratio, **EDFA S/N**, is performed at **Wavelength** and is defined by the following equation:

$$EDFA\ S/N = \left[\frac{P_{signal}}{N_{out} - N_{in}G}\right]\left(\frac{Noise\ BW}{Res\ BW}\right)$$

where:

P_{signal} is the output peak power at **Wavelength**.

N_{in} is average input noise power at **Wavelength**. It is calculated using the interpolation offset value. Refer to **Interpolation Offset** in this section. N_{in} is not measured during the Output Test.

N_{out} is average output noise power at **Wavelength**. It is calculated using the interpolation offset value. Refer to **Interpolation Offset** in this section. However, during a Output Test if integration is turned on (see the Setup menus's INTEGRT On Off softkey), N_{out} is the integrated noise value.

G is **EDFA Gain** as defined in this chapter.

Res BW is the optical spectrum analyer's true resolution bandwidth. It is a function of wavelength.

Noise BW is entered using the NOISE BW softkey. The default value is 1 nm.

Input S/N is not measured during the Output Test.

Input Noise

The **Input Noise** is defined by the following equation:

$$Input\ Noise = N_{in}\left(\frac{Noise\ BW}{Res\ BW}\right)$$

where:

N_{in} is the average input noise power at **Wavelength**. It is calculated using the interpolation offset value. Refer to **Interpolation Offset** in this section.

Res BW is the optical spectrum analyzer's true resolution bandwidth. It is a function of wavelength.

Noise BW is entered using the **NOISE BW** softkey. The default value is 1 nm. This value can be changed

Input Pwr

The **Input Pwr** is defined by the following equation:

$$Input\ Pwr = P_{signal} - N_{in}$$

where:

P_{signal} is the measured input power at **Wavelength**.

N_{in} is the interpolated average input noise power at **Wavelength**. It is calculated using the interpolation offset value. Refer to **Interpolation Offset** in this section.

Input Pwr is not measured during the Output Test.

Input S/N is not measured during the Output Test.

Input S/N

The input signal-to-noise ratio, `Input S/N`, is defined by the following equation:

$$Input\ S/N = \left(\frac{P_{signal}}{N_{in}}\right)\left(\frac{Noise\ BW}{Res\ BW}\right)$$

where:

P_{signal} is the input power at `Wavelength`.

N_{in} is the average input noise power at `Wavelength`. It is calculated using the interpolation offset value. Refer to `Interpolation Offset` in this section.

`Res BW` is the optical spectrum analyer's true resolution bandwidth. It is a function of wavelength.

`Noise BW` is entered using the `NOISE BW` softkey. The default value is 1 nm. This value can be changed

`Input S/N` is not measured during the Output Test.

Integrt ASE

The integrated amplified spontaneous emission, `Integrt ASE`, is defined by the following equation:

$$Integrt\ ASE = \sum_{\lambda=\lambda_1}^{\lambda=\lambda_2} N_\lambda$$

where:

λ_1 is the left-integration start wavelength set by the `INTEGRT STRT WL` softkey.

λ_2 is the right-integration stop wavelength set by the `INTEGRT STOP WL` softkey.

N_λ is the output noise at the wavelength. When

$$(\lambda_{pk} - i) \leq \lambda \leq (\lambda_{pk} + i),$$

the integrated noise value is the interpolated noise value for that wavelength range. The value i is the interpolation offset as described in `Interpolation Offset` in this section.

$$N_\lambda = \frac{N_{(\lambda+i)} + N_{(\lambda-i)}}{2}$$

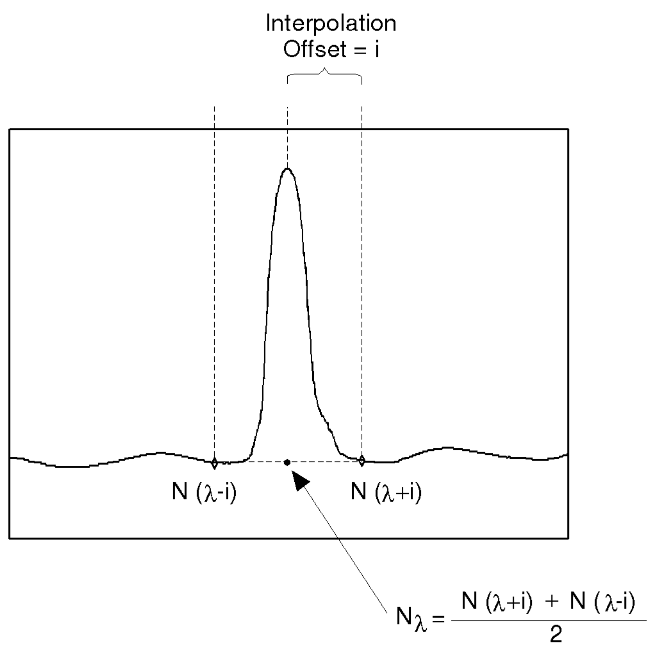
If integration is turned off during Swept λ Tests (see the Setup menus's `INTEGRT On Off` softkey), the program sets the `Integrt ASE` value to -100 dBm at each wavelength. This means that the displayed trace will also be at -100 dBm.

Integrt BW

`Integrt BW` is not a measurement. This value is the integration bandwidth used to measure `Integrt ASE`. The default value is the current wavelength span of the optical spectrum analyzer.

Interpolation Offset

The Interpolation Offset value is used to determine the noise level at the signal. As shown in the following figure, the interpolation offset locates two wavelength points for measuring noise. These noise measurements are then averaged to determine the noise level at the signal.



The average noise value is used between interpolation offset points.

The interpolation offset is entered using INTERP OFFSET in the Setup menu. It has a default value of 1 nm.

Noise BW

Noise BW is not a measurement. This value is the noise bandwidth value that is used in the calculations. The default value is 1 nm. This value can be changed using the **Noise BW** softkey.

Noise Fig

The **Noise Fig**ure at **Wavelength** is defined by the following equation:

$$Noise\ Fig = \frac{N_{out} - N_{in}G}{h\nu GB} + \frac{1}{G}$$

where:

N_{out} is average output noise power at **Wavelength**. It is calculated using the interpolation offset value. Refer to **Interpolation Offset** in this section.

N_{in} is average input noise power at **Wavelength**. It is calculated using the interpolation offset value. Refer to **Interpolation Offset** in this section.

G is the **EDFA Gain** as defined in this chapter.

B is the optical spectrum analyer's true resolution bandwidth in hertz.

h is 6.626×10^{-34}.

ν is $\frac{c}{\lambda}$.

c is 2.99711×10^{8}.

Noise Figure is not measured during the Output Test.

Output Noise

The Output Noise is defined by the following equation:

$$Output\ Noise = N_{out}\left(\frac{Noise\ BW}{Res\ BW}\right)$$

where:

N_{out} is the average output noise power at Wavelength. It is calculated using the interpolation offset value. Refer to Interpolation Offset in this section.

Res BW is the optical spectrum analyzer's true resolution bandwidth. It is a function of wavelength.

Noise BW is entered using the NOISE BW softkey. The default value is 1 nm.

Output Pwr

The Output Pwr is defined by the following equation:

$$Output\ Pwr = P_{signal} - N_{out}$$

where:

P_{signal} is the measured output power at Wavelength.

N_{out} is the average output noise power at Wavelength. It is calculated using the interpolation offset value. Refer to Interpolation Offset in this section.

Output S/N

The output signal-to-noise ratio, `Output S/N`, is defined by the following equation:

$$Output\ S/N = \left(\frac{P_{signal}}{N_{out}}\right)\left(\frac{Noise\ BW}{Res\ BW}\right)$$

where:

P_{signal} is the output power at **Wavelength**.

N_{out} is the average output noise power at **Wavelength**. It is calculated using the interpolation offset value. Refer to **Interpolation Offset** in this section.

Res BW is the optical spectrum analyzer's true resolution bandwidth. It is a function of wavelength.

Noise BW is entered using the NOISE BW softkey. The default value is 1 nm.

Res BW

Res BW is not a measurement. This value shows the optical spectrum analyzer's true resolution bandwidth. It is a function of wavelength and may differ slightly from the normal screen annotation value. The final step in calculating this value is applying a resolution-bandwidth correction factor:

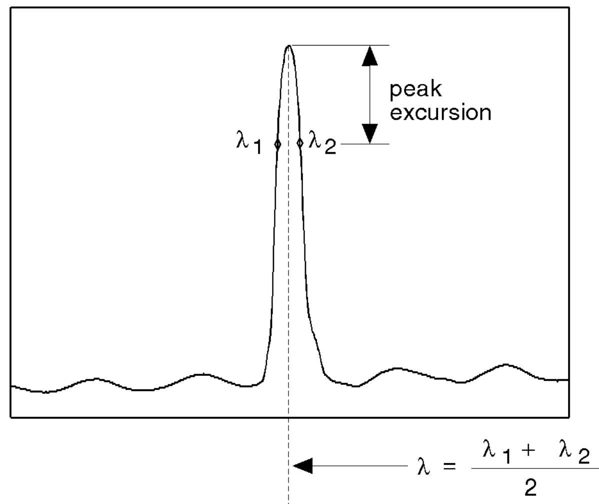
$$Res\ BW = (true\ Res\ BW)(correction\ factor)$$

This factor corrects for variations in the optical spectrum analyzer's slitwheel. (The slitwheel determines bandwidth, and is mounted in the monochromator.)

Use the Res BW CORRECT softkey to enter resolution-bandwidth correction factors. The default value is 1 (100%).

Wavelength

The source **Wavelength** is defined in the following figure and equation:



wavelen

where:

λ_1 is the wavelength to the left side of the signal peak that is equal in amplitude to the signal peak minus the peak excursion value.

λ_2 is the wavelength to the right side of the signal peak that is equal in amplitude to the signal peak minus the peak excursion value.

Peak excursion values can be entered using the PEAK EXCURSN softkey. The default value is 3 dB.

Measurement Uncertainty

The following table summarizes the error terms and shows the typical total measurement uncertainties. The total uncertainties are calculated as shown in the following equation:

$$uncertainty = 2\sqrt{\sum \frac{U^2}{3}}$$

where "U" is the uncertainty of each individual term.

These uncertainty calculations are based on the use of linear interpolation measurements, with an HP 71452B Optical Spectrum Analyzer. For more information, refer to application note HP 71452-1.

Measurement Uncertainties

	Single Wavelength Measurement	
	Gain	**Noise Figure**
Connector Uncertainty:		
with splices	3 x 0.05	2 x 0.05
with connectors	3 x 0.25	2 x 0.25
Source:		
Stability	0.05 dB	0.05
Optical Spectrum Analyzer:		
Absolute Accuracy	—	0.10
Polarization Sensitivity	2 x 0.05 dB	2 x 0.05 (gain)
Scale Fidelity	0.07	0.07 (gain)
	—	0.07 (N_{out})
	—	0.004 (N_{in})
Flatness	—	0.19 dB
Resolution Bandwidth Accuracy	—	0.12 dB
Internal Etalons	0.03 dB	0.03 dB
Dynamic Range	—	0.05 dB
Total Uncertainty:		
with splices	±0.17	±0.36
with connectors	±0.52	±0.53

Index

Index