SigWave User Guide

Product Version 23.1
September 2023

Contents

Preface

About This Guide

Who is this guide written for?

This guide is meant to assist the new or infrequent user. It provides general information about how to use SigWave to display, edit and analyze simulation waveforms.

Who will benefit from this guide?

Any beginning or intermediate user of SigWave will benefit from the information in this document. The information is organized and based on a typical workflow. It guides you through the various stages required to analyze a simulation waveform.

Note: In order to use SigWave effectively, you must first have a basic understanding of microelectronic circuits and how to analyze them with simulation tools.

Brief Outline of Chapters

This guide consists of three chapters and one appendix.

- Chapter 1: Getting Started with SigWave

  This chapter gives a brief introduction to the SigWave tool. It discusses the use model for SigWave and the various tasks you can perform with the tool.

- Chapter 2: Setting up the SigWave Environment

  This chapter describes the SigWave environment and explains how to change the default settings. It also discusses the basic concepts required to use the tool.

- Chapter 3: Viewing and Editing Waveforms

  This chapter provides general guidelines for using SigWave to view and edit simulation waveforms.

- Appendix A: Understanding Fast Fourier Transforms

This appendix provides explanations of the Fast Fourier Transform (FFT) operations and how they are used in SigWave.

Typographic and Syntax Conventions

This list describes the syntax conventions used in this guide.

`literal (LITERAL)`	Nonitalic or (UPPERCASE) words indicate key words that you must enter literally. These keywords represent command (function, routine) or option names.
argument	Words in italics indicate user-defined arguments for which you must substitute a value.
|	Vertical bars (OR-bars) separate possible choices for a single argument. They take precedence over any other character. For example, `command` argument | argument
[ ]	Brackets denote optional arguments. When used with OR-bars, they enclose a list of choices. You can choose one argument from the list.
{ }	Braces are used with OR-bars and enclose a list of choices. You must choose one argument from the list.
...	Three dots (...) indicate that you can repeat the previous argument. If they are used with brackets, you can specify zero or more arguments. If they are used without brackets, you must specify at least one argument, but you can specify more. *argument*...: specify at least one argument, but more are possible [*argument*]...: you can specify zero or more arguments
,...	A comma and three dots together indicate that if you specify more than one argument, you must separate those arguments by commas.
`Courier font`	Indicates command line examples.

1

Getting Started with SigWave

This chapter discusses the following:

- SigWave Overview on page 11
- SigWave Tasks on page 13

SigWave Overview

What is SigWave used for?

SigWave is a waveform viewer. It displays waveforms based on data generated by simulation tools, emulating the way an oscilloscope works. It also allows you to edit and annotate those waveforms. SigWave is closely integrated with Allegro SI for board-level signal integrity analysis.

Each time you analyze a new pair of driver-receiver pins, you can launch SigWave to display the results of the simulation for that pin pair. You can also load one or more previously saved waveform files in order to compare them.

SigWave supports the display of Time Domain, Bus, Frequency, Eye Diagram graphs and Smith Charts, as well as the application of Fast Fourier Transforms (FFT). It also provides the ability to view and edit spreadsheet data for the displayed waveform. With its annotation capabilities, SigWave allows you to prepare precise documentation of the waveform analyses you conduct.

How does SigWave work with other Cadence tools?

You can launch SigWave from Allegro SI to view and edit simulation waveforms that are generated by that application. Allegro SI provides high-speed design and simulation functionality that helps signal integrity engineers optimize key performance trade-offs in a PCB design. It reduces or eliminates design iterations caused by high-speed design issues.

SigWave is also integrated with the Model Integrity tool. Model Integrity is an Allegro SI tool that enables you to easily manage the integrity of the model data required for high-speed circuit simulations. This weeds out many potential simulator problems caused by faulty model code.

Additionally, SigWave works with SigXplorer. SigXplorer is another Allegro SI tool that is used to develop, simulate and analyze circuit topologies. SigWave can display a single pin pair waveform, as well as multiple waveforms that represent simulation sweeps and complex waveform pairs.

What does SigWave consist of?

The following diagrams explain the main SigWave User Interface (UI).

Figure 1-1 The SigWave user interface

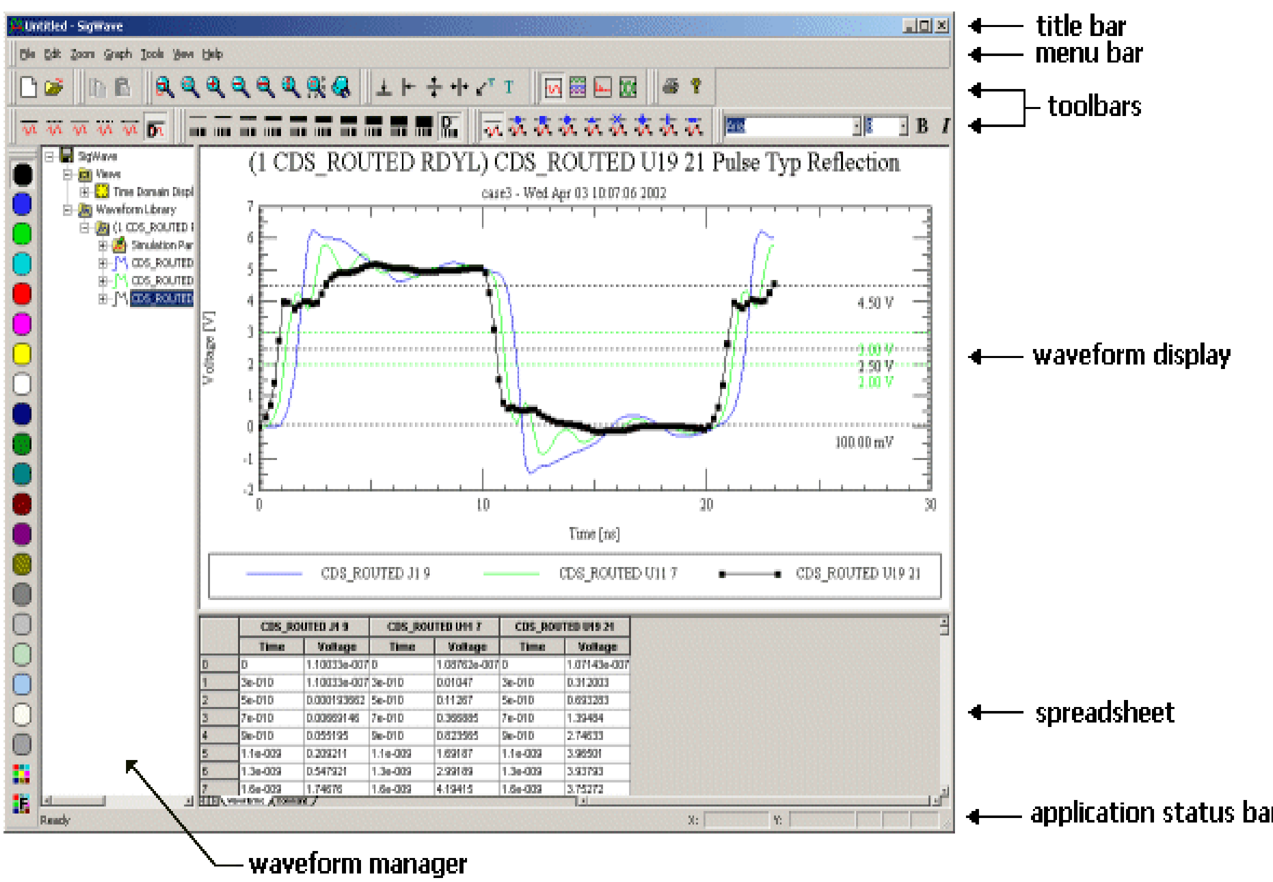
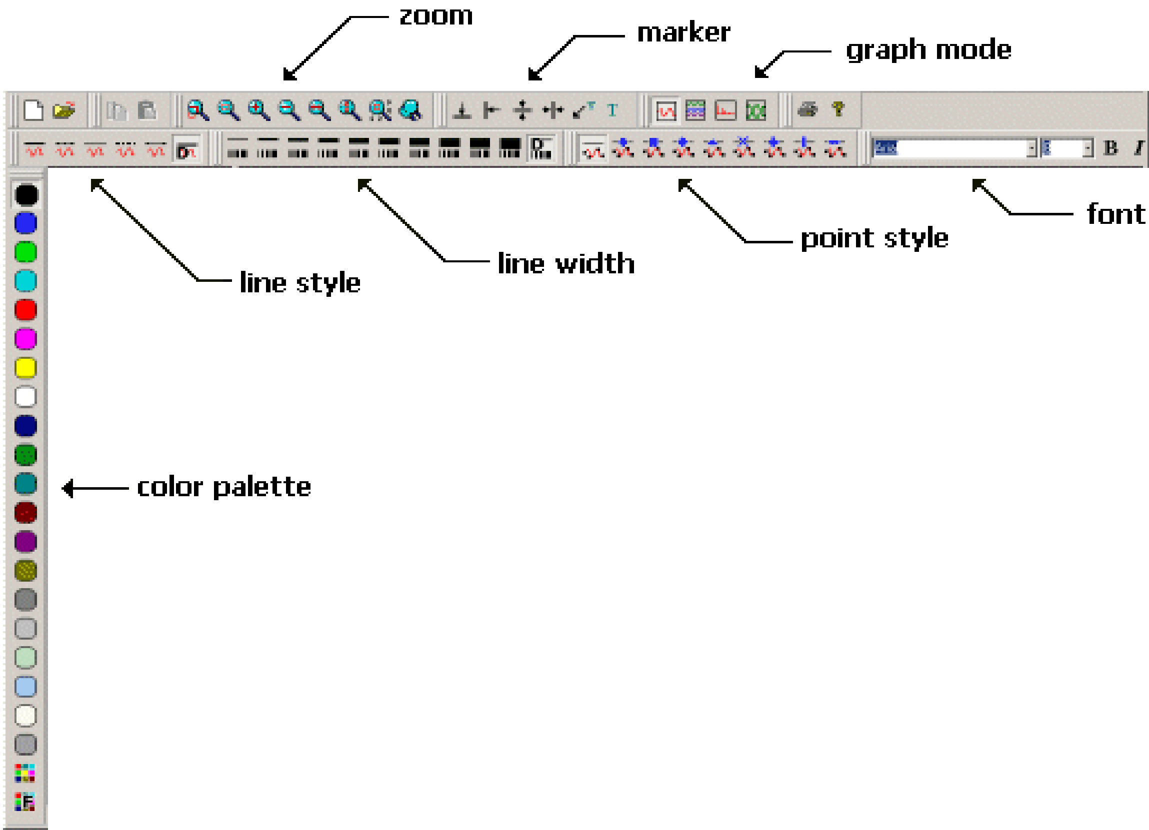


Figure 1-2 The SigWave toolbars



Note: For specific information about the SigWave menu commands, including procedures, see the *SigWave Command Reference*.

SigWave Tasks

What is the typical workflow for SigWave?

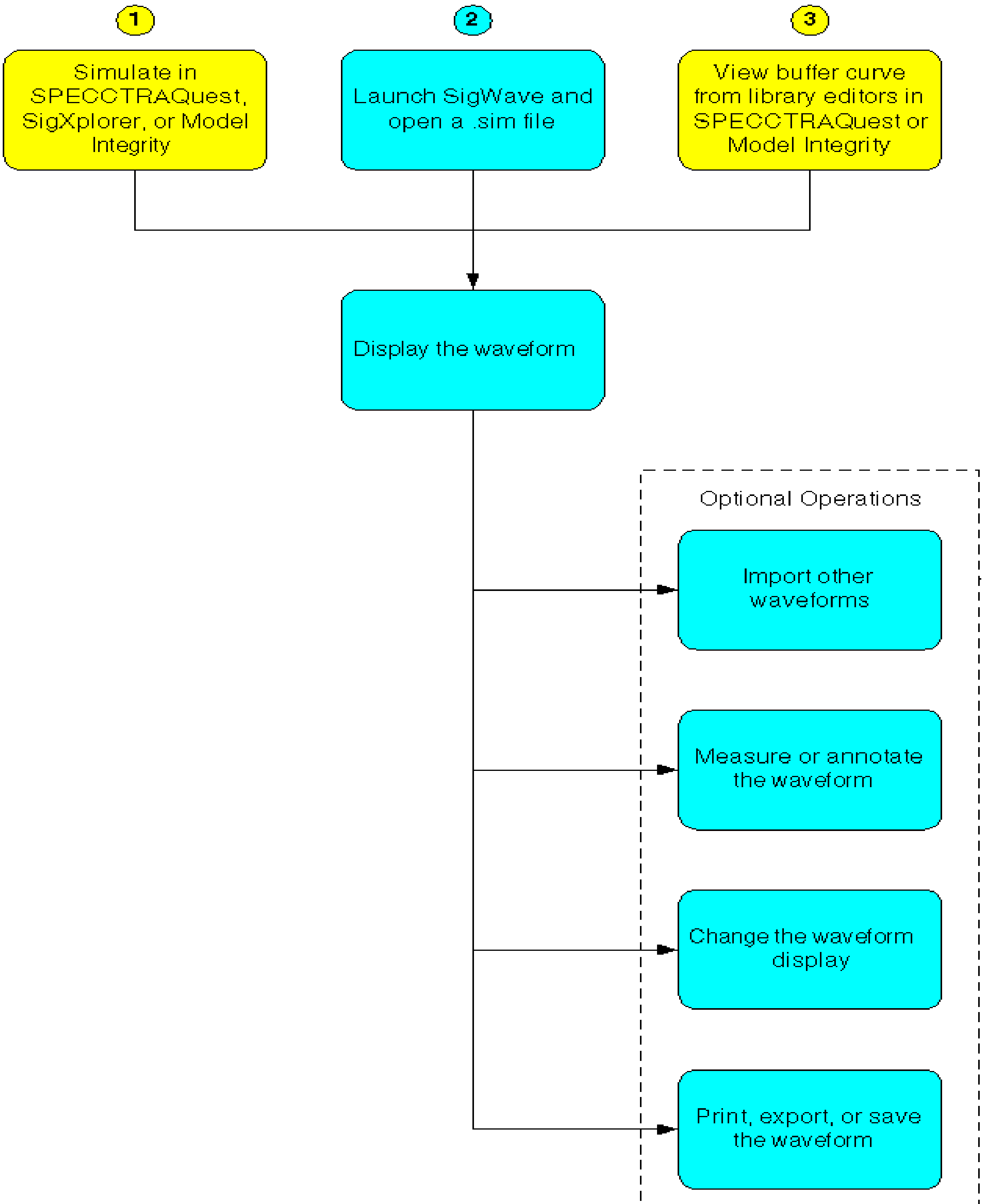
Most typically, engineers use SigWave to display a waveform and analyze the results of a simulation.

The following diagram illustrates the typical steps you would follow to analyze a waveform with SigWave. The most common way to use SigWave is to first perform a simulation with Allegro SI, SigXplorer or Model Integrity. After you run the simulation using these tools, you can display the resulting waveforms in SigWave.

In the diagram, the beginning tasks are labeled 1, 2, and 3, with the most common entry point to SigWave being 1, and the least common being 3. The blocks in the diagram that are colored yellow show operations that are performed outside of SigWave, using other Cadence tools. The blocks that are colored light blue show operations that are performed using SigWave.

The blocks that are grouped together inside the dashed box labeled “Optional Operations” represent the types of tasks that engineers might perform in order to better analyze the plots, or to prepare them for use in reports. These tasks can be carried out in any order, or not at all. SigWave provides these additional features to help you carry out the analysis and document the results of the simulation.

Figure 1-3 The SigWave workflow



2

Setting up the SigWave Environment

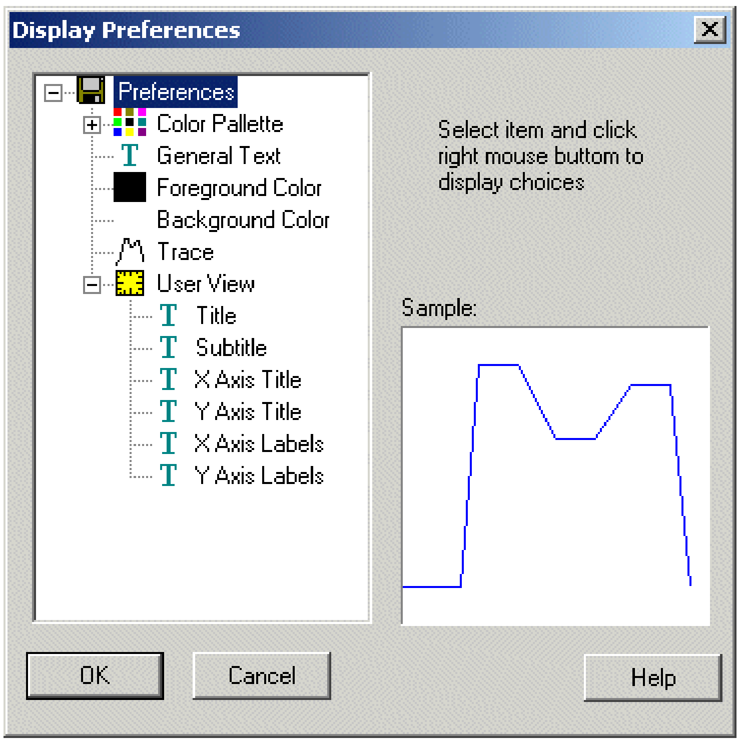
This chapter discusses the following:

Defining Display Preferences

In SigWave, you can define most of the display preferences in a single dialog box called Display Preferences. Any changes you make in the Display Preferences dialog box will be permanent: they will take effect for this and all future sessions. Choose *Tools – Preferences* to access the Display Preferences dialog box.

Note: To make dynamic changes (those that will only apply to the current work session), use the right-mouse button menus for editing (see Editing with the Right Mouse Button menus).

Figure 2-1 The Display Preferences Dialog Box



Defining Trace Colors

SigWave automatically assigns colors to the traces in the waveform display when the file is opened, based on a predefined color palette. The color palette is made up of 20 colors. SigWave assigns the colors in order from 0 to 19. You can change the color palette assignments to meet your preferences.

Defining Line Styles and Widths

SigWave automatically displays the traces as solid lines with the smallest width. You can change the line style or width assignments to meet your preferences.

Defining Text Styles

SigWave uses Times New Roman font, Regular, Size 12, Black, as the default settings for general text. You can change the general text style to meet your preferences.

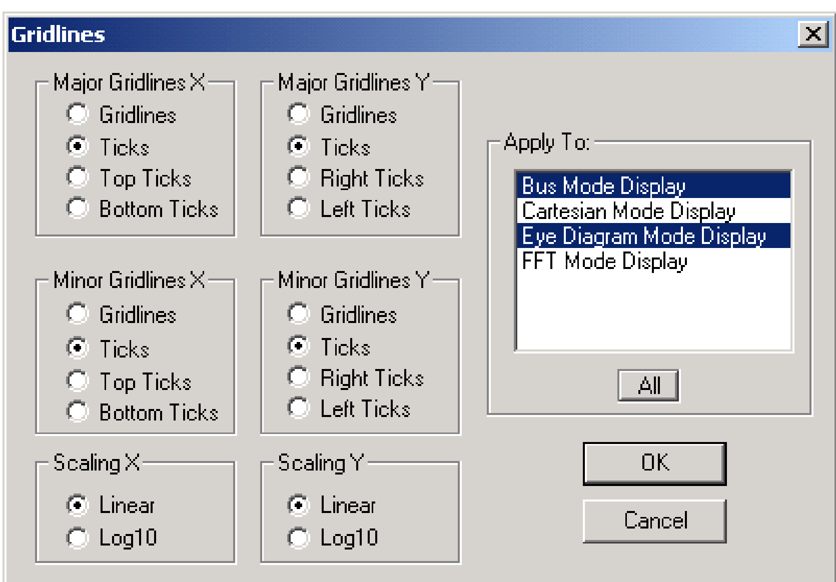
Defining Titles and Labels

You can define settings for titles and labels in the same way you change the text styles.

Defining Grids

By default, SigWave displays gridlines as tick marks along the X and Y axes. You can change the gridline display parameters using the Gridlines dialog box shown in Figure 2-2 on page 19. You access this dialog box by choosing *Graph – Grid*.

Figure 2-2 The Gridlines Dialog Box



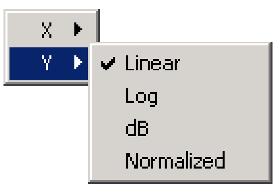
The *Apply To* list box, on the right side of the Gridlines dialog box, enables you to multi-select and apply the changes to several graph modes simultaneously.

Note: Graph modes are created when you first click on their icon in the Graph Mode toolbar. See Graph Modes on page 35 for further details.

Changing the Grid Scale Mode

Linear and Log grid scaling options (bottom of the Gridlines dialog box) are available for both X and Y axes in all Graph modes. You can change grid scaling options dynamically and access additional options for the Y axis by clicking-right in a blank area of the Analog View to display the Grid Scale menu shown in Figure 2-3 on page 20.

Figure 2-3 Grid Scale Menu



Note: The *dB* and *Normalize* options are available only for the Y axis in the Cartesian Graph mode.

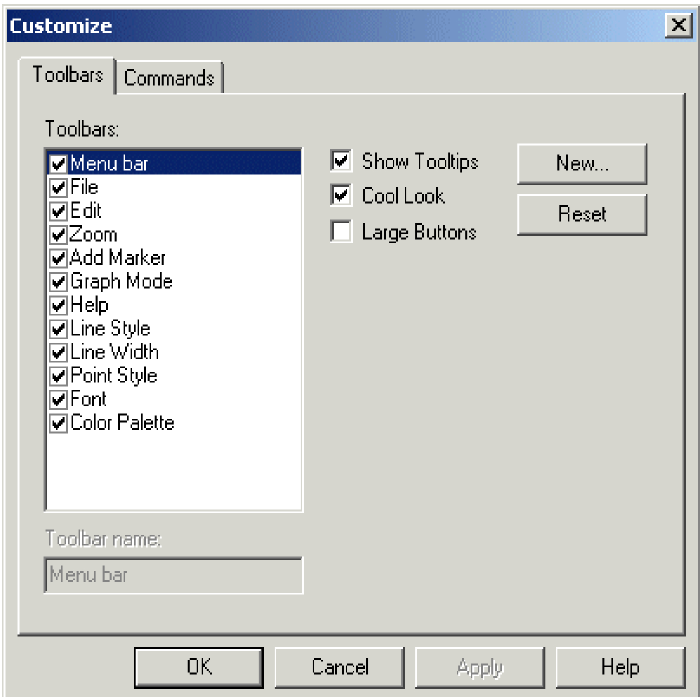
Customizing the Toolbars

The toolbars in SigWave are floating. You can relocate them anywhere you prefer by dragging them to a new position. You can customize the look of the toolbar buttons or change their size. You can also create new toolbars that contain any combination of the pre-defined commands.

Toggling the Display of Toolbars

By default, SigWave displays all of the defined toolbars. You can choose to hide particular toolbars to simplify the SigWave environment. Choose *Tools – Customize* to access the Customize dialog box.

Figure 2-4 The Customize Dialog Box



Changing the Look of Toolbar Buttons

You can choose from the following two basic toolbar styles:

- Cool Look (default)
- Traditional

You can also choose from two different sizes for the command buttons:

- Small (default)
- Large

Displaying ToolTips

By default, SigWave displays ToolTips for the toolbar buttons. ToolTips are command labels that appear when you hover the cursor over a command button. You can disable the display of ToolTips.

Creating New Toolbars

You can create new toolbars that contain different combinations of the existing command buttons. These can be modified in the same way as the pre-defined toolbars. You can also delete any toolbars you create.

Note: Once you define buttons for a new toolbar, you can change the location or arrangement of the buttons by dragging the icons along the bar using the left mouse button.

Modifying User-defined Toolbar Buttons

You can add or remove command buttons on a user-defined toolbar.

Note: You can only add pre-defined command buttons. You cannot define entirely new command buttons in SigWave.

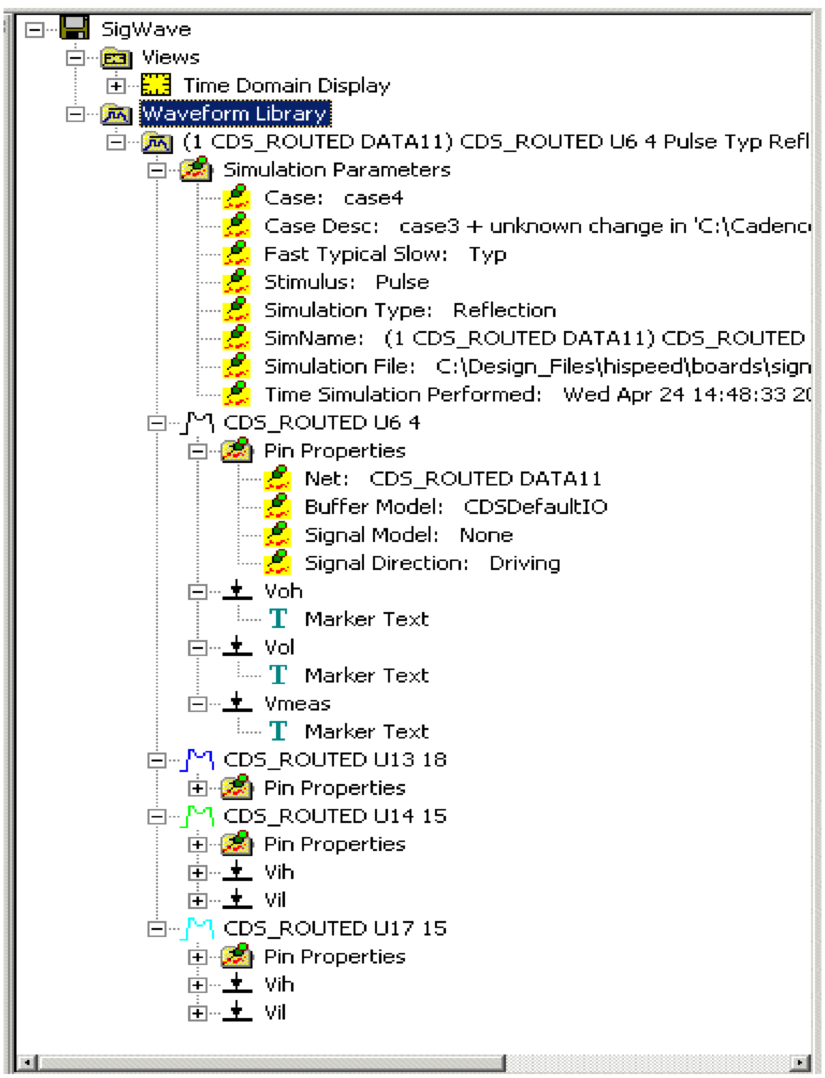
Resetting the Default Toolbars

If you make a mistake in modifying the toolbars, or simply want to restore them to their default settings, you can reset them to their default settings.

Working with the Waveform Library

The Waveform Library in SigWave is an efficient tool for managing the waveforms you display. The Waveform Library is part of the Waveform Manager, located in the left-hand portion of the SigWave window. It consists of several expandable folders that provide information about the simulation parameters, pin properties, and markers for the particular waveforms that are currently open in SigWave.

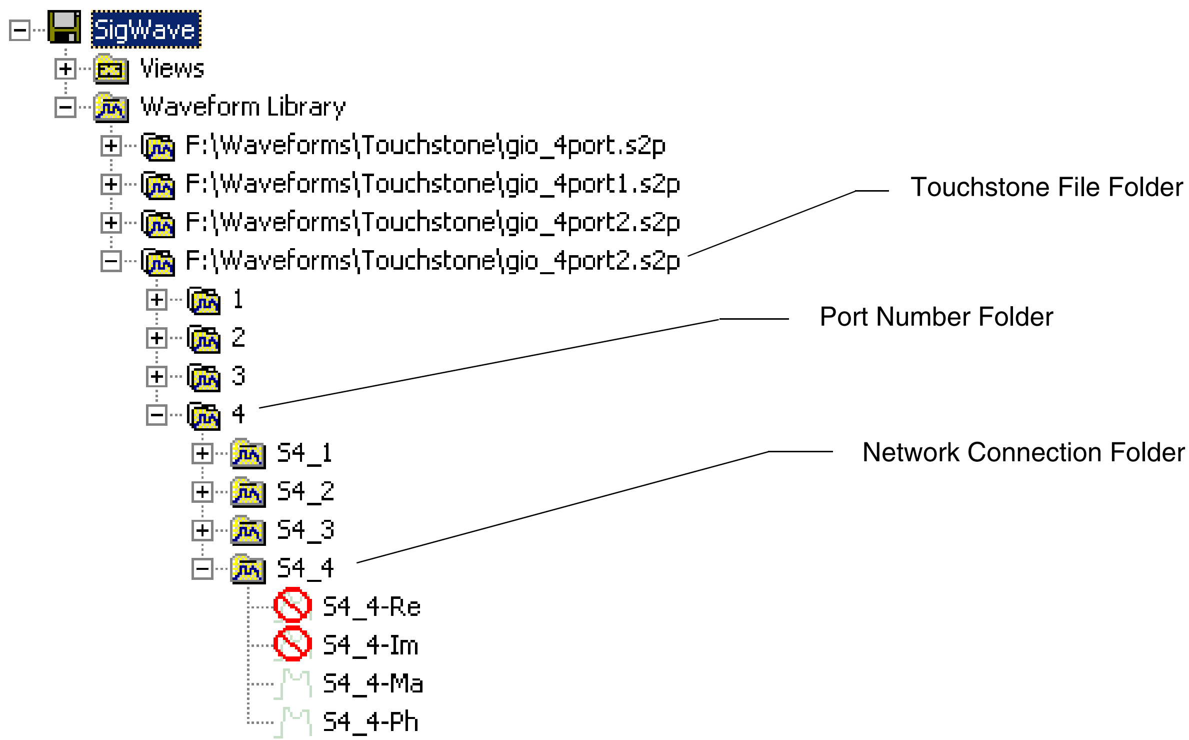
Figure 2-5 The Waveform Library



Complex Waveform File Hierarchy

The folder hierarchy for Touchstone (`.s?p`) files in the Waveform Library consists of a file folder, port number folder and a network connection folder for each port in the file. Figure 2-6 on page 24 shows an example. Note that each waveform component of the complex waveform is labeled according to its format.

Figure 2-6 Touchstone File Hierarchy



Editing the Waveform Display from the Waveform Library

You can modify the display parameters for waveforms and associated markers directly from the Waveform Library. For more information about editing, see Chapter 3, Editing the Plot.

Note: You cannot edit the simulation parameters or pin properties. This information is static and is determined by the simulation that generated the waveform data.

Display State Memory

You can choose to capture the display state of your most recently displayed folder in the Waveform Library for the purpose of automatically resetting a newly created folder to the same state. The new folder can be one created using the *File – Import* command or can be a waveform file coming into SigWave from another Allegro PCB SI tool such as SigXplorer or Model Integrity.

This feature enables you to easily maintain a preferred waveform display state when running successive circuit simulation and waveform analysis cycles between the SigXplorer and SigWave environments.

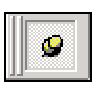
Note: Using Display State Memory neither creates nor destroys items in the Waveform Library.

With Display State Memory enabled, the following display items are saved in memory.

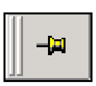
- Zoom scale in the Waveform window
- Folder expansion state within the Waveform Library
- Display state (on/off) of individual waveforms, markers, and folders

➤ To enable or disable Display State Memory, click the Pushpin icon in the toolbar as shown in the following figure.

Figure 2-7 Display Memory Pushpin Icon



Enabled



Disabled

Tip

Once you exit SigWave, display data held in memory is lost. To permanently save the display state as well as other environment settings in SigWave, you should save a workspace file. For further details, see Saving and Restoring Environment Settings on page 28.

Viewing Spreadsheets

In SigWave, you can display a spreadsheet that contains the relevant simulation data for the waveform file that you are analyzing. You can edit the data in the spreadsheet and see the resulting changes in the waveform immediately. You can save the spreadsheet data for use in documenting the analysis you have performed.

The spreadsheet window consists of two tabs:

- Waveforms
- Command

The Waveforms Tab

The Waveforms tab shows the datapoint information for the traces displayed in the Waveform window. You can copy and paste this data to other programs.

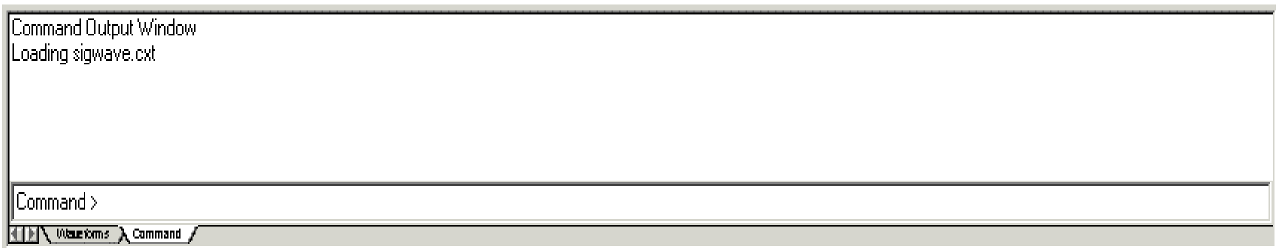
Figure 2-8 The Waveforms tab of the spreadsheet window

	CDS_ROUTED J1 5		CDS_ROUTED R1 2		CDS_ROUTED U10 4		CDS_ROUTED U11 15		CDS_ROUTED U19 2	
	Time	Voltage	Time	Voltage	Time	Voltage	Time	Voltage	Time	Voltage
0	0	1.42858e-007	0	1.44211e-007	0	1.45445e-007	0	1.44495e-007	0	1.45019e-007
1	3e-010	0.26052	3e-010	0.00681154	3e-010	1.45445e-007	3e-010	0.000651466	3e-010	4.26475e-006
2	5e-010	0.581297	5e-010	0.116137	5e-010	1.04672e-006	5e-010	0.0285281	5e-010	0.000596224
3	7e-010	1.11756	7e-010	0.315	7e-010	9.53691e-005	7e-010	0.171015	7e-010	0.0119203
4	9e-010	2.05574	9e-010	0.647191	9e-010	0.00245924	9e-010	0.4332	9e-010	0.0821011
5	1.1e-009	3.22143	1.1e-009	1.24654	1.1e-009	0.0252513	1.1e-009	0.839495	1.1e-009	0.28248
6	1.3e-009	3.63351	1.3e-009	2.17258	1.3e-009	0.131479	1.3e-009	1.54829	1.3e-009	0.64239
7	1.6e-009	3.62556	1.6e-009	3.03981	1.6e-009	0.65844	1.6e-009	2.9515	1.6e-009	1.6298
8	1.8e-009	3.50885	1.8e-009	3.39842	1.8e-009	1.39676	1.8e-009	3.32222	1.8e-009	2.66438
9	2e-009	3.42854	2e-009	3.52261	2e-009	2.59634	2e-009	3.4591	2e-009	3.62288
10	2.2e-009	3.4411	2.2e-009	3.59776	2.2e-009	4.22267	2.2e-009	3.73074	2.2e-009	4.15132
11	2.4e-009	3.45509	2.4e-009	3.84846	2.4e-009	5.78021	2.4e-009	4.08865	2.4e-009	4.53138
12	2.6e-009	3.50916	2.6e-009	4.28674	2.6e-009	6.06655	2.6e-009	4.41523	2.6e-009	5.14221
13	2.8e-009	3.68503	2.8e-009	4.62503	2.8e-009	6.02954	2.8e-009	4.85519	2.8e-009	5.7398
14	3e-009	3.93274	3e-009	5.03642	3e-009	5.94353	3e-009	5.47697	3e-009	5.80606
15	3.2e-009	4.18178	3.2e-009	5.55053	3.2e-009	5.97097	3.2e-009	5.74214	3.2e-009	5.84198
16	3.4e-009	4.48489	3.4e-009	5.59554	3.4e-009	5.95093	3.4e-009	5.63927	3.4e-009	5.81716
17	3.6e-009	4.75139	3.6e-009	5.33115	3.6e-009	5.92305	3.6e-009	5.56971	3.6e-009	5.77848

The Command Tab

The Command tab shows a running log of all the commands that have been executed during the current SigWave session.

Figure 2-9 The Command tab of the spreadsheet window



In the Command line at the bottom of the tab window, you can enter commands manually as an alternative to using the user interface.

Note: This Command line feature is still under development and should be available in its full implementation in a future release.

Displaying the Spreadsheet

You can toggle the display of the spreadsheet on or off, depending on how much space you need to view the waveform display.

Editing the Spreadsheet Contents

You can select one or more cells in the Waveforms tab, then copy and paste the values into a text editor, or alternately into the cells of another spreadsheet program. You can also paste text from another spreadsheet program into the SigWave spreadsheet.

Saving and Restoring Environment Settings

You can save a workspace file to permanently capture the current SigWave environment and preserve it between sessions. When you save the file, any existing waveform annotation that you created as well as other settings that are accessed via the Graphic User Interface are captured and written to the file. In a subsequent SigWave session, you can load the same workspace file to restore the environment and pick up your work right where you left off. For details on the annotation and settings that are saved, refer to Table 2-1 on page 29.

To save a workspace file

1. Choose *File – Save As.*

   The Save As dialog box appears.

2. In the *File name* field, type the new name for the file.
3. In the *Save as type* field, make sure *Workspace Files* (`*.sww`) is selected.
4. In the *Save in* field, specify the directory where you want to save the file.
5. Click *Save*.

   The workspace file is saved to the specified directory.

To load a previously saved workspace file

1. Choose *File – Open*.

   The Open file dialog box appears.

2. Select a workspace (`.sww`) file.
3. Click *OK*.

   SigWave reads the specified workspace file and restores waveform annotation and settings accordingly.

Managing Simulation and Workspace Files

Given that you can import multiple simulation waveform (`.sim`) files into SigWave, the workspace file may not necessarily be associated to a single simulation file. Often times, annotations that you create within SigWave apply more broadly to the entire SigWave environment. However, for file management purposes, it is recommended that you save one workspace file for every simulation file.

In addition to annotation information, the workspace file also stores Trace data points. This serves as a safety measure in the event that the simulation file is lost. It also serves as a compacted storage mechanism since the workspace file is much smaller than the combined simulation files. For example, a minimum workspace file with 5 simulation files is 24KB, the combined size of the 5 Simulation files is 45KB. Minimum workspace file represents a 46% reduction in file size. A minimum workspace file is one with no additional markers and storage of only the Time Domain data.

Note:

- Upon loading a workspace file, SigWave clears the Waveform Library of all objects including all file folders, markers and traces.
- When loading a workspace file, SigWave first determines the version of SigWave used to create the file. If the version of SigWave used to create the file is newer than the version attempting to load the file, SigWave terminates the process and alerts you with a message that indicates the appropriate version of SigWave needed to load the file.

Table 2-1 SigWave Annotation and Settings Saved to a Workspace File

	Name	Mode	Location	Display	Color	Linestyle	Linewidth	Pointstyle	Plotsyle	Font	Font Style	Font Size	Display Text	Snap Mode	Snap Mode Location	Allow Parent Change	Data Points	Markers
Title	X	X		X	X					X	X	X	X					
Subtitle	X	X		X	X					X	X	X	X					
X Axis Title	X	X		X	X					X	X	X	X					
Y Axis Title	X	X		X	X					X	X	X	X					
X Axis Labels	X	X		X	X					X	X	X	X					
Y Axis Labels	X	X		X	X					X	X	X	X					
Horizontal Major Gridlines	X	X		X	X	X	X						X					
Horizontal Minor Gridlines	X	X		X	X	X	X						X					

Vertical Major Gridlines	X	X		X	X	X	X						X					
Vertical Minor Gridlines	X	X		X	X	X	X						X					
Legend	X	X		X	X	X	X			X	X	X	X					
Grid Outline	X	X		X	X	X	X						X					
Waveform Library	X	X		X									X					X
File Folder	X	X		X									X					X
Trace		X		X	X	X	X	X	X				X				X	X
Horizontal Marker	X	X	X	X	X	X	X	X					X	X	X	X		
Vertical Marker	X	X	X	X	X	X	X	X					X	X	X	X		
Diff Horizontal Marker	X	X	X	X	X	X	X	X					X	X	X	X		
Diff Vertical Marker	X	X	X	X	X	X	X	X					X	X	X	X		
Angled Marker	X	X	X	X	X	X	X	X					X	X	X	X		
Pointer Text (Marker)	X	X	X	X	X					X	X	X	X					
Text (Marker)	X	X	X	X	X					X	X	X	X					

3

Viewing and Editing Waveforms

This chapter discusses the following:

Opening a File

After you perform a simulation with tools such as Allegro SI or SigXplorer, you can open the waveform file (`.sim` or `.cim`) and view it in SigWave. You can also import simulation files in formats other than the `.sim` format.

Loading a Simulation (.sim) File

If SigWave is already running, you can open a different simulation file and display the corresponding waveform in place of the current waveform.

Note: You can only open a simulation (`.sim`) file that was previously created by Allegro SI. To create a new waveform display based on a different simulation, you must simulate again in Allegro SI.

SigWave supports two variations of the `.sim` format. The newer format introduced in release 16.01 contains headers and data groups that enhance overall SigWave performance–particularly when working with very large waveforms–and which support more waveform properties, such as waveform display status. By contrast, old format `.sim` files do not contain header or data type information. A sample of this format is

```
(SimFile
(Header
(Version 16.01)
(EyesInfo
(Table
(Fields eyeName)
(Data
Eyemask1
Eyemask2)))

(WavesInfo
(Table
(Fields Name!Index!Display)
(Data
"DESIGN.IO1.1"!3!0
"DESIGN.IO2.1"!2!0
"DESIGN.IO5.1"!1!0
"DESIGN.IO6.1"!0!0))))

(DataGroup
(Eyes
(Eye
(Name Eyemask1)
(Table
(Fields x!rmin!fmax)
(Data
0.35!0!0
0.5!0.05!-0.05
0.65!0!0)))

(Eye
(Name Eyemask2)
(Table
(Fields x!rmin!fmax)
(Data
0.1!0!0
0.5!0.1!-0.1
0.9!0!0))))

(Waves
(Labels xLabel=Time!xUnit=s!yLabel=Voltage!yUnit=V)
(Table
(Fields Index!Time!Potential)
(Data
0!0!4.99998
```

```
1!0!4.99998
2!0!4.99998
3!0!4.99998
0!3e-010!4
1!3e-010!3
2!3e-010!2
3!3e-010!1
0!5e-010!1
1!5e-010!2
2!5e-010!3
3!5e-010!4
0!7e-010!5
1!7e-010!6
2!7e-010!7
3!7e-010!0
)))))
```

When you open a new format `.sim` file in SigWave, the tool checks the header information to determine what type of objects exist in the waveform; for example, `WavesInfo` contains waveform information.

```
(WavesInfo
(Table
(Fields Name!Index!Display)
(Data
"DESIGN.IO1.1"!3!0
"DESIGN.IO2.1"!2!0
"DESIGN.IO5.1"!1!0
"DESIGN.IO6.1"!0!0))))
```

Each line of information in the `WavesInfo` section represents a waveform's properties, specifically it's name, index, and whether or not it is displayed in SigWave. A `0` value indicates that the waveform is initially hidden in the waveform library, a `1` indicates that it is visible. When you open the `.sim` file, if all the waveforms are set to `0` (hide), SigWave retrieves only the waveform names and lists them in the tree view. If some waveforms are set to `1`, SigWave retrieves those waveforms' data for display.

A progress bar in the SigWave GUI tracks the progress of waveform data that you select to read into SigWave, as when you change the mode of a waveform in a library tree from Hide to Display, or if you open a `.sim` file whose waveforms are set to `1` (display). While the progress bar is active–that is, reading the data–you can cancel the action of displaying waveform data by pressing the STOP button. Completed waveform data will be displayed. If a waveform's data was not completely read, SigWave will indicate it by adding an exclamation mark on its icon.

SigWave also closes all other simulation files and removes them from the waveform library tree. To add a SigWave file to the waveform library for display and keep open other simulation files, choose *File – Import – SigWave* file (`.sim`).

Importing a Simulation File in a Different Format

SigWave can display waveforms from simulation files generated by other simulation tools. SigWave supports the conversion and import of files in the following formats:

- HSPICE
- Quad
- HP Scope
- Tek Scope
- LeCroy Scope
- Berkeley SPICE
- Spectre PSF ASCII
- Touchstone

Note: You can use the option *File – Import – Sigwave .sim file* to open an Allegro SI simulation file in the same way you would use the *File – Open* command.

Creating Custom Conversion Scripts

SigWave allows you to modify the pre-defined conversion scripts as needed, or create new ones to be used with other simulation file formats. (Such conversion scripts are typically written in programming languages such as Perl.)

If you create new conversion scripts for additional simulation file formats, you can add those to the list of available formats in the *File – Import* menu. This provides added flexibility to the display capabilities of SigWave.

Displaying the Waveform

You can view both single and complex waveforms. Complex waveforms consist of two waveform pairs (Real-Imaginary and Magnitude-Phase). These waveform pairs are linked together. That is, if a change is made to the data points of one pair, the datapoints on the corresponding pair adjust automatically. A complex waveform is represented as a folder in the Waveform Library and can be expanded to show its simple waveform components.

To enhance performance when you are opening a `.sim` file containing large numbers of data points, SigWave filters the number of data points that display along the waveform. By default, data point boxes are drawn at specific intervals, calculated by the total number of data points on the waveform.

SigWave provides a range of display modes and functions. Each display mode provides a different method for analyzing simulation results. The display functions allow you to easily view a particularly significant portion of the waveform display as you work in each mode.

You can also take advantage of SigWave powerful Fast Fourier Transform (FFT) operations. SigWave automatically performs FFT unless you specify not to. For more information about setting up the FFT operations, see Performing Fast Fourier Transforms (FFT).

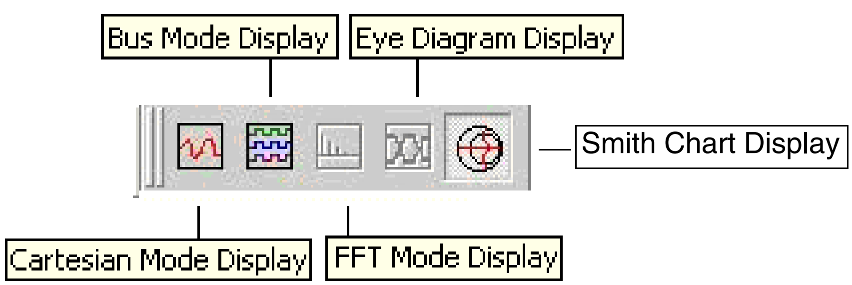
Displaying in Various Modes

Graph Modes

You can display waveforms in one of four different graph modes. You can switch quickly between these modes by clicking the appropriate button on the Graph Mode toolbar as shown in the following figure.

■

Figure 3-1 The Graph Mode toolbar with ToolTip descriptions



Note:

- These buttons do not control the visibility waveforms. To view waveforms in the Waveform Display window, you must have the display of individual waveforms as well as all folders in their hierarchy enabled in the Waveform Manager window.

- The Eye Diagram and FFT graph modes are only available for waveform folders whose x-axis is in time units, or S.

- The Smith chart graphs are available only for S-Parameter Touchstone files.

Cartesian (Voltage [V] / Time [ns]) graph

The Cartesian Mode emulates an oscilloscope function. This is the default mode for SigWave.

Cartesian graphs are used to show the behavior of one or more signals over the simulation time. Multiple signals are not offset from each other on the y-axis.

Bus (behavior over Time [ns]) graph

The Bus mode emulates a logic analyzer.

A Bus graph is a time domain graph which separates multiple signals on a single y-axis of a plot, so that they do not overlay.

FFT (Spectrum [dbV] / Frequency [GHz]) graph

The FFT mode emulates a spectrum analyzer.

FFT graphs are used to determine the amplitude of the various frequency components that make up a signal. This is particularly useful in EMI analysis.

Eye Diagram (Voltage [V] / Time [ns]) graph

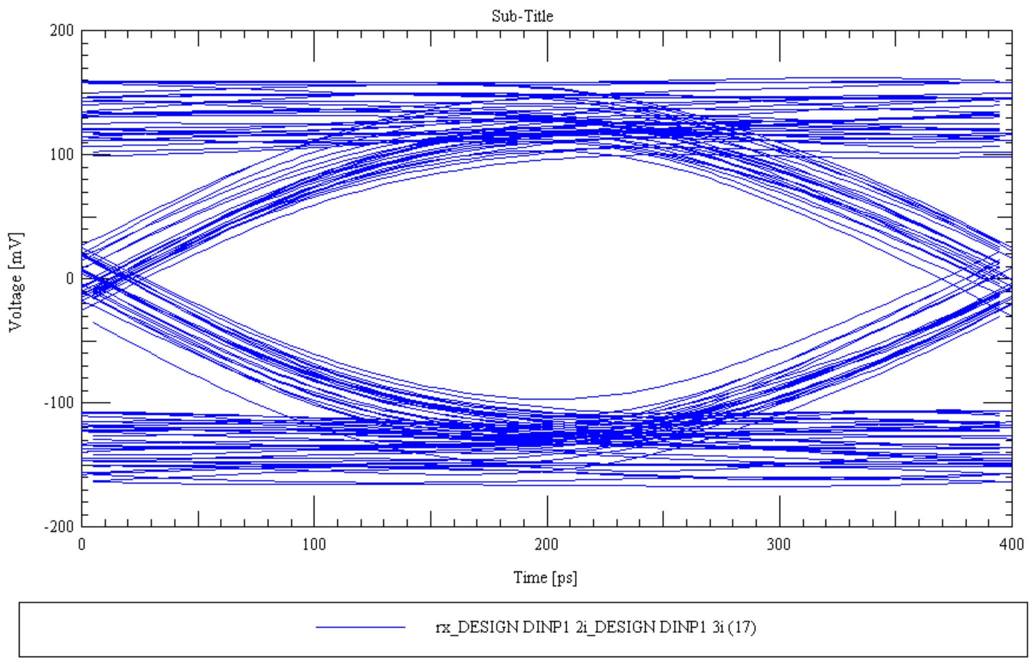
Eye diagrams are used in the analysis of high speed data and clock transmissions. The diagrams consist of waveform traces that are wrapped around on the display at a period equal to the clock period.

Data sequences are normally random so that the probability of a 0 or a 1 are equally likely for a given bit. For a long sequence of bits, an eye diagram has a transition region at the left and right sides of the diagram (where data is transitioning from 1s to 0s and vice versa) and lines at the high and low levels that define the bit logical level. The center of this region is normally where a clock edge samples the data bit.

For an ideal, lossless line without distortion, the eye pattern for a data signal looks like a dual-sided trapezoid with sharp, completely overlapping lines. For a lossy line with distortion, noise, or jitter, the eye pattern looks like a fuzzy-edged rounded trapezoid with a collapsing area in the center and an open background. This pattern resembles an open eye, thus the name.

Note: The Eye Diagram and FFT graph modes are only available for waveform folders whose x-axis is in time units. The graph mode is automatically reset to Cartesian when a file that does not have a time x-axis unit is opened or imported.

Figure 3-2 An example of an eye diagram plot



In extreme cases, the high, low, left, and right sides of the eye collapse, and data transfers are lost. For data streams with encoded clocks, noisy signals result in clocks with high jitter. This might result in sampling errors or setup and hold time violations.

Eye diagrams are commonly used in high speed serial links (Ethernet, FDDI, T1, Sonet) where line losses, dispersion, and noise effects can drastically affect the quality of the data transfer. This quality is often measured in terms of Bit Error Rate (BER) and is determined by the ratio of bit errors to bits transferred. For high speed links, these rates are in the range of 10e-9 to 10e-12.

From an eye diagram, you can estimate how low the BER will be. If you have information for the receiving clock, you can sweep the clock position through the eye and measure the BER. From this measurement, you can create a plot of BER versus clock edge position. This plot is commonly called a waterfall because its sides resemble a waterfall.

Buses or links that have a high BER will have a high rate of data errors. These errors can be corrected, either by applying an error correction algorithm or by retransmitting the data. In either case, errors result in lower throughputs and poorer system performance. In an extreme case, an uncorrected error can cause system errors or a crash. Links with low BERs run more efficiently and reliably.

Different types of waveforms provide different conditions of measurement. The measurement values of waveforms are dependent on where the simulations are derived, either from

Channel Analysis simulation in SigXplorer, Bus Simulation in PCB SI, or in a normal simulation. In SigWave, simulation measurement values are displayed thusly:
:

Simulation Type	Measurement Values
Channel Analysis	Eye Height, Eye Width, Jitter Margin, Noise Margin
Bus	Eye Height, Eye Width, Apt. Width, Jitter
Normal	Eye Height, Eye Width, Jitter, Jitter Margin, Noise Margin

Eye Diagram Display Preferences

You can set display parameters for eye diagrams at different levels by choosing *Graph – Eye Diagram Preferences* from the SigWave menu bar or by selecting the Eye Diagram option in the right button menu when you highlight a .sim file or a waveform.

- When you set the preferences by opening the dialog from the menu bar, the settings are applied globally
- When you set them by opening the dialog with a right button click on a highlighted waveform, the settings are applied only to that waveform.
- When you set them by opening the dialog with a right button click on a highlighted `.sim` file, the setting are applied to all the waveforms in the .sim file.

The Eye Diagram Preferences dialog box appears as shown in Figure 3-3 on page 38.

Figure 3-3 Eye Diagram Preferences Dialog Box

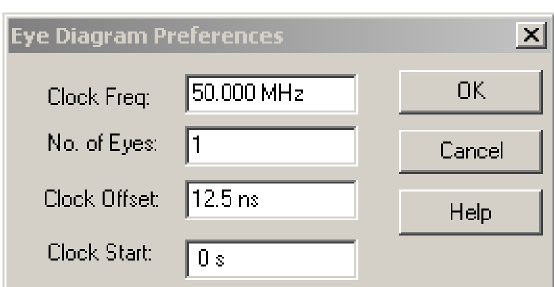


Table 3-1 Eye Diagram Preferences Dialog Box Options

Option	Description
Clock Freq(uency)	Lets you set a value that represents the clock frequency setting established in the custom stimulus controls in SigXplorer. The default is 50 MHz (except when the simulation has originated in a current session of SigXplorer).
No. of Eyes	Sets the number of eyes you want to display.
Clock Offset	Defines the amount of time by which the eye diagram view is shifted.
Clock Start	Enables you to define the point in time where the eye pattern data should start. Data points before this point are removed for display purposes.

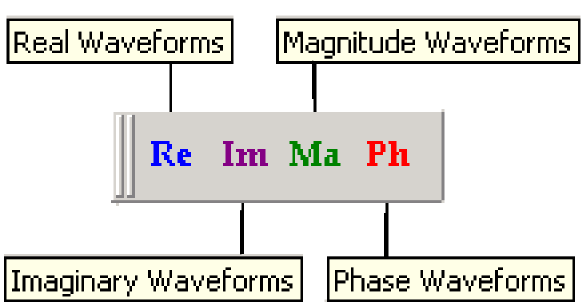
Complex Waveform Format Modes

You can display complex waveforms in one of four different format modes. Switching between these modes is done by clicking the appropriate button on the Format Mode toolbar as shown in Figure 3-4 on page 39.

Note:

- These buttons do not control the visibility of individual waveforms. To view waveforms in the Waveform Display window, you must have the display of individual waveforms as well as all folders in their hierarchy enabled in the Waveform Manager window.
- When changing waveform visibility in the Waveform Manager window, the format buttons do not reset to reflect the current format status of the visible waveforms.

Figure 3-4 The Format Mode toolbar with ToolTip descriptions



Complex Waveform Format buttons are currently used to support the display of S parameters. Since S parameters are in complex number form, they can be displayed either in real/imaginary or magnitude/phase format.

The following table briefly describes the function of these buttons.

This button. . .	Enables / disables the display of all waveforms that show. . .
Re	Real part of the S parameter.
Im	Imaginary part of the S parameter.
Ma	Magnitude of the S parameter.
Ph	Phase (in degree) of the S parameter.

Format and Plot Scale

The magnitude/phase format is often used for S parameter display and analysis.
The magnitude of the S parameter is closely related to the signal power that is transmitted and reflected. For this reason, in addition to the linear scale of the plot, a dB plot is provided for the magnitude display. The S parameter phase can be used to analyze the signal delay and distortion and is best plotted by linear scale.

Note: When using the real/imaginary format, the linear scale should be used.

Smith Charts

SigWave lets you view Smith charts in addition to the graph types previously described. You use Smith charts in SigWave to display the results of S-Parameter curves. Smith charts in SigWave support impedance, admittance, and polar views in S-Parameter Touchstone files. These view types are illustrated in the following figures:

Figure 3-5 Smith Chart Default View (Impedance)

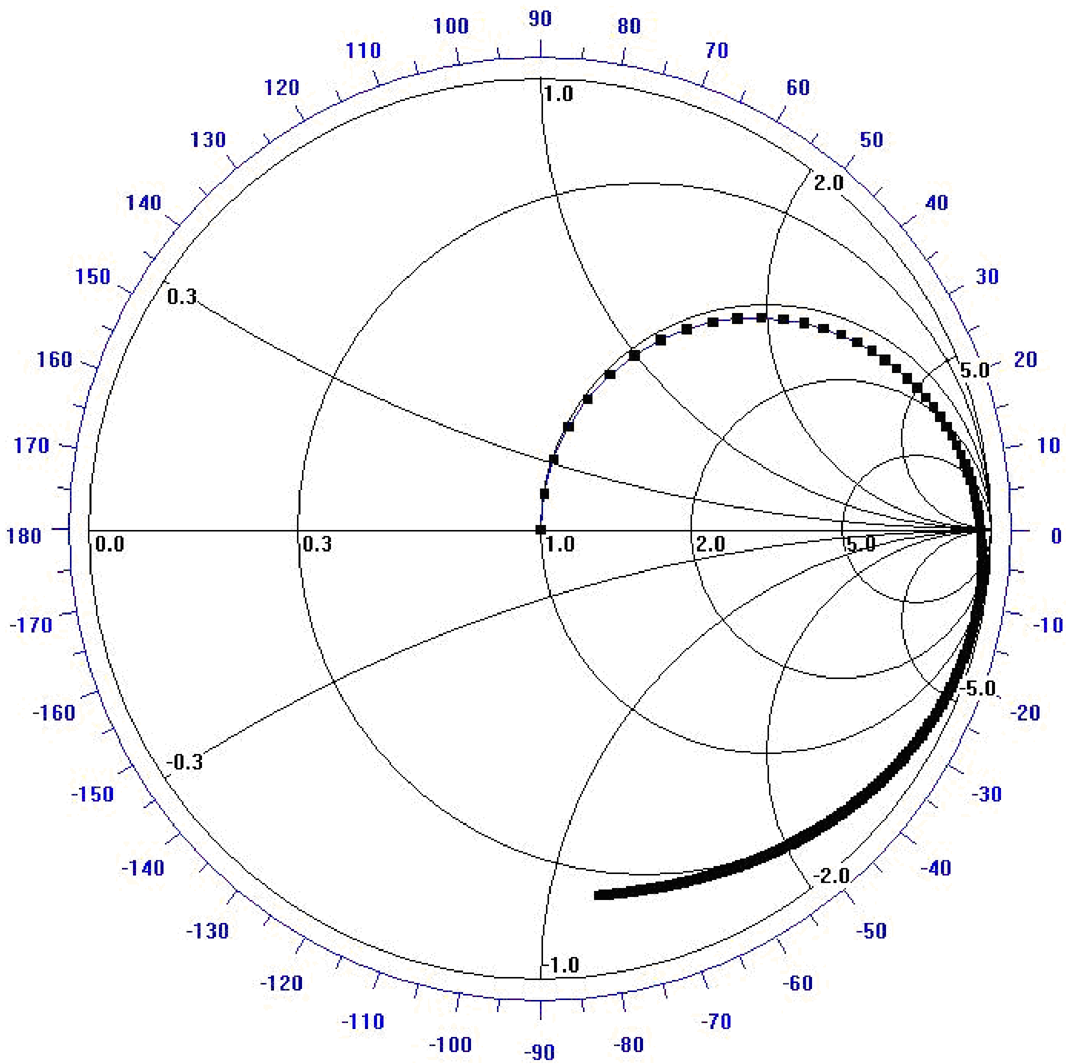


Figure 3-6 Smith Chart Admittance View

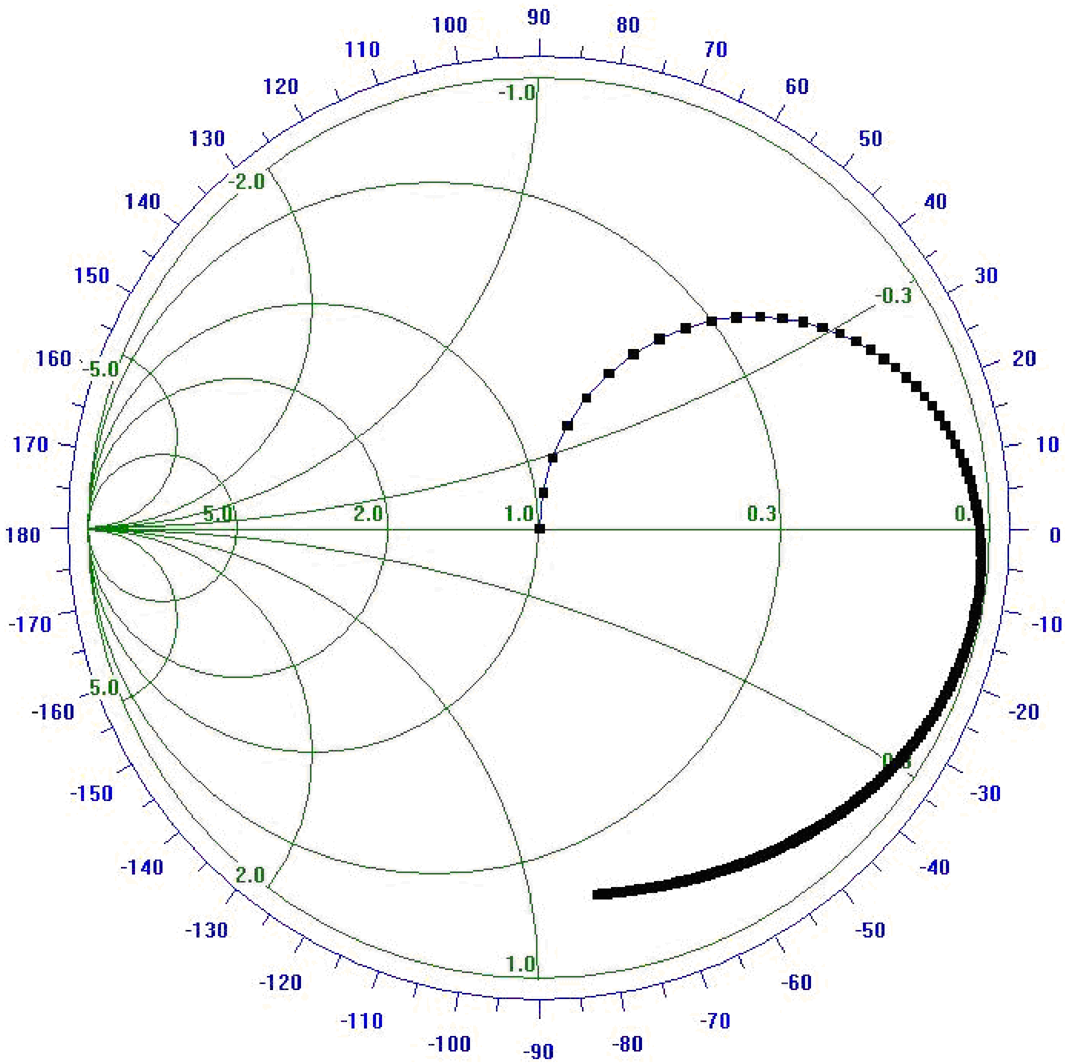


Figure 3-7 Smith Chart Combination View (Impedance and Admittance)

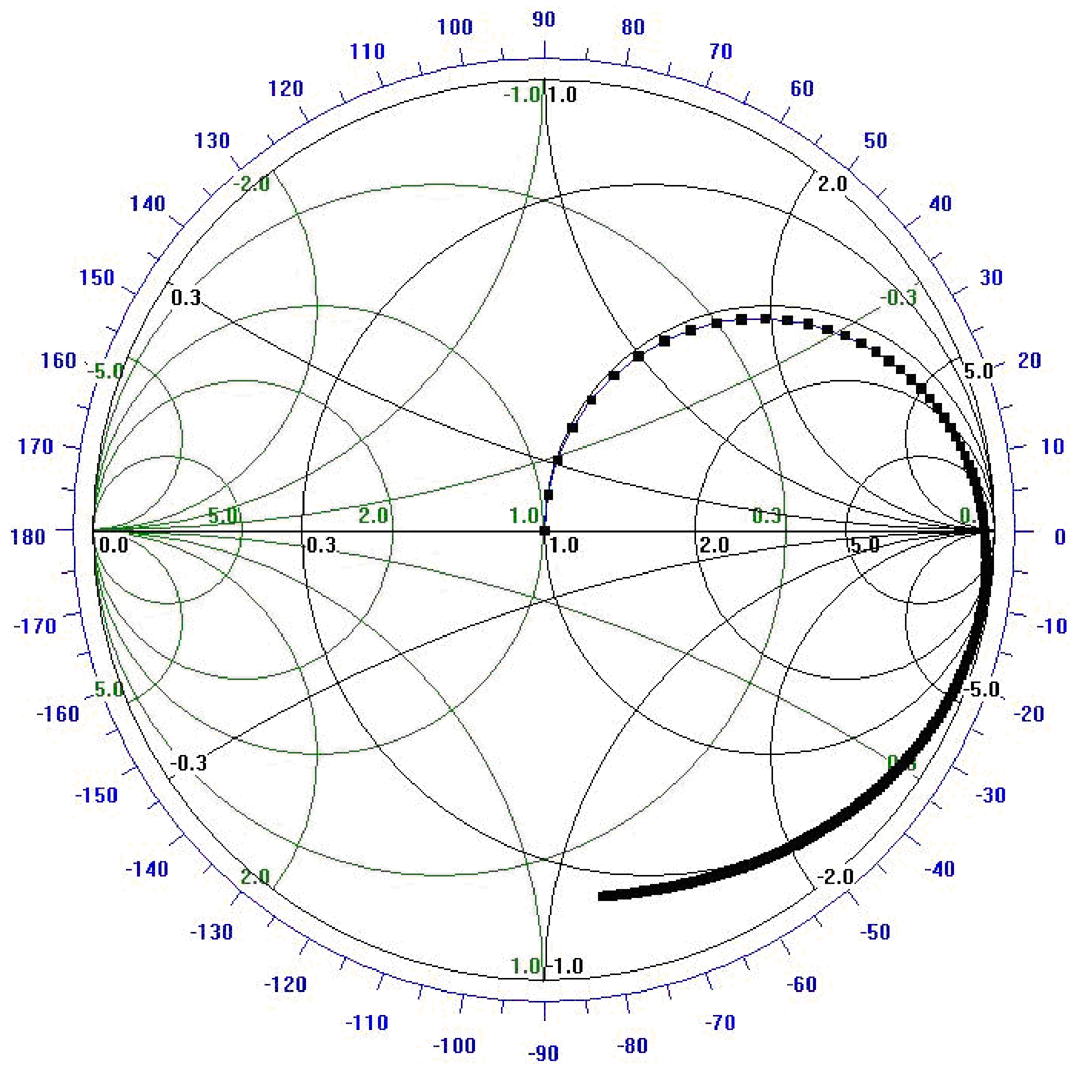
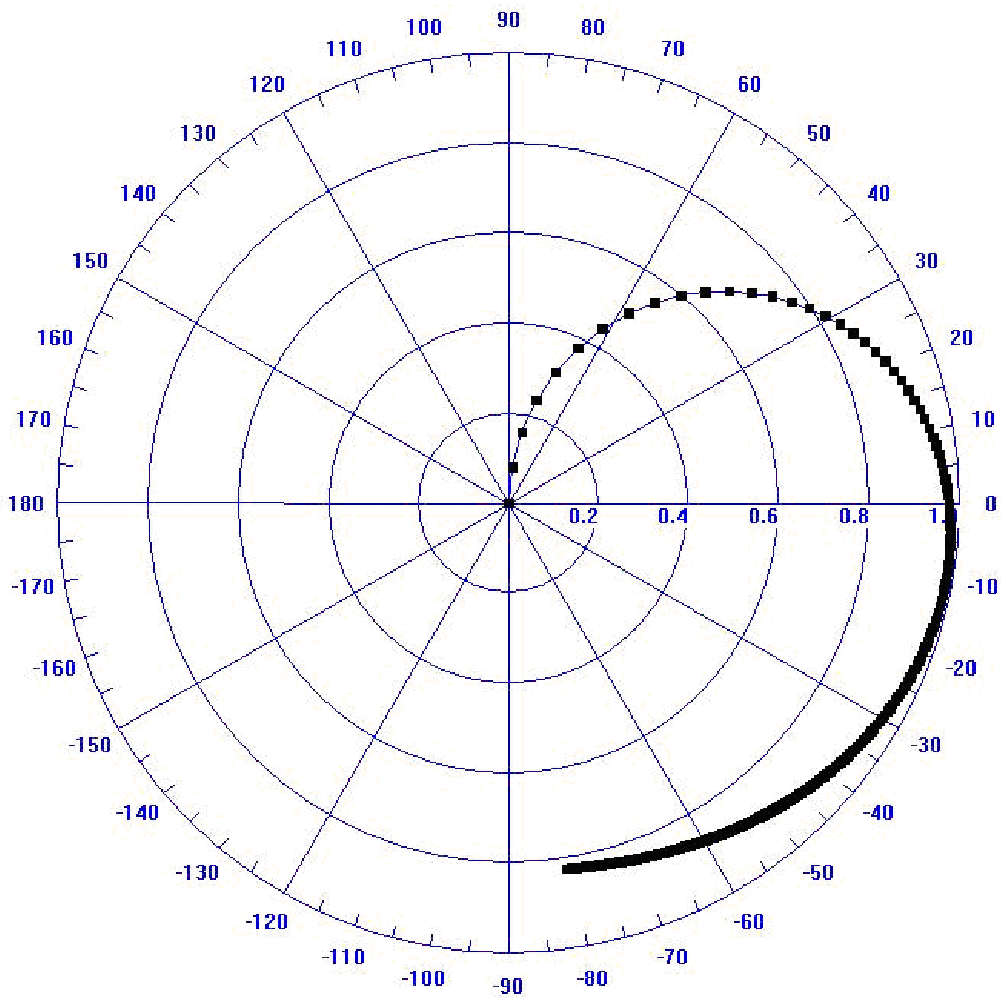
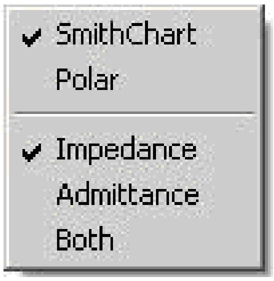


Figure 3-8 Smith Chart Polar View



You can alternate between each of these views by clicking the right mouse button in any blank area of the Smith chart to display a popup menu (Figure 3-9).

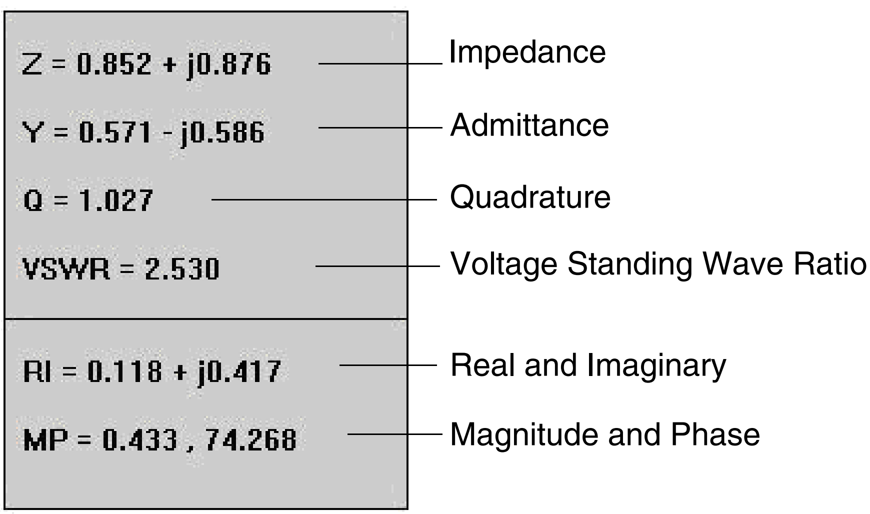
Figure 3-9 Right button popup



If the current Smith chart view is in Polar mode, the Impedance/Admittance views are inactive.

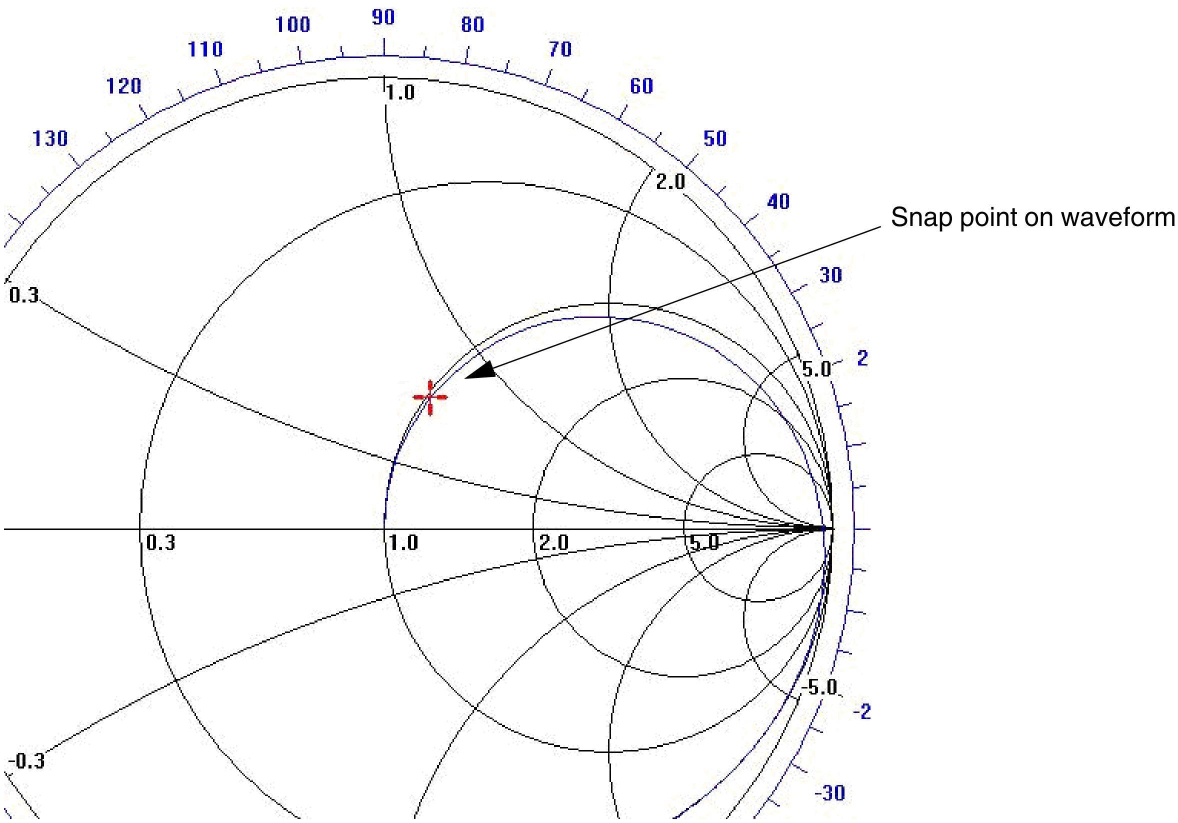
When you display a Smith chart in SigWave, it is accompanied by a data window (Figure 3-10) that displays the current point's parameters. The default current point is the general cursor position, unless you are in Snap mode, in which case, it is the position of the snap point in the selected waveform.

Figure 3-10 Smith Chart Data Window



To establish a snap point on a waveform in the Smith chart, right-click on a waveform and choose *Snap* from the popup. A red cross-like icon appears on the waveform line (Figure 3-11) and follows the movement of your mouse. To exit s nap mode, right-click and select *Unsnap* or press the `Esc` key.

Figure 3-11 Snap Point on a Smith Chart



When you open a Smith chart, the mode display tree lists the elements contained in the chart:

- Title
- Subtitle
- Resistance Grid
- Conductance Grid
- Susceptance Grid
- Degree Grid
- Polar Grid
- Legend

Displaying Multiple Waveforms

Single Waveforms

You can display more than one waveform plot at a time in order to compare and contrast. By adding the file to the waveform library list, you can display several waveforms simultaneously or switch back and forth to display different combinations.

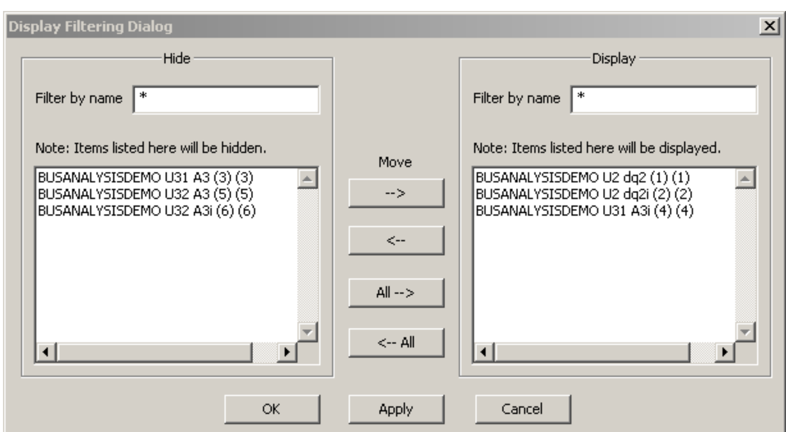
Complex Waveforms

All simple waveforms of a complex waveform can be viewed simultaneously. However, the y-axis range of data between format modes (for example, phase/magnitude) can differ to the point that simultaneous viewing may not be useful since one waveform has a distinct shape and the other is a flat line. In this situation, you can normalize the waveforms using the *Normalize* option in the Grid Scale menu while in Cartesian mode.

Filtering Waveforms

For simulation displays that contain very large numbers of waveforms, you may find it beneficial to display or hide individual waveforms using the Display Filtering dialog box.

Figure 3-12 Display Filtering Dialog Box

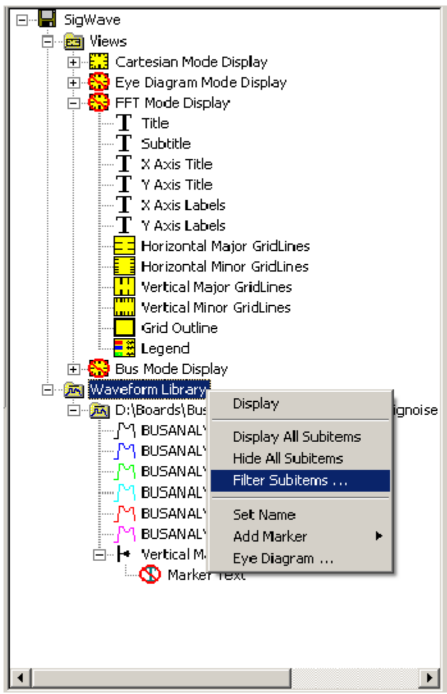


This dialog lets you scroll through a list of all the waveforms (subitems) in your simulation and select which ones you would like to display or hide.

Using the Display Filtering Dialog

1. Access this dialog box by clicking the right mouse button menu that is configured for waveform libraries or individual simulations, as shown in Figure 3-13.

Figure 3-13 Filter Subitems Selection



Opening the dialog box at the Waveform Library displays the list of `.sim` files in the library. Opening the dialog at the `.sim` file level displays all the waveforms for that simulation.

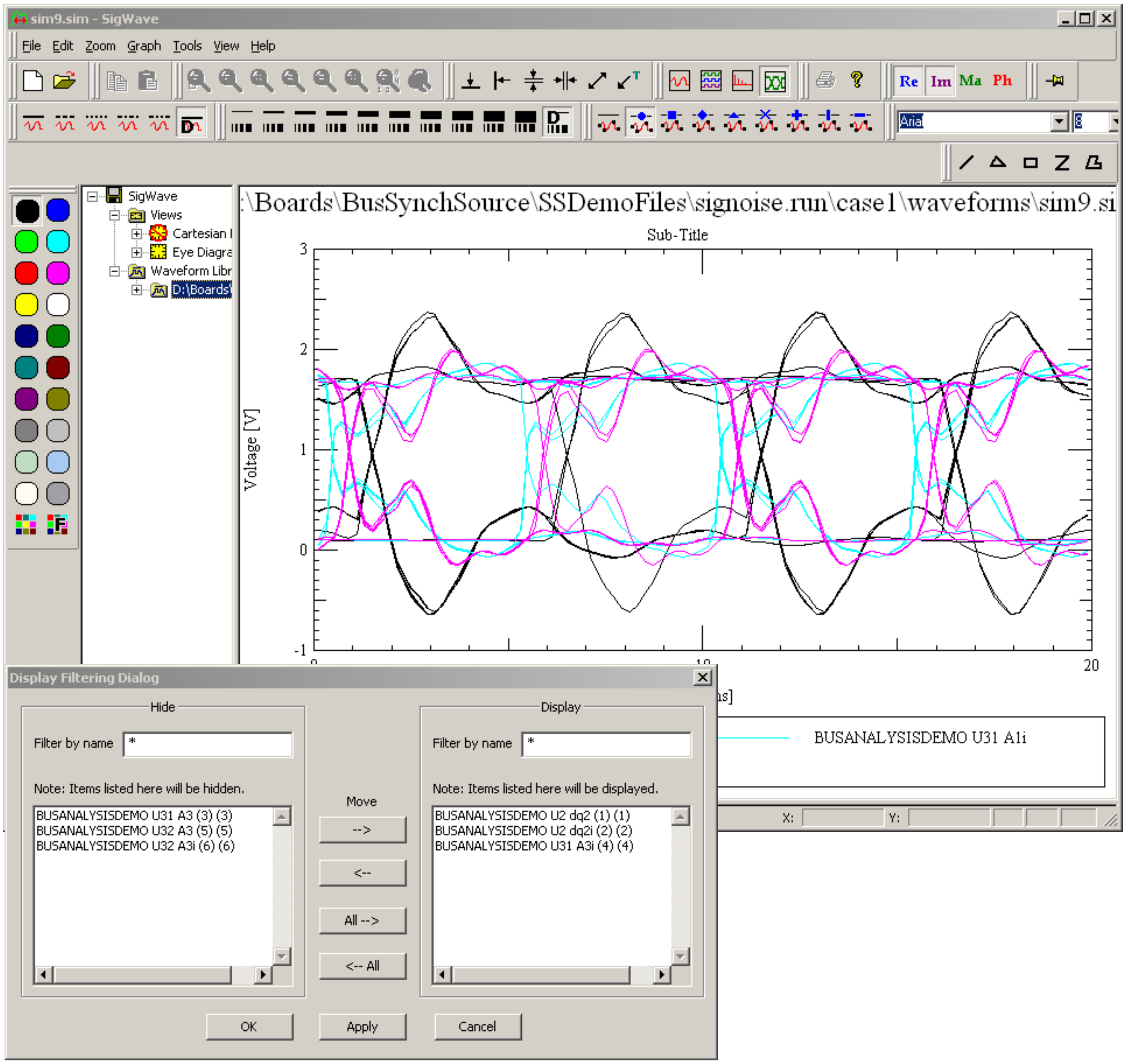
2. To select a filtered list of `.sim` files or waveforms, enter an appropriate string in the *Filter by name* field. Search selection supports standard wildcard characters.

3. Double-click the selected `.sim` or waveforms (or click *All*) to move them from one list window to the other.

4. When you have set up to display or to hide the desired `.sim` files and/or waveforms, click *OK*.

   The results are displayed in SigWave, as shown in the following figures.

Figure 3-14 Multiple Waveform Display



Performing Fast Fourier Transforms (FFT)

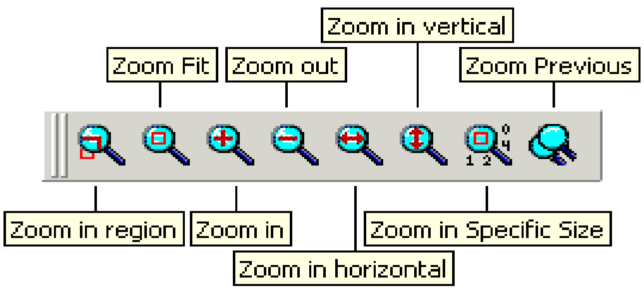
You set the parameters for performing Fast Fourier Transforms with the FFT Preferences dialog box.

Note: For a fuller explanation of Fast Fourier Transforms (FFT) and descriptions of how SigWave performs these operations on simulation waveforms, see Appendix A, Understanding Fast Fourier Transforms.

Panning and Zooming

To adjust your view of the waveform plot, you can zoom in and out using the buttons on the Zoom toolbar. SigWave provides a number of zooming functions, as shown in the figure below. You can also zoom by clicking the Zoom menu and choosing the appropriate command.

Figure 3-15 The Zoom toolbar with ToolTip descriptions



When you are zoomed in to a segment of the waveform, you can use the Pan Overview window to pan the zoomed view.

Hiding Objects

You can choose to hide certain objects, or all objects, in the display. You can also redisplay them. In the Views folder, SigWave denotes hidden objects with the international "not" symbol (a red circle and diagonal line inside) as shown in Figure 3-16 on page 50.

Figure 3-16 Hidden Object in the Waveform Manager

Editing the Plot

With SigWave, you can edit a plot in various ways. You can access the most useful editing functions by clicking the right mouse button (see Editing with the Right Mouse Button menus).

Dragging a Waveform

You can drag an entire waveform across the display to realign it on a different set of gridlines. By default, SigWave does not allow dragging. Choose *Graph – Waveform Edits* to enable this functionality.

Deleting a Waveform

Choose *Edit – Delete* to remove a waveform display. SigWave removes the waveform name from the Waveform Library list.

Editing a Waveform Using the Spreadsheet

You can edit datapoint values for the waveform by editing the spreadsheet cells. When you make changes to values in the spreadsheet, SigWave will automatically make the appropriate adjustments to the waveform, and will reflect these changes in the waveform display the next time you click in the waveform window.

Editing with the Right Mouse Button menus

The right mouse button menus provide a quick and easy way to access the most commonly used editing commands. The menus are context-sensitive so different menus appear depending on the element (views, waveforms, etc.) you select. The editing functions are largely self-explanatory.

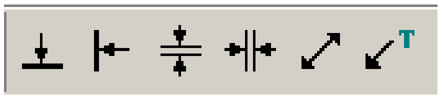
Annotating the Waveform

You can annotate a waveform in several ways. In addition to modifying the way the waveform appears, you can add text notes, add different types of markers to identify significant points on the waveform, or incorporate graphical elements to the waveform display.

Markers

Adding markers is the most common way to annotate a waveform. You can place markers using the marker toolbar in the GUI or by selecting the Add Marker option in the right button menu when you highlight a.sim file or a waveform.

Figure 3-17 Markers Toolbar in GUI



Adding

SigWave supports different types of markers. When you add a marker, SigWave displays text near the marker that provides corresponding datapoint information. The marker and its text are displayed in the same color that is used for the associated waveform.

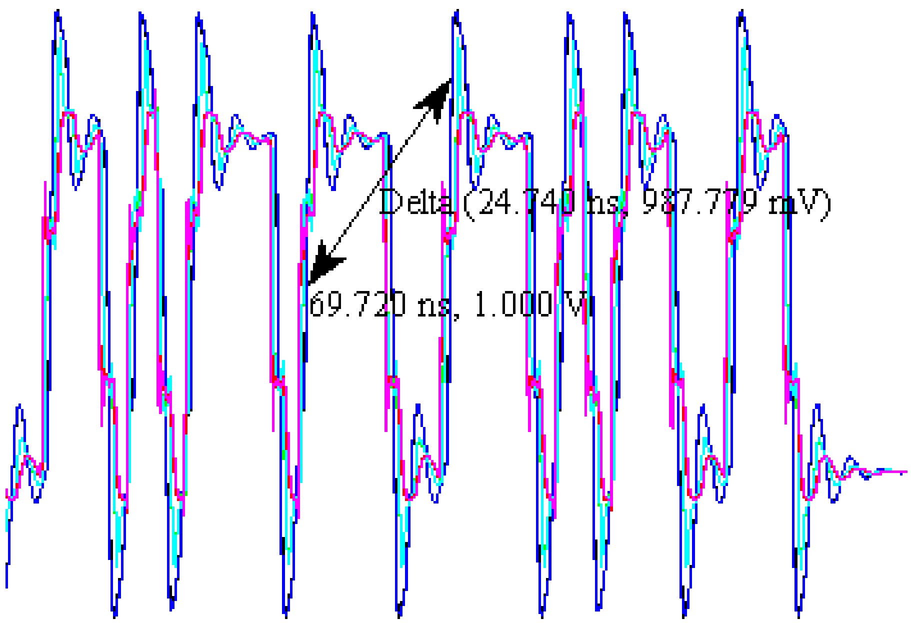
Table 3-2 on page 52 shows the types of markers you can use and the information that each displays.

Table 3-2 Waveform Markers

Marker Type	Text Information
Horizontal	The voltage.
Vertical	The current location in time (or frequency).
Differential Horizontal	Measure of the difference in the voltage between two markers.
Differential Vertical	Measure of the difference in the current location in time (or frequency) between two markers.
Angled	The start point and the delta values between the start point and end point.
Pointer Text	Highlight of the time/frequency text of the marker.
Text	User-defined note.

When you add an Angled marker, you must click a point on the waveform to establish a start-point, then move the cursor to an end-point location and click again to complete the marker. The coordinates of the start point and the delta values between the start point and end point appear at those locations. This is illustrated in Figure 3-18.

Figure 3-18 Angled Marker Display



Moving

The following tips will help avoid frustration in selecting a marker to be moved:

- When you move a vertical or horizontal marker, click on the dashed line rather than the handles when you select it.
- When you move an arrow text marker, click on the handles on either end of the marker when you select it.

Changing

By using the right mouse button menu, you can change the appearance of markers. You can change the color, line style, line width, and point style of a marker. You can also toggle the display of the marker on and off.

Deleting

You can delete markers by choosing the *Edit – Delete* command, or by choosing Delete from the right mouse button menu.

Using Snap Mode

Snap mode snaps the marker to the time or voltage points on the waveform.

Units and Labels

Markers inherit the units and labels of their parent item in the Waveform Library. The parent item is determined by the currently selected item at the time the marker is created. The marker item appears below its parent in the tree.

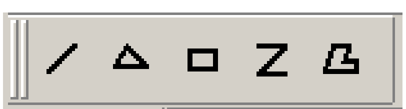
Adding Text

You can add and edit text elements anywhere on the waveform display. You can also edit the waveform title as well as the X and Y axis titles.

Adding Graphics

You can add rudimentary graphics to the waveform display using the drawing toolbar in the main GUI. Graphic functionality lets you easily append visual aids to waveforms.

Figure 3-19 Drawing Toolbar



You can add the following graphic elements to a waveform:

- Lines
- Triangles
- Rectangles
- Polylines
- Polygons

You can save graphic elements that you add to a waveform display by way of a Workspace (.sww) file. You can then import other simulation (.sim) files to it and overlay the graphic elements onto another waveform.

To add graphics

1. Click the desired graphic icon in the drawing toolbar (shown above).

    a. To draw a line, click on the start point within the waveform display and move the mouse to the end point, then double-click.

    b. To draw a polyline or a polygon, click on each point in sequence and double-click on the last point.

    c. If you select a rectangle or triangle, one will appear in the waveform display. You can reposition or reshape the graphic by dragging the edge or vertex of the shape.

    The selected icons appear in the tree view in the left-side pane.

2. To reselect any graphic for editing or to delete it, right-click on the icon in the tree view to display the pop-up menu. To delete a drawing before you complete it, click the Escape key or the appropriate icon in the toolbar to clear it.

To save graphics and overlay them onto another waveform

1. Once your graphic is drawn in the waveform display, select *File – Save As*.
2. In the Save As window, save the `.sim` file as a Workspace File (`.sww`) and assign a file name to it.
3. Select *File – SigWave or Touchstone File* and select a `.sim` file for import.

    The imported waveform will be displayed with the graphic elements overlaying it.

Viewing the Buffer Model

From SigWave, you can open the IOCell Editor to display the IOCell model for the selected waveform. This gives you detailed information about the model parameters used for the simulation. To view the buffer model, select the waveform, then click-right and choose *View Buffer Model*.

Note: When viewing buffer models, the option to Allow Parent Change (accessible from the right mouse button menu) will not function unless you initially loaded the `.sim` file directly from Allegro SI or SigXplorer.

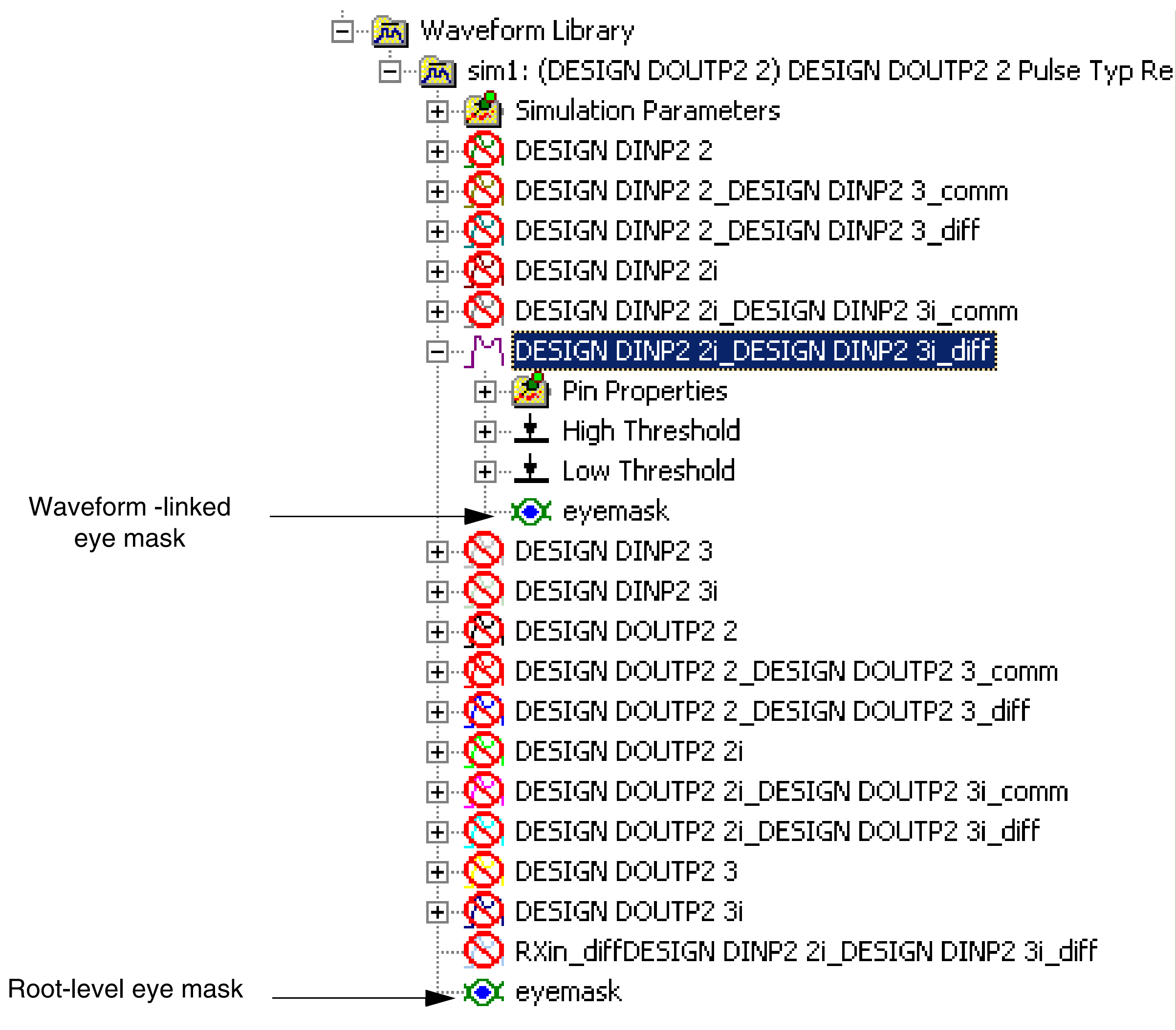
You can edit the model parameters and thereby modify the waveform. Any changes you make to the model in the IOCellEditor is also reflected in subsequent simulations you run with Allegro SI.

Using Eye Masks

Eye masks let you specify the acceptable parameters for what an eye should look like in order to extract clock transmissions and high-speed data to buffer models. SigWave lets you create eye masks that you can save in `.sim` files and view/edit when you display the waveform in Eye Diagram mode. Eye mask functionality in SigWave operates under the following conditions:

- Eye masks that you create in a simulation file reside at the root of the waveform library. These "root-level" eye masks are linked to the waveforms they are attached to, as shown in Figure 3-20.

Figure 3-20 Root-Level and Linked Eye Mask



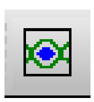
You can attach the same root-level eye mask to multiple waveforms. If you delete linked eye masks from a wave form, the root-level eye mask remains; however, if you delete the root-level eye mask, all iterations of the linked eye masks will also be deleted.

- You can attach only a single eye mask to a waveform; if you attempt to attach a second eye mask to a waveform, the second eye mask replaces the first.

Creating and/or Loading an Eye Mask

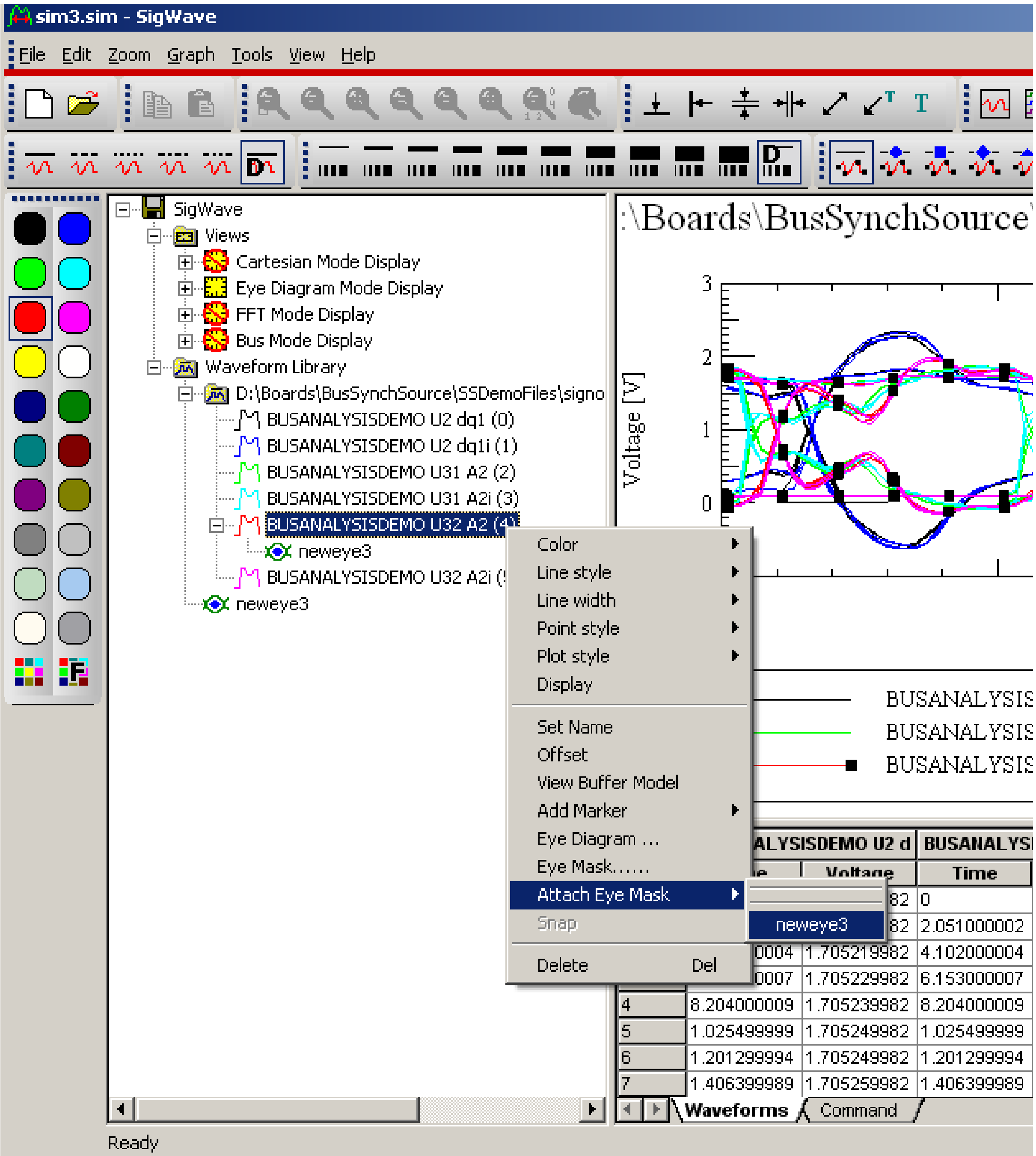
You create a new eye mask or load an existing one into SigWave by way of the Eye Mask toolbar icon, or by selecting a waveform and clicking the right mouse button to display the context pop-up menu.

Figure 3-21 Eye Mask Toolbar Icon



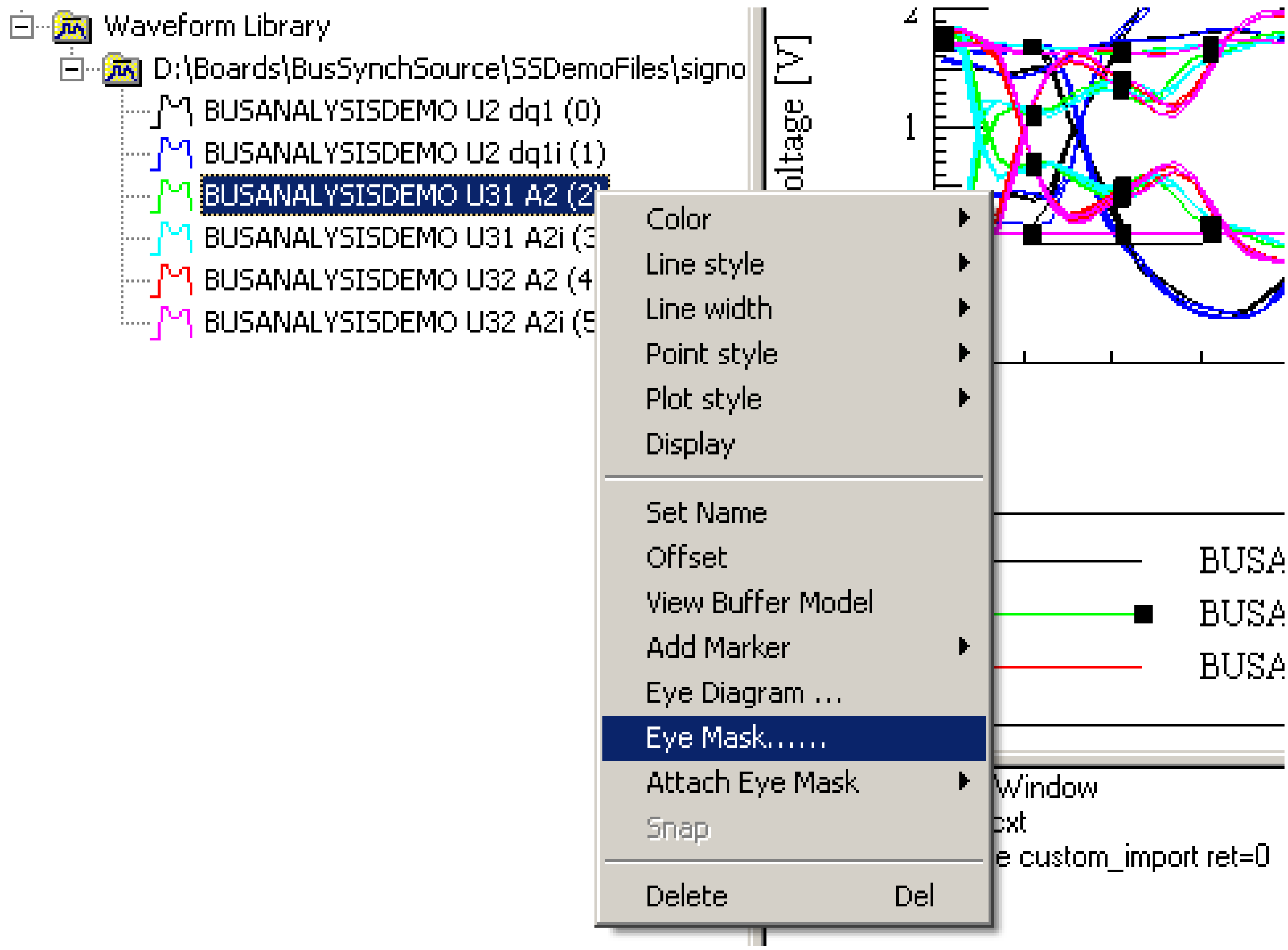
When you launch the Load/Create Eye Mask dialog box from the toolbar icon, the mask you create attaches itself to the waveform library of your `.sim` file, not to an individual waveform. You can attach an eye mask residing in the waveform library to an individual waveform by highlighting a waveform and selecting the *Attach Eye Mask* option from the right-button pop-up menu. You can then choose an eye mask from a list of the eye masks in the `.sim` file. This is depicted in Figure 3-22.

Figure 3-22 Attach Eye Mask Option



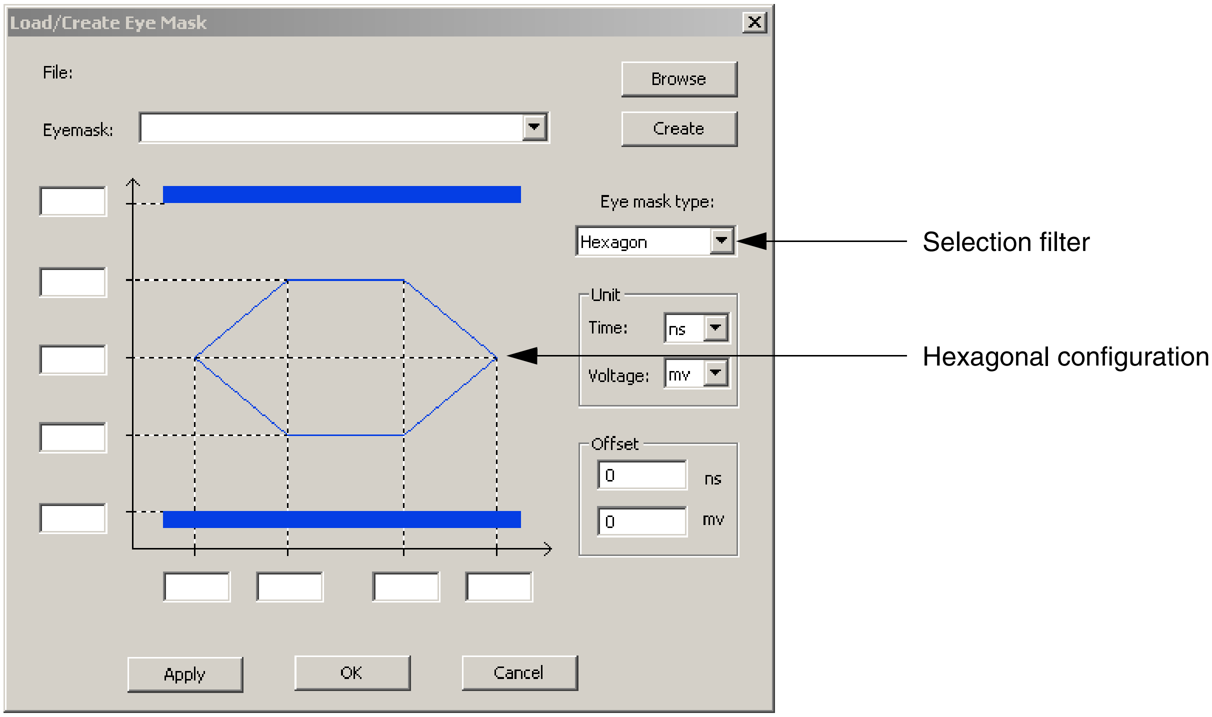
When you launch the Load/Create Eye Mask dialog box from *Eye Mask* option in the right-button pop-up menu (as shown in Figure 3-23), the mask will be created at root-level and a linked version of the eye mask will be attached to the highlighted waveform.

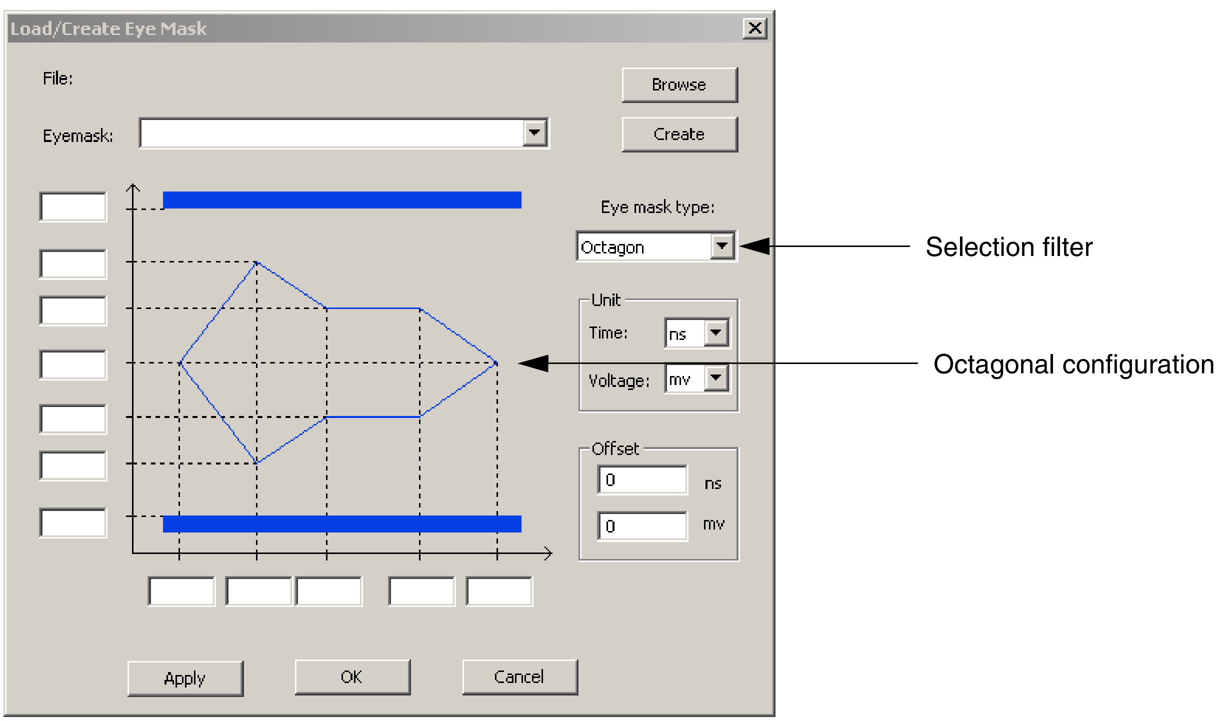
Figure 3-23 Context-Sensitive Pop-Up Menu Attached to Waveform



Eye mask configurations can be either hexagonal or octagonal. You select the appropriate configuration from the Load/Create Eye Mask dialog box, as shown in Figure 3-24.

Figure 3-24 Load/Create Eye Mask Dialog Boxes





You will configure the parameters of an eye mask that you either *create* or *load* from an eye mask library, as shown in Figures 3-25 and 3-26.

Figure 3-25 Create Eye Mask

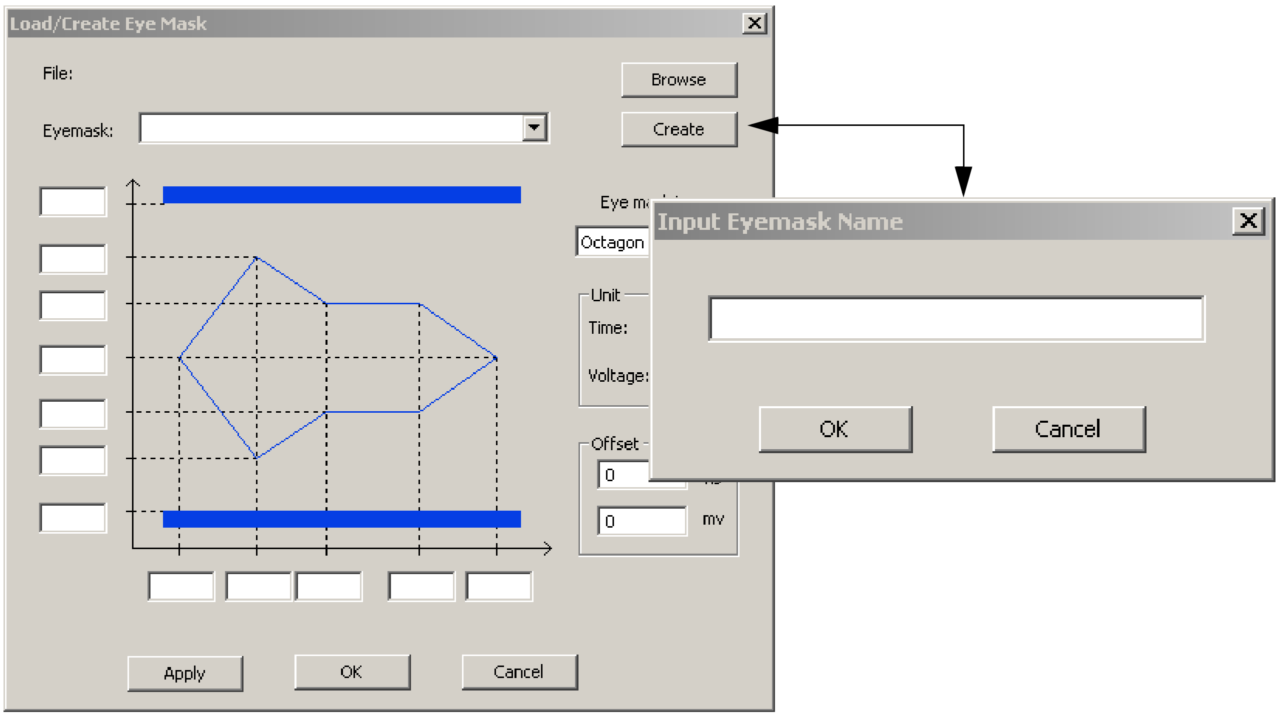
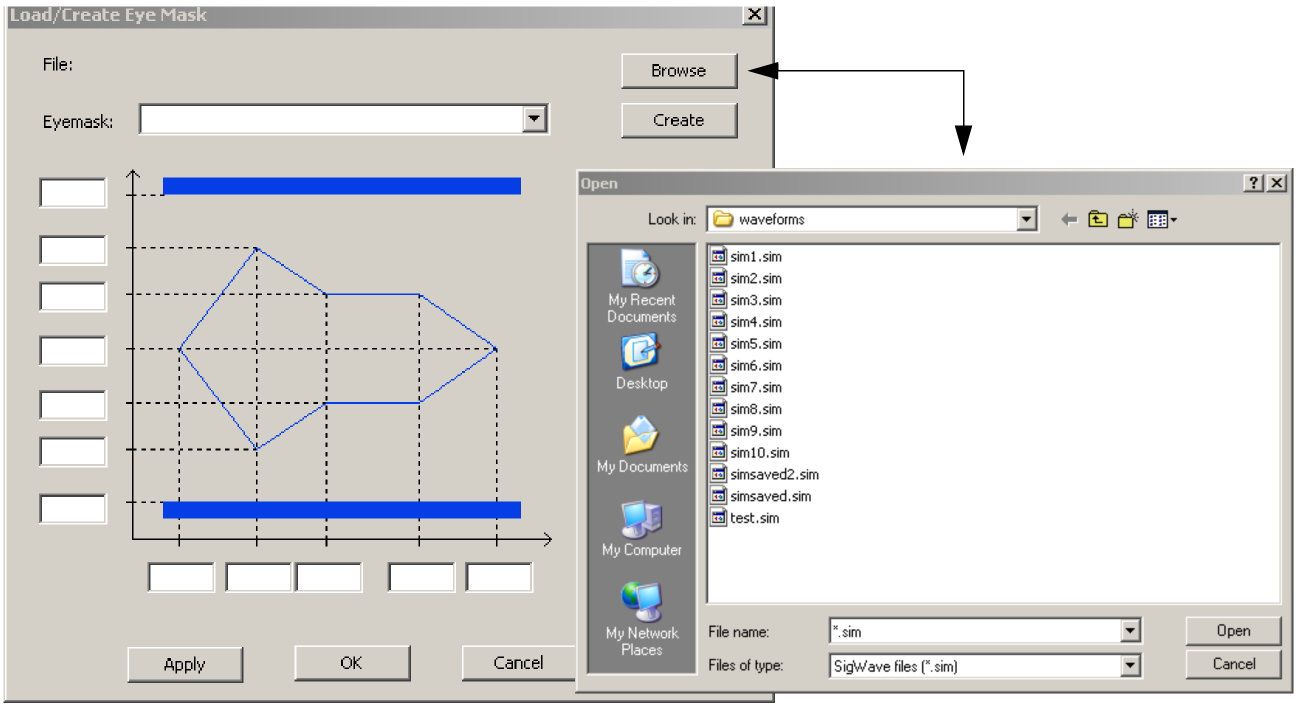


Figure 3-26 Load Eye Mask



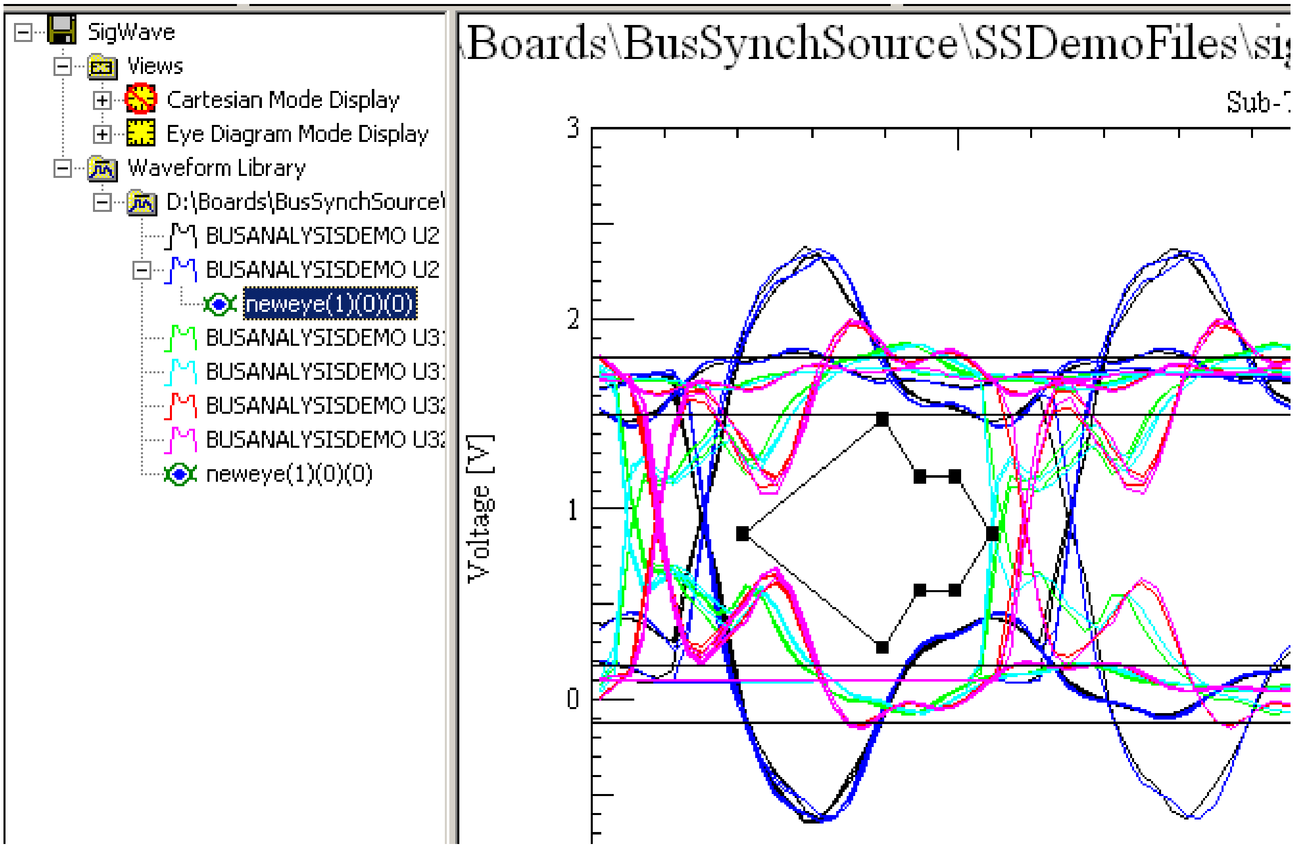
When you load an existing eye mask from the eye mask library, you can edit all the parameter settings *except* the hexagon/octagon eye mask type.

The control fields in the dialog box let you specify the eye mask point values in the vertical (time) and horizontal (voltage) dimensions. You can also offset the eye mask by specifying time and voltage values. Time measurement units are selectable in picoseconds (ps), nanoseconds, (ns) and unit intervals (UI). Voltage measurement units are selectable in volts (v) and millivolts (mv).

Note: The parameter settings in the dialog box are dynamic; that is, if you move the eye mask with your cursor to another point in the waveform, the settings in the dialog box will have changed to reflect the new position when you reopen it.

Figure 3-27 depicts an octagonal eye mask in a waveform.

Figure 3-27 Octagonal Eye Mask in Waveform



Editing an Eye Mask

You can edit the parameters of an existing eye mask through the use of the Edit Eye Mask toolbar icon (Figure 3-28) or by right-clicking on a selected eye mask to display the context-sensitive pop-up menu (Figure 3-29).

Figure 3-28 Edit Eye MaskToolbar Icon

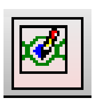
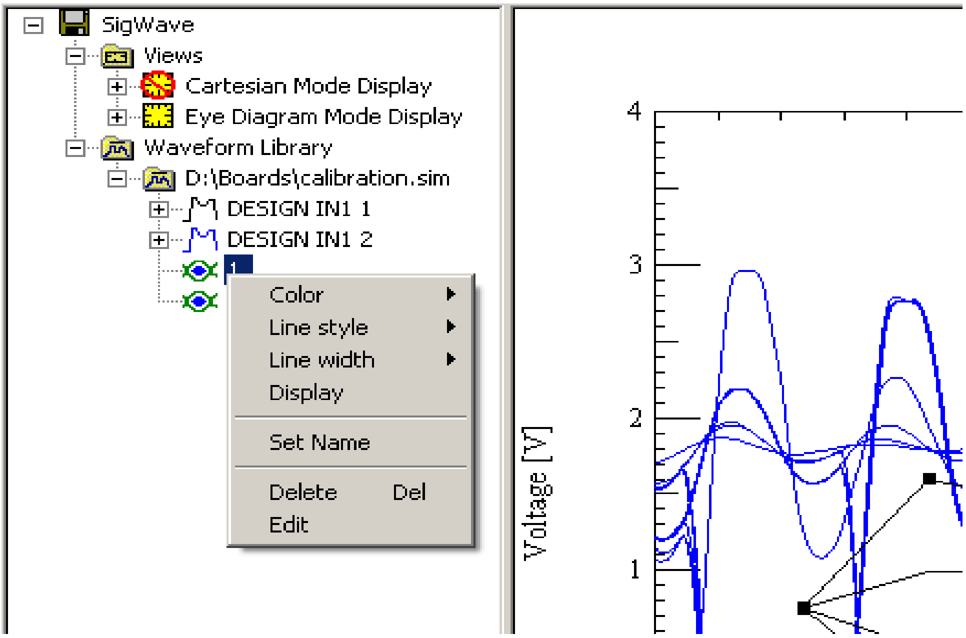
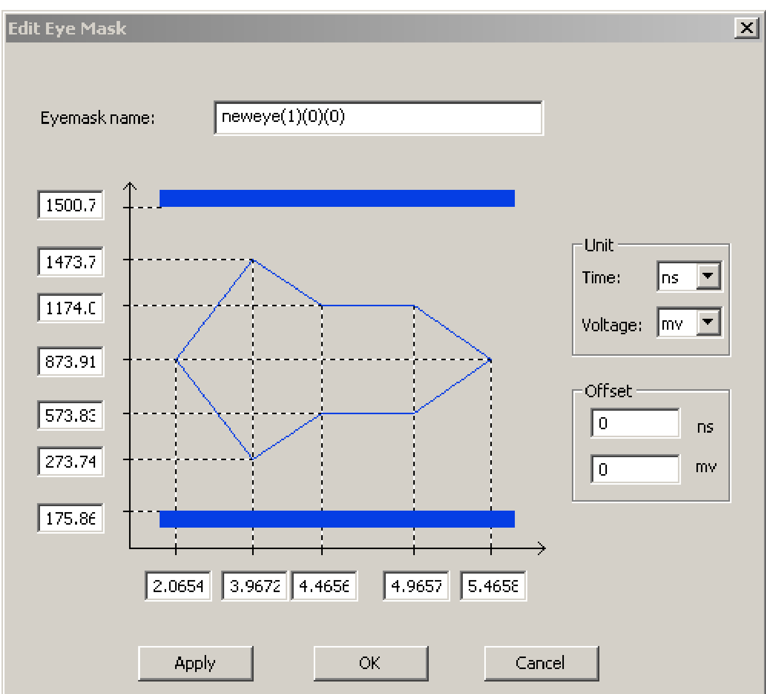


Figure 3-29 Context-Sensitive Pop-Up Menu for Editing Eye Masks



Either operation opens the Edit Eye Mask dialog box, shown in Figure 3-30.

Figure 3-30 Edit Eye Mask Dialog Box



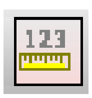
In addition to editing the parameters of the selected eye mask by way of the Edit Eye Mask dialog box, you can change the display of it through the right-button pop-up menu (shown in Figure 3-29). The options on the pop-up menu let you:

- Change the color of the eye mask outline
- Display/hide the eye mask
- Re-name the eye mask
- Export the eye mask data to a .sim file or a DML file
- Delete the eye mask

Displaying Eye Diagram Measurement Values

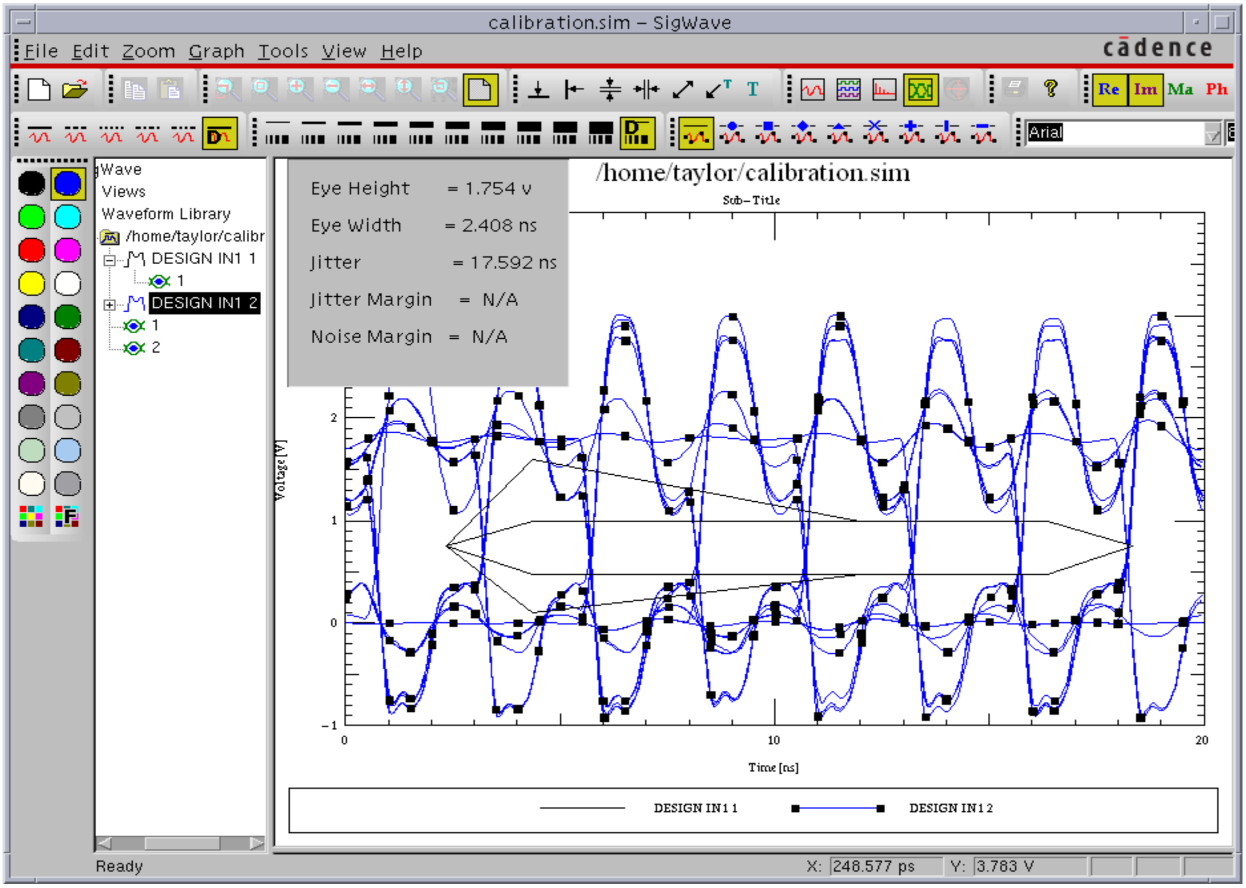
You can display the measurement values for eye diagrams containing an eye mask by turning on the Eye Measure Window on the tool bar (Figure 3-31).

Figure 3-31 Eye Measure Window Icon



When you toggle it on, eye diagram measurement values for selected waveforms are displayed on the waveform canvas. The measurement parameters that are displayed are dependent on the type of waveform you have selected. For example, Figure 3-32 illustrates a measurement display for a selected waveform simulated by SigXplorer.

Figure 3-32 Normal Measurement Values



The following table lists the measurement parameters for supported waveforms types:

Waveform Type	Measurement Parameters
Normal	Eye Height Eye Width Jitter Jitter Margin Noise Margin

Bus	Eye Height Eye Width Apt Width Jitter
Channel Analysis	Eye Height Eye Width Jitter Margin Noise Margin

DML Support for Eye Masks

Allegro's default DML buffer models support standardized mask definitions that you can modify for your specific uses (as shown in the DML sample, below). When models containing such data are assigned to design elements in Allegro high-speed products and run in simulations, the eye masks are displayed in waveform eye diagrams in SigWave. Additional examples of eye masks in DML files can be found in your installation hierarchy at `//share/pcb/examples/`.

Sample DML file with eye mask section

```
("sample.dml"
 (IbisIOCell
  (pci_xp_in
   (LogicThresholds
    (Input
     (High
      (maximum 0.0875 )
      (minimum 0.0875 )
      (typical 0.0875 ) )
     (Low
      (maximum -0.0875 )
      (minimum -0.0875 )
      (typical -0.0875 ) ) ) )

   (EyeMasks ; There can be multiple eye masks for different data rates.
             ; In this example eyemask is defined nominally for only one data
             ; rate.  The simulation software will use the eyemask nearest
             ; to its operating data rate.
     (DataRate
       (5e10 ; Data rate is bits per second.
         (Matrix ; Eye mask is a 3 column matrix.
                 ; The first column is dimensionless unit interval.
                 ; The second column is upper and the third is lower curve.
                 ; The upper curve and lower curve gives the voltage values.
```

```
                 ; The upper and lower are referenced to the midpoint
                 ;     voltage, which may be called vmeas.
           "0.3   0      0
            0.35  0.05  -0.05
            0.65  0.05  -0.05
            0.7   0      0"
         )
       )
     )
   )
   ;(ami  (chffefilt "/hm/ckumar/projects/test/dll_pak/work" ) (chcdr "/hm/ckumar/projects/test/
dll_pak/work") )
   (MacroModel
    (MacroType TDiffIO )
    (NumberOfTerminals 8 )
    (Parameters
     (MinTypMaxParams
      (rt 50 )
      (dfeOn 0 )
      (bitp 100p ) ) )
    (SubCircuits "
*  The comment character inside the SubCircuits double quotes is a *.
*
*  Handy bdrvr node cheat sheet:
*      power = 1
*      output = 2
*      ground = 3
*      input = 4
*      enable = 5
*      power clamp_reference = 6
*      ground clamp reference = 7
*
* ==========================================================================
.subckt pci_xp_in 1 2 3 4 5 6 7 8
+ padcap=2p
+ rt=50
+ dfeOn=0
+ bitp=100p

.param rxckt='if (dfeOn > 0) (1) else (2)'

x 1 2 3 4 5 6 7 8 rxckt
.subckt 2 1 2 3 4 5 6 7 8
rp 2 1 rt
rn 8 1 rt
cp 2 3 'padcap'
cn 8 3 'padcap'

erx rx 0 v='v(2,8)'

.node_param rx name=(RXin_diff name(2) _ name(8) _diff) print
```

```
.ends 2

.subckt 1 1 2 3 4 5 6 7 8

x 1 2 3 4 5 6 7 8 dfe39_rx bitp=bitp
.subckt dfe39_rx nvdd outp ngnd in en  pcl  gcl  outn bitp=100p
+ ff0=-0.377077 ff1=2.06626 ff2=-0.82244
+ cf2=-0.365234 cf3=0.0271399 cf4=-0.12127 cf5=0.0572522 cf6=0.0139165 cf7=-0.0515211 cf8=-0.0118379
cf9=-0.0127641 cf10=-0.0182494
+ padcap=2p
+ rt=50

* cf0 and cf1 are forward coefficients

cp outp ngnd 'padcap'
cn outn ngnd 'padcap'
rp outp nvdd 'rt'
rn outn nvdd 'rt'

* rx is the output node
erxin rxin ngnd v='v(outp, outn)'
.node_param rxin name=rxin print

erxdc rxdc ngnd v='((ff0 + ff1+ff2) * v(rxin, ngnd)) /(1 - cf2 - cf3 - cf4 - cf5 -df6 -cf7 -cf8 -cf9-
cf10)'
rrxdc rxdc ngnd 1e6

xrx1 rxin rx1 ngnd delayin del='bitp'
xrx11 rxin rx11 ngnd delayin del='2*bitp'
xrx2 rx rx2 ngnd delayin del='1*bitp'
xrx3 rx rx3 ngnd delayin del='2*bitp'
xrx4 rx rx4 ngnd delayin del='3*bitp'
xrx5 rx rx5 ngnd delayin del='4*bitp'
xrx6 rx rx6 ngnd delayin del='5*bitp'
xrx7 rx rx7 ngnd delayin del='6*bitp'
xrx8 rx rx8 ngnd delayin del='7*bitp'
xrx9 rx rx9 ngnd delayin del='8*bitp'
xrx10 rx rx10 ngnd delayin del='9*bitp'

erx rx ngnd v='if (time <= 0) (v(rxdc, ngnd)) else (ff0 * v(rxin, ngnd) + ff1 * v(rx1, ngnd) + ff2 *
v(rx11, ngnd) + cf2 * v(rx2, ngnd) + cf3 * v(rx3, ngnd) + cf4 * v(rx4, ngnd) + cf5 * v(rx5, ngnd) + cf6
* v(rx6, ngnd) + cf7 * v(rx7, ngnd)+ cf8 * v(rx8, ngnd))+ cf9 * v(rx9, ngnd)+ cf10 * v(rx10, ngnd)'

.node_param rx name=(RXin_diff name(outp) _ name(outn) _diff) print

.subckt delayin in1 in2 ngnd del=0
ein1 in2 ngnd pwl in1 ngnd delay=del
datapoints vv
0 0
1 1
end vv
r in2 ngnd 1e6
```

```
.ends delayin




.ends dfe39_rx


.ends 1
.ends pci_xp_in
" ) )
   (Model
    (C_comp
     (maximum 0 )
     (minimum 0 )
     (typical 0 ) )
    (ModelType Input ) )
   (Technology CMOS )
   (VIReferenceTemperature
    (maximum 100 )
    (minimum 0 )
    (typical 25 ) )
   )
  )
 )
```

Printing the Waveform or Spreadsheet

SigWave provides standard options and setup parameters for printing the waveform. You can also print the spreadsheet data if you have the spreadsheet displayed.

Note: Be sure to click in the appropriate window (Waveform or Spreadsheet) to make that the active window before using the print commands. If the cursor is in the Waveform window, the waveform will be printed. If the cursor is in the Spreadsheet window, the spreadsheet will be printed. The print function does not allow you to print both the waveform and the spreadsheet together. Always use *Print Preview* to confirm what you will be printing.

Exporting the Waveform

You can export the waveform data you edit in several ways. You can save the waveform graphics and annotation information in a variety of graphical file formats. Or, you can save the spreadsheet data so that it can be used in another spreadsheet program.

Note: To save the simulation file, see Saving the Waveform File.

Saving the Plot Graphics

SigWave allows you to save the plot graphics in the following file formats:

- Windows bitmap (.bmp)
- JPEG (.jpg)
- Targa bitmap (.tga)
- TIFF (.tif)

The graphic images of the plot can be useful in documenting the circuit analysis you are performing. They can also be used as a record of your work in progress as you experiment with different parameters in the simulation.

Saving the Spreadsheet Data

You can export the spreadsheet data from SigWave independently and use this as input to another spreadsheet editor. The spreadsheet data is saved as a tabbed text file (.txt).

Note: You do not need to have the spreadsheet view displayed in order to save the spreadsheet data.

Saving the Waveform

Saving the Waveform File

You can save the waveform file along with any changes you may have made to the simulation data. You can then reload this file later into SigWave to view it again, or to make further modifications. By saving the waveform file, you preserve any data changes you have made while working in SigWave.

When you first simulate with Allegro SI, the simulation (`.sim`) file is stored in the default subdirectory *signoise.run/case/waveforms.* If you save a file from SigWave, you can choose a different subdirectory to store the file. In order to preserve your work and manage the various files correctly, you should avoid overwriting the original simulation file in the default subdirectory.

Note: To save the waveform image along with the graphical annotations you may have added, you must save by using the *File – Export – Image* command (see Saving the Plot Graphics).

Translating the Waveform file to Analog Workbench format

SigWave provides a means of translating the SigNoise waveform file (`.sim`) into the file format (.wave) recognized by the Analog Workbench tool. This allows you to pass the data to Analog Workbench, where you can perform additional circuit simulation and analysis. To perform the translation, first locate the `.sim` file. (SigNoise writes waveform `.sim` files whenever simulation is invoked. The `.sim` files appear in the appropriate case subdirectory of the analysis directory.) Then, enter the sigwave2awb command in the following syntax:

`sigwave2awb filename.sim`

where *filename* is the name of the `.sim` file you want to translate.

The output from the sigwave2awb command is one or more files with the following name:

`waveforms.sim.spice_node_number.wave`

Note:

- The *spice_node_number* in these file names comes from the beginning of the `.sim` file. A `.sim` file contains voltage versus time data for multiple nodes that are probe points in the circuit. These nodes are the driver pin and the receiver pin.
- Nodes ending with `p` are external nodes representing the point where the package pin meets the interconnect. Nodes with no letter suffix are for the internal point where the lead wire attaches to the die in an integrated circuit. SigNoise uses internal node measurements for all data in the reports.

A

Understanding Fast Fourier Transforms

This appendix provides an explanation of Fast Fourier Transforms (FFT) and describes how SigWave performs these operations on simulation waveforms.

This appendix discusses the following:

Discrete Fourier Transforms

Discrete Fourier Transform (DFT) takes as input a time domain signal and converts it to its FFT representation. Given a periodic sequence $x_p(n)$ with a period of N samples, its Discrete Fourier Transform is:

$$x_p(\mathbf{k}) = \sum_{n=0}^{N-1} x_p(\mathbf{n})\mathbf{e}^{-j(2\pi/N)n}$$

If $x_p(n)$ is a time sequence, then $x_p(k)$ contains magnitude and phase information over a sequence of frequencies.

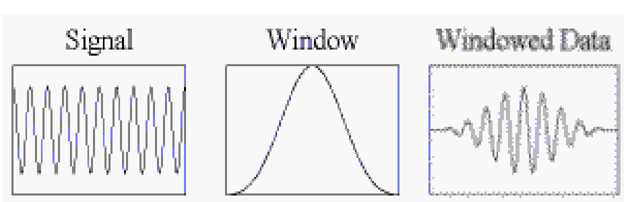
Fast Fourier Transform (FFT) is a computationally fast version of the DFT.

How Window Functions Work

A discontinuity in the levels of the beginning and ending points of the signal broadens the resulting Fast Fourier Transform (FFT) spectrum. You can minimize the discontinuity by applying a window function or eliminate the discontinuity by cropping the input data with the user-defined fields in the Window Extents section of the FFT Preferences dialog box.

A window function is a curve, usually bell-shaped, that is multiplied by the time domain data to minimize discontinuities. The figure below shows how a window function is applied. The signal (which in this case is a sine wave) is multiplied by the window function, resulting in the windowed data. The FFT is then taken on the windowed data. Note how the edges of the window function go to zero. In this way, if the signal period is not an integer multiple of the sampling period, the resulting edge discontinuity is minimized.

Figure A-1 The FFT window functions



Different window functions have different applications. Generally, to resolve a large signal next to a small signal, use a window function with a large sidelobe suppression. To reduce mainlobe width or gain frequency resolution, use a window function with a smaller sidelobe suppression.

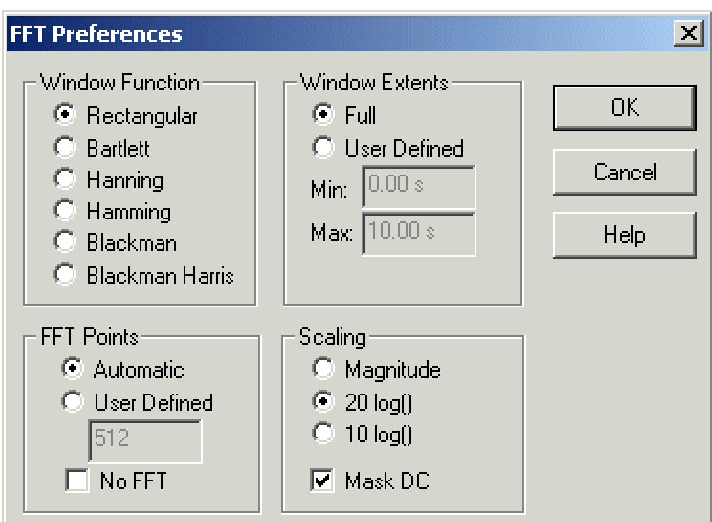
The following table summarizes the trade-offs for each of the six window functions.

Table A-1 The FFT window function descriptions

Window Function	Highest Sidelobe (dB)	Noise BW (Bins)	3dB BW (Bins)	6dB BW (Bins)
Rectangular	-13	1.00	0.89	1.21
Bartlett	-27	1.33	1.28	1.78
Hanning	-23	1.23	1.20	1.65
Hamming	-43	1.36	1.30	1.81
Blackman	-58	1.73	1.68	2.35
Blackman Harris	-92	2.00	1.90	2.72

You set the parameters for performing Fast Fourier Transform with the FFT Preferences dialog box.

Figure A-2 The FFT Preferences dialog box



Using the Rectangular Window Function

The rectangular window corresponds to directly truncating the time domain series.

The weighting function used by the rectangular window is:

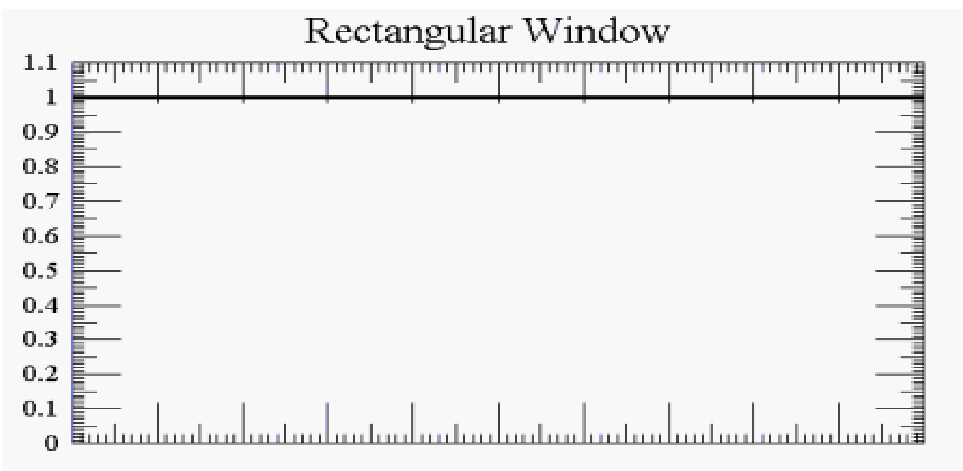
$$w(n) = \begin{cases} 1 \cdot 0 & -\left(\frac{N}{2}\right) \le n \le \frac{N}{2} \\ 0 \cdot 0 & \text{elsewhere} \end{cases}$$

where

N is the number of points.

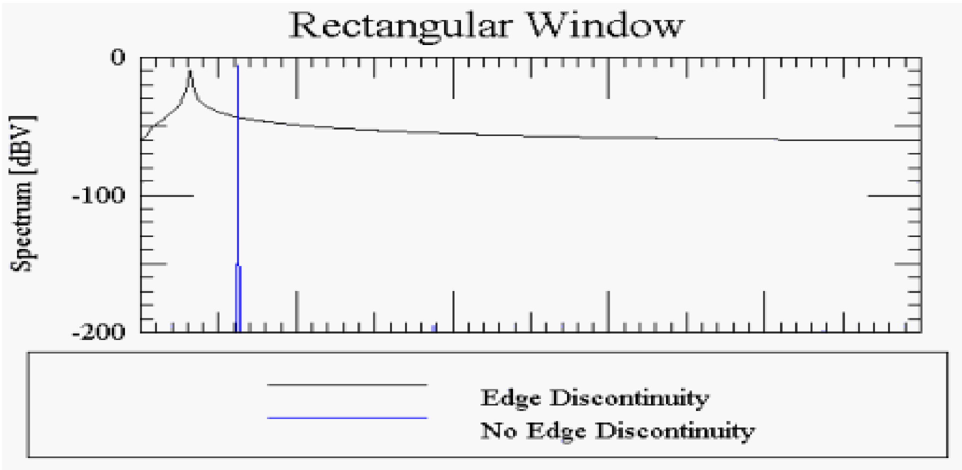
n is the sample value

Figure A-3 The Rectangular window function in the time domain



Note: In the following figure, the sinusoid that falls in the center of an FFT bin is a vertical line. The other sinusoid has a much broader spectrum. Compare this FFT to those of other window functions to see how using a different window function minimizes the disparity between the two window functions.

Figure A-4 FFT using the Rectangular window function



Using the Bartlett Window Function

The Bartlett window is triangular in shape.

The weighting function used by the Bartlett window is:

$$w(n) = \begin{cases} 1 - \dfrac{abs(2n - N + 1)}{N} & -\left(\dfrac{N}{2}\right) \le n \le \dfrac{N}{2} \\ 0 \quad \text{elsewhere} & \end{cases}$$

where

N is the number of points

n is the sample value

Figure A-5 The Bartlett window function in the time domain

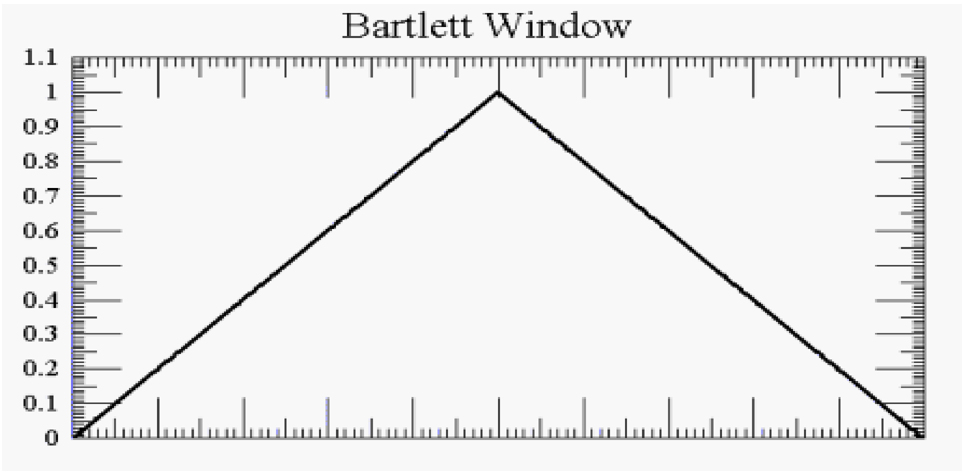
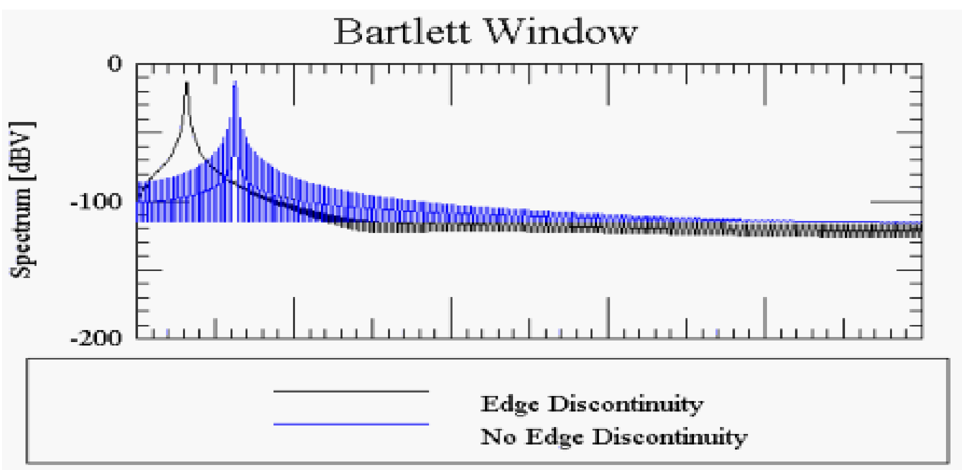


Figure A-6 FFT using the Bartlett window function



Using the Hanning Window Function

Both the Hanning window and the Hamming window have the shape of a raised cosine. These windows have twice the width of the main lobe of the frequency response of a rectangular window. Peak side lobe ripple is down about 40dB from the main lobe peak.

The weighting function used by the Hanning window is:

$$w(n) = \begin{cases} 0.5 - 0.5\cos\left(\dfrac{2\pi n}{N-1}\right) & -\left(\dfrac{N}{2}\right) \le n \le \dfrac{N}{2} \\ 0 \quad \text{elsewhere} \end{cases}$$

where

N is the number of points

n is the sample value

Figure A-7 The Hanning window function in the time domain

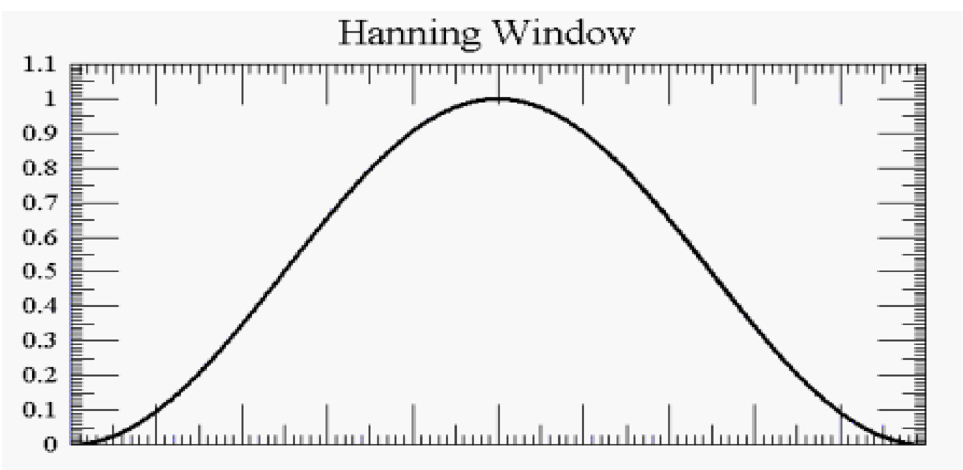
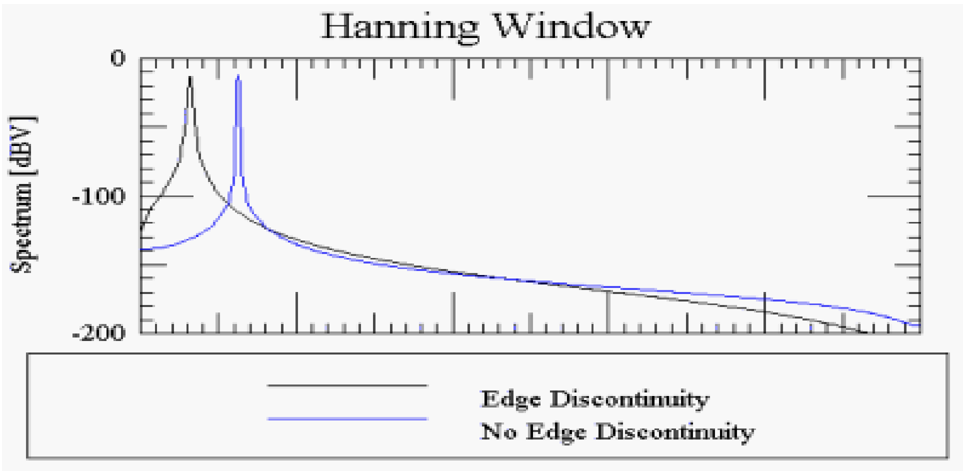


Figure A-8 FFT using the Hanning window function



Using the Hamming Window Function

See <u>Using the Hanning Window Function</u> for a description of the Hamming window.

The weighting function used by the Hamming window is:

$$w(n) = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right) & -\left(\frac{N}{2}\right) \le n \le \frac{N}{2} \\ 0 \quad \text{elsewhere} & \end{cases}$$

where

N is the number of points

n is the sample value

Figure A-9 The Hamming window function in the time domain

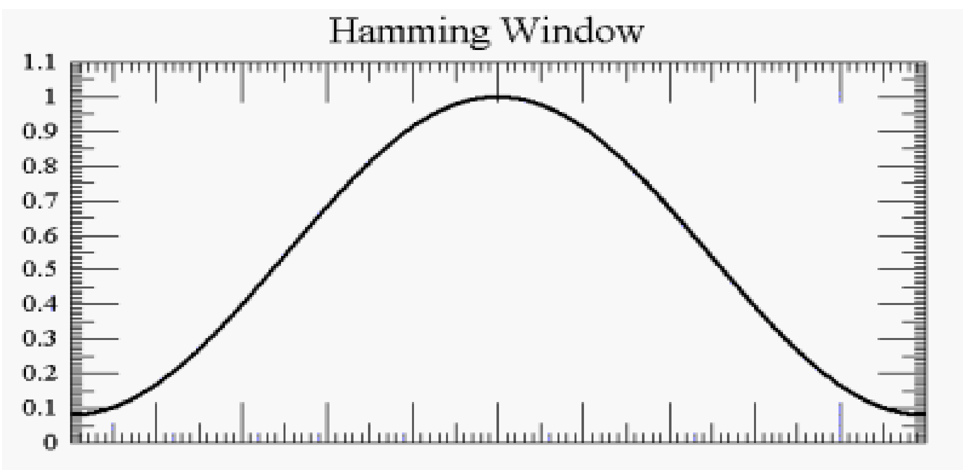
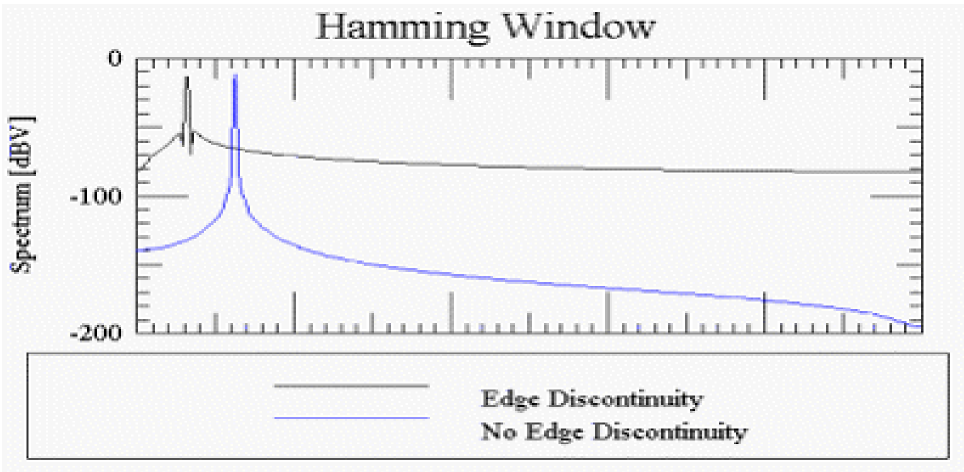


Figure A-10 FFT using the Hamming window function



Using the Blackman Window Function

The Blackman window has one additional cosine term to the Hanning and Hamming windows. This further reduces the ripple.

The weighting function used by the Blackman window is:

$$w(n) = \begin{cases} 0.42 - 0.5\cos\left(\frac{2\pi n}{N-1}\right) + 0.08\cos\left(\frac{4\pi n}{N-1}\right) & -\left(\frac{N}{2}\right) \le n \le \frac{N}{2} \\ 0 \quad \text{elsewhere} \end{cases}$$

where

N is the number of points

n is the sample value

Figure A-11 The Blackman window function in the time domain

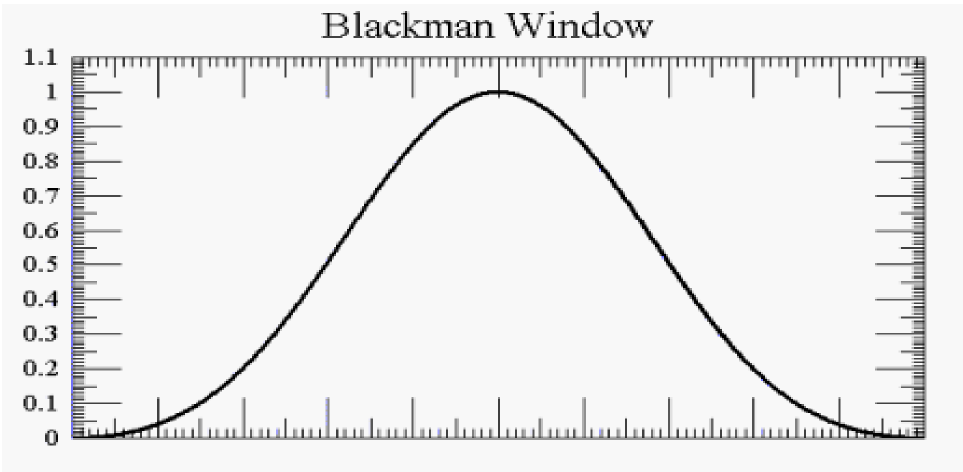
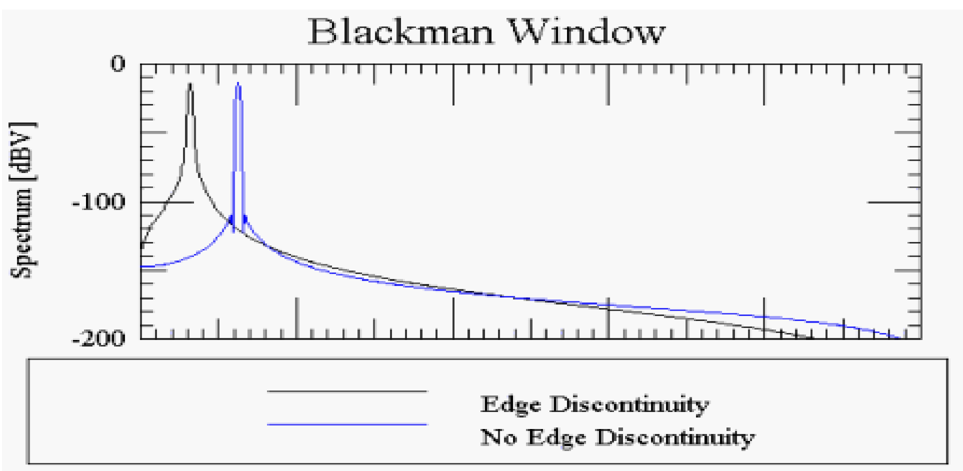


Figure A-12 FFT using the Blackman window function



Using the Blackman-Harris Window Function

The weighting function used by the Blackman-Harris window function is:

$$w(n) = 0.35875 - 0.48829\cos\left(2\pi\frac{n}{N-1}\right) + 0.14128\cos\left(4\pi\frac{n}{N-1}\right) + 0.01168\cos\left(6\pi\frac{n}{N-1}\right)$$

$$\text{for } -\left(\frac{N}{2}\right) \le \mathbf{n} \le \frac{N}{2}$$

where

N is the number of points

n is the sample value

Figure A-13 The Blackman-Harris window function in the time domain

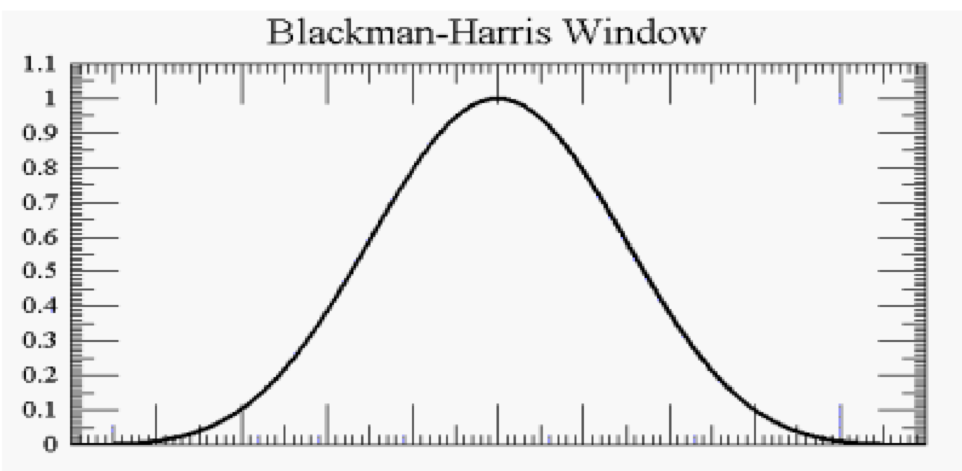
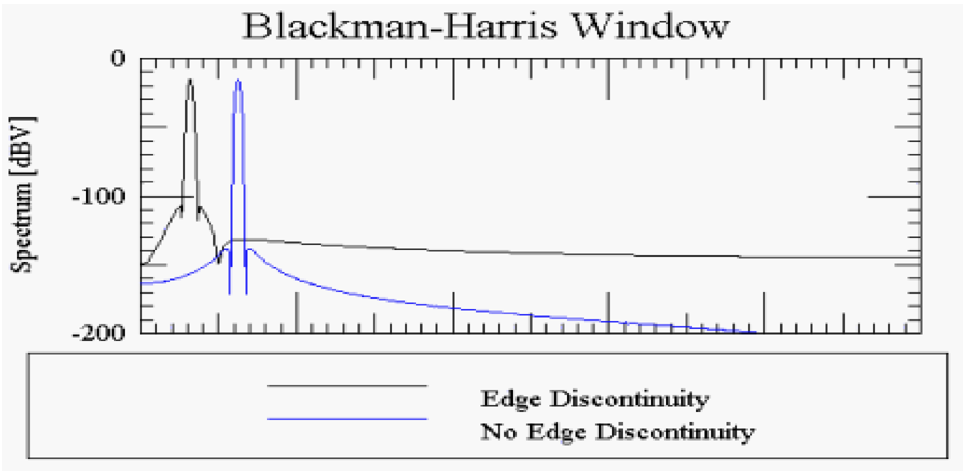


Figure A-14 FFT using the Blackman-Harris window function



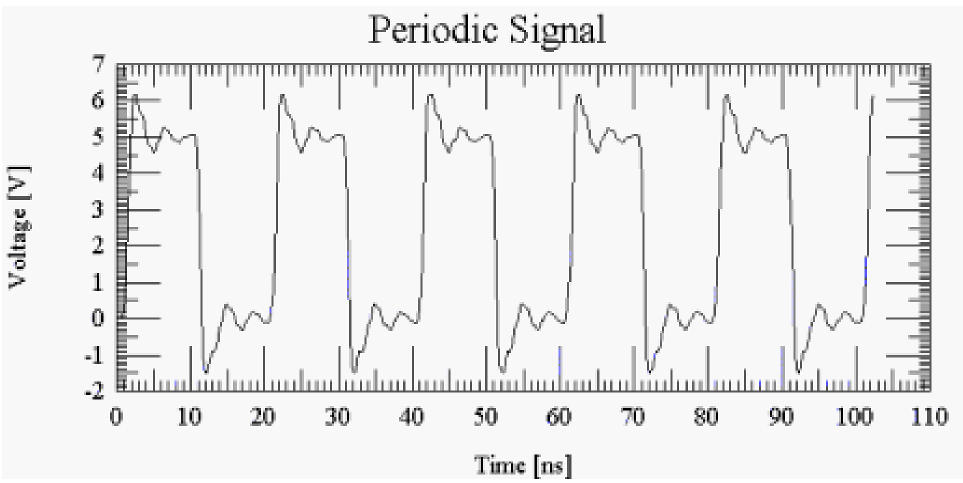
Notes on Selecting a Window

- If the input signal is periodic and a multiple of the sampling window, then use a Rectangular window.
- If the input is a short pulse or burst that starts and ends at the same amplitude, then use a Rectangular window (as long as the sampling window includes the entire transient).
- If the input is a section of a continuous waveform that is not periodic, then use the Hamming (or Triangular) window.

Using Window Extents

The Window Extents section of the FFT Preferences dialog box allows you to crop the time domain data before it is sent to the Fast Fourier Transform (FFT). A periodic pulse simulation most often results in waveforms that look like periodic square waves.

Figure A-15 A periodic signal graph



Fourier theory states that Fourier series representation of any periodic signal can be represented as the sum of an infinite number of sinusoids with a frequency spacing of 1/Tp between successive sinusoids, where Tp is the fundamental period.

The mathematical representation of a square wave and its Fourier transform is shown below:

Figure A-16 A Periodic Square Wave

$$g(t) = \begin{cases} A & \text{for } -\frac{T}{4} < t < \frac{T}{4} \\ 0 & \text{otherwise} \end{cases}$$

where T is the period of the square wave.

Figure A-17 The corresponding Fourier Transform

$$G(F) = \frac{AT}{2}\frac{\sin\left(\frac{\pi TF}{2}\right)}{\left(\frac{\pi TF}{2}\right)}$$

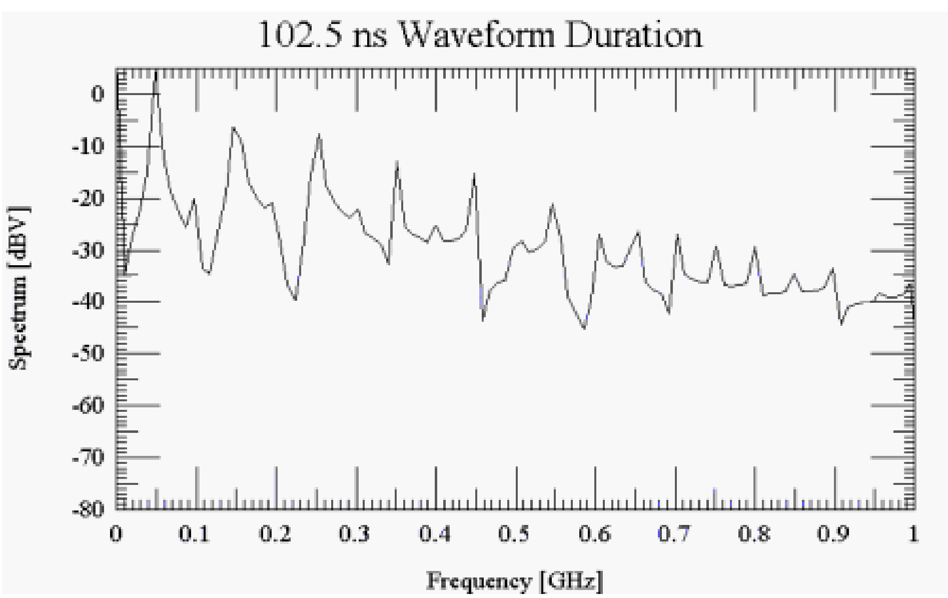
where T is the period of the square wave, and F is frequency.

Note: The frequency spectrum of the periodic square wave is an infinite number of harmonically related frequencies.

Example

Relate the sampling period to a signal's fundamental clock period. The periodic signal in the following figure has a simulation duration that is not harmonically related to the signal clock period. Using Windows Extents, you can crop the time domain data so that it is an integer multiple of the signal period (in this case, 100 ns).

Figure A-18 A graph of a 102.5ns waveform duration



Note how much cleaner the spectrum of the cropped waveform is in the following figure.

Figure A-19 A time domain graph cropped to 100ns