Saber® Tools Reference

Version Z-2007.03, March 2007

SYNOPSYS®

Copyright Notice and Proprietary Information

Right to Copy Documentation

The license agreement with Synopsys permits licensee to make copies of the documentation for its internal use only. Each copy shall include all copyrights, trademarks, service marks, and proprietary rights notices, if any. Licensee must assign sequential numbers to all copies. These copies shall contain the following legend on the cover page:

> "This document is duplicated with the permission of Synopsys, Inc., for the exclusive use of
> ____ and its employees. This is copy number __________."

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Registered Trademarks (®)

Synopsys, AMPS, Cadabra, CATS, CRITIC, CSim, Design Compiler, DesignPower, DesignWare, EPIC, Formality, HSIM, HSPICE, iN-Phase, in-Sync, Leda, MAST, ModelTools, NanoSim, OpenVera, PathMill, Photolynx, Physical Compiler, PrimeTime, SiVL, SNUG, SolvNet, System Compiler, TetraMAX, VCS, Vera, and YIELDirector are registered trademarks of Synopsys, Inc.

Trademarks (™)

AFGen, Apollo, Astro, Astro-Rail, Astro-Xtalk, Aurora, AvanWaves, Columbia, Columbia-CE, Cosmos, CosmosEnterprise, CosmosLE, CosmosScope, CosmosSE, DC Expert, DC Professional, DC Ultra, Design Analyzer, Design Vision, DesignerHDL, Direct Silicon Access, Discovery, Encore, Galaxy, HANEX, HDL Compiler, Hercules, Hierarchical Optimization Technology, HSIM^{plus}, HSPICE-Link, iN-Tandem, i-Virtual Stepper, Jupiter, Jupiter-DP, JupiterXT, JupiterXT-ASIC, Liberty, Libra-Passport, Library Compiler, Magellan, Mars, Mars-Xtalk, Milkyway, ModelSource, Module Compiler, Planet, Planet-PL, Polaris, Power Compiler, Raphael, Raphael-NES, Saturn, Scirocco, Scirocco-i, Star-RCXT, Star-SimXT, Taurus, TSUPREM-4, VCS Express, VCSi, VHDL Compiler, VirSim, and VMC are trademarks of Synopsys, Inc.

Service Marks (SM)

MAP-in, SVP Café, and TAP-in are service marks of Synopsys, Inc.

SystemC is a trademark of the Open SystemC Initiative and is used under license.
ARM and AMBA are registered trademarks of ARM Limited.
Saber is a registered trademark of SabreMark Limited Partnership and is used under license.
All other product or company names may be trademarks of their respective owners.

Printed in the U.S.A.

Saber® Tools Reference, Z-2007.03

Contents

1

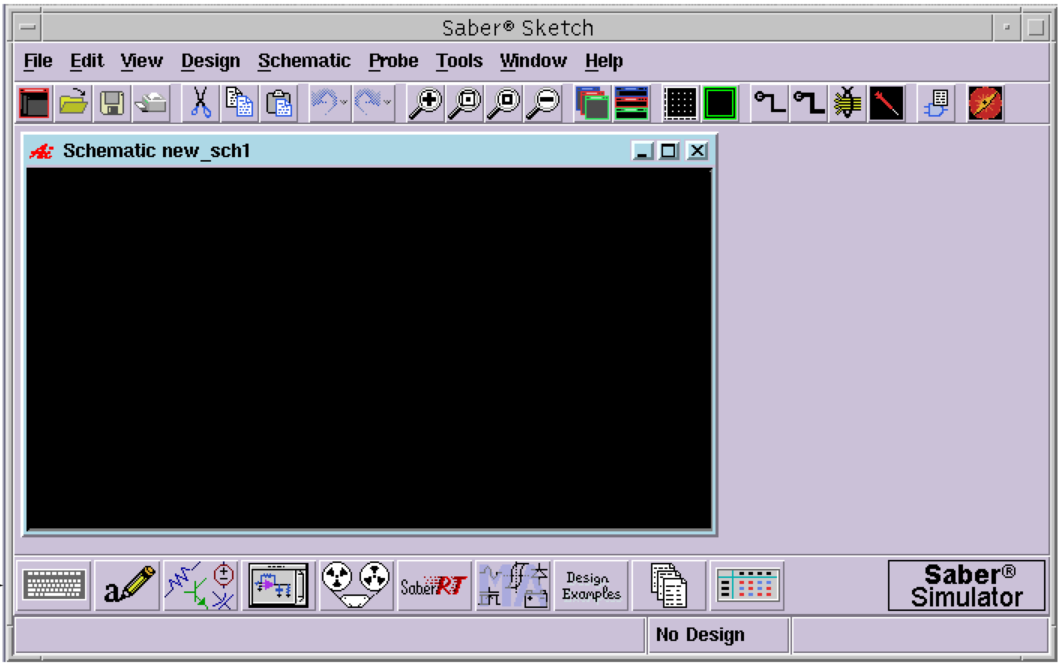
Overview

This chapter introduces the Saber Tools icon bar.

Saber Tools

Saber has a variety of tools that are available in the tools icon bar at the bottom of the work area, as shown in Figure 1.

Figure 1 Saber Tools Icon Bar



Tools icon bar

The tools in the icon bar changes depending on which application you are running. Figure 1shows the icon bar for Saber Sketch.

The following table lists all of the Saber tools. You may not see all of these tools in the icon bar. The tools are listed in the general order they appear in the toolbar.

Tool Icon	Function
Command Line Tool	Open the Command Line tool. This tool allows you to enter Aim commands, write scripts, and save them into files. For more information, see Using the Command Line Tool on page 7
AimDraw Tool	Open the AimDraw tool. This tool allows you to create symbols, general graphic objects and text. For more information, see Using the Draw Tool on page 11
Parts Gallery	Open the Saber Parts Gallery. This tool allows you to browse the parts libraries, select parts and place them in a schematic. For more information, see Using the Parts Gallery on page 35.
Design Tool	Open the Saber Design Tool. This tool displays a list of all schematics in a design hierarchy. For more information, see Using the Design Tool on page 69
Signal Manager	Open the CosmosScope Signal Manager, which manages the signals generated by a design analysis. For more information, see Using the Signal Manager on page 73
Measurement Tool	Open the CosmosScope Measurement Tool. It enables you to perform a variety of measurement operations on displayed waveforms in the CosmosScope Waveform Analyzer. For more information, see Using the Measurement Tool on page 91

Tool Icon	Function
Waveform Calculator	Open the CosmosScope Waveform Calculator, which operates as an RPN (Reverse Polish Notation) calculator or as an algebraic calculator. The calculator enables you to perform operations on AIM language expressions and waveforms as well as on numbers. For more information, see Using the Waveform Calculator on page 237
Macro Recorder	Open the Macro Recorder tool. This tool records a series of actions, allows you to edit these actions, and plays them back as a script. For more information, see Using the Macro Recorder on page 269
Saber RT SaberRT	Open SaberRT, an interface to the Saber Simulator allowing designs to be simulated interactively in a real-time or hardware-in-the-loop context. For more information, see Using the SaberRT Interface on page 285
Model Architect	Open the Model Architect palette. This is a collection of tools used to characterize or create models. For more information, see the *Saber Model Architect Tool User Guide*.
Design Examples Design Examples	Open the Design Examples browser where you can easily access design examples. For more information, see Viewing Design Examples on page 307.
RF Tool	Open the CosmosScope RF Tool. This tool enables you to perform special measurements and calculations when running RF analyses. For more information, see Using the RF Tool on page 309

Tool Icon	Function
MATLAB Interface	Open the CosmosScope Analysis Interface to MATLAB. The interface opens a transcript window that enables you to access MATLAB software and to transfer data between Saber applications and MATLAB applications For more information, see Using the CosmosScope MATLAB Interface on page 329
Report	Open the Saber Report tool. This tool allows you to edit Saber Simulator files or any ASCII text. It is also linked to the Saber Simulator so that new reports can be generated and displayed in the report window. For more information, see Using the Report Tool on page 323
Testify	Open the Saber Testify Test Manager tool. Testify allows you to use the Saber Simulator to develop and evaluate tests used for detecting fault conditions on a circuit board. For more information, see Using Testify on page 345
Table Manager	Open the Table Manager. This tool allows you to extract manufacturing information, such as wire lists, weight tables and bills of material from a design. For more information, see Generating Tables in the *Saber Harness User Guide*.
Connector Manager	Open the Connector Manager. This tool gathers inline connectors together and associates them with a connector block. For more information, see Connector Manager Overview in the *Saber Harness User Guide*.
Assembly Tool	Open the Assembly tool. This tool allows you to manage assemblies. For more information, see Creating an Assembly in the *Saber Harness User Guide*.

Tool Icon	**Function**
Bundle Tool	Open the Bundle tool. This tool allows you to manage wire bundles in assemblies. For more information, see Placing Elements on a Bundle Drawing in the *Saber Harness User Guide*.

2

Using the Command Line Tool

This chapter explains how to use the Command Line Tool.

The Command Line Tool allows you to enter AIM commands, write scripts, and save them into files. You can also view a command log, which is a transcript of the AIM language used to operate Saber or CosmosScope. You can use the AIM language displayed in the log to create scripts with the Macro Recorder tool.

Note:

> Saber Simulator commands must be entered in the Guide Transcript window.

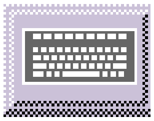
Text and waveforms can be selected from other sources and pasted directly into the Command Line Tool transcript window.

The following topics describe the Command Line Tool:

- Accessing the Command Line Tool
- Executing AIM Scripts and Commands
- Command Line Tool Menus

Accessing the Command Line Tool

The Command Line Tool icon is located in the Tool bar at the bottom of the work surface.



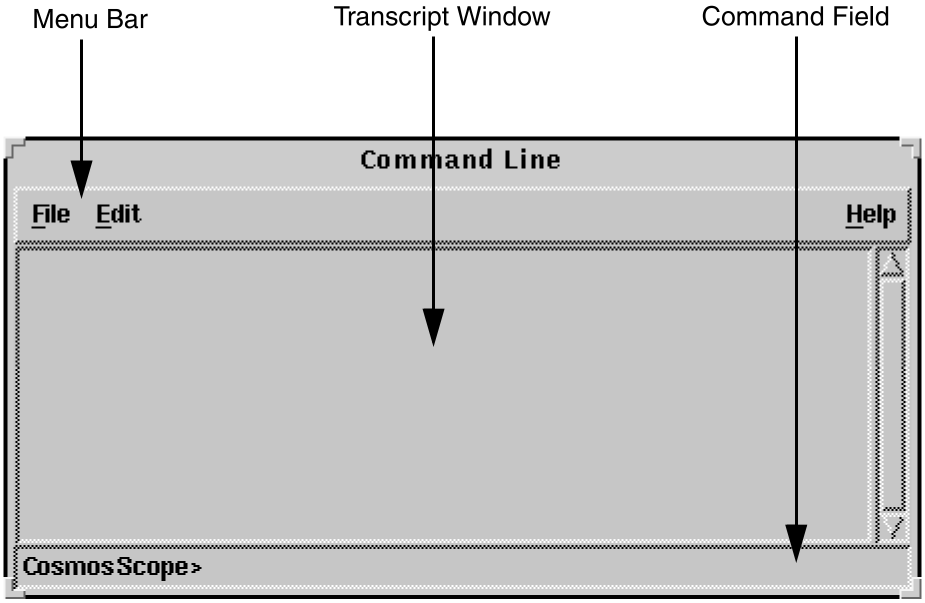
To open or close the tool, click the icon with the left mouse button.

Executing AIM Scripts and Commands

To run an existing AIM script, choose **File > Open** from the Command Line window menu bar.

To execute an AIM command, type the command in the Command field and press the Enter key.

Figure 2 AIM Command Line Tool Window



Command Line Tool Menus

The File menu allows you to open AIM script files, save the contents of the Command Line window, and close the Command Line Tool. The following File menu items are available:

Open	Opens and runs an AIM script.
Save	Saves the contents of the Transcript window with the specified current name and file location.
Save As	Saves the contents of the Transcript window with a name and file location of your choosing.

Close Window Closes the Command Line Tool.

The Edit menu allows you to cut, copy, and clear text from the transcript window, set the size of the transcript window, and toggle the logging function. The following Edit menu items are available:

Cut Cuts selected text from the Command Line window.

Copy Copies selected text from the Command Line window.

Paste Pastes text into the Command field.

Clear Deletes all text from the Transcript window.

Transcript Lines Sets the maximum number of lines of text the transcript window to display.

Display Logging Toggles the logging function.

Viewing Log Files

The log is a transcript of the AIM language used to operate Saber or CosmosScope. You can use the AIM language displayed in the log to create scripts with the AIM Macro Recorder tool.

An ASCII text record of the log, named your_application.log, is automatically created in the directory that you started your session.

To view the commands being logged, click the AIM Command Line Tool in the tool bar. The Command Line window appears. Next, choose **Edit > Display Logging** from the Command Line window. The log is displayed in the transcript window as operations are performed.

AIM Language References

AIM is a super-set of the Tcl/Tk scripting language developed by John K. Ousterhout. Detailed information on the AIM scripting language is beyond the scope of this manual. Information about Tcl/Tk is available in the book *Practical Programming in Tcl and Tk*, second edition by Brent B. Welch. Information about AIM is available in the AIM Reference documentation.

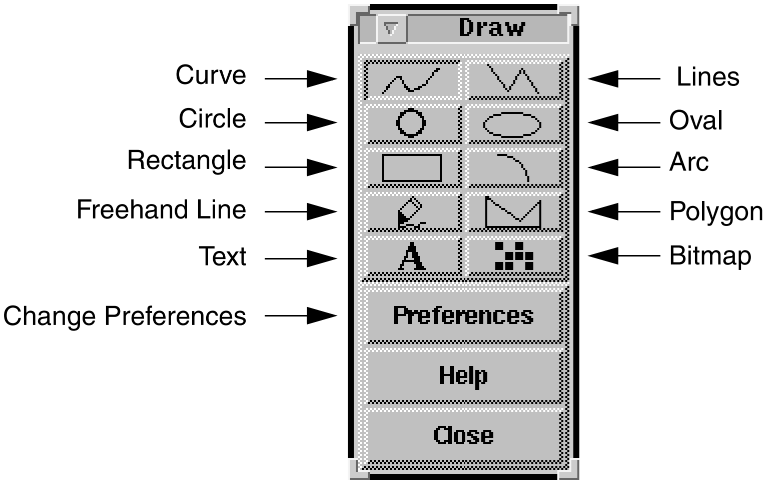
3

Using the Draw Tool

This chapter explains how to use the Draw Tool.

The Draw tool allows you to create both graphic objects and text in CosmosScope and any Saber graphics window. The Draw tool is also used to create and edit symbols in Saber Sketch.

The Draw tool provides Curve, Circle, Rectangle, Freehand Line, Lines (straight lines), Oval, Arc, and Polygon tools for drawing objects, the Text tool for inserting text, and the Bitmap tool for inserting objects.

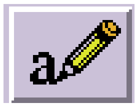


You can build any shape from these fundamental graphic objects and get exactly the look you want by specifying preferences like line thickness, fill patterns, arrow head styles, and colors.

In Saber, you can use style sheets to manage text, line and fill preferences for objects in a symbol or schematic. In the attribute dialog boxes in the Draw tool, the Format Style button opens the Manage Style Sheet form where you can change style sheets and styles. For more information about styles and style sheets, see the *Sketch User Guide* or the *Harness User Guide*.

Accessing the Draw Tool

The Draw tool icon is located in the Tool bar at the bottom of the main window.



To open or close the Draw tool, click the icon with the left mouse button. You can also choose **Tools > Draw Tool** or **Drawing Tool** from the main menu to open or close the Draw tool.

Creating Curves

The Draw Tool creates curves by drawing line segments between points specified by the user. To create a curve, click the Curve button in the Draw palette. The cursor changes to a cross shape.

To draw any irregularly curved object, draw a series of connected short segments to form the curves.

For example, to draw an S-shaped curve:

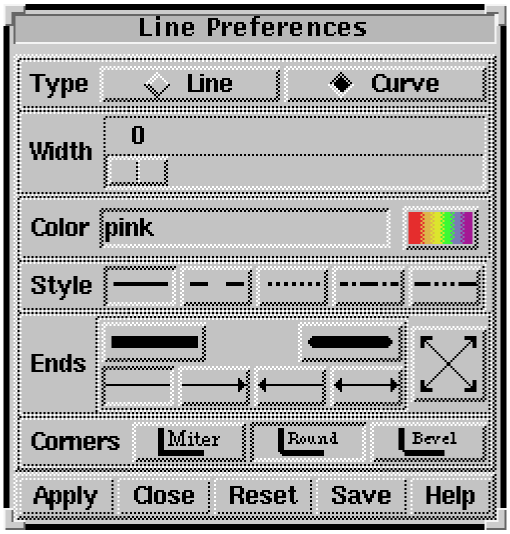
1. Hold down the left mouse button while you move the cursor to draw the first segment of the S-shape.
2. Click the left mouse button once, which ends the first segment and begins the second segment.
3. After drawing enough short segments to form the S-shape, double-click the left mouse button to complete the figure. You can smooth the resulting figure by changing the line attribute to Curve.

Changing Curve Preferences

Curve preferences can be changed in the Line Preferences dialog box.

To open the Line Preferences dialog box, use one of the following methods:

- Double-click on the object.
- Right-click the object, then choose Preferences from the Draw pop-up menu that appears.
- In the Draw palette, left click on the Curve, then click the **Preferences** button in the Draw palette.



Line Preferences

Type ◇ Line ◆ Curve

Width 0

Color pink

The Line Preferences dialog box elements are described as follows:

Type Field

Selecting the Line button shows every segment of the object as straight lines. Selecting the Curve button smooths the object and makes it look like a continuously drawn curved line.

Width Slider

Increases the width of a line when moving the slider to the right. Moving the slider to the left decreases the width of a line.

Color Field

Changes the color of the object.

You can change colors using one of the following methods:

- Type in the name of the color you want, and click the **Apply** button.
- Click the Rainbow button to bring up the Color Editor window.

Style Buttons

Changes the object’s lines to solid or dashed line styles.

Ends Buttons

Adds arrow heads, square ends or rounded ends to the line.

Corners Buttons	Changes corners to mitered, round, or beveled in an object composed of straight lines.
Reset Button	Restores the preferences dialog box fields to the values you previously specified.
Save Button	Saves the current preferences.
Close Button	Closes the dialog box.

Drawing Circles

To open the Circle tool, left-click the Circle icon in the Draw palette. The cursor changes to a cross shape.

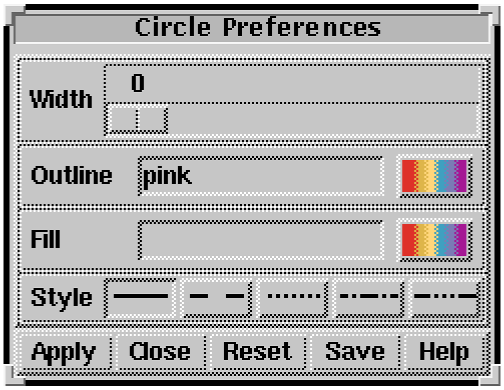
To draw a circle, hold down the left mouse button and drag the cursor to form the circle. Then, release the mouse button when the circle is the desired size. You can change Circle Preferences in the Circle Preferences dialog box.

Changing Circle Preferences

Circle preferences can be changed with the Circle Preferences dialog box.

You can open the Circle Preferences dialog box using one of the following methods:

- Double-click on the circle.
- Right-click on the circle, then choose **Preferences** from the pop-up menu that appears.
- Left-click the Circle button in the Draw palette, then click the **Preferences**.



Width 0

Outline pink

The following elements in the Circle Preferences dialog box are available:

Width Slider
Increases the width of the line. Move the slider to the right to increase width.

Outline Field
Changes the color of the outline of the object.

Fill Field
Changes the color of the interior of the object.

You can change colors using one of the following methods:

- Type in the name of the color you want, and click the **Apply** button.
- Click the Rainbow button to bring up the Color Editor window.

Style Buttons
Changes the line style to solid or dashed.

Reset Button
Restores the preferences dialog box fields to the values you previously specified.

Save Button
Saves the current preferences.

Close Button
Closes the dialog box.

Drawing Rectangles

To open the Rectangle tool, click the Rectangle icon in the Draw palette. The cursor changes to a cross shape.

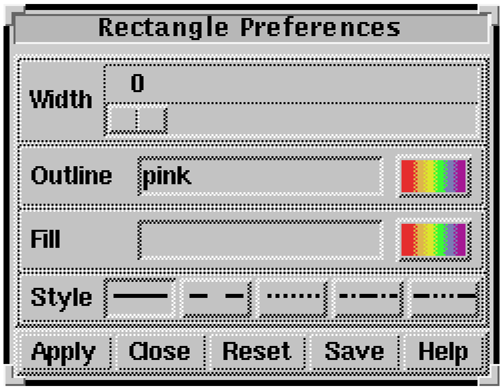
To draw a rectangle, drag the cursor to form the rectangle and Release the cursor when the rectangle is the desired size. Rectangle preferences can be changed in the Rectangle Preferences dialog box.

Changing Rectangle Preferences

You can change rectangle preferences in the Rectangle Preferences dialog box.

Use one of the following methods to open the Rectangle Preferences dialog box:

- Double click on the object.
- Right-click the object, then choose **Preferences** from the pop-up menu that appears.
- In the Draw palette, left-click on the Rectangle button, then click the **Preferences** button.



Rectangle Preferences

Width: 0

Outline: pink

Fill:

The following Rectangle Preferences elements are available:

Width Slider — Increases the width of a line by moving the slider to the right. Moving the slider to the left decreases the width of a line.

Outline Field — Changes the color of the outline of the object.

Fill Field

Changes the color of the interior of the object.

You can change colors in one of two ways:

- Type in the name of the color you want, and click on the Apply button.
- Click the Rainbow button to open the Color Editor window.

Style Buttons

Changes the object’s lines to solid or dashed line styles.

Reset Button

Restores the preferences dialog box fields to the values you specified previously.

Save Button

Saves the current preferences.

Close Button

Closes the dialog box.

Drawing Freehand Lines

To open the Freehand Line tool, click the Pencil icon within the Draw palette. The cursor changes to a cross shape.

To draw a freehand line:

1. Click the left mouse button to begin freehand drawing.
2. Move the cursor in any direction. A line follows the cursor.
3. Click the left mouse button again to stop drawing the line at the length you desire.

You can change freehand line preferences in the Line Preferences dialog box. See Changing Line Preferences on page 18 for more information.

Drawing Straight Lines

The Lines tool can be used to draw single straight lines or a series of connected straight lines.

To open the straight line tool, click the Lines icon in the Draw palette. The cursor changes to a cross shape.

To draw a single straight line:

1. Move the cursor to the starting point for the line and click the left mouse button.

2. Move the cursor to establish the desired length and direction of the line.
3. Double-click the left mouse button to stop drawing the line.

To draw several connected straight lines:

1. Click the left mouse button at the location where you want to begin drawing the line. click the mouse button once to end the first segment. Then move the mouse in a new direction for the second segment.
2. Move the first segment until it is correct,
3. Add as many segments as you need. Then double click to end the segmented line.

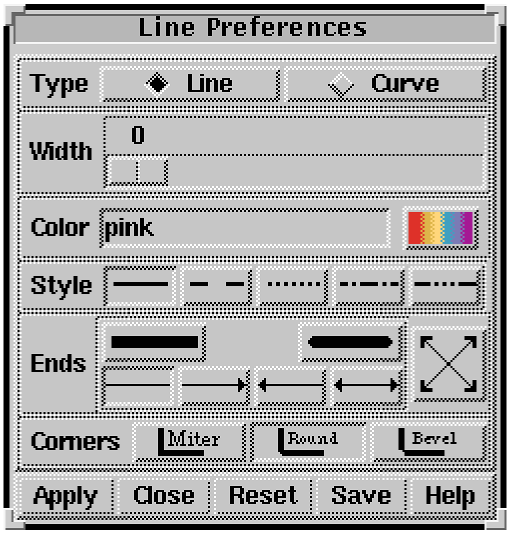
Straight line preferences can be changed with the Line Preferences dialog box.

Changing Line Preferences

Line preferences can be changed in the Line Preferences dialog box.

To open the Line Preferences dialog box, use one of the following methods:

- Double-click on the object.
- Right-click the object, then choose Preferences from the Draw pop-up menu that appears.
- In the Draw palette, left click on the line, then click the **Preferences** button in the Draw palette.



Line Preferences

Type ◆ Line ◇ Curve

Width 0

Color pink

The following Line Preferences dialog box elements are available:

Type Field
Selecting the Line button shows every segment of the object as straight lines. Selecting the Curve button smooths the object and makes it look like a continuously drawn curved line.

Width Slider
Increases the width of a line when moving the slider to the right. Moving the slider to the left decreases the width of a line.

Color Field
Changes the color of the object.

You can change colors using one of the following methods:

- Type in the name of the color you want, and click the **Apply** button.
- Click the Rainbow button to bring up the Color Editor window.

Style Buttons
Changes the object’s lines to solid or dashed line styles.

Ends Buttons
Adds arrow heads, square ends or rounded ends to the line.

Corners Buttons
Changes corners to mitered, round, or beveled in an object composed of straight lines.

Reset Button
Restores the preferences dialog box fields to the values you previously specified.

Save Button
Saves the current preferences.

Close Button
Closes the dialog box.

Drawing Ovals

To open the oval tool, left-click the Oval icon in the Draw palette. The cursor changes to a cross shape.

To draw an oval:

1. Click and hold the left mouse button.
2. Drag the cursor until the oval is the size and shape you desire.
3. Release the mouse button to lock the size and shape of the oval.

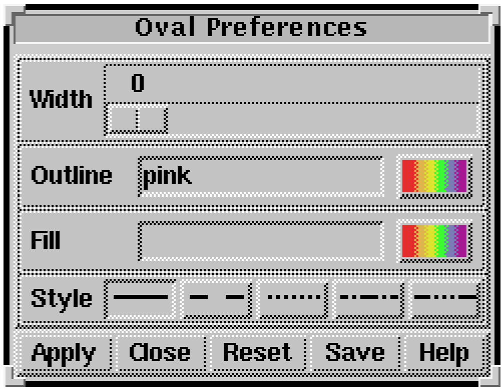
Oval preferences can be changed with the Oval Preferences dialog box.

Changing Oval Preferences

Oval preferences can be changed with the Oval Preferences dialog box.

You can use one of the following three methods to open the Oval Preferences dialog box:

- Double click on the object.
- Right-click on the object, and choose **Preferences** from the pop-up menu that appears.
- In the Draw palette, left-click on the Oval button, then click the **Preferences** button.



The following Oval Preferences dialog box elements are available:

Width Slider — Increases the width of a line when you move the slider to the right. Moving the slider to the left decreases the width of a line.

Outline Field — Changes the color of the outline of the object.

Fill Field — Changes the color of the interior of the object.

You can change the color of an oval using one of the following methods:

- Type in the name of the color you want, and click the **Apply** button.
- Click the Rainbow button to open the Color Editor window.

Style Buttons — Changes the line style of an oval to solid or dashed.

Reset Button — Restores the preferences dialog box fields to the values you previously specified.

Save Button	Saves the current preferences.
Close Button	Closes the dialog box.

Drawing Arcs

To open the Arc tool, click the Arc icon in the Draw palette. The cursor changes to a cross shape.

To draw an arc:

1. Click and hold the left mouse button.
2. Drag the cursor until the arc is seen along the circumference of the resulting oval.
3. Release the button to lock the size of the arc.

To adjust the shape of an arc:

1. Left-click on the arc. Four square handles appear around the arc once it is selected.
2. Left-click, hold, and then drag any one of the four square handles to alter the shape of the oval.
3. Release the mouse button when the arc is readjusted to the desired size.

To rotate an arc, move the arc around the circumference of the oval by grabbing any point on the arc between the handles on its ends (without touching the handles) and dragging. The length of the arc remains constant as its position changes.

To change the length of an arc, drag one of the handles at the end of the arc.

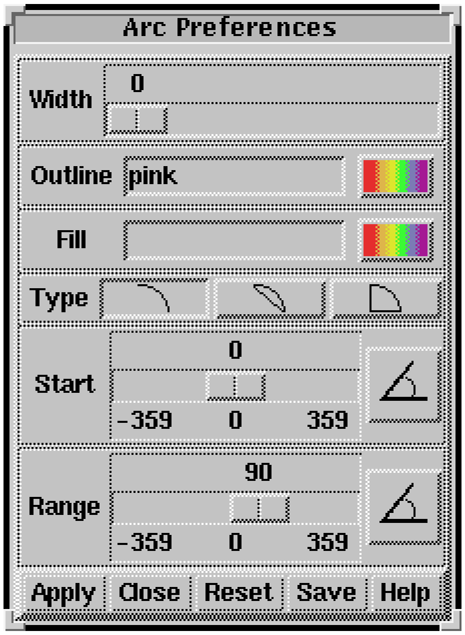
You can change the arc preferences with the Arc Preferences dialog box. See Changing Arc Preferences for more information.

Changing Arc Preferences

Arc preferences can be changed with the Arc Preferences dialog box.

You can open the Arc Preferences dialog box using on of the following methods:

- Double-click on the object.
- Right-click on the object to open the Draw popup menu, then left-click on the Preferences item.
- Left-click on the Arc button in the Draw palette, and then click the Preferences button.



Arc Preferences

Width: 0

Outline: pink

Fill:

Start: 0 (-359 0 359)

Range: 90 (-359 0 359)

The following Arc Preferences elements are available:

Width Slider — Increases the width of the line when you move the slider to the right. Moving the slider to the left decreases the width.

Outline Field — Changes the color of the outline of the object.

Fill Field — Changes the color of the interior of the object.

You can change colors using one of the following two methods:

- Type in the name of the color you want, and click on the Apply button.
- Click the Rainbow button to bring up the Color Editor window.

Type Buttons

Determine how the ends of an arc are closed and where the color fill occurs. The options are arc, chord, or pie slice as follows:

- Arc button (left most): no fill. Shows only the arc.
- Chord button (middle)—draws a chord line and fills between the chord and the arc.
- Pie Slice button (right most)—draws radii to the chord points and fills the area included by the radii and the arc.

Start Field

Specifies the beginning of the angular range occupied by the arc. Specify degrees counter-clockwise from the three o'clock position. Degrees can be negative values. The radii are visible when the pie slice Type button is selected.

A start value of zero draws a horizontal radius from the center of the figure, extending to the right as far as the circumference. Increasing positive values of Start swing this radius counter-clockwise.

You can change the Start value in one of two ways:

- Drag the slider.
- Click the Angle button at the right end of the field to open the Enter Start Angle dialog box. Enter a precise value in the box. Then click Apply.

Range Field

Specifies the size of the angular range occupied by the arc. Specify degrees counter-clockwise from the starting angle given by the Start field. Degrees can be negative values.

You can change the Range value in one of three ways:

- Drag the slider.
- Click the Angle button at the right end of the Range field to open the dialog box.
- Enter a precise value in the box. Then click Apply.

Reset Button

Restores the preferences dialog box field values to the previously- saved values.

Save Button

Saves the current preferences.

Close Button

Closes the dialog box.

Creating Polygons

To open the Polygon tool, click the Polygon icon in the Draw palette. The cursor changes to a cross shape.

To draw a polygon:

1. Click the left mouse button to begin.
2. Drag the cursor in any direction, and then click the left mouse button once.
3. Drag the cursor in another direction as a three-sided figure forms.
4. Click once again, and drag the cursor in another direction to form a four-sided closed figure.
5. Continue adding sides until you are done. Then double click the left mouse button to end the closed figure.

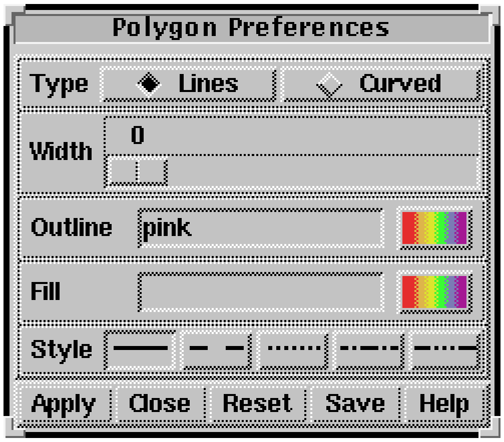
Polygon preferences can be changed with the Polygon Preferences dialog box. See Changing Polygon Preferences for more information.

Changing Polygon Preferences

You can change the polygon preferences in the Polygon Preferences dialog box.

To open the Polygon Preferences dialog box, use one of the following methods:

- Double-click on the object.
- Right-click the object to open the Draw popup menu, then left click on the Preferences item.
- Left-click the Polygon button in the Draw palette, and then click the **Preferences** button.



Type ◆ Lines ◇ Curved

Width 0

Outline pink

Fill

The following Polygon Preferences dialog box elements are available:

Type Field

Selecting the Line button shows every segment of the object as a straight line. Selecting the Curve button smooths the object and makes it look like a continuously drawn curved line.

Width Slider

Moving the slider to the right increases the width of a line. Moving the slider to the left decreases the width of a line.

Outline Field

Changes the color of the outline of the object.

Fill Field

Changes the color of the interior of the object.

There are two ways to change colors:

- Type in the name of the color you want, and click on the **Apply** button.
- Click the Rainbow button to bring up the Color Editor window.

Style Buttons

Changes the line style of an object to solid or dashed.

Reset Button

Restores the preferences dialog box field values to the previously- saved values.

Save Button	Saves the current preferences.
Close Button	Closes the dialog box.

Inserting Text

To open the Text tool, click the Text button within the Draw palette. The cursor changes to a cross shape.

To enter text:

1. Move the cursor to the point where the text is to begin.
2. Click the left mouse button. The cursor changes to a vertical, blinking line.
3. Begin typing your text. You can enter carriage returns to continue the text on a new line.
4. Exit text entry mode by clicking the left mouse button outside of the text area.

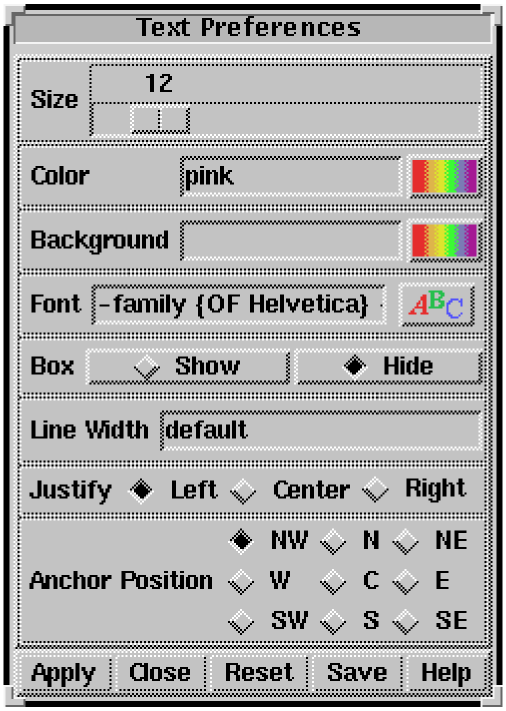
Text preferences can be changed with the Text Preferences dialog box. See Changing Text Preferences for more information.

Changing Text Preferences

You can change text preferences in the Text Preferences dialog box.

To open the Text Preferences dialog box, use one of the following methods:

- Double-click on the object.
- Right-click on the object to open the Draw popup menu, then left-click the Preferences item.
- Left-click on the Text button in the Draw palette, then click the Preferences button.



Text Preferences

Size: 12

Color: pink

Background:

Font: -family {OF Helvetica}

Box: ○ Show ● Hide

Line Width: default

Justify: ● Left ○ Center ○ Right

Anchor Position: ● NW ○ N ○ NE ○ W ○ C ○ E ○ SW ○ S ○ SE

The following Text Preferences dialog box elements are available:

Size Slider — Moving the slider to the right increases the point size of the text. Moving the slider to the left decreases the point size of the text.

Color Field — Changes the color of the text.

To change colors, use one of the following methods:

- Type in the name of the color you want, and click the **Apply** button.
- Click the Rainbow button to open the Color Editor window.

Font Field — Displays the current font set up. The **ABC Fonts** button opens the Font Selection dialog box.

Box Text Field	The **Show** button causes text to be enclosed in a box. The **Hide** button removes an existing box.
Line Width Field	Sets the maximum width, in points, allowed for a line of text. To change the line width, enter a value in this field and click the **Apply** button.
Justify Buttons	Select Left, Center, or Right justification.
Anchor Position Buttons	Position the location of the text anchor. The anchor is a small square symbol that indicates the starting point of the text. For example, placing the anchor in the NW position indicates that text appears to the right and below the anchor. As another example, placing the anchor in the E position indicates that text appears directly to the left of the anchor position.
Reset Button	Restores the preferences dialog box field values to the previously- saved values.
Save Button	Saves the current preferences.
Close Button	Closes the dialog box.

Creating Bitmaps

Note:

In CosmosScope, you can retrieve a bitmap file using the following method (this feature is not implemented in Saber Sketch):

To open the Bitmap tool, click the Bitmap button on the Draw palette. The Bitmaps window appears.

To open and place a bitmap:

1. Select the bitmap file you want to work with by clicking one of the files listed under the File Name heading. A preview version of the file appears to the right of the list.
2. Click the **Open** button.
3. Move the cursor to the graph window, and left-click once. The bitmap appears in the window.

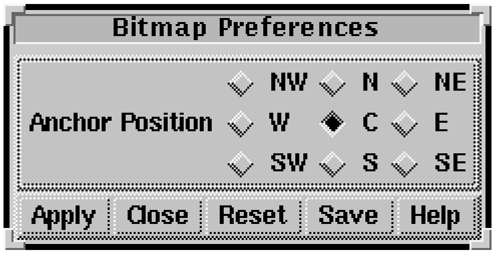
Bitmap preferences can be changed with the Bitmap Preferences dialog box. See Changing Bitmap Preferences for more information.

Changing Bitmap Preferences

You can change bitmap preferences in the Bitmap Preferences dialog box.

To open the Bitmap Preferences dialog box, use one of the following methods:

- Double-click on the object.
- Right-click on the bitmap to open the Draw popup menu, then left-click on the Preferences item.



Bitmap Preferences

Anchor Position: ◇ NW ◇ N ◇ NE / ◇ W ◆ C ◇ E / ◇ SW ◇ S ◇ SE

The following Bitmap Preferences dialog box elements are available:

Anchor Position Buttons — Position the location of the bitmap anchor. The anchor is a small square symbol that indicates the starting point of the bitmap. For example, placing the anchor in the NW position indicates that bitmap appears to the right and below the anchor. As another example, placing the anchor in the E position indicates that bitmap appears directly to the left of the anchor position.

Close Button — Closes the dialog box.

Reset Button — Restores the preferences dialog box fields to the values they had before you entered changes.

Save Button — Saves the current preferences.

Editing Colors

To open the Color Editor, click the Rainbow button in any Color Field.

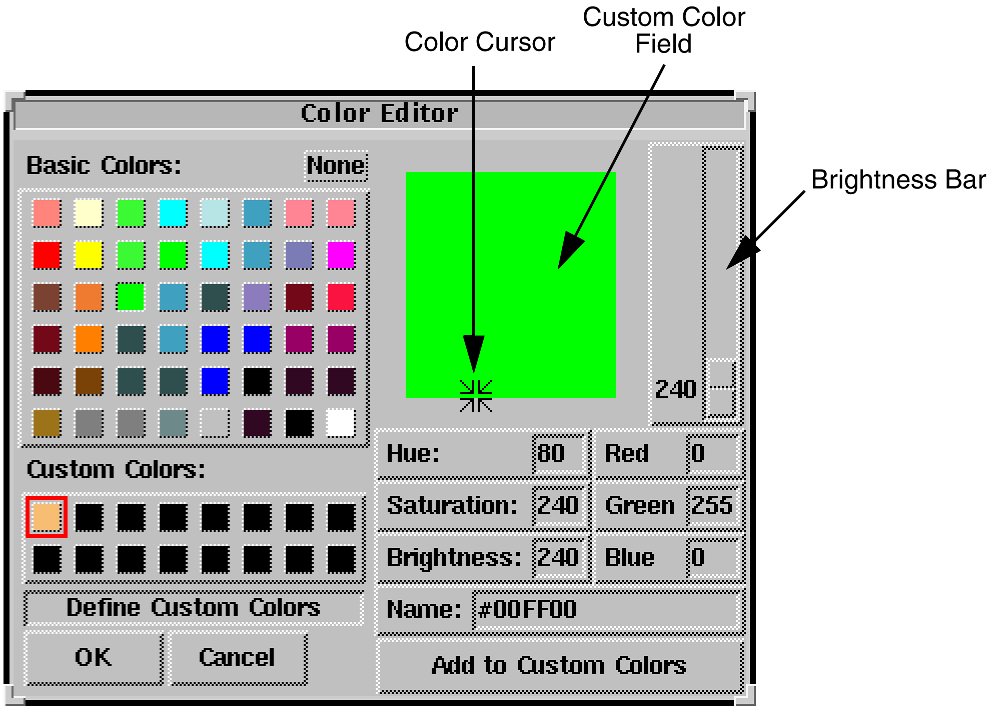
The number of available colors is dependent on your system. If you are not sure about the number of colors available, names, hexadecimal codes, or location, ask your system administrator.

To use any of the basic or existing custom colors, click on the desired color square, and then click the **OK** button. This process selects the color.

To define a new custom color, click the **Define Custom Colors** button. The Custom Color Editor dialog box appears.

Creating Custom Colors

To define a custom color you can use the Color Editor dialog box.



240

Hue: 80
Red 0
Saturation: 240
Green 255
Brightness: 240
Blue 0
Name: #00FF00

To create custom colors, use one of the following methods:

- Use the color cursor and brightness bar.
- Enter numeric values in the Hue, Saturation, Brightness, Red, Green, and Blue fields.
- Enter the name of a color in the Name field.

Using the Color Cursor and Brightness Bar

When you create custom colors, the color is displayed in the custom color field. Moving the color cursor inside of the custom color field displays all of the various colors available for use.

To create a custom color:

1. Place the mouse cursor over the color cursor in the custom color field.
2. Click and hold the left mouse button, and move the color cursor.
3. Release the mouse button when the desired color appears.

The sliding bar to the right of the custom color field controls the brightness of a custom color.

To change the brightness of a custom color:

1. Place the mouse cursor over the brightness bar, click and hold the left mouse button, and move the bar up or down.
2. Move the bar up to make the custom color darker, or move the bar down to make the custom color lighter.
3. Click the **Add to Custom Colors** button with the left mouse button to add the new color to the Custom Colors field.

Entering Color Values

You can directly enter numeric color values in the Hue, Saturation, Brightness, Red, Green, and Blue fields.

Generally, values from 0 to 255 are acceptable. If a number falls outside of acceptable limits, the highest or lowest acceptable value is automatically substituted.

To enter numeric color values:

1. Enter the numeric values.
2. Press the **Return** key.
3. Left-click the **Add to Custom Colors** button to add the new color to the Custom Colors field.

Recalling Custom Colors by Name or Hexidecimal Code

The Name field displays the hexadecimal code of the current custom color. The Name field can also be used to call up a color by its name or to call up a color by hexadecimal code.

To recall a color by name:

1. Clear the Name field with the Delete key, the Back Space key, or by pressing the Control-u keys simultaneously.
2. Type in the name of the color (such as green, or cyan), and press the Return key. The color appears in the custom color field.
3. Click the **Add to Custom Colors** button with the left mouse button to add the new color to the Custom Colors field.

To recall a color by hexadecimal code:

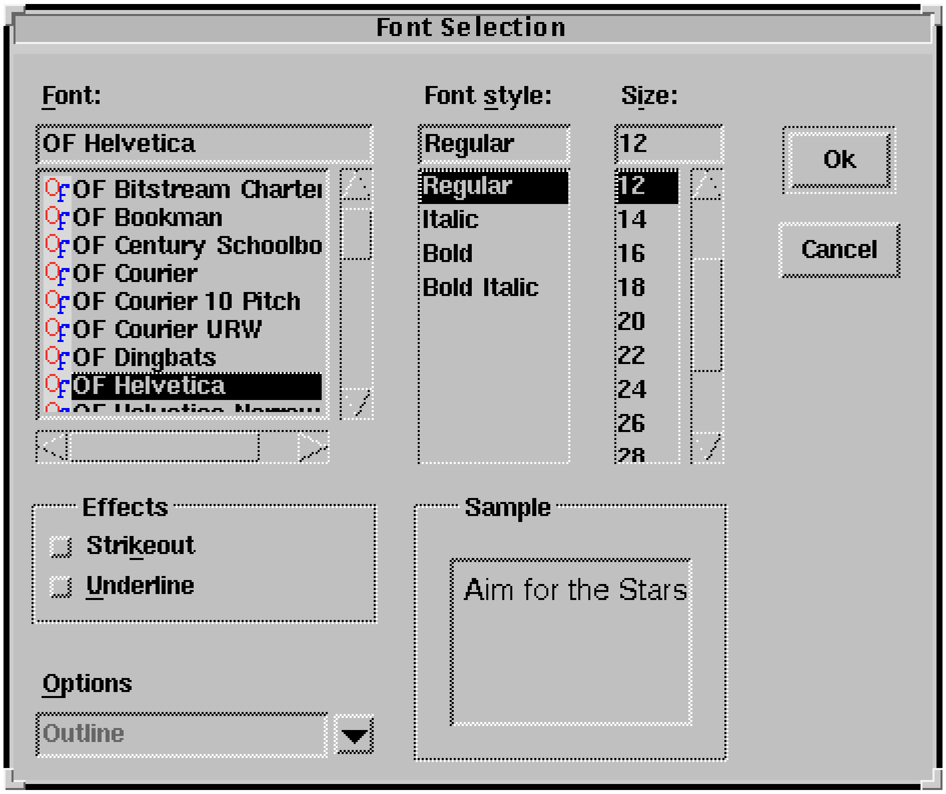
1. Clear the Name field with the **Delete** or **Backspace** key. then
2. Type in hexadecimal code, and press the **Return** key. The color appears in the custom color field.
3. Click the **Add to Custom Colors** button with the left mouse button to add the new color to the Custom Colors field.

Note:

> The hexadecimal code for a color depends on your system. For example, a system using 256 colors will have a six digit code. The first two digits represent the red value of the color, the next two digits represent the green value, and the last two digits represent the blue value. In this system, entering #00ff00 in the Name field displays a pure green color in the custom color field.

Selecting Fonts

To change the existing font, highlight your selections in the Font Selection dialog box and click the **OK** button. The following figure shows an example of the highlighted selections:



Font Selection

Font: OF Helvetica

Font style: Regular

Size: 12

Sample: Aim for the Stars

The following Font Selection dialog box elements are available:

Font Scrollable List — Displays all of the available fonts on your system. The name of the currently enabled font is displayed at the top.

Font Style Scrollable List — Displays all of the available styles for a selected font. The name of the currently enabled font style is displayed at the top.

Size Scrollable List — Displays all of the available font sizes (in points). The currently enabled font size is displayed at the top.

Strikeout Button — Puts a line through the middle of the current font.

Underline Button — Adds underlining to the current font.

Options checkbox List — Displays the font option types that are available.

Sample Text Field — Allows you to see the type of font, the font style and what the font size looks like before applying the font in your application.

4

Using the Parts Gallery

The Parts Gallery allows you to locate and place models and components (parts) in your schematics. The schematic can be created in Saber Sketch, Saber Harness or a Frameway.

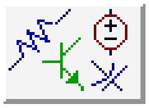
Parts are supplied in Saber libraries or furnished by you. You can easily add your own parts and categories to Parts Gallery and delete them. Entries for Saber libraries are protected and cannot be deleted.

As you work with a schematic, you can find specific parts by navigating the parts category tree or by issuing a text search that will find and display all parts with names that match the full or partial search string you supplied.

Most parts are written in the MAST and VHDL-AMS modeling languages and stored as template files. Simulator selectors allow you to see all parts or just those parts that are supported by a specific simulator, either Saber (MAST) or SaberHDL (MAST and VHDL-AMS).

Introduction to the Parts Gallery

The Parts Gallery icon is located in the Tool bar at the bottom of the work surface.



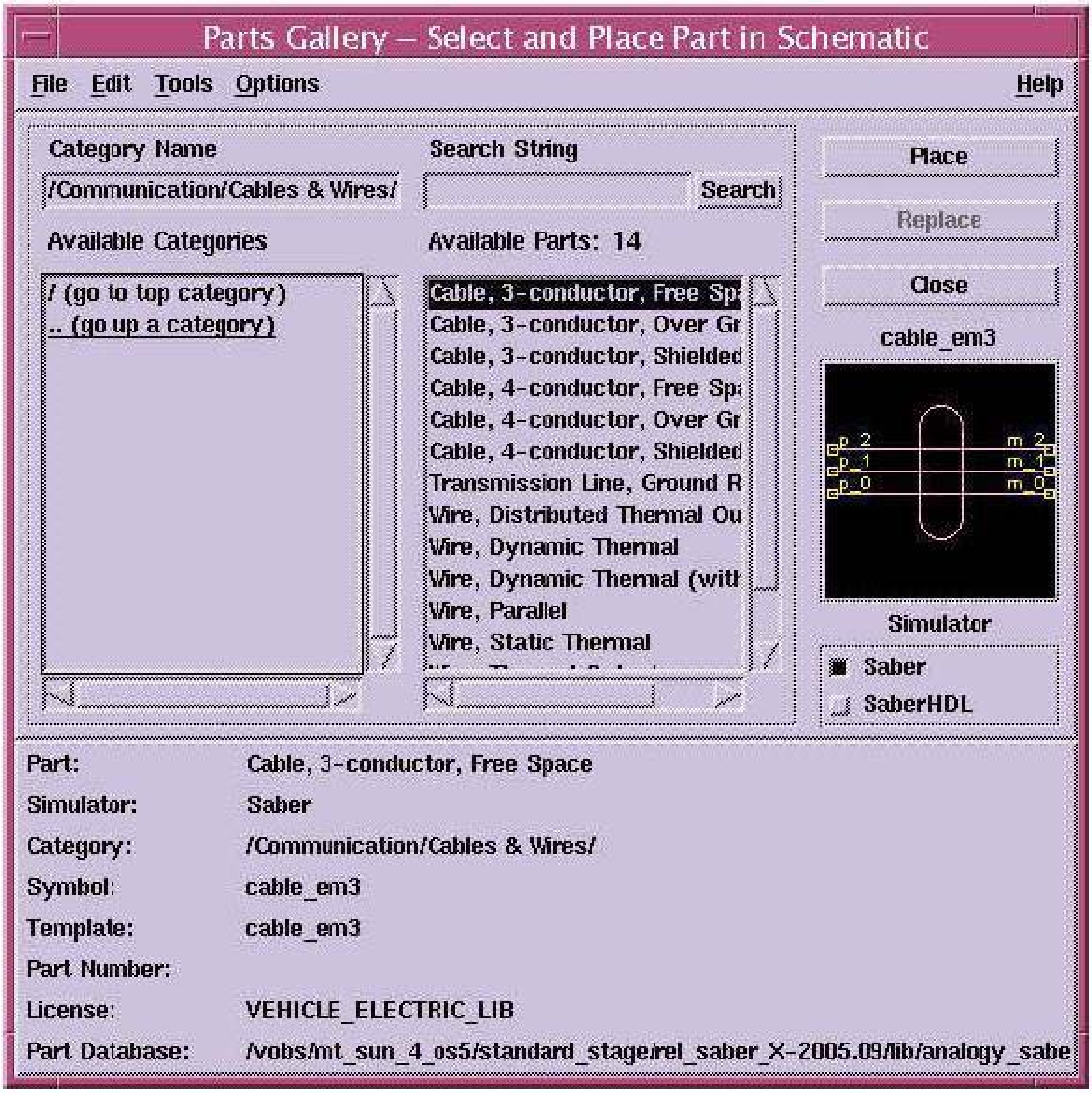
To open or close the Parts Gallery, click the icon with the left mouse button.

This is the same regardless of which user interface is set as the default.

Two User Interfaces

There are two user interfaces available to you for operating the Parts Gallery: the original, “classic” user interface and the “new” user interface.

Figure 3 Classic User Interface



Category Name
/Communication/Cables & Wires/

Search String

Available Categories
/ (go to top category)
.. (go up a category)

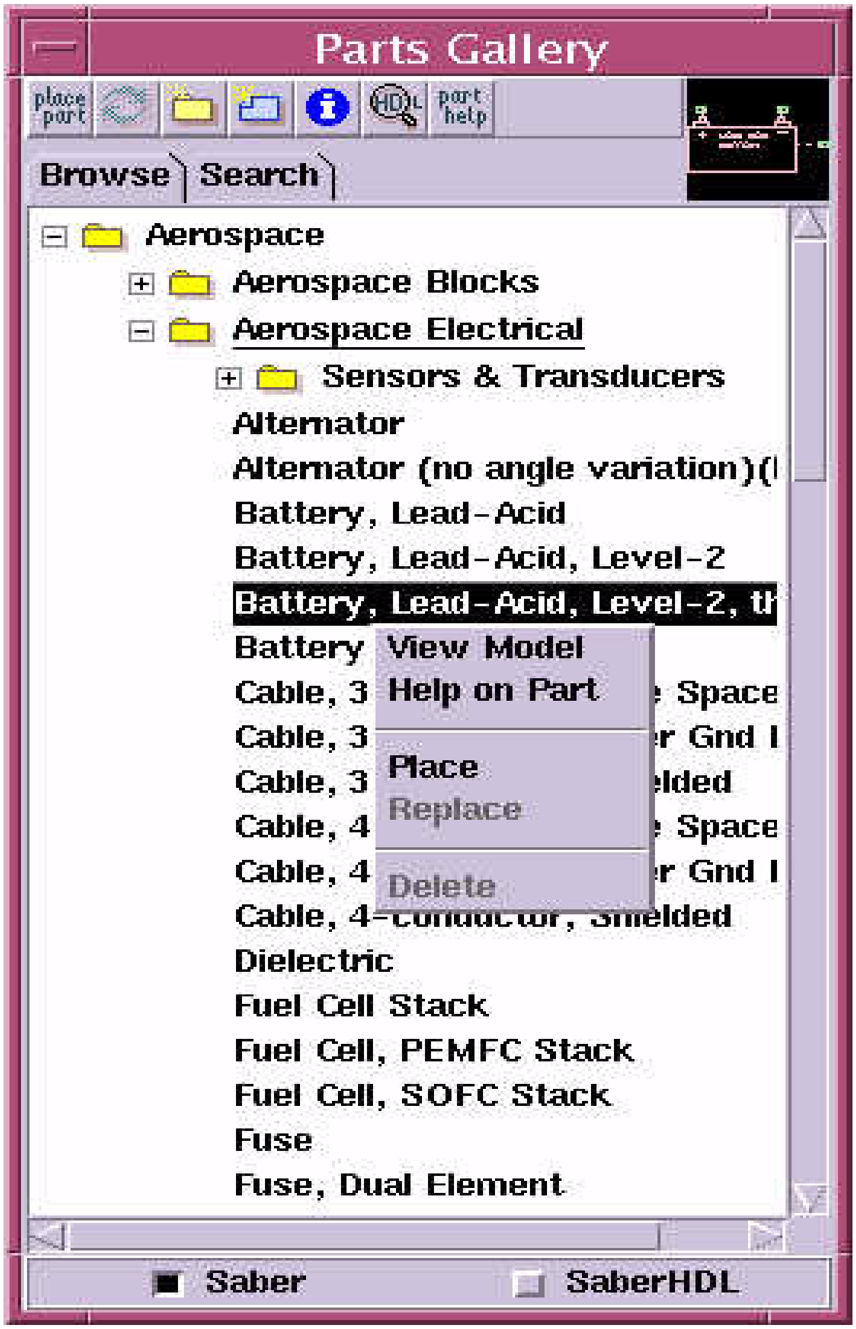
Available Parts: 14
Cable, 3-conductor, Free Sp
Cable, 3-conductor, Over Gr
Cable, 3-conductor, Shielded
Cable, 4-conductor, Free Spa
Cable, 4-conductor, Over Gr
Cable, 4-conductor, Shielded
Transmission Line, Ground R
Wire, Distributed Thermal Ou
Wire, Dynamic Thermal
Wire, Dynamic Thermal (with
Wire, Parallel
Wire, Static Thermal

Part:	Cable, 3-conductor, Free Space
Simulator:	Saber
Category:	/Communication/Cables & Wires/
Symbol:	cable_em3
Template:	cable_em3
Part Number:	
License:	VEHICLE_ELECTRIC_LIB
Part Database:	/vobs/mt_sun_4_os5/standard_stage/rel_saber_X-2005.09/lib/analogy_sabe

The classic user interface shows the parts and subcategories available under one category (the active one). Tree traversal is therefore necessary to select parts under different categories. A tree browser is present in the new user interface so you can navigate the hierarchy of categories as well as expand or collapse category contents as needed.

The new user interface takes up less screen space than the classic user interface, and the Parts Gallery window is dockable within the application window. These features help you better manage windows in your workspace.

Figure 4 Figure 1-2. New User Interface

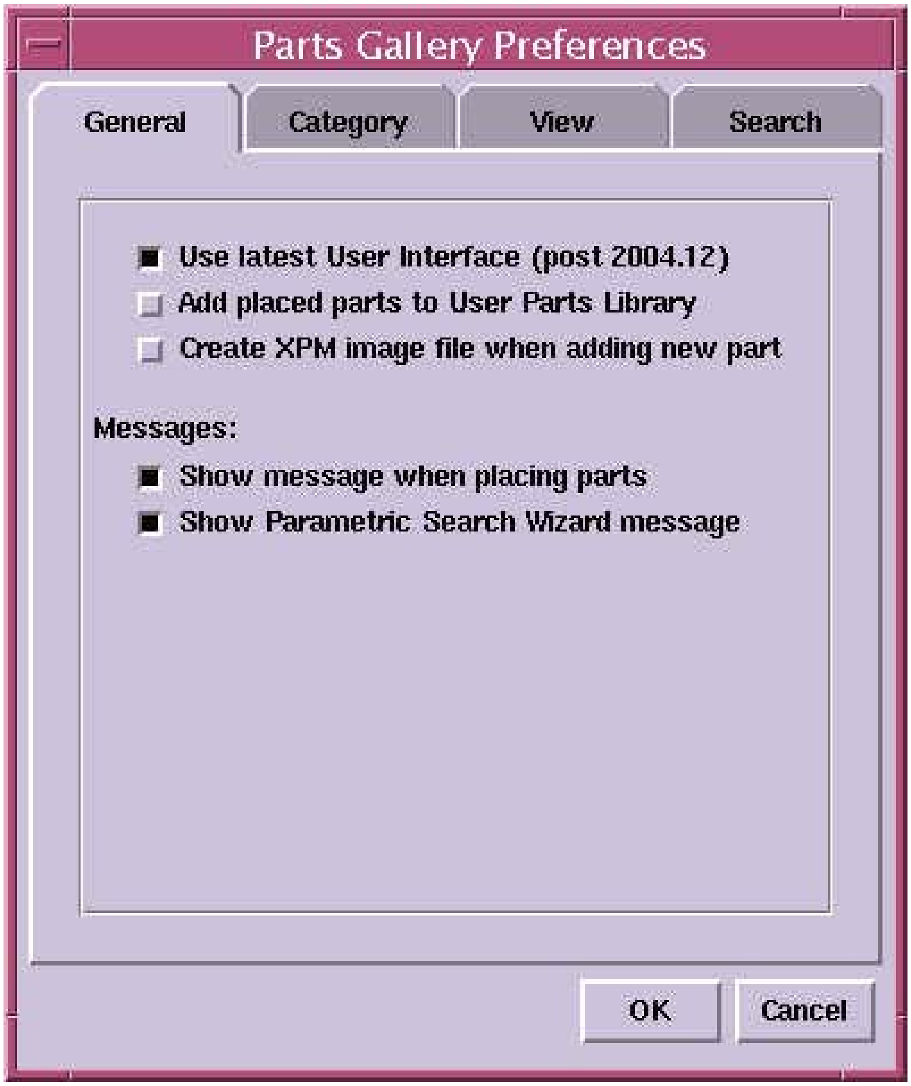


Setting the User Interface Preference

The new user interface is the default for Parts Gallery but you can it. Note that the classic and new versions of Parts Gallery can never be invoked at the same time.

The classic user interface or new user interface can be selected as the default for Parts Gallery in the General tab of the Preferences dialog form. This form can be invoked from the Saber Sketch Edit menu or from the Parts Gallery itself. In the classic Parts Gallery, it is invoked from the Options menu, and in the new Parts Gallery, from the right-click menu of the hierarchy browser (place the cursor in the white background area to select this menu). Selecting Use latest User Interface brings up the new user interface.

Figure 5 Figure 1-3: Parts Gallery Preferences form.



Parts Gallery Preferences

☒ Use latest User Interface (post 2004.12)

☐ Add placed parts to User Parts Library

☐ Create XPM image file when adding new part

Messages:

☒ Show message when placing parts

☒ Show Parametric Search Wizard message

Changing the user interface default with the Preferences dialog form has the following effects:

- The Parts Gallery user interface is rebuilt to reflect the selection you made.
- After you close the Parts Gallery and re-open it within the same session that the change was made, the Parts Gallery user interface stays consistent with the selection you made.
- The next time you invoke the application, the Parts Gallery user interface is consistent with the preference selection you made earlier; the user interface you selected is persistent and will continue to appear every time you open the Parts Gallery.

The Classic Parts Gallery User Interface

The Parts Gallery is your interface for finding, viewing and placing parts (models and components) in your design schematics. The Parts Gallery displays parts categories as well as part names, and graphical images for part symbols. Specific parts can be found by navigating the parts category tree or by issuing a text search that will find and display all parts with names that match the full or partial search string you supplied.

Saber libraries are represented in the Parts Gallery, and you can easily add and delete your own parts and part categories. Parts and categories for Saber libraries are protected and cannot be deleted.

When you have selected a part, a brief description of the part is shown in the window. Additional information can be viewed by selecting the pull-down menu item Help > Help On Part

Searching for Parts

The Parts Gallery provides access to thousands of parts. To find specific parts, you can do one of the following:

- Browse through the Available Categories and Available Parts scrollable lists.
- Use the Search String field.
- Use the Tools > Parametric Search menu item to open the Parametric Search Wizard. (Saber Sketch)
- Use the model documentation.

Search options are available through the Options pull-down menu item. The part name, category, symbol name, template name, and database location appear below the Available Categories, and Available Parts scrollable lists.

The Available Parts header displays the number of parts available in the Available Parts list.

Parts Gallery Database Files

When the Parts Gallery starts, it looks for parts databases in this order:

1. Model Parts Database
2. Site Parts Database (if it exists)
3. User Parts Database

To determine which database a part belongs to, select the Options > Preferences > View > Database menu item. The path to the database will be displayed at the bottom of the Parts Gallery window.

Model Parts Database The model database consists of all the parts and categories supplied with the simulation software. These parts and categories are protected, and cannot be deleted through the Edit menu.

Site Parts Database The site database consists of user parts and categories. These parts and categories are protected and cannot be deleted through the Edit menu. The site database file must be created manually.

The file called aimpart.site resides in a directory of your choice.

User Parts Database The user database consists of parts and categories you add through the Edit menu. These parts can be associated with a pre-defined category, or a category of your own. These parts are not protected and can be deleted.

If you wish to share the contents of your part database with other users on your system, copy the .aimpart_user file to a file called aimpart.site. If other users wish to access these parts they can put the location of aimpart.site in their AI_SITE_PATH environment variable.

The file defining the user database is called .aimpart_user and is located in your home directory.

Database Read Errors When the Parts Gallery starts, it checks the current user database against the .aimpart_user file. Two error messages are possible when the file does not match the database.

1. If the .aimpart_user file contains a reference to a part which is not in the database, an error message will appear suggesting that you update the database. If you click Yes, the Parts Gallery Database Update Wizard will take you through the update procedure.
2. If the .aimpart_user file and the database categories do not match, an error message will appear indicating that some parts or categories may be missing. You can clear this error in either of two ways:
   - If you have no custom parts in .aimpart_user, you may delete the .aimpart_user file.
   - If you have custom parts in .aimpart_user and do not want to remove the file, you will have to verify the file line-by-line.

Selecting and Placing Parts

Once you have located the part you are interested in, you must select it to place it in your schematic or access more documentation about it through the pull-down menu item: Help > Help On Part

To select and place a specific part perform the following steps:

1. Highlight the part in the Available Parts list, or the Parametric Search Wizard Search Results dialog box, by single clicking on the part name.
2. Put the selected part into the center of the schematic window by single clicking on the Place button or double clicking on the part name in the list.

   Put the part at the location of the mouse cursor by clicking on the window with the middle mouse button.

To replace several parts in the schematic with one from the Parts Gallery perform the following steps:

1. Select the first part in the schematic you wish to replace.
2. While holding down the Shift key, select any other parts in the schematic you wish to replace.
3. Select the part in the Parts Gallery that will replace the selected schematic parts.

4. Press the Replace button. All the parts that you had selected in the schematic have now been replaced with the part you selected in the Parts gallery.

Menus

File menu

Allows you to update the parts database and close your Parts Gallery session. For more information, see File Menu on page 44.

Edit menu

Allows you to create and delete parts and categories. For more information, see Edit Menu on page 45.

Tools menu

Allows you to view templates, open online help for a part, perform a parametric search for components with the Parametric Search Wizard, Invoke characterization tools, convert SPICE models to MAST with Nspitos, use the Scanned Data utility to import a scanned image, and access the Units Converter tool.

Note that some of these features are only available with Saber Sketch.

For more information, see Tools Menu on page 47.

Options menu

Allows you to display additional information about parts. Active display options are displayed in a field at the bottom of the Parts Gallery. For more information, see Options Menu on page 54.

Help menu

Displays information about the Parts Gallery and help on a specific, selected part (Help On Part). For more information, see Help Menu on page 56.

File Menu

Allows you to update the parts database and close your Parts Gallery session.

Update Database

Allows you to convert renamed part references to their new names and delete obsolete part name references so that databases can be loaded without error.

Close

Allows you to close the Parts Gallery.

Edit Menu

The parts and categories added through this menu are saved in the user database. This allows your parts and categories to be accessed by the Parts Gallery Search String field. These parts and categories are not protected and can be deleted.

You can add a new part to an existing category, create a reference in another category for a part, or delete a part or part reference from a category.

You can add a new category to the Available Categories scrollable list, or delete a category, with its related parts, from the Parts Gallery.

The Edit menu items are described as follows:

New Part

Allows you to add a new part to an existing category using the Create New Part dialog box.

Create Part Reference

Allows you to reference a part from the Available Parts scrollable list to any category listed in the Available Categories scrollable list using the Create Part Reference dialog box.

Delete Part

Allows you to delete a part or part reference from a category.

New Category

Allows you to add a new category to the Available Categories scrollable list

Delete Category

Allows you to delete a category, with its related parts, from the Parts Gallery.

Create New Part Dialog Box

Allows you to enter the information necessary to add a part.

To search for categories

- Click on the Category Browse button to open the Select Category dialog box.

To search for symbols

- Click on the Symbol Browse button to open the Select Symbol dialog box.

The Create New Part dialog box fields are summarized as follows:

Description — Allows you to type a part description which can be displayed in the Available Parts scrollable list.

Category — Allows you to choose the category for the part.

Symbol — Allows you to choose the symbol associated with the part.

Property Name and Property Value — Allows you to specify part properties. The property names and values must be valid for the symbol referenced by the part. You cannot create new properties in these fields.

Component (Frameway for Mentor Graphics) — Allows you to specify a component directory. This is a required entry.

If the Symbol field is left blank it is assumed that there is a symbol in the component directory with the same name as the component.

Library (Frameways for Cadence, Viewlogic and Mentor Graphics) — Allows you to specify the name of the symbol library. This is a required entry.

In the Frameway for Cadence, if you are using Saber symbols (the library name is SaberLib) a pixel map will be associated with the new part. Otherwise the part will have no associated pixel map.

Create Part Reference Dialog Box (Saber Sketch)

Allows you to enter the information necessary to add a part.

To search for categories

- Click on the In This Category Browse button to open the Select Category for New Part dialog box.

To create the new part reference

1. Select a part in the Available Parts listbox in the Parts Gallery.
2. Open the Create Part Reference dialog box.
3. Select a Category in the In This Category field in the Select Category for New Part dialog box.

4. Press the Create button in the Create Part Reference dialog box.

To add the part to a category

- Click on the Create button.

To close the dialog box

- Click on the Close button.

For information on creating parts

- Click on the Help button.

To reference the part to a different category

- Click on the Create button.

To close the dialog box

- Click on the Close button.

The Create Part Reference dialog box fields are summarized as follows:

Reference This Part	A part selected in the Available Parts listbox in the Parts Gallery is placed in this field.
In This Category	A category selected in the Category Name field of the Select Category for New Part dialog box (invoked with the Browse... button) is placed in this field.

Tools Menu

The Tools menu items are summarized as follows:

View Template	Opens a template viewing window, allowing you to view the underlying template of a selected part. The template cannot be edited in this window. (Saber Sketch)
Help on Part	Opens up the Template Description for a selected part, in the appropriate library, in the SaberBook online documentation.
Parametric Search	Opens the Parametric Search Wizard, which allows you to search for components in the MAST Component Libraries. (Saber Sketch)

nspitos

The nspitos menu item starts the nspitos translator user interface, which allows you to translate the specified file (in a SPICE 2G.6, SPICE3F, PSPICE, or HSPICE input format) to a format usable by the Saber Simulator (MAST). (Saber Sketch)

Units Converter

Opens the Units Converter interface, which provides unit conversion for the characteristic quantities of mechanical, hydraulic, magnetic, thermal, control, photonic, and custom-defined systems. (Saber Sketch)

Table Look-Up Table

The Table Look-Up Modeling tool is an interface used to create models from measurement, data sheets, or other simulation software for finite element analysis.

Magnetic Component

The Magnetic Component Characterization tool provides interactive characterization for the non-linear magnetic core templates corenl (based on the Jiles-Atherton model) and corenl2 (based on the Preisach model). (Saber Sketch)

Battery Tool

The Battery Tool provides support to characterize the Saber behavioral lead-acid battery model, batt_pb_1. (Saber Sketch)

Li-ion Tool

The Li-ion Battery Tool provides support to characterize the Saber behavioral lithium-ion battery model, batt_li_1. (Saber Sketch)

Fuse Tool

The Fuse Characterization tool provides an easy interactive way to characterize the fuse model available in the Saber Simulator. The graphical interface is also useful in understanding the functionality of the model. (Saber Sketch)

Diode Tool

The Diode Tool provides support to characterize the Saber diode behavioral model. (Saber Sketch)

MOSFET Tool

The MOSFET Tool provides support to characterize the Saber MOSFET behavioral model. (Saber Sketch)

Thermal Tool

The Thermal Tool provides support to characterize the Saber thermal model. (Saber Sketch)

Load Profile Tool

The Load Profile Editor provides support to characterize the Saber load models. These load models are available in the Parts Gallery in the Automotive category. (Saber Sketch)

Drive Cycle

The Drive Cycle Editor provides support to characterize the Saber dr_cycle template. (Saber Sketch)

Scanned Data

The Scanned Data menu item starts the utility which allows import of scanned data into the Table Look-Up tool and the Magnetic Component tool.

SaberRT

SaberRT allows you to apply interactive stimuli to a Saber design and generate a C interface for hardware-in-the-loop or real-time applications. (Saber Sketch)

DCPM Motor

The Motor Tool provides support to characterize the Saber behavioral DC Permanent Magnet Motor model, dc_pm2. (Saber Sketch)

Template Viewing Window (Saber Sketch)

Allows you to view, but not edit, templates associated with a part. The Template Viewing window items are described as follows:

Close button

Closes the template viewing window.

Find button

Opens an AimSearch - Find dialog box.

Dependencies button

Opens the Dependencies dialog box to display a list of templates which were used to develop a selected part.

You can view the template contents by clicking on the Template button at the bottom of the Dependencies dialog box.

Font button

Opens the Font Selection dialog box to allow you to change the text font size and style.

AimSearch - Find Dialog Box The AimSearch - Find dialog box allows you to search for text strings or parts of text strings.

Find field	Where you type the text string you want to locate. A default text search starts at the beginning of the template. The field is not case sensitive, and the text string may be a fragment of a larger text string.
Consider Case	Narrows the search by looking for text that matches the case of the text string typed in the Find field.
Whole Word	Narrows the search by looking for the entire text string. If the text string is a fragment of another text string, it will not be considered.
Backwards	Searches backwards from the insertion cursor position.
Find Next button	Press this button to start a search.
Close button	Closes the dialog box.

Parametric Search Wizard (Saber Sketch)

The Parametric Search Wizard is a database tool that searches for components in the MAST Component Libraries based on parameters that you enter in the tab forms.

To search for a component:

1. Click on the Tools > Parametric Search pull-down menu item in the Parts Gallery.
2. Select a category of components from the Select Category dialog box.
3. Press the Next button when you have made your selection. The Specify Attributes tab forms will be displayed.
4. Enter your search parameters in the General Information tab, the Maximum Ratings tab, and/or the Performance Specifications tab.
5. Click on the Finish button. The Search Results dialog box will display the results of the search.

Select Category Dialog Box Allows you to narrow your search by selecting a category of components before entering parameters in the Specify Attributes tab forms.

To select a category:

1. Single click on a category in the category list.
2. Click on the Next button to open the Specify Attributes tab forms.

General Information Tab Allows you to enter general search parameters for the component category you have selected.

There are four fields that are always available for every category. Other fields will be available depending on the category you have selected.

The selections for a field are displayed by clicking on the downward pointing arrow to the right of a field. The General Information tab items are summarized as follows:

Display check box
This check box to the right of the downward-pointing arrow toggles the display of that fields information in the Search Results dialog box.

Model Name
The Model Name option field allows you to search for text strings Containing, Beginning With, or Equal To text typed in the Model Name field. Selecting Any searches for any model names, without limitation.

Configuration
Allows you to select Dual or Single.

Part Class
Allows you to select a specific device class.

Package
Allows you to select a specific device package type.

Manufacturer
Allows you to select a specific manufacturer.

The buttons for the General Information, Maximum Ratings and Performance Specification tab forms are the same. For information on these buttons refer to the topic titled Specify Attributes Dialog Box Buttons.

Maximum Ratings Tab Allows you to enter maximum ratings for components. These fields will change depending on the component category you have chosen.

The Selections for a field are displayed by clicking on the downward pointing arrow to the right of a field.

The Display check box to the right of the downward pointing arrow toggles the display of that fields information in the Search Results dialog box.

The buttons for the General Information, Maximum Ratings and Performance Specification tab forms are the same. For information on these buttons refer to the topic titled Specify Attributes Dialog Box Buttons.

Performance Specifications Tab Allows you to enter performance specifications for components. These fields will change depending on the component category you have chosen.

The Selections for a field are displayed by clicking on the downward pointing arrow to the right of a field.

The Display check box to the right of the downward pointing arrow toggles the display of that fields information in the Search Results dialog box.

The buttons for the General Information, Maximum Ratings and Performance Specification tab forms are the same. For information on these buttons refer to the topic titled Specify Attributes Dialog Box Buttons.

Specify Attributes Dialog Box Buttons The buttons are summarized as follows:

Options button	Opens a dialog box which gives you the option of searching for components with undefined values, and allows you to limit the number of components that will be displayed in the Search Results dialog box.
Clear button	Sets all search criteria to their default values.
Back button	Takes you back to the Select Category dialog box.
Next button	Does not function in this dialog box.
Finish button	Closes the current dialog box, searches for components based on the entered search criteria, and opens the Search Results dialog box.
Cancel button	Closes the Parametric Search Wizard.

Generic Tab Form Field Selections The Parametric Search Wizard, Generic tab form field options are summarized as follows:

Any	Searches for any component parameters, without limitation, in that field group.
Between	Allows you to search between two entered values.
=	Allows you to search for a value equal to your entered value.
>=	Allows you to search for a value equal to, or greater than, your entered value.
<=	Allows you to search for a value equal to, or less than, your entered value.

Search Results Dialog Box Displays the results of the parameter search. Every parameter will have a separate column.

You can select and place parts from this dialog box, view more information about parts, or return to the Specify Attributes tab forms to change search parameters.

The Search Results dialog box buttons are described as follows:

Back button	Goes back to the Specify Attributes tab forms so you can change search parameters.
Place button	Places a selected part into your current schematic window.
Replace button	Replaces all selected parts in the schematic window with the part highlighted in the Search Results dialog box.
Template button	Opens a template viewing window, allowing you to view the underlying template of a selected part. The template cannot be edited in this window.
SaberBook button	Opens SaberBook to the section referring to the category you are using.
Close button	Closes the Parametric Search Wizard.

Options Menu

Allows you to set search parameters for the Search String field.

The Object Search Type and String Match Type menu items work together to search for parts.

The Preferences menu item allows you to access all Parts Gallery options.

The Options menu items are summarized as follows:

Object Search Type

Allows you to limit your ASCII text search in the Search String field to specific object types. You can limit searches to Part Name, Symbol Name, or Template Name.

The Any Field item allows searches of all of a category, and all of its associated sub-categories.

String Match Type

Allows you to search for text strings Containing, Beginning With, or Equal To text typed in the Search String field.

Preferences

Opens the Parts Gallery Preferences dialog box. This dialog box allows you to change your display preferences, set search parameters, and toggle warning message displays.

Preferences Dialog Box

Allows you to change your display preferences, set search parameters, and toggle warning message displays.

General Tab The General tab items are summarized as follows:

Add placed parts to User Parts Library

Automatically adds parts placed on your schematic into the User Parts Library.

Show Message when Placing Parts

Toggles the display of an informational message about placing parts.

Show Parametric Search Wizard Message

Toggles the display of an informational message about the Parametric Search Wizard.

Category Tab (Saber Sketch) The Category tab items are listed below:

Aerospace	Control Systems
Automotive	IC (Integrated Circuit)
Characterized Parts Library	MAST Parts Library
Communication	Power System

View Tab The View tab allows you to toggle the information display at the bottom of the Parts Gallery window. The tab items are summarized as follows:

Description — Displays the part name shown in the Available Parts scrollable list.

Category — Displays the category that contains the selected part.

Symbol Name — Displays the symbol name associated with the part.

Template Name — Displays the template name.

Part License — Displays the part’s license name.

Database — Displays the location of the part database.

Symbol Preview — Toggles the display of the part symbol bitmap at the right of the Parts Gallery window.

Search Tab The Search tab items are summarized as follows:

Search part by — Allows you to limit your ASCII text search in the Search String field to specific object types. You can limit searches to Part Name, Symbol Name, or Template Name.

The Any Field item allows searches of all of a category, and all of its associated sub-categories.

Search match — Allows you to search for text strings Containing, Beginning With, or Equal To text typed in the Search String field.

Ignore case when doing search	Toggles case sensitivity in the Search String field.

Help Menu

The Help menu displays information about the Parts Gallery. The menu items are summarized as follows:

Help on Part	Opens SaberBook online documentation and displays the Template Description for the selected part.
Help on Parts Gallery	Opens SaberBook online documentation and displays the instructions for Parts Gallery menus and features.
About Parts Gallery	Provides copyright and version information.

Fields and Lists

Category Name Field

The Category Name field displays the currently open category in the Available Categories scrollable list.

Search String Field

Allows you to search for parts by part name, symbol name, or template name. You can search for complete text strings, or fragments of text strings. You can set the field to be case sensitive or not case sensitive. Searches are performed globally across the database, scoped only the simulator selection settings; searches are not bounded by the current category (set in the Category Name field).

To set search parameters

- Use the Options pull-down menu.

To search for a subset of parts

- Type in the full name of a part, or type in a few letters of a name of a part and press the Search button. Parts matching the text in the Search String field will be displayed in the Available Parts scrollable list, and the number of parts in the list is displayed in the Available Parts count field.

To clear the Search String field

- Use the Back Space key on your keyboard.

Available Categories List

Displays all of the part categories that are available to you and allows navigation through all of your category levels. Results are further narrowed according to the simulator selection you have made; just the Saber Simulator or the SaberHDL simulator or the union of both. Part names associated with a selected category are displayed in the Available Parts scrollable list and the number of available parts is displayed in the Available Parts count field. By default, the first part in the list is selected, and the picture of the symbol, with the symbol name above it, appears to the right of the Available Parts scrollable list. Select another part from the list and the symbol image and name will be updated.

To display the parts available in a category, and to navigate category levels

- Double click on a category name.

To move to the top category level

- Double click on /(go to top category).

To move to go up a single category

- Double click on .. go up a category.

To go directly to a category

- Type the path of the category in the Category Name field, or type a part of the category name and press the Enter key on your key board.

To clear the Category Name field

- Use the Back Space key on your keyboard.

Available Parts List

Displays all of the parts for the selected simulator(s) that are available in the current category and match the string specified in the Search String field. The number of parts appearing in the list is displayed next to the Available Parts list header. A picture of the selected part in the list, with the symbol name above it, appears to the right of the Available Parts scrollable list. By default, the first part in the list is selected.

To select a part

- Single click on the part name.

To place a part

- Double click on the part name or press the Place button.

Simulator Selectors

Confines the display of parts to those appropriate to the selected simulator, either Saber (MAST) or SaberHDL (VHDL-AMS). When both simulators are selected, the combines results of each is displayed.

Once you have generated a list of available parts, you can change the simulator selection settings and the count and contents of the Available Parts list will dynamically update to reflect parts that are available for the new simulator selection.

Buttons

The Parts Gallery buttons are summarized as follows:

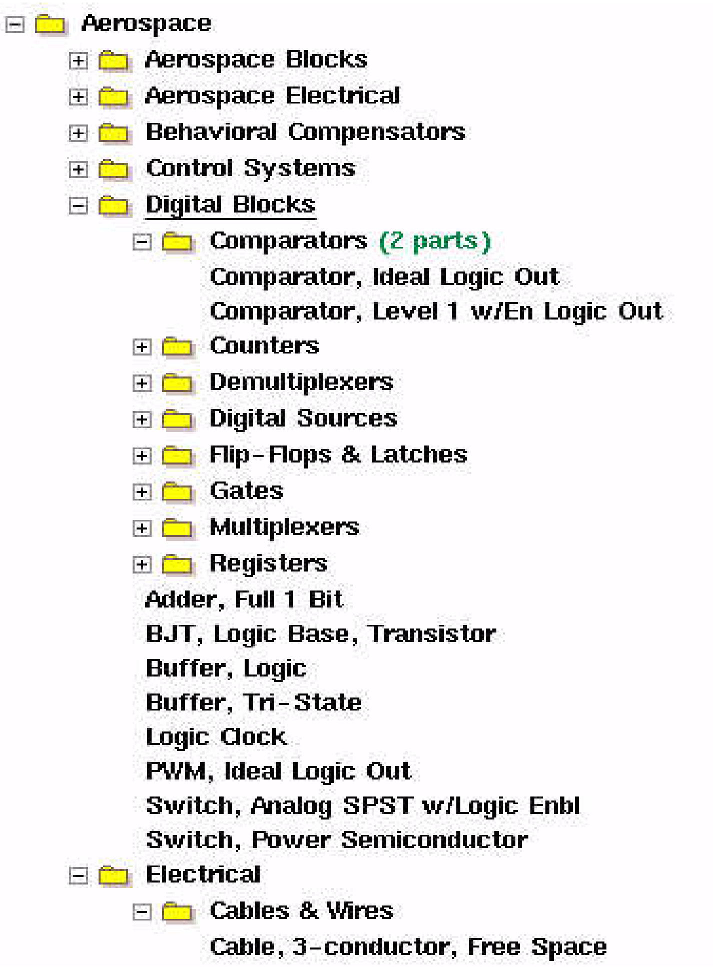
Place button	Places a selected part into your current Schematic window.
Replace button	Replaces all selected parts in the Schematic window with the part highlighted in the Available Parts scrollable list.
Close button	Closes the Parts Gallery.

The New Parts Gallery User Interface

Hierarchy Browser

The new Parts Gallery user interface shows categories and parts in a hierarchy tree. This provides you with a highly compact form that allows parts under more than one category to be viewed simultaneously

Figure 6 Hierarchy Browser.



- ⊟ Aerospace
  - ⊞ Aerospace Blocks
  - ⊞ Aerospace Electrical
  - ⊞ Behavioral Compensators
  - ⊞ Control Systems
  - ⊟ Digital Blocks
    - ⊟ Comparators (2 parts)
      - Comparator, Ideal Logic Out
      - Comparator, Level 1 w/En Logic Out
    - ⊞ Counters
    - ⊞ Demultiplexers
    - ⊞ Digital Sources
    - ⊞ Flip-Flops & Latches
    - ⊞ Gates
    - ⊞ Multiplexers
    - ⊞ Registers
    - Adder, Full 1 Bit
    - BJT, Logic Base, Transistor
    - Buffer, Logic
    - Buffer, Tri-State
    - Logic Clock
    - PWM, Ideal Logic Out
    - Switch, Analog SPST w/Logic Enbl
    - Switch, Power Semiconductor
  - ⊟ Electrical
    - ⊟ Cables & Wires
      - Cable, 3-conductor, Free Space

The hierarchy window displays two types of items: categories (branch items) and parts (leaf items).

The selection mechanism allows one category and one part to be selected at the same time. The selected part does not necessarily belong to the selected category. The text of the selected category is underlined whereas the colors (background and foreground) of the selected part are reversed. The selected part is shown in the preview area and can be exported through the clipboard for placement in the active schematic (middle mouse click in the schematic window). When the mouse cursor is moved on a category, the total number of parts it contains is shown on the right of the category name (including parts in subcategories).

To operate on the selections (category or part), the following mouse event and key stroke bindings are set:

- Single mouse click a category icon to toggle the state of the category (open or closed) and select the category. Note that closing a category containing the current part selection keeps the part selected.
- Single mouse click a category name to select the category (without opening or closing it).
- Double mouse click a category name to toggle the state of the category (expand or collapse the category).
- Single mouse click a part to select it and the category it belongs to.
- Double mouse click a part to select it and places an instance of the part in the active schematic. The category the part belongs to is also selected.
- Drag and drop a part name into the active schematic to place an instance of the part in the schematic.
- Right mouse click a part to select the part and access the Part menu with the following commands:

    a. View Model

    b. Part Info

    c. Help on Part

    d. Place

    e. Replace
    The menu item is disabled if no part is selected in the active schematic.

    f. Delete
    The menu item is disabled if the part is not a user-defined one.

- Click the <Up> and <Down> keys to change the part selection within a category where a part has previously been selected. There is no action if a part has not been selected in the tree. Also the selection does not move across categories.
- Right mouse click a category (icon and name) to select the category and display the Category menu with the following command items:

    a. Expand / Collapse the category.

    b. New Part
    Creates a part in the selected category. The newly created part is automatically selected in the hierarchy.

c. New Subcategory
Creates a subcategory for the currently selected category. The newly created category is automatically selected in the hierarchy.

To create a top-level category (which does not have a parent), right mouse click the white background area of the Hierarchy Browser and select the menu item New Top Category.

d. Delete
Disables the menu item when the category is not a user-defined one.

- Right mouse click the white background area of the Hierarchy Browser (where there is no text or icon) to access a menu with the following command items:

  a. Collapse All
  Collapses all categories.

  b. New Top Category
  Selects the newly created category.

  c. Parametric Search

  d. Simulator Filter: Saber or SaberHDL

  e. Update Database

  f. Preferences

  g. Help on Parts Gallery

- Click the <Delete> or <Backspace> key to delete the selected part (if a user-defined part is selected).
- Single mouse click in the background of the Hierarchy Browser to deselect the active category. This is necessary if you want to create a new top-level category using the New Category icon button. Otherwise, a subcategory of the selected category will be created.
- Click <Enter> or <Return> to place the selected part in the active schematic.
- Click the <g> key to toggle the visibility of Parts Gallery. This is a user definable hotkey (Sketch>Edit>Schematic Preferences>User Definable Hotkeys).

Note that operations on categories or parts outside the user-defined categories are not allowed. For unauthorized commands, the associated menu items are disabled.

Also, some of the key stroke bindings described earlier (<Up>, <Down> and <Delete>) requires the hierarchy window to grab the focus. A binding is added so that the focus is grabbed upon entry of the mouse cursor in the hierarchy window. Upon exit, the focus is restored to the user interface feature that previously had it.

Window Layout

Besides combining parts and categories in a single hierarchy tree, a compact layout is provided

The default size of the preview image is a relatively compact 50x50 image size. You can set the size of the preview (right mouse click) and it will be saved as a configuration. The Part Preview menu also has a command to launch the symbol editor. Double mouse click anywhere in the preview image to place the part in the active schematic.

In Saber Sketch, the symbol preview no longer relies on xpm files, but is created on the fly from the symbol (.ai_sym). The generation of xpm files when adding new parts is now optional (off by default). The option to still generate xpm files remains to support the Frameways where the Parts Gallery preview still relies on the xpm files.

The Parts Gallery window contains two tab forms: Browse and Search. The Browse tab form shows the Hierarchy Browser introduced earlier. The Search tab form allows you to perform a string based search. Note that the entry field displays a selection of previously typed entries matching the current user input (auto-complete feature). The result of the search is a list of parts shown in a list box with the following bindings:

- Single mouse click to select a part (any selection in the Hierarchy Browser is deselected at that point).
- Click the <Up> and <Down> keys to change the part selection.
- Right mouse click to access the same Part menu found in the Hierarchy Browser, with the additional command Find in Hierarchy (which expands the hierarchy to show the part). This command is useful to locate the part in the hierarchy.

You can also invoke the Parametric Search Wizard from the Search Object menu.

The <Tab> key allows you to switch tabs.

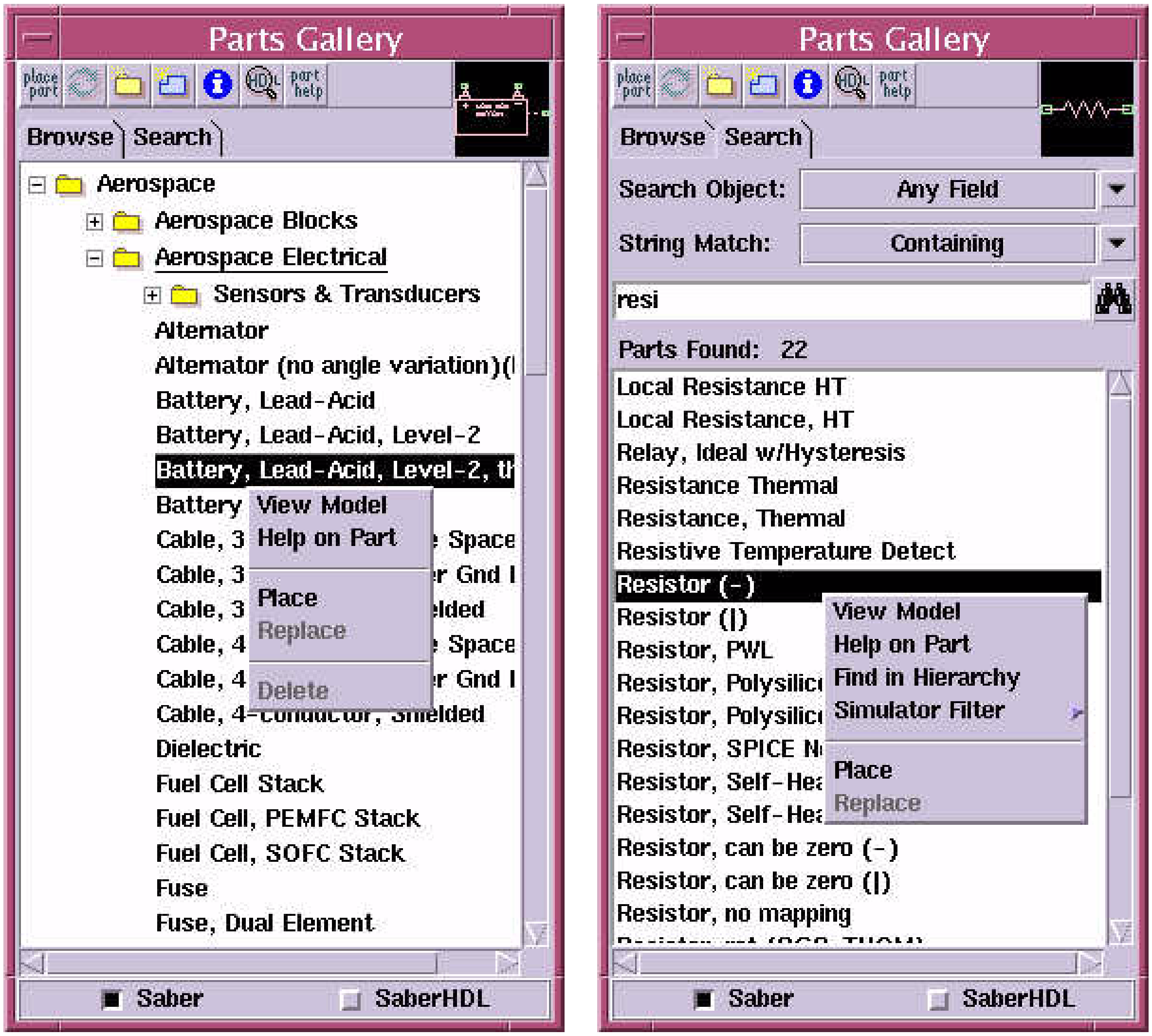
The toolbar contains icons for the following operations (from left to right).

- Place Part
- Replace Part
- Create Category
  The newly created category is a subcategory of the selected one. If no category is selected, the new category is a top-level one.
- Create Part
  The new part is placed in the selected Category. If no category is selected, the new part is added at the root of the hierarchy.
- Toggle Information View
  The part information is now available in a top-level window. The list of part attributes shown are: Part, Simulator, Category, Symbol, Template, Part Number, License and Database. You can filter out items from that list by using the Preferences form.
- View Model
  The MAST template or VHDL-AMS entity is displayed in the separate top-level window.
- Online Help on Part.

Beside the background popup menu of the Hierarchy Browser, the Preferences form is also available from the Edit menu (along with the other Preferences forms). Parts Gallery preferences can be edited without invoking Parts Gallery itself.

Similarly, online documentation for the application and Parts Gallery is available from the Help menu.

Figure 7 Browse (left) and Search (right) Tab Forms.



Parts Gallery

- Aerospace
  - Aerospace Blocks
  - Aerospace Electrical
    - Sensors & Transducers
    - Alternator
    - Alternator (no angle variation)(I
    - Battery, Lead-Acid
    - Battery, Lead-Acid, Level-2
    - Battery, Lead-Acid, Level-2, th
    - Battery
    - Cable, 3 … Space
    - Cable, 3 … r Gnd I
    - Cable, 3 … elded
    - Cable, 4 … Space
    - Cable, 4 … r Gnd I
    - Cable, 4-conductor, shielded
    - Dielectric
    - Fuel Cell Stack
    - Fuel Cell, PEMFC Stack
    - Fuel Cell, SOFC Stack
    - Fuse
    - Fuse, Dual Element

Parts Gallery

Search Object: Any Field

String Match: Containing

resi

Parts Found: 22

- Local Resistance HT
- Local Resistance, HT
- Relay, Ideal w/Hysteresis
- Resistance Thermal
- Resistance, Thermal
- Resistive Temperature Detect
- Resistor (-)
- Resistor (|)
- Resistor, PWL
- Resistor, Polysilic
- Resistor, Polysilic
- Resistor, SPICE N
- Resistor, Self-Hea
- Resistor, Self-Hea
- Resistor, can be zero (-)
- Resistor, can be zero (|)
- Resistor, no mapping

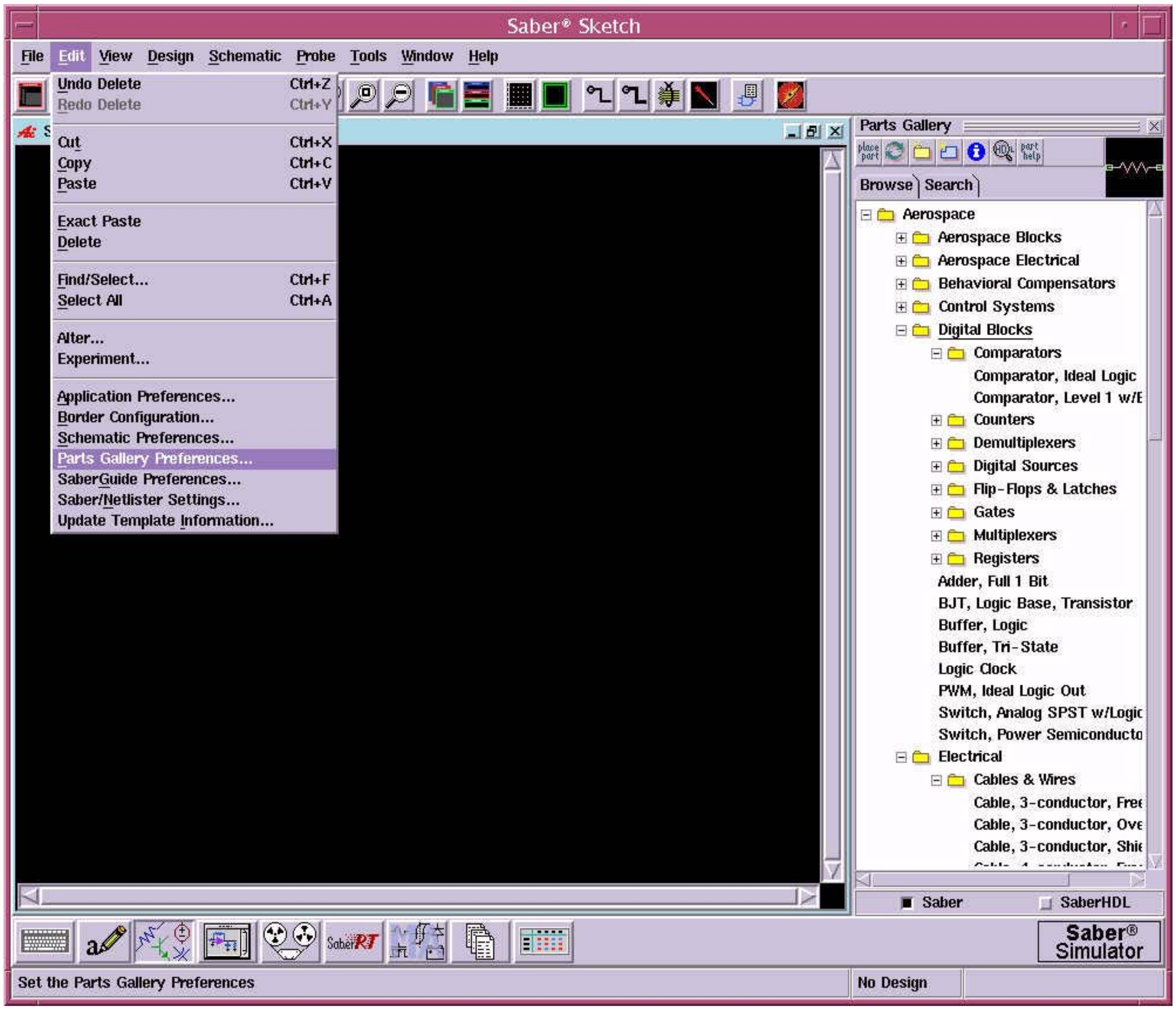
Dockable Top Level Window

In Saber Sketch and Saber Harness, Parts Gallery is a top-level window that can either be floating or docked on the right or left side of the schematic area. The window is initially docked on the left. You can drag it inside or outside the main application window by using the handle on top of the Parts Gallery window (vertical lines between the “Parts Library” title and the close button).

When the handle is released close to the right or left edge of the application main window, the Parts Gallery window re-docks itself. If it is released elsewhere, it becomes a floating top-level one. When undocked, the width and height of the window can be adjusted like any other top-level window. When docked, you can only change the width, the window stretches vertically to occupy the complete height of the schematic area.

The following illustration shows the Parts Gallery docked on the right side of the schematic area.

Figure 8 The Parts Gallery docked on the right side of Saber Sketch.



Set the Parts Gallery Preferences

The following settings are automatically saved at the end of a Saber Sketch or Saber Harness session and restored upon invocation of a new session:

- Last position where the window was docked (left or right)
- Width of the window
- Size of the preview

The visibility of Parts Gallery upon invocation of Saber Sketch or Saber Harness is a configuration that is saved with File> Configuration>Save.

Handling of the Frameways

On the Frameways, the new Parts Gallery differs from Saber Sketch and Saber Harness in several ways.

The preview image is expected to have a poorer resolution (for small size previews). The new Parts Gallery in Saber Sketch no longer relies on pre-built pixmap files for the symbol preview. The preview is drawn directly from the symbol file. This allows the preview to be scaled down. Since the routines for accomplishing this are not available on the Frameways, the current pixmap preview is used instead. Scaling down can still be performed using bitmap sub-sampling but the resolution isn't as good.

The Parts Gallery top-level window cannot be docked.

The drag-and-drop feature to place a part in the schematic is not supported.

The Parts Gallery top-level window contains a suite of pull-down menus to access the modeling tools, including Nspitos. The menu items are:

File->Update Database
File->Close
Edit->New Part
Edit->Part Reference
Edit->Delete Part
Edit->New Category
Edit->Delete Category
Edit->Preferences
Tools->View Template
Tools->Help on Part
Tools-><<the Model Architect tools suite>>
Help->Help on Part
Help->Help on Parts Gallery
Help->About Parts Gallery

Harness Modular Form

In Saber Harness, the Parts Gallery is invoked as a modal form to select Harness items such as bundles or shells. Invoked in that fashion, the new Parts Gallery user interface is modified as follows:

1. The OK and Cancel buttons are found at the bottom of the window. The focus is grabbed.
2. All icon buttons are gone (Place, Replace, View Model, Create Category, Create Part) except the Information one. The information view is no longer a separate top-level window but is packed within the Parts Gallery modal form (to avoid focus conflicts between windows).
3. There are no popup menus.
4. In the Search tab, the “Template Name” object option is removed.

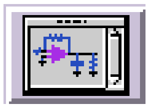
5

Using the Design Tool

The Design tool allows you to manage hierarchical schematics in a design. A list of all schematics related to a particular top-level schematic, if any, is displayed. The schematics are displayed hierarchically.

Accessing the Design Tool

The Design tool icon is located in the Tool bar at the bottom of the Saber Sketch work surface.



To open or close the Design tool:

- Single-click on the icon with the left mouse button.

Design Tool Hierarchy Browser

The Design Tool displays the content of a Sketch or Harness design within a hierarchy browser. At each level of the hierarchy, entries are listed in alphabetical order. To expand or collapse a node within the hierarchy, either single-click the +/- node icon, or double-click the node label or icon.

You can also use the Design Tool to archive a design and its related files in a user-selected directory.

Hierarchy Browser Menu Items

When you right-click in the *background* of the hierarchy browser (that is, in an area without an icon or text), the Design Tool displays a popup menu with the following items:

Collapse All — Collapses the entire hierarchy.

Search — Opens a window in which you can specify a search pattern. Saber searches the design hierarchy, recursively, for design entities whose label matches the specified pattern. You can use the wildcard character * in the search pattern.

Settings — Opens the Design Tool Settings form. You can choose the following:

- Whether to display sheets.
- Whether to display leaf symbols.
- Whether to expand the hierarchy beyond the leaf symbol to show the MAST template hierarchy.
- Whether to group and insert a level of hierarchy for instances that share the same definition.
- Color codes to identify the location of design entities in the file system.
- Registration of foreign routine files for archiving purposes. If a MAST template relies upon a user-defined foreign routine, the routine files must be registered as being tied to the template (primitive). If the routine reads a static data file, it also must be registered as a routine file. If the routine reads a data file passed as model argument, the argument name has to be registered.

Archive Design — Copies the files associated with this design into a directory that you specify. The archive directory must not be the same as the design directory.

Report on Design — Opens a report window that lists the constitutive elements of the entire design.

Leaf Symbol Menu Items

When you right-click on an instance of a leaf symbol (that is, a symbol with no schematic definition), the Design Tool displays a popup menu with the following items:

Select Symbol
Selects the symbol instance in the schematic.

View Template/ Entity
Loads the HDL code in the appropriate viewer; only applies if the model has an underlying model.

Properties
Opens the Property Editor.

Symbol Editor
Opens the symbol in a new symbol editor window.

Archive Symbol
Copies the files associated with this symbol into a directory that you specify. (Not currently supported in Saber Harness.)

Report on Symbol
Opens a report window that lists the constitutive elements of the selected symbol instance.

Hierarchical Symbol Menu Items

When you right-click on a hierarchical symbol, the Design Tool displays a popup menu with the following items:

Select Symbol
Selects the symbol instance in the schematic.

Open Schematic
Opens the schematic if it is not already open.

Properties
Opens the Property Editor.

Symbol Editor
Opens the symbol in a new symbol editor window.

Archive Symbol
Copies the files associated with this symbol into a directory that you specify. (Not currently supported in Saber Harness.)

Report on Symbol
Opens a report window that lists the constitutive elements of the selected symbol instance.

Sheet Icon Menu Items

When you right-click on a sheet icon (if Display Sheets is enabled and the schematic contains multiple sheets), the Design Tool displays a popup menu with the following items:

Select Symbol Selects the symbol instance in the schematic.

Open Sheet Opens the sheet in a new schematic editor window.

Rename Sheet Allows you to rename the sheet.

Template Icon Menu Items

When you right-click on a template icon, Saber displays a popup menu with the following items:

View Template/Entity Opens a Template Viewer

Archive Model Copies the files associated with this template into a directory that you specify. (Not currently supported in Saber Harness.)

Report on Template Opens a report window that lists the constitutive elements of the selected template.

6

Using the Signal Manager

This chapter explains how to use the Signal Manager.

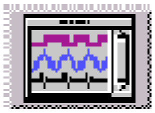
The Signal Manager is a CosmosScope tool, which is used for managing the signals generated by a design analysis. The Signal Manager allows you to open plotfiles, filter out unwanted signals, place signals into a CosmosScope graph window, and place signals into the Calculator.

The Signal Manager is described in the following topics:

- Accessing the Signal Manager
- Opening Plotfiles
- HSPICE Sweep Filtering
- Searching Multiple Plotfiles for Signals
- Signal Manager Dialog Box
- Signal Manager Menus
- Signal Manager Buttons
- Signal Manager Preferences
- Signal Manager Plotfile Window

Accessing the Signal Manager

The Signal Manager icon is located in the Tool bar at the bottom of the work surface.



To open or close the Signal Manager, click the icon with the left mouse button.

Opening Plotfiles

To open a plotfile:

1. Click the Signal Manager icon. The Signal Manager dialog box opens.
2. Click the **Open Plotfiles...** button to open the Open Plotfile dialog box.
3. Browse to the location of the plotfile you want to open, select the plotfile name, and click the **OK** button. The Plotfile window appears.
4. Select the signal(s) you want to plot. You can select more than one signal by holding the Control key as you select signals. If you want to deselect all highlighted signals, click the **Deselect** button.
5. Click the **Plot** button to plot the selected signal(s) in the current graph window.

You can also open plotfiles and open the Signal Manager by choosing **File > Open > Plotfiles...** from the main menu bar or by clicking the "Open" icon on the CosmosScope icon bar.

Copying a Signal into the Calculator

- To copy a signal into the calculator, highlight the signal in the plotfile window; place the mouse cursor in the calculator entry field, and click the middle mouse button.

Copying a Signal into Saber Simulator Guide

- When using either CosmosScope or Saber, copy a signal into a Guide form by highlighting the signal in the plotfile window, placing the mouse cursor in the form field, and clicking the middle mouse button.

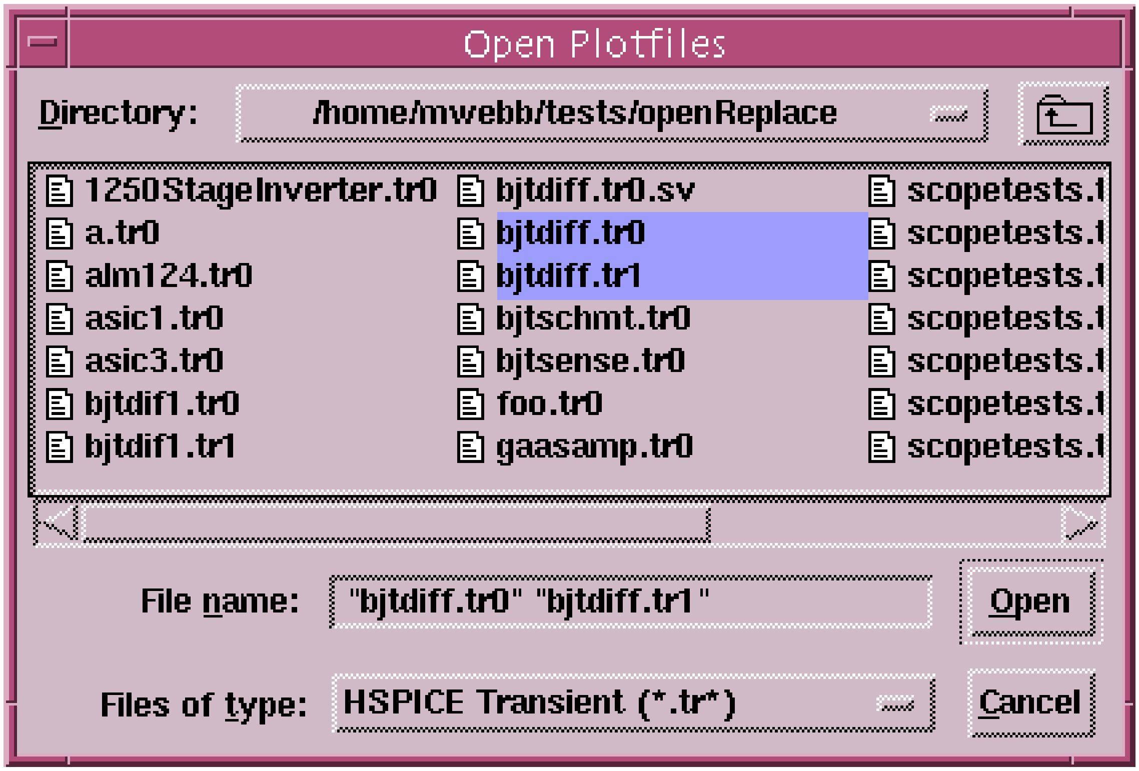
HSPICE Sweep Filtering

It is possible to have a single HSPICE run produce multiple output files. For example, you can set CosmosScope to produce multiple transient analysis files for each temperature setting with temperature as the “Parameter.”

To use this feature in the signal manager, click "Open Plotfiles." In the "Open Plotfiles" dialog, select two or more analyses output files with the same root name and the same type of analysis (e.g., bjtdiff.ac0, bjtdiff.ac1). If the measurement analysis files are available (in this case, bjtdiff.ma0, bjtdiff.ma1), the tool will permit you to specify the range of parameters you are interested in. (Select 2 or more runs from the same HSPICE execution .tr0 .t1.)

The HSPICE Sweep Filtering dialog will appear. Alter Parameter Name, Minimum, and Maximum will become enabled when Selection Range is set to Selection. (To select All output files, you must select two or more analyses first.)

The pull down menu will let you select sweep parameter or measurement. Minimum:/Maximum: will let you specify the range to open. Click OK, and the plotfiles whose parameters fall within that range will be opened and displayed in the signal list dialog.

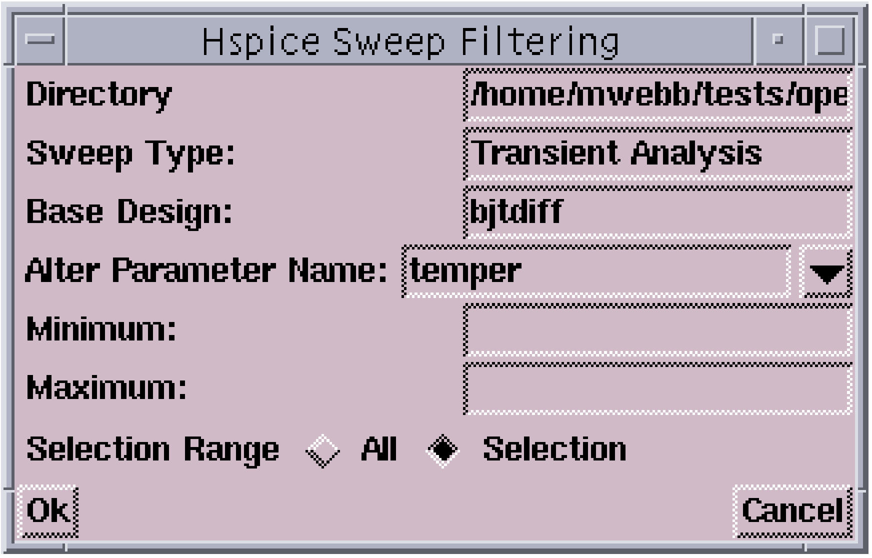


Open Plotfiles

Directory: /home/mwebb/tests/openReplace

File name: "bjtdiff.tr0" "bjtdiff.tr1"

Files of type: HSPICE Transient (*.tr*)



Hspice Sweep Filtering

Directory: /home/mwebb/tests/ope

Sweep Type: Transient Analysis

Base Design: bjtdiff

Alter Parameter Name: temper

Minimum:

Maximum:

Selection Range ○ All ● Selection

Searching Multiple Plotfiles for Signals

CosmosScope and Saber provide a way to search several waveform plotfiles for signals that may be of interest. For example, you may want to display signals for comparison from several different plotfiles. To do this, follow these steps:

1. Click the Signal Manager icon button on the lower part of the display.
2. In the Signal Manager window that appears, click Open Plotfiles.
3. In the Open Plotfiles window, navigate to the directory containing a waveform plotfile(s) of interest and select a plotfile.
4. From the swept results of a single design, type a pattern name on the Filter edit line.
5. Click the Filter down arrow to display the Filter Attributes menu.
6. In the Filter Attributes menu, select the desired filter attribute(s).
7. Signal names corresponding to your filter attributes will be highlighted in the File Name window.
8. You may plot any or all of these signals.

Note:

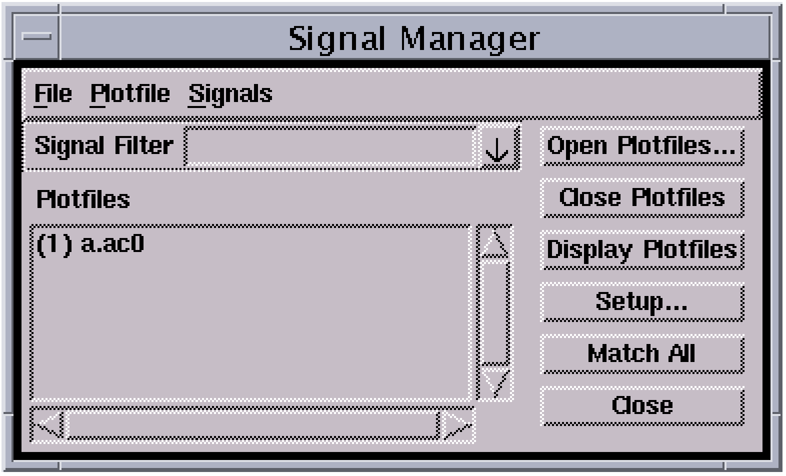
When you enter search criteria on the Filter edit line in the File Name window, regular expression syntax can be used to specify the search parameters.

Signal Manager Dialog Box

The Signal Manager dialog box allows you to open multiple plotfiles, and manage how they are displayed.

Open plotfiles are displayed in the Plotfiles scrollable list box. Plotfiles are numbered sequentially in the order that they were opened.

If a plotfile contains multi-member signals, this count will be reflected in brackets to the right of the signal name.



Signal Manager Menus

The following menus are available from the Signal Manager dialog box.

File Menu	Manages opens, loads, and closes plotfiles.
Plotfile Menu	Manages plotfiles displayed in the Plotfiles scrollable list.
Signals Menu	Plots or deselects selected signals in a plotfile.

Signal Manager File Menu Items

The following items appear in the File menu list in the Signal Manager dialog box:

Open Plotfiles	Opens the Open Plotfiles dialog box allowing you to open new plotfiles. (You can also open plotfiles by single clicking on the Open icon in the icon bar.)
Save As	Saves the selected plotfile to a Saber pl (*.*.ai_pl), Text (*.txt), or AWD (*.ai_awd) file type.

Reload All Plotfiles
Reloads all of the plotfiles in the Plotfiles list.

Reload Selected Plotfiles
Reloads plotfiles highlighted in the Plotfiles list.

Close Selected Plotfiles
Closes plot files highlighted in the Plotfiles list and removes the plotfile from the list.

Close All Plotfiles
Closes all plotfiles displayed in the Plotfile list whether or not they are highlighted and removes the plotfiles from the list.

Close Window
Closes the Signal Manager dialog box.

Signal Manager Plotfile Menu

The following items appear in the Plotfile menu list in the Signal Manager dialog box:

Display Selected
Opens plotfile windows for the plotfiles highlighted in the Plotfiles list.

Display All
Opens plotfile windows for all plotfiles displayed in the Plotfile list whether or not they are highlighted.

Hide Selected
Closes plotfile windows highlighted in the Plotfiles list. The plotfiles remain listed.

Hide All
Closes all plotfile windows displayed in the Plotfiles list whether or not they are highlighted. The plotfiles remain listed.

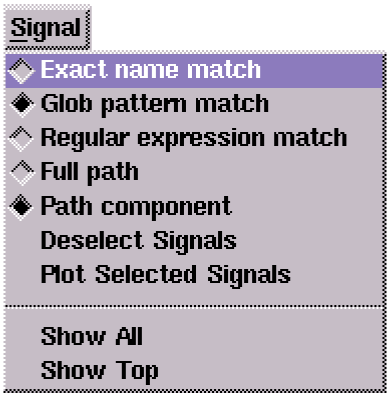
Stack Selected
Opens and stacks plotfile windows highlighted in the Plotfiles list. Stack orientation is managed with the Setup dialog box.

Stack Visible
Stacks plotfile windows which are visible on the work surface. Stack orientation is managed with the Setup dialog box.

Stack All
Stacks all plotfile windows in the Plotfiles list whether or not they are highlighted. Stack orientation is managed with the Setup dialog box.

Signal Manager Signals Menu Items

The following items appear in the Signals menu list in the Signal Manager dialog box:



Options under Signal Manager - Signals Menu

Note:

> On some Windows systems, the selection buttons do not appear as shown above. Click the left mouse button on the highlighted entry to activate the selection as a check mark.

There are three “Match” options from which to choose. these will be used in conjunction with the Signal Filter field:

- Exact name match: will search for an exact name match with your Signal Filter entry;
- Glob pattern match: filename pattern matching using *,?[abcd] wildcard characters
- Regular expression match: vi/emacs/tcl style patterns using the following wildcard symbols: ^ $. [ ] | ( )

The wildcard characters have the following meanings:

Pattern language	**Wildcard Character**	**Meaning**
Glob	*	any sequence of characters
Glob	?	any single character
Glob	[abc]	character a, or b, or c

Pattern language	**Wildcard Character**	**Meaning**
Regular Expression	.	any character
Regular Expression	<x>|<y>	anything matching regular expression <x>, or regular expression <y>
Regular Expression	( )	parenthesis used for grouping
Regular Expression	$	end of path component, or full path, depending on settings
Regular Expression	^	start of path component, or full path, depending on settings
Regular Expression	[abc]	character a, b, or c
Regular Expression	[^abc]	any character but a, b, or c

Second, select one of two target options:

Full Path, Path Component
: Specify whether the pattern is matched against single path components, or against full paths. This allows, for example, very specific selections of particular signals for only a given range of plotfile containers. The default target for pattern matching is “path component.”

The last two options on the Signal Manager Signals menu are:

Plot Selected Signals
: Plots signals that are highlighted in open plot file windows into the active graph window.

Deselect Selected Signals
: Removes the highlight from highlighted signals in displayed plot file windows.

Signal Manager Signal Filter Field

Enter your filtering pattern in the Signal Filter field; from the adjoining pulldown, select whether the pattern denotes a set to be Shown, Hidden, Selected, or

Deselected. Choosing the action causes it to be applied to the signals or containers matching the filter pattern in all plot files.

In addition, you may set a Cumulative mode. With Cumulative turned off (the default mode), each action on a set of signals or containers denoted by the filter pattern implies that signals or containers not matching the filter pattern have the reverse operation applied to them. For example, if you enter “v*” as a filter pattern, have “Show” as the action, and “Cumulative” off, all the v* entries appear, and other entries not matching that pattern disappear. Again, if the action had been “Select”, all of the v* entries would be highlighted (selected) and other entries not matching that pattern, deselected.

Signal Manager Buttons

The following buttons appear in the Signal Manager dialog box:

Open Plotfiles... Brings up the Open Plotfiles dialog box, allowing you to open new plotfiles.

(You can also open plotfiles by choosing **File** > **Open** > **Plotfiles...** in the menu bar or by single clicking on the Open icon in the icon bar.)

Close Plotfiles Closes plotfiles highlighted in the Plotfiles list and removes the plotfile from the list.

Display Plotfiles Opens plotfile windows for the plotfiles highlighted in the Plotfiles list.

Setup... Opens the Signal Manager Setup dialog box.

Match All Select this to match corresponding signals in all open plot files to the active graph.

Close Closes the Signal Manager dialog box.

Signal Manager Preferences

Clicking the **Setup** button or choosing **Edit > Preferences > Signal Manager** tab opens the Signal Manager Setup dialog box, which allows you to manage stack position, stack orientation, and plotfile window size. A stack is an ordered group of plotfile windows that are placed one on top of the other.

The Setup dialog box is summarized below:

Signal Listbox Options

The following signal listbox options are available:

- Anchor Position - Selects the position of the Signal Manager window stack on your screen. For example, if you choose C (center), the stack is centered in the screen. If you choose N, the stack is offset to the middle and top ("North") portion of the screen.
- Stacking Orientation - Changes the arrangement of plotfile windows in the stack.

  - Horizontal - Stacks the plotfile windows. Windows anchored to the NW, W, or SW are arranged from left to right. Windows anchored to the NE, E, or SE are arranged from right to left.

  - Vertical - Stacks windows from top to bottom.

  - Diagonal - Stacks windows diagonally from left to right. This is the default setting.

  - Multiple Listboxes - Creates one signal list window for each opened plot file.

  - Signal Listbox - Creates one signal list window for all open plot files to share. Double clicking on the plotfile name in the Signal Manager marks the plotfile as active.
- Locate signal in listbox - Selects a signal in the Graph window and automatically expands the hierarchy in the associated Signal Manager dialog box. The default is off.
- Match button: plot multi-member waveforms - Selects between automatically creating a multi-member waveform from the matched signals or plotting individual signals to the same graph region. The default is to create the multi-member waveform.
- Width - Changes the width of plotfile windows (in pixels). The default is 150 pixels.

- Height - Changes the height of plotfile windows (in pixels). The default is 200 pixels.
- Stacking Offset - Changes the overlap spacing of the stack (in pixels). The default is 30 pixels

Signal Manager Display

Selects which Signal Manager pop-up windows to display. At the end of the field is pull-down menu that contains the following options:

- Signal Manager and Listbox - Selecting this option displays both the Signal Manager dialog box and the Signal Manager plotfile window. This is the default.
- Signal Manager Only - Selecting this option displays only the Signal Manager dialog box.
- Signal Listbox Only - Selecting this option from displays only the Signal Manager plotfile window.
- None - No Signal Manager windows are displayed.

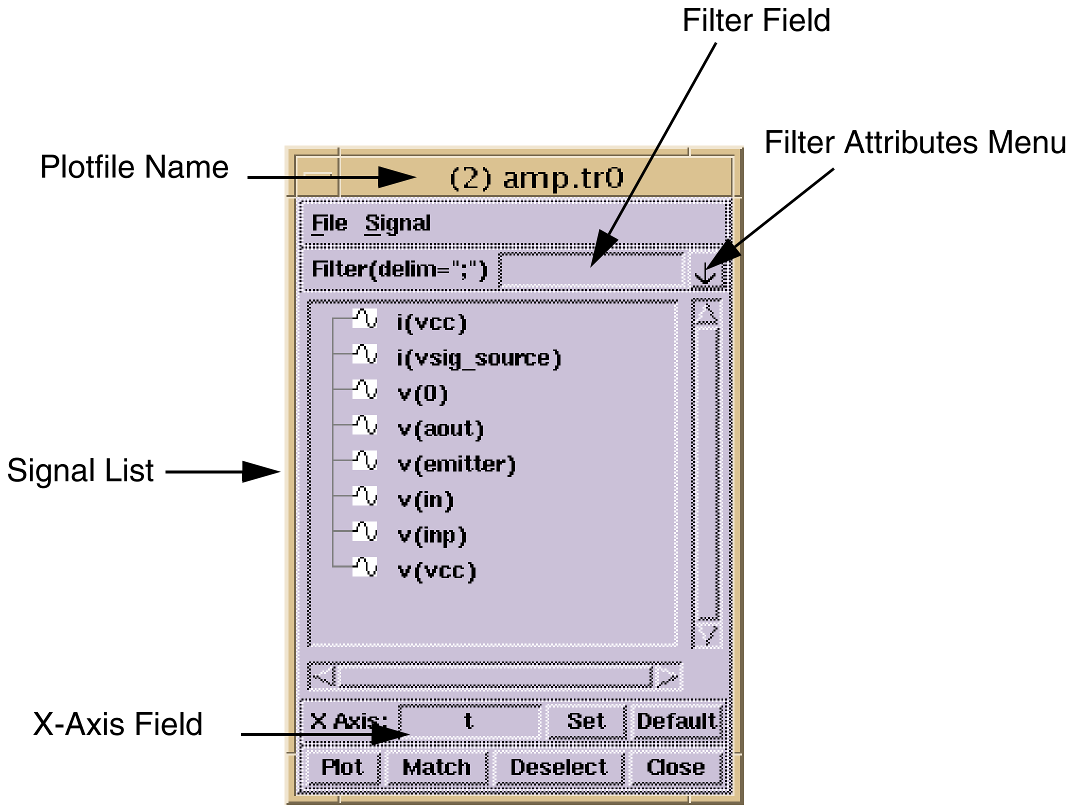
Apply To

This option designates which plotfiles are affected by the settings you specify in the Signal Manager Setup:

- Visible - All visible plotfiles.
- Selected - All selected plotfiles.
- All - All plotfiles currently open.

Signal Manager Plotfile Window

Choosing a plotfile in the Open Plotfiles dialog box opens the Plotfile window.



The Plotfile window displays the name of the plotfile and a scrollable list of signals that can be displayed in a graph window.

The Plotfile window is further described in the following topics:

- Plotfiles Dialog Box Menus
- Plotfiles Dialog Box Fields
- Plotfiles Dialog Box Use Notes

Plotfiles Dialog Box Menus

The Plotfile dialog box File menu items are summarized as follows:

Reload Plotfile	Reloads selected plotfiles.

Close Plotfile

Closes the plotfile window and removes the file from the Signal Manager Plotfiles list.

Close Window

Closes the plotfile window. The plotfiles remain listed.

The Plotfile dialog box Signal menu items are summarized below:

Match options

As in the main Signal Manager window, you may select the signal pattern matching language (exact, glob, or regular expressions).

Path options

Select either Full or Path Component for the pattern matching target.

Deselect Selected Signals

Removes the highlight from highlighted signals in displayed plot file windows.

Plot Selected Signals

Plots signals that are highlighted in open plot file windows into the active graph window.

Show All

Displays all signals in all levels of hierarchy.

Show Top

Displays only those signals at the top level of hierarchy.

Plotfiles Dialog Box Fields

The following list summarizes the Plotfile dialog box fields and buttons, from top to bottom:

Filter(delim=”?”) Field

This label next to the filter pattern provides the delimiter character used for signal path names for this plotfile. If you wish to match against full path names, separate individual path components with the character (“?” in this case). Enter your search pattern, then select the filter attribute from the options in the adjoining pulldown: Show, Hidden, Selected, or Deselected. This attribute is retained when you enter subsequent search patterns, until you reset the selected attribute option. And, as in the main Signal Manager window, you may set or unset the cumulative mode, with the same meaning.

Signal list

The Signal list is a scrollable list of signals that can be displayed in a graph window.

Signals are organized by type, such as analog, digital, or event; Verilog designations include bus, register, register_int, and string.

Match Button

Search the current Signal List to find and plot signals that have the same name as the signals in the active graph.

Plot Button

Plots highlighted signals to the active graph window.

Deselect Button

Deselects highlighted signals in the signal list.

Close Button

Closes the plotfile window. The plotfiles remain listed.

X-Axis Field

Changes the x-axis for plotting and measuring. You can change the x-axis when using HSPICE, HSPICE Meas, and Saber PL-type plotfiles that are analog waveforms with the same number of data points.

In the Open Plotfiles dialog box, the current x-axis appears in the x-axis field. The **Set** button sets the selected waveform in the Signal list as the active x-axis, and the **Default** button sets the active x-axis back to the default plotfile x-axis.

Plotfiles Dialog Box Use Notes

Signal Containers

- Within the Plotfile dialog box Signal List, signal containers are always preceded by a - or a + and contain sets of related signals. A - indicates that the signal container is open, and related signals are displayed beneath it. A + indicates that the signal container is closed, and no signals are displayed.

  To open a + signal container, double click on the container with the left mouse button.

To select signals for placement into the graph window

- Place the mouse cursor over the signal and single click the left mouse button. Any number of signals can be selected.

To unselect a signal

- Place the mouse cursor over the highlighted signal and single click the left mouse button.

To place the highlighted signals into the active graph window

- Single click on the Plot button, or double click on the highlighted signal, or use the middle mouse button to paste the signals into the desired graph region.

To display signal containers or individual signals in the Signal List

- Type the name in the Filter field and press the Return key on your work station keyboard.

If you want to display all signals and containers

- Delete all entries in the Filter field and press the Return key.

7

Using the Measurement Tool

This chapter explains how to use the Measurement Tool.

The Measurement Tool allows you to perform a variety of measurement operations on displayed waveforms in the CosmosScope Waveform Analyzer. The results of the measurements are displayed in the graph along with the waveform.

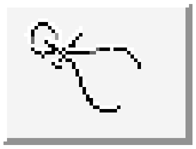
The Measurement Tool is similar to the functions available with the Saber Simulator through its command-line interface or via Saber Guide’s **Analysis > Batch** menu option, although some measurements may differ slightly.

The Measurement Tool description is divided into the following topics:

- Accessing the Measurement Tool
- List of Measurement Operations
- Using the Measurement Tool
- Managing Measurement Results
- Multi-Member Waveform Measurements
- Setting Measurement Preferences
- Topline/Baseline Calculation

Accessing the Measurement Tool

The Measurement Tool icon is located in the Tool bar at the bottom of the work surface.



To open or close the Measurement Tool, single click on the icon with the left mouse button.

List of Measurement Operations

The following list identifies the measurements that are available through the Measurement Tool:

Measurement	**Category**	**Description**
AC Coupled RMS	Levels	Displays the RMS value of the AC component of a waveform.
ACPR (Adjacent Channel Power Ratio)	RF	Displays the ACPR_upper and ACPR_lower ratios of an envelope power spectrum waveform.
Amplitude	Levels	Displays the amplitude of a waveform.
At X	General	Displays the Y-axis value at a particular X-axis point on a waveform.
Average	Levels	Displays the average value of a waveform.
Bandwidth	Frequency Domain	Displays the bandwidth, the low, high, or center frequency, or the level at which the measurement is made for a bandpass-shaped waveform.
Baseline	Levels	Displays the baseline level of a waveform.
Cpk	Statistics	Displays an indicator of the process capability for a waveform.
Crossing	General	Displays the crossing (intersection) points of two waveforms.

Measurement	**Category**	**Description**
Damping Ratio	s Domain	Displays the damping ratio of a point on a waveform.
dB	Frequency Domain	Displays the dB value on a point on a waveform.
Delay	Time Domain	Displays the delay between the edges of two waveforms.
Delta X	General	Displays the X-value difference between two Y-axis points on one or two waveforms.
Delta Y	General	Displays the Y-value difference between two X-axis points on one or two waveforms.
Dpu	Statistics	Displays the number of defects per unit of a scatter plot waveform.
Duty Cycle	Time Domain	Displays the duty cycle of a periodic waveform.
Eye Diagram	Time Domain	Displays the behavior of a waveform cycle during a specific period of time.
Eye Jitter (see Adding Jitter Measurements to Eye Diagrams)	Time Domain	Displays specified period or cycle-to-cycle jitter measurements.
Eye Mask	Time Domain	Display the maximum width and height of an eye diagram opening.
Falltime	Time Domain	Displays the falltime of a waveform.
Frequency	Time Domain	Displays the frequency of a periodic waveform.
Gain Margin	Frequency Domain	Displays the gain margin in dB of a complex waveform.
Highpass	Frequency Domain	Displays the corner frequency of a waveform with a highpass shape.

Measurement	Category	Description
Histogram	Statistics	Displays a histogram of a waveform.
Horizontal Level	General	Displays a moveable horizontal line to identify Y-axis levels.
Imaginary	Frequency Domain	Displays the imaginary value of a point on a waveform.
IP2	RF	Display Input/Output Second Order Intercept Point (I/O IP2)
IP3/SFDR	RF	Display Input/Output Third Order Intercept Point (I/O IP3) or Spurious-Free Dynamic Range (SFDR).
Jitter	Time Domain	Displays the deviation of the significant instances of a signal from their ideal location in time.
Length	General	Displays the length of a straight line that connects two X-axis points on a waveform.
Local Max/Min	General	Displays the local maximum or minimum point on a waveform.
Lowpass	Frequency Domain	Displays the corner frequency of a waveform with a lowpass shape.
Magnitude	Frequency Domain	Displays the magnitude of a point on a waveform.
Maximum	Levels and Statistics	Displays the maximum value of a waveform.
Mean	Statistics	Displays the mean value of a waveform.
Mean +3 std_dev	Statistics	Displays the (mean + 3s) value of a waveform.
Mean -3 std_dev	Statistics	Displays the (mean -3s) value of a waveform.
Median	Statistics	Displays the median value of a waveform.

Measurement	Category	Description
Minimum	Levels and Statistics	Displays the minimum value of a waveform.
Natural Frequency	s Domain	Displays the natural frequency of a point on a waveform.
Nyquist Plot Frequency	Frequency Domain	Displays the frequency at a point on a Nyquist (or Nichols) plot.
Overshoot	Time Domain	Displays the overshoot of a waveform relative to a default or specified topline.
P1dB	RF	Display 1DB Compression Point (CP).
Pareto	Statistics	Displays a Pareto chart of a multi-member analysis.
Peak-to-Peak	Levels	Displays the waveform’s peak-to-peak value.
Period	Time Domain	Displays the period of a periodic waveform.
Phase	Frequency Domain	Displays the phase value on a point on a waveform.
Phase Margin	Frequency Domain	Displays the phase margin of a complex waveform in degrees or radians.
Point Marker	General	Displays a moveable point marker on the waveform to display the X-value and Y-value.
Point to Point		Displays the Slope value between two selected points.
Pulse Width	Time Domain	Displays the pulse width of a waveform.
Quality Factor	s Domain	Displays the quality factor of a point on a waveform.
Range	Statistics	Displays the range of Y-axis values covered by the waveform.
Real	Frequency Domain	Displays the real value of a point on a waveform.

Measurement	Category	Description
Risetime	Time Domain	Displays the risetime of a waveform.
RMS	Levels	Displays the RMS value of a waveform.
Settle Time	Time Domain	Displays the settle time of a waveform.
Slew Rate	Time Domain	Displays the slew rate of a waveform.
Slope	Frequency Domain and General	Displays the slope (optionally as a per-octave or per-decade value) of a waveform.
Standard Deviation	Statistics	Displays the standard deviation of a waveform.
Stopband	Frequency Domain	Displays the stopband, the low, high, or center frequency, or the level at which the measurement is made for a stopband-shaped waveform.
THD/SNR/ SINAD	Frequency Domain	Displays the Total Harmonic Distortion (THD), Signal to Noise Ratio (SNR), and Signal to Noise and Distortion Ratio (SINAD) on the resulting FFT waveform.
Threshold (at Y)	General	Displays the X-axis values at a particular Y-value on the waveform.
Topline	Levels	Displays the topline level of a waveform.
Undershoot	Time Domain	Displays the undershoot of a waveform.
Vertical Cursor	General	Displays a vertical cursor that spans different graphs, for X-value, Y-value, and delta Y measurements.
Vertical Level	General	Displays a moveable vertical line to identify X-axis levels.
Vertical Marker	General	Displays the Y-axis value at a particular X-axis point on multiple waveforms.

Measurement	Category	Description
X at Maximum	Levels	Displays the X-value corresponding to the maximum value of a waveform.
X at Minimum	Levels	Displays the X-value corresponding to the minimum value of a waveform.
Yield	Statistics	Displays the ratio of data points that fall between the specified upper and lower Y-axis values of a waveform.

Using the Measurement Tool

To use the Measurement Tool you must have waveforms displayed in the graph window.

From the Measurement dialog box you can select the measurement operation you want to perform, select the signal on which to perform the measurement, and select measurement preferences.

After performing measurements with the Measurement Tool, you can manage your measurement results with the Measure Results dialog box.

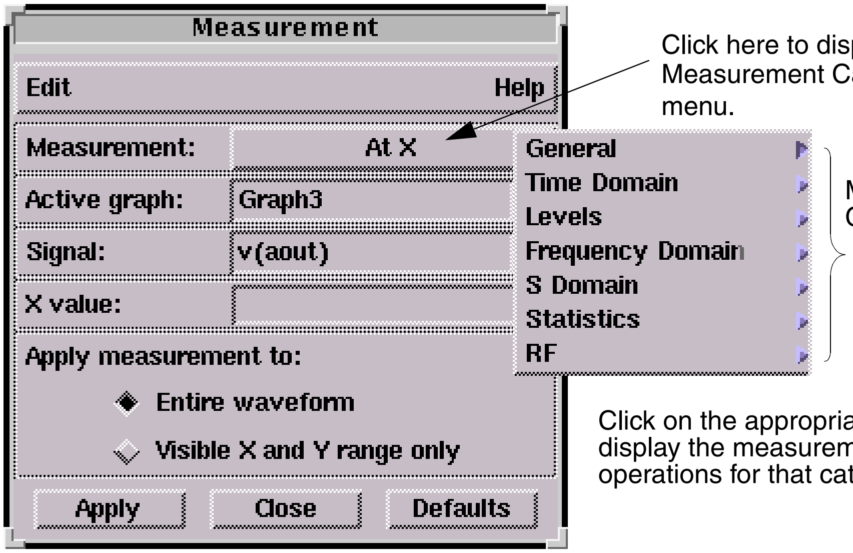
The following topics describe how to use the Measurement Tool:

- Measurement Dialog Box
- Selecting a Measurement
- Selecting a Signal for a Measurement
- Setting the Range of a Measurement
- Creating a New Waveform of Measurement Results

Measurement Dialog Box

The Measurement, Signal, and Apply Measurement to fields are always available in the Measurement dialog box. Other fields within the dialog box change or appear according to the particular measurement you are performing. These fields are described in the individual measurement operations.

The measurement operations are divided into categories as shown in the Measurement dialog box.



Measurement: At X

Active graph: Graph3

Signal: v(aout)

X value:

Apply measurement to:

◆ Entire waveform

◇ Visible X and Y range only

Click here to display the Measurement Categories menu.

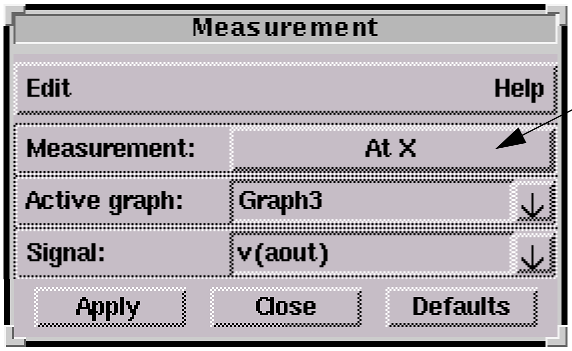
Measurement Categories.

Click on the appropriate choice to display the measurement operations for that category.

Selecting a Measurement

The Measurement field allows you to select the measurement operation.

Click the large **Measurement** button to expand the list of available measurements and select the one you want.



Measurement: At X

Active graph: Graph3

Signal: v(aout)

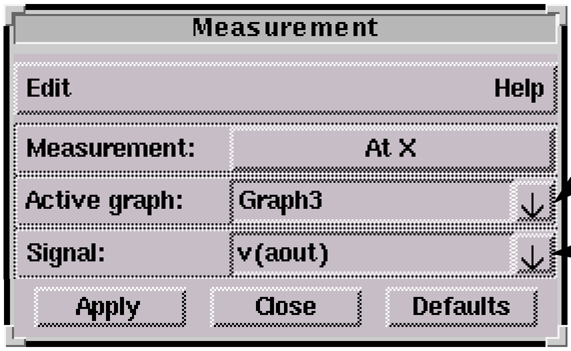
Click here to display the Measurement Categories.

Selecting a Signal for a Measurement

The Signal field allows you to select the signal from the active graph on which a measurement will be performed.

- Click on the associated arrow button to expand the list of available signals from the active graph and select the one you want, or you can click on a signal in the active graph.
- Click on the associated arrow button to expand the list of available graphs and select the one you want.

If more than one signal is selected on the active graph, the most recently selected signal is used for the measurement.



Measurement: At X

Active graph: Graph3

Signal: v(aout)

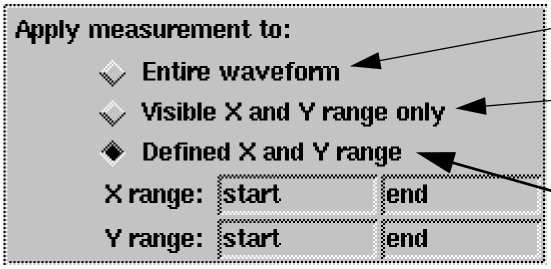
Click here to display the list of available graphs.

Click here to display the list of signals on the active graph or click on the desired signal in the active graph.

Setting the Range of a Measurement

The Apply Measurement to field provides two range choices for most measurements.

Choose one of the range choices for each measurement performed.



Apply measurement to:

◇ Entire waveform

◇ Visible X and Y range only

◆ Defined X and Y range

X range: start end

Y range: start end

The measurement is applied to the entire waveform, regardless of what is viewed.

The measurement is applied only to the the X and Y range that is displayed on the graph.

The measurement is applied only at the defined starting and ending X and Y range values.

Creating a New Waveform of Measurement Results

The result of some measurements produce other waveforms, especially with multi-member waveforms. The Create New Waveform in Active Graph field allows you the option of creating the new waveform in the active graph or creating a new graph to display the results.

1. Click on the downward pointing arrow to display your options.
2. Choose Active Graph or New Graph.

Note:

If any existing waveform is automatically updated due to an automatic plot action (such as Append) specified in an analysis, all measurement waveforms that depend on that waveform are updated at the same time.

Managing Measurement Results

The Measure Results dialog box displays measurement results, and manages the amount of data you view on a graph at one time.

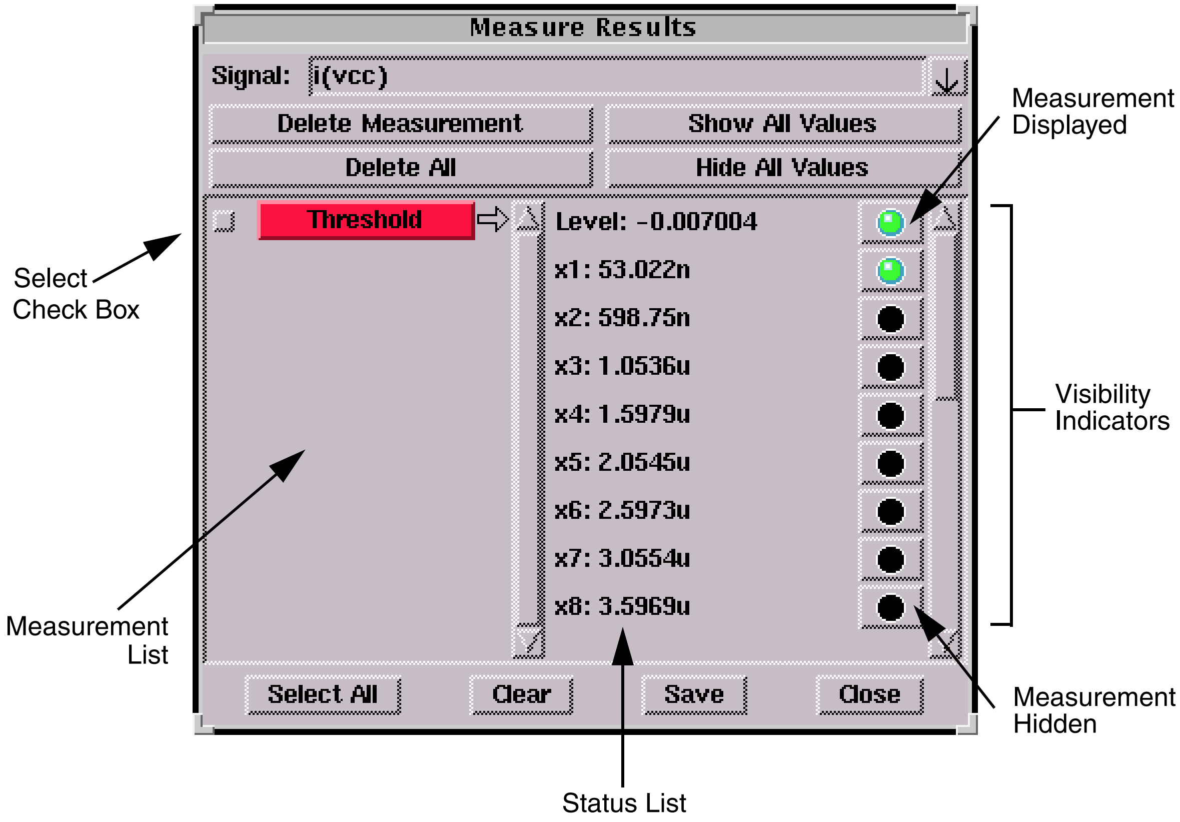
The Measure Results dialog box description is divided into the following topics:

- Accessing the Measurement Results Dialog Box
- Measurement List
- Status List
- Signal Field

Accessing the Measurement Results Dialog Box

There are various ways to open the Measure Results dialog box.

- Choose **Signal > Measure Results...** from the main menu.
- Double-click on any highlighted measurement value in a graph window.



Measure Results

Signal: i(vcc)

Threshold

```
Level: -0.007004
x1: 53.022n
x2: 598.75n
x3: 1.0536u
x4: 1.5979u
x5: 2.0545u
x6: 2.5973u
x7: 3.0554u
x8: 3.5969u
```

The **Delete Measurement** button permanently deletes the selected measurement.

The **Delete All** button permanently deletes all measurements for the selected signal.

The **Show All Values** button displays all values for the selected measurement.

The **Hide All Values** button hides the display of all values for the selected measurement. Passing the mouse cursor over a visibility indicator in the Status list will cause a highlighted measurement value to appear on the signal in the active graph.

The **Select All** button selects all measurements so information for them can be saved to a text file when the “Save” button is selected.

The **Clear** button deselects all the selected check boxes.

The **Save** button pops up a dialog window with information for the selected measurements. The **Save** button is located at the bottom of the window.

Clicking this button allows you to save information for the selected measurements to a text file.

The **Close** button closes the Measure Results dialog box.

Measurement List

The Measurement list displays all measurements that are active for a signal. Measurement values for the selected measurement are displayed in the Status list.

Status List

The Status list displays a list of measurement values associated with the selected measurement in the Measurement list. Each value has a visibility indicator associated with it.

A visibility indicator shows the display status of a measurement value.

- To change the state of the visibility indicator, single click the left mouse button.
- A black visibility indicator indicates that the measurement value is not displayed.
- A green visibility indicator shows that the measurement value is displayed.

Signal Field

The Signal field allows you to select any signal displayed in the graph window.

To display all available signals in the graph window, single click the down arrow at the right of the Signal field.

Even though you can select any signal in the graph window, the Measure Results dialog box does not show any data for signals that are not measured with the Measurement Tool.

Multi-Member Waveform Measurements

A multi-member waveform is created as a result of a Vary or Monte Carlo simulation. For example, if you are measuring the risetime of a multi-member waveform from a Monte Carlo analysis, you have the choice of measuring the risetime versus the run, the risetime plotted as a histogram, or both.

If you are measuring the risetime of a multi-member waveform from a Vary analysis, you have the choice of measuring the risetime versus the Vary parameter, the risetime plotted as a histogram, or both.

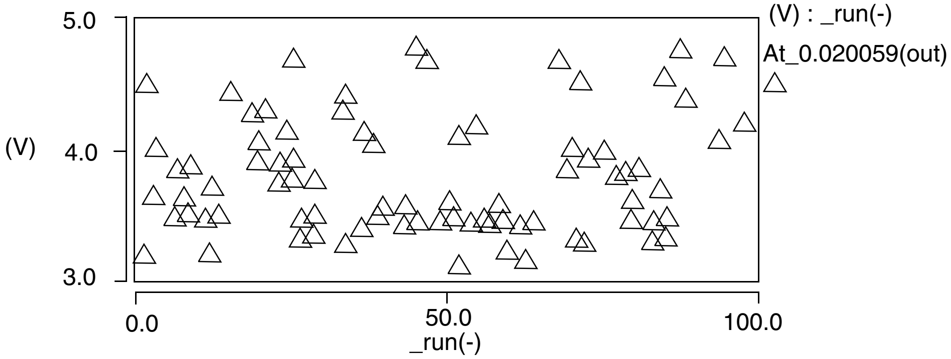
There can be other possible multi-member options provided in the Measurement dialog box depending on what type of measurement you are performing. Each of these options is described for each measurement operation.

Multi-member waveform measurements is further described in the following topics:

- Example of Creating a New Multi-Member Waveform
- Example of Creating a Multi-Member Histogram
- Multi-Member Count
- Multi-Member Count Example

Example of Creating a New Multi-Member Waveform

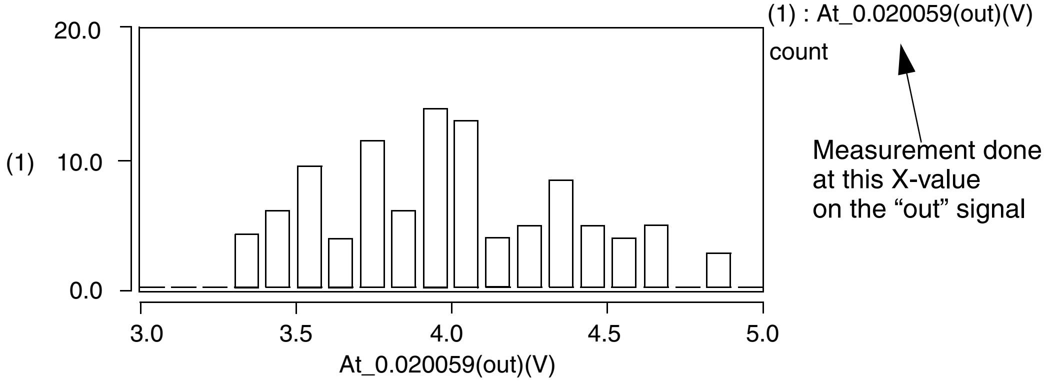
This scatter plot is generated by performing the At X measurement on a Monte Carlo generated, multi-member waveform that contains 100 members. In the Measurement dialog box, you should click on the **At X vs _run** button. Each particular Y-value (Y-axis) that occurs at the X-value of 0.020059 volts is plotted against the Monte Carlo run (X-axis) that generates the Y-value.



You can also choose any of the available parameters on the multi-member waveform, (as well as the index), to create a new waveform on the graph. Click the down arrow to the right of the At x vs. text box to open a list of parameters from which to choose.

Example of Creating a Multi-Member Histogram

This histogram is generated by performing the At X measurement on a multi-member waveform that contains 100 members. In the Measurement dialog box, you should click on the **At X** histogram button. The number of occurrences (Y-axis) is plotted against the Y-values (X-axis) found at the X-value of 0.020059 volts.

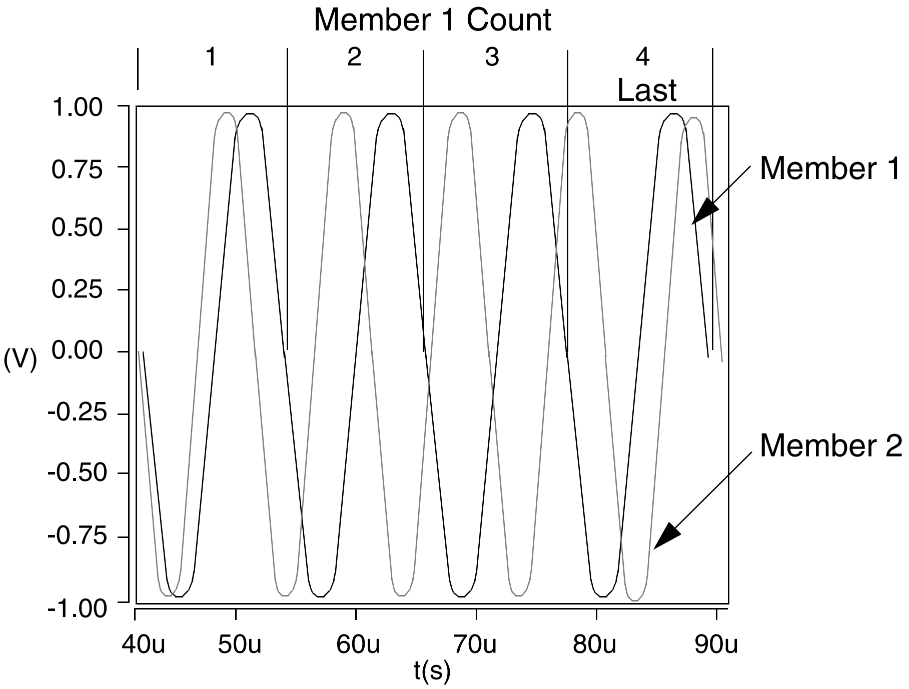


Multi-Member Count

You specify this location in the Measure Preferences dialog box, in the Multi-Member Count field. The waveform location is specified as a count value.

For an example, in the following figure, assume that a frequency measurement is applied to this waveform. This waveform has two members. Member 1 is shown divided into four counts, each representing periods for this measurement. If a measurement checks the rising edge, then there are four rising edge counts. Although not shown, Member 2 is also divided into counts.

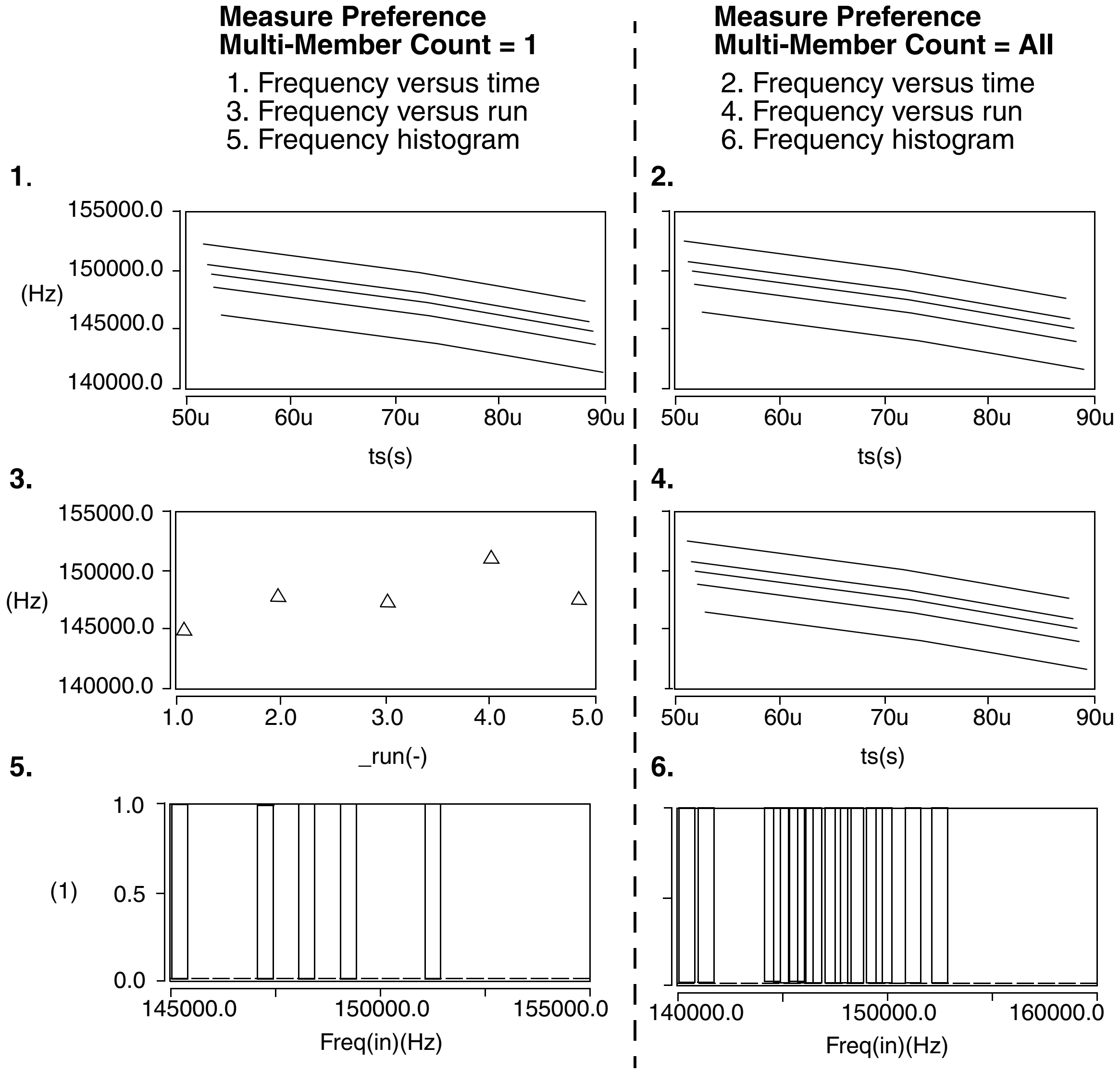
If you use the default count value of 1, the frequency measurement is applied to the first count of each member of the waveform. If you select Specify in the Multi-Member Count field, you can choose which waveform count the measurement will use. Specifying Last applies the measurement to the last count of each segment. If you specify All, the measurement calculation uses all the counts.



Multi-Member Count Example

This topic shows an example of how measurement results vary in response to changing the Multi-Member Count field in the Measure Preference dialog box. This example uses a frequency measurement.

The frequency measurement results shown in the example is for a multi-member waveform with five members, each having eight periods. The frequency of each member in this example is decreased over time. The example shows two measurements, each with three parts: the one on the left was made with the Measure Preferences dialog box, Multi-Member Count field set to 1. The measurement is made on the first period (or count) of each segment.



Measure Preference
Multi-Member Count = 1

1. Frequency versus time
3. Frequency versus run
5. Frequency histogram

Measure Preference
Multi-Member Count = All

2. Frequency versus time
4. Frequency versus run
6. Frequency histogram

Graph 1 shows how the frequency (Y-axis) of each member changes over time (X-axis). Graph 3 is a plot of the frequency (at count = 1) of each of the waveform members plotted against the individual simulation runs (X-axis). Graph 5 shows a histogram, which displays how many occurrences of each frequency value were encountered during all the simulation runs.Graphs 2, 4, and 6 are similar to their counterparts, but they represent all the counts of each waveform member.

Setting Measurement Preferences

Before running a measurement, you can preset Measurement Tool parameters by opening the Measure Preferences dialog box.

You can perform all your measurement operations without ever changing the default values. However, it might be helpful for you to be aware of what these settings are so that you can understand how the results are generated.

To access the Measure Preferences Dialog Box, choose **Edit > Preferences...** from the main menu, and click the Measurement tab in the Scope Preferences window that appears.

When you change parameters in the Measure Preferences dialog box, click the **Apply** button to cause the change to take affect on the next measurement and the **Save** button to take affect on subsequent invocations of the Measurement Tool.

The following table indicates Measure Preferences dialog box options. The bold choices in the table indicate the default settings.

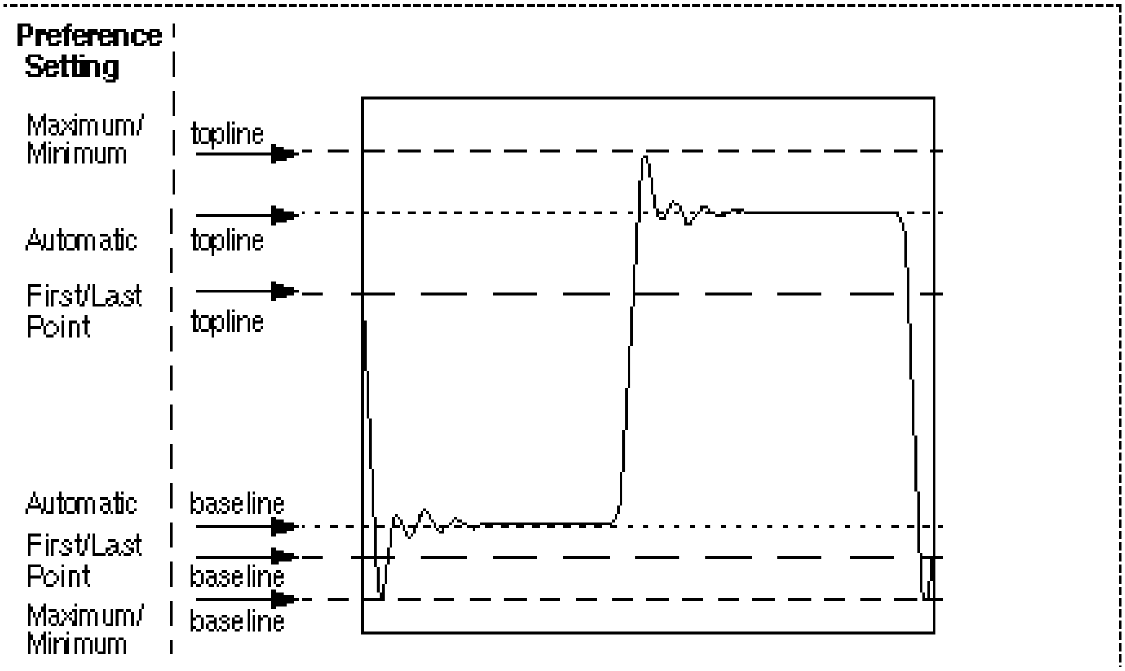
Parameter	**Choices**	**Description**
Display Precision	5 {a number}	Sets the precision of the numeric results displayed on the graph and in the Measurement Results dialog box.
Scatter Plots per Graph	1 - 10	Sets the Scatter Plots per Graph by clicking the up or down arrow to adjust the selection. (Note: this applies only to the Pareto measurement.)
Maximum Scatter Plots	Specified {a number}	Specifies the maximum number of scatter plots to be placed on the graph. (Note: this applies only to the Pareto measurement.)

Parameter	Choices	Description
Multi-Member Count	1	Specifies a location on a waveform where a measurement is taken. This parameter affects waveforms that meet the following criteria:
	All	Operates on a multi-member waveform or two waveforms of separate signals
	Last	Produces more than a single result (such as risetime, period, etc.)
	Specified {a number}	Generates a new waveform.
Histogram Bins	Specified {a number}	For histogram generation, this setting specifies the number of bins displayed. You can use the default value of 20 or specify your own integer.
Normalize Histogram	Selected/ Deselected	If selected, the histogram is normalized by dividing the total number of values into each bin count. The result is that each bin is assigned a value between 0 and 1, inclusive. This option is deselected by default.
Cumulative Histogram	Selected/ Deselected	If selected, the histogram is converted to a cumulative histogram by adding the count in each bin to the count of all preceding bins. This option is deselected by default.
Sort Result Waveforms	Selected/ Deselected	If selected, the result waveform is sorted if it is not monotonic. This option is deselected by default.
Default Topline/ Baseline	Automatic	When set to Automatic, the default is computed by using the method described in Topline/Baseline.
	Maximum/ Minimum	When set to Maximum/Minimum, the maximum and minimum points are used as the topline and baseline values.
	First/Last Point	When set to First/Last Point, the greater level of the first point and last point is used as the topline and the lower level of the first point and last point is used as the baseline.

Parameter	Choices	Description
Point to Point Measurement	X1, Y1	If selected, the Point to Point Measurement result includes the X and Y values for the first point by default.
	X2, Y2	If selected, the Point to Point Measurement result includes the X and Y values for the second point by default.
	DeltaX	If selected, the Point to Point Measurement result includes the X value difference between the two points by default.
	DeltaY	If selected, the Point to Point Measurement result includes the Y value difference between the two points by default.
	Length	If selected, the Point to Point Measurement result includes the length of the straight line that connects the two points by default.
	Slope	If selected, the Point to Point Measurement result includes the slope value of the straight line that connects the two points by default.
A2D and D2A Thresholds	Low (numeric)	The default low threshold value in Analog to Digital and Digital to Analog conversions.
	High (numeric)	The default high threshold value in Analog to Digital and Digital to Analog conversions.

Topline/Baseline Calculation

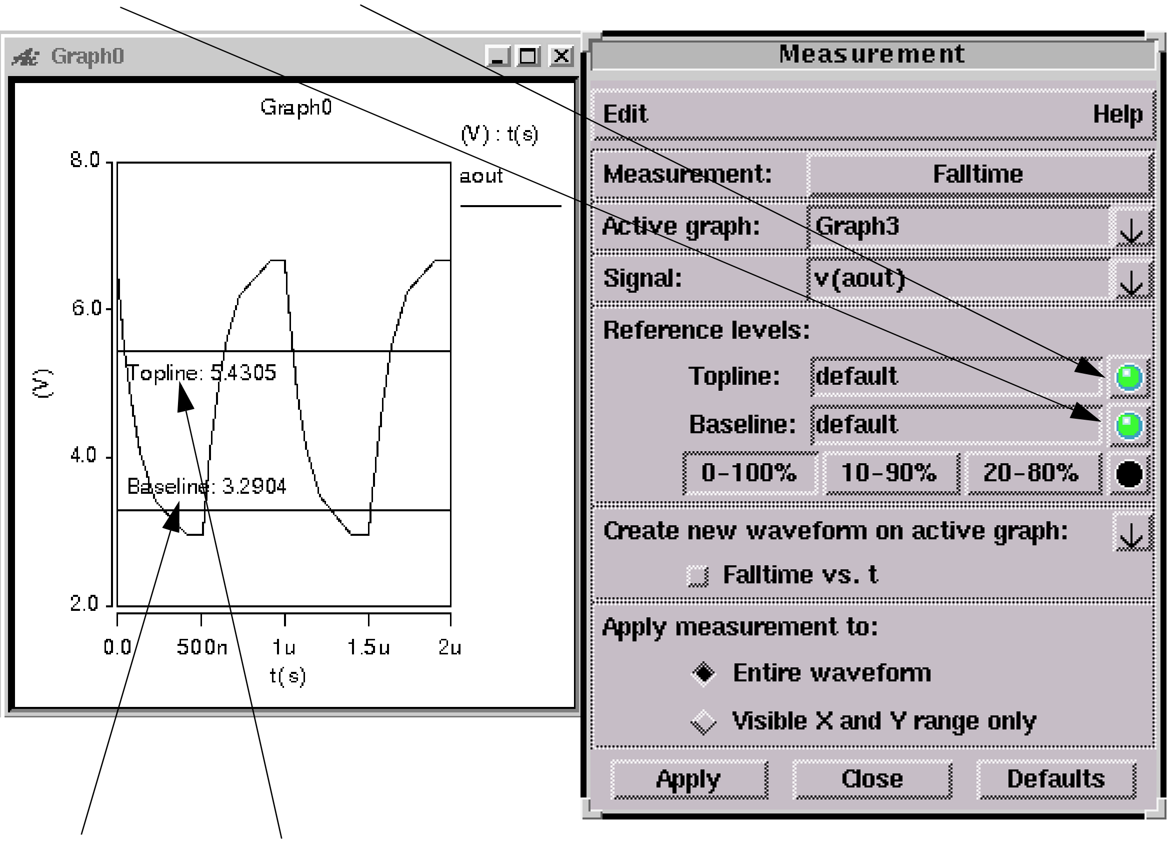
A number of measurements use either specified or default values for topline and baseline in their calculation. You set how your topline and baseline are calculated with the Measure Preferences dialog box. The following figure shows the various topline/baseline possibilities as set in the Measure Preferences dialog box.



Manually Set a Custom Topline/Baseline

There might be times when you want to perform a measurement using a custom top and/or base reference level. The following example shows how to change the topline and baseline references prior to computing a falltime measurement.

1. In the Measurement dialog box, turn on the Baseline and Topline indicator buttons.



Measurement: Falltime
Active graph: Graph3
Signal: v(aout)
Reference levels:
Topline: default ●
Baseline: default ●
Create new waveform on active graph:
□ Falltime vs. t
Apply measurement to:
◆ Entire waveform
◇ Visible X and Y range only

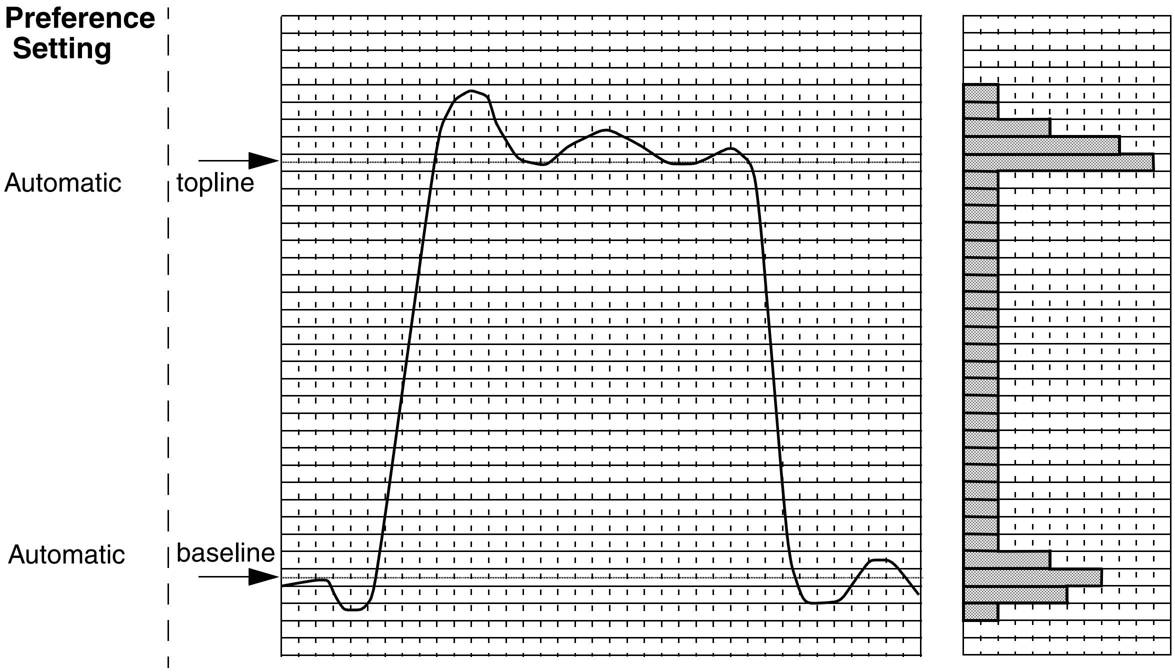
Baseline and Topline levels appear in the Graph window.

2. Move the cursor to Topline and left-click-and-hold to select it.
3. Move the Topline to the desired value. Repeat for the Baseline.

When using this procedure to change the Topline/Baseline levels, leave the Measurement Reference Level set to the 0-100% setting so that the measurement is based on the values of Topline and Baseline that you just set.

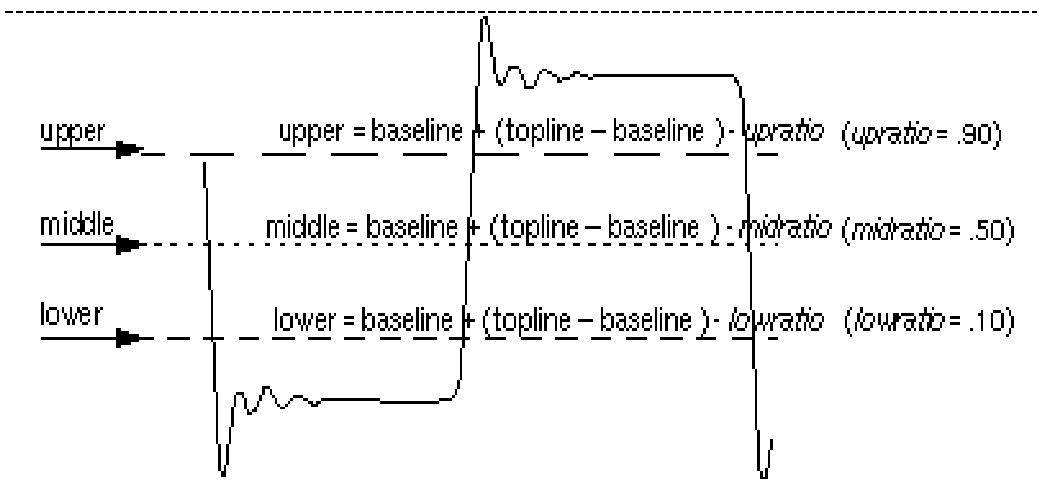
Default Calculation

If you use the default setting of Automatic in the Measure Preferences dialog box, the baseline and topline levels are calculated by using a probability density histogram method. The waveform is sampled at a number of equally spaced points along the X-axis. The Y-axis is then equally divided into a number of bins, and the number of sampled points that fall into each bin is determined. The Y-axis bin that contains the most points above the midpoint is used as the topline level. The Y-axis bin that contains the most points below the midpoint is used as the baseline level. This procedure is shown graphically in the following figure with the probability density histogram.



Waveform Reference Levels

In addition to the topline/baseline and maximum/minimum reference levels, other levels are calculated and used by various measurements as shown below.


upper
upper = baseline + (topline – baseline) - upratio (upratio = .90)
middle
middle = baseline + (topline – baseline) - midratio (midratio = .50)
lower
lower = baseline + (topline – baseline) - lowratio (lowratio = .10)

The reference levels upper, middle, and lower correspond to the distal, mesial, and proximal reference levels defined in the IEEE standard Pulse Terms and Definitions (IEEE Std 194-1977).

AC Coupled RMS

Description

Displays the RMS value of the AC component of a waveform.

Type of Measured Waveform

- Analog, event-driven analog

AC Coupled RMS Calculation

The AC coupled RMS value of a waveform is calculated as follows:

$$\left[\frac{1}{(x2-x1)}\int_{x1}^{x2}(W-\overline{W})^{2}dx\right]^{1/2}$$

In this calculation, W represents the waveform, $\overline{W}$ is its average value, and x1 and x2 are the starting and ending points for the waveform.

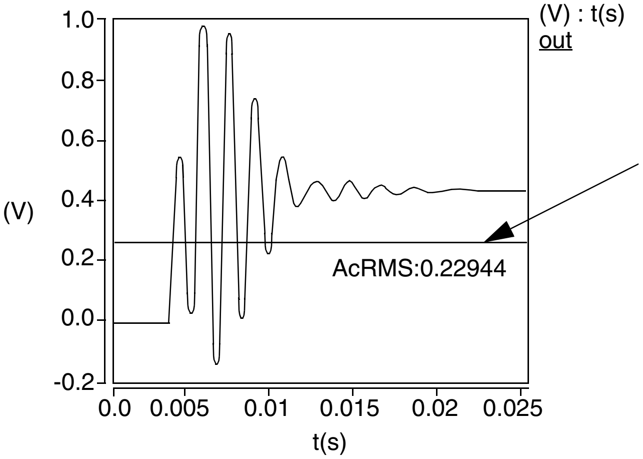
Command Group

Levels

Syntax

None

Example



Measurement shown as a non-moveable line and moveable text.

Graphical Interface Description

Dialog Box Fields

Category List

All Levels category items appear below the Signal field. Select the AC Coupled RMS item and any other items you want to measure.

Multi-Member Waveform Options

Create New Waveform on Active Graph or New Graph

Creates a new waveform in the Active Graph or in a New Graph to display the results.

Measurement vs. _run or vary_parameter -
Creates a scatter plot or analog waveform with the current measurement values (Y-axis) versus each Monte Carlo run (X-axis) or Vary parameter value (X-axis).

Measurement Histogram -
Creates a histogram with a count (Y-axis) of the current measurement value (X-axis) occurrences.

ACPR

Description

Displays the ACPR_upper and ACPR_lower ratios of an envelope power spectrum waveform. ACPR is "Adjacent Channel Power Ratio."

Type of Measured Waveform

- Analog real type envelope power spectrum waveform.

ACPR Calculation

ACPR calculations are performed in the following manner:

$$\mathrm{ACPR}_{\mathrm{UPPER}} = 10\log\frac{\int_{f2}^{f_3}\tilde{P}(f)df}{\int_{f_0}^{f_1}\tilde{P}(f)df}$$

$$\mathrm{ACPR}_{\mathrm{LOWER}} = 10\log\frac{\int_{f4}^{f5}\tilde{P}(f)df}{\int_{f_0}^{f_1}\tilde{P}(f)df}$$

ACPR measurements are in db units.

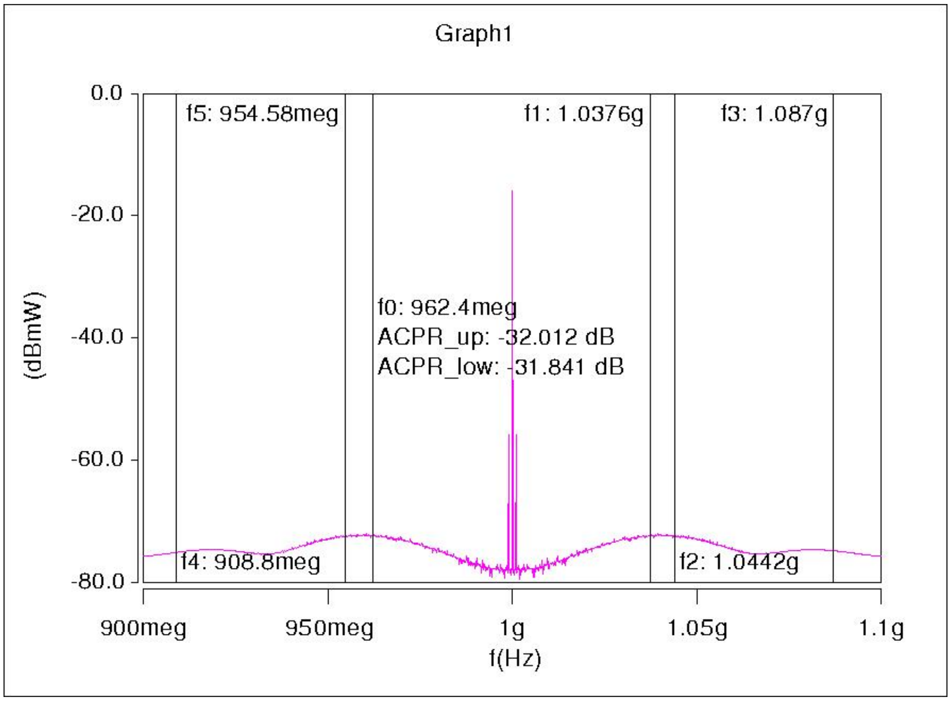
Command Group

RF

Syntax

None

Example



Graphical Interface Description

Dialog Box Fields

The following fields are optional. If no value is specified, the default value is used.

f0	minimum main channel frequency
f1	maximum main channel frequency
f2	minimum upper channel frequency
f3	maximum upper channel frequency
f4	minimum lower channel frequency
f4	maximum lower channel frequency

Amplitude

Description

Displays the amplitude of a waveform.

Type of Measured Waveform

- Analog, event-driven analog

Amplitude Calculation

The amplitude is calculated as the difference between the topline and the baseline reference levels.

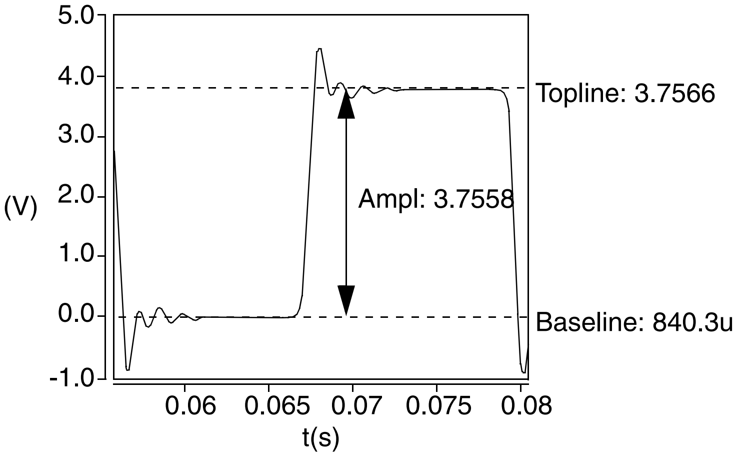
Command Group

Levels

Syntax

None

Example



Graphical Interface Description

Dialog Box Fields

Category List	All Levels category items appear below the Signal field. Select the Amplitude item and any other items you want to measure.

At X

Description

Displays the Y-axis value at a particular X-axis point on a waveform.

Type of Measured Waveform

- Analog, event-driven analog, digital, scatter plot, spectral, histogram, bus

Command Group

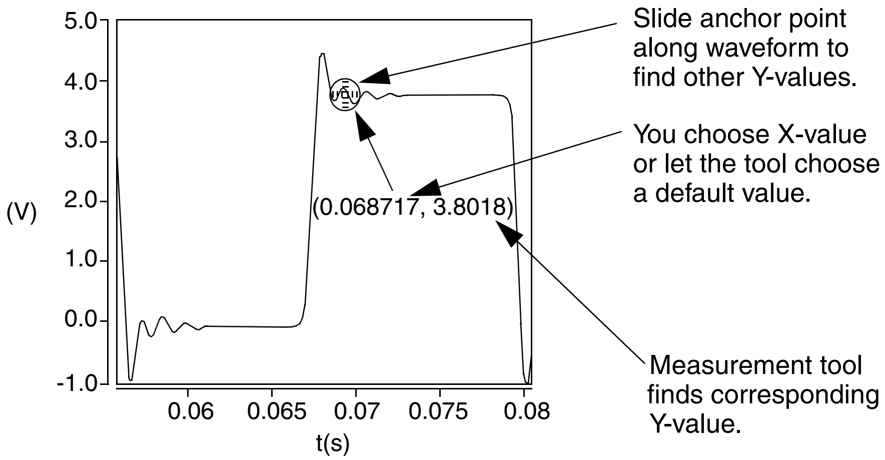
General

Syntax

None

Example

Example 1



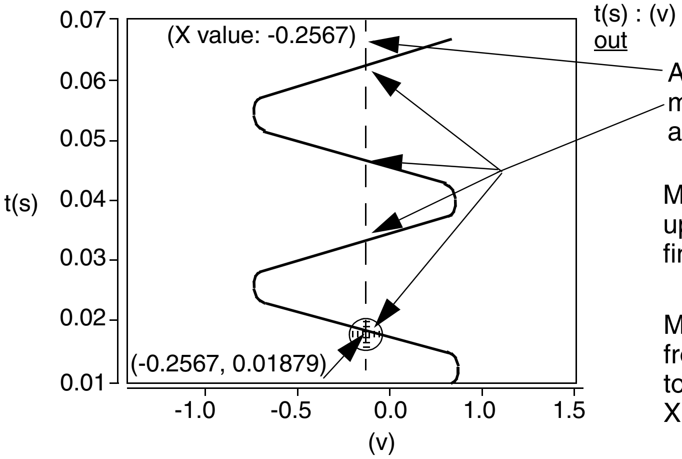
Slide anchor point along waveform to find other Y-values.

You choose X-value or let the tool choose a default value.

Measurement tool finds corresponding Y-value.

Example 2

This example shows a non-monotonic signal. The measurement result can display multiple Y-values for a given X-value with this type of signal.



At this X-value, multiple Y-values are found.

Move anchor point up and down to find other Y-values.

Move vertical line from side-to-side to specify alternate X-value.

Graphical Interface Description

X Value — Optional. You can provide an X-value and the tool will provide the Y-value at that coordinate. If you do not specify the X-value, a default is used.

Average

Description

Displays the average value of a waveform.

Type of Measured Waveform

- Analog, event-driven analog

Average Calculation

The average value of a waveform is calculated as follows:

$$\frac{1}{(x2-x1)}\int_{x1}^{x2} Wdx$$

W represents the waveform, and x1 and x2 are the starting and ending points for the waveform.

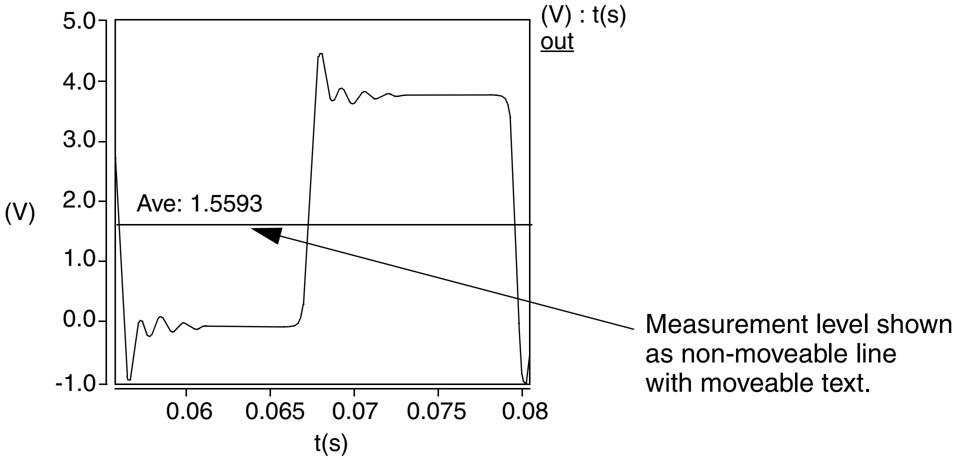
Command Group

Levels

Syntax

None

Example



Graphical Interface Description

Dialog Box Fields

Category List	All Levels category items appear below the Signal field. Select the item(s) you want to measure.

Bandwidth

Description

Displays the bandwidth, the low, high, or center frequency, or the level at which the measurement is made for a bandpass-shaped waveform. The measurement is made relative to a default or specified topline level and a specified offset.

Type of Measured Waveform

- Analog, spectral

Possible Errors

An error is reported if the first or last data point is above the measurement level.

Bandwidth Calculations

The following subtopics describe the calculations to determine the bandwidth of a waveform:

Topline

If you do not specify the topline, a default value is calculated by using a method specified in the Default Topline/Baseline field in the Measurement Preference dialog box.

Offset

Computed as one of
topline - offset_value,
topline + offset_value,
topline * offset_value, or
topline / offset_value,

The resulting offset (measurement) level is used to determine the bandwidth measurement as follows: (Also see the example.)

- fLow (frequency-low) is the first point that falls below the measurement level, before the maximum point. The fLow point can be shown on the graph using the Measure Results dialog box.
- fHigh (frequency-high) is the first point that falls below the measurement level after the maximum point. The fHigh point can be shown on the graph using the Measure Results dialog box.
- fCenter (frequency-center) is calculated as $\sqrt{\mathrm{fHigh} \cdot \mathrm{fLow}}$
  The fCenter point can be shown on the graph using the Measure Results dialog box.

Bandwidth

Calculated as fHigh – fLow.

Q (quality factor)

Calculated by dividing fCenter (frequency-center) by the bandwidth.

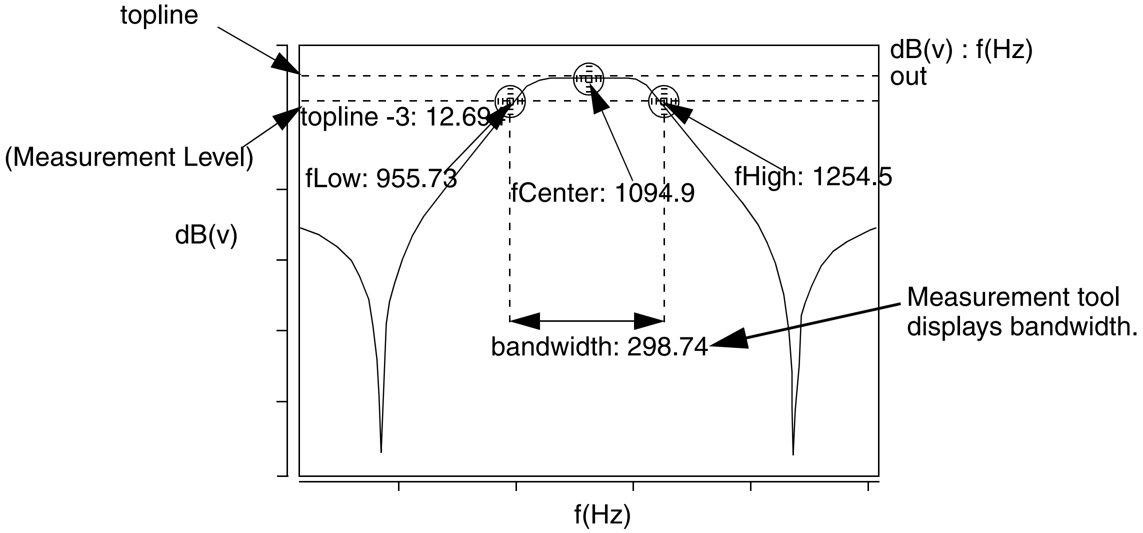
Command Group

Frequency Domain

Syntax

None

Example



Graphical Interface Description

Dialog Box Fields

Reference Levels

If you want to see the topline and/or offset level displayed on the waveform, you click on the Visibility Indicator to the right of the Topline or Offset field.

Topline

You set this field to a default or a specified level.

Offset

You specify an offset value, to be applied relative to the Topline value. The default is 3. You must also choose which operator to use (–, +, *, or /) along with the specified level. The default is the minus sign. This resulting level is also called the measurement level.

Bandwidth Multi-Member Waveform Options

Create New waveform on Active Graph or New Graph

Creates a new waveform in the Active Graph or in a New Graph to display the results.

Bandwidth vs. _run or vary_parameter - Creates a scatter plot or analog waveform with Bandwidth values (Y-axis) versus each Monte Carlo run (X-axis) or Vary parameter value (X-axis).

fLow vs. _run or vary_parameter -
Creates a scatter plot or analog waveform with low frequency values (Y-axis) versus each Monte Carlo run (X-axis) or Vary parameter value (X-axis).

fCenter vs. _run or vary_parameter -
Creates a scatter plot or analog waveform with center frequency values (Y-axis) versus each Monte Carlo run (X-axis) or Vary parameter value (X-axis).

fHigh vs. _run or vary_parameter -
Creates a scatter plot or analog waveform with high frequency values (Y-axis) versus each Monte Carlo run (X-axis) or Vary parameter value (X-axis)

Bandwidth Histogram - Creates a histogram with a count (Y-axis) of bandwidth value (X-axis) occurrences.

fLow Histogram - Creates a histogram with a count (Y-axis) of fLow value (X-axis) occurrences.

fCenter Histogram - Creates a histogram with a count (Y-axis) of fCenter value (X-axis) occurrences.

fHigh Histogram - Creates a histogram with a count (Y-axis) of fHigh value (X-axis) occurrences.

Baseline

Description

Displays the baseline level of a waveform as determined by the Measure Preferences Default Topline/Baseline setting.

Type of Measured Waveform

- Analog, event-driven analog

Baseline Calculation

The baseline calculation method is determined by the Default Topline/Baseline field in the Measurement Preference dialog box, which you can set before performing a measurement. If you do not set it, the default method is used.

Command Group

Levels

Syntax

None

Graphical Interface Description

Dialog Box Fields

Category List	All Levels category items appear below the Signal field. Select the Baseline item and any other items you want to measure.

Cpk

Description

Displays an indicator of the process capability for a waveform relative to specified upper and lower limits.

Type of Measured Waveform

- Scatter plot

CPK Calculation

$$\frac{(\text{upper-mean})}{3\ (\text{stddev})} \quad \text{or} \quad \frac{(\text{mean-lower})}{3\ (\text{stddev})}$$

In this calculation, mean represents the mean value of the scatter plot, upper and lower represent the specification limits you specify, and stddev represents the standard deviations of the scatter plot. When both the upper-mean and lower-mean values are provided, the smaller result of these two calculations is displayed as the measurement.

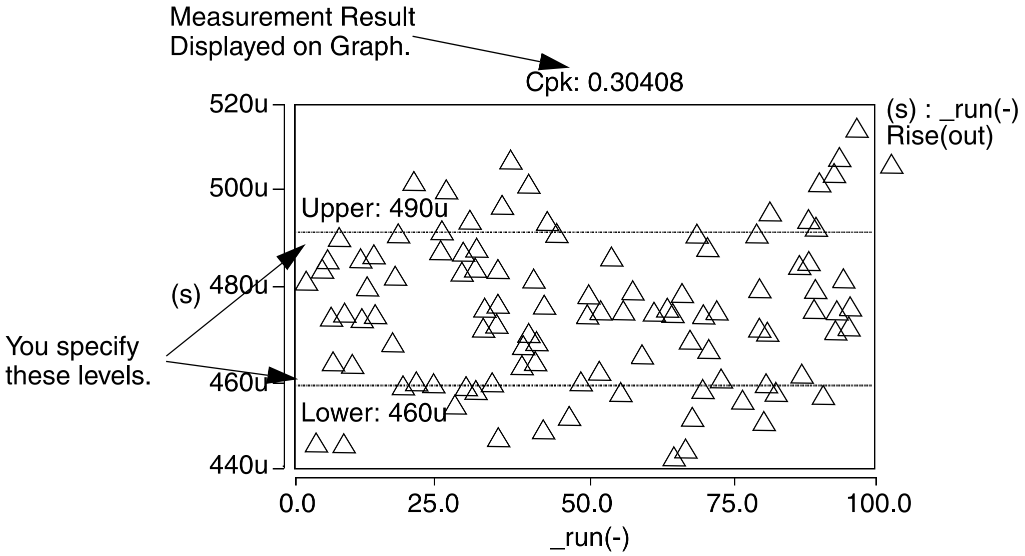
Command Group

Statistics

Syntax

None

Example



Graphical Interface Description

Category List	All Statistic category items appear below the Signal field. Select the Cpk item and any other items you want to measure.
Specification Limit	Required values you provide.
Upper	Specifies the upper specification limit.
Lower	Specifies the lower specification limit.

Crossing

Description

Displays the crossing (intersection) points of two waveforms.

Type of Measured Waveform

- Analog, event-driven analog (The two signals do not need to be the same type.)

Crossing Calculation

The particular crossing displayed is determined by the Multi-Member Count setting in the Measure Preferences dialog box and by the Slope Trigger field in the Measurement dialog box. If you need more information on the count setting, refer to the Multi-Member Count Example.

The slope of the crossing(s) to be selected can be designated using the Slope Trigger field in the Measurement dialog box. A positive crossing is one where the difference in the slope of the measured waveform and the slope of the reference waveform is greater than 0. A negative crossing is one where the difference of the slopes is less than 0.

Command Group

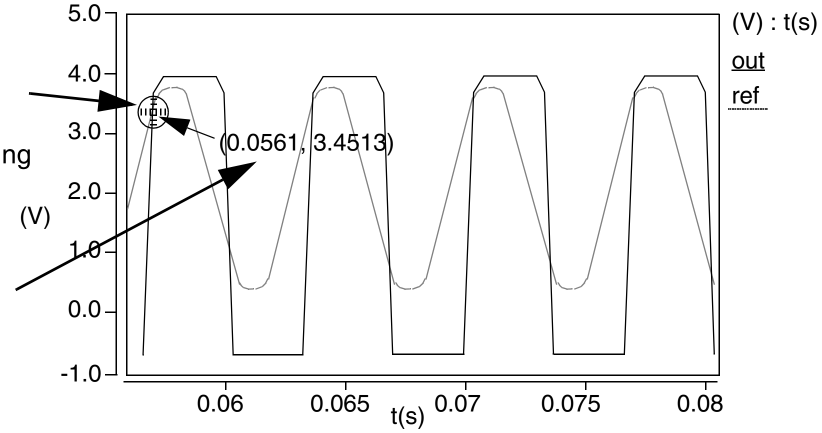
General

Syntax

None

Example

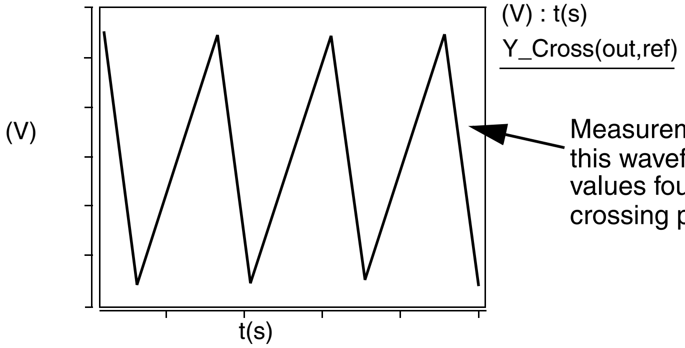
Example 1



Move anchor point along waveform to display other crossing points.

Measurement tool displays x/y coordinates at crossing point.

Example 2



Measurement tool produces this waveform of all Y-axis values found at the crossing points.

Graphical Interface Description

Dialog Box Fields

Signal and Ref. Signal — You specify the name of the signal as in all waveform measurements, but in addition, you specify a reference signal. By using the **Swap** button, you can alternate between the two signals to determine which one becomes the reference.

Slope Trigger	either - Measurement finds all crossings regardless of the slopes.
	positive - Measurement finds crossings where the slope difference between the two waveforms is positive.
	negative - Measurement finds crossings where the slope difference between the two waveforms is negative.
Create New Waveform on Active Graph or New Graph	X at Crossing vs. {X-axis} - Creates a new waveform with the X-values (Y-axis) of the crossing points versus the X-axis parameter.
	Y at Crossing vs. {X-axis} - Creates a new waveform with the Y-values (Y-axis) of the crossing points versus the X-axis parameter as in Example 2.

Multi-Member Waveform Options

Create New Waveform on Active Graph or New Graph	X at Crossing vs. _run or vary_parameter - A new waveform is computed with X-axis values (Y-axis) versus each run (X-axis) or vary parameter (X-axis).
	Y at Crossing vs. _run or vary_parameter - A new waveform is computed with Y-axis values (Y-axis) versus each run (X-axis) or vary parameter (X-axis).
	X at Crossing Histogram - A histogram is computed with X-axis values (Y-axis) versus each run (X-axis).
	Y at Crossing Histogram - A histogram is computed with Y-axis values (Y-axis) versus each run (X-axis).

Damping Ratio

Description

Displays the damping ratio of a point on a waveform.

Type of Measured Waveform

- Pole zero data, analog (must be complex)

Damping Ratio Calculation

The damping ratio of a waveform is calculated as -real(magnitude(value)).

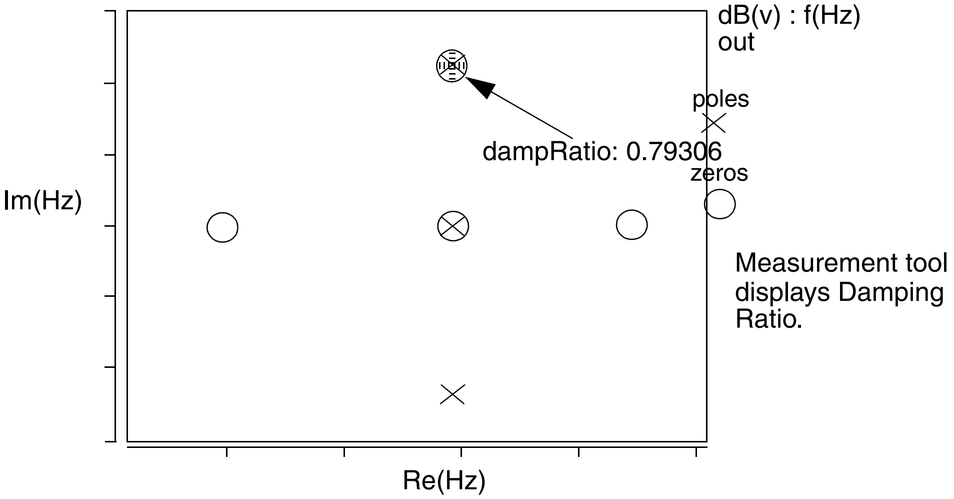
Command Group

s Domain

Syntax

None

Example



Measurement tool displays Damping Ratio.

Graphical Interface Description

Dialog Box Fields

X Value — Optional. You can provide an X-value and the tool will provide the Y-value at that coordinate. If you do not specify the X-value, a default is used.

dB

Description

Displays the dB value on a point on a waveform.

Type of Measured Waveform

- Analog (must be complex)

dB Calculation

The dB of a point is calculated by returning 20(log) of the absolute value of the point on the waveform.

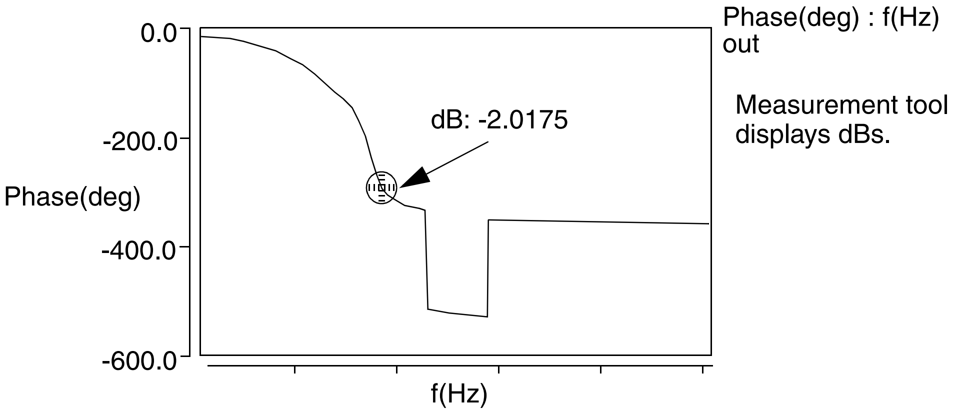
Command Group

Frequency Domain

Syntax

None

Example



Measurement tool displays dBs.

Graphical Interface Description

Dialog Box Fields

X Value	Optional. You can provide an X-value and the tool will provide the Y-value at that coordinate. If you do not specify the X-value, a default is used.

Delay

Description

Displays the delay between the edges of two waveforms relative to default or specified topline and baseline levels for both the measured waveform and the reference waveform. The two waveforms do not need to be the same type, but they must be in the same graph region. It is assumed that the rising or falling edge on the reference waveform causes the corresponding (rising or falling) edge on the measured waveform so that the reference edge occurs before the measured edge.

Type of Measured Waveform

- Analog, event-driven analog, digital

Delay Calculation

All rising or falling edges for the measured waveform are determined based on the Trigger setting in the Measure dialog box. From each edge, the corresponding previously-occurring edge on the reference waveform is determined. The difference on the X-axis between the two edges is the delay time.

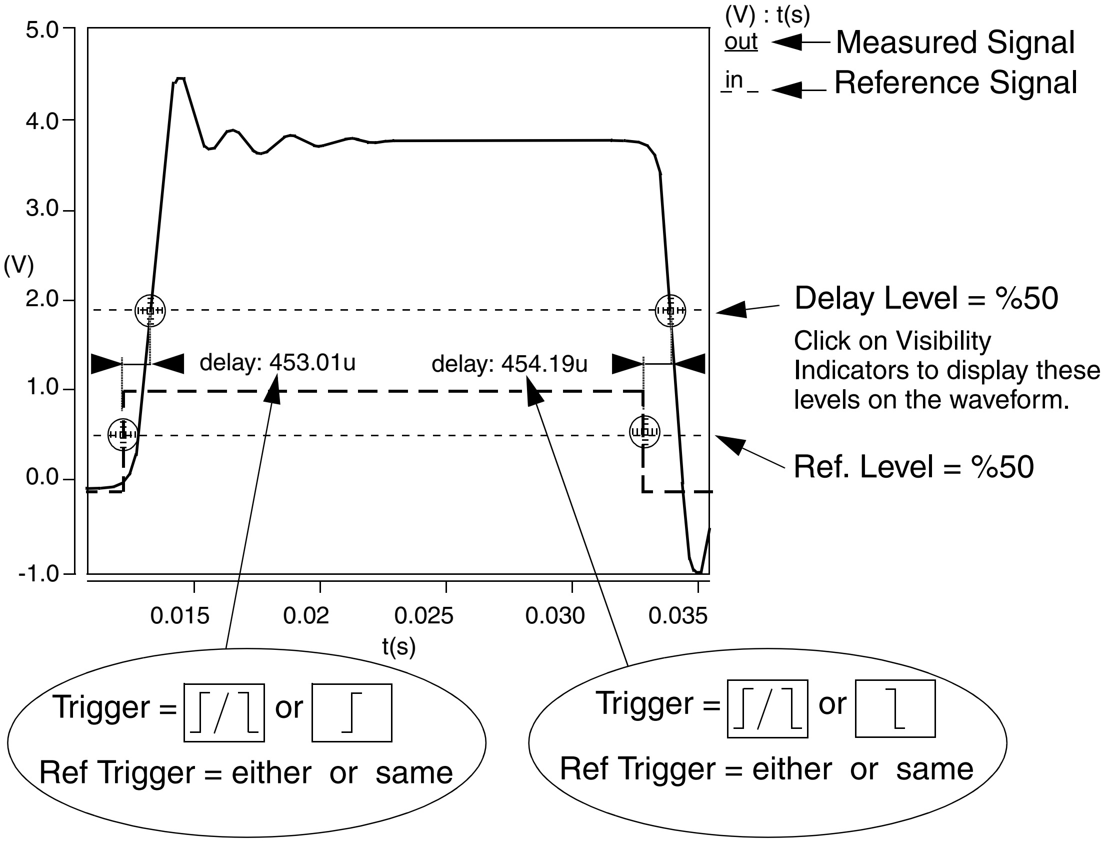
Command Group

Time Domain

Syntax

None

Example



Graphical Interface Description

Dialog Box Fields

Signal and Ref. Signal — You specify the name of the signal as in all waveform measurements, but, in addition, you specify a reference signal. By using the **Swap** button, you can alternate between the two signals to determine which one becomes the reference.

Reference Levels

The following four fields set the topline and baseline levels for the measured signal and the corresponding reference signal. You can display any of these four levels on the waveform by clicking on the corresponding Visibility Indicator to the right of each field.

Topline

Specify a topline value or use the default value.

Baseline

Specify a baseline value below the topline value or use the default value.

Ref. Topline

Same as Topline, but it applies to the reference signal.

Ref. Baseline

Same as Baseline, but it applies to the reference signal.

Delay Level

To set the waveform level where the delay is calculated on the measured signal, click on 10%, 50%, or 90%. See the Example.

Ref. Level

Click on 10%, 50%, or 90% to set the waveform level where the delay is calculated on the reference signal. See the Example.

Trigger

Specifies that the measurement starts from either a rising or falling edge.

Specifies that the measurement starts from a rising edge.

Specifies that the measurement starts from a falling edge.

Ref. Trigger	either - Selecting this button causes the measurement to trigger on the first edge of the reference waveform prior to the specified measured waveform edge, be it a positive or a negative edge. same - The measurement triggers on the reference waveform at the previous edge with the same polarity (set in the Trigger field) as the measured waveform. opposite - The measurement triggers on the reference waveform at the previous edge with the opposite polarity (set in the Trigger field) as the measured waveform. See the example.
Create New Waveform on Active Graph or New Graph	Delay vs. t - Creates a new waveform with the delay values (Y-axis) versus time (X-axis).

Delta X

Description

Displays the X-value difference between two Y-axis points on one or two waveforms. If two waveforms are selected, the two waveforms do not need to be the same type, but they must be in the same graph region.

Type of Measured Waveform

- Analog, event-driven analog, scatter plot

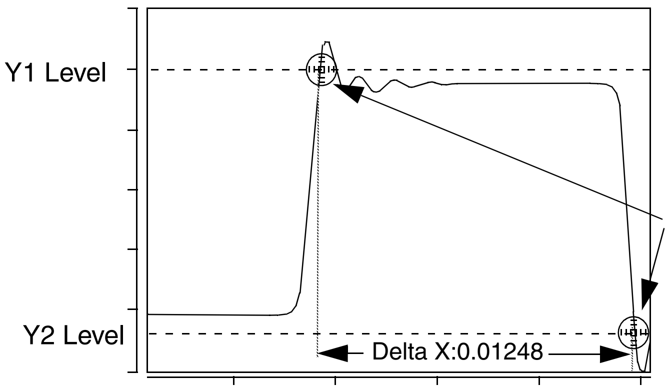
Command Group

General

Syntax

None

Example



Example measure showing anchor points that snap to “Signal.”

Slide anchor points along waveform to display other Delta-X values.

Graphical Interface Description

Dialog Box Fields

Anchor Snap — Signal - Generates moveable anchor points that snap to the signal waveform. See the Example.

Closest Measurement - When you have multiple anchor points visible on a graph from previous measurements, this setting causes the measurement to snap to one of those nearest points. If there is no visible measurement on the active graph from which to snap, an error message is displayed. Refer to the Point Marker Example.

Floating - Generates moveable anchor points that can be positioned anywhere within the graph. Refer to the Delta Y Example.

Location (Optional) — You can optionally specify two Y-values that are used to determine the X-value difference. See the Example.

Y1 Value — Optional value. You can change it by moving the anchor point after the initial measurement is made.

Y2 Value — Optional value. You can change it by moving the anchor point after the initial measurement is made.

Lock Vertical

An optional check box. If this is checked, the distance of “Y1 Value” and “Y2 Value” will be fixed, so the two anchors of the DeltaX measurement will move together.

Delta Y

Description

Displays the Y-value difference between two X-axis points on one or two waveforms. If two waveforms are selected, the two waveforms do not need to be the same type, but they must be in the same graph region.

Type of Measured Waveform

- Analog, event-driven analog, scatter plot

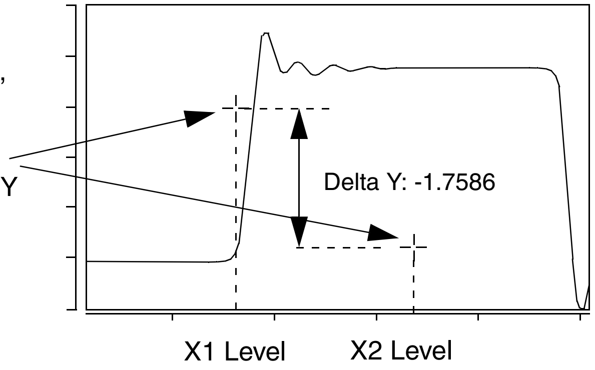
Command Group

General

Syntax

None

Example



Example measure shows anchor points that snap to “Floating.”

Move anchor points to display other Delta-Y values.

Graphical Interface Description

Dialog Box Fields

Anchor Snap	Signal - Generates moveable anchor points that snap to the signal waveform. To see an example of this type of snap, refer to the Delta X Example. Closest Measurement - When you have multiple anchor points visible on a graph from previous measurements, this setting causes the measurement to snap to one of those nearest points. If there is no visible measurement on the active graph from which to snap, an error message is displayed. Refer to the Point Marker Example. Floating - Generates moveable anchor points that can be positioned anywhere within the graph. See the example.
Location (Optional)	You can optionally specify two X-values that are used to determine the Y-value difference. See the Example.
X1 Value	Optional value. You can change it by moving the anchor point after the initial measurement is made.
X2 Value	Optional value. You can change it by moving the anchor point after the initial measurement is made.

Dpu

Description

Displays the number of defects per unit of a scatter plot waveform.

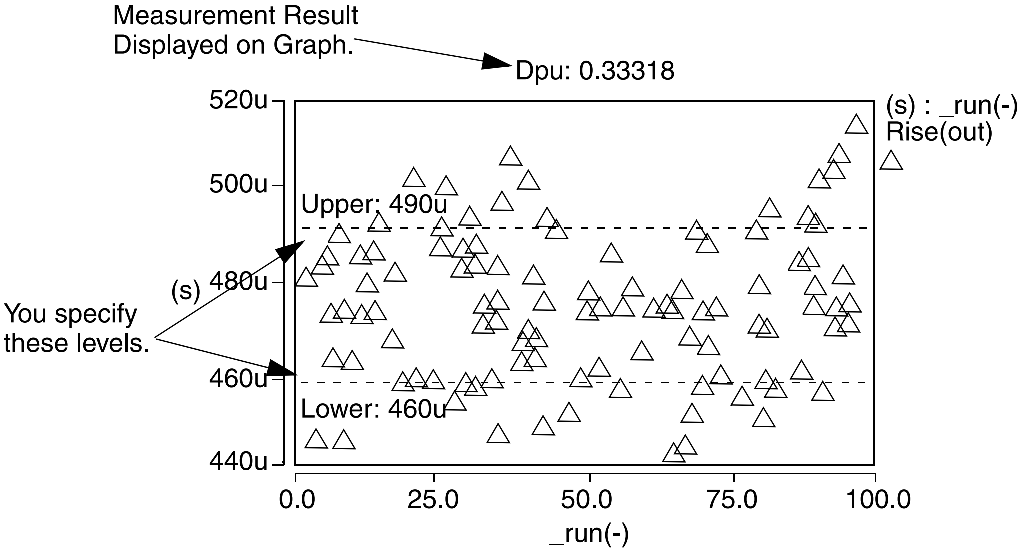
Type of Measured Waveform

- Scatter plot

Command Group

Statistics

Example



Graphical Interface Description

Dialog Box Fields

Category List	All Statistic category items appear below the Signal field. Select the Dpu item and any other items you want to measure.
Specification Limits	Required values you provide.
Upper	Specifies upper measurement limit.
Lower	Specifies lower measurement limit.

Duty Cycle

Description

Displays the duty cycle of a periodic waveform relative to default or specified topline and baseline levels.

Type of Measured Waveform

- Analog, event-driven analog, digital

Possible Errors

An error is reported if the waveform does not contain at least one complete cycle.

Duty Cycle Calculation

The duty cycle is calculated as the ratio of the “high” portion of the waveform to the length of the period. In the example, the duty cycle is t1/period. (The circled portion of the waveform in the example is considered “high” and does not influence the calculation since it does not fall below the lower level of the waveform.)

Further details of how the period is measured can be found in the description for the Period measurement. More information about reference levels is provided in “Waveform Reference Levels”.

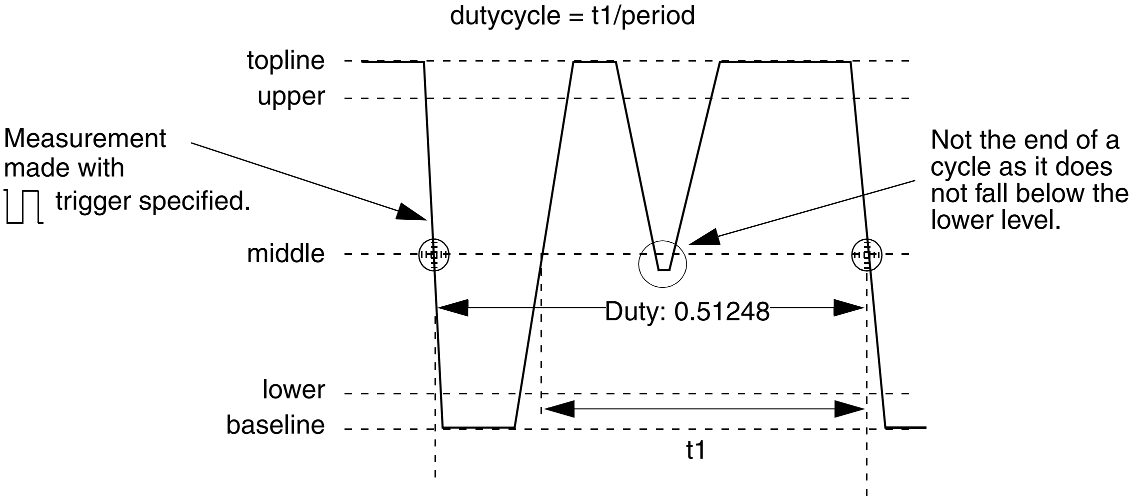
Command Group

Time Domain

Syntax

None

Example



Graphical Interface Description

Dialog Box Fields

Reference Levels

The following fields set the topline and baseline levels for the measured signal. You can display either of these levels on the waveform by clicking on the Visibility Indicator at the right of each field.

Topline

Specify a topline value within the upper and lower Y-axis values, or use the default value.

Baseline

Specify a baseline value below the topline value that is within the Y-axis values, or use the default value.

Trigger

⊓⊔/⊔⊓ Specifies that the measurement starts from a period with either a rising or falling edge.

⊓⊔ Specifies that the measurement starts from a period with a rising edge.

⊔⊓ Specifies that the measurement starts from a period with a falling edge.

Create New Waveform on Active Graph or New Graph

Duty Cycle vs. t - A new waveform is computed with duty cycle (Y-axis) versus time (X-axis).

Eye Diagram

Description

An eye diagram is used to display the behavior of a waveform cycle during a specific period of time. The eye diagram Measurement dialog has the ability to effectively overlap periods of time within a specified periodic waveform.

This type of periodic waveform display provides signal analysis characteristics much like the measurable variations found in periodicity or jitter output.

Two options are available to apply an Eye Diagram measurement:

- Searching the reference signal for points that contain a y value equal to the trigger value, getting the corresponding x values, and using the x values as the start x values. The time base value becomes the X axis length, and the the input signal is cut into multiple segments. This is the default trigger Eye Diagram.
- Specifying the start x value, end x value, and the time base, the measurement folds the input signal into multiple segments from the start x value to end x value. The X axis length of each segment is equal to the value of time base. This is called an ideal trigger Eye Diagram.

Type of Measured Waveform

- Analog

Command Group

Time Domain

Syntax

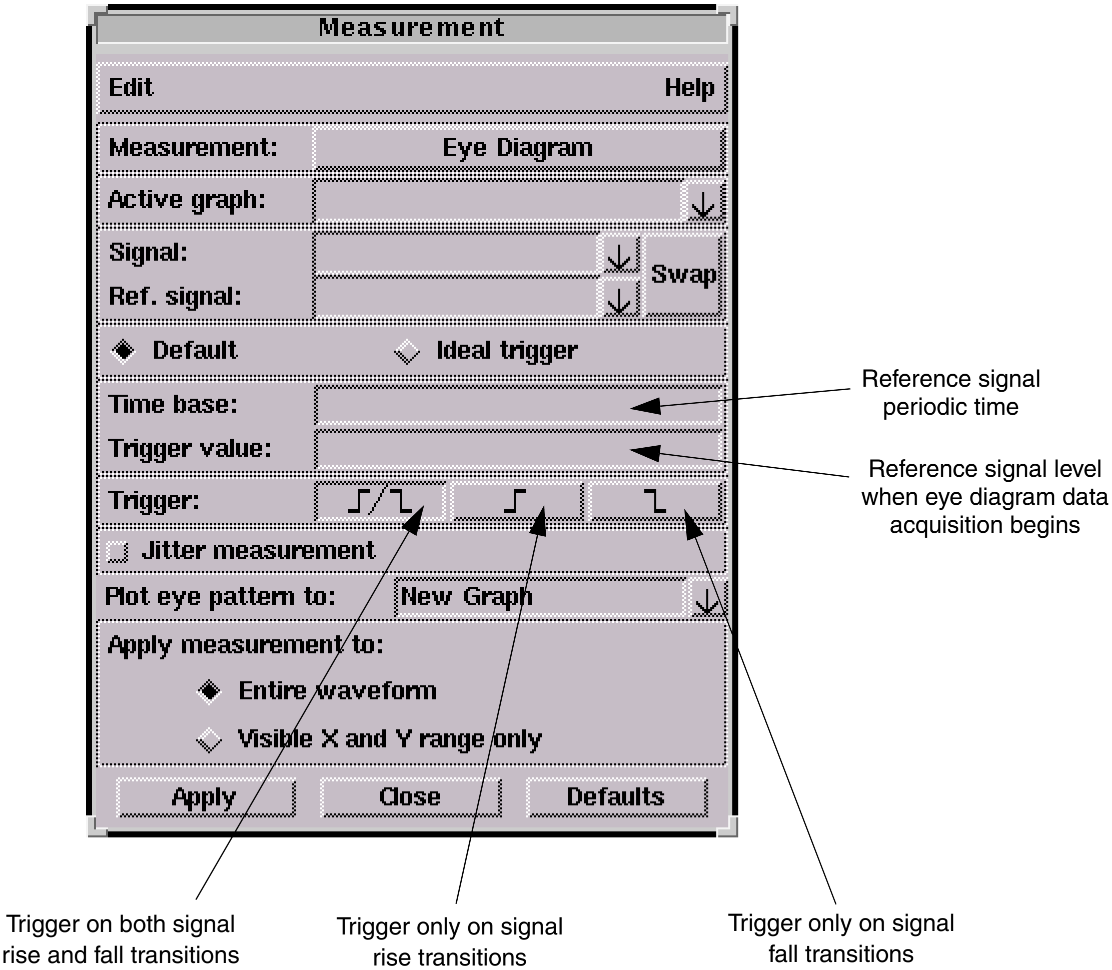
None

Examples

This section shows how you can perform a Default Trigger and Ideal Trigger Eye Diagram measurement.

Example 1: Default Trigger Eye Diagram

The Eye Diagram Measurement dialog box, as shown in the following figure, can be accessed by choosing **Tools > Measurement Tool**, and then choosing **Time Domain > Eye Diagram** from the Measurement button.



Measurement: Eye Diagram

● Default ○ Ideal trigger

□ Jitter measurement

Plot eye pattern to: New Graph

Apply measurement to:

● Entire waveform

○ Visible X and Y range only

Reference signal periodic time

Reference signal level when eye diagram data acquisition begins

Trigger on both signal rise and fall transitions

Trigger only on signal rise transitions

Trigger only on signal fall transitions

The following options are available when measuring a default trigger Eye Diagram:

- Time Base (optional): Specifies the width of the eye diagram X axis (or the X length of the eye diagram segments). When left blank, the measurement folds the input signal at the trigger points. The X axis lengths of segments are not fixed; instead, they are equal to the delta X value between trigger points.
- Trigger value: Specifies the trigger value at which the eye diagram segments start.
- Trigger: Specifies the type of transition for a trigger.

Example 2: Ideal Trigger Eye Diagram

You can access the Eye Diagram Measurement dialog box by opening the Measurement Tool, choosing the **Time Domain > Eye Diagram** menu item, and selecting the Ideal Trigger radio button.

The following options are available in the Eye Diagram Measurement dialog when performing an Ideal Trigger Eye Diagram measurement:

- Time Base: Specifies the width of the eye diagram X axis (or the X length of the eye diagram segments).
- Start X value: Specifies the start time (optional).
- End X value: Specifies the end time (optional).

The eye diagram measurement can be applied to the entire waveform or exclusively to the visible X and Y range of the output signal.

Graphical Interface Description

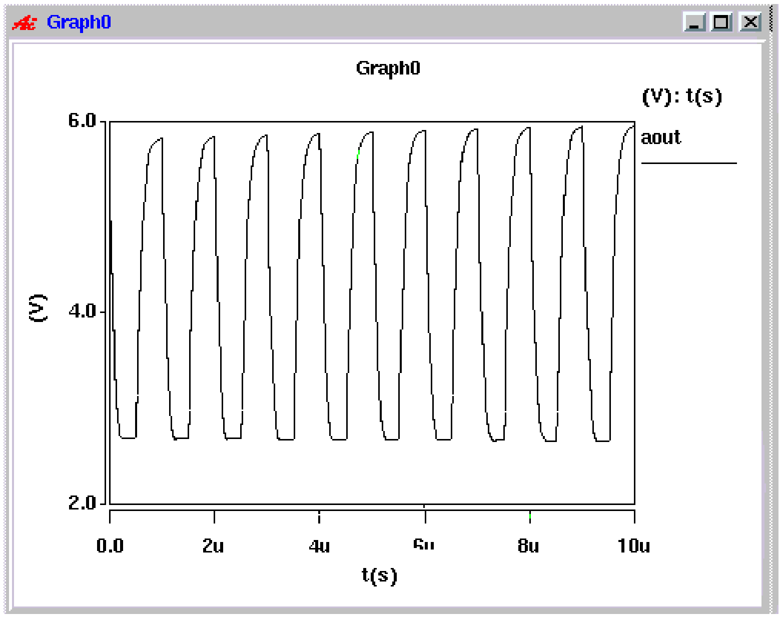
Field Names in the Eye Diagram Measurement Dialog:

- Time Base: Specifies the width of the eye diagram X axis (or the X length of the eye diagram segments). This value is optional when performing a default trigger Eye Diagram measurement.
- Trigger Value: Specifies the trigger value at which the eye diagram segments start.
- Trigger: Specifies the starting point for the measurement, which can start from rising edge only, fall edge only, or either rising or fall edge on the reference signal.

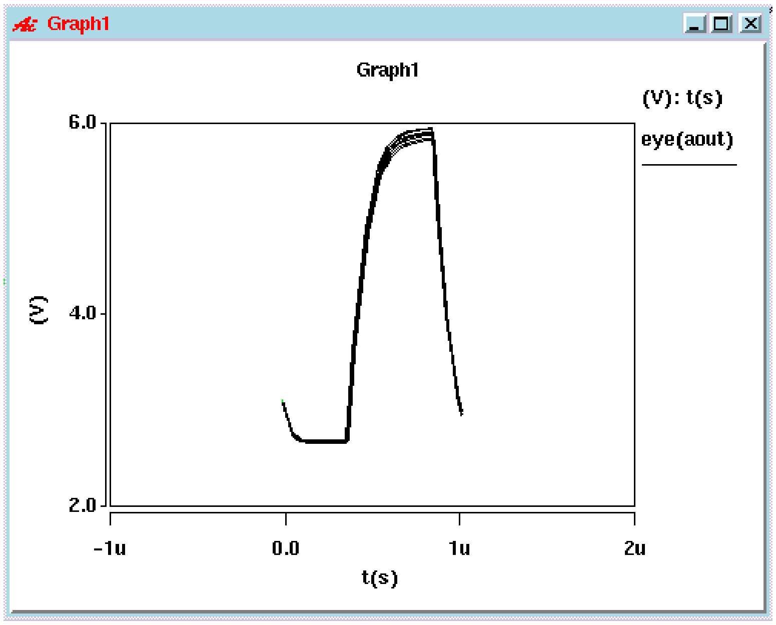
- Jitter Measurement: Applies Jitter measurements to the Eye Diagram. See Adding Jitter Measurements to Eye Diagrams for more information.
- Plot eye pattern to: Specifies the graph to which the eye pattern is plotted.

The eye diagram measurement has the ability to be applied to the entire waveform or exclusively to the visible X and Y range of the output signal.

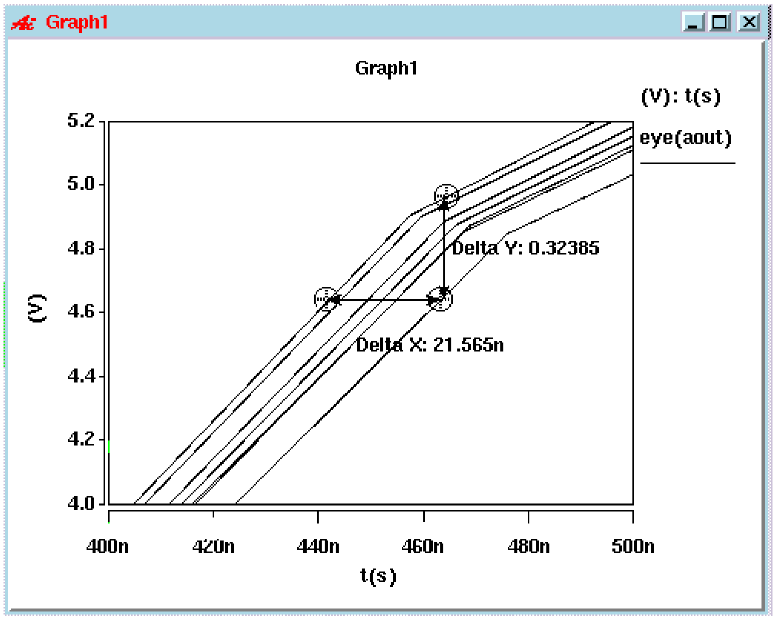
The following figure shows the aout waveform that is traced from the single_amp.tr.ai_pl plotfile. This example is located in the saber_amp examples directory.



The following figure shows the output of the signal in the aout waveform after the eye diagram measurement has occurred.

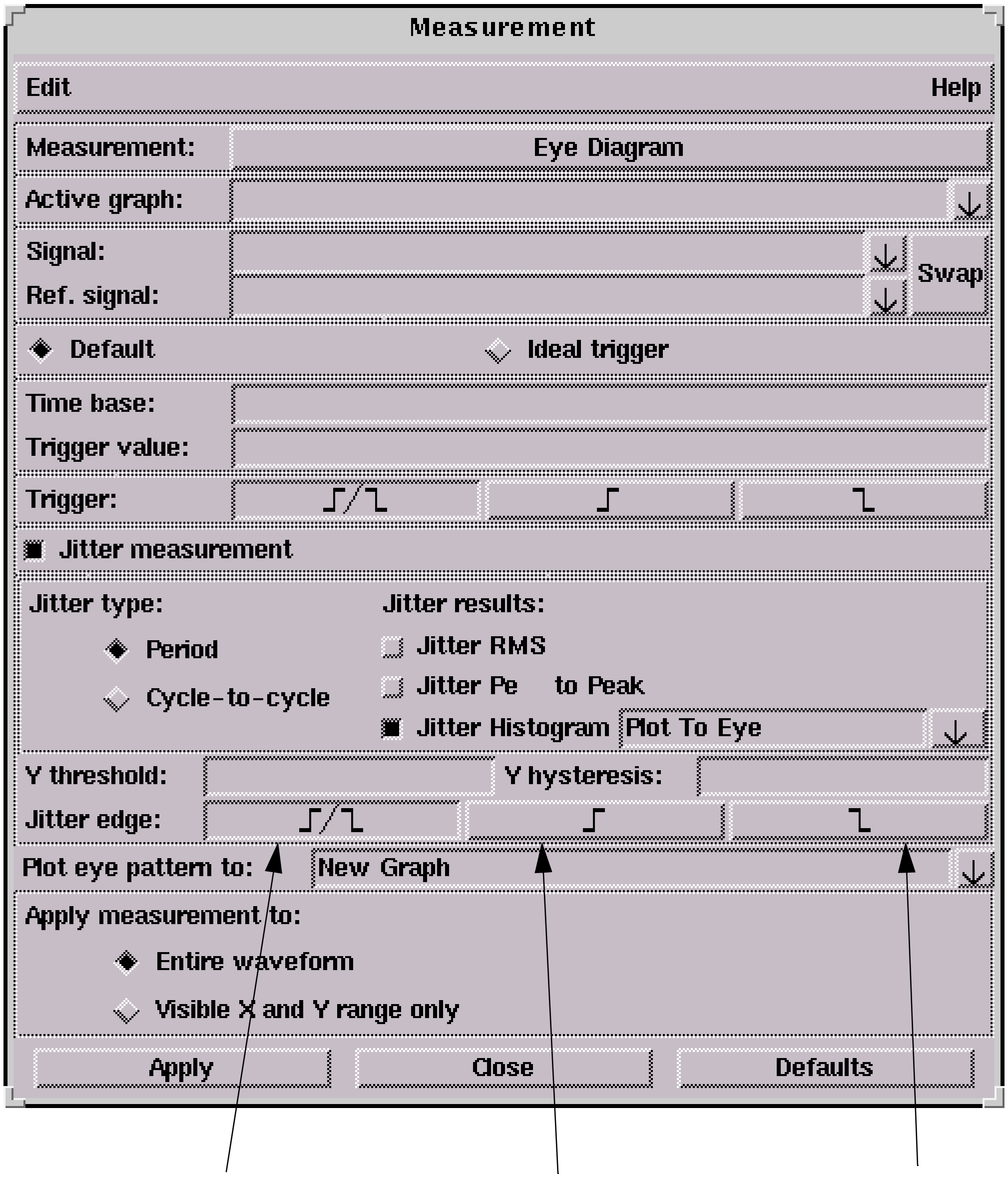


The figure below shows the signal gain after applying delta X and Y measurements to a portion of the output signal.



Adding Jitter Measurements to Eye Diagrams

To add jitter information when creating a new Eye Diagram, select the Jitter Measurement button. Selecting this button expands the eye diagram measurement dialog box to show the following options for jitter:



Measurement: Eye Diagram

Active graph:

Signal:

Ref. signal:

◆ Default ◇ Ideal trigger

Time base:

Trigger value:

Trigger:

■ Jitter measurement

Jitter type:

◆ Period

◇ Cycle-to-cycle

Jitter results:

□ Jitter RMS

□ Jitter Pe to Peak

■ Jitter Histogram Plot To Eye

Y threshold: Y hysteresis:

Jitter edge:

Plot eye pattern to: New Graph

Apply measurement to:

◆ Entire waveform

◇ Visible X and Y range only

Trigger on both signal rise and fall transitions

Trigger only on signal rise transitions

Trigger only on signal fall transitions

Dialog Box Fields

Jitter Type

Period Jitter measures a signal, generally a repeating signal, from one edge to another similar repeating edge. Period Jitter is selected by default.

Cycle-to-cycle Jitter measures the difference between two adjacent periods when the timing information is known. The difference between the periods is the cycle-to-cycle change: period one minus period two.

Jitter Results

Jitter RMS shows the average mean jitter. Jitter Peak to Peak shows jitter between rise and fall transitions.

Plot Jitter Histogram displays the jitter data in a new graph or the active graph, and the results are displayed in the histogram as well.

You can plot the Jitter Histogram to the Eye Diagram, Active Graph, New Graph, or any other open graph window.

Jitter results are displayed on the Eye Diagram relative to the threshold line. The results can be moved or turned off and on by using the Measure Results dialog box.

Y Threshold

This value is optional and if left empty is calculated to be the middle y value.

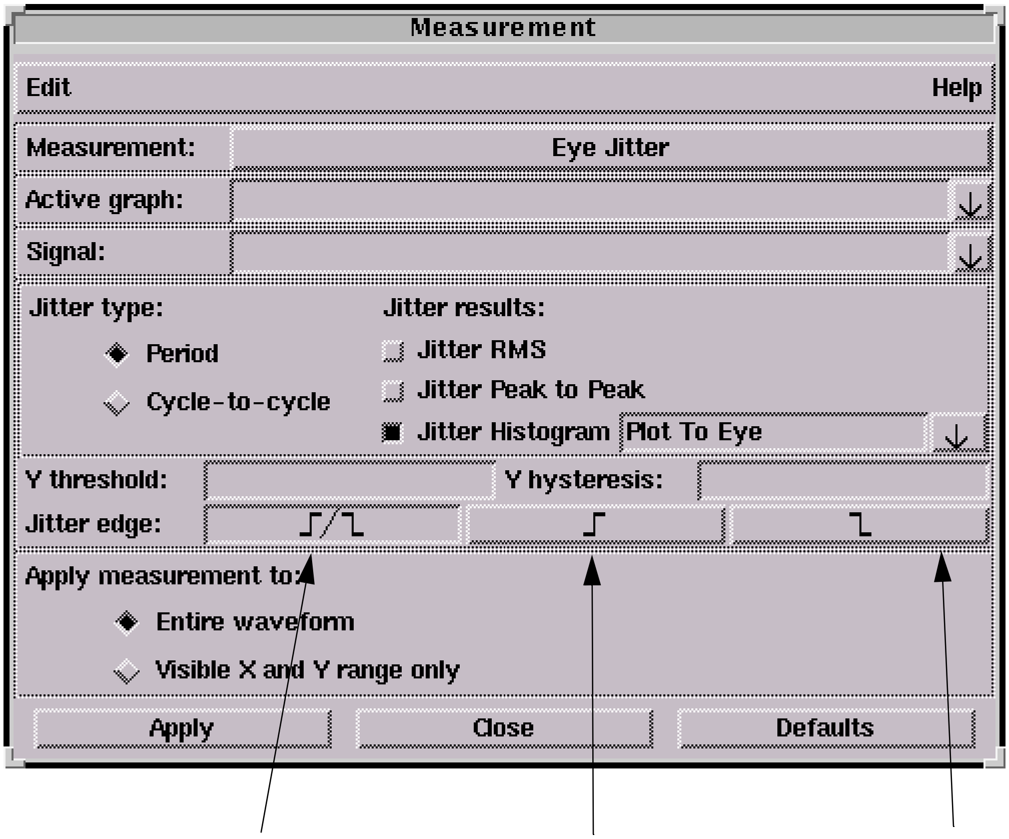
Y Hysteresis

You can specify an optional Y Hysteresis. If no value is specified, the default value is 0.

Jitter Edge

You can select from a three edges that are used to measure jitter measurement: Rising, Falling, and Either. Selecting the Rising edge calculates the jitter for the rising edges crossing the threshold. Selecting Falling calculates the jitter measurement for the falling edges crossing the threshold. Selecting Either calculates the jitter measurement for all points crossing the threshold. The default is Rising.

To apply a jitter measurement to an existing eye diagram, click the Measurement button and choose **Time Domain > Eye Jitter** from the menu that appears. The following Jitter Measurement dialog box appears.



Measurement: Eye Jitter

Active graph:

Signal:

Jitter type:
- ◆ Period
- ◇ Cycle-to-cycle

Jitter results:
- □ Jitter RMS
- □ Jitter Peak to Peak
- ■ Jitter Histogram Plot To Eye

Y threshold:

Y hysteresis:

Jitter edge:

Apply measurement to:
- ◆ Entire waveform
- ◇ Visible X and Y range only

Trigger on both signal rise and fall transitions

Trigger only on signal rise transitions

Trigger only on signal fall transitions

Eye Mask

Description

Eye Mask is used to display the maximum width and height of an eye diagram opening. This measurement provides three Eye Mask types: Diamond Eye Mask, Rectangle Eye Mask, and Hexagon Eye Mask.

Type of Measured Waveform

- Analog

Format for *.mask file import:

The *.mask file can contain one or more types of mask data. The first line in the file must indicate the version of the format and can contain one or more mask data lines. Each of the following lines after the version format represent one mask and its associated data. For example:

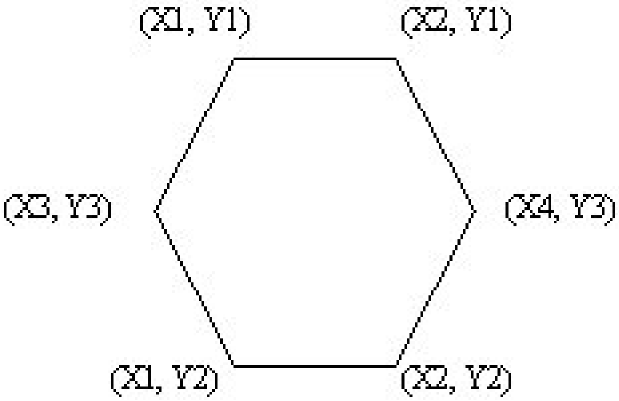
```
FORMAT_VERSION=1
hex_MASK  81.22  466.96  14.83   508.17  5.0   3.22  4.08
rec_MASK  81.22  466.96  5.0     3.22
dia_MASK  17.74  498.18  262.96  2.69    5.75  4.29
```

where hex_MASK, rec_MASK, and dia_MASK represent Hexagon, Rectangle, and Diamond masks, respectively.

Hexagon Format:

```
hex_MASK X1 X2 X3 X4 Y1 Y2 Y3
```

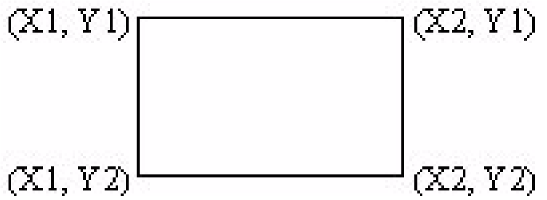
The hexagon mask data elements are placed as follows:



Rectangle Format:

```
rec_MASK X1 X2 Y1 Y2
```

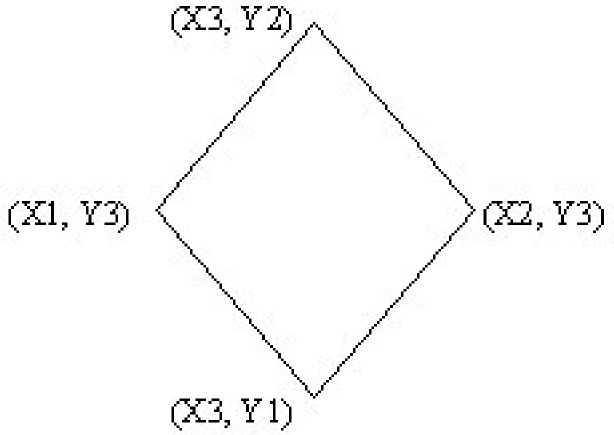
The rectangle mask data elements are placed as follows:



Diamond Format:

```
dia_MASK  X1 X2     X3 Y1 Y2 Y3
```

The diamond mask data elements are placed as follows:



Command Group

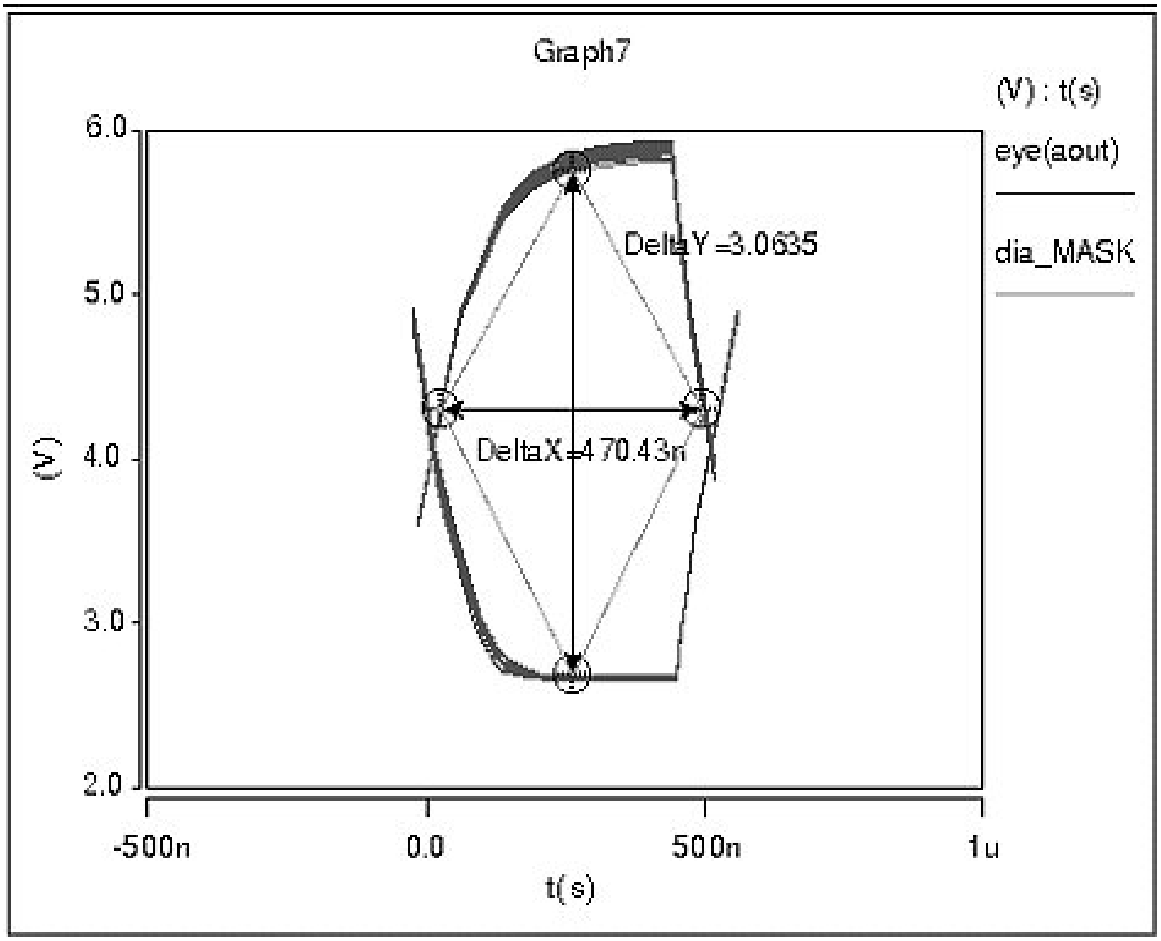
Time domain

Syntax

None

Example

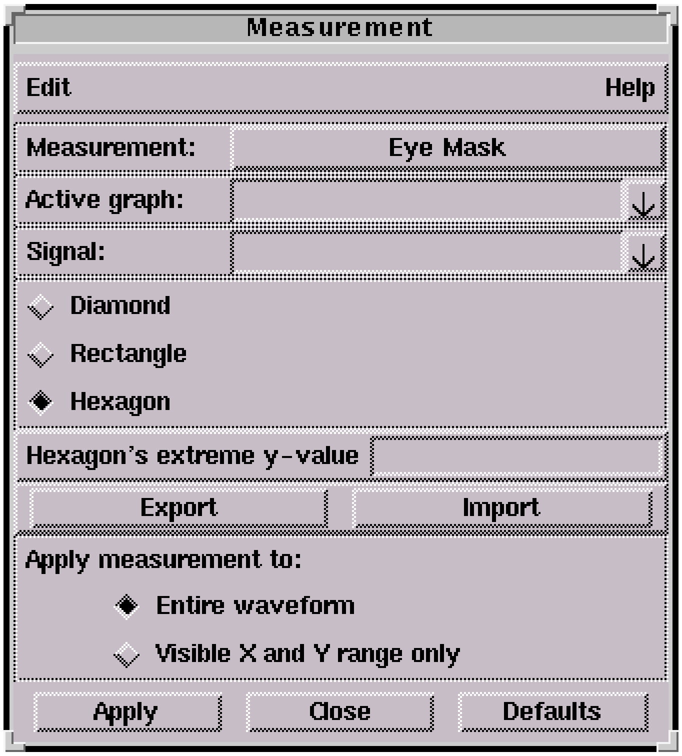
The following figure is the diamond eye mask with accuracy of x and y range 0% as set in the previous figure:



Graphical Interface Description

Dialog Box Fields

The Eye Mask Measurement dialog box, as shown in the following figure, is opened by choosing **Tools > Measurement Tool**, and then clicking the Measurement button and choosing **Time Domain > Eye Mask**.



Measurement: Eye Mask

Active graph:

Signal:

◇ Diamond

◇ Rectangle

◆ Hexagon

Hexagon's extreme y-value

Apply measurement to:

◆ Entire waveform

◇ Visible X and Y range only

Select the Eye Mask type:

- Diamond Mask: When selecting this Eye Mask type, enter the Accuracy (& of x and y range) to indicate the percentage of the x and y ranges within which the Diamond Mask is calculated.
- Hexagon: For this type, enter the Hexagon's extreme y-value to specify the maximum or minimum y value for the eye mask location.
- Rectangle: For this type, enter the Rectangle's extreme y-value to specify the maximum or minimum y value for the eye mask location.

Export:

- After you apply an Eye Mask measurement, click this button to export the masks to a text file (*.mask).

Import:

- Click this button to open a text file (*.mask); this will apply the masks in the file to the current signal.

Falltime

Description

Displays the falltime between selected upper/lower levels of a waveform. You can also compute the falltime based on manually-set upper/lower levels as described in the Measurement Tool topic titled “Manually Set a Custom Topline/ Baseline.”

Type of Measured Waveform

- Analog

Possible Errors

An error is reported if the waveform contains no falling edges.

Falltime Calculation

The falltime is calculated by finding a crossing with the middle level of the waveform. Looking forward from this point, the time when the waveform falls to the lower level is found. Looking backward, the time when the waveform rises to the upper level is also found. You set the upper and lower levels as 0-100%, 10-90%, or 20-80% of the Topline/Baseline levels. The difference in the times is the falltime, as shown in the example.

For more information about the lower, middle, and reference levels of a waveform, refer to "Waveform Reference Levels".

Command Group

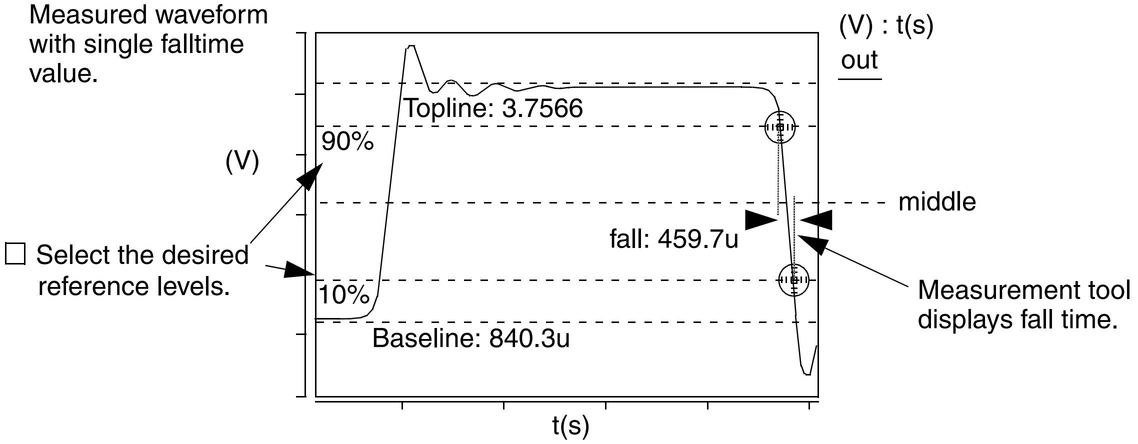
Time Domain

Syntax

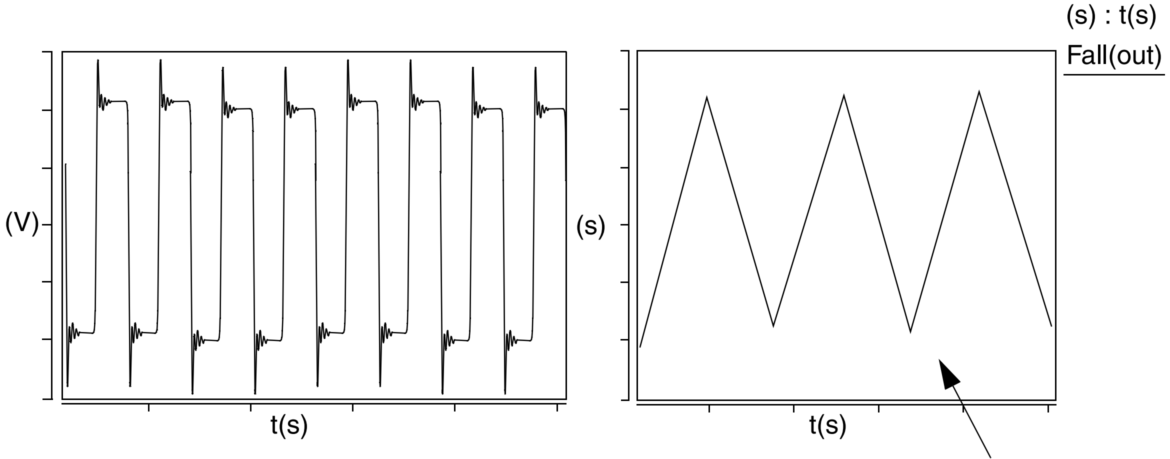
None

Example

Example 1



Example 2



Measured waveform with multiple falltime values.

Measurement tool produces this waveform showing falltime as a function of time.

Graphical Interface Description

Dialog Box Fields

Reference Levels	The following fields set the topline and baseline levels for the measured signal. You can display either of these levels on the waveform by clicking on the Visibility Indicator at the right of each field. Topline Specify a topline value or use the default value. Baseline Specify a baseline value below the topline value or use the default value.
0-100% 10-90% 20-80%	Click on one of these buttons to set an upper and lower range (in percent) relative to the topline/baseline levels. To compute a falltime based on a different percentage level than the defaults, refer to the topic titled Manually Set a Custom Topline/Baseline.
Create New Waveform on Active Graph or New Graph	Falltime vs. t - Creates a new waveform of falltime value(s) (Y-axis) versus elapsed time (X-axis). See Example 2.

Frequency

Description

Displays the frequency of a periodic waveform relative to default or specified topline and baseline levels.

Type of Measured Waveform

- Analog, event-driven analog, digital

Possible Errors

An error is reported if the waveform does not contain at least one complete cycle.

Frequency Calculation

The frequency is calculated as the reciprocal of the period. For information on how the period is computed, refer to the Period Calculation.

Command Group

Time Domain

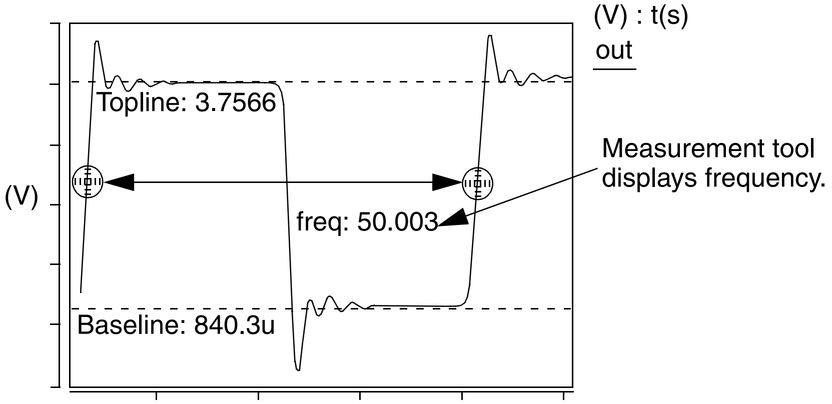
Syntax

None

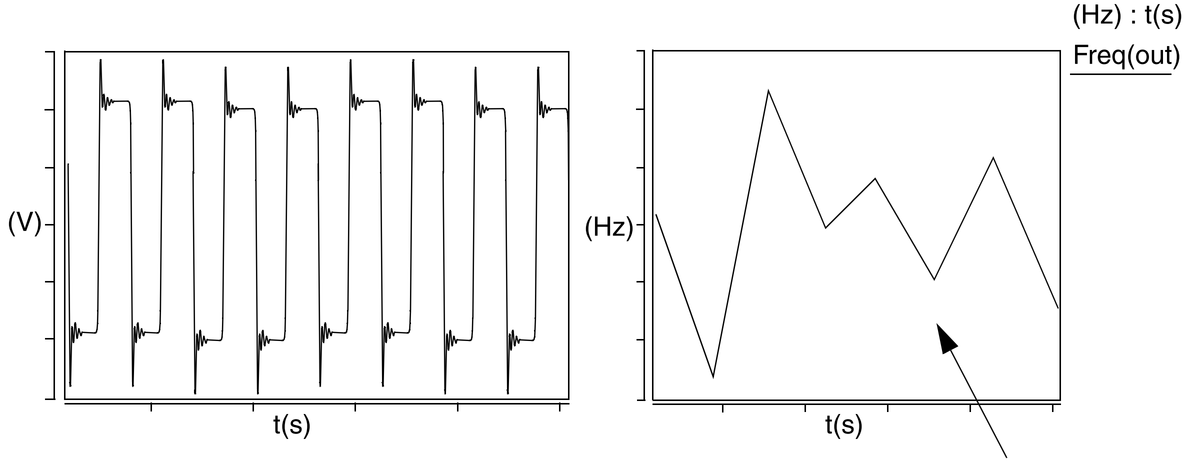
Example

Example 1

Measured waveform with single frequency value.



Example 2



Measured waveform with multiple frequency values.

Measurement tool produces this waveform showing frequency as a function of time.

Graphical Interface Description

Dialog Box Fields

Reference Levels

The following fields set the topline and baseline levels for the measured signal. You can display either of these levels on the waveform by clicking on the Visibility Indicator at the right of each field.

Topline

Specify a topline value or use the default value.

Baseline

Specify a baseline value below the topline value or use the default value.

Trigger

⎍⎍/⎍⎍ Specifies that the measurement starts from a period with either a rising or falling edge.

⎍⎍ Specifies that the measurement starts from a period with a rising edge.

⎍⎍ Specifies that the measurement starts from a period with a falling edge.

Create New Waveform on Active Graph or New Graph

Frequency vs. t - A new waveform is computed with frequency (Y-axis) versus time (X-axis). See Example 2.

Gain Margin

Description

Displays the gain margin in dB of a complex waveform.

Type of Measured Waveform

- Analog (must be complex)

Possible Errors

An error is reported if the phase of the measured waveform does not pass through –180 degrees or if the waveform is not complex.

Gain Margin Calculation

The gain margin is defined as the difference between the gain of the measured waveform and 0 dB at the frequency where the phase shift is –180 degrees.

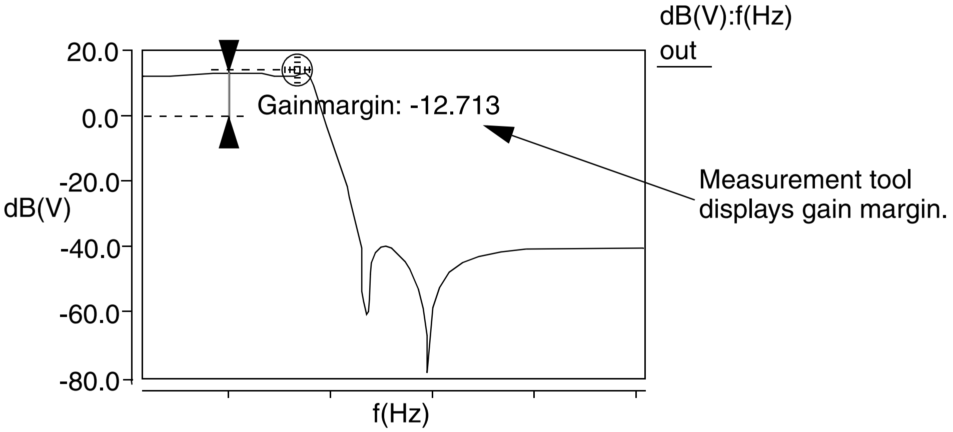
Command Group

Frequency Domain

Syntax

None

Example



Graphical Interface Description

Dialog Box Fields

There are no unique fields in this form.

Highpass

Description

Displays the corner frequency of a waveform with a highpass shape. The measurement is made relative to a default or specified topline level and a specified offset.

Type of Measured Waveform

- Analog

Highpass Calculation

The corner frequency is found by searching from right to left until the waveform first falls below the measurement level, which is determined by the offset (from the topline) that you specify.

Topline

If you do not specify the topline, a default value is calculated by using a method specified in the Default Topline/Baseline field in the Measurement Preference dialog box.

Offset

Computed as one of
  topline - offset_value,
  topline + offset_value,
  topline * offset_value, or
  topline / offset_value,
depending on which operator you choose. This level is also referred to as the Measurement Level, as shown in the Example.

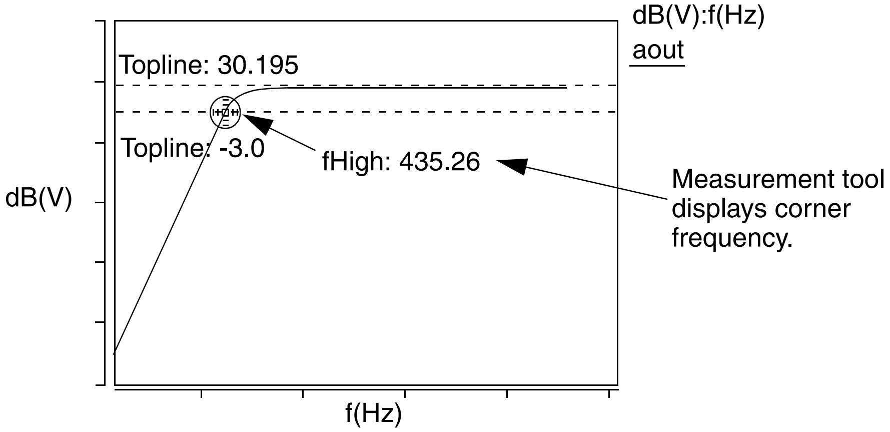
Command Group

Frequency Domain

Syntax

None

Example



Graphical Interface Description

Dialog Box Fields

Reference Levels	If you want to see the topline or offset level, or both, displayed on the waveform, click on the Visibility Indicator to the right the Topline or Offset field. Topline You set this field to a default or a specified level. Offset You specify an offset value, to be applied relative to the Topline value. The default is 3. You must also choose which operator to use (–, +, *, or /) along with the specified level. The default is the minus sign. This resulting level is also called the measurement level.

Histogram

Description

Displays a histogram of a waveform.

Histograms can display absolute values (such as the number of runs that fall into a certain range), where only integer numbers make sense. Normalized values can also be displayed (number of runs in a range divided by the total number of runs), where the values are fractions between 0 and 1. By design, the Y axis of histograms can have non-integer values.

Type of Measured Waveform

- Scatter plot, analog, event-driven analog

Command Group

Statistics

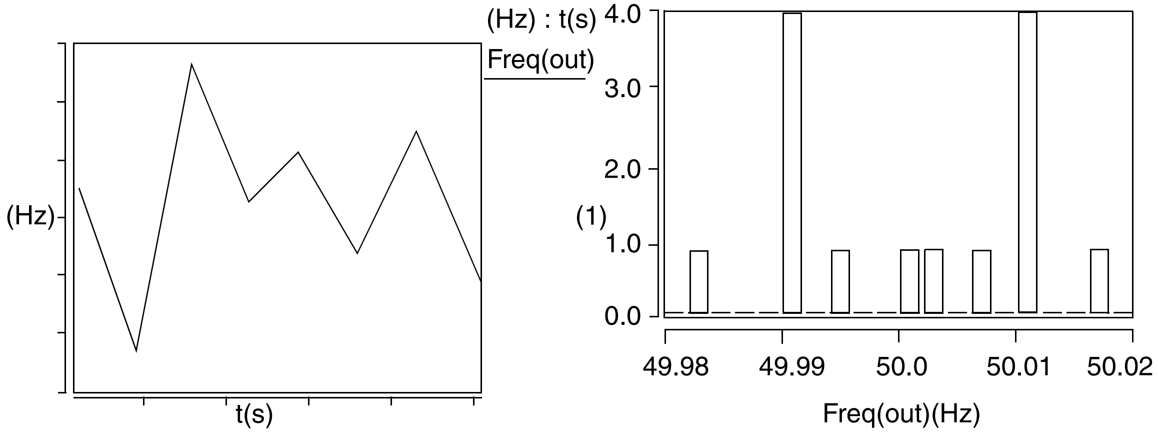
Syntax

None

Example

Previous measurement result of multiple frequency values.

Histogram of previous measurement result.



Graphical Interface Description

Dialog Box Fields

Category List	All Statistic category items appear below the Signal field. Select the Histogram item and any other items you want to measure.

Horizontal Level

Description

Displays a moveable horizontal line to identify Y-axis levels.

Type of Measured Waveform

- Analog, event-driven analog, scatter plot, histogram, spectral

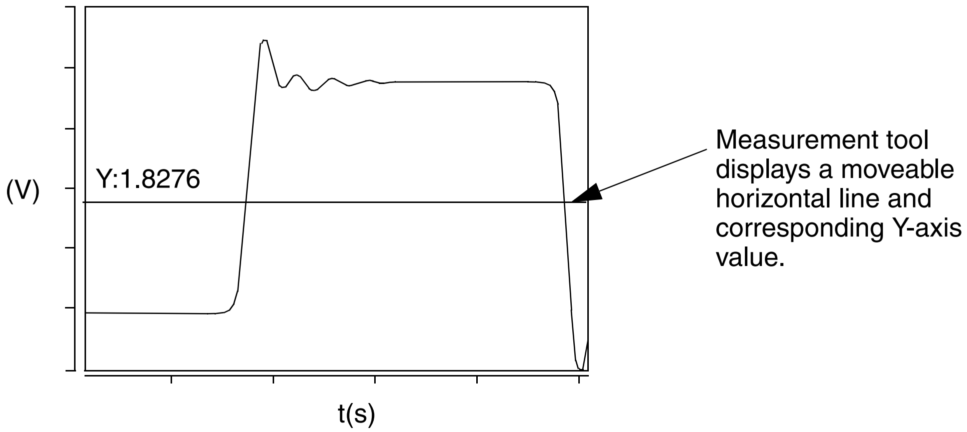
Command Group

General

Syntax

None

Example



Graphical Interface Description

Dialog Box Fields

Location (Optional) Y Value — You can specify a Y-value to place a moveable horizontal line on the waveform. If you do not specify one, a default value is computed.

Imaginary

Description

Displays the imaginary value of a point on a waveform.

Type of Measured Waveform

- Analog (must be complex)

Imaginary Calculation

The imaginary value of a waveform is the imaginary part of a complex argument. If there is no complex part then the value 0.0 is returned.

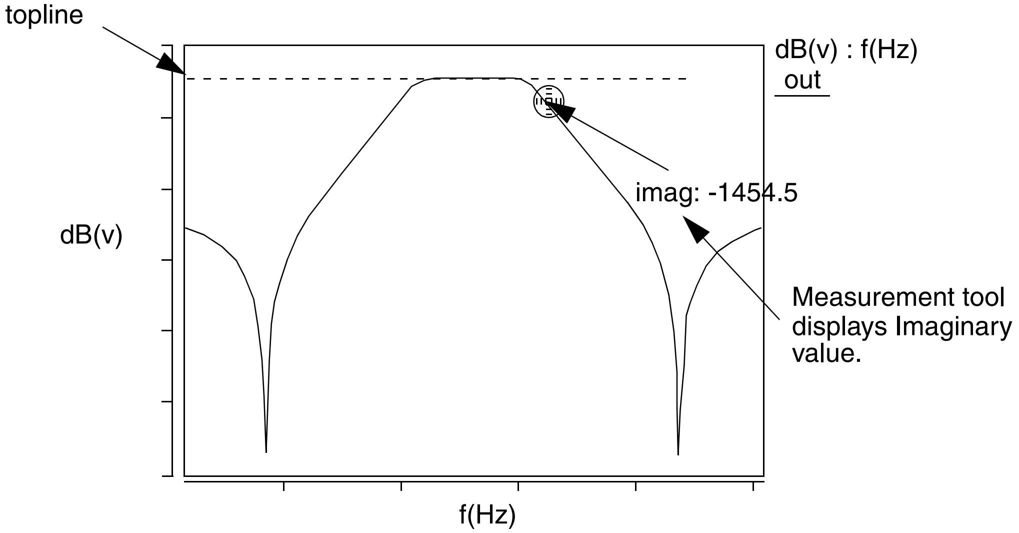
Command Group

Frequency Domain

Syntax

None

Example



Graphical Interface Description

Dialog Box Fields

X Value	Optional. You can provide an X-value and the tool will provide the Y-value at that coordinate. If you do not specify the X-value, a default is used.

IP2

Description

For a nonlinear system with an output and two fundamental excitation frequencies f1 and f2, the second order Intercept Point is the point on a set of PowerOut (PowerOut1 and PowerOut2) versus PowerIn curves at which a line extrapolated from PowerOut1 with a slope of 1, and a line with a slope of 2 extrapolated from PowerOut2 intersect. PowerOut1 represents the first order term, and PowerOut2 is the second order term. Both lines must be extrapolated from a region with sufficiently low input power. OIP2 is the value of PowerOut at which IP2 occurs, IIP2 is the value of the PowerIn value at which IP2 occurs.

Type of Measured Waveform

- Analog, Power type, either PowerOut versus PowerIn or PowerOut versus Frequency with PowerIn as a sweep parameter.

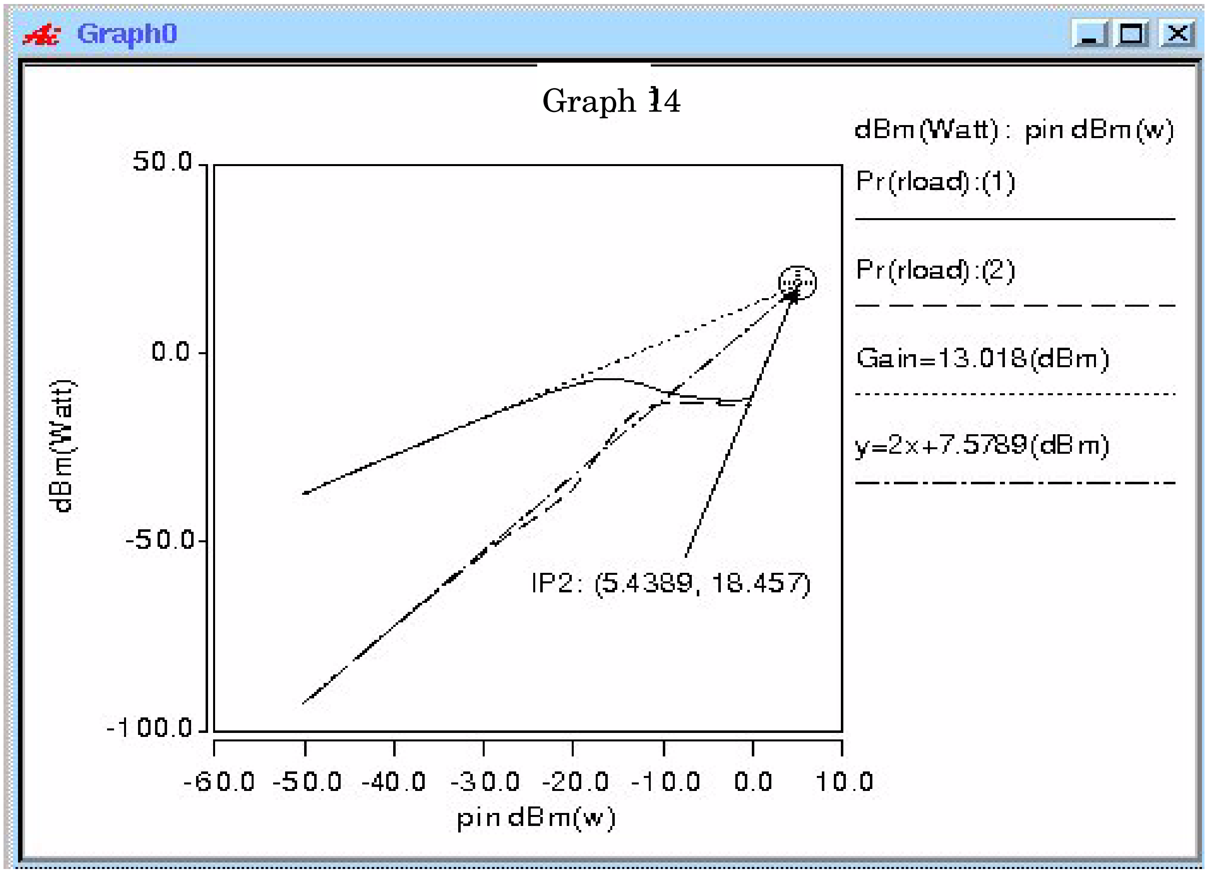
Command Group

RF

Syntax

None

Example

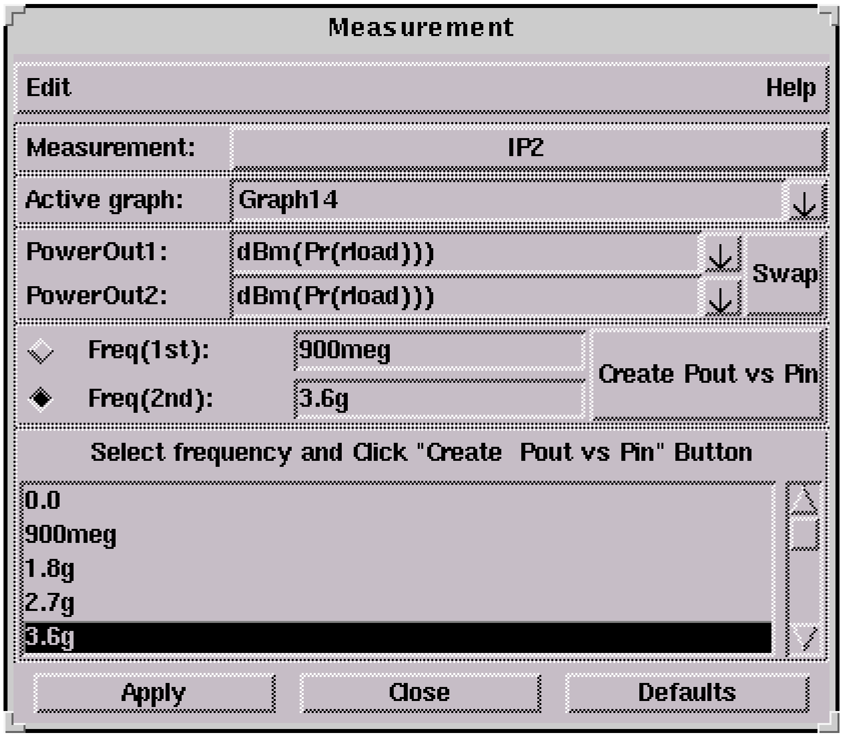


Given one waveform, PowerOut versus Frequency with PowerIn as a sweep parameter, the following Dialog Box Fields show:

PowerOut1	Waveform of PowerOut versus Frequency with PowerIn as a sweep parameter

PowerOut2	Second-order term PowerOut2 versus PowerIn curve
Freq (First)	Frequency to generate first-order term PowerOut1 versus PowerIn curve
Freq (Second)	Frequency to generate second-order term PowerOut2 versus PowerIn curve

Examples



Measurement: IP2

Active graph: Graph14

PowerOut1: dBm(Pr(rload)))

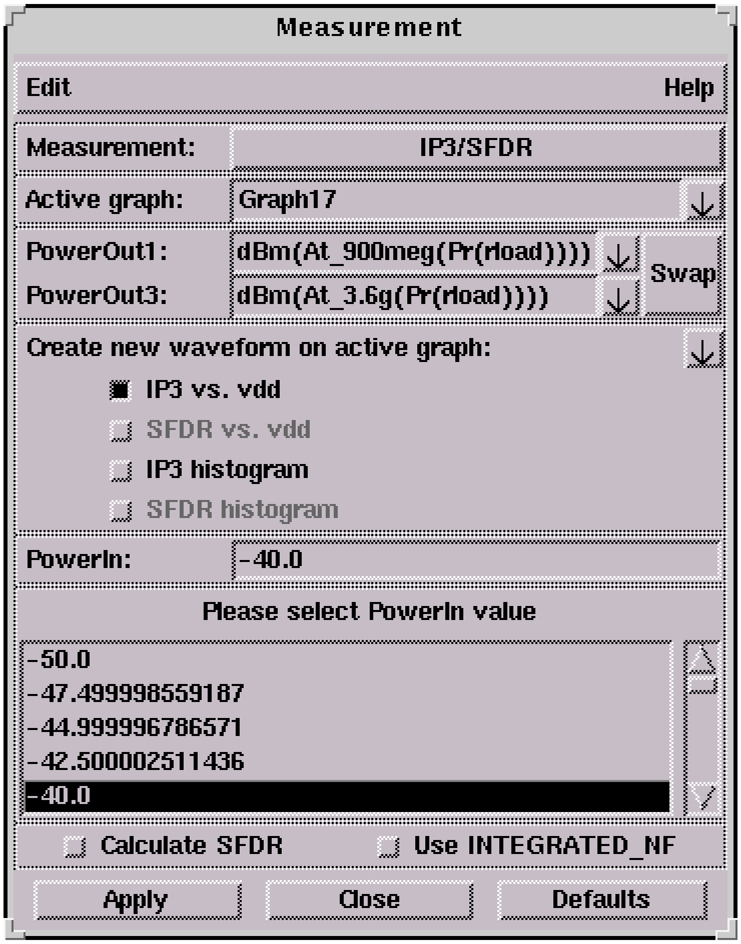
PowerOut2: dBm(Pr(rload)))

◇ Freq(1st): 900meg

◆ Freq(2nd): 3.6g

Select frequency and Click "Create Pout vs Pin" Button

0.0
900meg
1.8g
2.7g
3.6g



Measurement: IP3/SFDR

Active graph: Graph17

PowerOut1: dBm(At_900meg(Pr(rload)))

PowerOut3: dBm(At_3.6g(Pr(rload)))

Create new waveform on active graph:

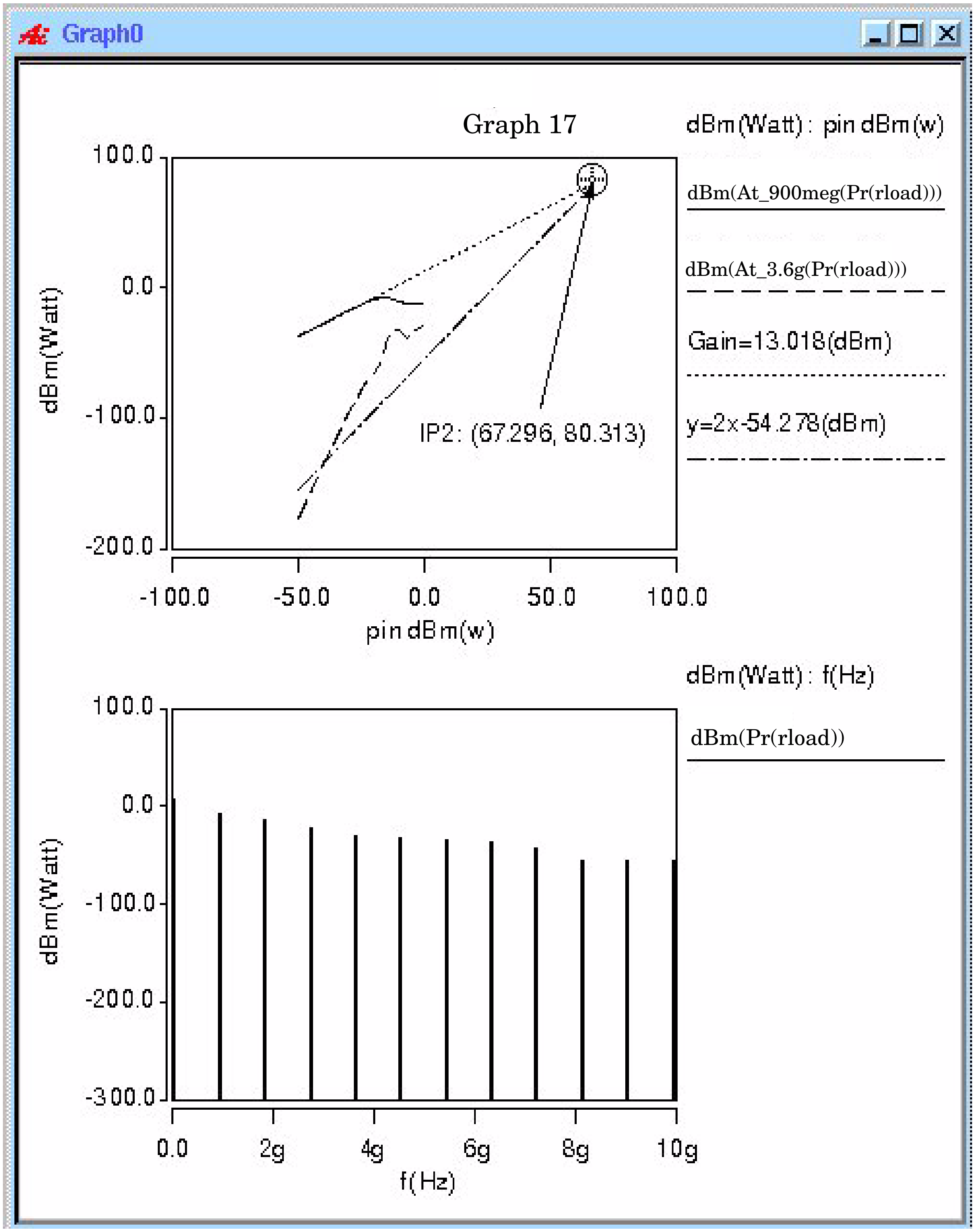
■ IP3 vs. vdd
□ SFDR vs. vdd
□ IP3 histogram
□ SFDR histogram

PowerIn: -40.0

Please select PowerIn value

```
-50.0
-47.499998559187
-44.999996786571
-42.500002511436
-40.0
```

□ Calculate SFDR □ Use INTEGRATED_NF


Graph0
Graph 17
dBm(Watt) : pin dBm(w)
dBm(At_900meg(Pr(rload)))
dBm(At_3.6g(Pr(rload)))
Gain=13.018(dBm)
y=2x-54.278(dBm)
IP2: (67.296, 80.313)
100.0
0.0
-100.0
-200.0
dBm(Watt)
-100.0
-50.0
0.0
50.0
100.0
pin dBm(w)
dBm(Watt) : f(Hz)
dBm(Pr(rload))
100.0
0.0
-100.0
-200.0
-300.0
dBm(Watt)
0.0
2g
4g
6g
8g
10g
f(Hz)

Graphical Interface Description

Dialog Box Fields

Given two waveforms, one is PowerOut1 versus PowerIn, the other is PowerOut2 versus PowerIn, the following Dialog Box Fields show:

PowerOut1	First-order term PowerOut1 versus PowerIn curve
PowerOut2	Second-order term PowerOut2 versus PowerIn curve
PowerIn	PowerIn value

IP3/SFDR

Description

For a nonlinear system with an output and two fundamental excitation frequencies f1 and f2, the third order Intercept Point is the point on a set of PowerOut (PowerOut1 and PowerOut3) versus PowerIn curves at which a line extrapolated from PowerOut1 with a slope of 1, and a line with a slope of 3 extrapolated from PowerOut3 intersect. PowerOut1 represents the first order term, and PowerOut3 is the third order term. Both lines must be extrapolated from a region with sufficiently low input power. OIP3 is the value of PowerOut at which IP3 occurs, IIP3 is the value of the PowerIn value at which IP3 occurs.

Type of Measured Waveform

- Analog, Power type, either PowerOut versus PowerIn or PowerOut versus Frequency with PowerIn as a sweep parameter.

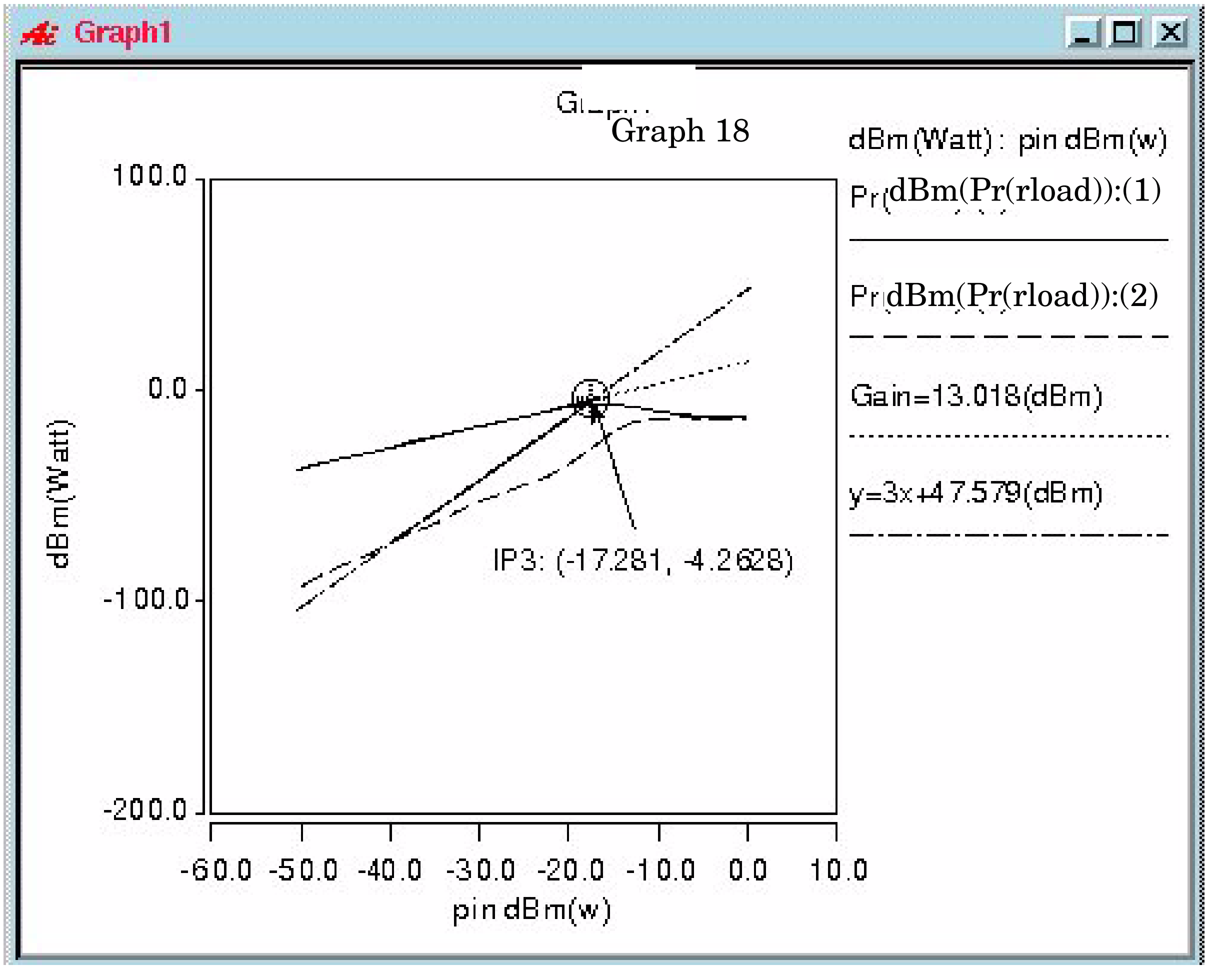
Command Group

RF

Syntax

None

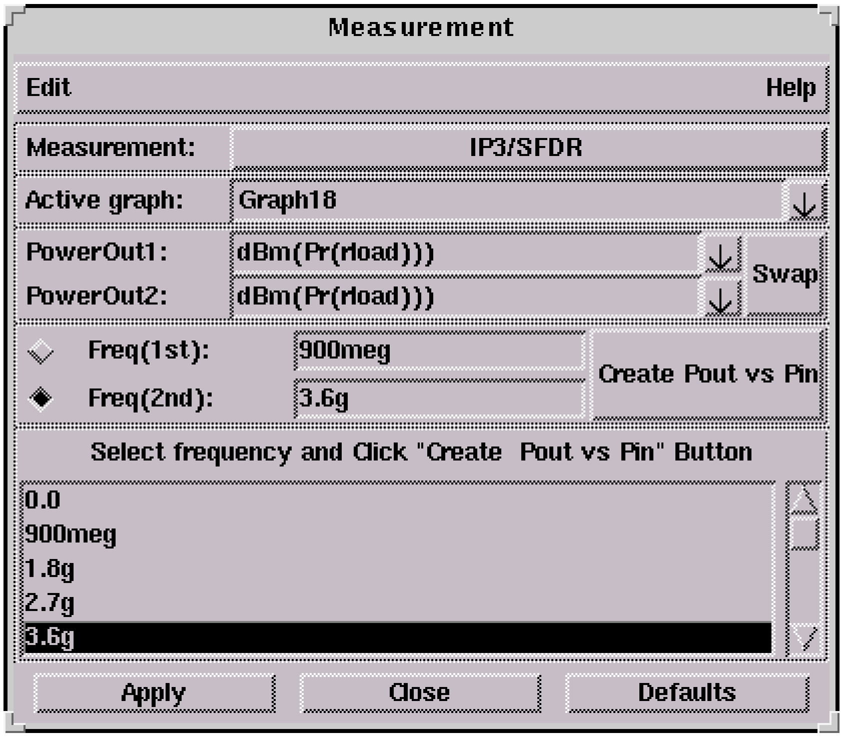
Example



Given one waveform, PowerOut versus Frequency with PowerIn as a sweep parameter, the following Dialog Box Fields show:

PowerOut1	Waveform of PowerOut versus Frequency with PowerIn as a sweep parameter
PowerOut3	This field is not used in this case
Freq (First)	Frequency to generate first-order term PowerOut1 versus PowerIn curve
Freq (Second)	Frequency to generate third-order term PowerOut3 versus PowerIn curve

Examples



Measurement: IP3/SFDR

Active graph: Graph18

PowerOut1: dBm(Pr(rload)))

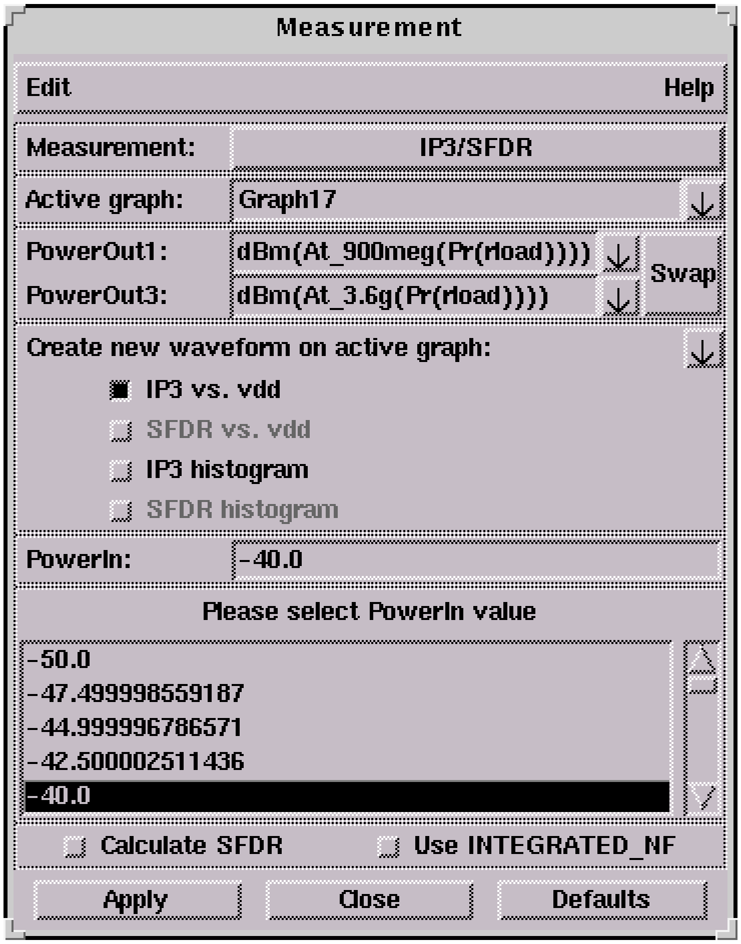
PowerOut2: dBm(Pr(rload)))

◇ Freq(1st): 900meg

◆ Freq(2nd): 3.6g

Select frequency and Click "Create Pout vs Pin" Button

0.0
900meg
1.8g
2.7g
3.6g



Measurement: IP3/SFDR

Active graph: Graph17

PowerOut1: dBm(At_900meg(Pr(rload))))

PowerOut3: dBm(At_3.6g(Pr(rload))))

Create new waveform on active graph:

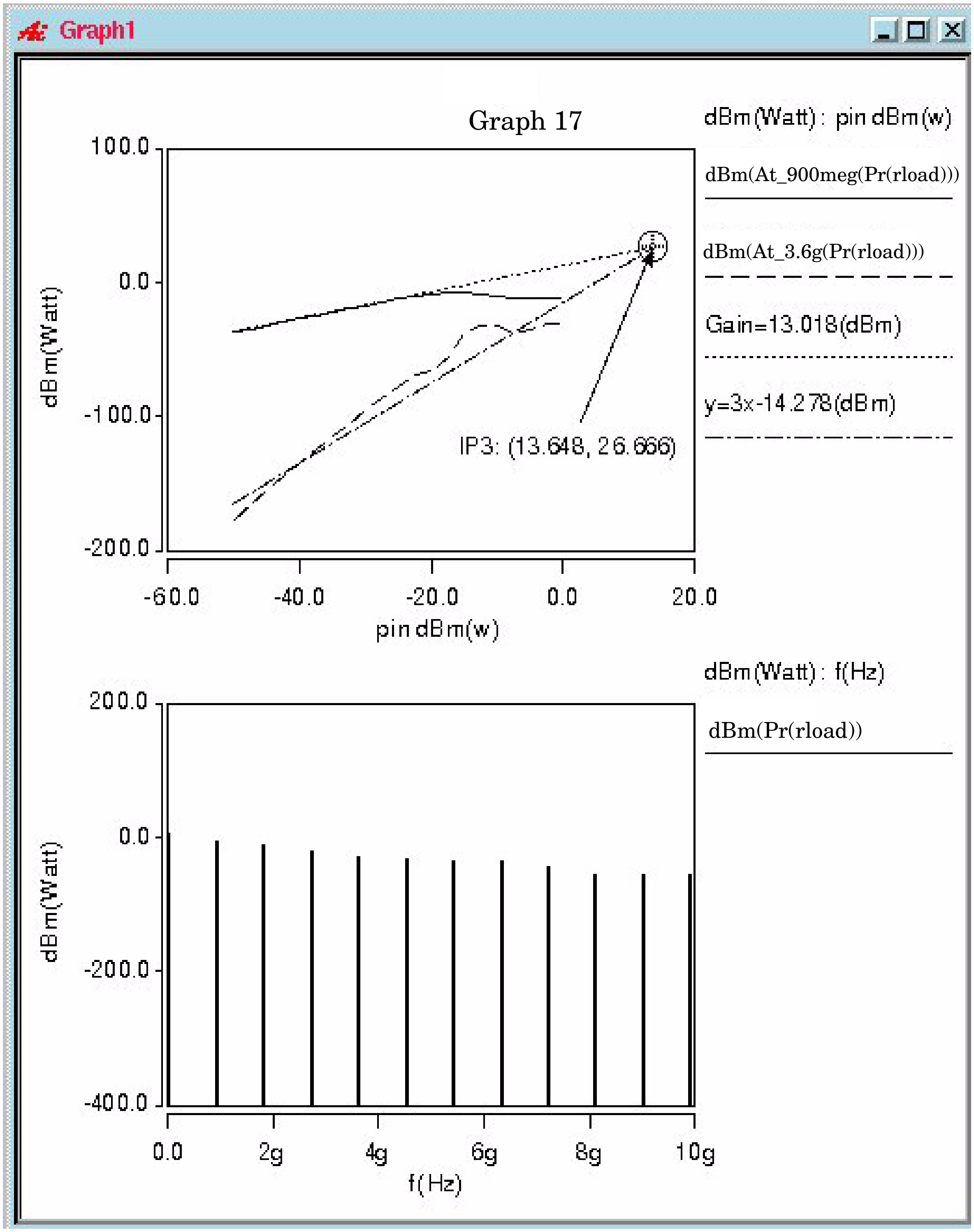
- ■ IP3 vs. vdd
- □ SFDR vs. vdd
- □ IP3 histogram
- □ SFDR histogram

PowerIn: -40.0

Please select PowerIn value

```
-50.0
-47.499998559187
-44.999996786571
-42.500002511436
-40.0
```

□ Calculate SFDR □ Use INTEGRATED_NF


Graph1
Graph 17
dBm(Watt) : pin dBm(w)
dBm(At_900meg(Pr(rload)))
dBm(At_3.6g(Pr(rload)))
Gain=13.018(dBm)
y=3x-14.278(dBm)
IP3: (13.648, 26.666)
100.0
0.0
-100.0
-200.0
dBm(Watt)
-60.0
-40.0
-20.0
0.0
20.0
pin dBm(w)
dBm(Watt) : f(Hz)
dBm(Pr(rload))
200.0
0.0
-200.0
-400.0
dBm(Watt)
0.0
2g
4g
6g
8g
10g
f(Hz)

Graphical Interface Description

Dialog Box Fields

Given two waveforms, one is PowerOut1 versus PowerIn, the other is PowerOut3 versus PowerIn, the following Dialog Box Fields show:

PowerOut1	First-order term PowerOut1 versus PowerIn curve
PowerOut3	Third-order term PowerOut3 versus PowerIn curve
PowerIn	PowerIn value

Jitter

Description

See Adding Jitter Measurements to Eye Diagrams for information on the Jitter measurement.

Syntax

None

Length

Description

Displays the length of a straight line that connects two X-axis points on a waveform or two X-axis points on two waveforms. If two waveforms are selected, the two waveforms do not need to be the same type, but they must be in the same graph region.

Type of Measured Waveform

- Analog, event-driven analog, scatter plot

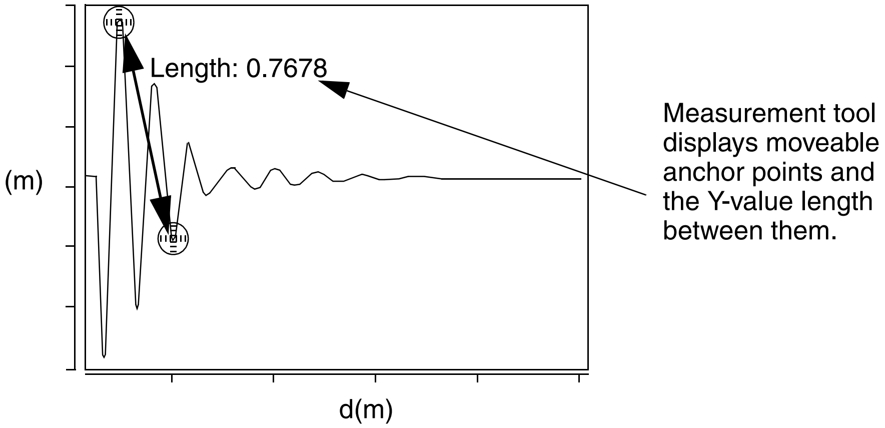
Command Group

General

Syntax

None

Example



Measurement tool displays moveable anchor points and the Y-value length between them.

Graphical Interface Description

Dialog Box Fields

Anchor Snap

Signal

Generates moveable anchor points that snap to the signal waveform. See the Example.

Closest Measurement

When you have multiple anchor points visible on a graph from previous measurements, this setting causes the measurement to snap to one of those nearest points. If there is no visible measurement on the active graph from which to snap, an error message is displayed. To see an example, refer to the Point Marker Example.

Floating

Generates moveable anchor points are float within the graph. To see an example, refer to the Delta Y Example.

Location (Optional)

You can optionally specify two X-values that are used

to determine the Y-value difference between the two X-values.

X1 Value

Optional value. You can change it by moving the anchor point after the initial measurement is made.

X2 Value   Optional value. You can change it by moving the anchor point after the initial measurement is made.

Local Max/Min

Description

Displays the local maximum or minimum point, or both, on a waveform.

Type of Measured Waveform

- Analog, event-driven analog

Possible Errors

An error is reported if a local minimum or maximum is not found within the waveform.

Local Min/Max Calculation

∩ A local maximum occurs when the slope of the waveform changes sign from positive to negative.

∪ A local minimum occurs when the slope of the waveform changes sign from negative to positive.

The end points of the waveform are not considered for either calculation.

The Peak Threshold value you specify determines whether a point at which a positive-to-negative or negative-to-positive slope change occurs should be selected as a local maximum or local minimum. The Peak Threshold value specifies the minimum change between an adjacent local maximum and local minimum relative to the peak-to-peak value of the waveform.

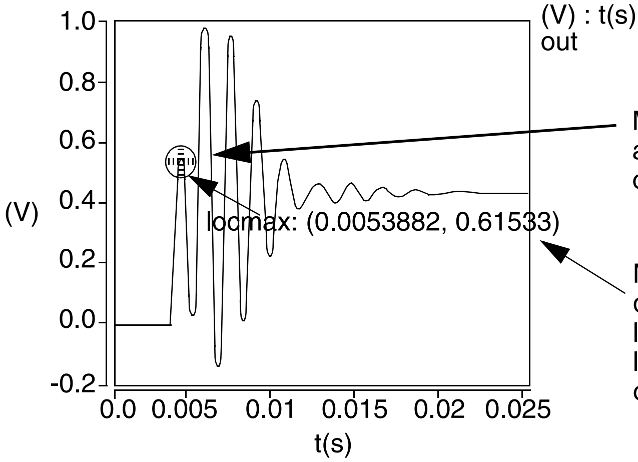
Command Group

General

Syntax

None

Example



Move the anchor point along waveform to find other local maximum values.

Measurement tool displays local minimum, local maximum, or local extreme, depending on the Peak Type setting.

Graphical Interface Description

Dialog Box Fields

Peak Type	∪	All local maximum and local minimum points are found.
	∩	Only the local maximum points are found.
	∩/∪	Only the local minimum points are found.
Peak Threshold	<n> %	Specify a value in percent of the peak to peak. Any peaks below this level are rejected by the measurement. The default setting is 2%.
Create New Waveform on Active Graph or New Graph		The peak type you have selected will determine which fields are displayed under this heading as follows:
	∩/∪	Local Extreme vs. t - A new waveform is plotted by connecting the Y-values found at the extreme points (Y-axis) against time (X-axis).
		X at Local Extreme vs. t - A new waveform is plotted by connecting the X-axis points found (Y-axis) against time (X-axis).

∩ Local Maximum vs. t -
A new waveform is plotted by connecting the Y-values found at the maximum points (Y-axis) against time (X-axis).

X at Local Maximum vs. t -
A new waveform is plotted by connecting the X-axis points found (Y-axis) against time (X-axis).

∪ Local Minimum vs. t -
A new waveform is plotted by connecting the Y-values found at the minimum points (Y-axis) against time (X-axis).

X at Local Minimum vs. t -
A new waveform is plotted by connecting the X-axis points found (Y-axis) against time (X-axis).

Multi-Member Waveform Options

Create New Waveform on Active Graph or New Graph

Creates a new waveform in the Active Graph or in a New Graph to display the results.

Local Extreme or Maximum or Minimum vs. _run or Vary_Parameter -
A scatter plot or analog waveform is displayed, which shows all the points found (Y-axis) against either the Monte-Carlo run (X-axis) or the Vary parameter value (X-axis).

X at Local Extreme or Maximum or Minimum vs. _run or Vary_Parameter -
A scatter plot or analog waveform is displayed, which shows all the points found (Y-axis) against either the Monte-Carlo run (X-axis) or the Vary parameter value (X-axis).

Local Extreme or Maximum or Minimum Histogram -
A histogram is plotted, which shows how many points were found (count, Y-axis) at each Y-value (X-axis).

X at Local Extreme or Maximum or Minimum Histogram -
A histogram is plotted, which shows how many points were found (count, Y-axis) at each X-value (X-axis).

Lowpass

Description

Displays the corner frequency of a waveform with a lowpass shape. The measurement is made relative to a default or specified topline and a specified offset.

Type of Measured Waveform

- Analog

Lowpass Calculation

The corner frequency is found by searching from left to right until the waveform first falls below the measurement level, which is determined by the offset (from the topline) that you specify.

Topline

If you do not specify the topline, a default value is calculated by using a method specified in the Default Topline/Baseline field in the Measurement Preference dialog box.

Offset

Computed as one of
  topline - offset_value,
  topline + offset_value,
  topline * offset_value, or
  topline / offset_value,
depending on which operator you choose. This level is also referred to as the Measurement Level, as shown in the Example.

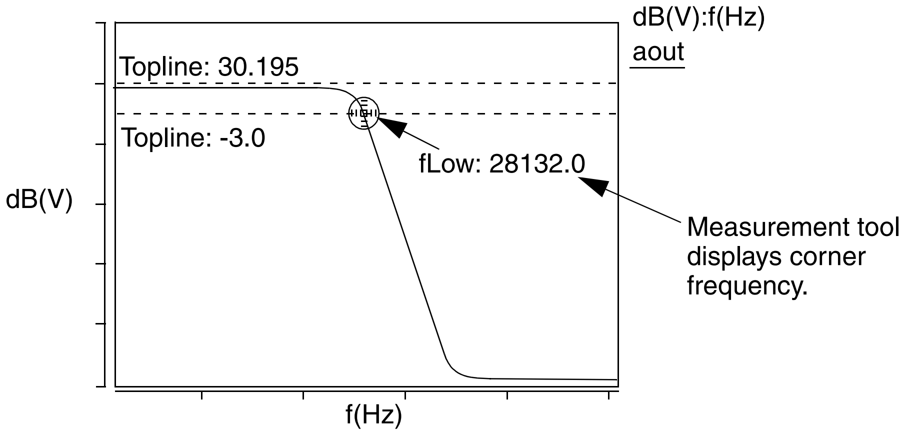
Command Group

Frequency Domain

Syntax

None

Example



Graphical Interface Description

Dialog Box Fields

Reference Levels	If you want to see the topline and/or offset level displayed on the waveform, click on the Visibility Indicator to the right the Topline or Offset field.

Topline

You set this field to a default or a specified level.

Offset

You specify an offset value, to be applied relative to the Topline value. The default is 3. You must also choose which operator to use (–, +, *, or /) along with the specified level. The default is the minus sign. This resulting level is also called the measurement level.

Magnitude

Description

Displays the magnitude of a point on a waveform.

Type of Measured Waveform

Analog (must be complex)

Magnitude Calculation

The magnitude of a waveform is calculated as the absolute value of an argument .

$$\text{mag} = \sqrt{(\text{real}^2 + \text{imag}^2)}$$

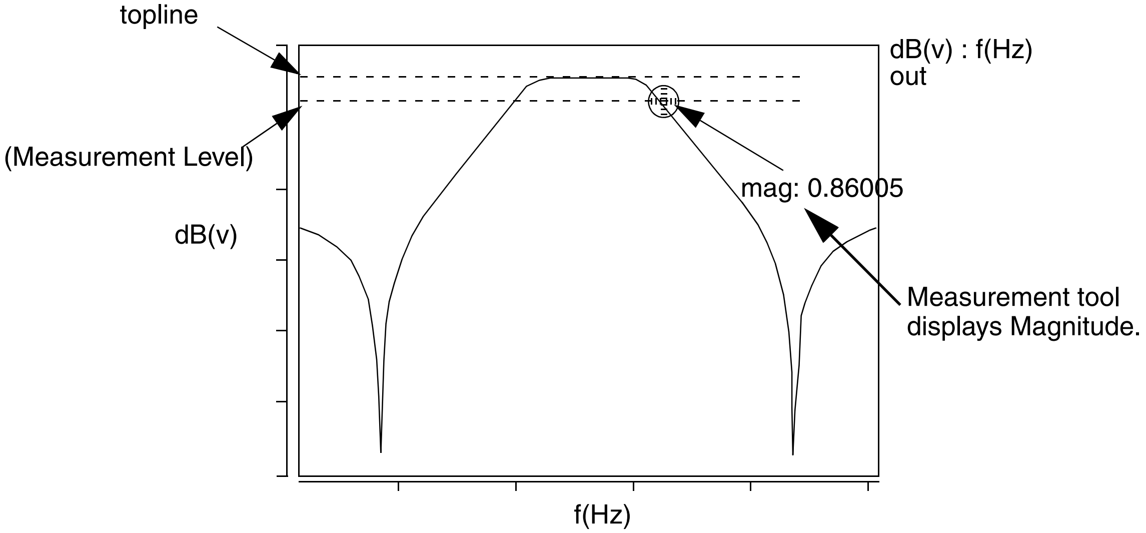
Command Group

Frequency Domain

Syntax

None

Example



Graphical Interface Description

Dialog Box Fields

X Value	Optional. You can provide an X-value and the tool will provide the Y-value at that coordinate. If you do not specify the X-value, a default is used.

Maximum

Description

Displays the maximum value of a waveform.

Type of Measured Waveform

- Analog, event-driven analog, scatter plot, histogram, spectral

Command Group

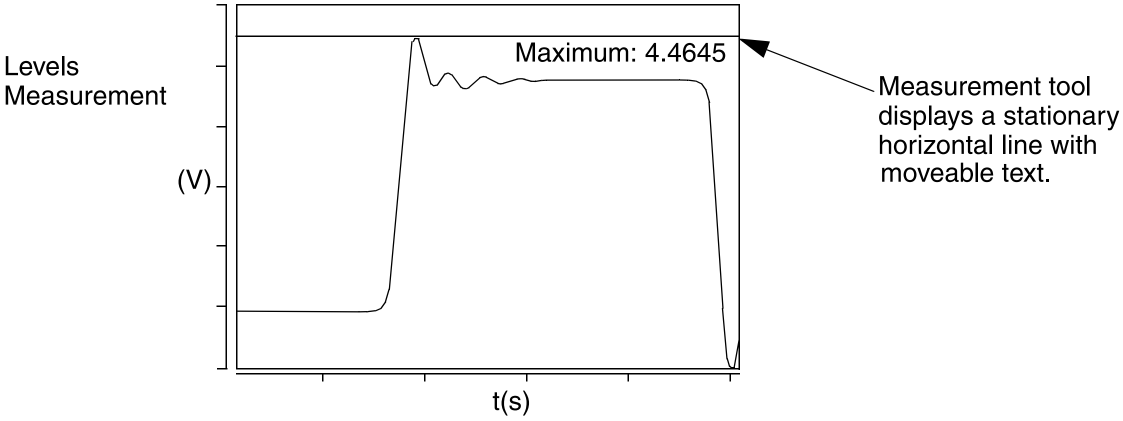
Levels, Statistics

Syntax

None

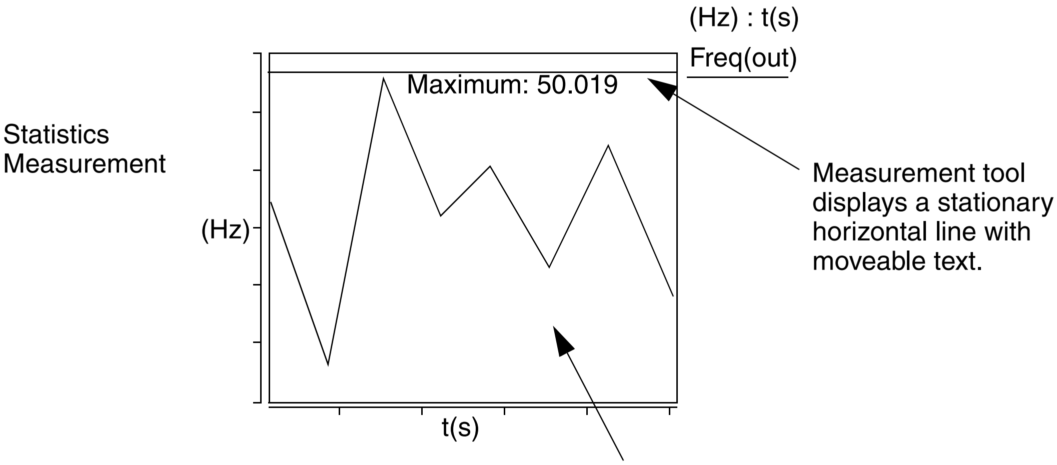
Example

Example 1



Measurement tool displays a stationary horizontal line with moveable text.

Example 2



Measurement tool displays a stationary horizontal line with moveable text.

Earlier measurement produced this waveform showing frequency as a function of time.

Graphical Interface Description

Dialog Box Fields

Category List	This measurement appears in the Statistics or Levels category. All Statistic or Levels category items appear below the Signal field (depending on which category you selected). Select the Maximum item and any other items you want to measure.

Mean

Description

Displays the mean value of a waveform.

Type of Measured Waveform

- Scatter plot, histogram, analog, event-driven analog

Mean Calculation

The mean value of a waveform is calculated as follows:

$$\frac{1}{N}\sum_{j=1}^{N} W_j$$

In this calculation, N is the number of points, and array Wj contains the individual points of the waveform.

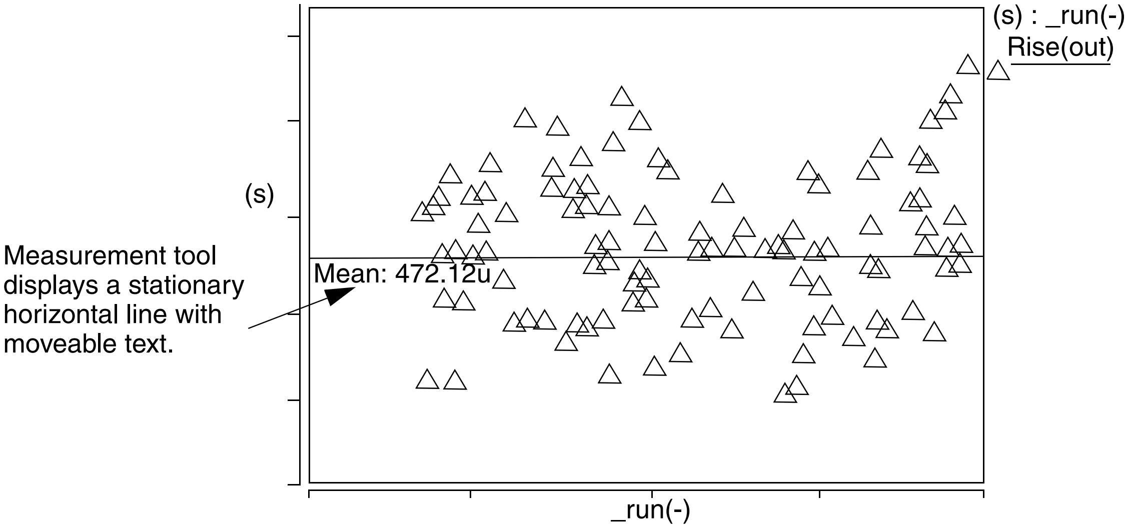
Command Group

Statistics

Syntax

None

Example



Measurement tool displays a stationary horizontal line with moveable text.

Graphical Interface Description

Dialog Box Fields

Category List	All Statistic category items appear below the Signal field. Select the Mean and other items you want to measure.

Mean +3 std_dev

Description

Displays the (mean + 3s) value of a waveform.

Type of Measured Waveform

- Scatter plot, histogram, analog, event-driven analog

Mean +3 std_dev Calculation

The value is calculated as mean + 3s, where mean is the mean value and s is the standard deviation.

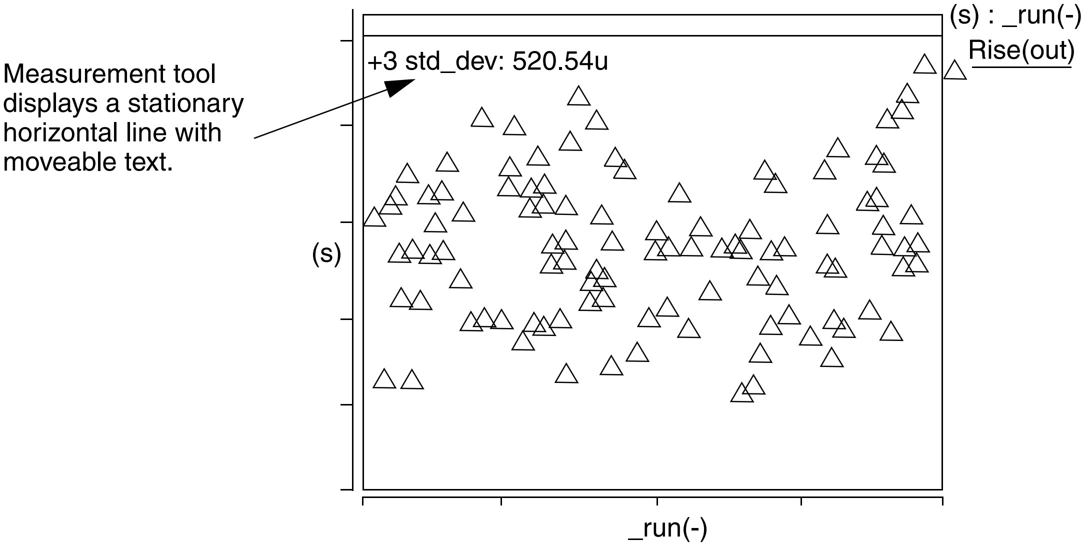
Command Group

Statistics

Syntax

None

Example



Measurement tool displays a stationary horizontal line with moveable text.

Graphical Interface Description

Dialog Box Fields

Category List	All Statistic category items appear below the Signal field. Select the Mean + 3 std_dev item and any other items you want to measure.

Mean -3 std_dev

Description

Displays the (mean – 3s) value of a waveform.

Type of Measured Waveform

- Scatter plot, histogram, analog, event-driven analog

Mean -3 std_dev Calculation

The value is calculated as mean – 3s, where mean is the mean value and s is the standard deviation.

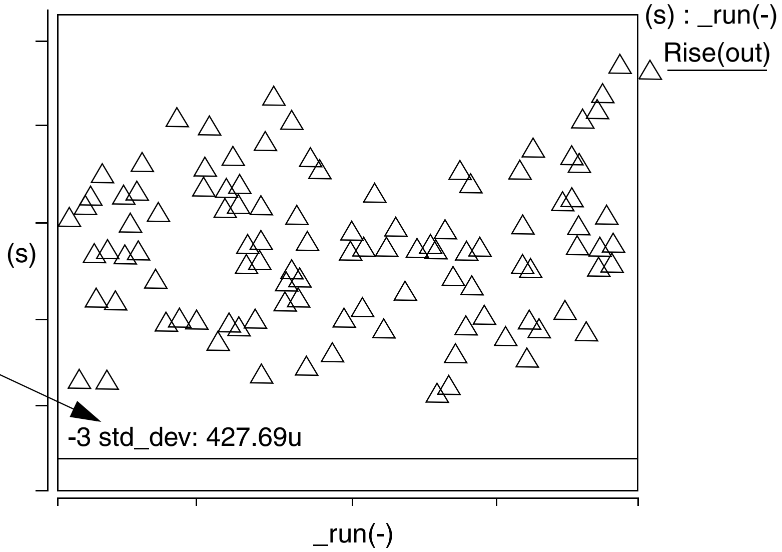
Command Group

Statistics

Syntax

None

Example



Measurement tool displays a stationary horizontal line with moveable text.

Graphical Interface Description

Dialog Box Fields

Category List

All Statistic category items appear below the Signal field. Select the Mean –3 std_dev item and any other items you want to measure.

Median

Description

Displays the median value of a waveform.

Type of Measured Waveform

Scatter plot, histogram, analog, event-driven analog

Median Calculation

The calculated median value represents the Y-axis point at which half of the data points are above and half the points are below the median value.

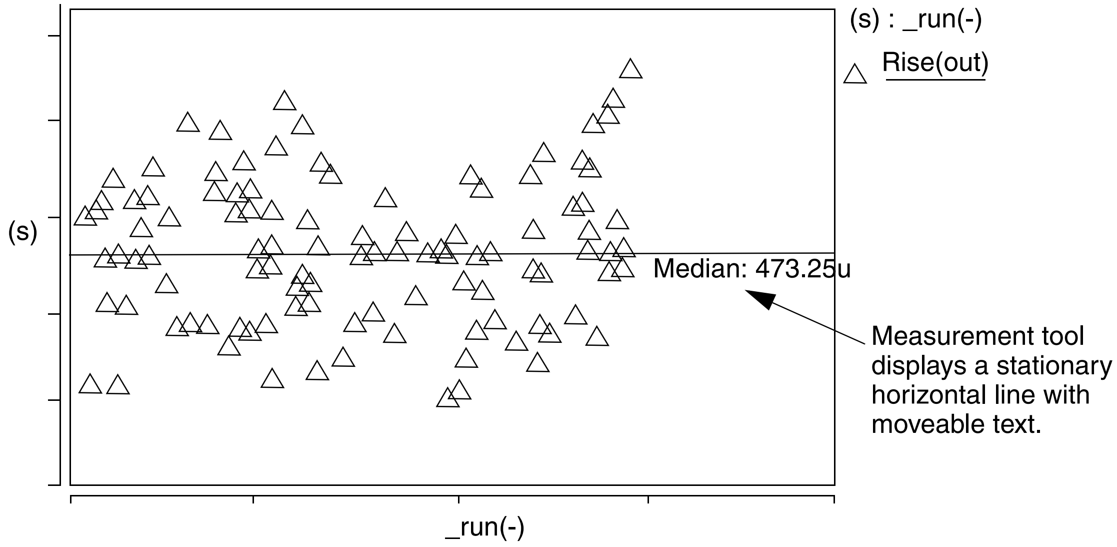
Command Group

Statistics

Syntax

None

Example



Measurement tool displays a stationary horizontal line with moveable text.

Graphical Interface Description

Dialog Box Fields

Category List — All Statistic category items appear below the Signal field. Select the Median item and any other items you want to measure.

Minimum

Description

Displays the minimum value of a waveform.

Type of Measured Waveform

- Analog, event-driven analog, scatter plot, histogram, spectral

Command Group

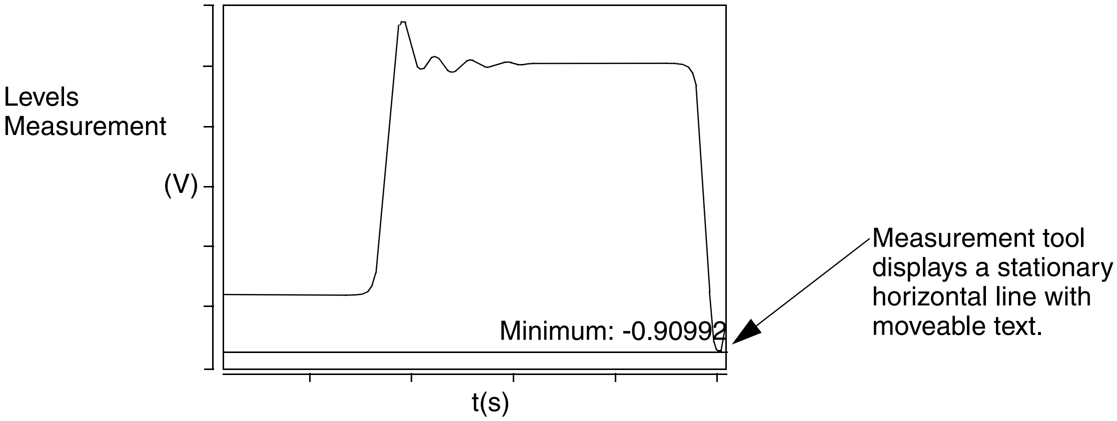
Levels, Statistics

Syntax

None

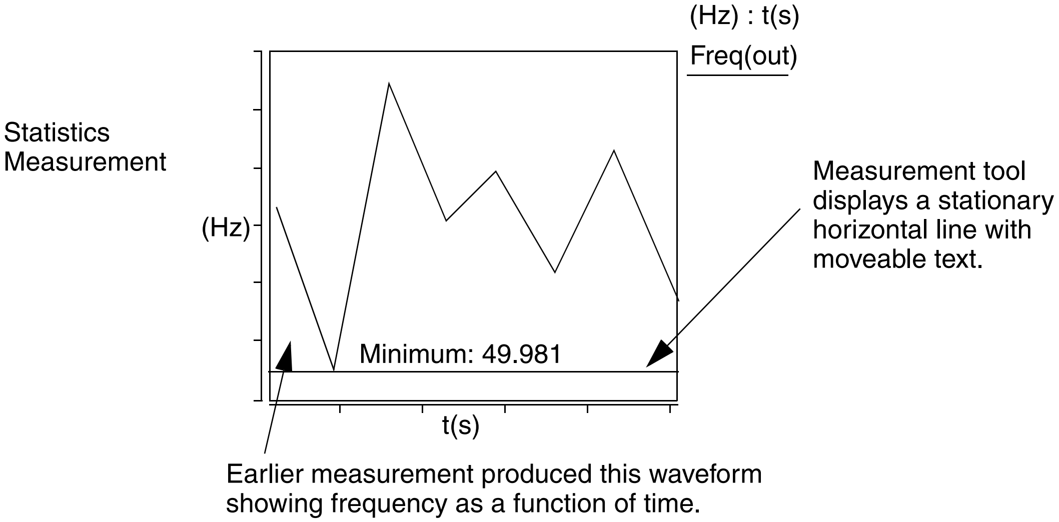
Example

Example 1



Measurement tool displays a stationary horizontal line with moveable text.

Example 2



Measurement tool displays a stationary horizontal line with moveable text.

Earlier measurement produced this waveform showing frequency as a function of time.

Graphical Interface Description

Dialog Box Fields

Category List	This measurement appears in the Statistics or Levels category. All Statistic or Levels category items appear below the Signal field (depending on which category you selected). Select the Minimum item and any other items you want to measure.

Natural Frequency

Description

Displays the natural frequency of a point on a waveform.

Type of Measured Waveform

- Pole zero data, analog (must be complex)

Natural Frequency Calculation

The natural frequency of a waveform is calculated as the absolute value of an argument .

$$\text{natural frequency} = \sqrt{(\text{real}^2 + \text{imag}^2)}$$

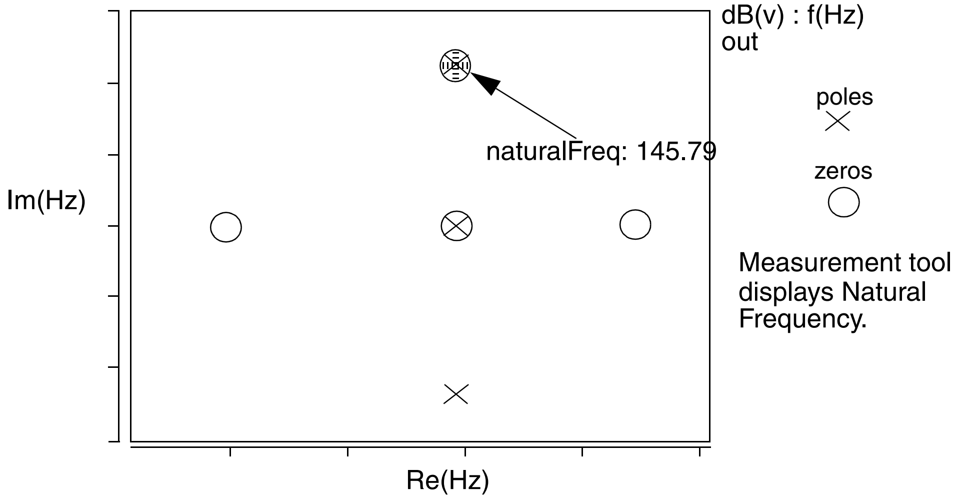
Command Group

s Domain

Syntax

None

Example



Graphical Interface Description

Dialog Box Fields

X Value	Optional. You can provide an X-value and the tool will provide the Y-value at that coordinate. If you do not specify the X-value, a default is used.

Nyquist Plot Frequency

Description

Displays the frequency at a point on a Nyquist (or Nichols) plot.

Type of Measured Waveform

- Analog (must be complex)

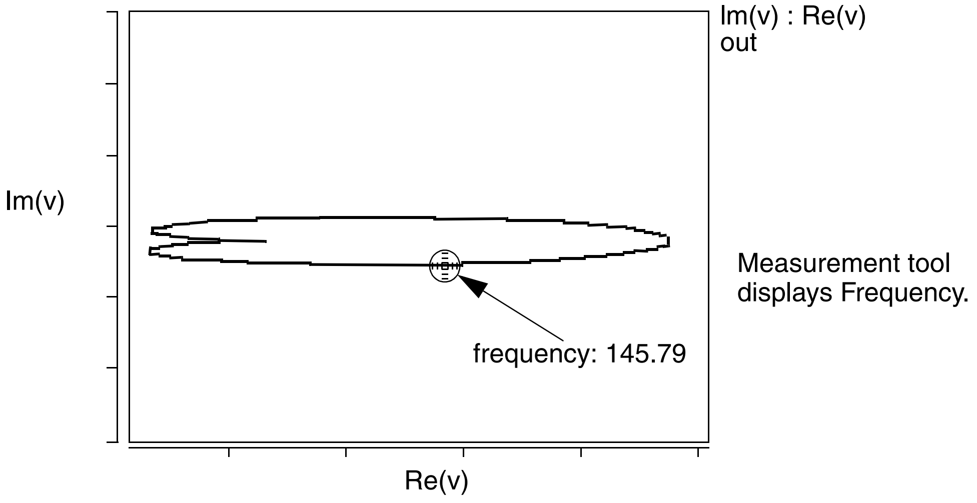
Command Group

Frequency Domain

Syntax

None

Example



Measurement tool displays Frequency.

Graphical Interface Description

Dialog Box Fields

X Value	Optional. You can provide an X-value and the tool will provide the frequency at that coordinate. If you do not specify the X-value, a default is used.

Overshoot

Description

Displays the overshoot of a waveform relative to a default or specified topline.

Type of Measured Waveform

- Analog, event-driven analog

Overshoot Calculation

The overshoot is calculated as the difference between the maximum point on the waveform and the specified (or calculated) Topline value. For more information on how the Topline value is calculated, refer to Topline/Baseline.

Command Group

Time Domain

Syntax

None

Graphical Interface Description

Dialog Box Fields

Reference Levels	You can display this reference level by clicking on the Visibility Indicator at the right of the Topline field. Topline Specify a topline value within the upper and lower Y-axis values, or use the default value. You can display this reference level by clicking on the Visibility Indicator at the right of the Topline field. Baseline Specify a baseline value within the upper and lower y-axis values, or use the default value. You can display this reference level by clicking on the Visibility Indicator at the right of the Baseline field.
Measure Format	Absolute The magnitude of the overshoot is calculated as the absolute value of an argument. Percentage The magnitude of the overshoot is calculated as the percentage of an argument.

P1dB

Description

When the amplitude of the input signal is small enough, it is almost the same as (linear) AC analysis. but when the input signal becomes large, circuit response is saturated and the output amplitude does not linearly increase. The 1dB compression point can be calculated from HB SWEEP results. Sweep the input amplitude and measure the output amplitude.

Type of Measured Waveform

- Analog, Power type, either PowerOut versus PowerIn or PowerOut versus Frequency with PowerIn as a sweep parameter.

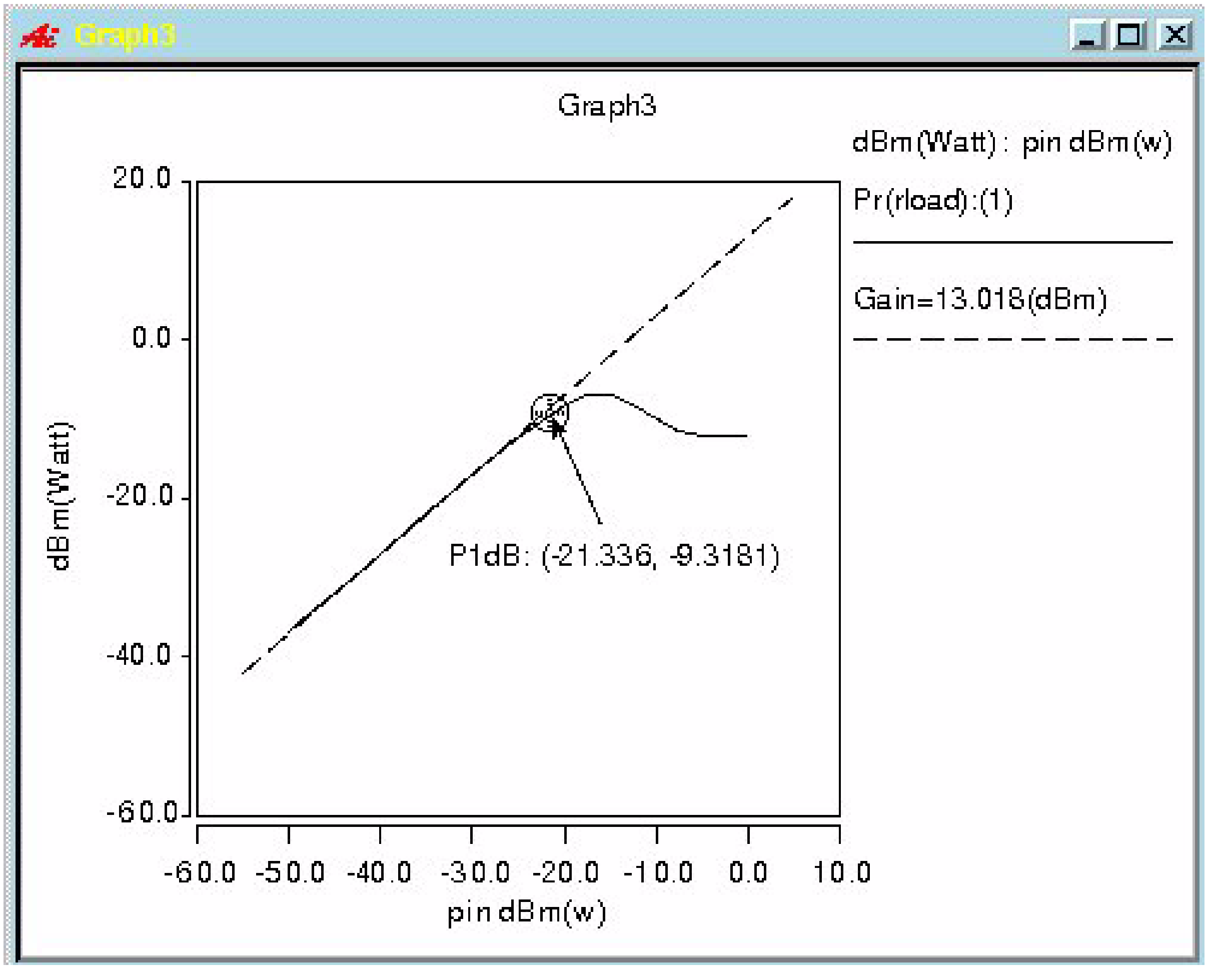
Command Group

RF

Syntax

None

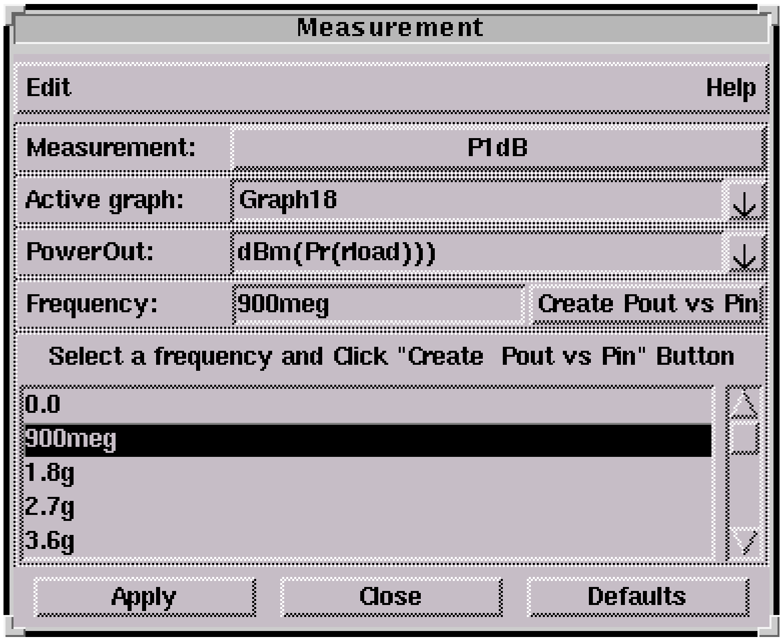
Example



Given one waveform, PowerOut versus Frequency with PowerIn as a sweep parameter, the following Dialog Box Fields show:

PowerOut	Waveform of PowerOut versus Frequency with PowerIn as a sweep parameter
Frequency	Frequency to generate PowerOut versus PowerIn curve

Examples



Measurement: P1dB

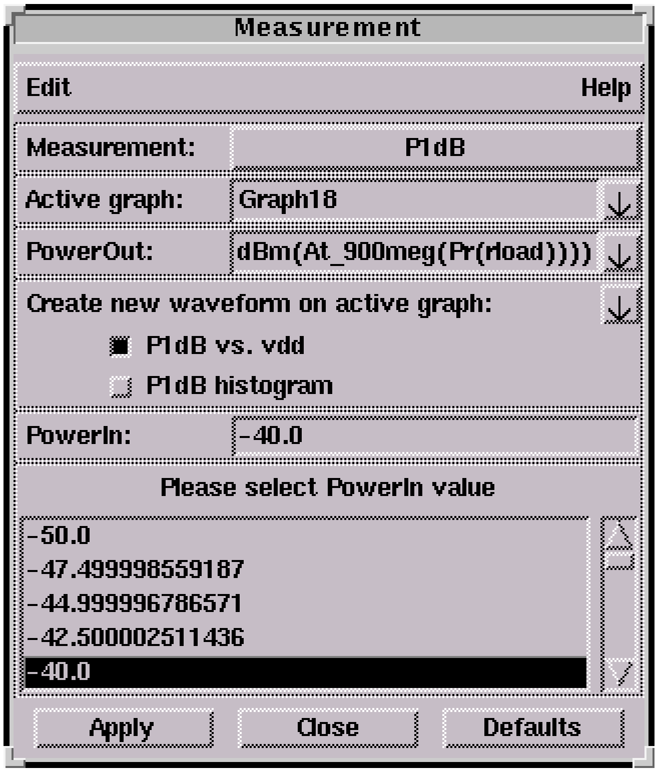
Active graph: Graph18

PowerOut: dBm(Pr(rload)))

Frequency: 900meg

Select a frequency and Click "Create Pout vs Pin" Button

0.0
900meg
1.8g
2.7g
3.6g



Measurement: P1dB

Active graph: Graph18

PowerOut: dBm(At_900meg(Pr(rload))))

Create new waveform on active graph:

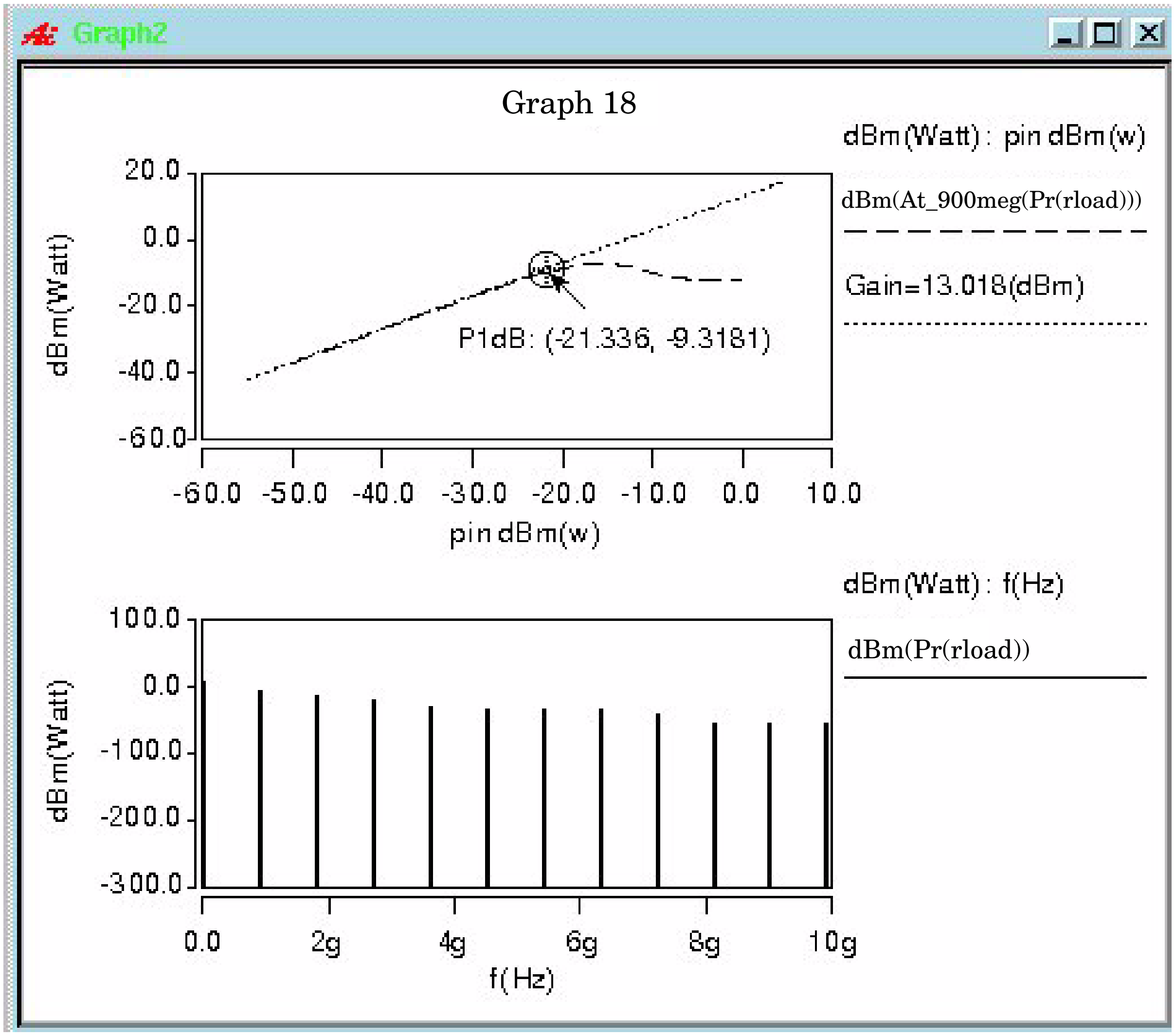
■ P1dB vs. vdd

□ P1dB histogram

PowerIn: -40.0

Please select PowerIn value

-50.0
-47.499998559187
-44.999996786571
-42.500002511436
-40.0



Graphical Interface Description

Dialog Box Fields

Given on waveform, PowerOut versus PowerIn, Dialog Box Fields show:

PowerOut	PowerOut versus PowerIn curve
PowerIn	PowerIn value

Pareto

Description

Pareto Analysis is a simple method for separating the major causes (the “vital few”) of a problem, from the minor ones (“trivial many”).

Type of Measured Waveform

- Multi-member

Pareto Calculation

The sensitivity histogram is similar in concept to the results from a Saber sensitivity analysis, except that the values from a Saber sensitivity analysis will be based on varying one parameter at a time, while the sensitivity histogram from the Pareto analysis accounts for the variation of all parameters at once. In addition, the size of the parameter change for a Saber sensitivity is a small percentage of the nominal parameter value, while the change from the Monte-Carlo analysis is related to the distribution of the parameter and is typically much larger. These are important distinctions for nonlinear circuits where two or more parameters may interact to influence the outputs being measured. A Saber sensitivity analysis cannot account for these influences.

The Pareto sensitivity values are calculated by fitting a straight line to the measured output versus each of the parameters. The normalized slope is the sensitivity value displayed in the histogram. The values in the histogram can be interpreted in the same way as the Saber sensitivity numbers. For example, if the sensitivity is -0.7 for the parameter rnom(r.r1) than a 1% change in rnom(r.r1) will lead to a -0.7% change in the measured output.

The R**2 (coefficient of determination) histogram is used in conjunction with the sensitivity histogram (which is why it is displayed on the same graph) and is a measure of the goodness of fit of the line that is used to calculate the sensitivity for each parameter. The R**2 values will always be between 0 and 1.

A value of R**2 close to 1 for a particular parameter indicates that the fit of the line is good and that there is a strong linear relationship between the measured value and that parameter, meaning that a change in that parameter will affect the measured value.

A value of R**2 close to 0 indicates that the fit is not good and that there is not a strong linear relationship between the measured value and that parameter, meaning that a change in that parameter will have little or no effect on the measured value.

The parameters with both a high sensitivity and a relatively high R**2 value are the ones that will have the most affect on the measured value. In other words, these are the parameters that should have a tighter tolerance to control the amount of variation in the output. In addition, these parameters can be used to change the value of the measured output. Answering the question of what is considered a high R**2 value is not easy since it depends on the interactions in the circuit. In practice, it is not uncommon for all parameters to have R**2 values less than 0.7. This does not mean that the parameters are not important, but is likely because two or more parameters interact to affect the measured output.

The scatter plots give more detailed information than is available from the R**2 and sensitivity histograms. Each scatter plot shows the measured values versus the parameter values. The best fit line through the data and the R**2 value are displayed. The more closely the points follow the best fit line, the higher the R**2 value. An R**2 value of 1 would mean that all of the points would be exactly on the line. Values of R**2 above approximately 0.7 will still show the scatter points to follow the tendency of the line. Low R**2 values, ones below approximately 0.2, will appear to be randomly placed on the graph. The slope of the best fit line is directly related to the sensitivity so the higher the slope, the more sensitive the measured value is to changes in this parameter. Of course, a sensitivity with a high slope is not very meaningful if the R**2 value is low.

Command Group

Statistics

Syntax

None

Example

Pareto Example

The following is a typical use case for the Pareto measure using the Saber Simulator:

1. Load a design into Saber and run a Monte-Carlo analysis. Be sure to save the parameters into a parameter file since they will be needed for Pareto.
2. Plot in CosmosScope one or more of the signals from the design.
3. Perform one or more measures on the signals. For example, you may be interested in the rise time at the output as well as the maximum power dissipation for a specific part. In this case, you would perform a rise time

measure on the output signal and a maximum measure on the power dissipation signal for the device. Each of these measures will create a new scatter plot waveform in CosmosScope.

4. Pareto can be run individually on both of these resulting scatter plots. Select the Pareto measure in the Measure tool and then select the signal to be used for the analysis.
5. On the Measure tool in the field labeled Parameter Plot File, type in the name of the plot file where the parameter values were saved from the Monte-Carlo analysis. In most cases it will be easier to use the browse button to the right of the entry field and select the parameter file directly from the file browser dialog box.
6. Optionally select which parameters are to be used by the Pareto analysis. The default is to use all parameters from the parameter plot file that had a variation in the Monte-Carlo analysis. These are referred to as the statistical parameters. If you want to choose a subset of these parameters, you can type in the parameter names directly or you can use the browse button on the right side of the entry field. In almost all cases, using the default, All Statistical Parameters, is recommended.
7. Select the type of output from the measure. There are 3 choices:
    - R**2 and Sensitivity Histograms
    - Scatter Plots
    - Save to File

Graphical Interface Description

Dialog Box Fields

Parameter Plot File — The plot file where the parameter values were saved from the multi-member analysis. Using the downward pointing arrow allows you to use the Open Parameter File dialog box.

Parameter Names — Select All Statistical Parameters, type in individual parameters, or pick parameters with the Browse in Plot File option. Using the downward pointing arrow displays the options for this field.

R**2 and Sensitivity Histograms — Creates a new graph that contains two histograms, one labeled R**2 (the coefficient of determination) and the other labeled Sensitivity.

Scatter Plots — Generates a scatter plot for all of the important parameters showing the measured values versus the parameter values.

Save to File

Allows all of the Pareto results to be saved to a text file. By default the file name is pareto.txt. Using the downward pointing arrow allows you to save the file in a specific directory with a specific name.

Minimum R**2 to Display

Specify the minimum values for R**2 (the coefficient of determination). The default value is 0.1. The histograms and scatter plots will only show those parameters that have both a sensitivity value and an R**2 value greater than these values. Changing these values to 0.0 will show the effect of all of the parameters in the plot file. Depending on the design, this may result in a large number of bars in the sensitivity and R**2 histograms and a large number of scatter plots. The minimum R**2 and minimum sensitivity values do not affect the results written to the text file if that option is selected.

Minimum Sensitivity to Display

Specify the minimum values for Sensitivity. The default value is 0.1. The histograms and scatter plots will only show those parameters that have both a sensitivity value and an R**2 value greater than these values. Changing these values to 0.0 will show the effect of all of the parameters in the plot file. Depending on the design, this may result in a large number of bars in the sensitivity and R**2 histograms and a large number of scatter plots. The minimum R**2 and minimum sensitivity values do not affect the results written to the text file if that option is selected.

Peak-to-Peak

Description

Displays the waveform’s peak-to-peak value.

Type of Measured Waveform

- Analog, event-driven analog, scatter plot

Peak-to-Peak Calculation

The peak-to-peak value is calculated as the difference between the maximum and minimum values of the waveform.

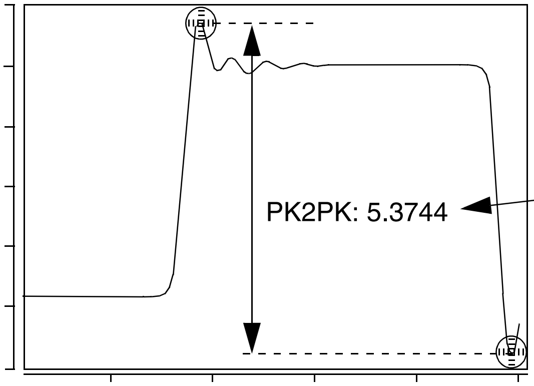
Command Group

Levels

Syntax

None

Example



Measurement tool displays peak-to-peak

Graphical Interface Description

Dialog Box Fields

Category List	All Levels category items appear below the Signal field. Select the Peak to Peak item and any other items you want to measure.

Period

Description

Displays the period of a periodic waveform relative to a default or specified topline and baseline levels.

Type of Measured Waveform

- Analog, event-driven analog, digital

Possible Errors

An error is reported if the waveform does not contain at least one complete cycle.

Period Calculation

The period is calculated as the difference in time between two consecutive edges of the waveform of the same polarity as shown in the example. First, a rising or falling edge (depending on your trigger setting) is found.

A rising edge starts below the lower level of the waveform and rises through the middle level to a value above the upper level. A falling edge starts above the upper level and falls through the middle level to a value below the lower level. For more information about the upper, middle, and lower reference levels, refer to "Waveform Reference Levels".

The waveform is then searched from the selected rising or falling edge to find the next edge of the same polarity. The period is calculated as the difference in time between the middle crossings of the two edges.

Command Group

Time Domain

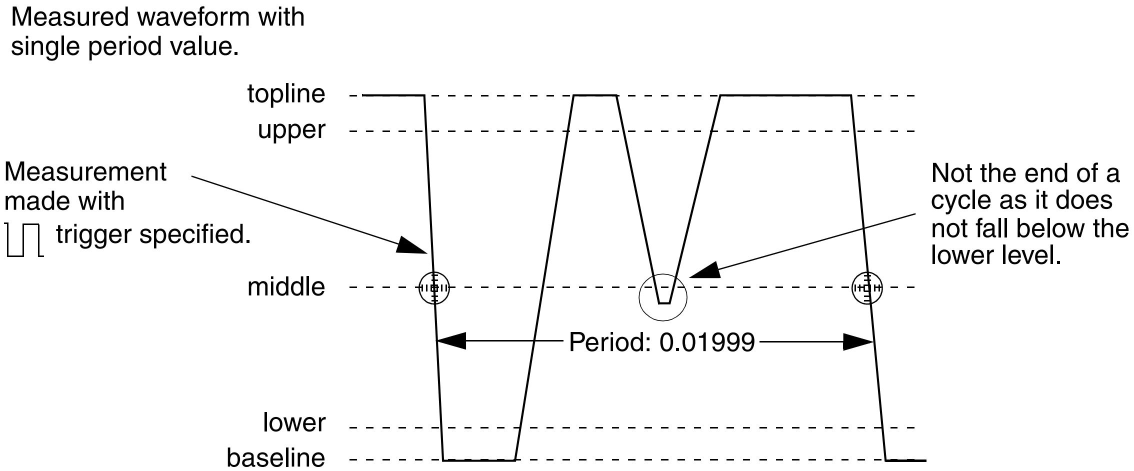
Syntax

None

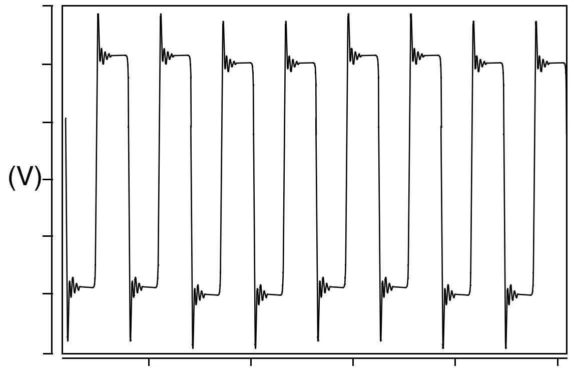
Example

Example 1

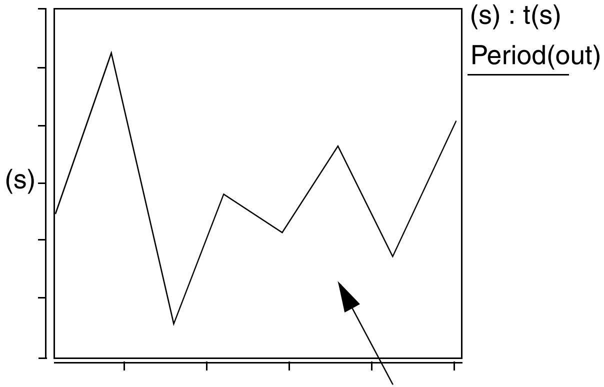
Example 1 shows that the circled portion of the waveform is not considered an edge since it does not fall below the lower level of the waveform.


Measured waveform with single period value.
topline
upper
Measurement made with trigger specified.
Not the end of a cycle as it does not fall below the lower level.
middle
Period: 0.01999
lower
baseline

Example 2



Measured waveform with multiple period values.



Measurement tool produces this waveform showing period as a function of time.

Graphical Interface Description

Dialog Box Fields

Reference Levels

The following two fields set the topline and baseline levels for the measured signal and the corresponding reference signal. You can display any of these levels on the waveform by clicking on the corresponding Visibility Indicator to the right of each field.

Topline

On the measured signal, specify a topline value

Baseline

On the measured signal, specify a baseline value below the topline value, or use the default value.

Trigger

⊓⊔/⊔⊓ Specifies that the measurement starts from a period with either a rising or falling edge.

⊓⊔ Specifies that the measurement starts from a period with a rising edge.

⊔⊓ Specifies that the measurement starts from a period with a falling edge.

Create New Waveform on Active Graph or New Graph

Period vs. t - A new waveform is computed with period (y-axis) versus time (x-axis). See Example 2.

Phase

Description

Displays the phase value on a point on a waveform.

Type of Measured Waveform

- Analog (must be complex)

Phase Calculation

The phase of a waveform is calculated as atan(imag/real).

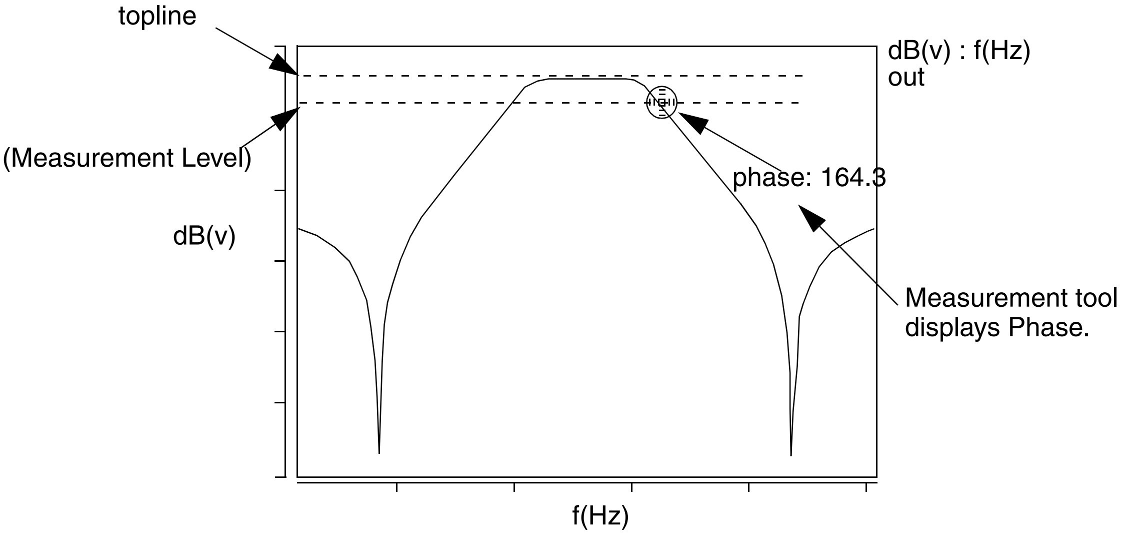
Command Group

Frequency Domain

Syntax

None

Example



Graphical Interface Description

Dialog Box Fields

X Value	Optional. You can provide an x-value and the tool will provide the y-value at that coordinate. If you do not specify the x-value, a default is used.

Units	You can select radians or degrees as the units of measurement. If you do not specify, degrees is the default unit.

Phase Margin

Description

Displays the phase margin of a complex waveform in degrees or radians.

Type of Measured Waveform

- Analog (must be complex)

Possible Errors

An error is reported if the magnitude of the waveform does not pass through 0 dB or if the waveform is not complex.

Phase Margin Calculation

The phase margin is defined as the difference between the phase of the measured waveform and –180 degrees at the unity gain frequency.

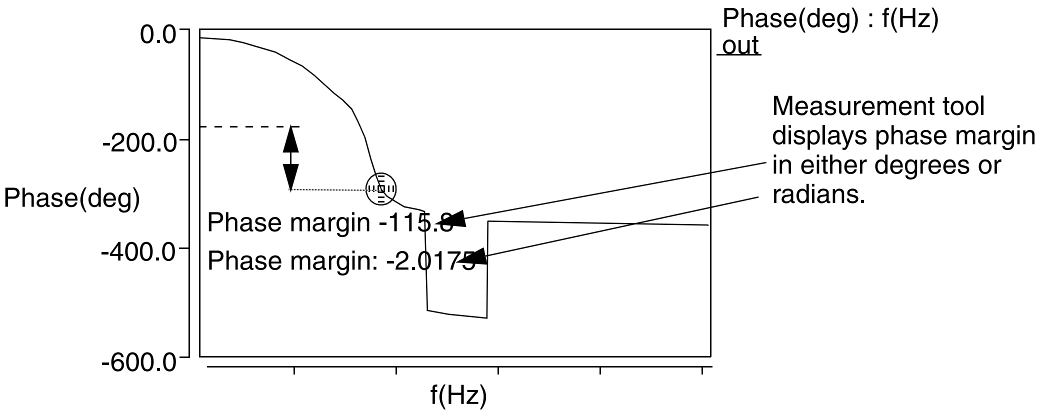
Command Group

Frequency Domain

Syntax

None

Example



Measurement tool displays phase margin in either degrees or radians.

Graphical Interface Description

Dialog Box Fields

Units	You can select radians or degrees as the units of measurement. If you do not specify, degrees is the default unit.

Point Marker

Description

Displays a moveable point marker on the waveform to display the x-value and y-value.

Type of Measured Waveform

- Analog, event-driven analog, digital, scatter plot, histogram, spectral

Command Group

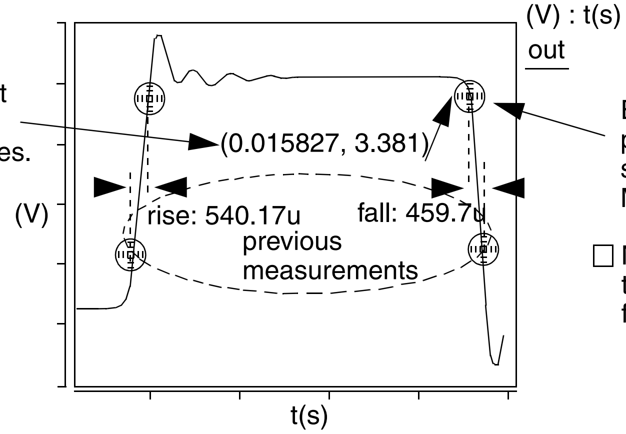
General

Syntax

None

Example

Measurement tool displays x,y-coordinates.



Example measure shows point marker anchor snapping to “Closest Measurement.”

□ Move point marker anchor to other existing anchors to find other x,y-coordinates.

Graphical Interface Description

Dialog Box Fields

Anchor Snap

Signal

Generates a moveable anchor point that snaps to the signal waveform. To see an example, refer to the Delta X Example.

Closest Measurement

When you have multiple anchor points visible on a graph from previous measurements, this setting causes the measurement to snap to one of those nearest points. If there is no visible measurement on the active graph from which to snap, an error message is displayed. See the Example.

Floating

Generates a moveable anchor point that can be positioned anywhere within the graph. To see an example, refer to the Delta Y Example.

Location (Optional) X Value

You can optionally specify an x-value that is used to determine the corresponding y-value.

Point to Point

Description

Display the following values between two X-axis points on one or two waveforms:

1. X and Y for the first point
2. X and Y for the second point
3. X-value difference between the two points
4. Y-value difference between the two points
5. Length of a straight line that connects two points
6. Slope of the two points.

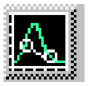
If two waveforms are selected, the two waveforms do not need to be the same type, but they must be in the same graph region.

Type of Measured Waveform

- Analog only

Short Cut Icon

A short cut Icon for the Point to Point Measurement is also available:



When selecting a signal or signals from a graph region, measurement results of X and Y for the first point, X and Y for the second point, and DeltaX, DeltaY, Length, and Slope values.

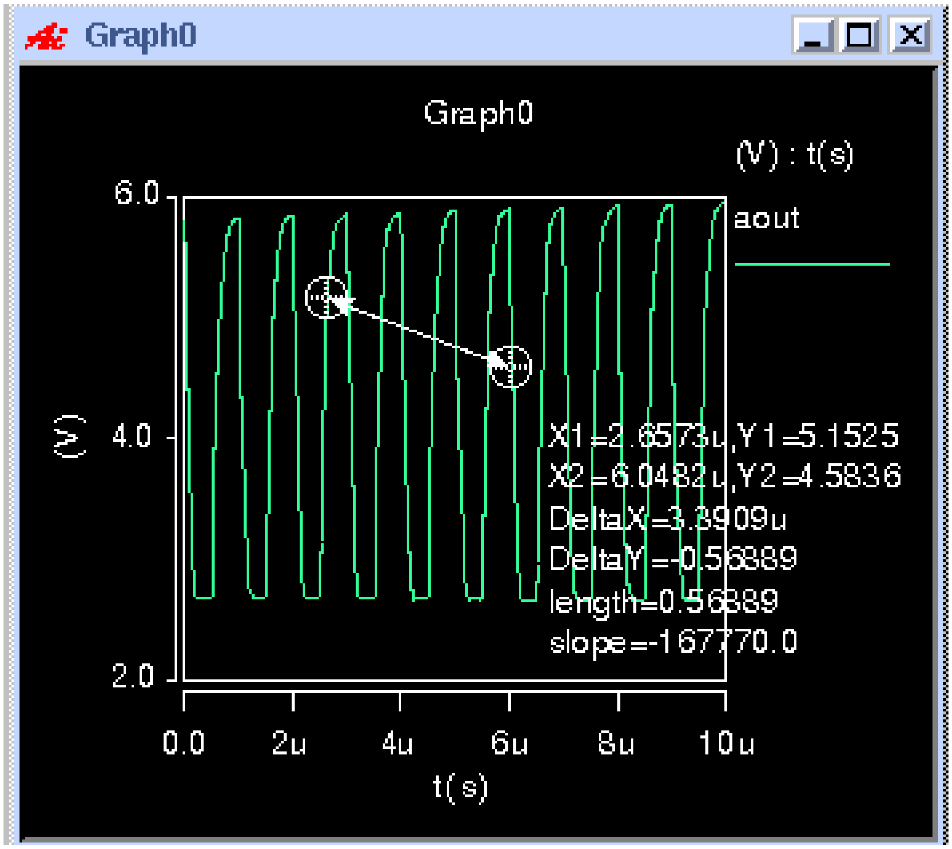
Command Group

General

Syntax

None

Example



```
X1=2.6573u,Y1=5.1525
X2=6.0482u,Y2=4.5836
DeltaX=3.3909u
DeltaY=-0.56889
length=0.56889
slope=-167770.0
```

Graphical Interface Description

Dialog Box Fields

Number of Signals	Select the number of signals to apply the measurement: 1 or 2.
Signal(s)	The signals from the active graph window are listed; select the desired signal(s) to measure. When measuring 2 signals, select from the pulldown list the Signal and Ref. Signal; a **Swap** button appears to allow you to reverse the measurement operation.
Location (Optional)	Specify, if desired, two X-values used to determine the Y-values. Adjust the X1 and X2 Value entries by moving the anchor point after making the initial measurement.

Check boxes: X1, Y1; X2, Y2; DeltaX; DeltaY; Length; Slope	Check the boxes to choose the values you want to display. All the boxes are checked by default.
Apply Measurement to:	Select from Entire Waveform or for the Visible X and Y range only.

Pulse Width

Description

Displays the pulse width of a waveform relative to default or specified topline and baseline levels.

Type of Measured Waveform

- Analog, event-driven analog, digital

Pulse Width Calculation

To be considered for a pulse width measurement, a pulse must rise above the upper level and fall below the lower level as shown in Example 1. The pulse width is measured at the middle level of the waveform. For more information about reference levels of a waveform, refer to "Waveform Reference Levels".

The pulse width is calculated as the difference in time between the middle level of a rising edge and the middle level of the next falling edge on the waveform.

Command Group

Time Domain

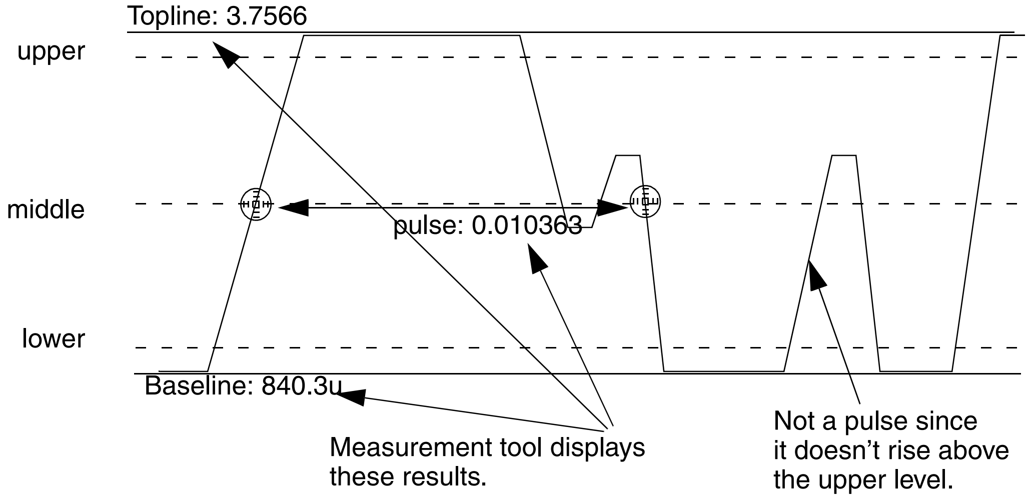
Syntax

None

Example

Example 1

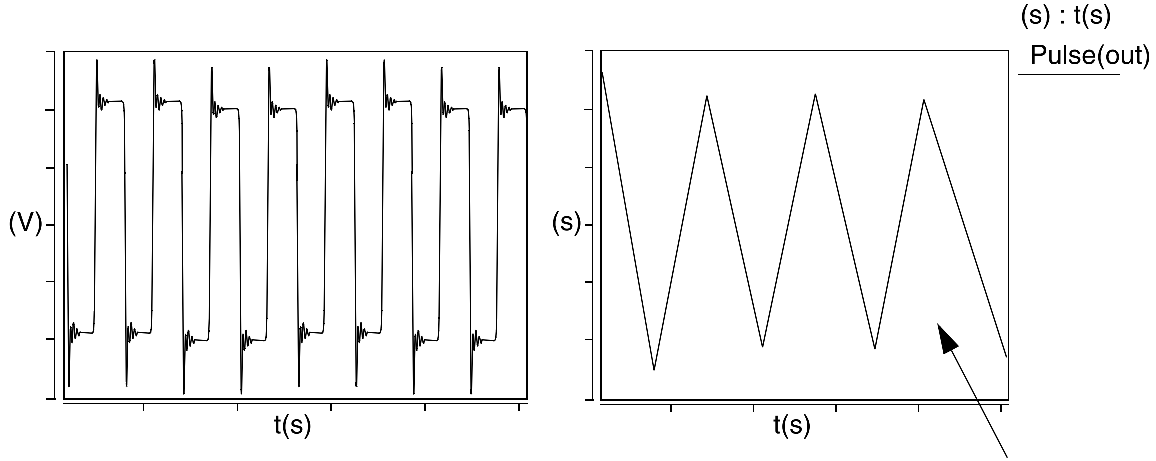
Measured Waveform with
Single pulse width value.



Measurement tool displays
these results.

Not a pulse since
it doesn't rise above
the upper level.

Example 2



Measured waveform with
multiple pulse width values.

Measurement tool produces this waveform
showing pulse width as a function of time.

Graphical Interface Description

Dialog Box Fields

Reference Levels

The following two fields set the topline and baseline levels for the measured signal and the corresponding reference signal. You can display any of these levels on the waveform by clicking on the corresponding Visibility Indicator to the right of each field.

Topline

Specify a topline value or use the default value.

Baseline

Specify a baseline value below the topline value or use the default value.

Pulse Type

⊓ Specifies that the measurement should find positive pulses.

⊔ Specifies that the measurement should find negative pulses.

Create New Waveform on Active Graph or New Graph

Pulse Width vs. t

A new waveform is computed with the pulse width values (y-axis) versus time (x-axis). See Example 2.

Quality Factor

Description

Displays the quality factor of a point on a waveform.

Type of Measured Waveform

- Pole zero data, complex set

Quality Factor Calculation

The quality factor of a waveform is calculated as 1/2(damping ratio).

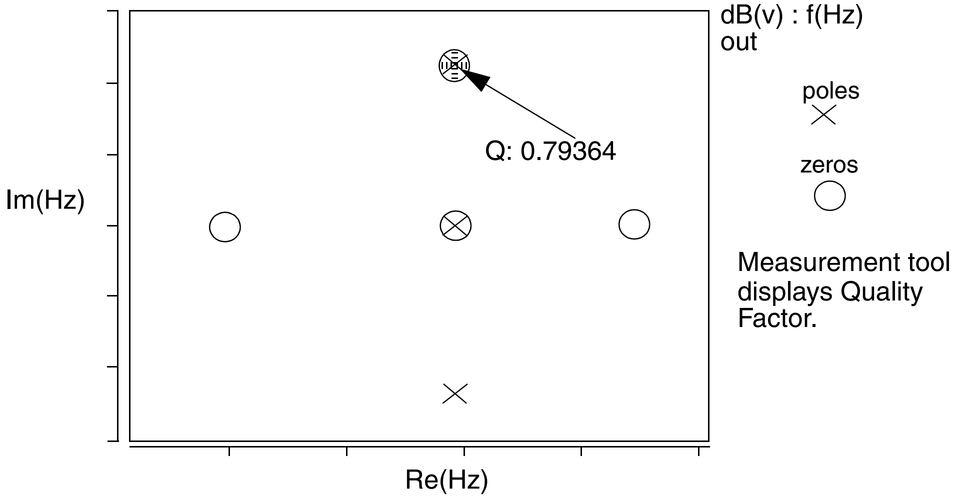
Command Group

s Domain

Syntax

None

Example



Graphical Interface Description

Dialog Box Fields

X Value	Optional. You can provide an x-value and the tool will provide the y-value at that coordinate. If you do not specify the x-value, a default is used.

Range

Description

Displays the range of y-axis values covered by the waveform.

Type of Measured Waveform

- Scatter plot, histogram, analog, event-driven analog

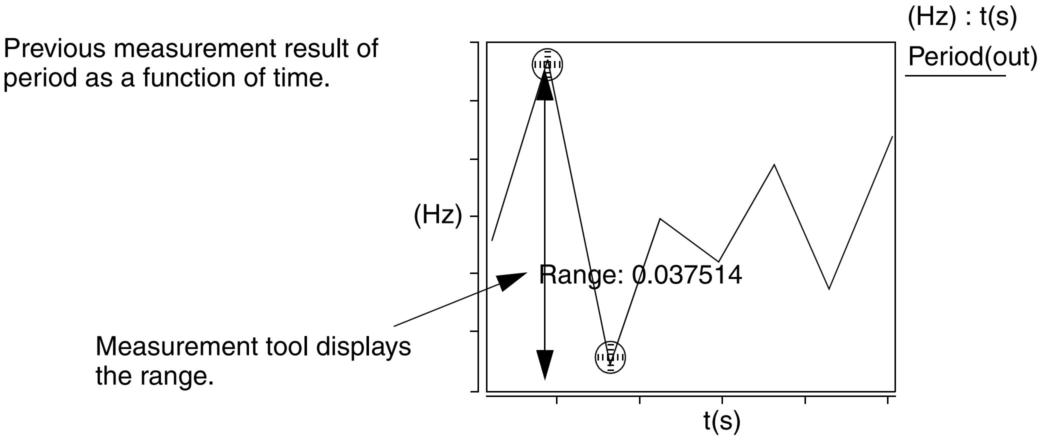
Command Group

Statistics

Syntax

None

Example



Graphical Interface Description

Dialog Box Fields

Category List	All Statistic category items appear below the Signal field. Select the Range item and any other items you want to measure.

Real

Description

Displays the real value of a point on a waveform.

Type of Measured Waveform

- Analog (must be complex)

Real Calculation

The real value of a waveform is the real part of a complex argument. If there is no real part then the value 0.0 is returned.

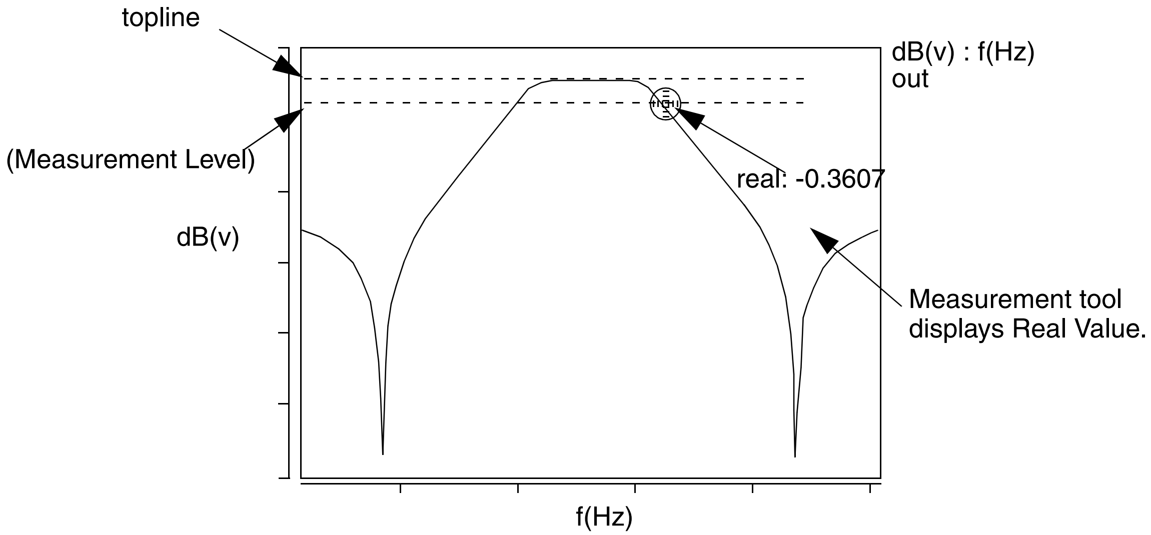
Command Group

Frequency Domain

Syntax

None

Example



Graphical Interface description

Dialog Box Fields

X Value	Optional. You can provide an x-value and the tool will provide the y-value at that coordinate. If you do not specify the x-value, a default is used.

Risetime

Description

Displays the risetime between default or selected upper and lower levels of a waveform. You can also compute the risetime based on manually-set upper/ lower levels as described in the topic titled "Manually Set a Custom Topline/ Baseline".

Type of Measured Waveform

- Analog

Possible Errors

An error is reported if the waveform contains no rising edges.

Risetime Calculation

The risetime is calculated by finding a crossing with the middle level of the waveform. Looking forward from this point, the time when the waveform rises to the upper level is found. Looking backward, the time when the waveform falls to the lower level is found. The difference in the times is the risetime.

For more information about the lower, middle, and upper reference level of a waveform, refer to "Waveform Reference Levels".

Command Description

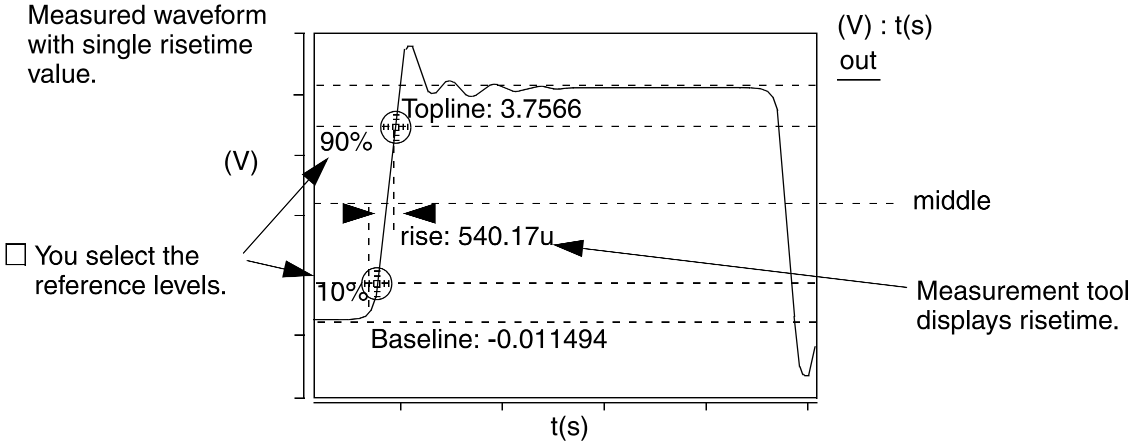
Time Domain

Syntax

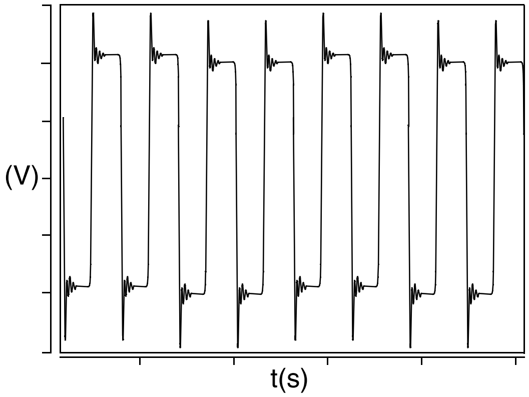
None

Example

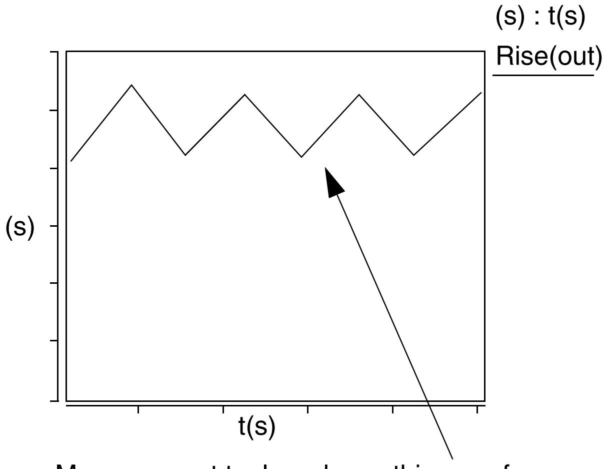
Example 1



Example 2



Measured waveform with multiple risetime values.



Measurement tool produces this waveform showing risetime as a function of time.

Graphical Interface Description

Dialog Box Fields

Reference Levels

The following two fields set the topline and baseline levels for the measured signal and the corresponding reference signal. You can display any of these levels on the waveform by clicking on the corresponding Visibility Indicator to the right of each field.

Topline

Specify a topline value or use the default value.

Baseline

Specify a baseline value below the topline value or use the default value.

0-100%
10-90%
20-80%

Click on one of these buttons to set an upper and lower range (in percent) relative to the topline/baseline levels. To compute a risetime based on a different percentage level than the defaults, refer to the topic titled "Manually Set a Custom Topline/Baseline".

Create New Waveform on Active Graph or New Graph

Risetime vs. t

Creates a new waveform with risetime (y-axis) versus time (x-axis). See Example 2.

RMS

Description

Displays the RMS value of a waveform.

Type of Measured Waveform

- Analog, event-driven analog, scatter plot, histogram

RMS Calculation

In this calculation, W represents the waveform and x1 and x2 represent the starting and ending points.

$$\left[\frac{1}{x2-x1}\int_{x1}^{x2}(W^{2}dx\right]^{1/2}$$

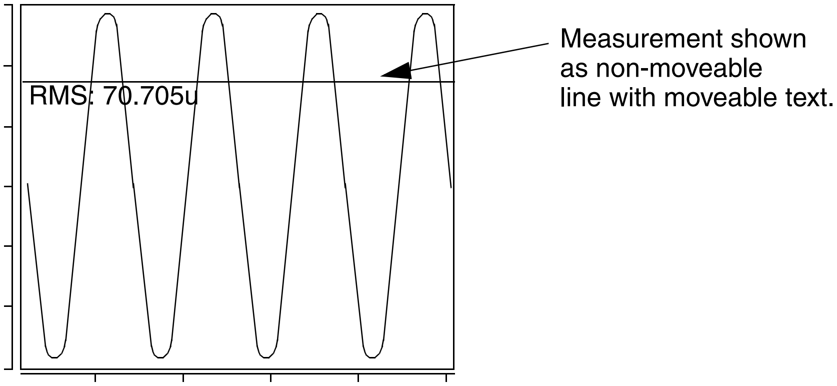
Command Group

Levels

Syntax

None

Example



Graphical Interface Description

Dialog Box Fields

Category List	All Levels category items appear below the Signal field. Select the RMS item and any other items you want to measure.

Settle Time

Description

Displays the settle time of a waveform with respect to a default or specified settle level and a specified settle band.

Type of Measured Waveform

- Analog, event-driven analog

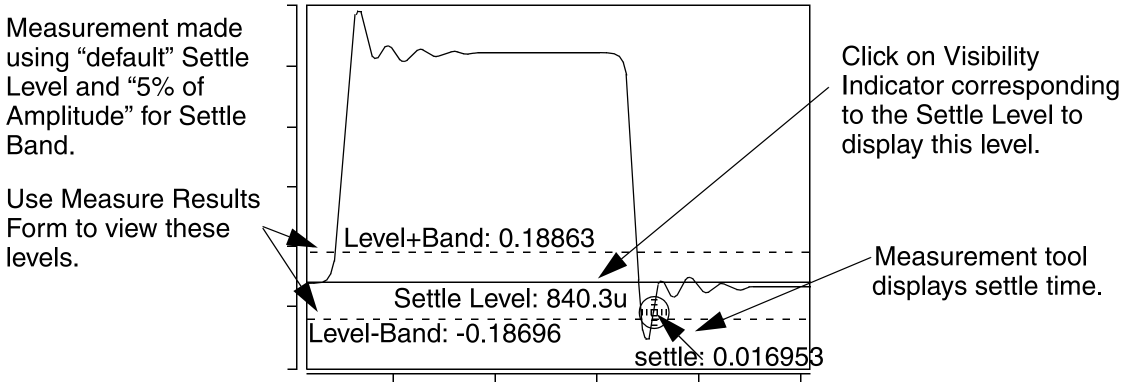
Command Group

Time Domain

Syntax

None

Example



Measurement made using “default” Settle Level and “5% of Amplitude” for Settle Band.

Use Measure Results Form to view these levels.

Click on Visibility Indicator corresponding to the Settle Level to display this level.

Measurement tool displays settle time.

Graphical Interface Description

Dialog Box Fields

Settle Level	You set a settle level or let the Measure tool calculate a default.

Settle Band You choose the size of the settle band on either side of the settle level. The default is 5 percent of the amplitude. Other choices are as follows:

Amplitude - % of amplitude of the waveform

Settle Level - % of the settled level of the waveform

Peak to Peak - % of the peak-to-peak value of the waveform

Absolute - an absolute value such as 0.3

Slew Rate

Description

Displays the slew rate of a waveform relative to default or specified topline and baseline levels.

Type of Measured Waveform

- Analog, event-driven analog

Slew Rate Calculation

The slew rate is calculated as the difference between the upper and lower levels of a waveform divided by the risetime or falltime of the edge. You select the upper and lower levels as a percent of topline/baseline.

For more information about the upper and lower reference level of a waveform, refer to "Waveform Reference Levels".

For more information on how risetime is calculated, refer to the Risetime Calculation. For more information on how falltime is calculated, refer to the Falltime Calculation.

Command Group

Time Domain

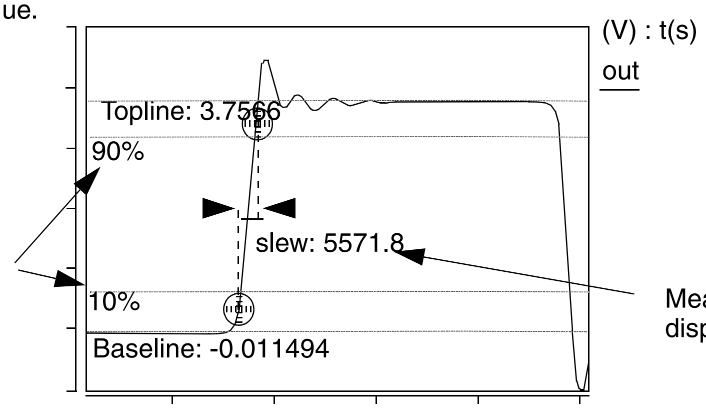
Syntax

None

Example

Example 1

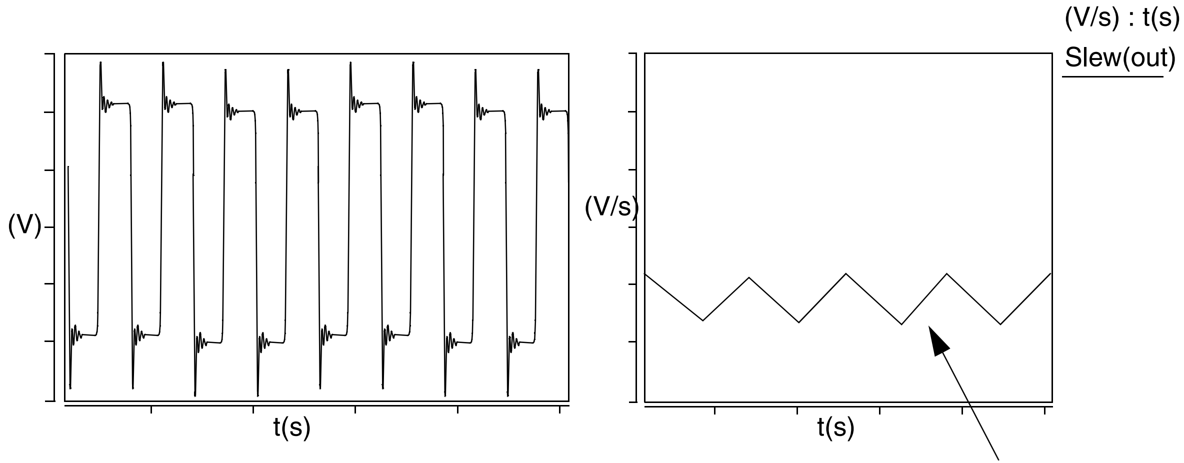
Measured waveform with single slew rate value.



□ You select the reference levels.

Measurement tool displays slew rate.

Example 2



Measured waveform with multiple slew rate values.

Measurement tool produces this waveform showing slew rate as a function of time.

Graphical Interface Description

Dialog Box Fields

Reference Levels — The following two fields set the topline and baseline levels for the measured signal and the corresponding reference signal. You can display any of these levels on the waveform by clicking on the corresponding Visibility Indicator to the right of each field.

Topline

Specify a topline value or use the default value.

Baseline

Specify a baseline value below the topline value or use the default value.

0-100%
10-90%
20-80% — Click on one of these buttons to set a range (in percent) relative to the topline/baseline levels.

Trigger

_|‾/‾|_ Specifies that the slew rate is calculated for rising or falling edges.

_|‾ Specifies that the slew rate is only calculated for rising edges.

‾|_ Specifies that the slew rate is only calculated for falling edges.

Create New Waveform on Active Graph or New Graph — Slew Rate vs. t

A new waveform is computed with slew rate (y-axis) versus time (x-axis). See Example 2.

Slope

Description

Displays the slope (optionally as a per-octave or per-decade value) of a waveform.

Type of Measured Waveform

- Analog, event-driven analog

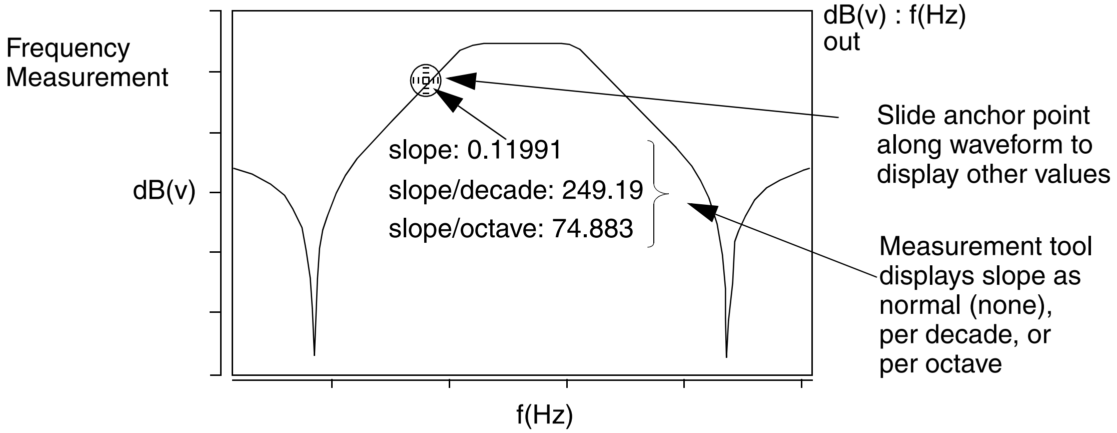
Command Group

Frequency Domain, General

Syntax

None

Example



Slide anchor point along waveform to display other values

Measurement tool displays slope as normal (none), per decade, or per octave

Graphical Interface Description

Dialog Box Fields

X Value

Optional. You can provide an x-value and the tool will provide the slope at that coordinate. If you do not specify the x-value, a default is used.

Option

(This field is visible in the Frequency Domain Category)

None
Displays the slope normally. See Example.

Per Decade
Displays the slope per decade. See Example.

Per Octave
Displays the slope per octave. See Example.

Standard Deviation

Description

Displays the standard deviation of a waveform. This measurement is intended for statistical (discrete) data such as histograms.

Type of Measured Waveform

- Scatter plot, histogram, event-driven analog

Standard Deviation Calculation

In this calculation, N is the number of points, Wj are the individual points of the waveform, and W is the Mean value.

$$\left[\frac{1}{N - 1}\sum_{j = 1}^{N}(W_j - \overline{w})^2\right]^{1/2}$$

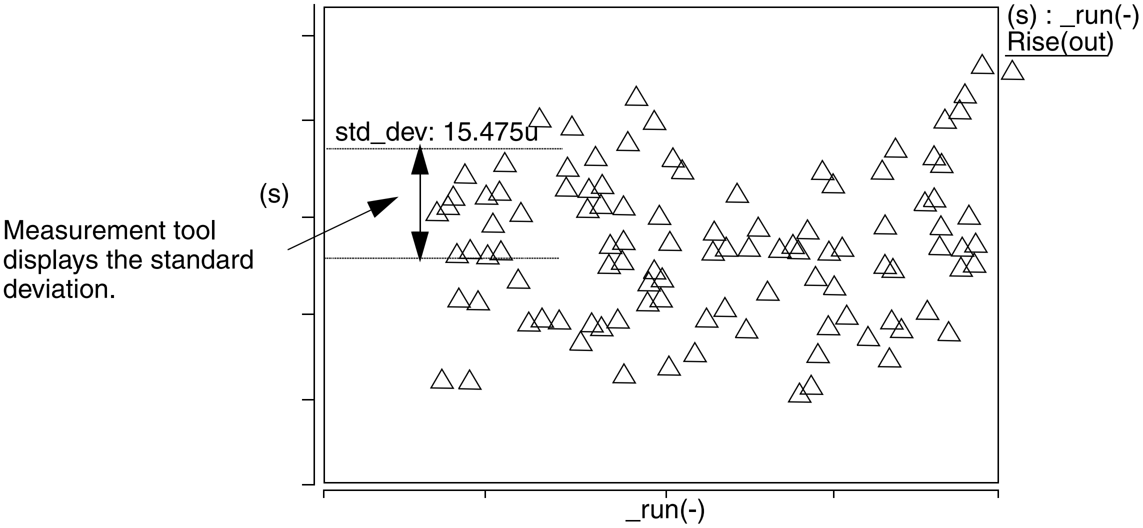
Command Group

Statistics

Syntax

None

Example



Graphical Interface Description

Dialog Box Fields

Category List	All Statistic category items appear below the Signal field. Select the Standard Deviation item and any other items you want to measure.

Stopband

Description

Displays the stopband, the low, high, or center frequency, or the level at which the measurement is made for a stopband-shaped waveform. The measurement is made relative to a default or specified topline level and a specified offset.

Type of Measured Waveform

- Analog

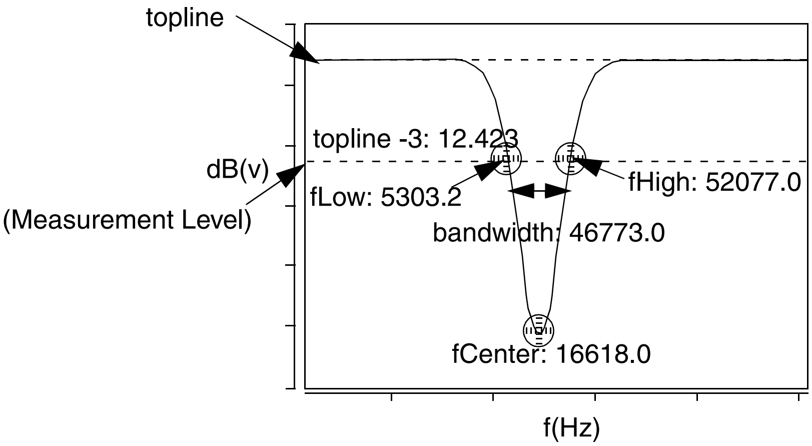
Command Group

Frequency Domain

Syntax

None

Example



Graphical Interface Description

Dialog Box Fields

Reference Levels — If you want to see the topline and/or offset level displayed on the waveform, click on the Visibility Indicator to the right of the Topline or Offset field.

Topline

You set this field to a default or a specified level.

Offset

You specify an offset value, to be applied relative to the Topline value. The default is 3. You must also choose which operator to use (–, +, *, or /) along with the specified level. The default is the minus (–) sign. This resulting level is also called the measurement level.

THD/SNR/SINAD

Description

Displays the Total Harmonic Distortion (THD), Signal to Noise Ratio (SNR), and Signal to Noise and Distortion Ratio (SINAD) on the resulting FFT waveform.

Type of Measured Waveform

- Analog

Graphical Interface Description

Dialog Box Fields

Max Harmonic — Specifies how many harmonics to use in the calculation of THD, SNR, and SINAD. Harmonics that are higher than the value specified for Max Harmonic are considered noise.

Bin Size — Specifies how many bins on each side of a harmonic frequency to include as the power of the harmonic. If you specify a windowing function other than rect, the Bin Size field is displayed. Windowing functions spread the original harmonics of the input signal spectrum into multiple bin components.

Fmin, Fmax — Specifies a range use to calculate the THD, SNR, and SINAD values. The default is the entire FFT waveform from start to end.

Threshold (at Y)

Description

Displays the x-axis values at a particular y-value on the waveform.

Type of Measured Waveform

- Analog, event-driven analog, scatter plot

Possible Errors

An error is reported if the waveform never crosses the measurement level.

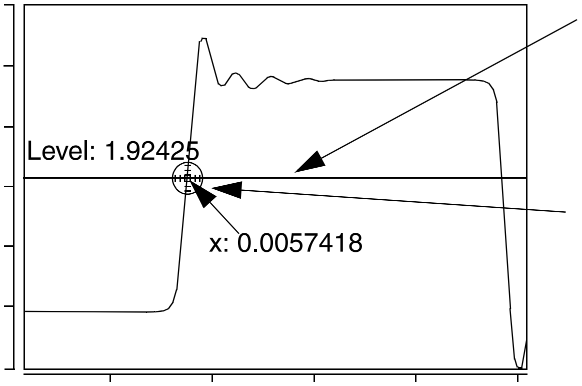
Command Group

General

Syntax

None

Example



Measurement tool displays a moveable horizontal line and one corresponding x-axis value at a waveform crossing.

Move anchor point along waveform to find other corresponding x values. Also use the Measure Results form to view all corresponding x values.

Graphical Interface Description

Dialog Box Fields

Y value	You can optionally supply a y-value, or a default will be calculated.	
Trigger	⌠/⌉	Specifies that the slope of the waveform at the y-value can be positive or negative.
	⌠	Specifies that the slope of the waveform at the y-value must be positive.

⌉⌊	Specifies that the slope of the waveform at the y-value must be negative.

Topline

Description

Displays the topline level of a waveform.

Type of Measured Waveform

- Analog, event-driven analog

Topline Calculation

If you do not specify the topline, a default value is calculated by using a method specified in the Default Topline/Baseline field in the Measurement Preference dialog box.

Command Group

Levels

Syntax

None

Graphical Interface description

Dialog Box Fields

Category List	All Levels category items appear below the Signal field. Select the Topline item and any other items you want to measure.

Undershoot

Description

Displays the undershoot of a waveform relative to a default or specified baseline level.

Type of Measured Waveform

- Analog, event-driven analog

Undershoot Calculation

The undershoot is calculated as the difference between the minimum point on the waveform and the specified (or calculated) Baseline value.

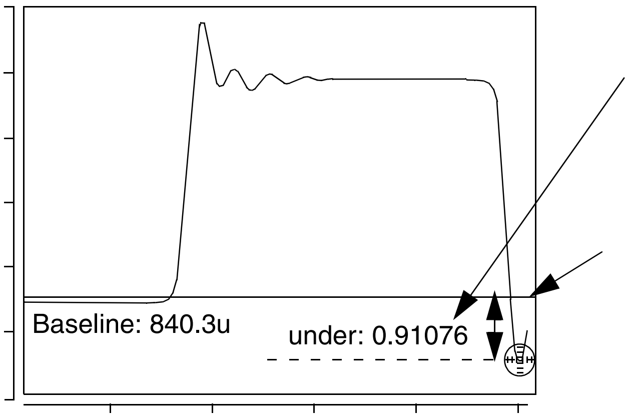
Command Group

Time Domain

Syntax

None

Example



Measurement tool displays undershoot result based on Baseline value.

Click on Visibility Indicator corresponding to the Baseline field to display this level.

Graphical Interface Description

Dialog Box Fields

Reference Levels

Topline

Specify a topline value within the upper and lower Y-axis values, or use the default value. You can display this reference level by clicking on the Visibility Indicator at the right of the Topline field.

Baseline

Specify a baseline value within the upper and lower y-axis values, or use the default value. You can display this reference level by clicking on the Visibility Indicator at the right of the Baseline field.

Measure Format

Absolute

The magnitude of the overshoot is calculated as the absolute value of an argument.

Percentage

The magnitude of the overshoot is calculated as the percentage of an argument.

Vertical Cursor

Description

Displays a vertical cursor that spans different graphs, for X-value, Y-value, and delta Y measurements.

Type of Measured Waveform

- Any

Vertical Cursor Measurement

To access the Vertical Cursor:

1. Select **Tools > Measurement** (or select the Measurement Tool button in the lower tool bar).
2. In the Measurement form, select **General > Vertical Cursor** and click the **Apply** button.

This measurement may be deleted in the Measure Results form or via a right mouse button form.

The Vertical Cursor measurement places vertical cursors in the regions of the selected signal and the reference signal, one marker in each region. You may move the marker if there are multiple Y values at that X value. You may also move the vertical cursor horizontally. The vertical cursors related to the same measurement in different regions move simultaneously.

Vertical Cursor measurement results are in two parts:

1. X-Y values of points indicated by the two markers.
2. Delta Y between the two markers.

Measurement results are displayed beside the markers and cursors. They can also be viewed in the Measure Results form.

Command Group

General

Syntax

None

Vertical Level

Description

Displays a moveable vertical line to identify x-axis levels.

Type of Measured Waveform

- Analog, event-driven analog, digital, scatter plot, histogram, spectral

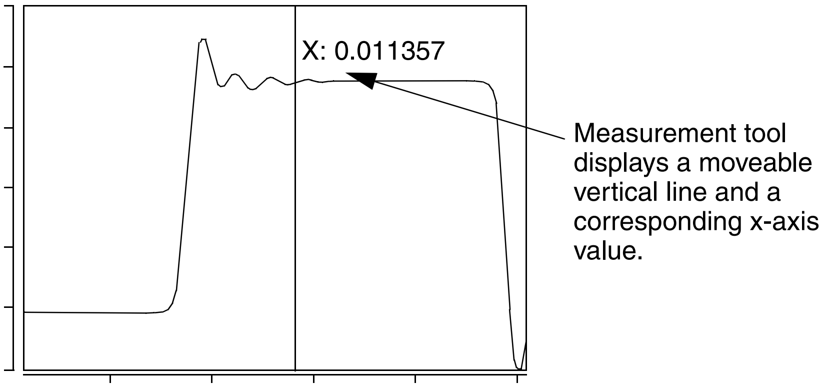
Command Group

General

Syntax

None

Example



Graphical Interface Description

Dialog Box Fields

Location (Optional) X Value	You can specify an x-value to place a vertical level on the waveform. If you do not specify one, a default value is computed.

Vertical Marker

Description

Displays the Y-axis value at a particular X-axis point on multiple waveforms. This measurement is only applied on the Graph Analog Region.

Type of Measured Waveform

- Analog, event-driven analog, digital, scatter plot, spectral, histogram, bus

Command Group

General

Syntax

None

Graphical Interface Description

Shortcut Button

A shortcut button is also available for the Vertical Marker Measurement:

When using this shortcut, select one or more x-axes and click the **Vertical Marker Measurement** button. The vertical marker measurements are applied to all signals with the specified x-axis or x-axes.

Dialog Box Fields

X Value — Optional. You can provide an X-value and the tool will provide the Y-value at that coordinate. If you do not specify the X-value, a default is used.

All signals in current graph — When you select this radio button, CosmosScope applies the measurement to all the signals with the specified x-axis in the current working graph.

All signals in selected region — When selected, this radio button opens the Region text field. Select the desired graph region from the pull-down menu, which contains a list of regions that contain the selected x-axis. The measurement is applied to all signals with the specified x-axis in the specified region.

Selected signals — When selected, this radio button opens the List dialog box, which contains all of the available signals that have selected x-axis. The measurement is applied to the signals you select.

X-Axis — Required. Select an x-axis from the pull-down menu, which contains a list of x-axes in the active graph. The measurement is applied to signals that have the x-axis you specify.

X at Maximum

Description

Displays the x-value corresponding to the maximum value of a waveform.

Type of Measured Waveform

- Analog, event-driven analog, scatter plot, histogram, spectral

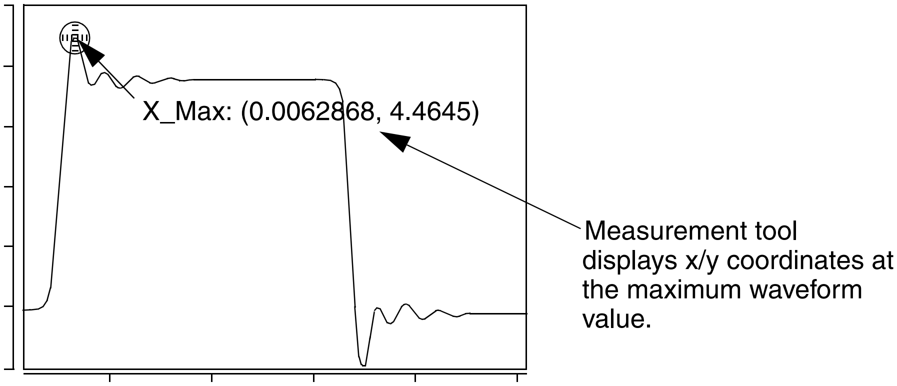
Command Group

Levels

Syntax

None

Example



Graphical Interface Description

Dialog Box Fields

Category List	All Levels category items appear below the Signal field. Select the X at Maximum item and any other items you want to measure.

X at Minimum

Description

Displays the x-value at the minimum value of a waveform.

Type of Measured Waveform

- Analog, event-driven analog, scatter plot, histogram, spectral

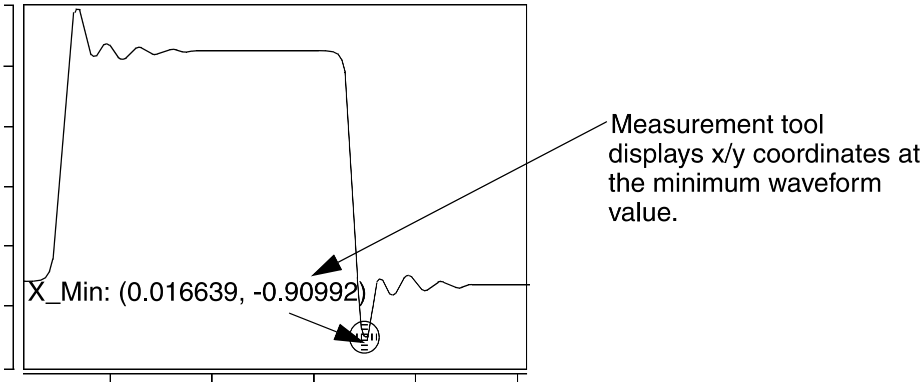
Command Group

Levels

Syntax

None

Example



Measurement tool displays x/y coordinates at the minimum waveform value.

Graphical Interface Description

Dialog Box Fields

Category List	All Levels category items appear below the Signal field. Select the X at Minimum item and any other items you want to measure.

Yield

Description

Displays the ratio of data points that fall between the specified upper and lower y-axis values of a waveform.

Type of Measured Waveform

- Scatter plot, histogram, analog, event-driven analog

Yield Calculation

The yield is calculated as the ratio of the number of data points between the y-axis levels Upper and Lower relative to the total number of data points.

The yield results are typically only meaningful if the input waveform is uniformly spaced along the x-axis (for example, the contents of a plot file generated by a measurement operation on the results of a Monte Carlo analysis).

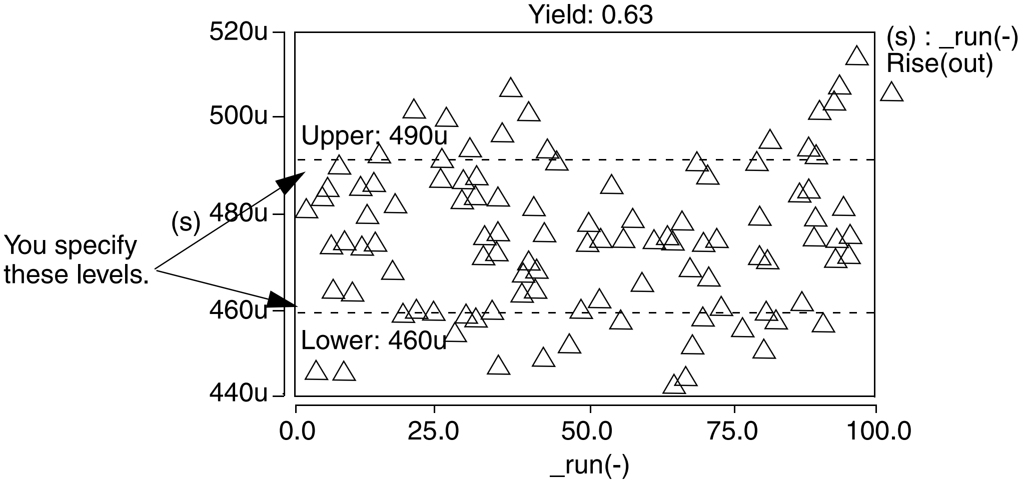
Command Group

Statistics

Syntax

None

Example



Graphical Interface Description

Dialog Box Fields

Category List	All Statistic category items appear below the Signal field. Select the Yield item and any other items you want to measure.

Specifications Limits Required values you supply.

Upper

Specifies the upper specification limit.

Lower

Specifies the lower specification limit.

8

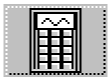
Using the Waveform Calculator

This chapter explains how to use the Waveform Calculator.

The Waveform Calculator operates as an RPN (Reverse Polish Notation) calculator or as an algebraic calculator. The calculator allows you to perform operations on AIM language expressions and waveforms as well as on numbers. The calculator can be programmed to perform custom operations or a series of operations.

Opening and Closing the Calculator

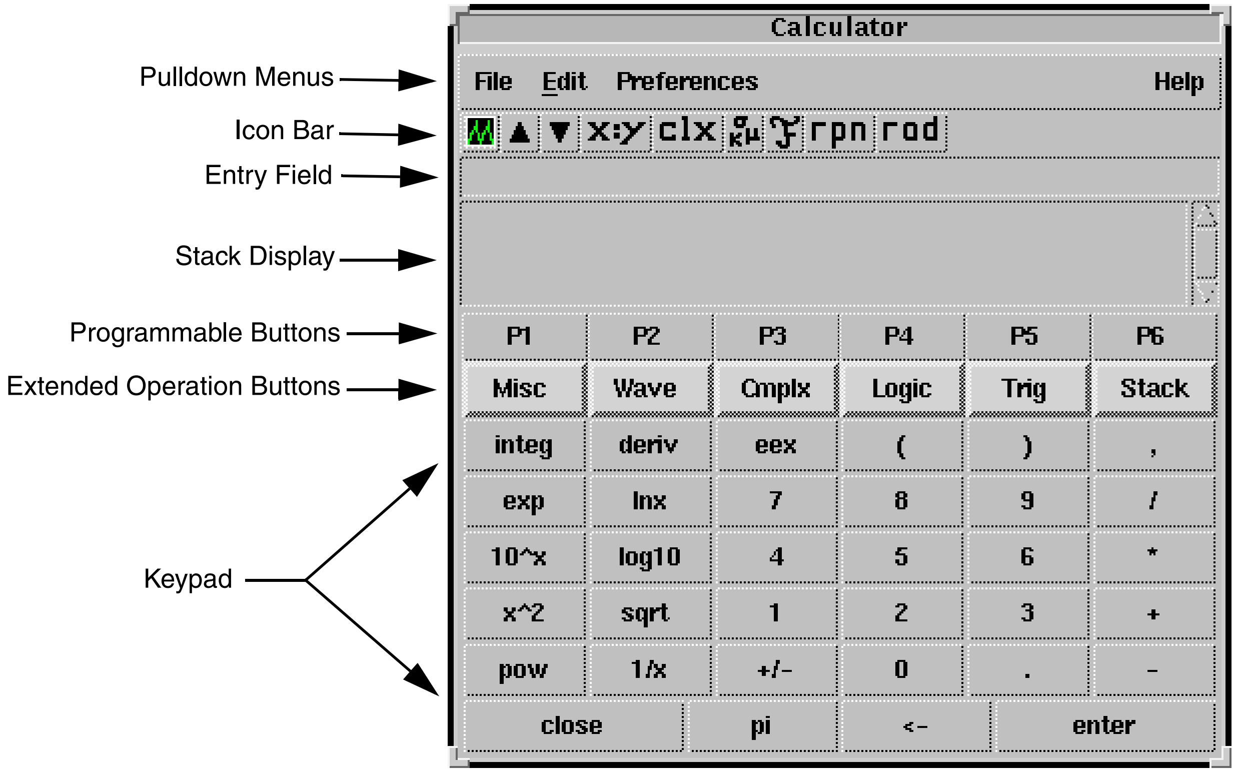
The Calculator icon shown below is located in the Tool bar at the bottom of the work surface.



To open or close the calculator, click the Calculator icon.

General Calculator Operation

The Programmable Waveform Calculator shown below is a resizable window consisting of a Pulldown menu bar, an Icon bar, an Entry field, a Stack Register display, Extended Operation buttons, and a basic calculator keypad.



Entering Operands

The calculator keypad, your computer keyboard, and your computer numeric keypad can all be used to input operands.

The Entry field is where numbers or waveforms appear and are evaluated before being pushed onto the Stack Display. The contents of the Entry field are always the same as that of the X-register.

The Stack Display is a scrollable list box that displays the stack registers. There is no limit to the number of stack registers.

To copy any register onto the X-register, click the register with the left mouse button, and then single click the middle mouse button.

Basic RPN Operation

The calculator is in rpn mode when the Input Mode icon displays rpn and the **Enter** button is visible in the lower right hand corner. Calculations are executed after the operands have been entered onto the stack.

The X-register and the Y-register

In RPN mode, the contents of the X-register and the Y-register are not always displayed in the same locations.

RPN Mode Example

Two Operand Operation

Two operand operations are performed on the contents of the X-register and Y-register.

Two Operand Example

One Operand Operation

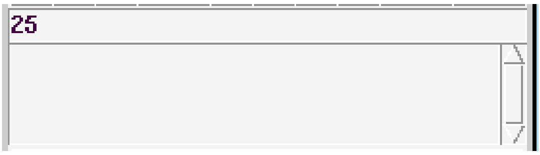
One operand operations are performed on the contents of the X-register only.

One Operand Example

RPN Mode Example

For example, the X-register will be the Entry field if only one operand has been input and the **Enter** button has not been pressed.

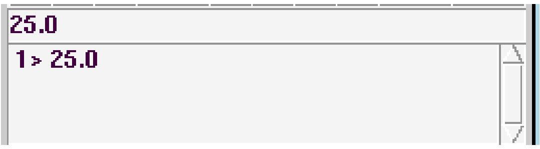
Input 25
(The Entry field is the X-register)



25

The X-register will be the Entry field and register one in the Stack Display if only one operand has been input and the **Enter** button has been pressed.

Press enter
(The Entry field and register one are the X-register)

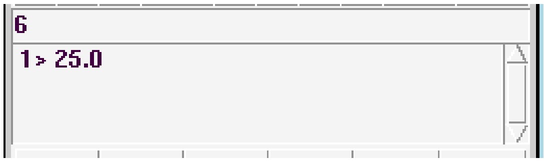


25.0

1> 25.0

The X-register will be the Entry field, and the Y-register will be register one, if a second operand is input and the **Enter** button has not been pressed.

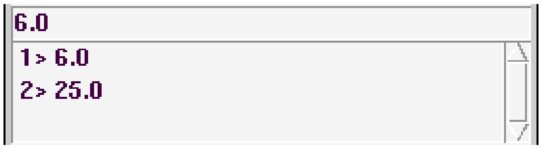
input 6
(6 is the X-register, 25 is the Y-register.)



```
6
1> 25.0
```

The X-register will be the Entry field and register one in the Stack Display, and the Y-register will be register two, if a second operand is input and the **Enter** button has been pressed.

Press enter
(The Entry field and register one are the X-register, register two is the Y-register)

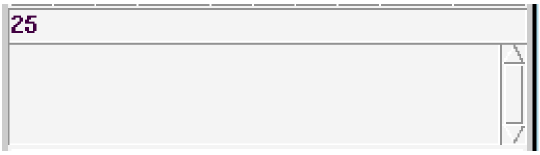


```
6.0
1> 6.0
2> 25.0
```

Two Operand Example

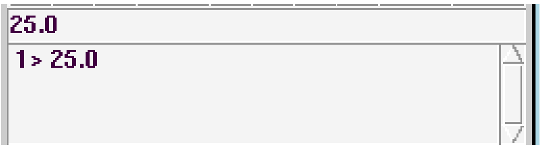
For example, to calculate 25 - 6, execute (y - x) by following these steps:

Input 25
(25 is displayed in the Entry field)



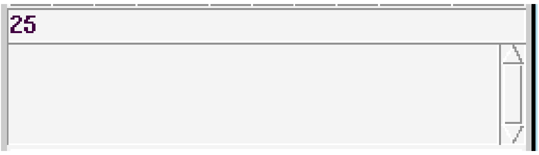
```
25
```

Press enter
(25 is pushed onto register one)



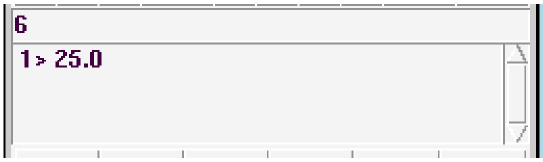
```
25.0
1> 25.0
```

Input 25
(25 is displayed in the Entry field)



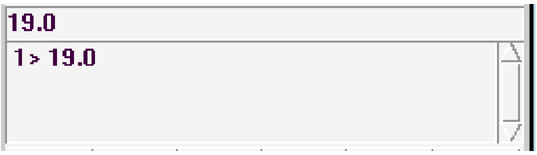
```
25
```

input 6
(6 is displayed in the Entry field)



```
6
1> 25.0
```

Press -
(The subtraction operation is performed on the two operands and the result is pushed onto register one)

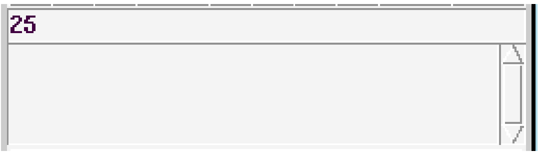


```
19.0
1> 19.0
```

One Operand Example

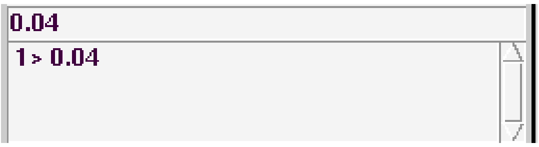
For example, to calculate the inverse of 25 (=1/x), follow these steps:

Input 25
(25 is displayed in the Entry field)



```
25
```

Press 1/x
(The inversion operation is performed on the operand, and the result is pushed onto the X-register)



```
0.04
1> 0.04
```

Basic Algebraic Operation

The calculator is in algebraic mode when the Input Mode icon displays alg, and the **=** button is visible in the lower right hand corner of the keypad.

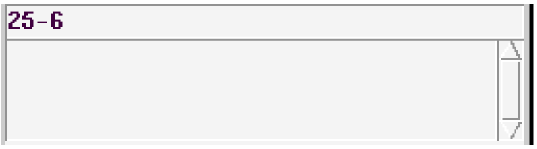
Calculations are executed after the **=** button is pressed.

Two Operand Operation	Two operand operations are performed in the Entry field. Two Operand Algebraic Example
One Operand Operation	One operand operations are performed in the Entry field. One Operand Algebraic Example

Two Operand Algebraic Example

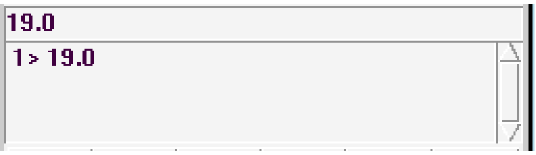
For example, to calculate 25 - 6, follow these steps:

Input 25-6
(25-6 is displayed in the Entry field)



```
25-6
```

Press =
(The subtraction operation is performed in the Entry field, and the result is pushed onto the X-register)

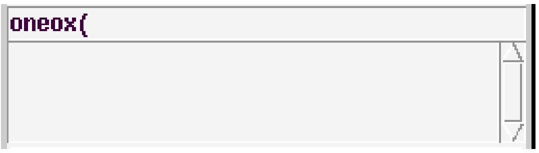


```
19.0
1> 19.0
```

One Operand Algebraic Example

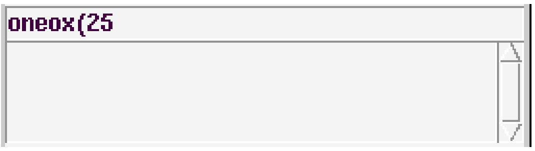
For example, to calculate the inverse of 25, follow these steps:

Press 1/x
(The operation with an open bracket is displayed in the Entry field)



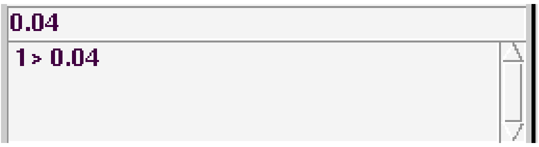
```
oneox(
```

Input 25
(25 is placed in the operation bracket)



```
oneox(25
```

Press =
(The brackets are closed, the inversion operation is performed on the Entry field, and the result is pushed onto the X-register)



```
0.04
1> 0.04
```

Performing Waveform Calculations

Calculations can be performed on waveforms as well as on numbers. Multiple waveforms can be pasted into the calculator.

In rpn input mode, multiple waveforms are pushed onto the stack, one waveform per register. In alg mode, the waveforms are pasted sequentially in the Entry field.

The results of these calculations can be plotted in an open graph window by single clicking on the Graph icon.

Note:

> When any previously graphed waveform is updated due to an automatic plot action (such as Append) specified in an analysis, all waveforms created by the calculator that depend on that waveform are updated at the same time.

To paste a waveform onto the Entry field:

1. Select the waveform in the Signal Manager or from a graph region.
2. Place the mouse cursor in the Entry field, and single click the middle mouse button.

To change the color of the waveform display:

1. Choose **Edit > Preferences...** from the main CosmosScope menu bar.
2. Click the Signal tab.
3. Choose the desired colors for both analog and digital signals.

Wave Extended Operation Button

The Wave Extended Operation button contains menu items that only perform operations on waveforms.

The FFT, IFFT, Limit X Range, Limit X and Y Range, Change X and Y View, Sample X Axis, Histogram, Extract Member, and Swap Parameters menu items open dialog boxes. Detailed instructions for filling out these dialog boxes are available by pressing the **Help** button in each respective dialog box.

Invoking the following functions will cause the corresponding abbreviations to appear in brackets [ ] in the Entry field:

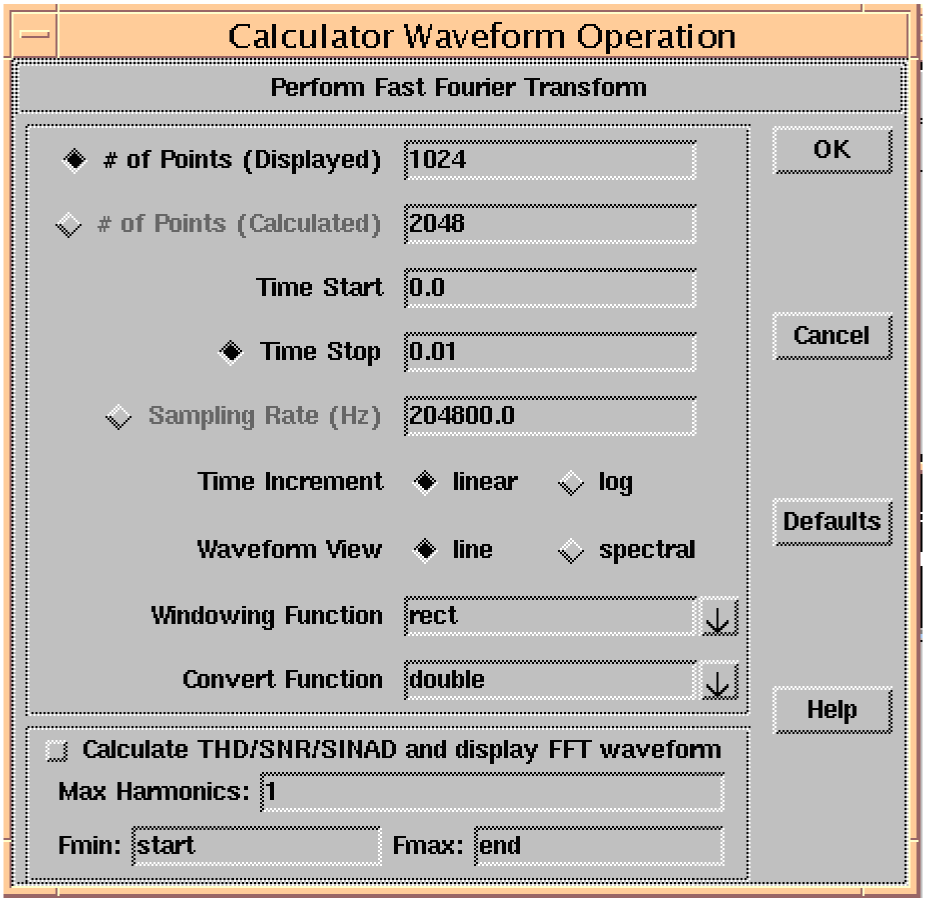
Limit to finite values	[finitelimit]
f(x)	[yvsx]
Swap X and Y Axes	[xyswap]
Reduce dimension	[flatten]

The name of the file to be acted upon by the function must be placed within the brackets, followed by a space, immediately after the function name. For example, if the waveplot file is named filt_out, then in order to apply f(x) to filt_out, first choose f(x) on the Wave menu, and then place filt_out in the brackets, as follows:

```
[yvsx filt_out]
```

FFT Calculation

Click the **Wave** button and choose **FFT** from the menu appears to open the following FFT Calculator Waveform Operation dialog box:



Calculator Waveform Operation

Perform Fast Fourier Transform

◆ # of Points (Displayed) 1024

◇ # of Points (Calculated) 2048

Time Start 0.0

◆ Time Stop 0.01

◇ Sampling Rate (Hz) 204800.0

Time Increment ◆ linear ◇ log

Waveform View ◆ line ◇ spectral

Windowing Function rect

Convert Function double

□ Calculate THD/SNR/SINAD and display FFT waveform

Max Harmonics: 1

Fmin: start Fmax: end

The following options are available:

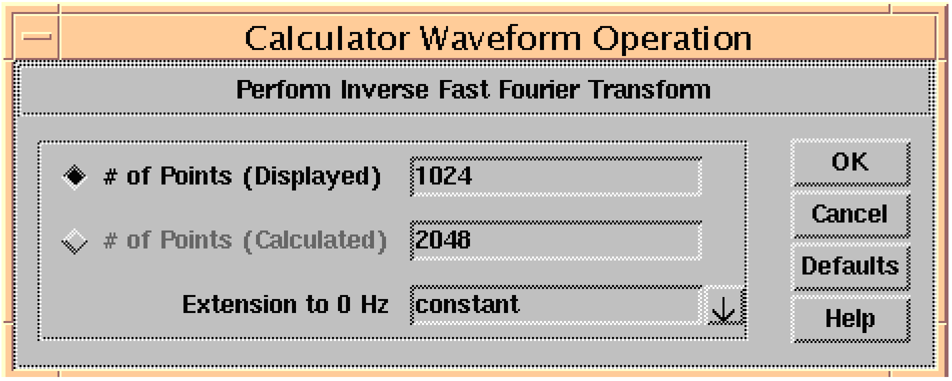
# of Points (Displayed)	Specifies the number of points to be displayed. If the **# of Points (Displayed)** radio button is not selected, click the **# of Points (Calculated)** radio button to toggle to this function.
# of Points (Calculated)	Specifies the number of points to be passed to the FFT algorithm. If the **# of Points (Calculated)** radio button is not selected, click the **# of Points (Displayed)** radio button to toggle to this function.

Time Start	The time start value.
Time Stop	The time stop value. If the **Time Stop** radio button and value are not selected, click the **Sampling Rate (Hz)** radio button to toggle to this function.
Sampling Rate (Hz)	Specifies the frequency of the sampling signal for the FFT operation. If the **Sampling Rate (Hz)** radio button and value are not selected, click the **Time Stop** radio button to toggle to this function.
Time Increment	Specifies either the linear or log time increment value.
Waveform View	Specifies either the line or spectral waveform view style.
Windowing Function	Specifies the window function. You can choose from the following functions: ▪ rect ▪ bartlett ▪ hann ▪ hamming ▪ blackman ▪ flattop ▪ user
Convert Function	Specifies the convert function. You can choose from double, one's complement, or two's complement.

Calculate THD/ SNR/SINAD and display FFT waveform	Calculates the Total Harmonic Distortion (THD), Signal to Noise Ratio (SNR), and Signal to Noise and Distortion Ratio (SINAD) on the resulting FFT waveform. The Max Harmonic value specifies how many harmonics to use in the calculation of THD, SNR, and SINAD. Harmonics that are higher than the value specified for Max Harmonic are considered noise. If you specify a windowing function other than rect, the Bin Size field is displayed. Windowing functions spread the original harmonics of the input signal spectrum into multiple bin components. The Bin Size value specifies how many bins on each side of a harmonic frequency to include as the power of the harmonic. Fmin and Fmax specifies a range use to calculate the THD, SNR, and SINAD values. The default is the entire FFT waveform from start to end.

IFFT Calculation

Click the **Wave** button and choose **IFFT** from the menu appears to open the following IFFT Calculator Waveform Operation dialog box:



Calculator Waveform Operation

Perform Inverse Fast Fourier Transform

◆ # of Points (Displayed) 1024

◇ # of Points (Calculated) 2048

Extension to 0 Hz constant

The following options are available:

# of Points (Displayed)	Specifies the number of points to be displayed. If the **# of Points (Displayed)** radio button is not selected, click the **# of Points (Calculated)** radio button to toggle to this function.

# of Points (Calculated)	Specifies the number of points to be passed to the FFT algorithm. If the **# of Points (Calculated)** radio button is not selected, click the **# of Points (Displayed)** radio button to toggle to this function.
Extension to 0 Hz	Specifies the extension type to 0 Hz. You can choose from constant, zero, or linear types.

Entering Complex Numbers

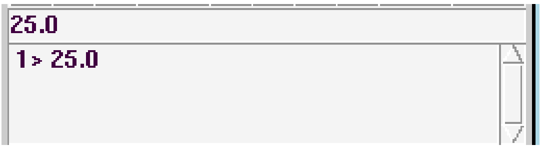
Complex numbers are input in a different manner from natural numbers and waveforms.

RPN Mode	In rpn mode, complex numbers can be entered into the calculator by using the letter j after the imaginary portion, or by using the j or complex items from the Cmplx extended operation button. Complex Number - RPM Mode - Example
Algebraic Mode	In algebraic mode, complex numbers can be entered as a numeric string. Complex Number - Algebraic Mode - Example

Complex Number - RPM Mode - Example

For example, to enter the complex number (25 + 2j), follow these steps:

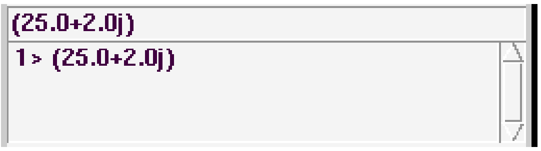
Input 25, and Enter
(The real number 25 is pushed onto the X-register)



```
25.0
1> 25.0
```

Input 2j+
or

Input 2, and then choose complex from the Cmplx menu
(The complex number 25 + 2j is pushed onto the X-register)



```
(25.0+2.0j)
1> (25.0+2.0j)
```

Complex Number - Algebraic Mode - Example

For example, to enter the complex number (25 + 2j), enter the numbers and text followed by entering the equals sign (=). The number will be pushed onto the X-register.

Entering Vectors, Matrices, and Arrays

Vectors, matrices, and arrays (vmas) are input in a different manner from that of natural numbers and waveforms.

A vma can be entered into the calculator by using the AIM vma command. (For details on this command refer to the AIM Command Reference Manual.) Generally, the command is used in the following manner:

```
[vma vma_type list {vma values}]
```

vma type can be vector, matrix, diag, array3, array 4, -datatype complex, -datatype real, or waveform.

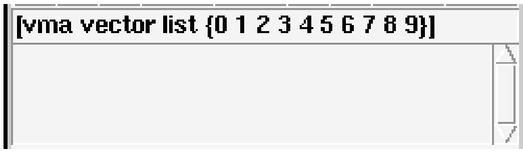
vma values are the list of numbers that make up the vma. Each number must be separated by a space.

The vma calculator operations are available under the **Misc** button with the vma menu item. For details on these vma operations, refer to the AIM Command Reference Manual.

VMA Example

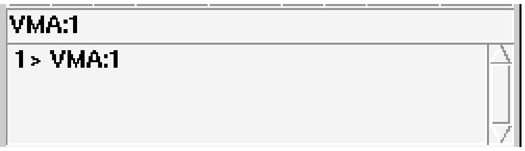
For example, to calculate the average of 0, 1, 2, 3, 4, 5, 6, 7, 8,and 9 follow these steps.

Input the vma string exactly as shown on the right.



```
[vma vector list {0 1 2 3 4 5 6 7 8 9}]
```

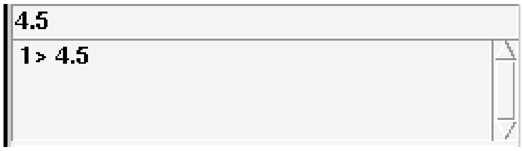
Press enter
(The vma string is converted to an AIM representation and pushed onto the X-register).



```
VMA:1
1> VMA:1
```

Choose Misc >

vma > mean
(The average of the vector is pushed onto the X-register).



```
4.5
1> 4.5
```

Using Constants

The calculator provides a Constants dialog box containing a list of mathematical and physical constants. You can also create your own set of constants. The icon for constants is σ κμ .

To open the Calculator Constants dialog box, click the Constants icon. Mathematical constants are listed under the math tab, physical constants are listed under the physical tab, and any user-created constants are listed under the user tab.

For additional information about any constant, select the constant with a single click and press the calculator **Help** button.

To input a constant into the calculator, double-click the constant or select the constant with a single click, and press the **OK** button.

To create a User Constant:

1. Click the User tab to display the User tab form.
2. Click the **Add** button. The Add User Constant dialog box opens with instructions for creating a user constant with a custom Help message.

   The format for adding a constant is the constant name, constant value, and a brief description of the constant (or whatever you choose to put in the help text string).

   ```
   (<name>=<value>;<help_text>).
   ```

To save user-defined constants between sessions, click the **Cancel** button to save your constant and close the Constants dialog box. User-defined constants are treated as preferences. You must choose **File > Preferences > Save** to save the user-defined constants.

To delete a user-defined constant, select the constant and click the **Remove** button.

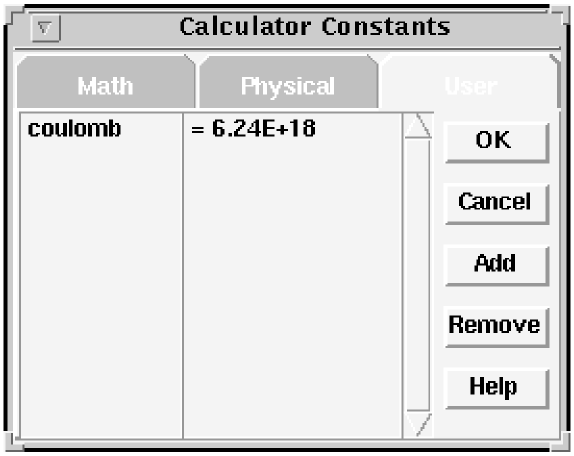
Constants Example

To create a constant for the number of electrons in a coulomb (6.24 x 10^{18})

1. Click the **Add** button.
2. In the dialog box type the following:

coulomb=6.24E18;number of electrons
in a coulomb

3. Single click the **OK** button to add the constant to the User list.



coulomb = 6.24E+18

4. Selecting the constant and clicking on the calculator **Help** button displays the following:

coulomb: 6.24E+18, number of electrons

in a coulomb

Programming the Calculator

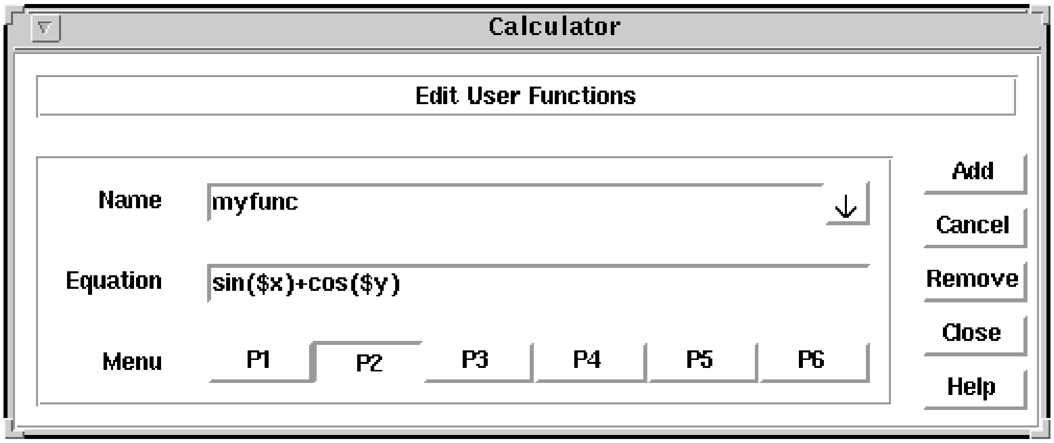
Buttons **P1** through **P6** can be programmed with user-defined menus containing user-defined operations. These programmable buttons function like the Extended Operation buttons. Each programmable button can contain as many operations as you desire.

You can program or edit menus in the programmable buttons by using Edit User Function dialog box.

To access the Edit User Function dialog box, click the Function icon located on the icon bar, or choose **File > Add Function**.

To add a user function:

1. Type in a new function name in the Name field.



Name: myfunc

Equation: sin($x)+cos($y)

Menu: P2

2. Input the operation you want the menu item to perform in the Equation field. Operations must be written in the AIM language.

   Valid equation examples are given below:

sin($x)

sin($x)+cos($x)

sin($x)+cos(exp($y))

deriv(log($x-$z-$signal))

$var1 + $var2

[Measure:Frequency $wf]+20

13/27

   Your equations should adhere to the format of these examples.

   The dollar sign indicates an input to the function. Any number of inputs may be specified. Any combination of letters and numbers may be used for input names.

   You may use any available AIM math function. AIM procedures may also be used, as the [Measure:Frequency $wf]+20 example shows, providing the procedure returns a number or a waveform.

3. Choose the location of your new menu item in the Menu field. You can place a new menu item under any button from **P1** through **P6**. You can also place multiple menus under a single button.
4. Press the **Add** button to add the item to the calculator.

To save user-defined functions between sessions, click the **Close** button to save your information and exit the Edit User Functions dialog box. User-defined functions are treated as preferences. You must choose **File > Preferences > Save** to save any user-defined functions.

To close the dialog box without saving your program, click the **Cancel** button.

To delete a function, click the **Remove** button.

To edit an existing function:

1. Click the down arrow at the right of the Name field in the Edit User Functions dialog box. A list of functions are displayed.
2. Select the function you would like to edit.
3. Make changes in the Name and Equation fields or **P1** - **P6** buttons as necessary
4. Click the **Add** and **Close** buttons to save changes.

Calculator Menus

File Menu	Allows you to manage preferences, add functions, save waveforms to a file, and close the calculator.
Edit Menu	Cuts, copies and pastes items between the graph window and the Entry field.
Preferences Menu	Allows you to select rpn or alg Input Mode, type of display, degree of precision, operand numeric base, and waveform color.

File Menu

Menu Item	Description
Preferences > Save	Saves precision, mode, constants and user functions.
Preferences > Restore	Restores the previously saved set of preferences.

Preferences > Default	Sets preferences to default values:
	precision = 6
	mode = rpn,
	notation = engineering,
	base = decimal,
	waveform color = blue.
	User defined constants and functions are retained.

Add Function	Opens the Add Function dialog box.
Save X	Saves the waveform in the X-register to a file.
Save Selected	Saves the selected waveforms in the stack display to a file.
Save All	Saves all waveforms in the stack display to a file.
Close Window	Closes the calculator.

Edit Menu

Menu Item	Description
Cut	Removes a selected object and moves it into the clipboard.
Copy	Copies a selected object into the clipboard.
Paste	Pastes whatever is in the clipboard into the Entry field.

Preferences Menu

Menu Item	Description
rpn	Converts to Reverse Polish Notation calculator operation.
algebra	Converts to Algebraic calculator operation.
degrees	Performs trigonometric calculations in degrees.
radians	Performs trigonometric calculations in radians.
grads	Performs trigonometric calculations in grads.
engineering	Displays operands in engineering notation.
scientific	Displays operands in scientific notation.
fixed point	Displays operands in fixed decimal point notation.
Precision >	Sets the number of digits displayed for floating point numbers.
decimal	Displays operands as base 10.
octal	Displays operands as base 8.
hexadecimal	Displays operands as base 16.

Menu Item	Description
Waveform Color...	Opens the Draw Color Editor dialog box to allow the customizing of waveform color.

Help Menu

Menu Item	Description
Help on Calculator	Opens online documentation.
About AimCalc	Displays the AimCalc version and copyright information.

Calculator Icons

Icon	Description
Graph	Pastes the contents of the X-register into the active graph window.
Up ▲	Rolls the stack up by one register. The X-register moves to the bottom of the stack, and the Y-register moves to the top of the stack.
Down ▼	Rolls the stack down by one register. The bottom register moves to the top of the stack, and the X-register contents are pushed onto the Y-register.
X:Y x:y	Swaps the X and Y registers.
Clear X clx	Clears the X-register. The Y-register contents are pushed onto the X-register.
Constants σ κμ	Displays the Constants dialog box.

Icon	Description
Add Function ℱ	Opens the Add User Function dialog box.
Input Mode rpn	Toggles between Reverse Polish Notation calculator operation and Algebraic calculator operation.
Trigonometric Mode rad	Toggles between degrees, radians, and grads for trigonometric operations (RPN mode only).

Calculator Extended Operation Buttons

Misc Button	Performs various mathematical operations as well as vector and matrix array operations.
vma Menu	Performs vector, matrix, and array operations. This menu is selected from **Misc** button options.
Wave Button	Performs waveform operations.
Cmplx Button	Performs complex number operations.
Logic Button	Performs logical operations.
Trig Button	Performs trigonometric operations.
Stack Button	Manages the stack registers.

Misc Button

Operation	Description
abs	The absolute value of the X-register.
mod	x modulo y.
fmod	Floating point x modulo y.
ceil	Rounds off floating point values to the next highest integer.
floor	Rounds off floating point values to the next lowest integer.
round	Rounds off floating point values.
hypot	The square root of the sum of the square of the X-register and the square of the Y-register.
vma	Opens Vector/Matrix/Array menu.

VMA Menu

Operation	Description
vmax	Returns the largest value of all of the elements in a vma argument.
vmaxi	Returns the index for the largest value of all of the elements in a vma argument.
vmin	Returns the smallest value of all of the elements in a vma argument.
vmini	Returns the index for the smallest value of all of the elements in a vma argument.
mean	Returns the mean value of all of the elements in a vma argument.
sum	Returns the sum of all of the elements in a vma argument.
var	Returns the computed sample variance of all the elements in a vma argument.
variance	Returns the biased variance of all the elements in the vma argument.
specvariance	Returns the spectral variance of all the elements in a vector vma argument. The argument is assumed to be the result of a Fourier transform.

Wave Button

Operation	Description
FFT	Opens the Fast Fourier Transform dialog box.
IFFT	Opens the Inverse Fast Fourier Transform dialog box.
Swap X and Y Axes	Swaps the x and y axes of the waveform in the X-register.

Operation	Description
Limit X Range	Limits the range of the x axis of the waveform in the X-register.
Limit X and Y Range	Limits the range of the x and/or y axes of the waveform in the X-register.
Limit to Finite Values	Limits the range of the x and/or y axes of the waveform to finite values.
Change X and Y View	Applies a view transform to the x and/or y axes of the waveform in the X-register
Sample X Axis	Applies X-sampling to the waveform in the X-register.
f(x)	Creates a parametric function of y1 vs y2, with X as the parameter for waveforms in the X-register and Y-register.
Histogram	Converts the waveform in the X-register to a histogram.
Extract Member	Extracts a single member from a multi-member waveform in the X-register.
Swap Parameters	Swaps the waveform parameter order of the waveform in the X-register.
Reduce Dimension	Reduces the dimension of the waveform in the X-register by 1.
Group Delay	Creates a new waveform displaying Group Delay time.
Phase Delay	Creates a new waveform displaying Phase Delay time.
Concat	Takes the waveform in the X register and concatenates it to the waveform in the Y register. If the waveforms have the same name, the resulting waveform also has that name. If the waveforms have different names, the resulting waveform is named concat.

Cmplx Button

Operation	Description
J	Places a “j” onto the X-register to indicate a complex number.
complex	Creates a complex number with the X-register as the real part, and the y register as the imaginary part.
real	Places the real part of a number or waveform onto the X-register.
imag	Places the imaginary part of a number or waveform onto the X-register.
mag	Absolute magnitude.
db	Decibels.
phase	Phase of the X-register, always in radians, limited to values between 0 and 2p.
cphase	Phase of the X-register, always in radians, with unlimited bounds.
phasedeg	Phase of the X-register, always in degrees, limited to values between 0 and 360.
cphasedeg	Phase of the X-register, always in degrees, with unlimited bounds.
conjugate	Complex conjugate of the X-register.
polar	Converts a complex number to polar notation.

Logic Button

Operation	Description
y or x	OR the X and Y-registers (bitwise operation).

Operation	Description
y and x	AND the X and Y-registers (bitwise operation).
y xor x	Exclusive OR the X and Y-registers (bitwise operation).
not x	NOT the X-register (bitwise operation).
y<<x	Left shift the Y-register by the number of bits in the X-register.
y>>x	Right shift the Y-register by the number of bits in the X-register.
y<x	The X-register is set to 0 if the X-register is less than the Y-register. Otherwise, it is set to 1.
y<=x	The X-register is set to 0 if the X-register is less than or equal to the Y-register. Otherwise, it is set to 1.
y>x	The X-register is set to 0 if the X-register is greater than the Y-register. Otherwise, it is set to 1.
y>=x	The X-register is set to 0 if the X-register is greater than or equal to the Y-register. Otherwise, it is set to 1.

Trig

Operation	Description
sin	Sine of the X-register.
cos	Cosine of the X-register.
tan	Tangent of the X-register.
asin	Inverse sine of the X-register.
acos	Inverse cosine of the X-register.
atan	Inverse tangent of the X-register.
atan2	Inverse tangent of the X-register divided by the Y-register.
sinh	Hyperbolic sine of the X-register.
cosh	Hyperbolic cosine of the X-register.
tanh	Hyperbolic tangent of the X-register.

Stack

Operation	Description
Clear All	Clear all registers.
Clear Selected	Clear selected stack registers.
Clear X	Clear the first stack register.
Rename X	Rename the item in the X-register
Copy X	Copy waveform in the X-register. Specify new axis names and units of scale.
Swap X and Y	Swap the first two stack registers.
Roll Up	Roll the stack up one register.

Operation	Description
Roll Down	Roll the stack down one register.

Calculator Keypad

Button	Description
integ	Integrate the X-register.
deriv	Differentiate the X-register.
eex	Exponent 10x.
(	Left parentheses.
)	Right parentheses.
,	Comma, for multi-argument functions.
exp	Natural exponential.
lnx	The natural logarithm of the X-register.
/	The Y-register divided by the X-register.
10^x	The X-register raised to the power 10.
log10	Base 10 logarithm of the X-register.
*****	The X-register multiplied by the Y-register.
x^2	The X-register squared.
sqrt	The square root of the X-register.
+	The X-register plus the Y-register.
pow	The Y-register raised to the power of the X-register.
1/x	1 divided by X.
+/-	Change the sign of the X-register.
.	Decimal point.
-	The Y-register minus the X-register.
close	Close the calculator.

Button	Description
pi	Enter the value of p (3.14159266535898) onto the X-register.
<-	Backspace one character in the Entry field.
enter	In RPN mode only, evaluate the X-register and push the result onto the stack.
=	In algebraic mode only, evaluate the X-register and push the result onto the stack.

Calculator Computer Keyboard Operation

Many calculator operations can be performed from your computer work station keyboard.

Key	Description
Return **Enter**	Evaluates the X-register.
Numeric Keypad	Enters numerals and operands.
Up Arrow	Rolls the stack up.
Down Arrow	Rolls the stack down
Left Arrow	Moves the cursor to the left.
Right Arrow	Moves the cursor to the right.
Escape	Functions like the **Cancel** button in forms.
Help	Functions like the **Help** button in forms.
F1	Functions like the **Help** button in forms.
Tab	Moves to the next field in a form.
Delete	Clears selected register.
Back Space	Backspaces one character in the Entry field.

9

Using the Macro Recorder

This chapter explains how to use the Macro Recorder.

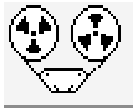
The Macro Recorder tool records a series of actions performed in Saber Guide, CosmosScope, Saber Sketch, and the Saber Simulator. You can edit these actions and play them back as a script.

For example, to perform a measurement on a signal, you must open a graph window, graph a specific signal, and perform the measurement. By recording these actions and editing the script, you can automate the measurement when you play back the recording.

Every significant operation is saved in a your_application.log file in the AIM scripting language. The Macro Recorder records the AIM language script in this file. You are not required to view the log to operate the Macro Recorder, but viewing operations as you record them can be helpful.

Accessing the Macro Recorder

The Macro Recorder icon is located in the Tool bar at the bottom of the work surface.



To open or close the Macro Recorder, click the icon.

Saber Log Files

The log is a transcript of the AIM language used to operate Saber applications. The AIM language displayed in the log can be used to create scripts with the Record/Playback tool.

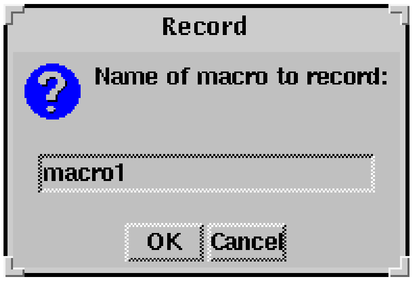
An ASCII text record of the log, named your_application.log, is automatically created in the directory where you started your session.

To view the logged commands, click the AIM Command Line tool in the tool bar, which opens the Command Line window. Next, choose **Edit > Display Logging**, which selects the radio button next to the **Display Logging** menu item. When operations are performed in the Command Line window, the log is displayed.

Recording Macros

To record a macro:

1. Click the **Record Macro** button ● in the Macro Recorder button bar. The Record dialog box appears.



Name of macro to record: macro1

2. Enter a name for the macro you are about to record.
3. Click the **OK** button to begin the recording process. From this point forward, every action you perform is recorded.

To temporarily pause a recording, click the **Pause** button ⏸.

To stop recording, click the **End** button ■.

Playing Macros

To play a macro, the Browser must be open. Open the browser by clicking the **Browse** button ▤ on the Macro Recorder Button bar, then select the macro you want to play from the list.

To play a macro, click the **Play** button > .

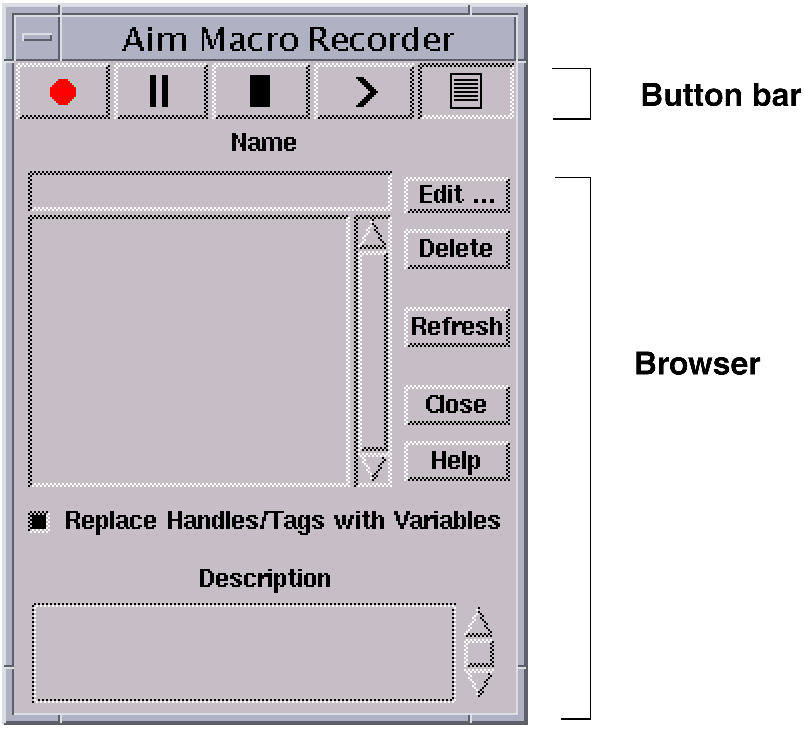
Macro Recorder Controls

The Button bar at the top of the Macro Recorder window controls recording and playback of macros.

The Browser section below the Button bar allows you to manage and edit macro files.

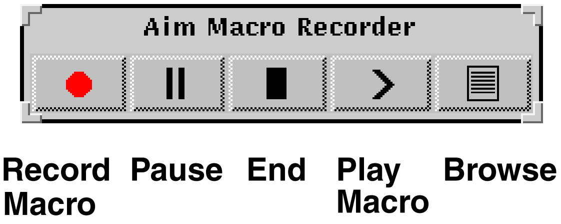
Note:

> The Replace Handles/Tags with Variables option is only available when CosmosScope is invoked. This option does not appear in the Macro Recorder window if CosmosScope is not invoked.



■ Replace Handles/Tags with Variables

You control the recorder using the five Macro Recorder control buttons.



Record Macro button — Begins recording every operation you perform. Clicking this button closes the browser and leaves the button bar displayed.

Pause button — Halts the recording process. The recorder does not collect any more actions for this macro file until you click this button again. The file remains open.

End button — Stops the recording process. Your actions are collected in the macro file you named, and the file is closed.

Play Macro button — Immediately executes all the recorded instructions. The active script can be viewed in the Command Line window if logging is displayed.

Browse button — Displays the Macro Recorder Browser.

Macro Recorder Browser Controls

You manage and edit macro files in the Browser section of the AIM Macro Recorder window.

Name Field — Contains the name of the macro file that is highlighted in the list of macros.

Scrollable List — Displays previously-recorded macros.

Description Field — Displays your description of the highlighted macro.

Edit Button — Allows you to edit a selected macro.

Delete Button — Removes the selected macro from the list of macros.

Refresh Button — Restores the list of macros to the previously-recorded macros available at the start of your session.

Close Button — Closes the Macro Recorder tool.

Editing Macro Files

To display the Edit window, click the **Edit** button. The name of the file you are editing is displayed in the title bar at the top of the window.

The AIM script stored in the macro file is displayed in the Macro field. You can change or delete existing code or add more code to this file.

In the Description field, you can enter text to describe the function of the macro file. This text is displayed in the AIM Macro Recorder Description field whenever this file is highlighted.

Use the **File** menu to save the file or close the file without saving any changes. If you use a new name for a saved file, that new file is listed the next time you open the AIM Macro Recorder.

To save the changed text without closing the Edit window, click the **Apply** button.

Macro Recorder Examples

CosmosScope numbers each graph window and waveform according to where you are in a session. If you play back a recorded macro in a different CosmosScope session, the numbered designations in the macro may not match the numbering scheme in the new session and could generate error messages.

When using the Macro Recorder with CosmosScope, the **Replace Handles/ Tags with Variables** option corrects this problem by replacing the numbered designations with variables. You do not have to edit the macro when you enable this option except for the activeframe variable needed for AimPrint.

In order to guarantee an accurate playback, you must make a few changes to the recorded script. The following examples demonstrate some suggested edits:

- Macro Recorder Example 1: Running a DC and Transient Analysis
- Macro Recorder Example 2: Performing Measurements on Selected Signals

Macro Recorder Example 1: Running a DC and Transient Analysis

Recorded Macro 1 records the following tasks in this order:

1. Running a DC and a transient analysis
2. Opening the Signal Manager
3. Loading a specific waveform into a graph window
4. Turning on the axis grids
5. Measuring the overshoot
6. Printing the graph window

Recorded Macro 1

```
Saber:Send {dc
tr (dfile tr1,pfile tr1,tend 50m,tstep 1u}
Saber:Send {pl (placation openonly}
ScopeSigMgr:loadpffile /tmp/RLC/ex_rlc.tr1.ai_pl 2 openonly
GrXY:NewGraph
pf:read PF:1 vout
Graph addsignal WF:1:1 -region new -yview real(y) -xview real(x)
   -tracehi 2
Graph axisconfig {AxisY(0,0) AxisX(0)} -gridvis {yes yes}
Graph config -grid yes
GrMeas:Overshoot Graph0 Signal0 default default absolute xyrangeAll
Graph:PrintCB mono
AimPrint:Hardcopy -image .frame1.mdi.mdi1.work.graph1 -prname lw3
   xoffset 0.3i -yoffset 0.3i
Graph:PrintCB restore
```

In Recorded Macro 1, the user records running a dc and a transient analysis, opening the Signal Manager, loading a specific waveform into a graph window, closing the Signal Manager, turning on the axis grids, opening the Measurement tool, measuring the overshoot, closing the Measurement tool, and printing the graph window.

Selecting the "Replace Numbered Handles/Tags with Variables" Option for Recorded Macro 1 (CosmosScope Only)

The following macro is the result of recording the same procedure as Recorded Macro 1, except with the Replace Numbered Handles/Tags with Variables option selected:

```
Saber:Send {dc
tr (dfile tr1,pfile tr1,tend 50m,tstep 1u}
set pf [ScopeSigMgr:loadpffile /tmp/RLC/ex_rlc.tr1 2 openonly]
set graph1 [GrXY:NewGraph]
set wf [pfread $pf {vout}]
set sig0 [Graph addsignal $wf -region new -yview real(y) -xview real(x)
   -tracehi 2]
Graph axisconfig {AxisY(0,0) AxisX(0)} -gridvis {yes yes}
Graph config -grid yes
set meas1 [GrMeas:Overshoot $graph1 $sig0 default default absolute
   xyrangeAll]
Graph:PrintCB mono
AimPrint:Hardcopy -image .frame1.mdi.mdi1.work.graph1 -prname lw3
   xoffset 0.3i -yoffset 0.3i
Graph:PrintCB restore
```

Edited Macro 1

```
Saber:Send {dc
tr (dfile tr1,pfile tr1,tend 50m,tstep 1u}
set pf [ScopeSigMgr:loadpffile /tmp/RLC/ex_rlc.tr1]
set wf [pfread $pf vout]
GrXY:NewGraph
Graph addsignal $wf -region new -tracehi 2
Graph axisconfig {AxisY(0,0) AxisX(0)} -gridvis {yes yes}
Graph config -grid yes
GrMeas:Overshoot $Graph(graph)  Signal0 default xyrangeAll
Graph:PrintCB  mono
set activeframe [set $Graph(graph)]
AimPrint:Hardcopy -image $activeframe -prname lw3 -xoffset 0.3i
   -yoffset 0.3i
Graph:PrintCB  restore
```

In Edited Macro 1, the pf and wf variables have been created to load the desired waveform into a graph at any point in a Scope session. The activeframe variable has been created to print the current graph window at any point in a Scope session. Edited Macro 1 also removes some superfluous recorded commands.

The following table compares the recorded macro with the edited macro. Bolded text indicates edited code.

Recorded Macro 1	Edited Macro 1

`Saber:Send {dc`
`tr (dfile tr1,pfile tr1,tend 50m,tstep 1u}`
Finds the DC operating point and runs a transient analysis.
	`Saber:Send {dc`
	`tr (dfile tr1,pfile tr1,tend 50m,tstep 1u}`
	Finds the DC operating point and runs a transient analysis.

`Saber:Send {pl}` Sends the plot file to CosmosScope, opens the Signal Manager, and loads the plot file into the signal list.	`set pf [ScopeSigMgr:loadpffile /tmp/RLC/ex_rlc.tr1]` Opens the Signal Manager, loads the plot file into the signal list, and assigns the variable `pf` as a plot file descriptor pointing to the plot file `ex_rlc.tr1` that was just loaded.
`GrXY:NewGraph` Opens a new graph window.	`GrXY:NewGraph` Opens a new graph window.
	`set wf [pfread $pf vout]` Selects the `vout` signal from the signal list, and assigns it to the variable `wf` as a waveform descriptor, which allows the waveform to be manipulated later.
`Graph addsignal WF:3:6 -region new -tracehi 2`	`Graph addsignal $wf -region new -tracehi 2` Graphs the waveform designated in the `wf` variable (namely vout) into a new region on the active graph.

`Graph axisconfig {AxisY(0,0) AxisX(0)} -gridvis {yes yes}`
`Graph config -grid yes`
Turns on the axis grids.
	`Graph axisconfig {AxisY(0,0) AxisX(0)} -gridvis {yes yes}`
	`Graph config -grid yes`
	Turns on the axis grids.

Recorded Macro 1	Edited Macro 1

`GrMeas:Overshoot Graph0`
`Signal0 default xyrangeAll`
Measures the overshoot on graph window `Graph0`.
	`GrMeas:Overshoot $Graph(graph)`
	`Signal0 default xyrangeAll`
	Measures the overshoot on the active graph window.

`Graph:PrintCB  mono` Converts the graph window colors to black and white.	`Graph:PrintCB  mono` Converts the graph window colors to black and white.

	`set activeframe [set`
	`$Graph(graph)]`
	Creates the `activeframe` variable, which contains the canvas description of the active graph.

`AimPrint:Hardcopy -image`
`.frame1.mdi.mdi1.work.`
`graph0 -prname lw3`
`-xoffset 0.3i -yoffset 0.3i`
Prints `graph0`.
Sends a print job to the `lw3` printer and restores the graph window colors. The monochrome and printer settings are established in the Printer Setup dialog box; otherwise, these logged commands are typical when choosing **File > Print**.
	`AimPrint:Hardcopy -image`
	`$activeframe -prname lw3`
	`-xoffset 0.3i -yoffset 0.3i`
	`Graph:PrintCB  restore`
	Prints the graph designated in the activeframe variable.
	Sends a print job to the `lw3` printer and restores the graph window colors. The monochrome and printer settings are established in the Printer Setup dialog box; otherwise, these logged commands are typical when choosing **File > Print**.

`Graph:PrintCB restore` Restores the graph window colors.	`Graph:PrintCB restore` Restores the graph window colors.

Macro Recorder Example 2: Performing Measurements on Selected Signals

This example shows how to create a script from a recorded macro that performs measurements on any selected signals in CosmosScope.

Recorded Macro 2

```
GrXY:NewGraph
ScopeSigMgr:loadpffile /tmp/RLC/ex_rlc.tr1 2 openonly
pf:read PF:1 vout
Graph addsignal WF:1:1 -region new -yview real(y) -xview real(x)
   -tracehi 2
Graph axisconfig {AxisY(0,0) AxisX(0)} -gridvis {yes yes}
Graph config -grid yes
GrMeas:Overshoot Graph0 Signal0 default default absolute xyrangeAll
```

In Recorded Macro 2, the following tasks are recorded:

1. Opening a new graph window
2. Opening the Signal Manager
3. Loading a specific waveform into the graph window
4. Turning on the axis grids
5. Measuring the overshoot

Selecting the "Replace Numbered Handles/Tags with Variables" Option for Recorded Macro 2 (CosmosScope Only)

The following macro is the result of recording the same procedure as Recorded Macro 2, except with the Replace Numbered Handles/Tags with Variables option selected

```
set graph1 [GrXY:NewGraph]
set pf1 [ScopeSigMgr:loadpffile /tmp/RLC/ex_rlc.tr1 2penonly]
set wf1 [pf:read $pf1 {vout}]set sig0 [Graph addsignal $wf1 -region new -
   yview real(y) -xview real(x)-tracehi]
Graph axisconfig {AxisY(0,0) AxisX(0)} -gridvis {yes yes}
Graph config -grid yes
set meas1 [GrMeas:Overshoot $graph1 $sig0 default default absolute
   xyrangeAll]
```

Edited Macro 2

```
set signals [selection get WF]
foreach sig $signals {
set graph [GrXY:NewGraph]
Graph addsignal $sig -region new -tracehi 2
Graph itemselect Signal1 add
Graph axisconfig {AxisY(0,0) AxisX(0)} -gridvis {yes yes}
Graph config -grid yes
GrMeas:Overshoot $graph Signal0 default xyrangeAll
}
```

In Edited Macro 2, the signal variable is created to accept highlighted signals from the Signal List in the Signal Manager. The graph variable is created to point to a new graph window. A waveform loads, and an overshoot measurement is performed on each graph. A loop command repeats the open, load, and measure sequence on all selected signals.

To run Edited Macro 2:

1. Open the Signal Manager.
2. Open one or more plot files.
3. Select any number of signals from the signal list.
4. Select the macro from the Name list in the Macro Recorder tool.
5. Press the **Play** button to run the macro.

The macro opens a graph window for each selected signal and performs an overshoot measurement on each signal.

The following table compares the recorded macro with the edited macro. Bolded text indicates edited code.

Recorded Macro	Edited Macro

`ScopeSigMgr:loadpffile /tmp/RLC/ex_rlc.tr1`
Opens the Signal Manager tool and loads the plot file into the signal list.
	`set signals [selection get WF]`
	Creates the variable signals, which contains all selected signals from the Signal Manager tool Signal List.
	`foreach sig $signals {`
	Creates a loop that performs all of the following operations on each selected signal.

`GrXY:NewGraph` Opens a new graph window.	`set graph [GrXY:NewGraph]` Creates the variable graph which points to a new graph window.

`Graph addsignal WF:3:6 -region new -tracehi 2`
`wm withdraw .sigmgr._PF:3`
Loads waveform WF:3:6 into the graph window.
	`Graph addsignal $sig -region new -tracehi 2`
	Loads the waveform designated in the sig variable into the graph window.

	`Graph itemselect Signal1 add` Selects the second signal in a graph region if more than one signal is displayed (the first signal is signal 0).

`Graph axisconfig {AxisY(0,0) AxisX(0)} -gridvis {yes yes}`
`Graph config -grid yes`
Turns on the axis grids.
	`Graph axisconfig {AxisY(0,0) AxisX(0)} -gridvis {yes yes}`
	`Graph config -grid yes`
	Turns on the axis grids.

`GrMeas:Overshoot Graph0 Signal0 default xyrangeAll`
Measures the overshoot on the signal.
	`GrMeas:Overshoot $graph Signal0 default xyrangeAll`
	`}`
	Measures the overshoot on the signal.

Macro File Naming and Directory Conventions

The Macro Recorder uses the following file naming and directory conventions:

- Macro file extension: filename.ai_mcr

  Only files with the ai_mcr extension are considered by the Macro Recorder for macro definition. To utilize previously created files, add the ai_mcr extension to their filename. The internal format remains unchanged.

- Creation of .aimMacro directory and content

  The default location macro directory, .aimMacro, is in the user home directory. This directory is now automatically created when the Macro Recorder is opened. If user_grp files have been installed, the contents of the AimMacro subdirectory is copied verbatim to .aimMacro.

- Application-specific macro directory

  Macro files (filename.ai_mcr) located in ~/.aimMacro directory are considered common to all applications. Application-specific macro files are located in the corresponding sub-directory. All macros created using the Macro Recorder are saved in the corresponding application-specific directory.

- User-defined macro files location preferences

  You can define locations for macro files using either ~/.aim_user or aim.site files by adding the following line to the file:

  ```
  AimMacro(prefs) directory list
  ```

  For example, the following line indicates that directory1 and directory2 must be checked for .ai_mcr files:

  ```
  AimMacro(prefs) {directory1 directory2}
  ```

- Macro Recorder directory precedence

  The Macro Recorder checks for .ai_mcr files in the directories, in the following order:

  a. Local directory

  b. User-defined directories

  c. Application-specific directory

  d. ~/.aimMacro directory

e. First definition—when multiple definitions for the same macro exist, only the first one is considered.

10

Using the SaberRT Interface

This chapter describes the Saber Simulator Real-Time Interface.

Overview

SaberRT is an interface to the Saber Simulator allowing Saber designs to be simulated interactively in a real-time or hardware-in-the-loop context. Previously, Saber sources were deterministic functions of time. With SaberRT, they become interactive and can be altered as the time-domain analysis proceeds. Changes in stimulus values take immediate effect in the simulation. The design response can be monitored through gauges, walking waveforms, or custom animations. SaberRT relies on the proven algorithms of the Saber Simulator, and minimal changes to Saber designs are required.

Off-line Design Validation and Calibration

SaberRT can be invoked from Saber Sketch or Saber Harness and used as a design validation tool or as an interactive design test bench. Test vectors can be recorded and re-applied to Saber designs, ensuring improved test coverage. Model parameters can be directly altered through the SaberRT interface.

SaberRT also allows a simulation to be calibrated with respect to sampling period, truncation error, sample point density, and other simulation parameters. Achieving optimum speed/accuracy trade-off is essential to real-time simulation. The real-time profiling function of SaberRT helps determine whether the system can be simulated in a real-time manner.

Hardware-In-The-Loop (HIL) Verification

After the simulation has been calibrated, you can generate a C interface of the model. This interface connects the model to real-time systems, such as manually-controlled systems, or HIL equipment. The SaberRT model then acts as a virtual component interacting with real hardware, allowing system testing in early design stages when not all components are available.

Making Simulation Easy for Non-Expert Users

Pre-setup simulations can be run by users unfamiliar with Saber, allowing them to control design parameters and stimuli. This can allow marketing teams, for example, to demonstrate a concept or a product to potential customers.

Using SaberRT

This section describes how to use SaberRT. It is divided into the following topics:

- Using Designs with SaberRT
- The Model C Interface
- Creating Animations with SaberRT

Using Designs with SaberRT

To run a design with SaberRT, you first need to replace some of the deterministic—predefined functions of time—sources in your design with the appropriate SaberRT real-time sources.

SaberRT Real-Time Stimulus Sources

Similar to the Hypermodels used by Saber for co-simulation with digital simulators, the SaberRT stimulus templates rely on foreign state variables to receive the design input values. Design output is returned through the probe function of the Saber Mixed-Mode Interface (SMMI) so that no probe models are needed. "The Model C Interface" has more information on the SMMI.

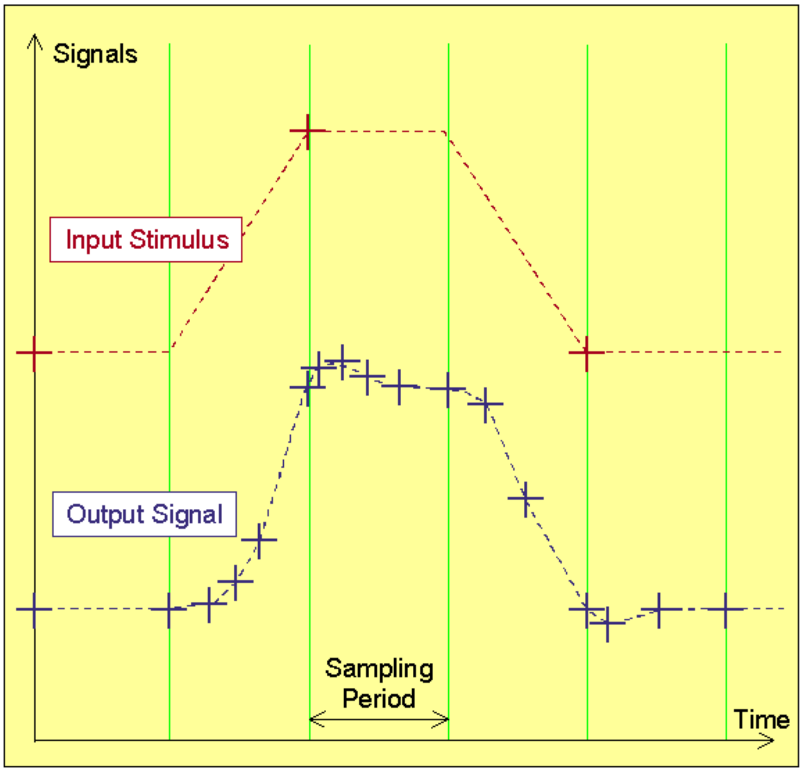
The MAST implementation of the voltage stimulus is shown below.

```
element template real2v p m = dt
electrical p,m
number dt=20m         # sampling period
{
   foreign state nu value
   state nu clock,
          slope,
          tn=0,    # next time point
          vn=0     # next stimulus value
   val v v
   var i i

   when (dc_done) {
      schedule_event(0,clock,1)
   }
   when (event_on(value)) {
      slope=(value-vn)/dt
      vn=value
      if (time==0) tn=time
      else tn=time+dt
      schedule_next_time(time)
      schedule_next_time(tn)
   }
   when (event_on(clock)) {
      schedule_event(time+dt,clock,1)
      schedule_next_time(time)
   }
```

```
    values {
       if (time>tn) v=vn
       else v=vn+(time-tn)*slope
    }
    equations {
       i(p->m)+=i
       i:v(p,m)=v
    }
}
```

Except for the digital and state sources (real2bit and real2state), the SaberRT stimuli are piece-wise linear function of time. The sampling period dt defines the update rate of the model. In each stimulus template, a clock combined with the schedule_next_time function forces the design to be evaluated at each clock tick. This ensures that the model is updated at regular time intervals. The passing of new input through the foreign state variable value is also synchronized with the clock, but it does not necessarily occur at each clock tick. The model can simulate for several sampling periods without receiving any new input. In such a case, the stimulus model maintains the last input value received. Note also that despite a constant sampling period, Saber still uses its variable time step algorithm. As shown in the figure below, the simulator may evaluate the design more than once within a sampling period in order to meet its accuracy criteria.



Sampling period

The table below gives the list of stimuli available:

Table 1 Stimulus Templates 1 of 2

Variable type	Source
Digital	real2bit
State/Z domain	real2state
Control	real2ctrl
Conservative	(see below)

Table 2 Stimulus Templates 2 of 2

Technology	Across variable source	Through variable source
Electrical	real2v	real2i
Magnetic	real2mmf	real2flux
Translational	real2pos	real2force
Rotational Ang.	real2angle	real2torque
Rotational Vel.	real2w	real2torque
Hydraulic	real2pressure	real2flow
Thermal	real2temp	real2power

Running a Design with SaberRT

There are three steps to running a design with SaberRT:

1. Inserting stimulus sources
2. Creating a SaberRT Model
3. Running Interactive Simulations

Inserting Stimulus Sources To insert stimulus sources:

1. Insert a SaberRT stimulus in the design either directly, by modifying the netlist, or on the schematic using the schematic capture tool.

   Stimulus sources, representing the interactive input stimuli to your model, can be found in the Parts Gallery under the category:

   ```
   /MAST Parts Library/SaberRT Stimuli
   ```

2. Give a meaningful reference designator to the stimulus since this will be used as a label for the stimulus.

   Also, the reference designators assigned to SaberRT stimuli should not be shared by other models in the netlist, as this would cause an error during dc analysis.

   Once the interactive sources have been inserted, the design is ready to be loaded into SaberRT.

Creating a SaberRT Model To create a SaberRT Model:

1. In Saber Sketch and Saber Harness, SaberRT is invoked from the Tool Bar. It is also available from the Tools menu of Parts Gallery. (This is the only method available for the Frameways.)
2. Select File > New to invoke a wizard that asks for the design. SaberRT automatically identifies the real-time sources from the design netlist.

   To complete the wizard:

   a. First, select the desired design by browsing to the directory and selecting the file.
   b. Enter the initial value, range, and resolution for each stimulus. This information is used to configure the scale widgets in the stimuli tab form. Note that only the initial value is required for digital stimuli (real2bit). The domain of definition for an initial value is {0, 1, x, z}.
   c. Next, define the model output (or probe signals) by selecting the signals to include in the Probe List.
   d. Specify the sampling period (update rate of the model), as well as the simulation parameters (time step, sample point density, and maximum truncation error). Note that you can re-enter the values that you assigned to the simulation parameters by selecting the Edit > Calibration... menu choice.
   e. Click Finish.
3. The Stimuli tab on the left side of the SaberRT application window shows the list of stimulus scale widgets. Double-clicking a stimulus label invokes a form that allows range and resolution to be re-defined. This can also be accomplished with the Edit > Stimuli... command.
4. The Parameters tab is initially empty. Selecting Edit > Parameters... allows design parameters to be altered. Note that parameters get effectively altered when the simulation is re-started from the dc operating point. In order to avoid discontinuity, SaberRT does not allow parameters to be altered in the middle of a transient analysis.
5. The right field of the window is the probe widget area. This field is initially empty. The View menu allows probes to be selected among the list of output signals.
6. Save the setup of your SaberRT model by selecting File > Save.

   Your model is now ready to be simulated interactively through the SaberRT user interface.

Running Interactive Simulations To run interactive simulations:

1. Load the design by selecting Simulation > Init. Saber performs the dc operating point analysis.
2. Start the transient analysis by selecting Simulation > Start.
3. You can alter the stimulus values using the scale widgets as the simulation proceeds.

   The design response is observed through probe widgets. Clicking with the right mouse button on a probe allows a type of display to be selected among gauge, walking waveform, horizontal or vertical meter. Other probe configurations are available from the same menu, such as logarithmic scale or viewing range.
4. The simulation can be interrupted at any time by selecting Simulation > Stop, or the Stop icon.

   The scale widget on the icon bar allows you to roll-back in time and observe the model state at previous instants of simulation.
5. With the File > Export > TLU Source... selection, it is possible to save any stimulus or probe signal as a Table Look-Up model. This model can subsequently be used to replace an interactive source for off-line simulations with Saber.
6. The Edit > Plotfile Signals... selection allows you to log simulation results in a plotfile.

   To gain speed, it is recommended to have no plotfile signal selected when running real-time simulations.
7. Select Simulation > Continue to restart the simulation from the time when it was interrupted.

Calibrating the Model

SaberRT allows the simulation to be calibrated for best convergence and speed performance. The Edit > Calibration... form gives access to a list of simulation parameters including:

- Sampling Period: defines the update rate of the model, which is the frequency at which the Saber simulator receives stimulus values and returns output values. The sampling period implicitly defines the simulation maximum time step. Increasing the sampling period typically helps improve the real-time capability of the model.
- Time Step: by default, can be set equal to the sampling period.
- Sample Point Density: used by Saber to approximate non-linear characteristics with piece-wise linear equivalents. This parameter primarily affects the accuracy of the simulation.
- Maximum Truncation Error: affects both the speed and the convergence performance of the design. This parameter is of primary importance for model calibration.
- Truncation Error Norm: the default value 6 usually yields accurate results for electronics systems. Mechanical designs including multi-body systems can simulate better with the truncation error norm set to 3.
- Trace File: this debugging option is not a simulation parameter. Set to yes, it allows the function calls to the SMMI interface to be logged in a C file (smmitrace_***.c). The SMMI interface handles the interprocess communication between the SaberRT GUI and the Saber simulator (see Model C Interface section). Be aware that the trace option can slow down the simulation and should only be used for debugging purposes.

Real-Time Profiling

Achieving optimum speed/accuracy trade-off is essential to real-time simulation. The real-time profiling function of SaberRT helps determine whether the model is real-time capable. This function is available under the Tools menu of SaberRT.

The real-time profile curve is an instantaneous measure of the real-time capability of the simulation. This curve does not account for any graphical display update and shows the intrinsic performance of the simulator. The x-axis is the simulation time and the y-axis is the ratio of the time needed by the simulator to compute the transient analysis over a sampling period dt divided by dt.

profile(t)=(CPU time of simulation step)/dt

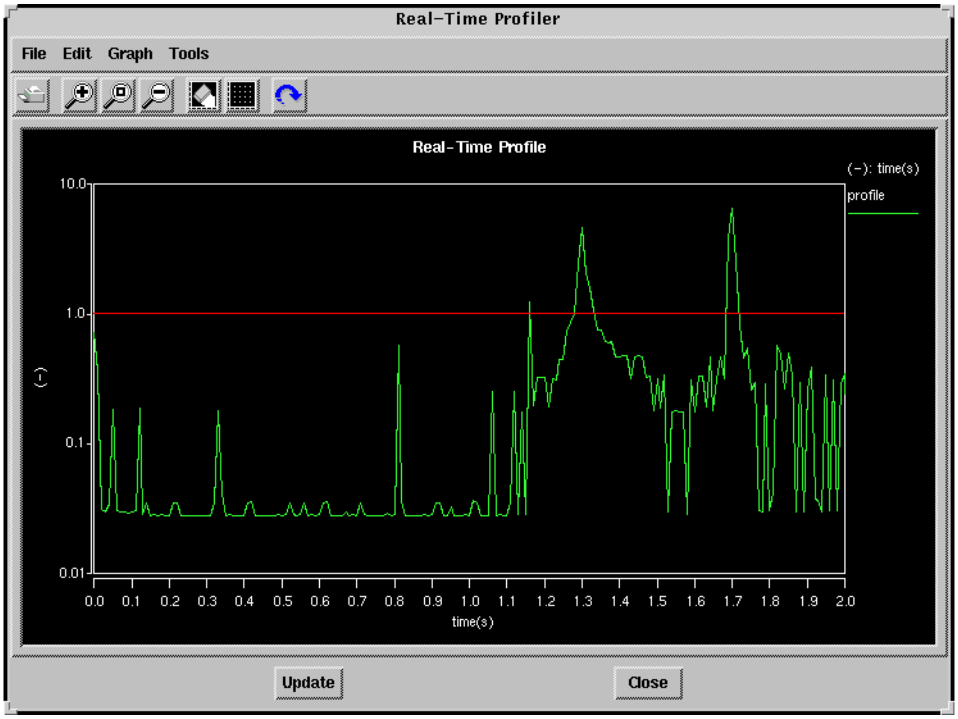
When this ratio is greater than 1, the time needed to simulate a sampling period is larger than the sampling period. This means that, in a real-time context, the simulator will not be able to return the output values to the simulation backplane in due time. The model will have to be revisited or calibrated in order to lower the profile and increase the latency.

Undoubtedly there is a limit to design size, stiffness, and time constant spread to achieve real-time capability. To ensure that your design simulates within this limit, you may consider:

- reducing its complexity
- choosing simpler models
- increasing the truncation error
- increasing the sampling period
- using a faster computer
- stopping all other system activity taking CPU resource

The File > Export > Graph... and File > Import > Graph... selections allow model performance to be compared between SaberRT sessions (possibly on different computers). Within a session, real-time profiles can be appended by changing the view update mode (View > Update Preferences...).

The profile curve shown in the figure below goes several times above 1. These are instants when the model loses its real-time capability. "The Model C Interface" discusses how the simulation back-plane should be prepared to handle such situations.



Real-time profile

The Model C Interface

The Model C Interface is intended for simulation outside the SaberRT user interface, in a real-time environment such as on a hardware-in-the-loop platform. This C interface can also be used for co-simulation purposes.

Exporting the Interface

The File > Export > C Interface... command brings a wizard that first asks for a type of interface. The Generic C Interface is the default one. Third party interfaces are not discussed in this document.

You then specify whether your simulation configuration is local or remote. In a remote configuration, the simulation process and the process calling the interface are running on separate computers through a socket connection. You also need to specify the operating system on which Saber is running (UNIX, Windows or Linux).

If you choose a remote configuration, you need to specify the host name of the computer where Saber is intended to run. Either the domain name or the IP address is acceptable as the host name.

You are then requested to specify a directory location for saving the interface files. At this point, you can rename the interface so that it does not share the same name as the SaberRT document (which by default is the design name). This is necessary if you intend to concurrently run several model instances based on the same design.

A list box shows the files which are part of the interface. The README.c, the stimulus file, the model.scs file and the model.h file are the only files which are model specific. The stimulus file contains the stimulus profiles played in the tool before exporting the interface. These profiles are used by the README.c program.

Using the Generic C Interface

The generic C interface consists of one function which has a prototype of:

```
int model (int cycle, double *input, double *output);
```

The first argument is the cycle number. The corresponding simulation time is cycle*dt where dt is the sampling period. The initial call of the function should have this argument set to zero, corresponding to the initialization of the model. The design is then loaded in Saber, and the dc analysis is performed for the specified input values.

Note that if some design parameters were altered in the SaberRT tool before exporting the interface, these parameters would be altered the same way during this initialization phase. The alter commands as well as the simulation settings are actually retrieved from the scs file created when generating the C interface.

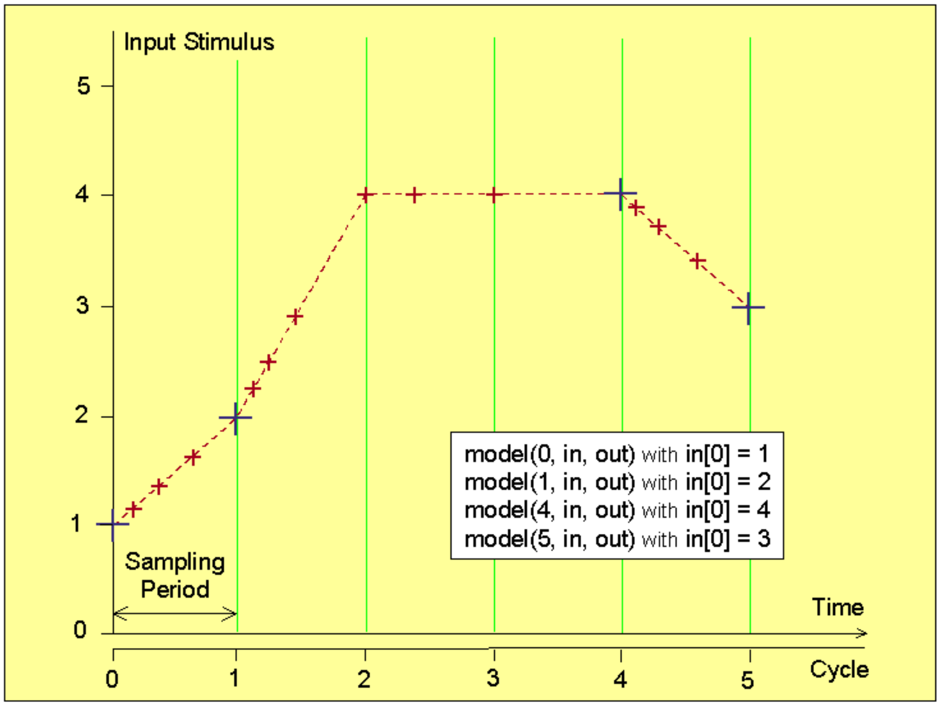
Subsequent function calls only accept strictly increasing cycle values (except for cycle = -1, which is used to terminate the simulation). The difference in cycle values between two consecutive calls can be greater than 1, allowing faster progress in simulation time. This is desired in a real-time context when the simulation lags behind the clock.

The input array contains the stimulus values for the current cycle time. For each function call, the simulator performs a time domain analysis between the previous cycle time and the current cycle time. If this time interval is equal to dt, the analog stimulus sources are linear interpolations between previous and current time points. If it is a multiple of dt, the interpolation takes place during the first sampling period, and the stimuli are maintained constant for the remaining periods. Zero-delay events associated with event-driven stimuli (real2bit and real2state) take place at the current time cycle*dt. When the simulation is completed, the function returns the probe values computed at the current time into the output array.

The input and output values respectively taken by digital stimuli (real2bit) and logic probes (logic4) are integer values (coded as double in the input and output arrays). Their domain of definition can be set by the user from the menu “Edit>Logic Mapping Table...”. By default, the 0 logic state is mapped to the integer value 0, logic 1 to 1, logic x to 2 and logic z to 3. If a digital stimulus (real2bit instance) receives a value outside the mapping table (like 3.14159), the stimulus generates the x logic state.

To terminate the simulation and unload the model, the function must be called with cycle = -1.

The function returns a status flag 1 if OK, 0 in case of an error.



Interface calls

A sequence of calls to a model interface that includes one continuous stimulus is shown in the figure above. The corresponding stimulus profile illustrates how the interface handles jump in cycles (from cycle 1 to cycle 4).

Remote Simulation

If you choose a remote configuration, make sure that you invoke SaberRT on the local computer in the directory where the design resides, before calling the C interface on the remote computer. Then select the Tools > Simulation Server.... menu choice. On Windows, also make sure that your design is not currently used by a session (including the SaberRT tool).

When the C interface executes the model initialization from the remote computer, a signal is sent to the SaberRT simulation server through a socket channel, starting the Saber process. At this point a new socket channel is established between Saber and the simulation backplane (remote process). This socket handles the passing of model input/output values.

Note:

> In the first release of SaberRT, you can not have the C interface running on Windows and the design running on a remote computer.

Error Tracing

To trace an error (returned status flag 0), you can:

1. Look into the output file (*.out).
2. Invoke the C interface under the trace mode. To activate this mode, you need to set the Trace File option to yes in the calibration form (Edit > Calibration...) before exporting the interface. In this mode, the function calls to the SMMI interface are logged in a C file (smmitrace_***.c). Be aware that this mode can slow down the simulation and should only be used for debugging purposes.

The Saber Mixed-Mode Interface (SMMI)

The Saber Mixed-Mode Interface is a C API which is primarily used for co-simulation between Saber and digital simulators. It handles the interprocess communication between Saber and a partner process. SaberRT makes use of the SMMI protocol in a configuration where Saber is the slave process.

The smmi.h file lists the functions available in the API. The SMMI source code is provided to allow more flexibility in porting or installing the model interface on platforms currently not supported by Saber. This should also make it easier to modify the generic interface for user specific requirements.

Example of Real-Time Implementation: README.c

The SaberRT graphical interface performs simulation in an interactive fashion but not in real-time. Running real-time simulation means slowing down the simulation with respect to a real clock. For this to be possible, the design should simulate faster than real-time. However, the simulation back-plane calling the interface must be ready to handle situations when the simulator lags behind the clock.

The program README.c generated with the interface is an example of a real-time implementation of the generic C interface. Below is the README.c file on Windows obtained with the demo grand_prix (File > Install > Demo Examples...). The header of the file contains compilation instructions (not shown below).

```
#define _WIN32_WINNT 0x0400
#define WINVER 0x0400
#include <windows.h>
#include <mmsystem.h>
#include <winbase.h>
#include "grand_prix.h"

main()
{
   FILE *f;
   char line[256];
   int cycle, maxCycle=0, nbLine=0;
   LARGE_INTEGER frequency, start, current;
   HANDLE timer;
   double **stimulus, input[1], output[9], time_diff, dt=0.025;
```

```
/* Load the stimulus values from the stimulus file */
  f=fopen("grand_prix.stimulus","r");
  if (f == NULL) {
     printf("Can not open stimulus file\n");
     exit(0);
  }
  stimulus = (double **)malloc(1*sizeof(double *));
  while (fgets(line, 256, f) != NULL) nbLine++;
  stimulus[0] = (double *)malloc(nbLine*sizeof(double));
  rewind(f);
  while (fgets(line, 256, f) != NULL) {
     if (sscanf(line, "%lf ", \
        &stimulus[0][maxCycle]) == 1) maxCycle++;
  };


    fclose(f);

  if (maxCycle == 0) {
     printf("No data in stimulus file\n");
     exit(0);
  }

  /* Load the model in Saber and perform the dc analysis */
  if (grand_prix(0, input, output) == RT_ERROR) {
     printf("Error at time 0 second\n");
     exit(1);
  }

  cycle=1;
  timer = CreateWaitableTimer(NULL, FALSE, NULL);
  QueryPerformanceFrequency(&frequency);
  QueryPerformanceCounter(&start);
```

```
while (cycle < maxCycle) {
    printf("cycle = %d\n", cycle);
    /* Evaluate the model */
    input[0] = stimulus[0][cycle];
    if (grand_prix(cycle, input, output) == RT_ERROR) {
        printf("Error at time %g second\n", cycle*dt);
        exit(1);
    }
    /* Check timing */
    QueryPerformanceCounter(&current);
    time_diff = ((double)(current.LowPart-start.LowPart))/ \
        ((double)frequency.LowPart)-cycle*dt;
    if (time_diff < 0) {
        /* Pause the simulation for "time_diff" */
        WaitForSingleObject(timer, (DWORD)(-time_diff*1e3));
        cycle += 1;
    } else {
        cycle += 2+(int)floor(time_diff/dt);
        printf("...\n");
    }
}
free(stimulus[0]);
free(stimulus);
```

The program README.c is based on the grand_prix model which contains 1 input (thrust) and nine output values. The program performs in real-time the last interactive simulation run in SaberRT before the interface was generated. The update period dt is equal to 25 ms.

The thrust stimulus profile is first retrieved from the grand_prix.stimulus file and stored in the stimulus array. The program then loads the design and performs the dc analysis. The time before the start of the transient analysis is recorded (QueryPerformanceCounter). This reference time will be used to establish whether the model runs real-time or not.

The program then enters a loop that exits when reaching the last pre-recorded cycle. Within this loop, the model is simulated from the previous cycle time to the current cycle time. The program then computes the difference between the overall real time elapsed since the beginning of the simulation and the current simulation time cycle*dt. If this difference is smaller than zero, the simulation, which proceeds faster than real-time, is paused and resumed after the lead time over real-time has elapsed (WaitForSingleObject). The cycle number is incremented by 1 for the next iteration of the while loop.

If the simulation is running late, the cycle number is increased by a number of sampling periods corresponding to the gap between real-time and simulation time. In the nested loop iteration, the simulation will simulate a larger time interval during which the input stimulus will be maintained as a constant. This helps the simulation catch up with the clock. Be aware that whenever a cycle jump larger than 1 occurs, pre-recorded stimulus values are ignored, which could potentially result in divergent or unreliable simulation results.

Creating Animations with SaberRT

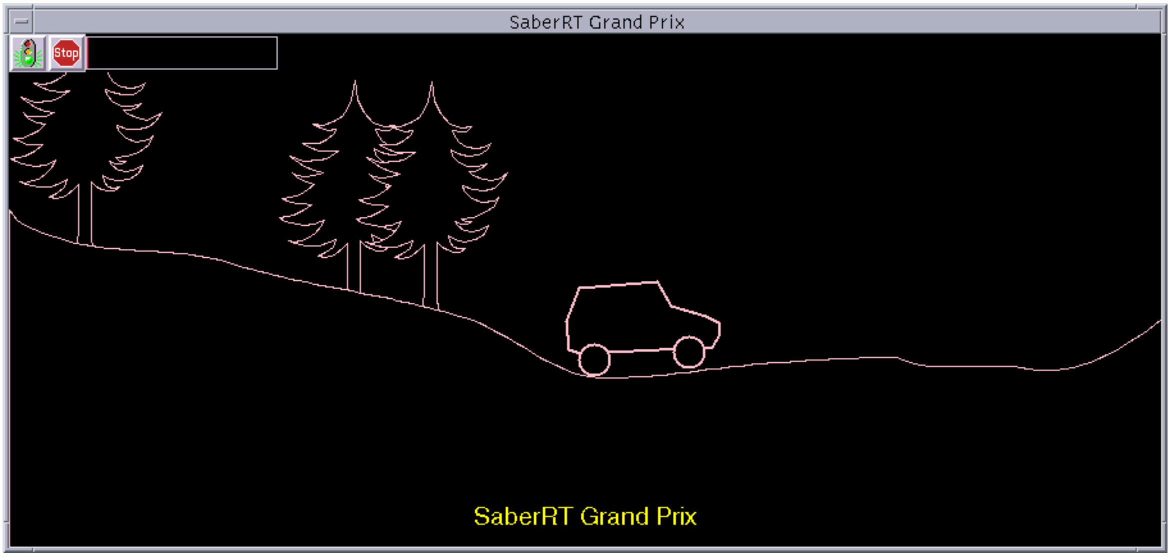
SaberRT offers a means to animate your simulation by allowing custom Tcl/Tk scripts to be executed within a simulation process.

Exploring the SaberRT demos

Clicking File > Install > Demo Examples... or Help > Demo Examples... gives you access to several demos that you can run interactively. Some of these demos (such as the SaberRT Grand Prix demo shown in the next figure) accept input from the joystick on Windows.

The Tcl/Tk or AIM scripts associated with these demos can be viewed and edited by clicking the Edit AIM Script icon. These scripts are provided as examples that you can adapt for your own animations.

You can disable an animation by toggling the Edit > Run AIM Script menu choice.



Example of a SaberRT animation

AIM Script Template

The template of an animation script can be generated for a given SaberRT model by clicking the File > Export > AIM Script... command.

This template contains a global array Model and two procedures Model:Init and Model:Update where your custom code should be inserted.

The global array Model is intended to store persistent information from one procedure call to the next.

The procedure Model:Init is called after the dc operating point analysis. Its arguments are the incrTcl saberRT object followed by the list of model input/output values at time 0. The saberRT object is typically stored in the Model array so that it can be referenced in the Model:Update procedure. This object gives you some control over the simulation process from your script. For example, you can stop the simulation with:

```
$saberRT terminate
```

The procedure Model:Update is called after each dt transient analysis. Its arguments are the current simulation time followed by the list of model input/output values at that time.

The Model:Init and Model:Update procedures do not return anything.

Frequently Asked Questions

1. Question: How does the SaberRT remote simulation work?

   Answer: When invoking the design from a remote computer, you need to make sure that the design is not already used by another Saber session, including the current SaberRT session. In such a case, you will have to exit SaberRT (File > Exit), then re-invoke SaberRT in the directory where the design resides and select the Tools > Simulation Server... menu choice.

   The C function produced by SaberRT is compiled on the real-time platform. When this function is executed, a signal is sent to the SaberRT simulation server through a socket channel that starts the Saber process and loads the design. At this point a new socket channel is established between Saber and the simulation backplane to exchange the model input/output values.

2. Question: Can SaberRT handle multiple models?

   Answer: Several models can be combined into one design, at the expense of speed. A large design that includes independent models will simulate more slowly than smaller designs running with separate SaberRT (especially if the simulation of the models is distributed across several computers). There is no automation to combine several models into one design. This will have to be done the traditional way in Saber Sketch (copy/paste).

   Note that the SaberRT model interface is control flow. Problems will arise if you want to simulate a virtual model which is driven through conservative nodes. Getting the right loading effects may still be a challenge with SaberRT. SaberRT is well fit for simulating islands of conservative systems in which the interface is control-flow.

3. Question: What is the impact of socket communication between SaberRT and RT platform on the overall simulation performance?

   Answer: The impact should be quite acceptable for sampling periods above 10 ms.

4. Question: Does the real-time profile take the communication overhead into account?

Answer: To some extent, it does. When running the simulation interactively, Saber is a slave process receiving the input values from, and returning the output values to, the SaberRT user interface process. This interprocess communication is based on sockets. When running interactive off-line simulations, the SaberRT user interface and Saber processes are running on the same computer. The real-time profile is therefore not representative of a remote simulation where the interprocess communication would take place across a network.

5. Question: Does the real-time profile account for the time spent running the animation script?

   Answer: No. The real-time profile measures the time difference between the instant when the input values are sent to Saber and the instant when the Saber output values are received. The time taken by the subsequent execution of an animation script is not included in this measurement. It also does not account for the update of the probe widgets.

6. Question: Does the SaberRT generic C interface allow simulation time back-tracking?

   Answer: No, the cycle values passed to the interface function must be strictly increasing.

7. Question: Is it possible to run a model with no stimulus sources in SaberRT?

   Answer: Yes. Your design still needs a clock element that enforces the simulator to evaluate the system at a rate equal to the sampling period This will set a maximum value for the time step and ensure smooth results. SaberRT Real-Time Stimulus Sources on page 286 has more information on this subject.

8. Question: Does simulation from the SaberRT user interface run real-time?

   Answer: No. There is no reference to a real-time clock in a simulation run from the SaberRT user interface. The progress of the simulation time depends on the complexity and stiffness of the design as well as the time spent on display operations. It will run slower if you have time-consuming animations or if you choose walking waveform probes.

9. Question: Can several SaberRT models based on the same design be run concurrently with the generic C interface?

   Answer: The generic interface function includes static variables that store the simulation context of the mode from one call to the next. This context can not be shared between several models. Therefore the C function can only be used by one model at a time. It is however possible to save the model C interface with a different name for each instance and have the instances running remotely on separate computers.

10. Question: How does the sampling period affect the simulation speed?

    Answer: Decreasing the sampling period tends to decrease the simulation speed. The sampling period implicitly defines the maximum time step the simulator can take. By decreasing it, the density of time points increases. At the same time, the interprocess communication frequency increases, which also tends to slow down the simulation.

11. Question: The C interface generated by SaberRT for remote simulation does not execute properly. How can the problem be traced?

    Answer: Check that the communication between the process calling the interface and the simulation server has been established properly. The Simulation Server window in SaberRT should show that a connection has been accepted.

    The simulation server then invokes Saber which loads the design. If this does not work, you need to make sure that:

    - the design resides in the current directory. To query the current directory, type pwd in the AIM command line. Note that the current directory is not necessarily the one from which Sketch or SaberRT was launched.
    - no other Saber or SaberRT sessions are currently using the design; this is especially important on Windows.
    - the reference designators assigned to the SaberRT stimuli are not shared by other models in the design netlist. This would cause an error when trying to the run the DC analysis.

    If the problem persists, you can trace it by:

    a. Looking into the Saber output file (*.out).

    b. Invoking the C interface under the trace mode. To activate this mode, you need to set the Trace File option to yes in the calibration form (Edit > Calibration...) before exporting the interface. In this mode, the function calls to the SMMI interface are logged in a C file (smmitrace_***.c). Be aware that this mode can slow down the simulation and should only be used for debugging purposes.

11

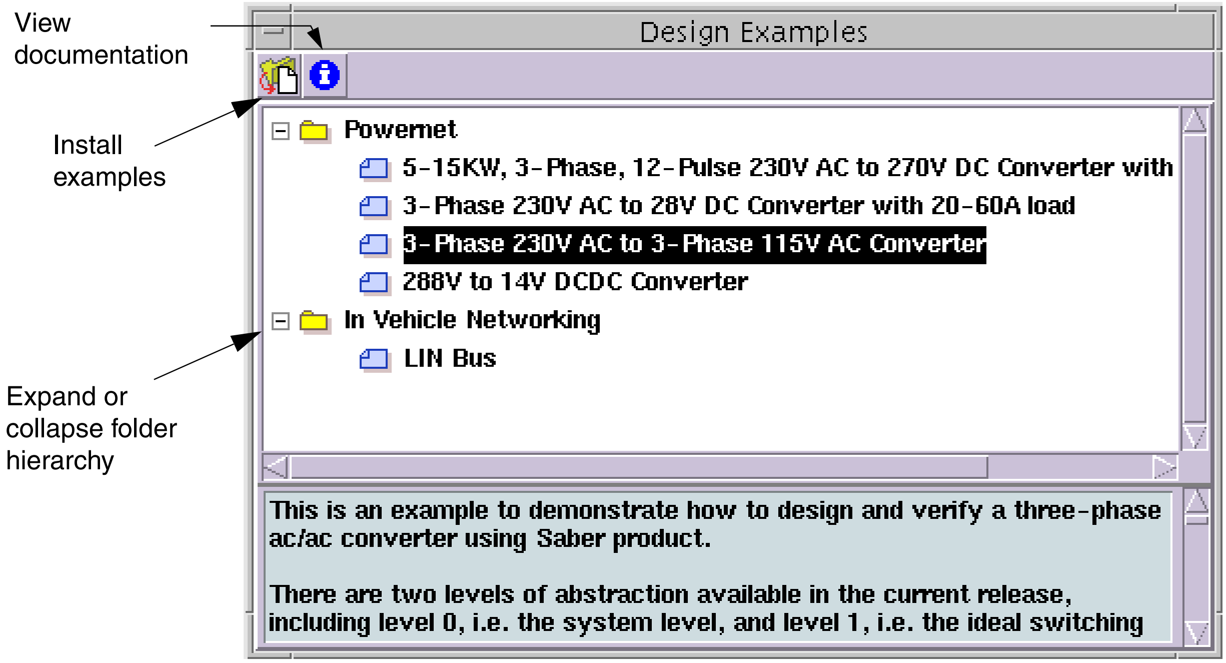
Viewing Design Examples

From Saber Harness and Saber Sketch, there is now an easy way to access design examples.

Using the Design Examples Browser

To open the Design Examples browser:

- From the Tools menu or the Tool bar, select Design Examples.



- Powernet
  - 5-15KW, 3-Phase, 12-Pulse 230V AC to 270V DC Converter with
  - 3-Phase 230V AC to 28V DC Converter with 20-60A load
  - 3-Phase 230V AC to 3-Phase 115V AC Converter
  - 288V to 14V DCDC Converter
- In Vehicle Networking
  - LIN Bus

This is an example to demonstrate how to design and verify a three-phase ac/ac converter using Saber product.

There are two levels of abstraction available in the current release, including level 0, i.e. the system level, and level 1, i.e. the ideal switching

The upper half of the Design Example browser shows a hierarchical list of design examples organized by application.

The lower half of the Design Example browser shows a brief description of the selected example.

You can do the following tasks in the Design Example browser:

- To expand or collapse a folder hierarchy, click on the Plus or Minus icon next to a folder.
- To select an example and show its description, click on an example icon or the example text.
- To install an example, click the Install Examples button. Follow the installation instructions to install the example.
- To view the full documentation for the selected example, click the View Documentation button.

12

Using the RF Tool

This chapter explains how to use the RF Tool.

To perform special measurements and calculations when running RF analyses, CosmosScope provides the RF Tool, which you can use to select signals and apply a Point Trace measurement. You can also select a circle item and input necessary configuration information to calculate RF circles.

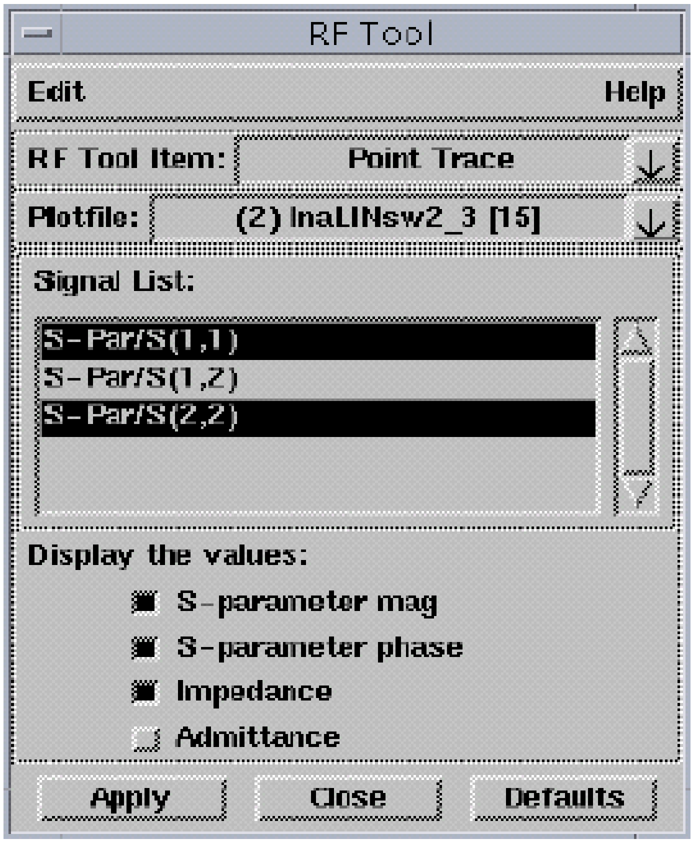
Invoking the RF Tool

To invoke the RF Tool, select **Tools > RF Tool** from the menu, or click the RF Tool icon on the Tool Bar. In the RF Tool dialog box, click the down arrow button in the RF Tool Item field. You can select the item you want to apply from the selection menu.

Point Trace Measurements

You can place Point Trace Markers for one or more signals from the same output file on Smith/Polar Charts. These markers are secured to one another—moving a marker from one chart moves the rest of the markers as well. The markers follow a trace and sequentially move from one data point to another on the same frequency value. If the signals are multi-member signals, you can make markers jump to different segments by right-clicking the mouse on the marker and selecting Next Segment from the pop-up menu that appears. Markers will then be displayed on the Smith/Polar Chart. A Point Trace Table will be displayed as a separate window to show the values of the markers. The results include the frequency range, the system impedance, the current frequency value, the signal name, magnitude, phase, impedance and admittance values for S-Parameters. In addition, results also include the segment index and sweep parameters for multi-member signals.

RF Tool - Point Trace Dialog



RF Tool Item: Point Trace

Plotfile: (2) lnaLINsw2_3 [15]

Signal List:

- S-Par/S(1,1)
- S-Par/S(1,2)
- S-Par/S(2,2)

Display the values:

- ■ S-parameter mag
- ■ S-parameter phase
- ■ Impedance
- □ Admittance

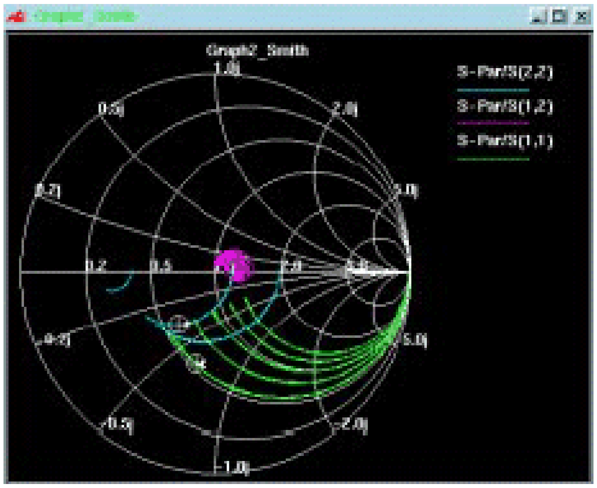
Point Trace dialog

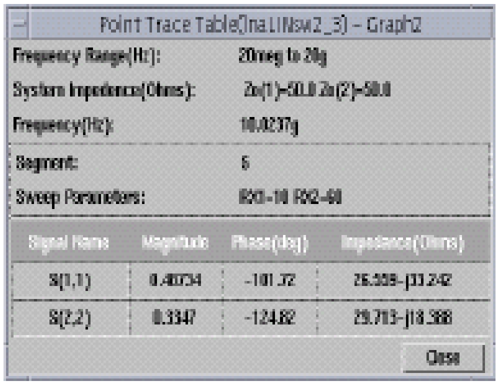
Plot File — Defines the plot file you want to use.

Signal List — Defines the signals you want to put a Point Trace on.

Display the values — Specifies the value types you want to show in the Point Trace table.

Point Trace Markers and Table




Point Trace Table(lna.LINsw2_3) - Graph2

Frequency Range(Hz): 20meg to 20g

System Impedance(Ohms): Zo(1)=50.0 Zo(2)=50.0

Frequency(Hz): 10.0237g

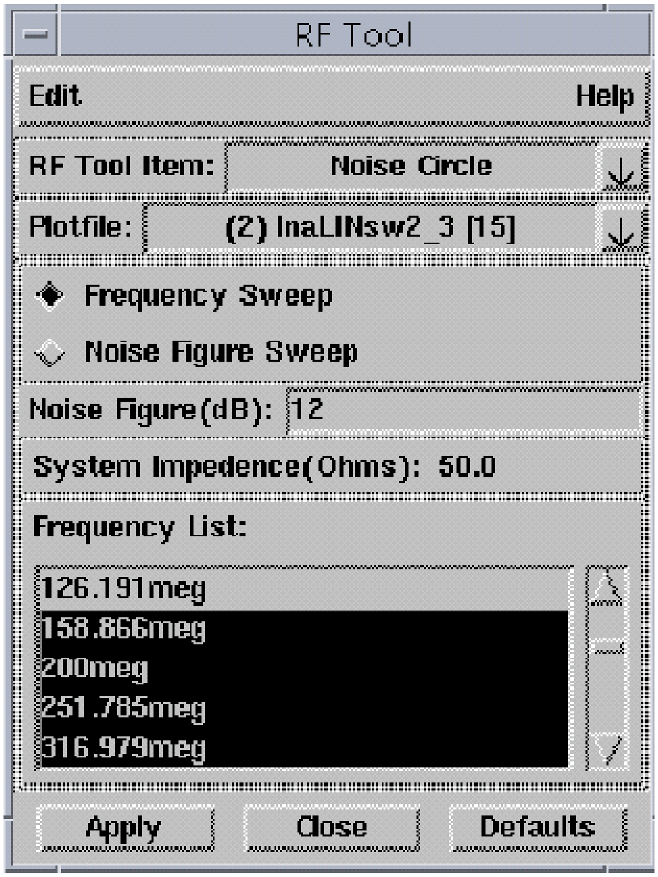
Segment: 5

Sweep Parameters: RX1=10 RX2=40

Signal Name	Magnitude	Phase(deg)	Impedance(Ohms)
S(1,1)	0.40734	-101.72	26.559-j33.242
S(2,2)	0.3347	-124.82	29.713-j18.388

Point Trace Markers and Table

Noise Circle



RF Tool Item: Noise Circle

Plotfile: (2) lnaLINsw2_3 [15]

◆ Frequency Sweep

◇ Noise Figure Sweep

Noise Figure(dB): 12

System Impedence(Ohms): 50.0

Frequency List:

126.191meg
158.866meg
200meg
251.785meg
316.979meg

Noise Circle

Plot File

Defines the plot file you want to use.

Frequency List

Defines a range of frequencies to apply to the circle. If you select Frequency Sweep as a sweep method, you can select multiple frequencies by holding the <Ctrl> key and selecting frequencies.

Frequency Sweep

Selects Frequency Sweep as sweep method for drawing the circle.

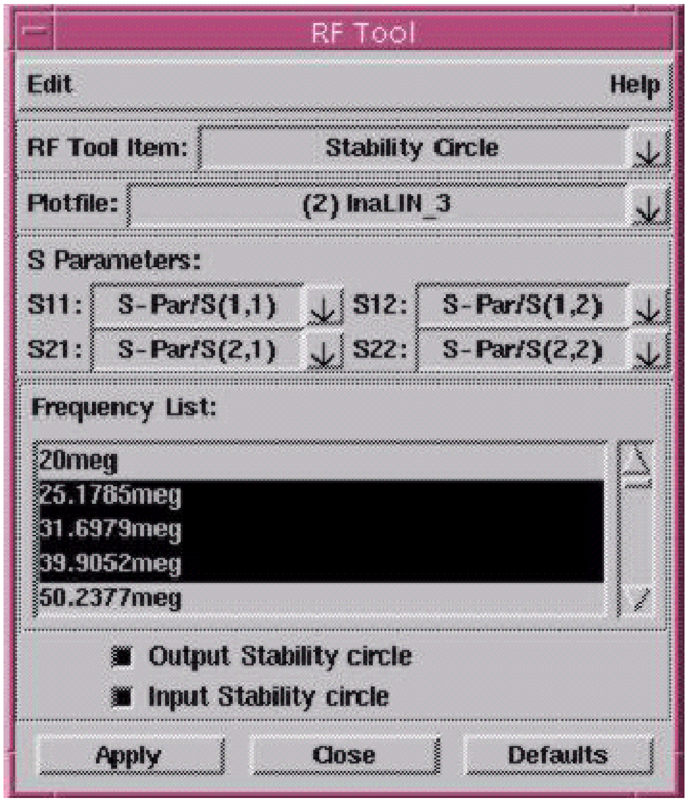
Noise Figure Sweep

Selects Noise Figure Sweep as sweep method for drawing the circle.

Noise Figure (db) Defines a value for the noise figure.

System Impedance (ohm) Defines a value for system impedance.

Stability Circle



RF Tool Item: Stability Circle

Plotfile: (2) lnaLIN_3

S Parameters:

S11: S-Par/S(1,1) S12: S-Par/S(1,2)

S21: S-Par/S(2,1) S22: S-Par/S(2,2)

Frequency List:

20meg
25.1785meg
31.6979meg
39.9052meg
50.2377meg

■ Output Stability circle

■ Input Stability circle

Stability Circle

Plot File
Defines the plot file you want to use.

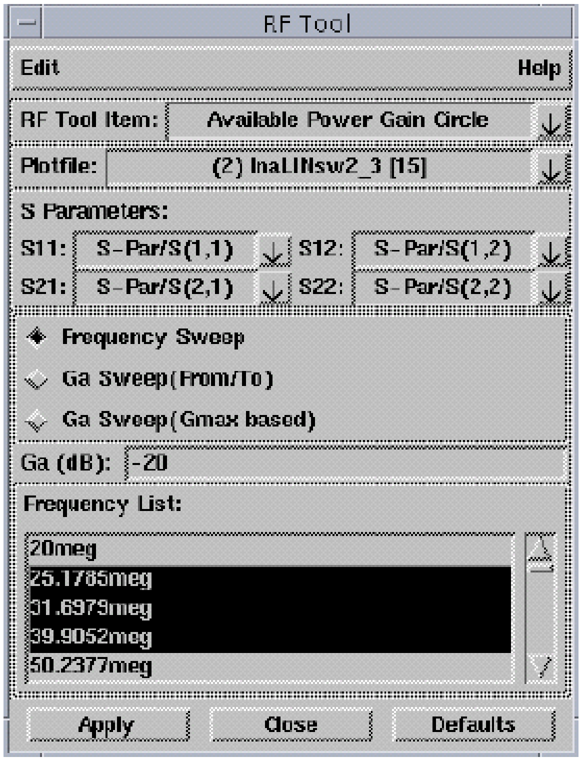
S-Parameters (S11, S12, S21, S22)
Defines the values for S-Parameters.

Frequency List
Defines a range of frequencies to apply to the circle.

Output Stability Circle
Selects the Output Stability circle.

Input Stability Circle
Selects the Input Stability circle.

Available Power Gain Circle



RF Tool Item: Available Power Gain Circle

Plotfile: (2) lnaLINsw2_3 [15]

S Parameters:

S11: S-Par/S(1,1) S12: S-Par/S(1,2)

S21: S-Par/S(2,1) S22: S-Par/S(2,2)

◆ Frequency Sweep

◇ Ga Sweep(From/To)

◇ Ga Sweep(Gmax based)

Ga (dB): -20

Frequency List:

20meg
25.1785meg
31.6979meg
39.9052meg
50.2377meg

Available Power Gain Circle

Plot File — Defines the plot file you want to use.

S-Parameters (S11, S12, S21, S22) — Defines the values for S-Parameters.

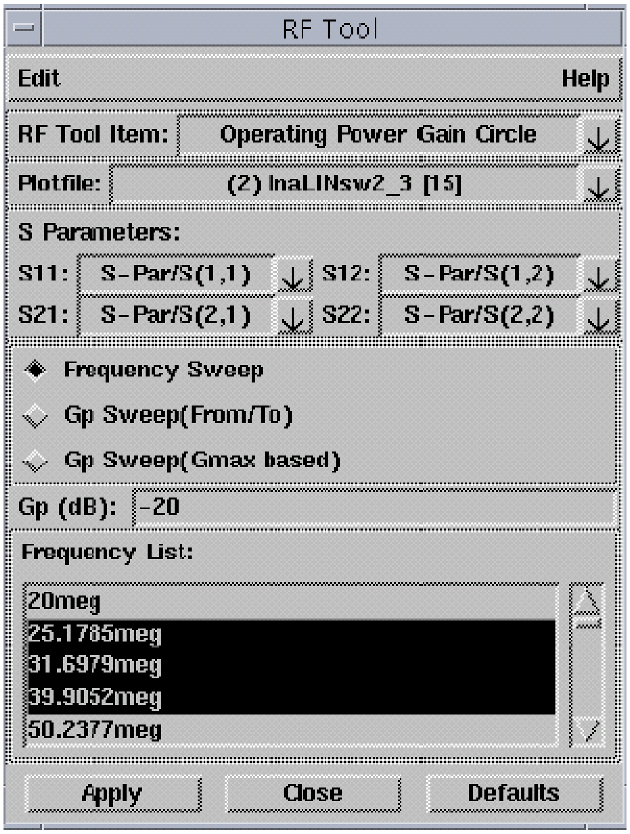
Frequency List — Defines a range of frequencies to apply to the circle.

Frequency Sweep — Selects Frequency Sweep as the sweep method.

Ga Sweep (From/To) — Selects Ga Sweep (From/To) as the sweep method.

Ga Sweep (Gmax based)	Selects Ga Sweep (Gmax based) as the sweep method.

Operating Power Gain Circle



RF Tool Item: Operating Power Gain Circle

Plotfile: (2) lnaLINsw2_3 [15]

S Parameters:

S11: S-Par/S(1,1) S12: S-Par/S(1,2)

S21: S-Par/S(2,1) S22: S-Par/S(2,2)

● Frequency Sweep

○ Gp Sweep(From/To)

○ Gp Sweep(Gmax based)

Gp (dB): -20

Frequency List:

20meg
25.1785meg
31.6979meg
39.9052meg
50.2377meg

Operating Power Gain Circle

Plot File
Defines the plot file that you want to use.

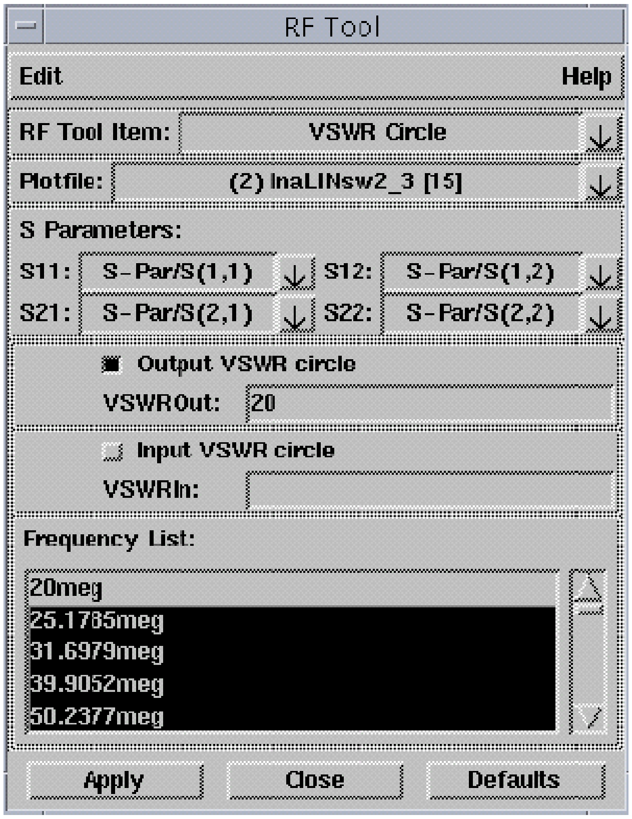
S-Parameters (S11, S12, S21, S22)
Defines the values for S-Parameters.

Frequency List
Defines a range of frequencies to apply to the circle.

Frequency Sweep
Selects Frequency Sweep as the sweep method.

Ga Sweep (From/To)	Selects Ga Sweep (From/To) as the sweep method.
Ga Sweep (Gmax based)	Selects Ga Sweep (Gmax based) as the sweep method.
Gp (db)	Defines the value of Gp.

VSWR Circle



RF Tool Item: VSWR Circle

Plotfile: (2) lnaLINsw2_3 [15]

S Parameters:

S11: S-Par/S(1,1) S12: S-Par/S(1,2)

S21: S-Par/S(2,1) S22: S-Par/S(2,2)

■ Output VSWR circle

VSWROut: 20

□ Input VSWR circle

VSWRIn:

Frequency List:

20meg
25.1785meg
31.6979meg
39.9052meg
50.2377meg

VSWR Circle

Plot File	Defines the plot file you want to use.
S-Parameters (S11, S12, S21, S22)	Defines the values for S-Parameters.
Frequency List	Defines a range of frequencies to apply to the circle.
Output VSWR Circle	Selects the output VSWR circle.
VSWROut	Defines a value for VSWROut.
Input VSWR Circle	Selects the input VSWR circle.

VSWRIn	Defines a value for VSWRIn.

The RF Tool locates the associated waveforms from the selected plot file by default. To change the default signal, pull down the list box and select one from the Signal List dialog box.

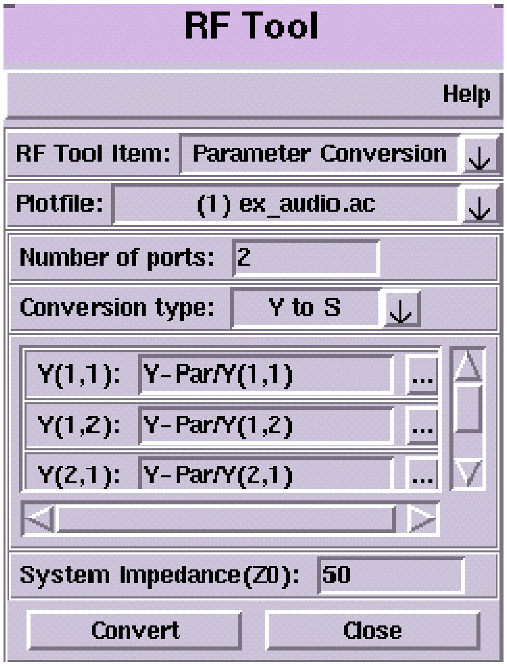
Converting Parameters

Since only S Parameters are meaningful on Smith/Polar Charts, and typically plot information is described only in terms of Y or Z parameters, the RF Tool allows for the conversion of parameters. See Conversion Equations for more information on the equations that are used for parameter conversion.

To convert parameters:

1. Select Parameter Conversion in the RF Tool Item field of the RF Tool.
2. Select the plot file from the Plotfile pulldown menu.
3. Enter the number of ports in the Number of Ports field, with a minimum of one port.
4. Choose one of the following conversion types from the menu: S to Y, S to Z, Y to S, Y to Z, Z to S, or Z to Y.
5. Set the signals for conversion.
6. Enter the system impedance (Z0) value in the System Impedance field.

The following RF Tool dialog box shows a sample parameter conversion configuration:



RF Tool Item: Parameter Conversion

Plotfile: (1) ex_audio.ac

Number of ports: 2

Conversion type: Y to S

Y(1,1): Y-Par/Y(1,1)

Y(1,2): Y-Par/Y(1,2)

Y(2,1): Y-Par/Y(2,1)

System Impedance(Z0): 50

7. Click the **Convert** button to perform the conversion.

   The conversion results are displayed in a separate window containing a list of converted signals. You can select and plot any of the signals in the list. You can also select and save any of the signals to a *.ai_pl plot file.

Conversion Equations

The following equations are used in the conversion:

```
[Y0]=[Z0]**(-1)

[Z]={{[1]+[S]}*{[1]-[S]}**(-1)}*[Z0]
[Y]={{[1]-[S]}*{[1]+[S]}**(-1)}*[Y0]
```

where

S are the S parameters
Y are the Y parameters
Z are the Z parameters
Z0 is the system impedance.

13

Using the Report Tool

This chapter describes the Saber Report Tool.

This chapter includes the following sections:

Using the Report Tool

From the Report tool, you can view/edit ASCII files and use Saber commands to generate analysis reports such as stress or sensitivity reports.

The Report icon is located in the Tool bar at the bottom of the work surface.

The following list shows a typical scenario for using the Report to edit a file generated by Saber:

1. To open or close the tool, single click on the icon with the left mouse button.
2. To open a file, click on File > Open, or run a new Saber simulator report by clicking on File > New.
3. Split the report window by clicking on the Split Window icon .

   The top window is for reference only and cannot be edited.

The upper window is an exact copy of the lower window. As edits are performed in the lower window, the upper window is updated to reflect the changes.

4. Add comments, change values, perform a global search for text strings, and rearrange text by cutting and pasting.
5. Save as a new text file with the File > Save As item.

Using the File Menu in the Report Tool

The File menu allows you to generate new Saber Guide reports, open existing reports or files, save reports or files, close the current report, and close the report tool window.

Menu Item	Description
New	Generates a new Saber Guide report. These reports are the same as reports generated with the Results pulldown menu. Saber must be running with an active design. For information on Results menu choices, see: Initial Point Results Print Results Stress Report Sensitivity Report Small Signal Parameters Pole-Zero Report
Open	Opens existing report files. The Open dialog box will be displayed with the List Files of Type field set for Reports (.rpt). More than one report can be opened at one time.
Save	Saves the current report with the current name and file location.
Save As	Saves the current report with a name and file location of your choosing.
Close	Closes the current report. A message will be displayed asking if you want to save the report if changes have been made and not saved previously.

Menu Item	Description
Close Window	Closes the Report tool. A message will be displayed asking if you want to save reports.

Using the Edit Menu in the Report Tool

The Edit pulldown menu allows you to perform word-processing operations on displayed text. You can undo and redo operations, cut/copy/paste text, select sections of text, and perform global search and replace operations.

Menu Item	Description
Undo	Cancels the previous operation that was performed. Can be performed multiple times.
Redo	Cancels the previous Undo operation. Can be performed multiple times.
Cut	Cuts selected text from the lower window and saves it to a clipboard.
Copy	Copies selected text the from the lower window to a clipboard.
Paste	Pastes text from the clipboard into the lower window.
Select All	Selects all items in text flow.
Select Line	Selects the line where the insertion cursor is located.
Deselect All	Deselects all items.
Find/Change	Opens the AimSearch - Find/Change dialog box.

Using the Format Menu in the Report Tool

The Format pulldown menu allows you to perform word-processing operations on displayed text. You can change the margins and add or delete text before or after selected lines.

Menu Item	Description
Adjust Margins	Allows you to change the margins on selected lines.
Add prefix/suffix	Allows you to globally add text before or after selected lines.
Delete prefix/ suffix	Allows you to globally delete text before or after selected lines.

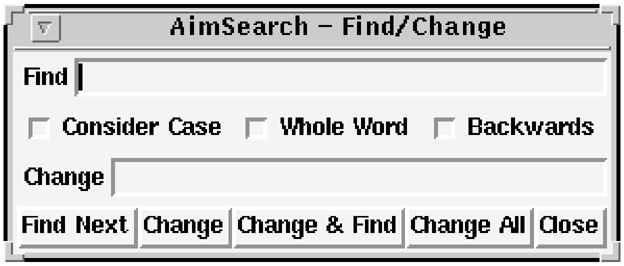
Using the Window Menu in the Report Tool

The Window menu allows you to manipulate the windows containing currently open report. A list of current reports is displayed at the bottom of the Window menu. To view one of the reports in the list, click on the button to the left of the report name.

Menu Item	Description
Split Window	Splits the report window into a top and bottom view. The top view cannot be edited and is only for reference. The bottom view has complete editing capabilities.
Undo Split Window	Closes the top window.
Next File	Rotates to the next report file when multiple files are loaded.

The Find/Change Dialog Box

This dialog box allows you to search for text strings or parts of text strings and to replace them with new text if desired.



The Find field is where you type the text string you wish to locate. A default text search starts at the beginning of the report. Case is not considered, and the text string may be a fragment of a larger text string. A search is started by pressing the Find Next button.

The Consider Case button narrows the search by looking for text that matches the case of the text string typed in the Find field.

The Whole Word button narrows the search by looking for the entire text string. If the text string is a fragment of another text string, it will not be considered.

The Backwards button searches backwards from the insertion cursor position.

The Change field is where you type the text you wish to insert.

The Find Next button highlights the next occurrence of the text string typed into the Find field. No changes are made.

The Change button replaces highlighted text with the text string typed in the Change field.

The Change and Find button highlights the next occurrence of the text string typed in the Find field. If you wish to change the highlighted text, press the Change and Find button again. The highlighted text will be replaced with the text string typed in the Change field, and the next occurrence of the text string typed in the Find field will be highlighted.

The Change All button changes all occurrences of the text string typed in the Find field to the text string typed in the Change field.

The Close button closes the Find/Change dialog box.

14

Using the CosmosScope MATLAB Interface

This chapter explains how to use the MATLAB Interface in CosmosScope.

MATLAB Interface Tool

The CosmosScope Analysis Interface to MATLAB opens a transcript window that allows access to MATLAB software application and enables data transfers between Saber applications and MATLAB applications.

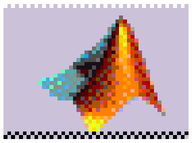
Text, waveforms, plot files, and Vector/Matrix/Arrays can be selected from other sources and pasted directly into the MATLAB window.

You can also write AIM language scripts to operate MATLAB through CosmosScope. The Macro Recorder tool can be used to facilitate developing these scripts. For more information on writing scripts refer to CosmosScope MATLAB-specific AIM commands.

Users should have some experience with MATLAB and the Saber Simulator.

Accessing the MATLAB Interface Tool

The MATLAB Interface tool icon is located in the CosmosScope Tool Bar.

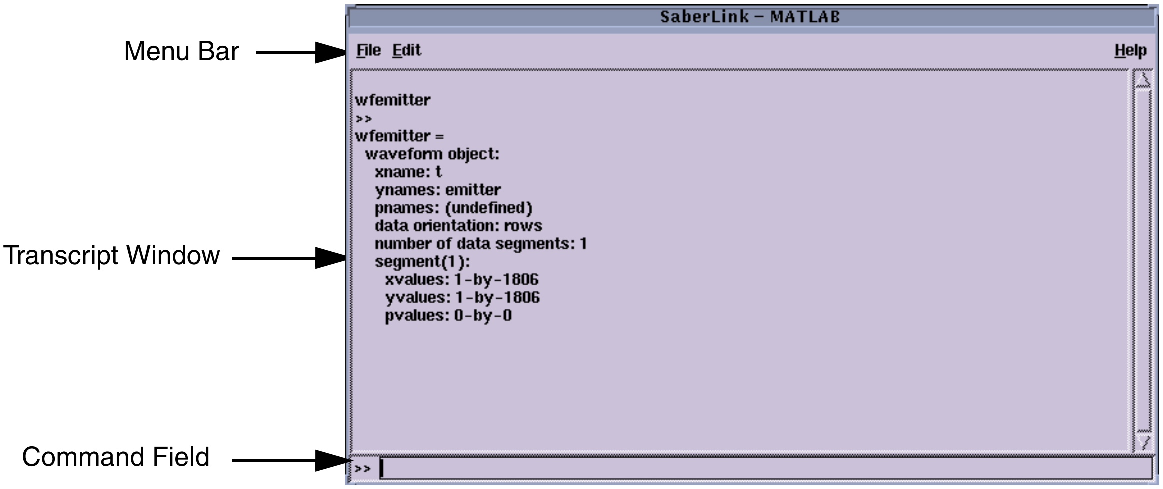


To open or close the MATLAB Interface tool, click the MATLAB Interface icon.

MATLAB Interface Window Description

The MATLAB Interface window allows you to interactively enter MATLAB commands in the Command field and transfer data between MATLAB and Saber applications or CosmosScope.

The scrollable transcript window displays MATLAB Interface commands and responses.



```
wfemitter
>>
wfemitter =
  waveform object:
    xname: t
    ynames: emitter
    pnames: (undefined)
    data orientation: rows
    number of data segments: 1
    segment(1):
      xvalues: 1-by-1806
      yvalues: 1-by-1806
      pvalues: 0-by-0
```

MATLAB Interface Menus

File > Save
Saves the contents of the transcript window as an ASCII file. The file is saved under the current file name (if any).

File > Save as
Saves the contents of the transcript window as an ASCII file. The file is saved under a file name of your choice.

File > Close Window
Closes the current CosmosScope window without closing the current MATLAB Interface session.

File > Close Session
Closes the current MATLAB Interface window and MATLAB Interface session.

Edit > Cut
Removes selected text and moves it to the clipboard. Cut acts like a copy command in the transcript window.

Edit > Copy
Copies selected text in the transcript window to the clipboard.

Edit > Paste
Pastes the contents of the clipboard into the transcript window.

Edit > Clear
Clears the contents of the transcript window.

MATLAB Interface Fields and Lists

The command field is where MATLAB Interface commands are entered.

To execute a MATLAB Interface command, type the command in the Command field and press the **Return** key.

There are several keyboard shortcuts which can be used to facilitate command entry. See MATLAB Interface Keyboard Shortcuts for a list of these shortcuts.

MATLAB Interface Keyboard Shortcuts

The following MATLAB Interface keyboard shortcuts are available:

Keyboard Key	Operation
Control + u	Clears the command line field.
Control + a	Moves the cursor to the beginning of a command line string.
Control + e	Moves the cursor to the end of a command line string.
Left Arrow	Moves the cursor to the left one character at a time.
Right Arrow	Moves the cursor to the right one character at a time.
Up Arrow	Displays the previous command in a succession of commands.
Down Arrow	Displays the next command in a succession of commands.

MATLAB Interface Data Transfer

MATLAB Interface data translation between matrix tools and Saber applications or CosmosScope can be accomplished using the mouse or the AIM command language.

Using the mouse selection method to transfer data is preferred when using the CosmosScope MATLAB Interface window. Using the AIM command method allows you to develop scripts to control both the CosmosScope MATLAB Interface tool and Saber applications with a single command stream. The AIM command method requires knowledge of the AIM command language.

To transfer data, select and copy one or more objects in the source window and paste them into the destination window. The type of translation is determined by the source and destination windows, as well as the type of data you are transferring. For MATLAB Interface translation purposes, window objects have the following corresponding object types:

Window	Object Type
MATLAB Interface tool	MATLAB Interface name

Window	Object Type
CosmosScope Graph	waveform
Guide Transcript	Plot File path name
AIM Command Line	Vector/Matrix/Array (VMA)
Other windows	ASCII string

Both waveforms and matrices in a session in the tool window are converted to MATLAB Interface waveforms when pasted into CosmosScope. When the same selections are pasted into the Saber Guide transcript window, a plot file is created, and the plot file name is displayed in the Saber Guide transcript window. If pasted into an AIM Command Line window, MATLAB Interface waveforms are translated into AIM waveforms and matrices are translated into VMAs.

Similarly, a waveform selected in a CosmosScope graph and pasted into the MATLAB Interface tool window results in the creation of a waveform object containing an independent and dependent variable. A waveform selected in an AIM Command Line window and pasted into the MATLAB Interface tool is translated into a MATLAB Interface waveform; a VMA is translated into a MATLAB Interface matrix.

In the AIM language, vectors are different from matrices. In MATLAB, vectors are interpreted as matrices. Because of this difference, row and column vectors are indistinguishable in AIM. As a result, when a vector is transferred from the AIM Command Line window to the MATLAB Interface window, it is treated as a row-vector for the purposes of the transfer.

Transferring from Saber Applications to MATLAB

Transferring using mouse buttons:

1. Select the item you wish to paste in the source window using the left mouse button.
   a. In CosmosScope graph windows, place the mouse cursor over the signal name and single click the left mouse button.
   b. From a Guide transcript window, press and hold the left mouse button and drag the mouse cursor over the ASCII string representing the name of the plot file you are transferring.

c. In the AIM Command Line window select the VMA or wf handle, or the variable containing the handle.

d. In the Signal Manager and the Calculator press and hold the left mouse button and drag the mouse cursor over the ASCII string representing the name of the plot file you are transferring.

2. Place the mouse cursor in the MATLAB Interface command field, and single click the middle mouse button.

To transfer using the clipboard:

1. Select the item as in the previous method.
2. In the Saber application, choose **Edit > Cut** or **Edit > Copy**.
3. In the MATLAB Interface tool, choose **Edit > Paste**.

There are a number of MATLAB Commands that allow you to manipulate your data such as the wfdata command.

Transferring from MATLAB to Saber Applications

Transferring using mouse buttons

1. Select the item you wish to paste in either the transcript window or the command field using the left mouse button.
2. Press and hold the left mouse button, and drag the mouse cursor over the ASCII string representing the name of the item (waveform or matrix) you are transferring.
3. To complete data transfer, place the mouse cursor in the CosmosScope graph window, the Guide transcript window, or the AIM Command Line window, and single click the middle mouse button.

Note:

> You cannot transfer from MATLAB to either the Calculator or the Signal Manager.

Transferring using the clipboard.

1. Select the item as in the previous method.
2. In the MATLAB Interface tool, choose **Edit > Cut** or **Edit > Copy**. Note that Cut behaves like Copy in the transcript window.

3. In the Saber applications or CosmosScope, choose **Edit >Paste**.

CosmosScope Window

When transferring data to a CosmosScope graph window, the data is translated into an AIM waveform.

You can transfer matrices or MATLAB Interface waveforms. Note that when matrices are transferred, they first get converted into MATLAB Interface waveforms.

Once data has been transferred into AIM waveform format, all Saber tools, including the Calculator, are available for use.

Guide Transcript Window

When transferring data to the Guide transcript window the data is translated into pf (plot file) format.

AIM Command Line Window

When transferring data to the AIM Command Line window, AIM waveforms are translated into MATLAB Interface waveforms, and matrices are translated into VMA (Vector/Matrix/Array) format.

CosmosScope AIM Commands

The AIM Command Line window can be used to directly interface with MATLAB. You can also write AIM scripts to automate procedures.

The Macro Recorder tool can be helpful in writing AIM scripts. Every significant operation performed in the Saber applications or CosmosScope is saved in the scope.log file in the AIM scripting language. The Macro Recorder records the AIM language script in this file, and allows you to edit the script.

To execute a script:

- Use the Command field in the AIM Command Line tool to type source pathname/filename

Several AIM commands are available to facilitate MATLAB Interface usage and allow you to control MATLAB Interface sessions, to transfer data between

MATLAB and the Saber Simulator, and to perform MATLAB operations from an AIM command line.

AIM Command	Description
matlab open session_name	Opens a MATLAB session.
matlab close session_name	Closes a MATLAB session.
session_name get source_name destination_name	Transfers data from a MATLAB session.
session_name put source_name destination_name	Transfers data to a MATLAB session.
session_name query exists|dtype|size|vtype source_name	Displays information on MATLAB data.
session_name eval {command_string}	Performs MATLAB operations.
MtiTrans:mti2var session_name MATLAB_variable	Converts a MATLAB variable to an AIM VMA.
MtiTrans:mti2wf session_name MATLAB_variable	Converts MATLAB waveform object to a CosmosScope waveform.
MtiTrans:mti2pf session_name MATLAB_variable plotfile_name	Converts MATLAB waveform object to a Saber Simulator plot file.
MtiTrans:var2mti session_name AIM_variable MATLAB _variable	Converts an AIM variable to a MATLAB variable.
MtiTrans:wf2mti session_name waveform MATLAB_variable	Converts a CosmosScope waveform to a MATLAB waveform object.
MtiTrans:pf2mti session_name plotfile_name MATLAB_variable	Converts a Saber Simulator plot file to a MATLAB waveform object.

AIM Overview

AIM is a super-set of the Tcl/Tk scripting language developed by John K. Ousterhout. Detailed information on the AIM scripting language is beyond the scope of this manual. Information about Tcl/Tk is available in the book Practical Programming in Tcl and Tk, second edition by Brent B. Welch. Refer to the AIM documentation for additional details and information.

MATLAB Interface Waveform Commands

The following table summarizes the MATLAB Interface waveform commands:

Command	Description
waveform	Creates a waveform object using the given arguments, or displays information on a waveform object.
wfdata	Sets or gets the data for a waveform object.
wfdatatype	Returns the datatype of the dependent variable value in a waveform object.
wfnames	Sets or gets the names for an existing waveform object.
wfnpars	Returns the number of parameters defined for the waveform object.
wfnsegs	Returns the number of data segments in a waveform object.
wfparsizes	Gets or sets parameter sizes for a waveform object.
wfparvalues	Returns the parameter values for a given parameter in the waveform object.

waveform

Using CosmosScope, waveforms are represented as structured data types. For MATLAB, CosmosScope provides a “waveform” class.

The waveform command creates a waveform object using the given arguments. The dependent variable names and parameter names are specified as cell character arrays. If the values are specified without names, the waveform object is created with default names. These names may subsequently be changed with the wfnames command. The independent, dependent, and variable values are all numeric arrays.

The waveform command has the following formats:

```
waveform_name=waveform()
```

With no arguments, the waveform command creates an empty waveform object:

```
waveform_name=waveform(waveform_name)
```

If a waveform object is specified as input, the waveform command returns a copy of that object:

```
waveform_name=waveform(independent _variable_name,
  dependent_variable_names)
waveform_name=waveform(independent _variable_name,
  dependent_variable_names, parameter_names)

waveform_name=waveform(dependent _variable_name)

waveform_name=waveform(independent _variable_name,
  dependent_variable_names, independent_variable_values,
  dependent_variable_values)

waveform_name=waveform(independent_variable_values,
  dependent_variable_values)
```

waveform_name	The name of a MATLAB Interface waveform object.
independent _variable_name	The name of the independent variable component of a MATLAB Interface waveform object.

dependent_variable_names
The names of the dependent variable components of a MATLAB Interface waveform object.

parameter_names
The names of the parameters of a MATLAB Interface waveform object.

independent_variable_values
The values of the independent variable component of a MATLAB Interface waveform object.

dependent_variable_values
The values of the dependent variable components of a MATLAB Interface waveform object.

wfdata

Sets or gets the data for an existing waveform object. The data includes independent variable values, dependent variable values, and, optionally, parameter values which correspond to a given segment.

In the first two forms (set forms), the data values are appended to the existing waveform data as a new data segment. In the other forms (get forms), The segment selector may be a scalar value or a vector of valid segment values. Use of the keyword all returns data for all segments. If data for more than one segment is returned, the size of the data will be that of the largest segment selected. If no segment selector is specified, the first segment (1) is returned.

```
waveform_name=wfdata (waveform_name,
independent_variable_values, dependent_variable_values)
```

```
waveform_name=wfdata (waveform_name,
independent_variable_values,
dependent_variable_values,parameter_values)
```

```
[independent_variable_values,
dependent_variable_values,parameter_values] =
wfdata(waveform_name)
```

```
[independent_variable_values,
dependent_variable_values,parameter_values] =
wfdata(waveform_name,selector)
```

```
[independent_variable_values,
dependent_variable_values,parameter_values] =
wfdata(waveform_name,’all’)
```

waveform_name	The name of a MATLAB Interface waveform object.
independent_variable_value s	The values of the independent variable component of a MATLAB Interface waveform object.
dependent_variable_ values	The values of the dependent variable components of a MATLAB Interface waveform object.
parameter_values	The values of the parameters of a MATLAB Interface waveform object.
selector	The default value is 1. The value may be: a scalar—the segment index of the segment to be retrieved. a vector—the range of segment indices to be retrieved, e.g., [1 2 3] selects segment 1-3. the string all—selects all segment values

wfdatatype

Returns the datatype of the dependent variable value in a waveform object.

```
data_type=wfdatatype(waveform_name)
```

data_type	The name of the MATLAB Interface waveform object data type.
waveform_name	The name of a MATLAB Interface waveform object.

wfnames

Sets or gets the names for an existing waveform object. The names include an independent variable name, one or more dependent variable names, and, optionally, one or more parameter names. The dependent variable names and parameter names are specified as cell character arrays.

```
waveform_name=wfnames(waveform_name,
independent_variable_name, dependent_variable_names)
```

```
waveform_name=wfnames(waveform_name,
independent_variable_name,
dependent_variable_names,parameter_names)
```

```
[independent_variable_name, dependent_variable_names,
parameter_names] = wfnames(waveform_name)
```

waveform_name	The name of a MATLAB Interface waveform object.
independent _variable_name	The name of the independent variable component of a MATLAB Interface waveform object.
dependent_variable_ names	The names of the dependent variable components of a MATLAB Interface waveform object.
parameter_names	The names of the parameters of a MATLAB Interface waveform object.

wfnpars

Returns the number of parameters defined for the waveform object and takes the following form:

```
number_of_parameters=wfnpar(waveform_name)
```

waveform_name	The name of a MATLAB Interface waveform object.
number_of_parameters	The number of parameters present in the waveform object.

wfnsegs

Returns the number of data segments in the given waveform object and takes the following form:

```
number_of_segments=wfnsegs(waveform_name)
```

waveform_name	The name of a MATLAB Interface waveform object.
number_of_segments	number_of_segments

wfparsizes

Returns a waveform with its parameter sizes set to the values contained in parameter_size_vector. If parameter_size_vector is specified, it must be a vector of length number _of_parameters, which is the number of parameter values defined in the waveform object.

```
parameter_sizes=wfparsizes(waveform_name)
```

Returns the parameter sizes for the waveform object.

```
waveform_name=wfparsizes(waveform_name,
  parameter_size_vector)
```

parameter_sizes	The name of the parameter sizes in a MATLAB Interface waveform object.
waveform_name	The name of a MATLAB Interface waveform object.
parameter_size_vector	The number of parameter values in a MATLAB Interface waveform object.

wfparvalues

Returns the parameter values for the index_number parameter in the waveform object. The first index_number is one.

```
parameter_values=wfparvalues(waveform_name,
  index_number)
```

parameter_values	The values of the parameters of a MATLAB Interface waveform object.
waveform_name	The name of a MATLAB Interface waveform object.
index_number	The index, starting at one, of the parameter whose values you wish to display.

MATLAB Interface Command Limitations

The MATLAB Interface window interface is implemented through the use of calls to the MATLAB engine library. The mechanism for MATLAB Interface command execution involves the following steps:

- Fielding a command from user input in the Command field and transferring that command to MATLAB for execution
- Collecting the resulting output to the MATLAB Interface window in paged form based on additional user input in the Command field

Limitations in the MATLAB engine library affect the ability to process command input, and can affect the way in which some output is displayed.

For example, in the MATLAB “native” command entry window, a long list is displayed, a screen at a time. If you wish to view more information, entering the more command displays the next screen. In the MATLAB Interface window, a long list will be displayed in its entirety, whether or not the information fits in one screen. Use the scroll bars to view the information one screen at a time.

The other commands that do not behave as expected are described in the following table. This limitation does not apply to multi-line commands. These commands will behave as expected.

MATLAB Interface Command	Expected Behavior	Actual Behavior
input	Displays the provided text string, waits for input from the keyboard, and returns the value entered.	The command does not wait for the keyboard entry. It continues execution as if an empty string was entered at the keyboard.
more	Controls the paged output for the MATLAB Interface command window. Pressing the Return key advances to the next line or pressing the Space bar advances to the next screen	The command does not wait for input from the keyboard and displays the entire contents at once.
pause	Causes M-files to stop and wait for the user to press any key before continuing.	The pause(n) behaves as expected. Otherwise the command does not pause.

15

Using Testify

This chapter describes how to use the Testify with the Saber Simulator.

Introduction

Testify allows you to use the Saber Simulator to develop and evaluate tests used for detecting fault conditions on a circuit board. Generally implemented on Automatic Test Equipment (ATE), these tests are designed to detect faults on the Unit Under Test (UUT) by comparing measured quantities against an expected range of performance.

For example, if a measured output voltage is outside of its specified tolerance window, this test, in combination with other tests, can be used to identify a particular fault condition, such as a component open or short, a pin-to-pin short, a pin-to-supply short, or a pin-to-ground short.

Testify helps you develop these tests by taking a schematic of an existing design and generating a netlist containing fault models of the parts in the design. You set up tests likely to reveal faults in the design, select the faults you would like to evaluate from a list of all possible faults, and then let Testify run the fault simulations and create a report of test results.

Determining Fault Conditions

Testify allows you to use the Saber Simulator to develop and evaluate tests used for detecting fault conditions on equipment such as circuit boards. Testify inserts each fault, one at a time, and runs tests of your own design to measure out-of-specification performance.

- Testify Process Steps
- Using the PinFault Editor

- Using Testify with Hierarchy
- Testify Fault Wrappers

The Getting Started with Testify on page 362tutorial uses an example circuit to demonstrate how Testify can be used to determine the fault conditions of a simple electrical circuit.

Testify Process Steps

To determine the fault conditions of your design using Testify, use the following process:

1. Using the Netlister with Testify
2. Displaying the Testify Form
3. Setting Up the Tests
4. Determining Nominal Operation and Operational Limits
5. Specifying the Faults
6. Running the Fault Simulations
7. Displaying the Testify Test Results
8. Debugging Fault Runs that Don’t Converge

Using the Netlister with Testify

Testify requires a special netlist in order to insert faults into your design. This netlist contains fault models of the parts whose failures are to be simulated by the test suite. A fault model is a standard MAST template that has a special section of code, called a fault wrapper, providing the details of a part’s fault behavior. These fault models are inserted by the netlister when you netlist your design; you do not have to place special models in your schematic. Fault models appear in the netlist as the standard part name appended with _f. For example, the c capacitor model would become the c_f fault model.

The two netlister settings you need to set are found on the Saber/Netlister Settings form, on the Advanced sub-tab of the Netlister tab:

- Fault simulation mode instructs the netlister to insert fault models when the Yes button is depressed. The No button allows the netlister to produce a standard netlist.
- Generate fault wrappers provides three options for inserting fault wrappers into your design's models:
    - Yes—fault wrappers are generated for all of the required parts the first time you netlist.
    - As Needed—the netlister will generate fault wrappers only for those parts in the design that do not have fault models either pointed to by SABER_DATA_PATH or in the current directory. This feature allows you to create your own library of custom fault models if you need to simulate faults differently than the generic fault wrappers allow.
    - No—never create fault wrappers.

Displaying the Testify Form

Once you have created a netlist of your design that contains fault models, you are ready to use Testify to determine fault conditions.

You invoke the Testify form

either by clicking the Testify icon on the Tool Bar or by selecting Tools > Testify from the Pulldown Menu Bar.

Setting Up the Tests

To set up the tests:

1. Click the Test Setup tab

   The Test Setup tab is where you define the tests performed on your circuit after each fault has been applied. You add the tests, one at a time, to the Test List column. The Saber commands making up the highlighted test in the Test List are displayed in the Selected Test column.

2. Add a test

   Each test is created by using the Experiment form, displayed either by clicking the Add button or by selecting the Edit > Add Test menu choice.

3. Specify the analyses

You can add any analysis listed in the Analysis pulldown menu to the test. Once you select an analysis, Saber Guide adds a button, denoted by the command name, to the Experiment form. You can specify the arguments of the analysis either by clicking on the analysis button or by selecting the Edit item from the arrow pulldown menu.

Because adding faults usually affects the operating point of the design, the first analysis placed in the test is typically a DC analysis. Placing this analysis first allows the remaining analyses to use an accurate operating point, given the current fault configuration of the circuit.

4. Optionally, add Vary or Monte Carlo loops

   Using the Loop menu item in the Experiment dialog box, you can add Vary and Monte Carlo loops to the test. Note that you cannot nest a Monte Carlo loop within another Monte Carlo loop.

5. Add post-processing functions

   The final addition that you make to the Experiment dialog box is to add any post-processing functions. Saber executes these functions after it completes the final command. Post-processing functions are listed under the PostProcessing pulldown menu. The process of adding and editing post-processing functions to the test is identical to the process of adding analyses.

6. Verify the test

   Before you execute the Experiment dialog box, you should verify the order of the analyses and the post-processing commands.

   The following functions, accessible from the arrow pulldown menus, allow you to change the order of commands within this form:

   - Edit displays the corresponding form for editing.
   - Delete removes the command from the test.
   - Move Up moves the command up one level in the test.
   - Move Down moves the command down one level in the test.

   After you verify the structure of the Experiment dialog box, you can click the Accept button to place this test in the Test List column of the FMEA/Testify window.

7. Optionally, add, edit, or delete tests from the Test List

   The Add, Edit, and Delete buttons, as well as the Edit pulldown menu, allow you to modify the Test List.

8. Optionally, validate the test suite

   You can validate the functionality of all of the tests listed in the Test List by selecting the Run > Validate menu choice and confirming the results in the Saber Guide Transcript window.

   If you exercise this option, the nominal calculated values of each test will automatically be placed in the Nominal column on the Test Ranges tab.

Determining Nominal Operation and Operational Limits

To determine nominal operation and operational limits:

1. Click the Test Ranges tab

   The Test Ranges tab is where you determine your design's normal operating region.

   - A filled-in check box beside the test name will include that test in the test suite.
   - The Name column lists the tests included in the suite, in order.
   - The Lower Limit column defines the lower operational limits of each test.
   - The Upper Limit column defines the upper operational limits of each test.
   - The Nominal column displays the value returned by the test when the circuit elements are all at their nominal values.

2. Calculate the nominal values of the tests

   Because the circuit's operational limits can be defined as a function of nominal operation, nominal values for each test measurement must be calculated. You do this by clicking the Nominal button.

3. Calculate the test limits

   A circuit has a range of acceptable operation that is usually specified by its design specification. If a fault does not cause the circuit to go beyond this acceptable range, the fault will go undetected. Because of this, the operational limits of a test are a crucial factor in determining how many faults are detected by a given test.

   The test limits are calculated by one of the two methods defined in the Limits Calculated From field:

   - Monte Carlo

The default method of calculating test limits is to perform a Monte Carlo analysis of fifteen runs for each test. You can change the parameters of the Monte Carlo analysis, including the number of runs, by clicking the Edit... button and filling out the Monte Carlo Analysis form following the procedures of "Executing the Monte Carlo Analysis".

To execute the Monte Carlo analysis click the Limits button.

- % Nominal

  The arrow pulldown buttons next to the Lower and Upper fields allow two choices for entering limits: Specify Each and Apply to All.

  Specify Each lets you set the percentage of nominal for each test by typing the values directly into the Lower Limit or Upper Limit fields.

  Apply to All lets you set the percentage of nominal for all of the tests by typing the value in either the Lower or Upper fields at the bottom of the form.

Specifying the Faults

To specify the faults:

1. Click the Fault List tab

   The Fault List tab is where you define what faults will be tested by the test suite.

   You compile the possible faults in the Possible Faults listbox; of those, you place the ones you want tested into the Chosen Faults listbox.

2. Compile the possible faults

   Clicking the Get Fault List button will place all of the faults that can be exercised in your circuit—such as open, short, pin-to-pin short, pin-to-supply short, and pin-to-ground short—into the Possible Faults listbox, making them available for inclusion in the test suite.

   The Get Parameters button allows you to insert parametric faults—relative changes such as a large value deviation, rather than absolute changes like short- or open-circuits—into the Possible Faults listbox.

   To insert parametric faults:

   a. Click the Get Parameters button to invoke the Parameter Fault List dialog box.

b. Select Parameters > Browse... to instruct Saber to compile a list of all of the parameters in the circuit and display them in the Set Parameter List dialog box

c. Highlight the parameters of interest and place them in the Parameter Fault List by clicking OK.

d. Type the failure value of the parameter in the Value column.

e. Once you have filled out the Parameter Fault List, click OK to place the parametric fault in the Testify Possible Faults listbox.

3. Specify the faults

   You specify the faults you want in the test suite by placing them in the Chosen Faults listbox.

   a. Highlight the items in the Possible Faults list either with the mouse cursor or by selecting the Edit > Select All > Possible Faults menu choice.

   b. Filter the possible faults by part type by setting the Part Filter to one of the following:

Option	Description
All Parts in the design	
Toplevel Parts	Relevant only for hierarchical designs
Passive Parts	Use for resistors, capacitors and inductors
Active Parts	Use for integrated circuits, transistors and other semiconductor parts
Specified for filling in part types	For example r* for all of the resistors in the design

   c. Filter the possible faults by fault type by setting the Fault Filter to one of the following:

Option	Description
All Faults in the design	
Analog Faults	Use for shorts and opens of all varieties

Option	Description
Digital Faults	Use for digital outputs stuck high or low
All Pin Opens	
All Pin Shorts	
Pin-Pin Shorts	
Pin-Supply Shorts	
Stuck Faults	Use for digital outputs stuck high or low.
Specified	Use for filling in fault types not covered in the above choices

d. Click the --> button to move the faults into the Chosen Faults listbox.

Running the Fault Simulations

Testify creates the fault/test result matrix on the Results tab.

1. Click the Results tab
2. Specify simulator preferences

   You are able to control several of the simulator's options from the Simulator tab of the FMEA/Testify Preferences form (Edit > Preferences...):

   - Limit Time will allow simulation time to be limited if the Yes button is depressed.
   - Fault Time Multiplier is multiplied by the nominal simulation time to determine the simulation time limit. This value is important only if Limit Time is set to Yes.

     For example, if a nominal transient analysis takes 12 seconds, and Fault Time Multiplier = 5 and Limit Time is Yes, then a simulation will be interrupted when it reaches 5x12 = 60 seconds.
   - Maximum Time sets the simulation time limit in seconds.
   - Open Resistance is the finite resistance that represents an open circuit.
   - Short Resistance is the finite resistance that represents a short circuit.

   "Debugging Fault Runs that Don't Converge" explains how to use the Open Resistance and Short Resistance values in debugging.

3. Specify how the results are displayed on the Results form

   You can control the color of the test data on the Results tab from the FMEA/ Testify Preferences form (Edit > Preferences...) for each of the three result conditions, Below Lower Limit, Between Limits, Above Upper Limit:

   a. Click the Display tab.
   b. Click the rainbow color band to invoke the Color Editor.
   c. Click on the color you want.
   d. Click Apply.

   The editable Text fields contain the words used to describe each test result condition in reports generated by the Report Tool; the Text fields do not affect display on the Testify Results tab.

   Disable Probes During FMEA/Testify Run prevents multiple simulations from updating the probe windows each time a test is run. The default is that Probes are disabled. This feature affects Saber Sketch only, not the Frameways.

4. Optionally, disable Smart Simulation

   Smart simulation is a time saving feature and is selected by default. Smart Simulation prevents Testify from running redundant DC simulations for each test run. You can disable this feature by selecting Edit > Smart Simulation to un-check the box.

5. Run the simulations

   Clicking the Run button begins the test suite. You can pause the test runs by clicking the Pause button.

Displaying the Testify Test Results

Once you have run the fault simulations, you can display the results in a text file using the Report Tool.

To display Testify test results:

1. Select the Edit > Preferences... menu choice.

   You are able to select from a set of possible parameters to include on the report.

2. Select the Reports tab.

Do any of the following:

- The Format listbox allows you to select either of the predefined report formats, Short or Verbose, as defined by the check boxes in the Include section. These check boxes can also be turned on or off individually.
- You can create a “spreadsheet style” report using tabs and ready for export into Excel, by selecting the Spreadsheet Style box.
- Automatic Report Name, if the check box is filled in, will name the report either Short_Report or Verbose_Report, depending upon the Format selected. If the check box is not filled in, you will be queried for the report name.

3. Click Apply if you have made changes you want to keep.
4. Click Close.
5. To generate the report and open the Report Tool, click the Report button at the bottom of the Results tab.
6. To view the Report form, see the document for the Report Tool for instructions for managing the report text file.

Debugging Fault Runs that Don’t Converge

Sometimes the insertion of short-circuit and open-circuit faults may cause nonconvergence. The short-circuit default value is 100mW ($1x10^{-4}$), and the open-circuit default value is 100MW ($1x10^{8}$).

Fault Resistances If you have convergence problems, you can try changing the values of the fault resistances. The value of the fault resistance that allows the simulation to converge—while still giving meaningful fault detection results—depends on the circuit elements surrounding the fault, and will therefore vary from circuit to circuit. However, as a first step, try either increasing the value of the short-circuit resistance to 1mW (10^{-3}), or decreasing the value of the open-circuit resistance to 1GW (10^{9}) or 1MW (10^{6}). (Note that me is the MAST abbreviation for 10^{6} .)

To change either of the default resistances, do the following:

1. Select the Edit > Preferences... menu choice from the Testify Pulldown Menu Bar to invoke the FMEA/Testify Preferences Dialog Box.
2. Select the Simulator tab.

3. Insert the value of resistance you want into the appropriate field.
   - The Open Resistance field specifies the open-circuit fault resistance.
   - The Short Resistance field specifies the short-circuit fault resistance.
4. Click Apply to set your changes.
5. Click Close to close the form.

You are now ready to re-run your tests.

DC Operating Point If you have problems with DC operating point convergence under fault conditions, try determining these operating points by replacing the default initial point of zero volts with the nominal, non-fault operating point value. This should improve both convergence and the speed of the simulation for DC.

Make this change as follows.

1. Select the Analyses > Operating Point > DC Operating Point... menu choice from the Saber Guide Pulldown Menu Bar. This selection brings up the Operating Point Analysis form.
2. Select the Input/Output tab on the Operating Point Analysis form.
3. Change the Ending Initial Point File from the default name dc to a non-default name, such as dcip.
4. Click the OK button to run the analysis.

In the Testify experiments, to change the Operating Point Analysis forms:

1. Click the dcanalysis button on the Experiment form to invoke the Operating Point Analysis form.
2. Click on the Input/Output tab.
3. Change the name in the Starting Initial Point file field from the default of zero to the Ending Initial Point File you created in the last operation, in this case dcip.
4. Click the Accept button.

Now you are ready to re-run your tests.

Note that this procedure may not help convergence if you are altering source values as part of your tests.

Using the PinFault Editor

Parts capable of fault mode behavior have a symbol port property called pin_fault whose value can be changed using the PinFault Editor. You are able to select a combination of possible values to assign to pin_fault. The possibilities are:

- open
- all possible shorts to the part's other pins
- stuck at digital 1 or 0

These choices are hard-coded and cannot be changed.

In order to change the value of pin_fault on a part, follow this procedure:

1. Open the Symbol Editor on the part in your schematic.

   Note that you are modifying the symbol, not the instance.
2. In the Symbol Editor, open the Property Editor on the port that you are interested in.
3. Click in the Value field next to the pin_fault property to invoke the PinFault Editor.
4. Select the faults you want to be able to choose from when creating your test suite and click OK.
5. Click OK on the Property Editor.
6. Close the Symbol Editor.

Using Testify with Hierarchy

Testify allows you to insert faults on one level of hierarchy at a time. You are able to control whether or not the faults are inserted on the pins of the hierarchical symbol or on the parts making up the underlying schematic. You must choose one or the other for any given run of a Testify test suite.

- Inserting Faults on a Hierarchical Symbol
- Inserting Faults within a Hierarchical Symbol

Inserting Faults on a Hierarchical Symbol

This is a useful means of inserting faults when you have a hierarchical symbol representing an integrated circuit or an ASIC whose internal faults are not covered by your test suite. You can use this method of fault insertion when you want to simulate faults on the pins of the IC or ASIC.

1. Open the Symbol Editor on the hierarchical symbol.

   You must modify the ports on the symbol, not the ports on the instance.
2. Open the Property Editor on the first port you want faults applied to.
3. Add a property called pin_fault, without giving it a value, and apply the change.
4. Once the pin_fault property has been applied to the port, click in the Value field of pin_fault to invoke the PinFault Editor.
5. Select the pin faults you want to be available for Testify to apply, and click OK on the PinFault Editor.
6. Apply the pin_fault property value change to the port.
7. Repeat steps 2-6 for the other ports you want to apply the pin_fault property to.
8. Once you have finished applying pin_fault to the symbol ports, save your changes and close the Symbol Editor.

You will need to netlist your design again before running Testify.

Inserting Faults within a Hierarchical Symbol

To allow Testify to be able to activate the faults on parts in a hierarchical symbol's underlying schematic, you must assign the use_fault_model property to specific instances of the hierarchical symbol.

1. Open the Property Editor on the instance of a hierarchical symbol in your schematic.

   Note that you are modifying an instance of the symbol rather than the symbol itself.
2. Add a property called use_fault_model, without giving it a value, and apply the change.
3. Once the use_fault_model property has been applied to the instance, give it a value of no. The possible values are:

no	turns the faults at the highest level of hierarchy for that instance off, allowing the faults of the underlying schematic to be activated by Testify.

yes turns the faults at the highest level of hierarchy for that instance on, preventing the faults of the underlying schematic from being activated by Testify.

4. Apply the use_fault_model value change to the instance.

You will need to netlist your design again before running Testify.

Testify Fault Wrappers

A fault wrapper is a section of code within a model that provides the details of a part's fault behavior.

- If you are writing your own template, all you have to do is add the pin_fault property on the symbol's ports in order to allow the netlister to add the fault wrapper.

  The value or values taken by pin_fault represent the faults requiring modeling.

- If you are writing your own template and want to add a custom fault wrapper to it, write the fault wrapper in the template using the example as a guide, add a pin fault to the symbol, and select Generate Wrappers as Needed in the netlister settings form. This last step keeps the netlister from overwriting the custom fault wrapper you have written.

When a fault wrapper is added to a template of name, a new template called name_f, is created. This template has all the arguments of the original plus some fault parameters, such as failures. The original part is called inside the wrapper template, with all of its arguments passed in. The faults indicated in the properties are constructed using pin fault templates on the pins of the original part. Different faults are introduced into the wrapper by setting the failures argument to the desired value, which is then passed to the relevant pin fault part.

Every wrapper has in its parameters section a message that it prints out when its external parameter print_failures is set to true. This message is a list of the instance name and the possible faults. Therefore, when a netlist is loaded into Saber the parameter print_failures can be changed to true to make Saber print out a list of all of the faults in all of the instances in the entire design.

In the following example of a template with a fault wrapper, the symbol c_f has the property pin_fault(p):(open;short=m). The part of the template extracted from the original template c is shown in courier, and the fault wrapper is shown in bold. The arguments and the original header of the template c are used, along with the default values (strucs are not redefined). The fault parameters are declared and based on the pin_fault property. The pin_fault2.p fault is connected to the port named p of the template c. Normally there are no faults, and the value is none. The other possible values are open and short_m.

```
component element template c_f p m = c,model,l,w,ic,esr,rleak,tc,
                              tnom,ratings,rth_ja,part_type,
                             part_class,failures,use_failures

electrical p, m

process..imodel model = (tc1 = 0,tc2 = 0)

number c = undef,
    esr = 0,
    rleak = inf,
    l = 0,
    w = 0,
    ic = undef,
    tc[2] = [0,0],
    tnom = 27,
    rth_ja = undef

external number temp, include_stress, c_tol, c_vmax, c_vrmax

external standard..pdist pdist

c..ratings ratings=()

string part_type    = “capacitor”,
    part_class   = “generic”

export val p     pwrd
export val tc    tempj
export val joule energy
```

```
# added for fault simulation

fault_modes..use_failures use_failures=yes

struc { enum {short_m,open,none} p=none } failures=()

{
external string print_failures

enum {a_short, in_open, fault_off} p_fail=fault_off

parameters {
    if ((print_failures == “true”) & (use_failures == yes)) {
        message(“%, p short_m”, instance())
        message(“%, p open”, instance())
        }

    p_fail = fault_off

    if (failures->p == off) p_fail = fault_off
    else if (failures->p == open)  p_fail = in_open
    else if (failures->p == short_m)  p_fail = a_short
    else message(“Invalid p failure, %”, instance())
    }
# End of Failure Checking
c.1 p:pi m:m = c=c, model=model, l=l, w=w, ic=ic, esr=esr,
               rleak=rleak, tc=tc, tnom=tnom, ratings=ratings,
               rth_ja=rth_ja, part_type=part_type,
               part_class=part_class

pin_fault2.p in:p out:pi a:m = fault=p_fail,
                       use_failures=use_failures

pwrd = pwrd(c.1)
tempj = tempj(c.1)
energy = energy(c.1)
}
```

Getting Started with Testify

The Testify tutorial consists of two parts:

- Invoking Testify on page 362
- Testing a Design with Testify on page 373

Invoking Testify

In order to run Testify, you must have a schematic of the circuit that you want to test. In this tutorial, you will be developing tests that detect the fault modes of a simple mixed-signal differential amplifier, called DiffAmp.

You can bring DiffAmp up in the schematic capture environment of your choosing. From the following list, select the tool that applies to your environment:

- Setting Up the Design in Saber Sketch
- Setting Up the Design in icms (Cadence)
- Setting Up the Design in DVE (Mentor Graphics)
- Setting Up the Design in Workview Office (Viewlogic on Windows)

Setting Up the Design in Saber Sketch

Note:

This section is for users that are using the Saber Sketch schematic capture program only. If you are using Cadence, Mentor Graphics, or Viewlogic application programs, please refer to Invoking Testify.

Copying the Differential Amplifier Design To start the tutorial, you need to make personal copies of the tutorial files.

1. Make sure that the all necessary environment variables are set for use with Saber. These are typically set up by a System Administrator. For additional installation and system setup information, please refer to the Installing Saber Products manual and Release Notes.
2. Create (if necessary) a directory where you would like the files to be copied. For example, create a directory called synopsys_tutorial.
3. Copy the DiffAmp directory from the following location to the directory you created in the previous step:

UNIX source - `install_home/example/SaberSketch/DiffAmp`
Windows - `install_home\example\SaberSketch\DiffAmp`

4. (Windows only) You must change the file permissions of your local copy of the files so that they are no longer read-only as follows:
   a. From Windows Explorer navigate to your local copy of the design.
   b. Select all the files (Edit > Select All).
   c. Open the Properties dialog box (File > Properties) and select the General tab.
   d. Un-check the Read-only box.
   e. Click OK.

Viewing the Schematic with Saber Sketch You can view the differential amplifier circuit by performing the following steps:

1. Invoke Saber Sketch (sketch). An empty schematic window will appear.
2. Select the File > Open > Design... menu choice to invoke the Open Design Dialog Box.

3. Navigate to the directory containing your copy of the schematic file, diffamp.ai_sch (path/synopsys_tutorial/DiffAmp).
4. Click on the diffamp.ai_sch file.

   This selection will add diffamp.ai_sch to the File Name field.
5. Click on the Open button.

   This selection will open the diffamp schematic in Saber Sketch.

Netlisting the Design and Invoking Testify with Saber Sketch Once the differential amplifier schematic is opened in Saber Sketch, do the following:

1. Choose the Design > Use diffamp menu item. This selection makes the diffamp file your current design.
2. Click on the Testify icon on the Tool Bar at the bottom of the Saber Sketch window to invoke Testify.

   The FMEA/Testify menu form will be displayed.
3. Choose the Edit > Saber/Netlister Settings... menu item from the Saber Guide Menu Bar to display the Saber/Netlister Settings form.
   a. Select the Netlister tab and then the Advanced tab. Set Fault Simulation Mode to Yes. This process will generate a netlist containing the fault models.
   b. Click Apply.
   c. Click Close.
4. Choose the Design > Simulate diffamp menu item to generate the netlist and load it into Saber Guide. (If a dialog box appears, asking if you want to save the design, click on Yes.)
5. Display the Saber Guide icon bar by clicking the Show/Hide Saber Guide icon.
6. Click on the >cmd button on the Saber Guide icon bar. This command invokes the Saber Guide Transcript window.

In the Transcript window, after Saber invocation, you will see the fault templates (_f appended to existing templates) on the netlist.

You are now ready to go to Testing a Design with Testify.

Setting Up the Design in icms (Cadence)

Note:

> This section is for users that are using the Cadence schematic capture program only. If you are using Saber Sketch, Mentor Graphics, or Viewlogic application programs, please refer to Invoking Testify.

Copying the Differential Amplifier Design To start the tutorial, you need to make personal copies of the tutorial files.

1. Make sure that all the necessary environment variables are set for use with Saber. These are typically set up by a System Administrator. For additional installation and system setup information, please refer to the Installing Saber Products manual and Release Notes.
2. Create (if necessary) a directory where you would like the files to be copied. For example, create a directory called synopsys_tutorial.
3. Copy the install_home/example/Cadence/DiffAmp directory to the directory you created in the previous step.

Viewing the Schematic with icms You can view the differential amplifier circuit by performing the following steps:

1. Invoke icms as follows:
    a. Navigate into the DiffAmp directory.
    b. Type icms.
2. Add a new library definition for the diffamp design as follows:
    a. From the icms - Log window, choose the Tools > Library Manager menu item. The Library Manager window will appear.
    b. In the Library Manager window, choose the Edit > Library Path menu item. The CdsLibEditor window will appear.
    c. Follow the instructions at the bottom of the CdsLibEditor window to add the following Library/Path information:

Library	Path
DiffAmp	{path}/synopsys_tutorial/DiffAmp/DiffAmp

    d. When you are finished, choose the File > Save menu choice.
    e. Close the CdsLibEditor window (File > Exit).

   f. Close the Library Manager window (File > Exit).

3. Open the differential amplifier design as follows:
   a. From the icms - Log window, choose the File > Open menu item. The Open File dialog box will appear.
   b. In the Open File dialog box, set the fields to the values as follows:

Library Name	DiffAmp
Cell Name	diffamp
View Name	schematic
Mode	edit or read
Library path file	{path to your cds.lib file}

   c. Once the fields are set correctly in the Open File dialog box, click the OK button. The Open File dialog box closes, and the differential amplifier schematic will appear.
   d. Choose the Design > Check and Save menu item.

4. Set the Project Directory

   Select the Saber > Set Working Directory menu item. The Project Information dialog box will appear.

   Insert your working path, path/synopsys_tutorial/DiffAmp, into the Project Directory field, and click the OK button.

Netlisting the Design Once the differential amplifier schematic is opened, do the following:

1. Choose the Saber > Saber/Netlister Settings... menu item. This selection displays the Saber/Netlister Settings form.
   a. Select the Netlister tab and then the Advanced tab. Set Fault Simulation Mode to Yes. This process will generate a netlist containing the fault models.
   b. Click Apply.
   c. Click Close.
2. Start the netlister by selecting the Saber > Netlist > Start Netlister menu item.

3. Invoke Saber by selecting the Saber > Start Saber Guide menu item.

Invoking Testify With Saber Guide opened and the netlist loaded, you can invoke Testify.

- Click on the Testify icon on the Tool Bar at the bottom of the Saber window to invoke Testify.

  The FMEA/Testify menu form will be displayed.

You are now ready to go to Testing a Design with Testify.

Setting Up the Design in DVE (Mentor Graphics)

This section is for users that are using the Mentor Graphics schematic capture program only. If you are using Saber Sketch, Cadence, or Viewlogic application programs, please refer to Invoking Testify.

Copying the Differential Amplifier Design To start the tutorial, you need to make personal copies of the tutorial files.

1. Make sure that all the necessary environment variables are set for use with Saber. These are typically set up by a System Administrator. For additional installation and system setup information, please refer to the Installing Saber Products manual and the Release Notes.
2. Create (if necessary) a directory where you would like the files to be copied. For example, create a directory called synopsys_tutorial.
3. Using Design Manager (dmgr), copy the install_home/example/MentorGraphics/DiffAmp directory to the directory you created in the previous step.

Viewing the Schematic in Design Viewpoint Editor You can view the differential amplifier circuit by performing the following steps:

1. Create a SABER_EXAMPLE environment variable, whose value is your current working directory, the directory that contains DiffAmp as follows:

```
setenv SABER_EXAMPLE your_data_path
```

2. Change your working directory to DiffAmp as follows:

```
cd $SABER_EXAMPLE/DiffAmp
```

3. Start the Design Viewpoint Editor application by typing the following:

```
dve diffamp
```

4. Setup Saber by selecting the Setup > Saber menu item.
5. Select the File > Save Design Viewpoint > With Same Name > Cleanup Unused References menu item.
6. Click on the OPEN SHEET icon to open the schematic.

Netlisting the Design Once the differential amplifier schematic is opened, do the following:

1. Choose the Saber > Saber/Netlister Settings... menu item to display the Saber/Netlister Settings form.
    a. Select the Netlister tab and then the Advanced tab. Set Fault Simulation Mode to Yes. This process will generate a netlist containing the fault models.
    b. Click Apply.
    c. Click Close.
2. Start the netlister by selecting the Saber > Netlist > Start Netlister menu item.
3. Invoke Saber by selecting the Saber > Start Saber Guide menu item.

Invoking Testify With Saber Guide opened and the netlist loaded, you can invoke Testify.

1. Click on the >cmd button on the Saber Guide icon bar to invoke the Saber Guide Transcript Window.

    You will see the fault templates (_f appended to existing templates) invoked.
2. Click on the Testify icon on the Tool Bar at the bottom of the Saber window to invoke Testify.

    The FMEA/Testify menu form will be displayed.

You are now ready to go to Testing a Design with Testify.

Setting Up the Design in Workview Office (Viewlogic on Windows)

Note:

> This section is for users who are using the Viewlogic Workview Office applications on Windows only. If you are using either Cadence, Mentor Graphics, or Viewlogic Powerview applications, please refer to Invoking Testify.

Copying the Design To start the tutorial, you need to make personal copies of the tutorial files.

1. Make sure that all the necessary environment variables are set for use with Saber. These are typically set up by a System Administrator. For additional installation and system setup information, please refer to the Installing Saber Products manual and Release Notes.
2. Create (if necessary) a directory where you would like the files to be copied. For example, create a directory called synopsys_tutorial.
3. Copy the following directory to your current directory:

```
install_home/example/Viewlogic/DiffAmp
```

   where install_home is the path to the Saber installation.

4. Change your working directory to DiffAmp.
5. Copy the following file to your DiffAmp directory:

install_home/framework/standard/viewdraw.ini

   This file is template initialization file for ViewDraw that you will use as a convenient way to incorporate certain library search paths in later steps.

6. You must change the file permissions of your local copy of the files so that they are no longer read-only as follows:
   a. In your local DiffAmp directory, select all the files (Edit > Select All).
   b. Open the Properties dialog box (File > Properties) and select the General tab.
   c. Un-check the Read-only box.

   d. Click OK.

   e. Repeat steps a through d for the Audio\RLC directory.

7. Open the viewdraw.ini file with a text editor. You must edit the library search paths contained in this file to match the search paths required for your local Saber installation. This a plain-text file, so be sure to save it as a text file after editing.

   Show below is an excerpt from a viewdraw.ini file, showing a few of the library search-path entries.

Modify to point to your local Workview Office installation

```
dir [p] C:\WVOFFICE\wv_libraries (WVLIBRARY)
dir [rm] C:\WVOFFICE\wv_libraries\anlgdev (analog)
dir [rm] C:\Synopsys\saber5.1\framework\viewlogic\symbols\comp (sbr_comp)
.
.
.
```

Modify to point to your local Saber installation

   Use the search and replace capabilities of your text editor to modify the installation-dependent portions of the search-paths so they are correct for your installations.

8. From the Start menu (or WorkView Office shortcut if present) invoke Workview Office (if it is not already active).

9. Start the Project Manager by clicking on the Project Manager icon in the Workview Office task bar (or use the Start menu or a shortcut, as appropriate).

10. Set up a project file in your DiffAmp directory as follows:

   a. In the Project Manager dialog box, choose the File > New menu item. This activates the Project Manager wizard.

   Note:

   If you have previously created a project file you may see a Project Manager Wizard message box asking whether you want to copy the library search paths that were used in that project. If this happens, click on the Don’t Copy button.

   b. In the first dialog box of the Project Manager wizard, enter diffamp.1 in the Project Name field.

c. Edit the Project Directory field, if necessary, to include the correct path to your DiffAmp directory (for example, C:\vwlogic_ex\DiffAmp) then click on the Next button to display the next dialog box.

d. Confirm that the directory location for the project file (DiffAmp.vpj) is the same as given in the preceding step, then click the Next button to display the next dialog box.

e. Again click on the Next button in the dialog box. (No FPGA libraries need to be added.)

f. In the “ViewDraw libraries to use” dialog box, add the library search paths by clicking on the Import button. The Open dialog box appears.

g. In the Open dialog box, click on the viewdraw.ini file name and click the Open button. This adds the library search paths automatically from the viewdraw.ini file.

h. Click on the Finish button.

i. Confirm that the New Project Information dialog box contains the correct project information, then click on the OK button. This completes the System project file setup.

j. Save the project file by selecting the File > Save menu item.

Viewing the Schematic in ViewDraw Open the Differential Amplifier schematic as follows:

1. In the Project Manager, navigate to the DiffAmp directory and open the project file (DiffAmp.vpj).
2. Click on the ViewDraw icon in the Workview Office task bar (or use the Start menu or a shortcut). The Viewdraw session window appears.
3. In the File Open dialog box, double-click on the schematic named diffamp. The schematic appears in ViewDraw.
4. Pull down the ViewDraw Tools menu and check to see if a menu item containing word Saber is present. This menu item will have several Saber-related entries below it, beginning with Start Saber Guide. If this menu item is not present, add it to the Tools menu as follows:

   a. Select the Tools > Customize menu item.

   b. Click on the User Menu selector button.

   c. Type a name such as Saber in the Menu Text entry box. (The name Saber will be used for this menu item in all subsequent instructions.)

   d. Click on the Browse button next to the Command entry box and navigate to the bin directory under your Saber installation directory (typically C:\Synopsys\saberdesigner\bin). Under the bin directory, select the file named menu.exe, then Click the OK button.

   e. Click on the Add button, then the OK button. The Saber menu item should now appear in the Tools menu.

5. From the Viewdraw session window, choose the Tools > Saber menu item to activate the Saber Menu window. This window contains a Saber menu, which you should use throughout the remainder of this tutorial when told to select an item from the Saber menu.

Netlisting the Design Once the differential amplifier schematic is opened, do the following:

1. Choose the Saber > Saber/Netlister Settings... menu item to display the Saber/Netlister Settings form.

   a. Select the Netlister tab and then the Advanced tab. Set Fault Simulation Mode to Yes. This process will generate a netlist containing the fault models.

   b. Click Apply.

   c. Click Close.

2. Start the netlister by selecting the Saber > Netlist > Start Netlister menu item.

3. Invoke Saber by selecting the Saber > Start Saber Guide menu item.

Invoking Testify With Saber Guide opened and the netlist loaded, you can invoke Testify.

1. Click on the >cmd button on the Saber Guide icon bar to invoke the Saber Guide Transcript Window.

   You will see the fault templates (_f appended to existing templates) invoked.

2. Click on the Testify icon on the Tool Bar at the bottom of the Saber window to invoke Testify.

   The FMEA/Testify menu form will be displayed.

You are now ready to go to Testing a Design with Testify.

Testing a Design with Testify

This tutorial shows how to use Testify to develop tests that detect the fault modes of a simple mixed-signal differential amplifier, called diffamp, by using the following process:

1. Setting Up the Tests
2. Determining Nominal Operation and Operational Limits
3. Specifying the Faults to Be Simulated
4. Running the Fault Simulations
5. Displaying the Test Results
6. Interpreting the Results

Setting Up the Tests

The FMEA/Testify menu form has four tabs: Test Setup, Test Ranges, Fault List, and Results. You will process these tabs in order from left to right. This section explains the Test Setup tab.

In the following procedure, you will set up three tests to be performed on the differential amplifier design that you loaded into your schematic capture tool.

1. Select the Test Setup tab on the FMEA/Testify form.
2. To add a test, select Edit > Add Test from the Testify Pulldown Menu Bar. This process invokes the Query pop-up.
3. Enter RiseTime1 in the Test Name field. This test will measure the rise time of the signal out1. Note that Testify can automatically measure any system variable or operational parameter, not just node voltages.
4. Click the OK button to display the Experiment form.

First Test: RiseTime1 The process of entering a test is similar to that of entering a loop or Monte Carlo analysis in Saber Guide. Fill out the Experiment form by performing the following steps:

1. Choose the Analysis > DC Operating Point menu item to add a DC operating point analysis to the experiment.

   Saber requires the DC operating point of the circuit before a transient analysis can be executed. The DC operating point is used as the first data point in the transient analysis.

For this tutorial, you will be using the default settings. However, you can invoke the Operating Point Analysis form and edit these settings either by clicking the dcanalysis button or by clicking the arrow button and selecting the Edit menu item.

2. Choose the Analysis > Transient menu item to add a transient analysis to the experiment. Set the transient analysis up as follows:

   a. Click either the tranalysis button or the arrow button, and select the Edit menu choice. These selections bring up the Time-Domain Transient Analysis form.

   b. Leaving all other entries on the form at their default values, fill out the following fields as indicated:

End Time	20m	Saber will simulate the circuit for 20ms.
Time Step	0.2m	This value gives Saber a time increment to begin solving the circuit.
Monitor Progress	100	Up to 100 data points will be displayed in the Saber Guide Transcript window so that you can monitor the progress of the simulation.

   c. Click Accept. This command closes the form.

3. Choose the PostProcessing > Measures menu item, adding a measurement to the experiment. This capability allows you to perform various measurements on specified waveforms, in this case, out1.

4. Set the measurement up as follows:

   a. Click either the measure button or the arrow button and select the Edit menu choice. These selections bring up the Measurement form.

   b. Click the arrow button next to the Measure field, and select Time Domain > Risetime.

   c. In the Input Plot File field, enter tr. This command specifies the plot file used by the measurement as input data.

   d. In the Curve Name field, enter out1. This command specifies the signal the measurement is applied to.

   e. Click Accept to accept the measurement.

5. Click Accept on the Experiment form to accept the experiment.

Second Test: RiseTime2 This test is similar to the first test.

1. Add another test by selecting Edit > Add Test from the Testify Pulldown Menu Bar. This selection invokes the Query pop-up.
2. Enter RiseTime2 in the Test Name field. This test will measure the rise time of the signal out2.
3. Click the OK button to display the Experiment form.
4. Edit the information left over in the Experiment and Measurement forms from RiseTime1 by doing the following:
   a. Change the measurement by clicking on the arrow button and selecting Edit. The Measurement form will appear.
   b. In the Curve Name field, replace signal name out1 with out2.
   c. Click Accept to accept the measurement.
5. Click Accept to accept the experiment.

Third Test: LowPass3 Unlike the previous two tests, which tests the transient response of the differential amplifier, LowPass3 tests the circuit's frequency response.

1. Add another test by selecting Edit > Add Test from the Testify Pulldown Menu Bar. This selection invokes the Query pop-up.
2. Enter LowPass3 in the Test Name field. This test will measure the low-pass 3dB point of the signal out3.
3. Click the OK button to display the Experiment form.
4. Delete the transient analysis by clicking on the arrow button and selecting Delete.
5. Choose the Analysis > Small Signal AC menu item. This selection adds a small-signal frequency analysis to the experiment.
6. Move the small-signal frequency analysis so that it is performed before the measurement by clicking on the arrow button next to the acanalysis button and selecting Move Up.
7. Click either the acanalysis button or the arrow button, and select Edit, to bring up the Small-Signal Frequency Analysis form.

8. Leaving all other entries on the form at their default values, fill out the following fields as indicated:
   a. Enter 10 in the Start Frequency field.
   b. Enter 100k in the End Frequency field.
   c. Enter 100 in the Monitor Progress field.
   d. Click Accept.
9. Bring up the Measurement form by clicking on the arrow button and selecting Edit. Change the measurement as follows:
   a. Click on the arrow button next to the Measure field and select Frequency Domain > LowPass (3dB point).
   b. Enter ac in the Input Plot File field.
   c. In the Curve Name field, replace signal name out2 with out3.
   d. Click Accept to accept the measurement.
10. Click Accept to accept the experiment.

You now have a demonstration simulation test program that controls the application of stimuli and takes measurements much as it would be done on ATE.

Determining Nominal Operation and Operational Limits

In this part of the tutorial, you will determine the nominal values and the upper and lower limits of the quantities measured by the three tests you set up on the Test Setup tab.

1. Select the Test Ranges tab on the FMEA/Testify form. Your three tests, RiseTime1, RiseTime2, and LowPass3, are listed in the Name column.
2. Make sure that the check boxes next to the test names are filled in, specifying that these three tests should be activated in the fault simulation analysis to be run later in this tutorial.
3. Click on the Nominal button to run simulations under nominal conditions and without faults inserted into the circuit. Once the simulations are finished, the results of the measurements specified by the tests will be put into the cells in the Nominal column.
4. Click on the Limits button to run a Monte Carlo analysis establishing an acceptable range of operation of this circuit. In this tutorial, you will use the default Monte Carlo setup as indicated in the Command field at the bottom of the Test Ranges tab.

If you would like to view the other default values of the Monte Carlo analysis, click on the Edit... button to invoke the Monte Carlo Analysis form.

Once these simulations are finished, the results of the measurements specified by the tests will be put into the cells in the Lower Limit and Upper Limit columns.

Monte Carlo is only one method of establishing limits. You can also enter data, say from test requirements documents, directly into the text boxes, or you can change the Limits Calculated From field to % Nominal and specify lower and upper percentage tolerances.

If you want to rerun the nominal or limits simulations, first you will have to erase those measurements from the Test Ranges tab by selecting Edit > Clear > Ranges from the Testify Pulldown Menu Bar. To erase only the Limits measurements, click in the appropriate boxes and erase the text.

Specifying the Faults to Be Simulated

Now that you have set up the tests you want to run on the circuit and have established acceptable ranges of operation on these parameters, you will specify the circuit faults you want simulated.

1. Select the Fault List tab on the FMEA/Testify form.

   The Possible Faults and Chosen Faults windows are empty.

2. Click on the Get Fault List button at the bottom of the form. The default of this command places all the possible faults for this circuit into the Possible Faults window. Testify generates this list automatically.

3. Highlight the following faults in the Possible Faults window:

   - All of the capacitor faults, such as the following example:

     ```
     /c_f.c2, p open
     ```

   - All of the resistor faults, such as the following example:

     ```
     /r_f.r1, p open
     ```

   - All of the transistor faults, such as the following example:

     ```
     /q2n2222_f.q1, c open
     ```

- None of the lm324 faults

You may need to select the faults of interest that are visible and then scroll the list to select the rest of the faults.

4. Click on the --> button between the windows to copy the selected faults into the Chosen Faults window.

   When the tests are run, each chosen fault will be sequentially inserted into the circuit and simulated, and the specified result will be measured.

You have a great deal of flexibility in selecting the faults you want to activate. If you want to select all possible faults in the design, choose the Edit > Select All > Possible Faults menu choice from the Testify Pulldown Menu Bar. Faults in the Chosen Faults window can be resolved further by using the Part Filter and the Fault Filter at the bottom of the FMEA/Testify form. To see what selections are available to you, click the arrow buttons next to these fields.

Running the Fault Simulations

With the tests set up, the test ranges determined, and the faults of interest chosen from the list of all possible faults, the only step left for you to do is to run the fault simulations.

1. Select the Results tab on the FMEA/Testify form.

   Activated faults are listed in the Faults column. Titles of the tests checked on the Test Ranges tab are at the head of columns where test data will be listed. This data will be the elements of the matrix formed by faults in the vertical and tests in the horizontal.

2. Click on the Run button to perform the fault simulations.

   If the measurement falls within the specified test limits, the fault has not been detected by the test and the measurement is red. If the measurement falls outside the specified test limits, the fault has been detected by the test and the measurement is blue.

   A “No Measurement” condition indicates that there was no information with which to take the measurement, such as a flatline output. This is a detected fault.

   You can edit the colors in the Preferences form.

While the simulations are running, select Edit > Preferences... to look at the option fields in the FMEA/Testify Preferences dialog box. For this tutorial, none of the default values need to be changed, but you may want to change them in the future.

You may pause or stop the simulations while the fault runs are processing. To pause, click on the Pause button at the bottom of the Results tab. The simulation run in process will complete and then processing will pause.

Displaying the Test Results

When the simulations are completed, you can view a report of the test results in either of two formats.

1. To see the default Short Report, click on the Report button to display the Detect/Non-Detect status of the tests versus the inserted faults.
2. To see the Verbose Report:
    a. Close the Short Report.
    b. Select Edit > Preferences... to invoke the FMEA/Testify Preferences dialog box.
    c. Select the Report tab.
    d. Click on arrow next to the Format field and select Verbose.
    e. Click on Apply.
    f. Click Close to close the form.
    g. Click on the Report button at the bottom of the Results tab for a report listing the measurement values as well as the Detect/Non-Detect status.

Both reports show the operational limit values at the top of the report and a fault-detection summary at the bottom.

Interpreting the Results

The Short_Report generated in the previous section compiles several useful items:

- the operational range defined for each test
- a table of chosen faults versus specified tests, with detect (D) or non-detect (N) indicators as elements
- the total number of tested faults
- the total number of detected faults
- the percentage of covered faults
- a list of all non-detected faults.

You can modify report options by using the Edit > Preferences > Reports tab in the Testify dialog box.

Index