Verdi® and Siloti® Command Reference

Version O-2018.09-SP2, March 2019

SYNOPSYS®

Copyright Notice and Proprietary Information

Third-Party Software Notices

Verdi® Automated Debug Platform includes or is bundled with software licensed to Synopsys under free or open-source licenses. For additional information, see the third_party_notices.txt file in the *INSTALL_PATH/doc* directory of the Verdi software installation.

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.

Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at https://www.synopsys.com/company/legal/trademarks-brands.html.
All other product or company names may be trademarks of their respective owners.

Free and Open-Source Software Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

www.synopsys.com

Contents

About This Manual

Purpose

This is a reference manual for the Verdi platform and Siloti system and provides detailed descriptions for every command and commonly used functions. For use model and tutorial, refer to the appropriate User Guide and Tutorial document.

This manual does not need to be read from beginning to end. You should read this chapter and the *Introduction* chapter completely and then refer to the other chapters of interest.

- For browsing and debugging your HDL source code, refer to the *nTrace* chapter.
- For browsing and understanding your design schematically, refer to the *nSchema* chapter.
- For viewing your simulation waveforms, refer to the *nWave* chapter.
- For viewing and debugging your finite state machines, refer to the *nState* chapter.
- For analyzing and viewing your design activity over time, refer to the *Flow View* chapter.
- For speeding up your simulation while maintaining full visibility, refer to the *Visibility* chapter.
- For comparing different simulation results files, refer to the *nCompare* chapter.

The following topics are scattered throughout the document; search for the name of the component related to the topic:

- For analyzing your clock and reset trees and switching activity, or debugging timing analysis results, search for *nAnalyzer*.
- For implementing Engineering Change Orders (ECOs) in flexible schematic views, search for *nECO*.

Audience

The audience for this manual is any engineer who needs to look up the details of a command available in the Verdi platform or Siloti system, *nAnalyzer*, or *nECO*.

This document assumes that you have a basic knowledge of the platform on which your version of the Verdi platform and Siloti system runs: Unix or Linux, and that you are knowledgeable in design and verification languages, simulation software, and digital logic design.

Manual Organization

This manual is organized as follows:

- *About This Book* provides an introduction to this manual and explains how to use it.
- *Introduction* provides an overview of the Verdi and Siloti platforms.
- *nTrace* describes the commands used in the *nTrace* window.
- *nSchema* describes the commands used in the *nSchema* window.
- *nWave* describes the commands used in the *nWave* window.
- *nState* describes the commands used in the *nState* window.
- *Flow Views* describes the commands used in the *Flow View* frames.
- *nCompare* describes the commands used in the *nCompare* window.
- *Visibility* describes the commands in the **Tools** -> **Visibility** command menu of the *nTrace* window.
- *Editable Schematic* describes the commands used in the *Editable Schematic* window.
- *nECO* describes the commands used in an *nECO* schematic window.
- *nRegister* describes the commands used in the *nRegister* window.
- *nEditor* describes the commands used in the *nEditor* window.
- *Memory/MDA Window* describes the commands used in the *nMemory* window.
- *Transaction/Message Analyzer* describes the commands used in the *Transaction/Message Analyzer* window.
- *Clock Analyzer* describes the commands used in the *Clock Analyzer* window.
- *Assertion Debug* describes the commands and frames used with assertion debug.
- *Power Aware Debug* describes the commands and frames used with power aware debug.
- *Testbench Debug* describes the commands and frames used with testbench debug.
- *Interactive Simulation Debug* describes the commands and frames used with interactive simulation.
- *UVM Interactive Simulation Debug* describes the commands used in UVM debug frames, including *Resource View*, *Factory View*, *Phase View*, *Sequence View*, and *Register View*.

- *FFT* describes the commands used in the *FFT* frame.
- *Language Support and Compile/Import Methods* describes the import and viewing capability for supported languages.
- *Preferences* describes the preference options available in the Verdi platform and Siloti system.
- *Utilities* describes the utilities available in the Verdi platform and Siloti system.
- *Appendix A* describes the common usage environment variables.
- *Appendix B* explains how to customize the Verdi platform to specific applications.
- *Appendix C* describes the formats of the *.register* file used by *nRegister*.
- *Appendix D* summarizes the command syntax and error formats for *nCompare*.
- *Appendix E* provides details about power aware formats.
- *Appendix F* lists each available message number and its mapping usage when specifying the **+disable_message+<message_serial_numbers>** option in the *novas/verdi*, *vericom*, and *nrun* utility commands.
- *Appendix G* describes how to use Verdi for debugging large gate-level designs.

Conventions Used in This Book

The following conventions are used in this book:

- *Italic font* is used for emphases, book titles, section names, design names, file path, and file names within paragraphs.
- **Bold** is used to emphasize text and highlight titles, menu items, and other Verdi terms.
- `Courier type` is used for program listings. It is also used for test messages that the Verdi platform displays on the screen.
- **NOTE** describes important information, warnings, or unique commands.
- **Menu -> Command** identifies the path used to select a menu command.
- Left-click or Click means click the left mouse button on the indicated item.
- Middle-click means click the middle mouse button on the indicated item.
- Right-click means click the right mouse button on the indicated item.
- Double-click means click twice consecutively with the left mouse button.
- Shift-left-click means press and hold the <Shift> key then click the left mouse button on the indicated item.
- Drag-left means press and hold the left mouse button, then move the pointer to the destination and release the button.
- Drag means press and hold the middle mouse button on the indicated item then move and drop the item to the other window.

Functions of Common Buttons

This section describes the most common buttons that appear in the GUI forms.

- The **OK** button confirms the settings in the form and closes the form.
- The **Apply** button applies any changes but does not close the form.
- The **Cancel** button discards the settings in the form and closes the form.
- The **Close** button closes the form without applying any changes.

Bind Keys

Bind keys bind a command to a key on the keyboard. The key is either a single character or a multi-character sequence (such as, <key>, **Shift**+<key>, **Ctrl**+<key>, **Ctrl**+**Alt**+<key>) where the character can be alphabetic, numeric, function, or symbol. In this document, all alphabetic characters are indicated as an uppercase letter; however, this is a single keystroke (that is, Shift is not

required). Shift is a modifier used for a different keystroke sequences; it is not used to change the case. For example, **Ctrl+A** and **Ctrl+Shift+A** are different bind keys.

NOTE: If the Caps Lock key is enabled, the bind keys do not work.

Related Publications

- *Installation and System Administration Guide* - provides information on installing Verdi and Siloti platforms.
- *Verdi and Siloti Quick Reference Guide* - provides a quick reference for using the Verdi platform and Siloti system with typical debug scenarios.
- *Linking Novas Files with Simulators and Enabling FSDB Dumping* - provides detailed information on linking Novas object files with supported simulators for FSDB dumping and the related dumping commands.
- *Verdi User Guide and Tutorial* - provides detailed information on using the Verdi platform.
- *Siloti User Guide and Tutorial* - provides detailed information on using the Siloti system.
- *nAnalyzer User Guide and Tutorial* - provides detailed information on using the *nAnalyzer* Design Analysis module.
- *nECO User Guide and Tutorial* - provides detailed information on using the *nECO* Automated Netlist Modification module.
- *Release Notes* - provides current information about the latest software version. Click the '*View release notes*' link on the product downloads page on SolvNet.
- Language Documentation

  Hardware description (Verilog, VHDL, SystemVerilog and so on) and verification language reference materials are not included in this manual. For language related documents, refer to the appropriate language standards board (www.ieee.org, www.accellera.org) or vendor (www.synopsys.com, www.cadence.com) websites.

Introduction

Overview

The Verdi® Automated Debug Platform and the Siloti® Visibility Automation System are advanced solutions for debugging digital designs that increase design productivity with complex System-on-Chip (SoC), ASIC, and FPGA designs.

The Verdi platform provides powerful technology to help users comprehend complex and unfamiliar design behavior, automate difficult and tedious debug processes, unify diverse and complicated design environments, and infer the dynamic behavior of a design over time.

The Siloti system speeds up simulation by eliminating the overhead associated with recording signal values and by enabling full visibility of design activity based on a relatively small amount of signal data captured during full-chip simulation, regression simulation, and hardware emulation/acceleration. The Siloti system is fully integrated with the Verdi platform so that users can leverage the benefits of full visibility for gate or RTL debug.

Synopsys also provides the following two modules that complement the Verdi platform:

- The *nAnalyzer*^{TM} Design Analysis module for debugging the implementation aspects of the design such as clock trees and static timing failures.
- The *nECO*^{TM} Automated Netlist Modification module for solving problems associated with manual netlist changes.

Verdi and Siloti systems and the additional *nAnalyzer* and *nECO* modules enable engineers to locate, isolate, understand, and resolve bugs in a fraction of the time of traditional solutions. This maximizes the efficiency and productivity of expensive engineering resources, significantly reduces costs, and dramatically accelerates the process of getting silicon to market.

Product Overview

Verdi

The Verdi platform includes the following primary components. With the exception of the *nCompare* module which has its own document, each component has its own chapter in this manual.

Refer to the *Utilities* chapter for details on the Verdi command line options and other provided utilities. Refer to the *Preferences* chapter for details on setting preferences for the Verdi components.

nTrace

nTrace is a source code viewer and analyzer that operates on the Knowledge Database (KDB) to display the design hierarchy and source code (Verilog, VHDL, SystemVerilog, mixed, and power languages like CPF/UPF) for selected design blocks. *nTrace* quickly identifies signal connectivity information (drivers and loads) without any simulation overhead. With the FSDB, the simulation results can be back-annotated in the source code and then *nTrace* can analyze and determine a signal's active driver at a particular simulation time.

Refer to the *nTrace* chapter for details. Also refer to the *Language Support and Compile/Import Methods* chapter for details on language support.

nWave

nWave is a state-of-the-art graphical waveform viewer and analyzer that is fully integrated with the Verdi platform's source code, schematic, and flow views. A waveform search engine combined with backward and forward navigation allows users to search for signal transitions, bus values, discrepancies, or user-defined events easily. *nWave* also offers flexible signal group management, user-customizable glitch detection, a built-in logic analyzer, logical operations, events, display of delays back-annotated from SDF files, mixed analog/digital (A/D) display capabilities (including overlap, vertical zoom, delta x and y, arithmetic operations, analog-to-digital signal conversion, and others), transaction/message display, and support for power aware simulation results.

Refer to the *nWave* chapter for details. Also refer to the *FFT Pane* chapter for details on analog waveform support.

nSchema

nSchema is a schematic viewer and analyzer that generates interactive debug-specific logic diagrams showing the structure of selected portions of a design. RTL diagrams show the interconnection of finite state machines, storage elements, and multiplexers; gate-level diagrams show the interconnection of semiconductor vendor cells; and special flattened diagrams cut through the design hierarchy to isolate connected design elements. *nSchema* dynamically generates partial schematics to focus on the circuits of interest within a large design. Basic editable schematic capabilities and support for power design intent are also available.

Refer to *nSchema* and *Editable Schematic* chapters for details.

nState

nState is a finite state machine (FSM) viewer and analyzer that generates bubble diagrams for visualization of state machines that are automatically recognized by the Verdi platform when compiling the Verilog/VHDL source code modules. States and transitions are annotated with logic conditions and animated with simulation results. *nState* analyzes the simulation results to determine state and transition coverage.

Refer to the *nState* chapter for details.

Flow Views

The Flow Views are unique temporal views and analysis tools that allow visualization and analysis of the design's behavior through time. The *Temporal Flow View* identifies and displays causal control and data paths - the registers and signals that actually cause the erroneous values to occur - through multi-level combinational logic within one or more register-to-register transfer stages. This type of intelligently filtered temporal representation allows users to locate and identify problems in the most efficient manner possible. The *Temporal Flow View* also supports power aware debug.

Refer to the *Flow Views* chapter for details.

nCompare

The *nCompare* Waveform Comparison module compares simulation results stored in FSDB dump files using flexible, user-specified comparison criteria. Optimized for the extremely fast comparison of large data sets, the *nCompare* module is fully integrated with *nWave* to intuitively display any differences between runs.

Refer to the *nCompare* chapter for details.

The Verdi platform also includes these secondary components. Each component has its own chapter in this manual.

Assertion Debug

Assertion Debug support provides visualization and analysis capabilities for assertions and assertion results.

Refer to the *Assertion Debug* chapter for details.

Testbench Debug

Testbench Debug provides visualization and comprehension capabilities for designs including SystemVerilog Testbench code.

Refer to the *Testbench Debug* and *Interactive Simulation Debug* chapters for details.

Power Aware Debug

Power Aware Debug provides visualization and analysis capabilities for designs including power design intent source code (that is, CPF or UPF).

Refer to the *Power Aware Debug* chapter for details.

Transaction/Message Analyzer

The *Transaction/Message Analyzer* window provides visualization and analysis capabilities for transactions and message data.

Refer to the *Transaction/Message Analyzer* chapter for details.

nMemory

The *nMemory* window allows multi-dimensional memory arrays (including two-dimensional) to be viewed in a tabular format. The display of word size and radix can be modified and the array searched for a specific value.

Refer to the *Memory/MDA Pane* chapter for details.

nRegister

The *nRegister* window provides a graphical window for viewing simulation results. Lines, rectangles, arrows, and text can be added and the objects aligned to create a unique representation of the results.

Refer to the *Register Window* chapter for details.

nEditor

nEditor is a complete text editor. The text is color-coded by language syntax.

Refer to the *nEditor* chapter for details.

Siloti

The Siloti system provides the Visibility Analysis engine to determine the minimum set of signals for full visibility of the design and the Data Expansion engine to compute the remaining signal values 'on-demand'. In addition, the optional Abstraction Correlation module maps gate-level signal data to the RTL for easier debug and understanding.

Refer to the *Visibility* chapter for details.

Additional Modules

The following modules can be purchased and used in conjunction with the Verdi platform.

nAnalyzer

The *nAnalyzer* module provides the ability to analyze clock and reset trees (including crossing paths), to qualify Clock Tree Synthesis (CTS), to annotate standard delay format (SDF) files and CTS results, to load and display timing results from standard timing analysis tools, and to perform switching analysis on the design. These functions build on top of the functional debug aspects of the Verdi platform.

Search for *nAnalyzer* in this document to locate related commands. Refer to the *Clock Analyzer* chapter and the *nAnalyzer User Guide and Tutorial* for other details.

nECO

The *nECO* module provides the ability to perform gate-level engineering change orders (ECOs) in the Verdi platform's flexible schematic views. The *nECO* module takes full advantage of the Verdi platform's sophisticated capabilities to propagate changes throughout the design hierarchy and automatically creates new nets and ports.

Search for *nECO* in this document to locate related commands. Also refer to the *nECO* chapter for *Eco Window* related commands, the *Preferences* chapter for details on setting preferences for *nECO,* and the *nECO User Guide and Tutorial* for other details.

User Interface Overview

This section introduces the main aspects of the User Interface. This includes information on the Welcome page and associated options, framework and customization options, and global toolbar icons.

Welcome Page

The *Welcome* page can be accessed by clicking the **Welcome** icon in the lower right corner of the main window or by selecting the **Help -> Welcome** command. This page displays icons that provide easy access to the new features list, work modes, history, and documentation.

Figure: Welcome Page

What's New

Click the **What's New** icon to display a list of documents available as part of Verdi.

Work Modes

Click the **Work Modes** icon to display a page with the available work modes. After selecting the desired work mode, click the **Go to Work** button to begin debugging. The work panes are maximized for the selected mode.

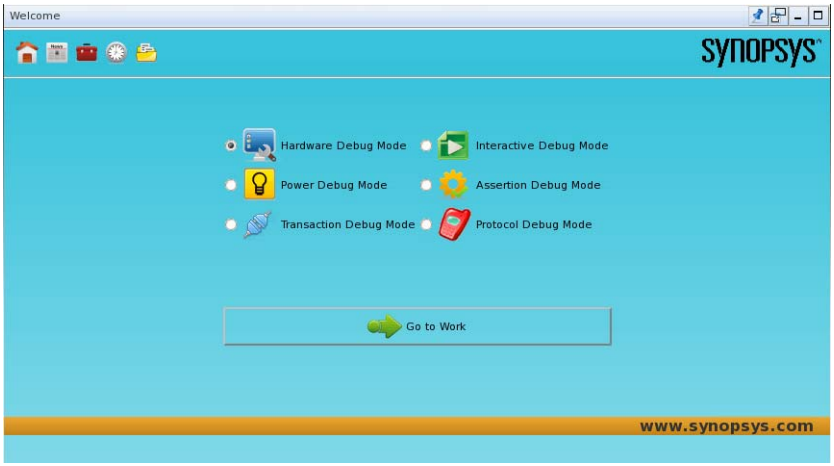


Figure: Work Modes Page

The following work modes are available:

- Hardware Debug Mode
- Interactive Debug Mode
- Power Debug Mode
- Assertion Debug Mode
- Transaction Debug Mode
- Protocol Debug Mode

The default panes that are activated for Hardware Debug Mode, Testbench Debug Mode, Power Debug Mode, Assertion Debug Mode, Transaction Debug Mode, and Interactive Debug Mode are displayed in a tip. Refer to the *Window Commands* section of the *nTrace* chapter for details on the different work modes and instructions for saving a user mode.

History

Click the **History** icon  to display a page with the five most recent debug sessions listed. After selecting the session, click the **Restore Session** button to begin debugging. Click the **Restore Others** button to open the *Restore Session* form and then locate other session files.

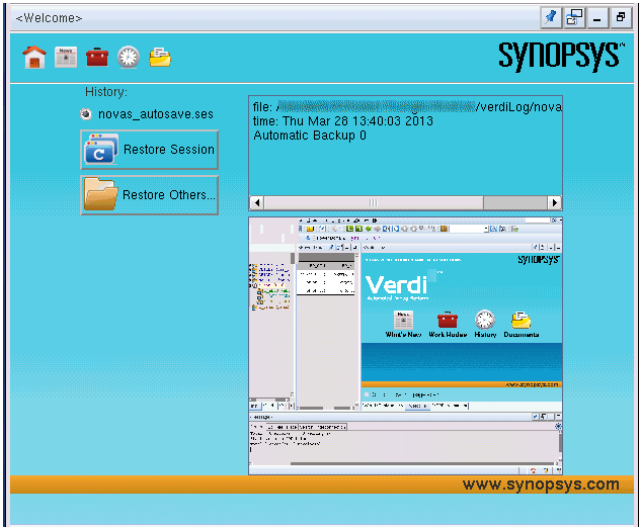


Figure: History Page

Documents

Click the **Documents** icon  to display a page with the available documents, video demos, and links to the VIA Apps Exchange website and customer support.

Figure: Documents Page

Customized links can be added to the *Documents* page by creating a *user_documents.txt* file and placing it under the doc directory in the Verdi

platform's installation path. The content of the file can be a link to a PDF or an HTML file with the following syntax:

```
"<file_path>" "<TitleOnPage>"
"<url_path>" "<TitleOnPage>"
```

Example:

```
"http://www.google.com" "Google Webpage"
"/myhome/doc/training.pdf" "Internal Training"
```

Framework and Customization

The User Interface consists of a central framework where all components are docked. Each component (window or widget) has a default pane location. The default pane locations may be changed (undocked) by dragging (press and hold the middle mouse button) the component and dropping (release the middle mouse button) it in another location (docked). If the default/new pane location is on top of an existing pane, a new tab gets added. All components may be undocked (standalone panes) and some components (such as, *nSchema*, *nWave*, *Flow Views*, and *nState*) may also become fully-functional separate windows. When a widget is undocked, it always remains in the foreground.

When multiple panes are undocked as standalone windows, window titles of these active windows are listed under the **Window** command. For the top window, the undocked window titles are listed based on the window creation order. For a standalone window, except for the top window, the top window title is listed first, followed by the remaining undocked window titles based on their creation order. Select the window title of interest to switch to the target window.

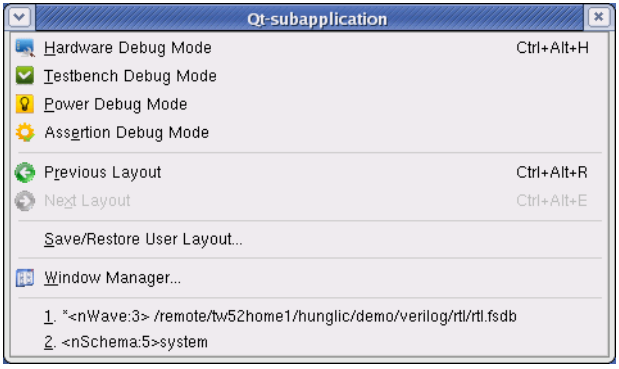


Figure: Windows Titles Listed on Top Window

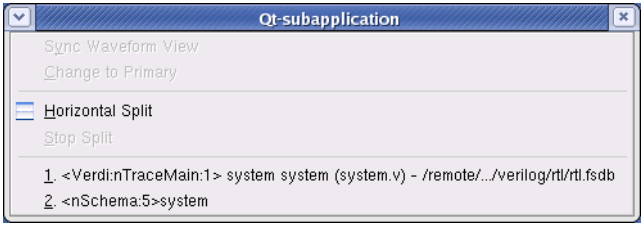


Figure: Windows Titles Listed on Standalone Window

In addition, menu commands, bind keys, and toolbar icons can all be customized (such as, names, locations, and symbols). Refer to the **Tools -> Customize Menu/Toolbar** command in the *nTrace* chapter for details.

Definitions

- **Component**: A window or widget that is either a part of the main window by default or can be added by invoking a command.
- **Pane**: An area in the main window that is occupied by one or more components.
- **Widget**: A component that is a subsidiary pane of the main window. A widget does not have its own toolbar or menu commands. A widget cannot be maximized to full screen size.
- **Window**: A component that has its own toolbar and menu commands. A window can be docked in the main window as a pane or a tab or it may be undocked as a standalone window. A window can be maximized to full screen size.

Icons for Dockable Panes

The icons in this section are provided on the pane banner and menu bar of most panes that may be docked and undocked. Toolbar icons are described in other chapters.

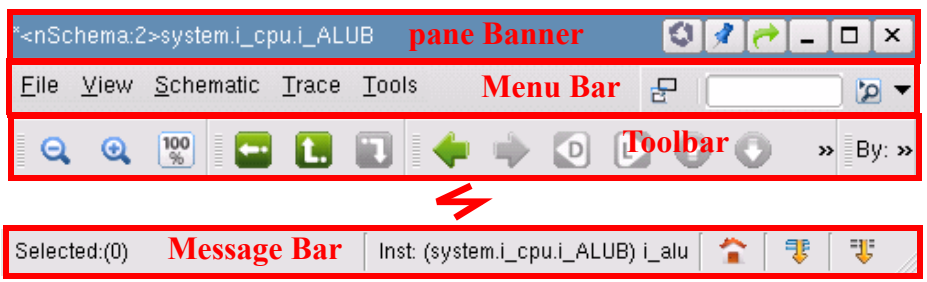


Figure: Icon Locations

Pane Banner

The following icons are available in the pane banner of most dockable panes.

Sync. Signal Selection (Enabled /Disabled)

Click the **Sync. Signal Selection** disabled icon (gray version) to enable synchronization of a selected signal in the associated pane/window. When two or more panes have enabled this toolbar icon, these panes are synchronized. To disable signal synchronization in a pane, click the **Sync. Signal Selection** enabled icon (blue version).

Refer to the **View -> Sync. All Signal Selection** and **Release Signal Selection Sync.** command descriptions in the *nTrace* chapter for global synchronization details.

Keep as Top

Click the **Keep as Top** icon to place the pane on top of other panes in the same display area when a new tab is created (such as, select the **New Schematic** icon). If a pane is moved to a different area, it automatically becomes the top pane even if this icon is enabled (such as, if a waveform pane is dragged and dropped to the source code area).

Undock /Dock

Click the **Undock** icon to release the pane from the main window and have it become a standalone window. Click the **Dock** icon to dock the window as a pane in the main window. Alternatively, a pane can be undocked by dragging (press and hold the middle mouse button) and dropping (release the middle mouse button) it outside the main window.

Hide

Click the **Hide** icon to hide the pane from view. Right-click the main toolbar or a title bar to display a list of available panes and select a hidden pane to restore.

Maximize /Restore

Click the **Maximize** icon to expand the pane and occupy the entire main window. Other panes are hidden. Click the **Restore** icon to restore the original layout. Double-clicking the title pane also maximizes/minimizes the pane.

NOTE: If the maximized pane is hidden, deleted, or closed, or if the pane becomes a window, the original layout is restored without this pane.

Delete or Close 

Click the **Delete** icon to remove the current pane from the window. Click the **Close** icon on the main window to exit the tool completely.

Menu Bar

The following icons are available on the menu bar of most windows and panes:

Spotlight 

Click the triangle on the right to select the topic to search for (**Menu**, **Reference Manual**, **Tcl**, or **Preferences**), and then specify a string to search for in the text field to dynamically search for the first object that matches in the current pane/window. Press the **Spotlight** icon to display a list of objects matching the search string. A partial string is valid and searching is case-insensitive. The **Spotlight** text field and icon are available in all standalone windows.

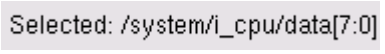
NOTE: Searching with a space is not supported when **Reference Manual** and **Tcl** topics are selected.

Toolbar

The toolbar icons are different for each pane. Refer to the *Toolbar Icon and Fields* section in each chapter for details.

Message Bar

The following icons are available in the message bar of the main window:

Selected 

The **Selected** text field shows details about the selected object in the active pane.

Welcome 

Click this icon to display the *Welcome* page as a new tab in the same pane location as the source code pane.

Hide Banners 

Bind Key: Ctrl+Shift+B

Click this icon to hide the banner in all panes. Click this icon again to restore the pane banners.

Hide Menus 

Bind Key: Ctrl+Shift+E

Click this icon to hide the menu bar in all panes. Click this icon again to restore the menu bars.

Global Bind Keys

The bind keys in this section are provided for global commands.

Next Dock Tab

Bind Key: Ctrl+Shift+Page Down

NOTE: The **Page Down** key on the keyboard number pad is not supported for this bind key usage.

When the mouse is placed on a dockable pane/widget, this command switches to the next tab within multiple tabs of this dockable pane/widget.

Previous Dock Tab

Bind Key: Ctrl+Shift+Page Up

NOTE: The **Page Up** key on the keyboard number pad is not supported for this bind key usage.

When the mouse is placed on a dockable pane/widget, this command switches to the previous tab within multiple tabs of this dockable pane/widget.

nTrace

Overview

When the Verdi platform is started and a work mode is selected, the *nTrace* main window is displayed and serves as the central framework from which other views are created. The main window consists of several frames that may be docked/ undocked and moved within the framework. Refer to the *User Interface* chapter of the *Verdi User Guide and Tutorial* for details.

The menu bars when the Verdi platform is started are as follows:

Figure: nTrace Menus

The menus for *nTrace* are summarized and explained in the following sections.

Menu Summary

The *nTrace* menu commands are summarized below. All commands can be double-clicked to jump to the corresponding command description. For right-click menu options, refer to the *Right-Click Options* section for details.

File Commands

Import Design
Reload Design
View Import Log
Compile Design
Open File
Open Waveform File
Load Aux Annotation Waveform File
Close Waveform File
Load Behavior Database
Import CPF/UPF Files
Reload CPF/UPF Files
SDF ->
  Load SDF Files - nAnalyzer
  Close SDF Files - nAnalyzer
  View SDF Log - nAnalyzer
  Report Undefined Paths - nAnalyzer
Import Path Data Files - nAnalyzer
Import Analysis Results - nAnalyzer
Load Clock Domain Results - nAnalyzer
Load Clock Tree Database - nAnalyzer
Print
Capture Window
Restore Session
Save Session
Save Design Snapshot
ECO ->
  Save ECO Netlist - nECO
  ECO Report - nECO
  Save ECO Script - nECO
  Load ECO Script - nECO
  Save Spare Cell Script - nECO
Exit

View Commands

Source Tab ->
  New Source Tab
  Close Source Tab
  Close Inactive Tabs
  Show Active Scope Tab
Signal List
Constraint View
Inheritance View
FSDB Message View
Line Number
Auto Wrap
X-propagation Instrument
Enable Syntax Color
Identify False Logic
Source Code Folder ->
  Expand All in File
  Collapse All in File
  Expand All in Design
  Collapse All in Design
  Statement Folder ->
    Expand All in File
    Collapse All in File
    Expand All in Design
    Collapse All in Design
  Comment Folder ->
    Expand All in File
    Collapse All in File
    Expand All in Design
    Collapse All in Design
  Comment Folder ->
    Expand All in File
    Collapse All in File
    Expand All in Design
    Collapse All in Design
  Port List Folder ->
    Expand All in File
    Collapse All in File
    Expand All in Design
    Collapse All in Design
  Sub-program Folder ->
    Expand All in File
    Collapse All in File
    Expand All in Design
    Collapse All in Design
  Compiler Directive Folder
    Expand All in File
    Collapse All in File
    Expand All in Design
    Collapse All in Design
  False Logic Folder
    Expand All in File
    Collapse All in File
    Expand All in Design
    Collapse All in Design
Macro Folder ->
  Expand All in File
  Collapse All in File
  Expand All in Design
  Collapse All in Design
Implicit Port Folder ->
  Expand All in File
  Collapse All in File
  Expand All in Design
  Collapse All in Design
Hierarchy Tree by Level
Sync. All Signal Selection /
Release Signal Selection Sync.

Source Commands

Find Scope
Find Signal/Instance/Instport
Find String
Go To ->
  Line
  1st Executable
  Current Statement
Show ->
  Calling
  Definition
  Entity
  Reference
Edit Source File
Parameter Annotation
Active Annotation
Function Annotation
Leading Zeros
Expand Macro
Expand Decrypted Code
Expand Implicit Port
Signal Value Radix
Signal Value Notation
Set/Unset Bookmark
Previous Bookmark
Next Bookmark
Manage Bookmarks
Manage User-defined Folders
Recent Files ->
  [Filename]

OneTrace Commands

Value Change
Driver
Load
Connectivity
Chain Driver
Fan-in
Fan-out
Mapped Signal
Trace across Hierarchy
Stop on Complex Exp Instport
Trace Including Top Level Port
Trace across Power Design
Bus Driver
Bus Load
Bus Connectivity
Pop View Up from Port
Push View In from Port
Show Previous
Show Next
Show Previous in Hierarchy
Show Next in Hierarchy
Reset Traced Signal's Color
Forward History
Backward History
Reset History
Interface Mapping
From HDL to HVL Interface
From HVL to HDL Interface

Simulation Options

Run/Continue
Save State
Step
Step in Thread
Next in Thread
Step in Testbench
Step Out
Quit
Kill Simulator Process
Save State
Restore State
Add Checkpoint
Delete Checkpoint
Interactive Debug Panes
Set Time Scale Display
Go to Source Position

Interactive Debug Panes

Manage Breakpoints
Interactive Debug Panes
Delete Watch
Rename Watch

Tools Commands

Visibility ->
  Essential Signal Analysis ->
    Generate ESDB
  Data Expansion ->
    Setup Data Expansion
    Enable Data Expansion
    Refresh Signal Values
  Gate/RTL Correlation ->
    Open Slave Process
    Prepare CRDB
    Correlation Console
    Generate Result
    Advanced Commands ->
      Extract CRDB
      Perform CRDB Correlation
      Save CRDB File
      Load CRDB File
      Correlation Analyzer
VC Apps Toolbox ->
New SmartLog
New Waveform
New Schematic from Source ->
  New Schematic
  Browser Window
  Flattened Window
  Hierarchical Flattened View
  ECO Window for Selected
  Editable Window for Selected
  Fan-in
  Fan-out
  Driver
  Load
  Connectivity
  Clock Tree
  Reset Tree
Temporal Flow View ->
 Auto Trace
 Trace Glitch
 New Temporal Flow View
 Add Reference Signals
Trace X
Behavior Analysis
Extract Interactive FSM
Browse/Watch/List ->
  Interface Browser
  Browse Cell Summary
  Watch Signals
  Watch Expressions
  List X
  Browse SDF Path
Register
Memory ->
  Memory Definition Table
  Memory/MDA
Property Tools ->
  Evaluator
  Statistics
  Analyzer
  Add Temporary Assertions
Transaction Debug ->
  Transaction and Protocol Analyzer
  Transaction Table View
  Classic Transaction ->
    Evaluator
    Analysis Window
  Classic Message ->
    Analysis Window
 UVM/OVM/VMM Debug ->
  All Views
  Resource/Config View
  Factory View
  Phase View
  Sequence View
  Register View
Clock Analyzer ->
  Extract Clock Information
  List Clock Domains
  List Specified Clock Domains
  Highlight Clock Domains
  Check Crossing Paths
  List Crossing Paths
Switching Analysis ->
  New Query
  Manage Report
Invoke HW/SW Debug
Highlight
Preferences
Customize Menu/Toolbar
Run Simulation
Attach to Simulation
Detach Simulation
Stop Interactive Debug

Power Commands

Show Power State Table
Show Power Mode Table
Report Impacted Signals by Scope
Check Power Sequence

List HDL Signals
Highlight Power Domain
Full Power Map

Window Commands

Hardware Debug Mode
Testbench Debug Mode
Power Debug Mode
Assertion Debug Mode
Transaction Debug Mode
Interactive Debug Mode

Preserve Visible Windows/Widgets
Previous Layout
Next Layout
Save/Restore User Layout
Window Manager

Bind Keys

For a complete list of bind keys used in *nTrace*, refer to the *nTrace* section in the *Bind Key Summary* for details.

File Commands

Import Design

Menu Bar: File -> Import Design

Toolbar Icon: 

The **File -> Import Design** command opens the *Import Design* form.

NOTE: Even if the symbol library is specified in the **Schematics** folder **-> Symbol Library** page of the *Preferences* form (invoked with the **Tools -> Preferences** command), when importing the design with the HDL code and symbol library at the same time, VHDL and Verilog treats the design as a user-defined module in *nTrace*, instead of a symbol library cell. The module is treated as a symbol library cell when **-v** or **-y** is specified.

The *Import Design* form includes the **From Library** tab and the **From File** tab. After selecting the required design and top modules on either tab, click **OK** to close the form and import the design with the specified options.

From Library Tab

A Verilog or VHDL design can be imported from a pre-compiled library. The Verilog and VHDL source can be incrementally compiled into the defined libraries using these batch mode compilers.

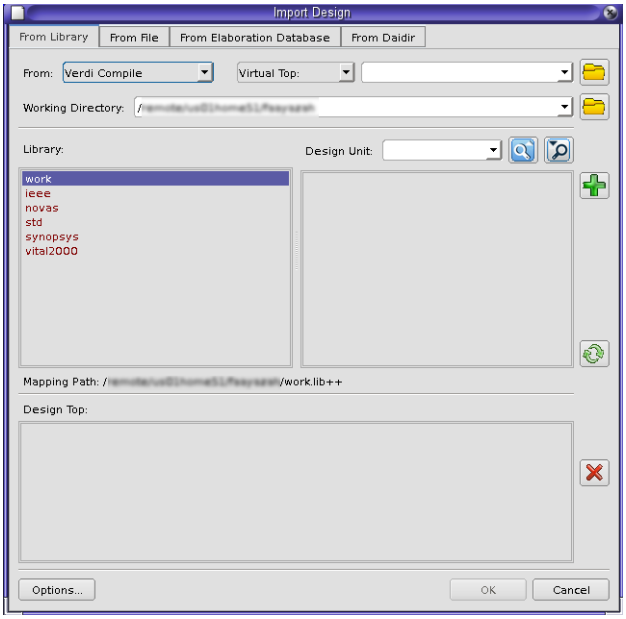


Figure: Import Design Form - From Library Tab

The *Import Design* form includes the following fields, sections, and buttons on the **From Library** tab:

- **From:** Select if the Verdi Knowledge Database (KDB) is generated by **Verdi Compile** or **VC/VCS Native Compile**. When **Verdi Compile** is selected, Verdi loads the Verdi Knowledge Database generated by the *vericom* or *vhdlcom* utilities and the library mapping from the *novas.rc* resource file is used. When **VC/VCS Native Compile** is selected, Verdi loads the KDB generated by VCS and the library mapping from the *synopsys_sim.setup* setup file is used.

  When the **-simflow** option is specified on the Verdi command line, **VC/VCS Native Compile** is selected when the *Import Design* form is opened.

- **Virtual Top:** This field can be used to import only a portion of the design. Refer to the *Virtual Top Example* section for details.

  Configuration File: Toggle the **Virtual Top** field to select **Configuration File** and import part of the design into the Verdi platform to enhance performance. Refer to the *verdi* section in *Utilities* chapter for details.

- **Working Directory:** Specify the working directory for the library in this text field.

- **Library:** This section lists the available libraries in the current working directory. Library names in the **Library** section are listed by library type first (standard libraries have higher priority than user-defined libraries), then by letter case (uppercase has higher priority than lowercase), and the remaining are sorted by alphabetical order.
- **Design Unit:** This section lists all top objects from the selected pre-compiled library. The design can be specified by selecting a design in the **Design Unit** list or entering a search string for the design name in the field. Click the **Find Previous** icon  and the **Find Next** icon  (or press **Enter**) to search for the design in the design unit. The design unit matching the specified string is selected in the list.

  When importing a Verilog or VHDL library, multiple top modules can be selected in the **Design Unit** list. Click the **Refresh** icon  to see the updated library generated in the working directory and click the **Add** icon  to add the design unit to the **Design Top** list.
- **Design Top:** This section displays the selected design tops to be added to the design browser frame. Click the **Delete** icon  to remove the selected design unit from the **Design Top** list.

NOTE: 1. If a library is imported from **Virtual Top**, **Design Top** is disabled (and the **Delete** and **Add** icons is disabled).
2. In the case of a mixed-language (Verilog and VHDL) design, the **Import Design From Library** method must be used.
3. The **Library** section of the *novas.rc* resource file is used to specify the logical and physical library path mapping for the **Import Design From Library** approach.

- **Options:** Click **Options** to open the *Library Options* form in which the library names and their library paths can be specified or deleted. Click the **Browse** icon  to view the directory structure and locate a library to add to the list. Click the **Delete** icon  to remove the library from the list.

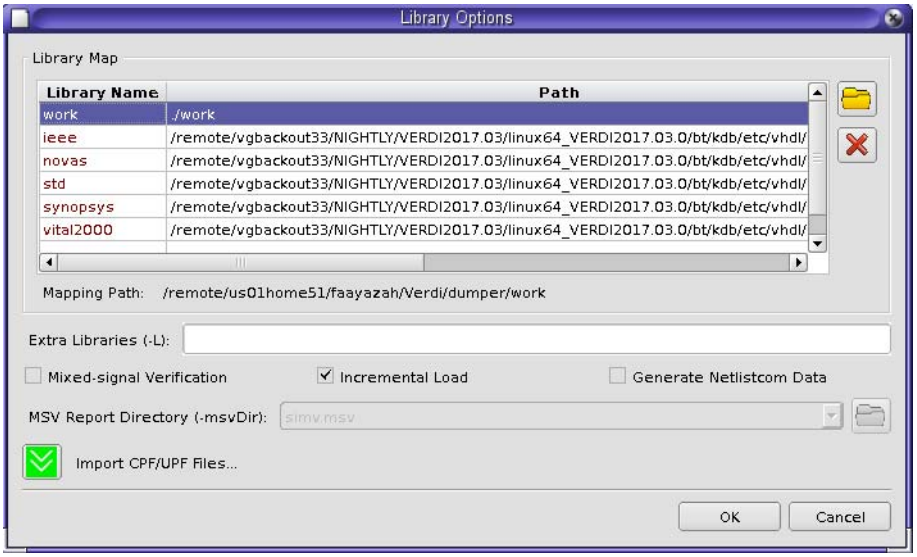


Figure: Library Options Form

The *Library Options* form includes the following fields and options:

- **Extra Libraries (-L):** Specify extra libraries to be imported.
- **Interface Element Report (-ieFile):** Specify the report file of the interface element to be imported.
- **Incremental Load:** When this option is turned *on*, the library is loaded incrementally.
- **Generate Netlistcom Database:** When this option is turned *on*, the Verdi platform translates libraries to detailed schematic views. Refer to the *netlistcom* section in the *Utilities* chapter for details.
- **Import CPF/UPF Files:** Click the icon to expand the section where CPF/UPF files can be imported.

From File Tab

This tab opens the design and compiles the source code. The Verilog/VHDL compilation options can be specified in addition to the file names. The Verdi platform parses the Verilog/VHDL source code to recreate their data structures in the KDB, with the advantage that the Verdi platform serves as a Verilog/VHDL syntax checker.

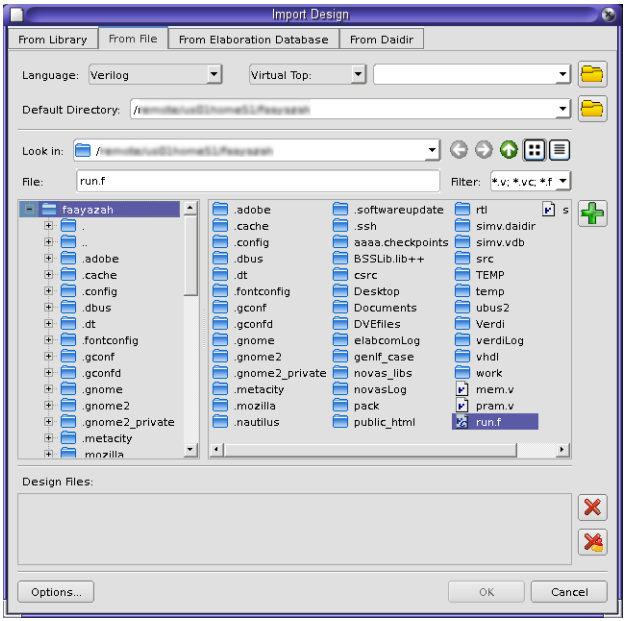


Figure: Import Design Form - From File Tab

The *Import Design* form includes the following fields, sections, and buttons on the **From File** tab:

- **Language:** Select the design language to be imported using the **Language** options: **Verilog**, **Verilog-2001**, **SystemVerilog**, **SystemVerilog-2009**, **SystemVerilog-2012**, **VHDL-87**, **VHDL-93**, **VHDL-2000**, or **VHDL-2008**. In the case of VHDL, enter the top design unit in the **Virtual Top** selection field. If the top design unit is not specified here, a *Set Top Module* form opens so the top design unit can be selected when the VHDL design is imported. Click **OK** to complete the selection.
- **Virtual Top:** This field can be used to import only a portion of the design. Refer to the *Virtual Top Example* section for details.

  Configuration File: Toggle the **Virtual Top** field to select a **Configuration File** and import part of the design into the Verdi platform to enhance performance. Refer to the *verdi* section in *Utilities* chapter for details.
- **Default Directory:** This field specifies the file path for files without a full path in the command argument file (for example, *run.f)*. Specify the **Default Directory** by entering the details in the field or click the **Browse** icon .

Alternatively, when any *run.f* is added to the **Design Files** list, the directory path for this *run.f* file is automatically added to the field.

- **Filter:** This field filters the file format for the file list. The default file extensions are **.v*, **.vc*, **.f*, and **.sv*. Additional, filters can be added in this selection field using a semicolon (;) as the separator.
- **Design Files:** This list specifies the Verilog files to be imported into the Verdi platform. *run.f* is selected as the default file from the **File List**. Add or delete files in the **Design Files** list by selecting the file and then click the **Add** icon  , **Delete** icon  , or **Delete All** icon  .
- **Options:** Click **Options** to open the *Import Design Options* form in which the **Library Directories**, **Library Files**, or **Other Options** can be specified for the current design. These options serve as a command line option when a design file is imported into the Verdi platform. If options are already specified in the **Importing Design** page under the **Tools -> Preferences -> Source Code** page (which is the global setting), these options is copied into the *Import Design Options* form. Different options can be specified in the *Import Design Options* form, but these options are only valid for the current design.

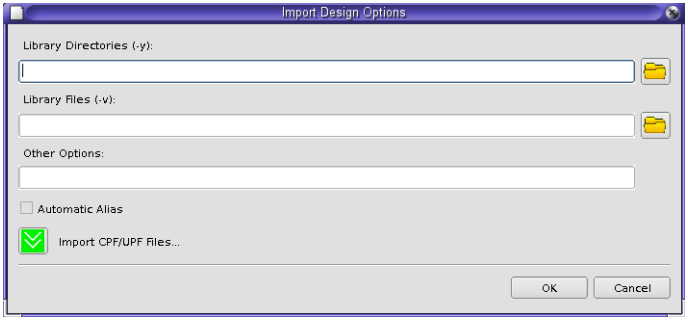


Figure: Import Design Options Form

The *Import Design Options* form includes the following fields and options:

- **Library Directories (-y):** Specifies the library directory, which appends to the command line option **-y**. Type in the directory path or click the **Browse** icon  to browse using the *Choose Library Directory* form. Refer to the **-y** option in the *verdi* utility description for details.
- **Library Files (-v):** Specifies the library file that appends to the command line option **-v**. Enter the file name or click the **Browse** icon  to browse in the *Choose Library File* form.

- **Other Options:** See the supported Verilog reference manual for available options. For these options, invoke the Verdi platform with the command line option **-h**.
- **Automatic Alias:** This option instructs the Verdi platform to automatically create signal aliases according to Verilog's constant parameter. If a parameter is created as a constant and then is assigned or compared to other signals, the Verdi platform automatically creates the signal alias as that parameter.
-  **Import CPF/UPF Files:** Click the  icon to expand the section where CPF/UPF files can be imported.

Virtual Top Example

To maintain the consistency of the design hierarchy, the correct hierarchical paths of the top sub-hierarchy modules must be specified. The definitions are stored in a *.map* file and the syntax is as follows:

```
module_name = hierarchical_instance_name
```

The Verdi platform generates a virtual top automatically according to the definitions in the *.map* file and imports only a subset of the design.

NOTE: The hierarchical instance name in the virtual top (*.map*) file accepts environment variables to specify the complete hierarchical path.

Assume there is a VHDL design with a top module, *system*, which contains two instances: *cpu* and *pram* and that cpu contains an instance *alub*. Assume only *alub* and *pram* should be loaded into the Verdi platform instead of loading the entire *system* hierarchy. First, specify the design files associated with *cpu*, *pram*, and their sub-instances in a file called *vtop.f*. Then there are two methods for defining the virtual top file.

Method 1. Define the contents of a virtual top file, vtop.map, as follows:

```
alub = system.i_cpu.i_alub
pram = system.i_pram
```

Method 2. Alternatively, specify environment variables in UNIX as follows:

```
setenv vtop1 system
setenv vtop2 i_cpu
```

Then, define the contents of a virtual top file, *vtop.map*, as follows:

```
alub = $vtop1.$vtop2.i_alub
pram = $vtop1.i_pram
```

After the virtual top file and design source list file are created, this subset of the design can be imported with the virtual tops *system* and *i_cpu* using either one of the following two methods:

1. Import the design from file.
   a. Invoke the Verdi platform from the command line:
   ```
   verdi -vtop vtop.map -vhdl -f vtop.f
   ```
   b. Use the following commands in the Verdi GUI:
   - Select the **File -> Import Design** command.
   - Select the **From File** tab.
   - Set the **Language** field to *VHDL-87* or *VHDL-93*.
   - Set the **Virtual Top** field to *vtop.map*.
   - Set the **Design File** field to *vtop.f*.
   - Click **OK**.
2. Import the design from library. Invoke the Verdi platform from the command line (only the command line mode works):
   ```
   verdi -vtop vtop.map -lib work
   ```

NOTE: The virtual top is effective both when importing a design from a library and when importing a design from source (or from files). However, when importing from source, the **-vtop** option does not have an effect if the specified top is included in the imported source. For example, assume that a design has the following design hierarchy:

```
system              ----> (system.v)
- i_cpu (CPU)   ----> (CPU.v)
   - i_ALUB (ALUB) ----> (ALUB.v)
   - i_CCU (CCU)      ----> (CCU.v)
   - i_PCU (PCU)      ----> (PCU.v)
```

Assume a *run_all.f* design file list contains:

```
system.v
ALUB.v
CCU.v
PCU.v
CPU.v
```

Assume a *run_cpu.f* design file list contains:

```
ALUB.v
CCU.v
PCU.v
CPU.v
```

Assume a *vtop.map* map file contains:

```
CPU=system.i_cpu
```

The following two commands have the same imported design hierarchy and the same contents in the *system* module (the same as *system.v*):

verdi -vtop vtop.map -f run_all.f
verdi -f run_all.f

The following two commands have the same imported design hierarchy, but different contents in the *system* module:

verdi -vtop vtop.map -f run_cpu.f
====> The system module is built virtually.
verdi -f run_all.f

The following two commands have a different imported design hierarchy:

verdi -vtop vtop.map -f run_cpu.f
verdi -f run_cpu.f

From Elab Database Tab

The elaboration database can be loaded directly in this tab.

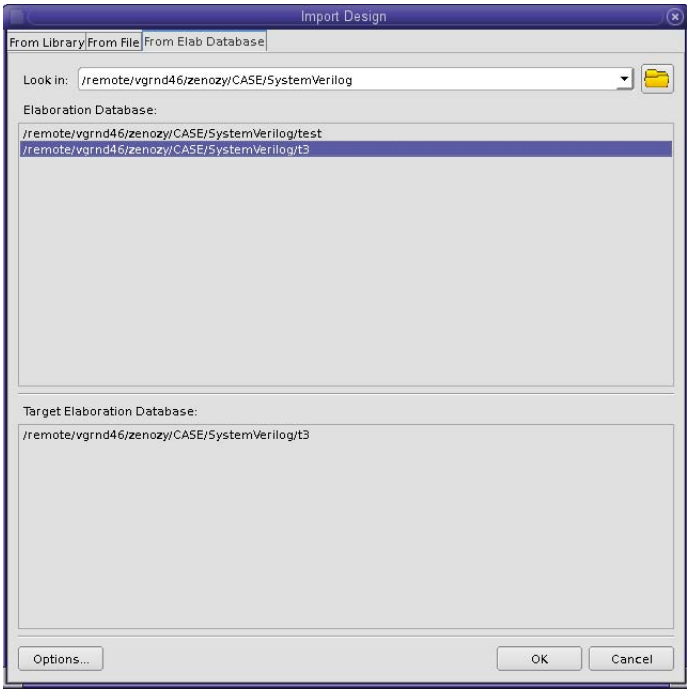


Figure: Import Design Form - From Elab Database Tab

- **Look in**: Specify the searching directory of or click the **Browse** icon to browse directories. All valid elaboration database under the specified

directory is listed. To import an elaboration database, select a elaboration database and click **OK**. Only one elaboration database can be imported at one time.

- **Options**: Specify advanced importing option.
  - **Import CPF/UPF Files:** Click the  icon to expand the section where CPF/UPF files can be imported.

Reload Design

Menu Bar:	File -> Reload Design
Bind Key	Shift+L

This command reloads the design files and simulation results. The Verdi platform maintains the current debugging environment with the latest source and simulation results but the trace results from previous operations in each window are removed. When the design is imported from source files and one or more source files are modified with the **Source -> Edit Source File** command, the **Reload Design** command can be invoked to recompile the design and all opened windows are updated with the new design.

NOTE: When the design is imported from a pre-compiled library and one or more source files are modified, the design must be recompiled outside the Verdi platform using the *vericom*/*vhdlcom* utilities first and then reloaded with the updated library using the **Reload Design** command.

View Import Log

Menu Bar:	File -> View Import Log

This command opens the *Text Viewer* frame to view the compile log file.

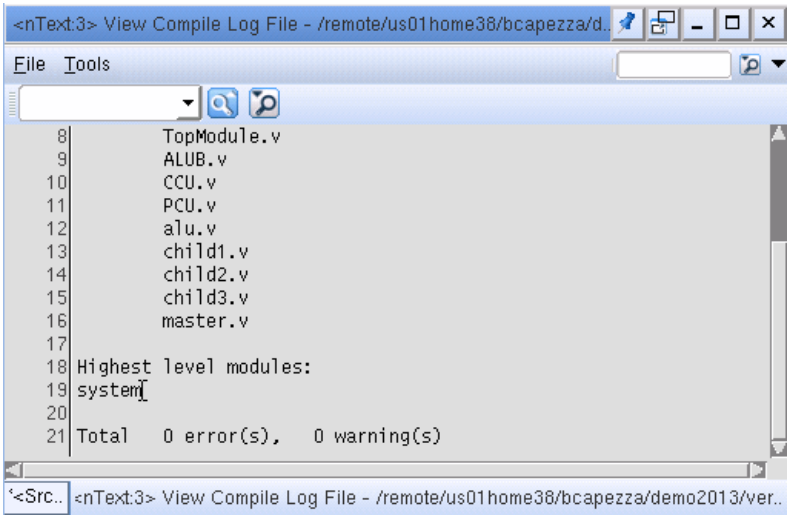


Figure: Text Viewer Frame - Compile Log

Refer to the *Text Viewer Frame* for details.

Compile Design

Menu Bar: File -> Compile Design

This command compiles a Verilog or VHDL design from a pre-compiled library that is already created by using the utilities *vericom* or *vhdlcom*.

Select a design with the browser or the command line window then click **OK**.

Verilog and VHDL source can be incrementally compiled into the defined libraries using the batch mode compilers *vericom* and *vhdlcom* respectively. For details, refer to the *vericom*, *vhdlcom*, and *verdi* sections in the *Utilities* chapter.

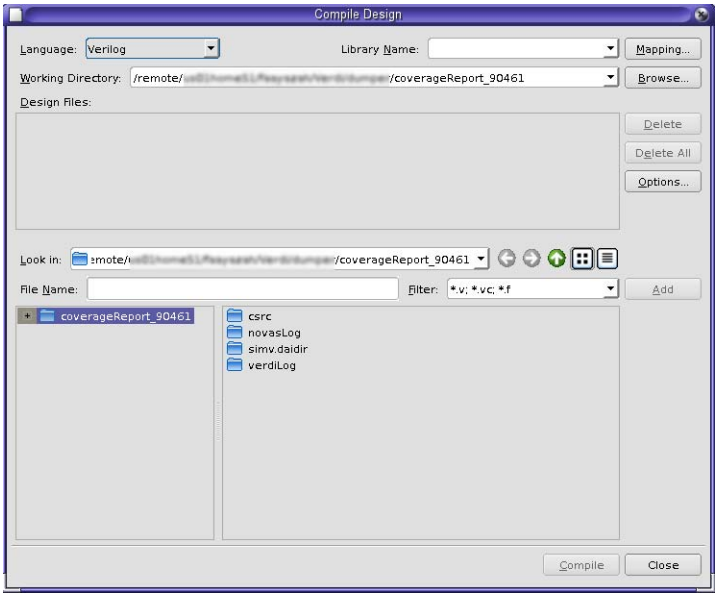


Figure: Compile Design Form

Language: Select the design language to be compiled using the Language options: **Verilog, Verilog-2001, SystemVerilog, SystemVerilog-2009, SystemVerilog-2012, VHDL-87, VHDL-93**, **VHDL-2000**, or **VHDL-2008**.

Library Name: Specify the library names and their library paths by clicking Mapping button on the right side.

Working Directory: Input the working directory of the library in the **Working Directory** text field.

Design Files: Displays the Verilog files to be compiled into the Verdi platform. Add or delete files in the **Design Files** list by selecting the file and clicking **Add** or **Delete**.

Filter: This is the file format filter for the **File List**. Additional filters can be added in this selection field using a semicolon (;) as the separator.

Options: Clicking the **Options** button opens the *Compile Design Options* form.

Compile Design with Verilog Options
If Verilog is selected in the **Language** option of the *Compile Design* form, the following options are displayed and serve as command line options when design files are compiled into the Verdi platform.

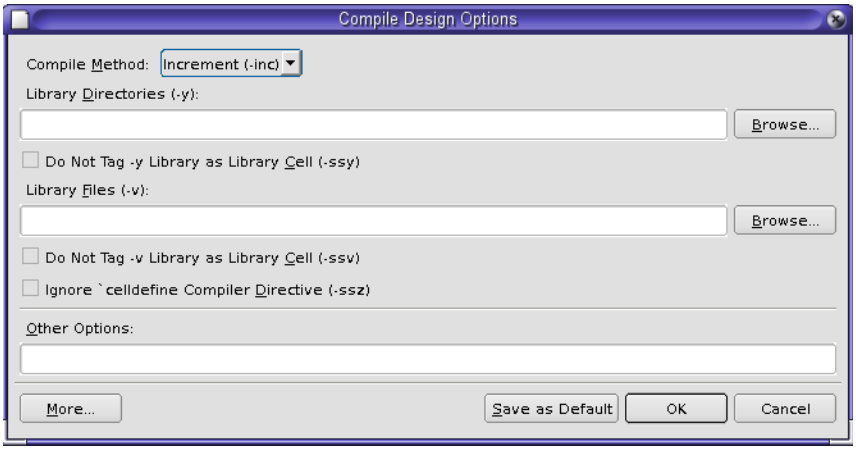


Figure: Compile Design Options Form (Verilog)

Compile Method: Choose to either incrementally add the compiled Verilog modules into the library (**Increment (-inc)**) or remove the previously saved Verilog modules (**Replacement (-rep)**).

Library Directories (-y): Specify the library directory, which appends on the command line option **-y**. Type in the directory path text field or click the **Browse** button to view the directory structure and specify a library path.

Do Not Tag -y Library as Library Cell (-ssy): When this option is turned *on*, library modules in the library directory (**-y**) is not automatically tagged as library cells.

Library Files (-v): This text field is used to specify the library file, which appends on the command line option **-v**. Either type in the library file text field or click the **Browse** button to view the directory structure and specify a library file.

Do Not Tag -v Library as Library Cell (-ssv): When this option is turned *on*, library modules in the library file (**-v**) is not automatically tagged as library cells.

Ignore 'celldefine Compiler Directive (-ssz): When this option is turned *on*, *'celldefine* compiler directives is ignored.

Other Options: See the supported Verilog reference manual for available options. For these options, invoke the Verdi platform with the command option **-h**.

More: Click this button for more compile options as follows:

- **Run as Mentor (-mentor):** This option works with the Mentor simulator.

- **Compile as NC Library Format (-nclib):** This option uses the NC binding scheme. With this option, *vericom* looks for *libMap, workLibrary, viewMap*, and *defaultView* as specified in the `[NC_Sim]` section of the *novas.rc* resource file for compiling Verilog source files into the Verdi Knowledge Database (KDB).

  If the **Compile as NC Library Format (-nclib)** option is turned *on*, the **NC Library Name (-work)** and **NC View Name (-view)** options is turned *on*.

  - **NC Library Name (-work):** Specify the NC library name (valid if **-nclib** is specified).

  This option provides the same function as *ncvlog -work library_name* to use the specified library name for the design units in the input Verilog source files. For example:

  ```
  vericom -nclib -work novas -view novas -f ../src/novas.f
  ```

  All design units in the file *../src/novas.f* are compiled into the library *novas* and have a view name *novas*.

NOTE: Using the command line option, **-work**, overrides the *workLibrary* and *libMap* variables in *novas.rc*.

  - **NC View Name (-view):** Specify the NC view name (valid if **-nclib** is specified).

  This option provides the same function as *ncvlog -view view_name* to use the specified view name for the design units in the input Verilog source files. For example:

  ```
  vericom -nclib -view novas ../src/novas.v
  ```

  All design units in *../src/novas.v* have a view name *novas*.

NOTE: Using the command line option, *-view*, overrides the *defaultView* and *viewMap* variables in *novas.rc*.

- **Extract RTL (-extractRTL):** This option automatically recognizes RTL storage elements. The extracted storage elements are saved into the library.
- **Specify novas.rc (-rcFile):** Specify the resource file - *novas.rc*.

Set as Default: Click this button to save all the settings as default compile options. You can then click the **Compile** button right away to compile the design with the default options just set.

Compile Design with VHDL Options

If VHDL is selected in the **Language** option of the *Compile Design* form, the following options is displayed and serve as command line options when design files are compiled into the Verdi platform.

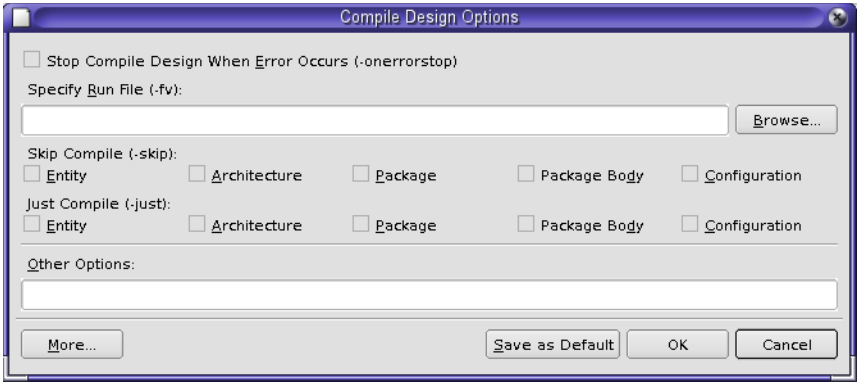


Figure: Compile Design Options Form (VHDL)

Stop Design Compile when Error Occurs (-onerrorstop): This option stops compiling the design if an error occurs.

Specify Run File (-fv): If many VHDL files are to be compiled, pack them sequentially into a pack file and pass this file to *vhdlcom*. For example, if you have three source files called *v1.vhd*, *v2.vhd* and *v3.vhd*, you can create a file called *run.f* (or whatever name you wish to use) and create its contents as follows (note that each file name must occupy its own line):

```
v1.vhd
v2.vhd
v3.vhd
```

Skip Compile (-skip): You can specify the compiler to skip. You can skip **Entity**, **Architecture**, **Package**, **Package Body**, or **Configuration** structures.

Just Compile (-just): You can specify the compiler to compile. You can just compile **Entity**, **Architecture**, **Package**, **Package Body**, or **Configuration** structures.

Other Options: See the supported VHDL reference manual for available options. For these options, you can invoke the Verdi platform with the command option *-h*.

More: Click this button to get more compile options as follows:

- **Compile All Entities (architecture) as Symlib (-libcell):** Compiles all entities (architectures) the same as the symbol library.
- **Verbose Messages (-verbosemsg):** Verbose mode. Displays status messages to the screen.
- **Preload Basic Library (-prelib):** If **Yes** is specified, *vhdlcom* loads two standard packages (*standard* and *textio*) before parsing the design (the default value is *Yes*).
- **Extract RTL (-extractRTL):** This option automatically recognizes RTL storage elements. The extracted storage elements are saved into the library.
- **Specify novas.rc (-rcFile):** Specifies the resource file - *novas.rc*.

NOTE:

1. To save both Verilog/VHDL options, click **Save as Default** twice for both language types.
2. After compiling, a message, "`You may have to reload design after compilation if library is the same.`" is shown to notify that reloading the design may be necessary if compiling something into the same library which may cause bad behavior in the Verdi platform. If a design is not currently imported into the Verdi platform, the message is not shown. Also, the message can be turned *off* on the **Tools -> Preferences -> General** page.
3. If the Verdi platform exits, the compile language type (Verilog, Verilog-2001, VHDL-87, VHDL-93, and VHDL-2000) is saved to *novas.rc* as the default language for compiling the design for the next time the Verdi platform runs.

Open File

Menu Bar: File -> Open File

This command opens the *Open File* form where one or more files can be selected to be loaded in the *nTrace* window.

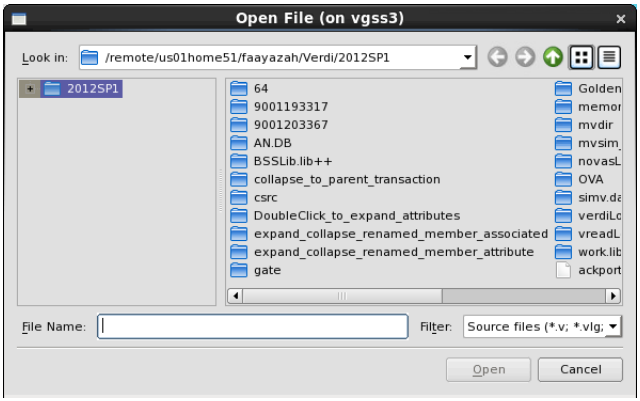


Figure: Open File Form

The *Open File* form includes the following fields, buttons, and options:

- **File Name:** This field displays the FSDB files to be opened in *nTrace*.
- **Filter:** Select the file format filter for the file list.

Open Waveform File

Menu Bar: File -> Open Waveform File

This command opens the *Load Simulation Results* form where simulation results for Active Annotation™ can be loaded in the source code or schematic frame. After specifying the options, click **OK** to load the simulation results, or click **Cancel** to close the *Load Simulation Results* form without any changes.

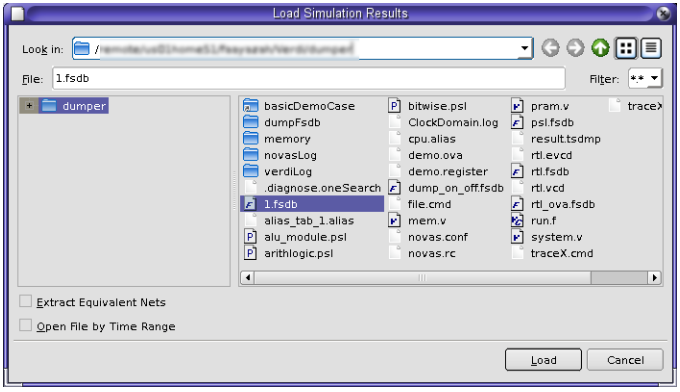


Figure: Load Simulation Results Form

The Verdi platform performs an FSDB (Verdi's simulation result file) conversion for the following file types: VCD dump file and ViewSim. Before the conversion of FSDB files, the Verdi platform checks for write permission and overwrites existing files. Only digital simulation results can be loaded using this form.

The command status is shown in the *Message* frame. For example:

```
File /myproject/demo/verilog/rtl/rtl.fsdb is loaded
```

If loaded successfully, the simulation results automatically apply to the *nWave* window when invoking the **Tools -> New Waveform** command. The *Load Simulation Results* form performs the same function as the **File -> Open** command in the *nWave* window and automatically shows G1 in the *nWave* signal pane.

The Verdi time unit is determined in the following priority order:

1. The setting of the **Waveform -> Waveform Time -> Set Window Time Unit** command or the  icon when a waveform frame is opened.
2. The setting of `windowTimeUnit` in the *novas.rc* resource file if the waveform is not opened or the opened waveform does not include the time unit setting.
3. The time unit of the FSDB file specified in the **Open Waveform File** command.

NOTE: For more information on how to generate a simulation result file (**.fsdb*) using Novas object files for FSDB dumping, refer to the *Linking Novas Files with Simulators and Enabling FSDB Dumping* document.

The *Load Simulation Results* form includes the following options and field:

- **Extract Equivalent Nets:** When this option is turned *on*, the equivalent signals of every signal stored in the FSDB file are found and added to the signal list. For example, if a signal *system.clock* is in the FSDB file, the equivalent signals *system.i_cpu.clock*, *system.i_cpu.i_ALUB.clock*, *system.i_cpu.i_CCU.clock*, and *system.i_pram.clock* is included. The waveforms associated with these signals are now available, since they are the same as *system.clock*. If this option is turned *off* and only *system.clock* was stored in the FSDB file, the waveforms associated with the other signals are not available.
- **Filter:** This is the file format filter for the file list in the right pane. The default file extensions in the **Filter** field are * and **.fsdb*; **.fsdb.gz*; **.fsdb.bz2*;**.ff*; **.dump*. Additional filters can be added in this selection field using a semicolon (;) as the separator.

- **Open File by Time Range:** When this option is turned *on*, a *Time Range of Opened File* form opens where the time range of interest can be specified in the **From Time** and **To Time** text fields. This option speeds up the loading process when you want to load a partial FSDB file.

Load Aux Annotation Waveform File

Menu Bar: File -> Load Aux Annotation Waveform File

This command appends one more waveform file in the Verdi platform so you can see active annotation and perform active trace. If a signal's value is not found in the primary waveform file, the Verdi platform automatically looks in the auxiliary waveform file for the value. After the second simulation file (waveform file format only) is specified in the form, clicking **OK** appends the waveform file as the secondary result. The *Message* frame shows `Auxiliary file <file_name> for annotation is loaded`, where the <*file_name*> is the full path file name of the secondary waveform file name.

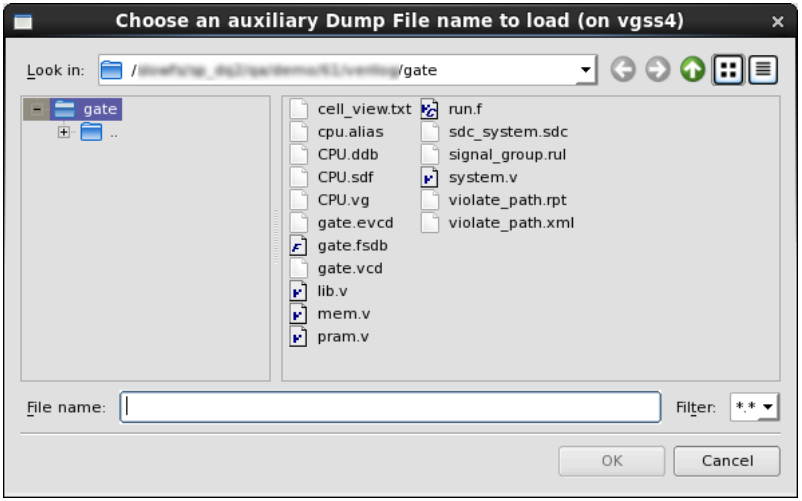


Figure: Append Simulation Result Form

Close Waveform File

Menu Bar: File -> Close Waveform File

This command closes the simulation results so that they are not loaded when you open a new *nWave* window.

Load Behavior Database

Menu Bar: File -> Load Behavior Database

NOTE: This command is available when the design is loaded from a pre-compiled library.

This command opens the *Load Behavior Database* form where a previously created behavior database can be loaded. A behavior database is created using the *bacom* utility.

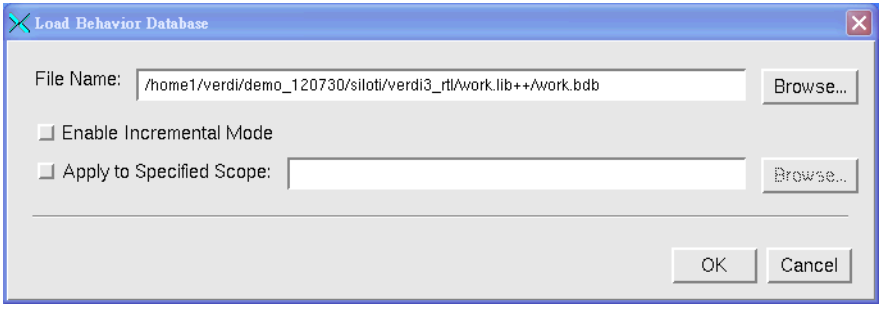


Figure: Load Behavior Database Form

The following fields and option are available:

- **File Name:** Enter the file name directly or click the **Browse** button to open the *Load BDB File* form and view the directory structure to load a Behavior Database (*.*bdb*) file.
- **Enable Incremental Mode:** When this option is turned *on,* the Behavior Database is loaded incrementally. When this option is turned *off*, the entire Behavior Database is loaded. The default value of this option is *off*.

  When the **-bdb_incr** option is specified on the command line of the *verdi* utility, this option is turned *on* automatically.
- **Apply to Specified Scope:** When this option is turned *on*, the behavior database is only applied to the specified scope. When this option is turned

off, the behavior database applies to the entire design. The default value of this option is *off*.

After turning the option *on*, specify the path to load the top module of the Behavior Database (BDB) file. Either type in the text field or click the **Browse** button to open the *Instance Manager* form where the design hierarchy can be browsed and one scope can be selected at a time. Alternatively, drag a scope from the *nTrace* design browser frame and drop it to the **Apply to Specified Scope** text field.

Import CPF/UPF Files

Menu Bar: File -> Import CPF/UPF Files

NOTE: This command is available when CPF/UPF files have already been loaded from the command line.

This command loads a CPF/UPF file. Refer to the **Import CPF/UPF Files** command description in the *Power Aware Debug* chapter for details.

Reload CPF/UPF Files

Menu Bar: File -> Reload CPF/UPF Files

Bind Key: Shift+R

NOTE: This command is available when CPF/UPF files have already been loaded from the command line.

This command reloads the current CPF/UPF files. Refer to the **Reload CPF/UPF Files** command description in the *Power Aware Debug* chapter for details.

SDF

Load SDF Files - nAnalyzer

Menu Bar: File -> SDF -> Load SDF Files

This command opens the *Import SDF* form where an SDF (Standard Delay Format) file (**.sdf*) can be imported.

NOTE: Verdi closes the loaded SDF file when a design is imported or reloaded.

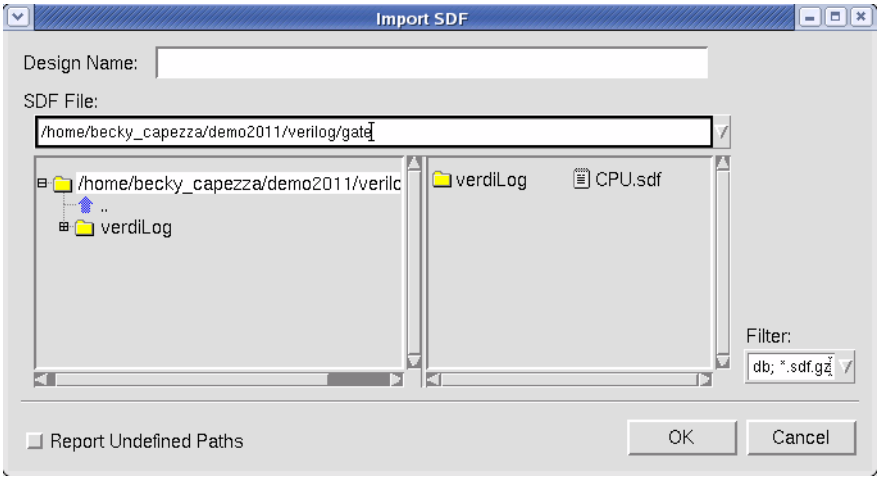


Figure: Import SDF Form

An SDF library file is created by running the *sdfin* utility, a batch mode compiler for SDF files. The *sdfin* utility parses and saves the SDF file into a **.sdb* file. The usage of *sdfin* is as follows:

```
> sdfin [-vhdl] sdfFile [sdfFile2]
```

Refer to the *sdfin* section in the *Utilities* chapter in this manual and the *nAnalyzer User's Guide and Tutorial* for details.

The *Import SDF* form includes the following field and option:

- **Design Name:** This text field identifies the region or scope of the design for the cell entry that contains timing data. The design name does not need to be specified if the timing data is for the top module. If the timing data is only for a lower-level design, to view the SDF annotation (when the **Schematic -> SDF Annotation** option is turned *on*) for the instance in *nSchema,* the full hierarchical instance name must be in this text field. For example, if the cell entry starts from the *system.i_cpu.i_alub*, `system.i_cpu.i_alub` should be specified in the **Design Name** text field.

  To map the SDF file correctly to the design, the following two methods are available:

1. Loading the design from the `system` level testbench. Use the SDF virtual top and specify the complete hierarchical instance name in the **Design Name** text field.

In an SDF file with the following design hierarchy in the header:

```
(DELAYFILE
    (SDFVERSION "sdf 3.0")
    (DESIGN "system.i_cpu")
```

Enter `system.i_cpu` in the **Design Name** text field.

2. Loading the design from a sub-module of the CPU (that is, the *ALUB* module with instance *i_ALUB*). Use the design virtual top.

In an SDF file with the following design hierarchy in the header:

```
(DELAYFILE
    (SDFVERSION "sdf 3.0")
    (DESIGN "system.i_cpu")
```

Create a map file (that is, map.dat) that includes `ALUB = system.i_cpu.i_ALUB` in the **Design Name** text field. Then, load the map file with the design.

- **Report Undefined Paths:** When this option is turned *on*, any cell/path delays exist in an SDF file that does not match the current design is reported. The default value of this option is *off*.

Close SDF Files - nAnalyzer

Menu Bar: File -> SDF -> Close SDF Files

This command closes the SDF file that were previously loaded by the **Load SDF Files** command.

View SDF Log - nAnalyzer

Menu Bar: File -> SDF -> View SDF Log

This command opens the *sdf.log* file in a *Text Viewer* frame. The *sdf.log* file contains the summary of import status of the SDF file.

Report Undefined Paths - nAnalyzer

Menu Bar: File -> SDF -> Report Undefined Paths

This command checks whether any cell/path delays exist in the SDF file that do not match the current design.

Import Path Data Files - nAnalyzer

Menu Bar: File -> Import Path Data File

This command opens the *Import Report File* form where a timing report file (**.xml* extension) can be imported.

NOTE: More than 20 different types of reports can be generated from Primetime timing analysis, such as report_annotated_delay, report_cell, report_timing, report_constraint, report_delay_calculation, and so on. The Verdi platform with the *nAnalyzer* module only accepts the reports that were created with report_timing and report_constraint from Primetime.

NOTE: The Verdi platform does not extract all of the information from a Primetime report.
The following list of information is extracted:
- Start point & End point
- Incremental Delay & Accumulated delay
- Rising & Falling
- Data Arrival Time & Data Require time
- Clock network delay
- Clock uncertainty delay
- Library setup time
- Slack

The following list of information is ignored:
- Capacitance
- Transition time
- Fan In/Out number
- Any item that we does not list in "Extract"

The timing report file can be imported directly from the original report file or from the converted XML file which can be created from tool outputs by using the conversion utilities (that is, *pt2Xml.pl*, *AMBIT.pl*, *astrol2Xml.pl*, *DC.pl*, *EinsTimer.pl*, *HAL2Xml.pl*, *magma.pl*, *pearl.pl*, *RTLC2Xml.pl*, and *SE2Xml.pl*). Refer to the *Timing* section in the *Utilities* chapter for details.

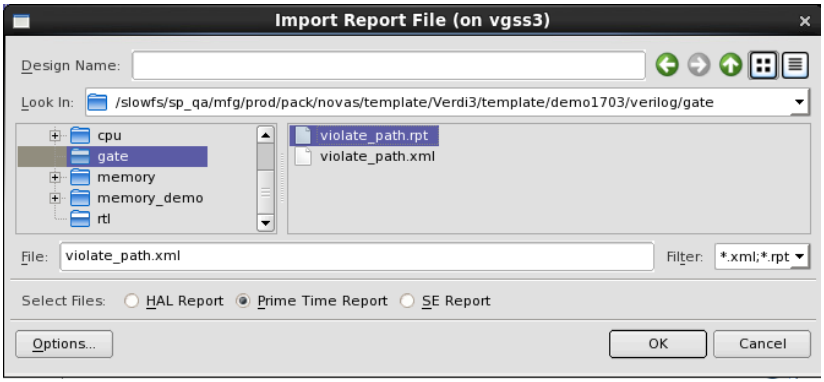


Figure: Import Report File Form

The *Import Report File* form includes the following fields and options:

- **Design Name:** Specify the design name in this field and the root path of the design name is concatenated to each element in the timing report.
- **Select File:** Select an import file type from the selections of **HAL Report**, **Prime Time Report** and **SE Report**. The default value of this option is *Prime Time Report*.
- **Filter:** Use this dynamic selection field for the file format filter of the file list. The default file extensions are **.xml* and **.rpt*. The desired filter can be added by using a semicolon (;) as the separator.
- **Filter:** After the **Prime Time Report** option is selected, the **Filter** option can be turned *on*. Clicking the **Filter** button opens the *Filter* form where viewing of the display area in the *Timing Report* form can be configured. Refer to the *Filter Form* for details.
- **Directory for Converting XML:** When this option is turned *on*, the *Set the Directory to Converting XML File* form is opened so the directory for the converted XML file can be set.

After the timing report output file is read, the path information is displayed in the *Timing Report Form* where the path to trace can be selected.

The Prime Time Report (PTR) file (**.rpt* extension) can be loaded directly. The PTR file is automatically translated to an XML timing report with the *pt2Xml.pl* utility. A sample PTR file is shown below:

```
Design: src_pci
Version: 1999.05-PT2.1
Date: Oct 15 20:13:18 2000
Delim: \
****************************************************************************
StartPoint: CORE/LI/CCW/BF_RingNotAvailQF1_reg
EndPoint: CORE/LI/CCW/BF_FR_Cntr_reg_5_
```

```
Path Group: DSK/Qclk
Point: CORE/LI/CCW/BF_RingNotAvailQF1_reg/CP     Incr: 0.00    Path: 2.13
Point: CORE/LI/CCW/BF_RingNotAvailQF1_reg/Q      Incr: 0.49    Path: 2.62
Point: CORE/LI/CCW/BF_RingNotAvailQF1            Incr: 1       Path: 0.02
Point: CORE/LI/CCW/BF_WRMUX_U19/ZN               Incr: 0.61    Path: 3.23
Point: CORE/LI/CCW/BF_WriteRt                    Incr: 1       Path: 0.06
Point: CORE/LI/CCW/U3199/Z                       Incr: 0.26    Path: 3.50
Point: CORE/LI/CCW/n2080 (net)                   Incr: 9       Path: 0.40
Point: CORE/LI/CCW/BF_FR_U213/ZN                 Incr: 0.36    Path: 3.86
Point: CORE/LI/CCW/BF_FR_Cntr123_5_              Incr: 1       Path: 0.02
Point: CORE/LI/CCW/BF_FR_Cntr_reg_5_/D           Incr: 0.00    Path: 3.86
Arrive Time: 0.38
*****************************************************************************
StartPoint: CORE/LI/CCW/BF_RingNotAvailQF1_reg
EndPoint: CORE/LI/CCW/BF_FR_Cntr_reg_5_
Path Group: DSK/Qclk
Point: CORE/LI/CCW/BF_RingNotAvailQF1_reg/CP     Incr: 0.00    Path: 2.13
Point: CORE/LI/CCW/BF_RingNotAvailQF1_reg/Q      Incr: 0.49    Path: 2.62
Point: CORE/LI/CCW/BF_RingNotAvailQF1            Incr: 1       Path: 0.02
Point: CORE/LI/CCW/BF_WRMUX_U19/ZN               Incr: 0.61    Path: 3.23
Point: CORE/LI/CCW/BF_WriteRt                    Incr: 1       Path: 0.06
Point: CORE/LI/CCW/U3199/Z                       Incr: 0.26    Path: 3.50
Point: CORE/LI/CCW/n2080 (net)                   Incr: 9       Path: 0.40
Point: CORE/LI/CCW/BF_FR_U213/ZN                 Incr: 0.36    Path: 3.86
Point: CORE/LI/CCW/BF_FR_Cntr123_5_              Incr: 1       Path: 0.02
Point: CORE/LI/CCW/BF_FR_Cntr_reg_5_/D           Incr: 0.00    Path: 3.86
Arrive Time: 0.38
*****************************************************************************
....
```

NOTE:

1. **Design, Version, Date, Delim, StartPoint, EndPoint, Path Group, Point, Arrive Time, Incr,** and **Path** are the key words in PTR format.
2. Paths are separated by a line containing only "*"s.

Timing Report Form

After clicking the **OK** button in the *Import Report File* form, the *Timing Report* form opens.

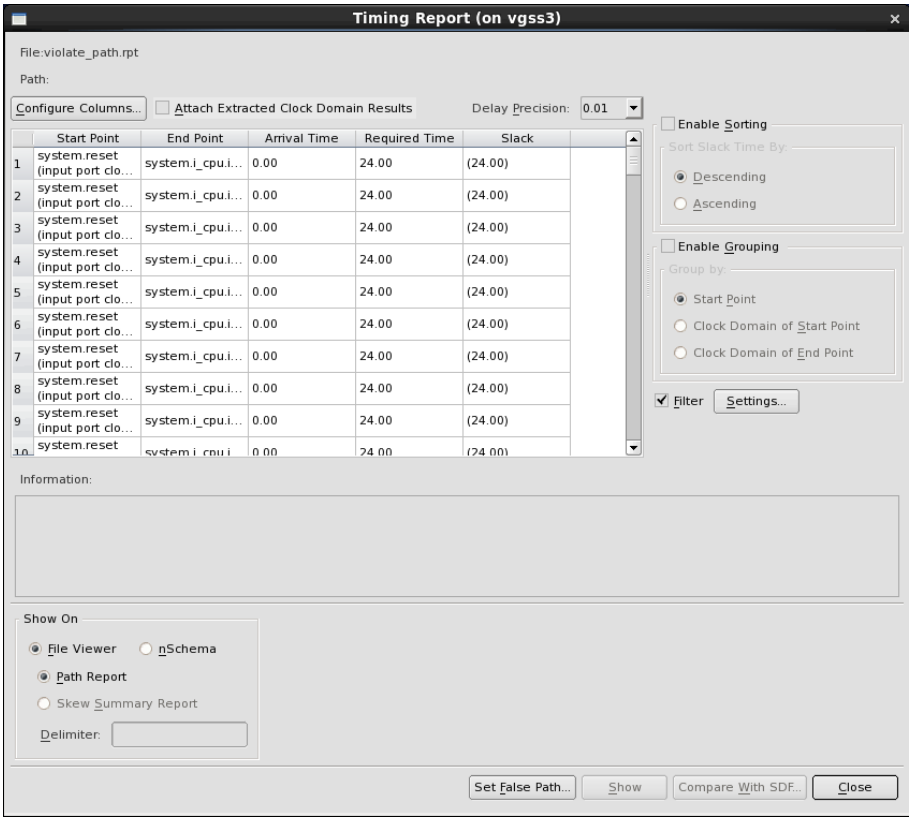


Figure: Timing Report Form

The main display area in the *Timing Report* form shows the paths from static timing analysis (STA) reports. If a path has a timing violation, the path is highlighted in red.

Five default columns are available in the main display area. The width of each column can be adjusted by placing the cursor over the vertical bar in the heading row and then pressing and holding the left mouse button. The column headings are summarized as follows:

- **Start Point:** Indicates the point where the timing path starts.
- **End Point:** Indicates the point where the timing path ends.
- **Arrival Time:** Indicates the time when the path arrives at the end point.
- **Required Time:** Indicates the expected time from the start point to the end point of a path.

- **Slack:** Indicates the difference between the analyzed time and the require time.

The following two column headings can be optionally added with the **Attach Extracted Clock Domain Results** option:

- **Clock Source of Start Point:** Indicates the point where the clock source starts.
- **Clock Source of End Point:** Indicates the point where the clock source ends.

If an SDF file is already loaded, the following three column headings is displayed in the **Timing Report** table. Click the **Configure Columns** button to view them.

- **Insertion Delay of Start Point:** Indicates the insertion delay of the specified path's start point.
- **Insertion Delay of End Point:** Indicates the insertion delay of the specified path's end point.
- **Clock Skew:** Indicates the difference between the insertion delay of the end point and the insertion delay of start point.

The *Timing Report* form includes the following options:

- **Configure Columns:** Click this button to customize the display in the main display area of the *Timing Report* form. Refer to the *Configure Columns Form* section for details.
- **Attach Extracted Clock Domain Results:** After clock domains in the *nTrace* or *nSchema* windows are extracted and this option is turned *on*, the **Clock Source of Start Point** and **Clock Source of End Point** information is available for viewing on the main display area of the *Timing Report* form. Click the **Configure Columns** button to view the **Clock Source of Start Point** and **Clock Source of End Point** columns. If clock domains are not extracted before enabling the **Attach Extracted Clock Domain Results** option, a message prompting to extract clock domains is issued.
- **Delay Precision:** Select a preferred time precision from 0.1, 0.01, 0.001, and 0.0001. The default value of this option is *0.01*.
- **Enable Sorting:** When this option is turned *on*, paths can be sorted by ascending slack time or descending slack time. By default, the paths are sorted by **Ascending** slack time.
- **Enable Grouping:** When this option is turned *on*, paths can be sorted by group. The paths can be grouped by **Start Point**, **Clock Domain of Start Point**, or **Clock Domain of End Point**. Only one option can be selected at a time. If the **Enable Sorting** option is also turned *on*, the sorting order is by group first and then the paths is sorted by slack time within the group.

- **Filter:** Click this button to open the *Filter* form where viewing of the main display area in the *Timing Report* form can be customized. Refer to the *Filter Form* section for details.
- **Information:** This section displays the original timing report information.

The **Show On** section has the following options. Select one or more timing paths (either drag-left or ctrl-left-click for multiple selections) in the main display area of the *Timing Report* form and then click the **Show** button to view the path(s) in either text (**File Viewer** option) or schematic (**nSchema** option) format. Path(s) can also be shown by dragging the path(s) in the *Timing Report* form and dropping the path(s) to the *nSchema* window.

- **File Viewer:** Select this option to view the selected path(s) in the *Text Viewer* frame.
  - **Path Report:** When this option is turned *on*, click the **Show** button to view the total delay, delay, type, arrival time, require time and slack for the selected path(s) in the *Text Viewer* frame.
  - **Skew Summary Report:** When this option is turned *on* and an SDF file is imported, click the **Show** button to view the skew summary report for the selected path(s) in the *Text Viewer* frame.
  - **Delimiter:** Specify the delimiter separating the hierarchy names. The default is the forward slash (/) character.
- **nSchema:** Select this option to view the timing report in the *nSchema* window.
- **Set False Path:** Click this button to open the *Set False Path* form where the selected path(s) in the *Timing Report* form can be added as false path to the *Set False Path* form. False path(s) can also be added by dragging the path(s) highlighted in the *Timing Report* form and dropping the path(s) to the *Set False Path* form.

Figure: Set False Path Form

The **Type** column shows the false path type as **InstPin** (instant pin). The **From** and **To** columns refer to the start point and end point of the false path respectively.

Refer to the **Tools -> Set False Path** command in the *Crossing Paths* window for details about options in the *Set False Path* form.

- **Show:** After selecting a viewing option, click this button to display the results in the specified format.
- **Compare with SDF:** Click this button to open the *Compare with SDF* form where the SDF results for the selected path are displayed. After selecting the **Delay Scale** and **Delay Type** for the SDF file, click the **OK** button to apply the changes and close the *Compare with SDF* form. Click the **Cancel** button to close the form without making any changes.

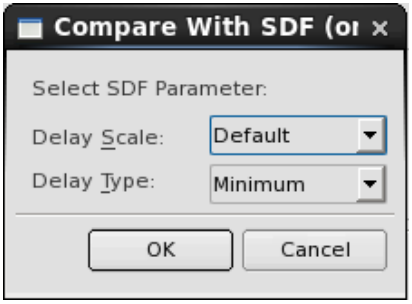


Figure: Compare with SDF Form

The *Compare with SDF* form includes the following options:

- **Delay Scale:** Specify the delay scale. Options include **Default, 10fs, 100fs, 1ps, 10ps, 100ps, 1ns**, and **10ns**.
- **Delay Type:** Specify the delay type. Options include **Minimum**, **Typical**, and **Maximum**.

Configure Columns Form

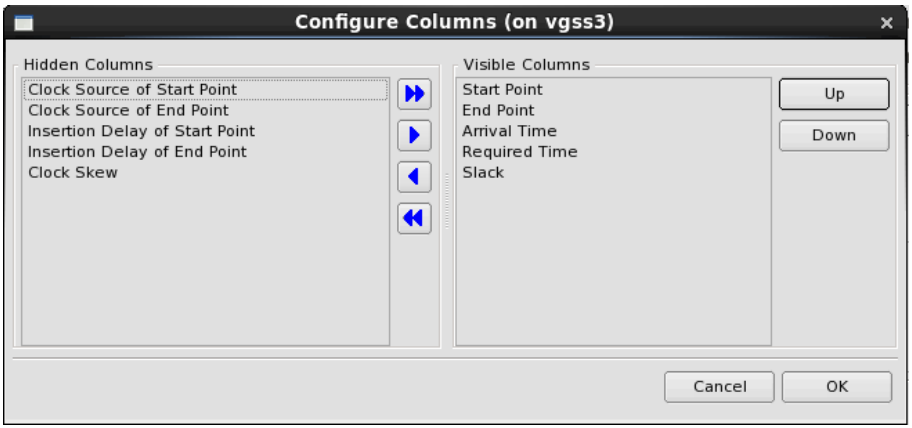


Figure: Configure Columns Form

The *Configure Columns* form has two main sections: the **Hidden Columns** which lists the columns not to display and the **Visible Columns** which lists the columns to display. Use the >> > < << buttons to specify which columns to view in the *Timing Report* form. The selections can be moved between the **Hidden Columns** and **Visible Columns**. By default, the **Start Point**, **End Point**, **Arrival Time**, **Required Time** and **Slack** columns are placed in the **Visible Columns**.

>> : This button moves all selections from **Hidden Columns** to **Visible Columns** side.

> : This button moves a single selection from **Hidden Columns** to **Visible Columns** side.

< : This button moves a single selection from **Visible Columns** to **Hidden Columns** side.

<< : This button moves all selections from **Visible Columns** to **Hidden Columns** side.

UP: Click this button to move the selected column to an upper display order. The column at the top of the list corresponds to the left-most column.

DOWN: Click this button to move the selected column to a lower display order. The column at the bottom of the list corresponds to the right-most column.

Filter Form

The **Filter** button on the *Timing Report* form opens the *Filter* form where you can customize viewing of the main display area in the *Timing Report* form through the **General**, **By Name of Start/End Point**, **By Clock Domain of Start/End Point** and **By Pass/Bypass Components** tabs for filtering the timing results.

General Tab

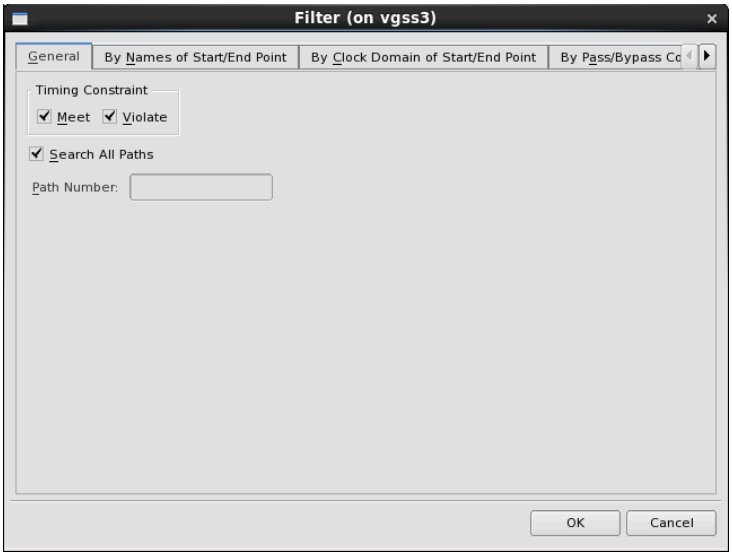


Figure: Filter Form - General Tab

The **Timing Constraint** section includes two options: **Meet** and **Violate**. By default both **Meet** and **Violate** are *on*. If **Violate** is turned *on*, only the paths including a timing violation is listed. If **Meet** is turned *on*, only the paths meeting the timing constraint is listed.

When the **Search All Paths** option is turned *on*, the report contains all possible paths. If the **Search All Paths** option is turned *off*, a value must be entered in the **Path Number** field for the number of paths to see in the report. If no value is specified, all paths is reported. The default value of this option is *on*.

By Names of Start/End Point Tab

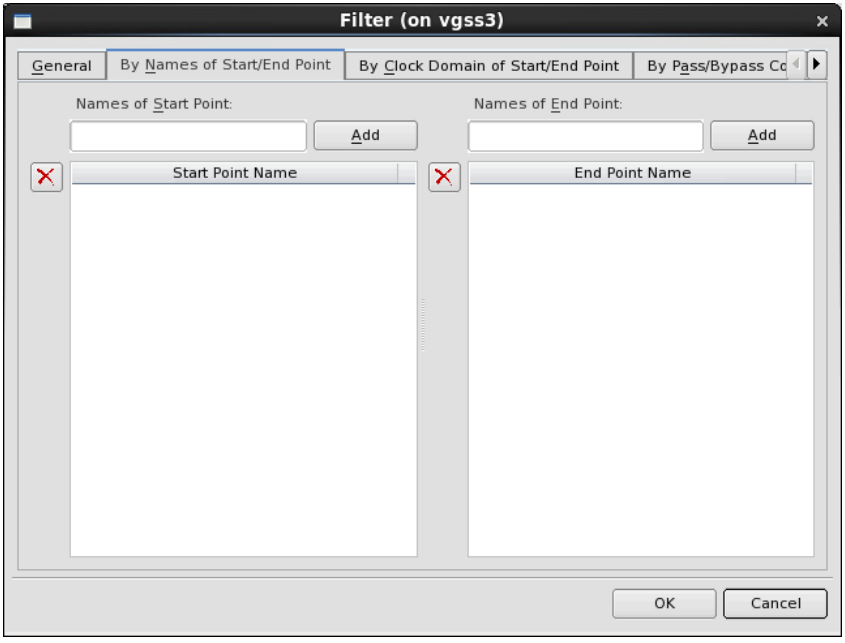


Figure: Filter Form - By Names of Start/End Point Tab

To view path(s) of the specified start point(s) or end point(s) on the **Timing Report** table, you can specify the path start point(s)/end point(s) by entering the name(s) of the start point(s)/end point(s) in the **Name of Start Point/Name of End Point** text fields. Wildcards are supported when entering a name. You can also drag the desired start/end point from the **Timing Report** table and drop it in the corresponding text field. Click the **Add** button to specify multiple filtering start/end points. To delete the current selected filtering setting, click the  button.

By Clock Domain of Start/End Point Tab

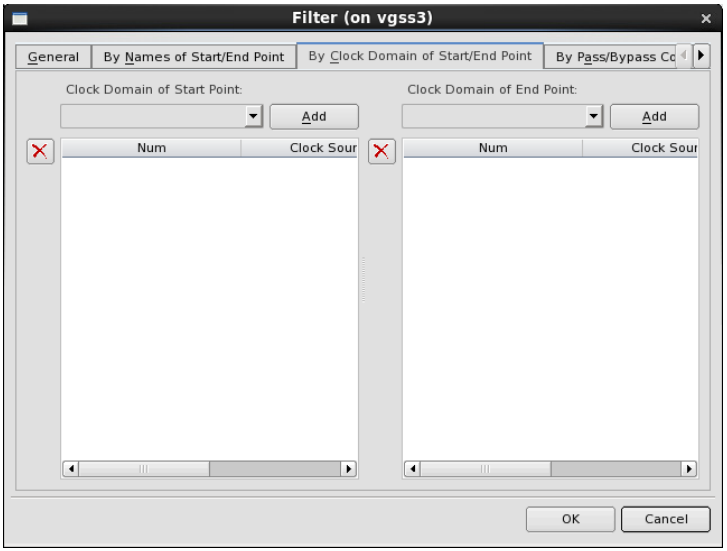


Figure: Filter Form - By Clock Domain of Start/End Point Tab

After clock domains are extracted, you can specify a clock domain for the clock domain start/end point by selecting the desired clock domain with the **Clock Domain of Start Point** /**Clock Domain of End Point** selection fields. After **OK** is clicked on the *Filter* form, the clock domain meeting the specification is displayed on the **Timing Report** table. Click the **Add** button to specify multiple clock domain start/end points. To delete the current selected filtering setting, click the  button.

Pass/By Pass Components Tab

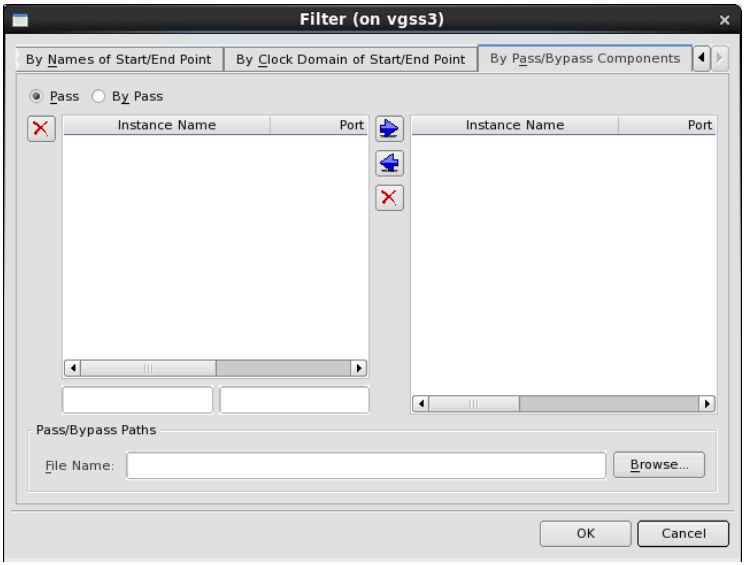


Figure: Filter Form - Pass/By Pass Components Tab

Use the **Pass/Bypass Components** tab of the *Filter* form to specify if the specified component wants to be **PASS** (gone through) or **By Pass** (stopped at or ignored). Drag a desired instance from the *nSchema* window, *nTrace* window or the **Timing Report** table and drop it in the left table. The instance name and port name of the selected instance is displayed in the left table automatically. The right table displays the instance type, pass or by pass, that you have specified. Move the instance between the left and right tables with the **Add**, **Remove**, or **Remove All** icons to set up the desired instances.

In the **File Name** text field of the **Pass/Bypass Paths** section, you can specify a path to a file name containing Pass/ByPass paths.

Import Analysis Results - nAnalyzer

Menu Bar: File -> Import Analysis Results

This command opens the *Import Report File* form where a HAL (HDL Analysis and Linting) report file can be imported.

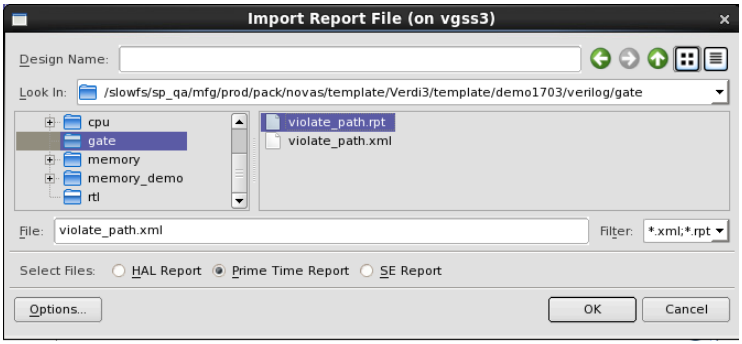


Figure: Import Report File Form

This *Import Report File* form is similar to the form opened by invoking the **Import Path Data File** command, but two differences exist:

1. The default for the **Select File** section is **HAL Report**.
2. The default file extensions in the **Filter** dynamic selection field are **.xml* and **.log*.

Refer to the **Import Path Data Files - nAnalyzer** command for details.

After clicking the **OK** button on the *Import Report File* form, the *Analysis Report* form opens.

Analysis Report Form

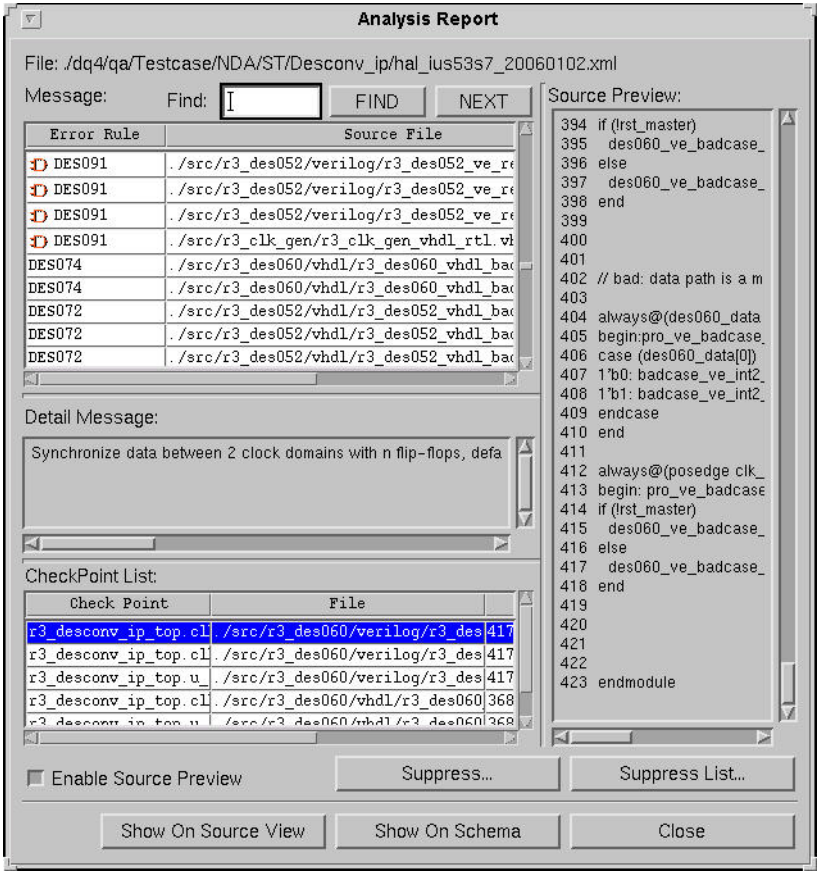


Figure: Analysis Report Form

Message

In the **Find** text field, enter the name of the error rule to find. Click the **FIND** button to locate the matching text in the **Error Rule** column. Click the **NEXT** button to go to the next line which has the same error rule name.

The message table lists the messages from the HAL report in a tabular format. Any message can be selected with a left-click anywhere along the row. The selected row is highlighted.

Five columns are available in the table. The width of each column can be adjusted by placing the cursor over the vertical bar in the heading row and then pressing and holding the left mouse button. You can sort by any column by clicking the column headers. The column headings are summarized as follows:

- **Error Rule:** Display the name of the error rule.

NOTE: The icon is displayed before an error rule in the **Error Rule** column, if the rule can be shown on the *nSchema* window. The supported rules include DES052=CMBPAU, DES060=CLKDMN, DES090=CLKDAT, DES091=CDFDAT, DES092=CAAFSR, DES151=CBPAHI, RTL041=UNCONI and RTL042=UNCONO.

- **Source File:** Display the source file.
- **Line:** Display the line number.
- **Severity:** Display the severity. It can be either **Error**, **Note**, or **Warning**.
- **Category:** Display the category. It can be either **halcheck**, **halstruct**, or **halsynth**.

Detail Message
This section display details about the selected message in the display area. It is from the HAL report.

CheckPoint List
The check point list lists the messages from the HAL report in a tabular format. Any message can be selected with a left-click anywhere along the row. The selected row is highlighted.

Four columns are displayed. The width of each column can be adjusted by placing the cursor over the vertical bar in the heading row and then pressing and holding the left mouse button. You can sort by any column by clicking the column headers. The column headings are summarized as follows:

- **Check Point:** Display the name of the module or signal.
- **File:** Display the source file.
- **Line:** Display the line number.
- **Content:** Display the content.

Source Preview
If the **Enable Source Preview** option is turned *on*, the source code from the loaded design in the *Analysis Report* form can be viewed in the **Source Preview** section. When the **Enable Source Preview** option is turned *off*, the **Source Preview** section is not displayed in the *Analysis Report* form. Below is an example of the figure when the **Enable Source Preview** option is turned *off*.

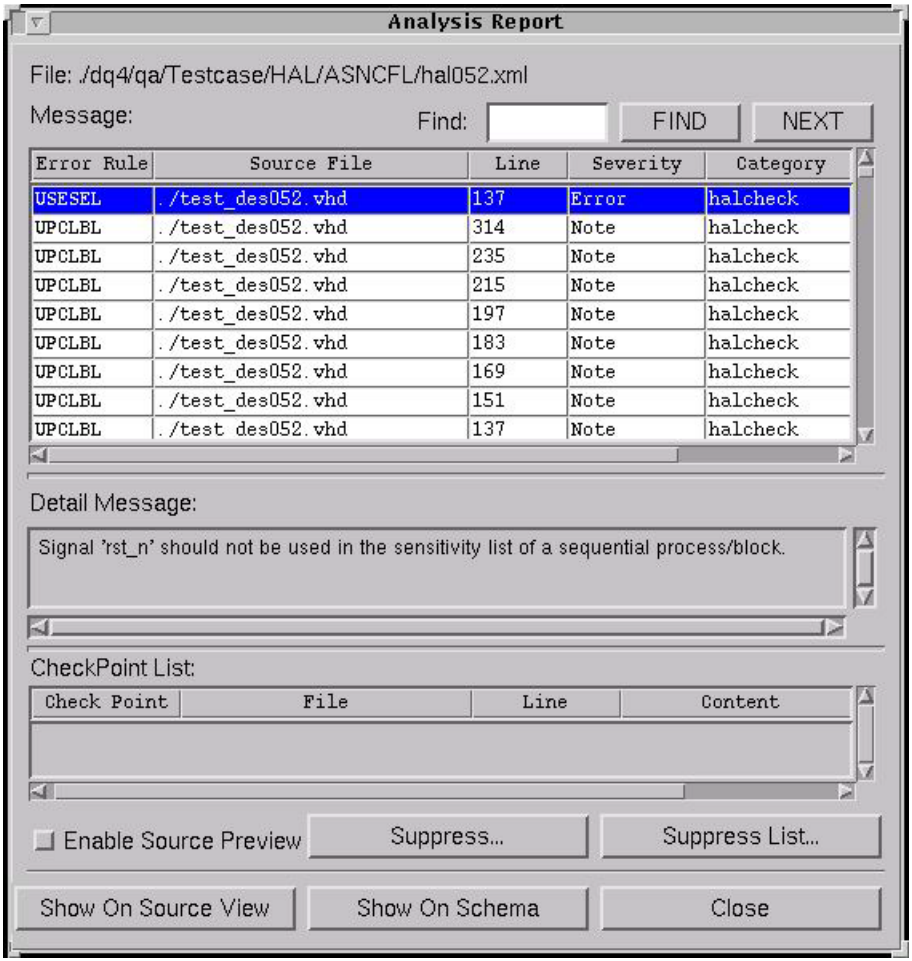


Figure: Analysis Report Form with Enable Source Preview off

Suppress: Click this button to open the *Edit* form for specifying messages to suppress. Refer to the *Edit Form* section for details.

Suppress List: Click this button to open the *Suppress List* form where the suppressed messages are displayed. Refer to the *Suppress List Form* section for details.

Show on Source View: Click this button to display the selected message in text format in a new source code frame.

Show on Schema: Click this button to display the selected message in schematic format in a new *nSchema* window.

Close: Click this button to close the *Analysis Report* form.

Edit Form

The fields in this form are automatically filled in based on a selected message. Specifying any value in any of the fields and messages matching your specifications is suppressed. Wildcard characters are not supported.

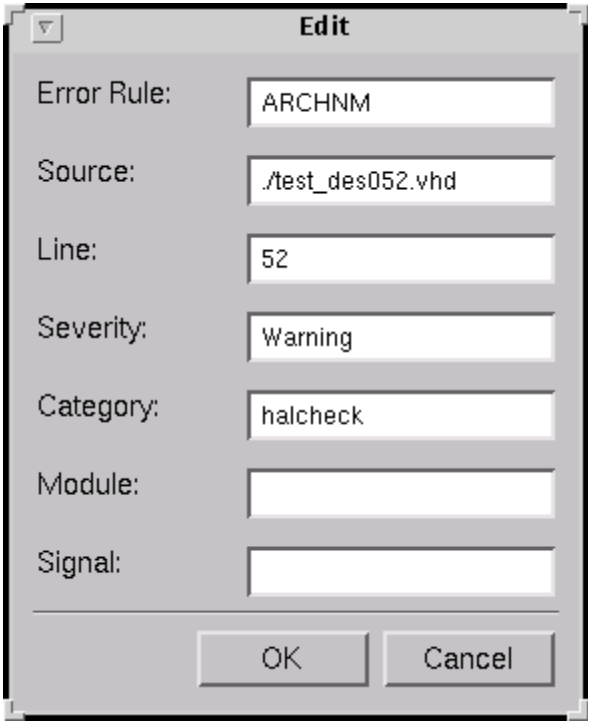


Figure: Edit Suppress List Form

The *Edit* form includes the following text fields:

- **Error Rule:** Display the name of the error rule.
- **Source:** Display the source file.
- **Line:** Display the line number.
- **Severity:** Display the severity. It can be either **Error**, **Note**, or **Warning**.
- **Category:** Display the category. It can be either **halcheck**, **halstruct**, or **halsynth**.
- **Module:** Display the module name.
- **Signal:** Display the signal name.

A workaround is to delete the line number and then rules with similar criteria are suppressed at any line of the same design file.

Suppress List Form

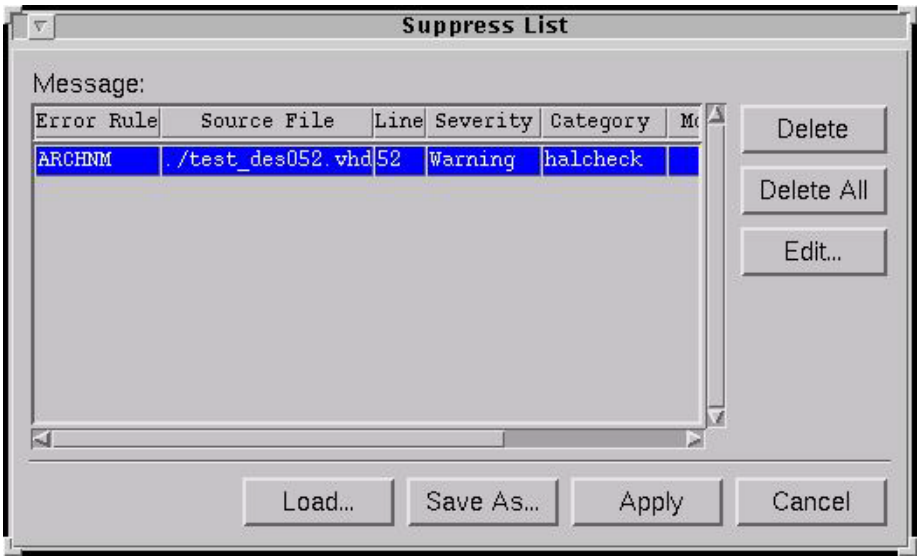


Figure: Suppress List Form

The **Message** table includes the following columns:

- **Error Rule:** Display the name of the error rule.
- **Source File:** Display the source file.
- **Line:** Display the line number.
- **Severity:** Display the severity. It can be either **Error**, **Note**, or **Warning**.
- **Category:** Display the category. It can be either **halcheck**, **halstruct**, or **halsynth**.
- **Module:** Display the module name.
- **Signal:** Display the signal name.

The *Suppress List* form includes the following options:

Delete: Click this button to delete the selected message from the *Suppress List*.

Delete All: Click this button to delete all messages from the *Suppress List*.

Edit: Click this button to open the *Edit* form.

Load: Click this button to load another suppress list file.

Save As: Click this button to open a *Save As* form where you can view the directory structure and specify a text file name for saving the *Suppress List* contents to.

Apply: Click this button to apply all of the changes in the *Suppress List* and leave the *Suppress List* form open.

Cancel: Click this button to cancel changes to the *Suppress List* of the selected message and close the *Suppress List* form.

Load Clock Domain Results - nAnalyzer

Menu Bar: File -> Load Clock Domain Results

This command opens the *Import Clock DB* form where a CDB (Clock Database) file can be loaded.

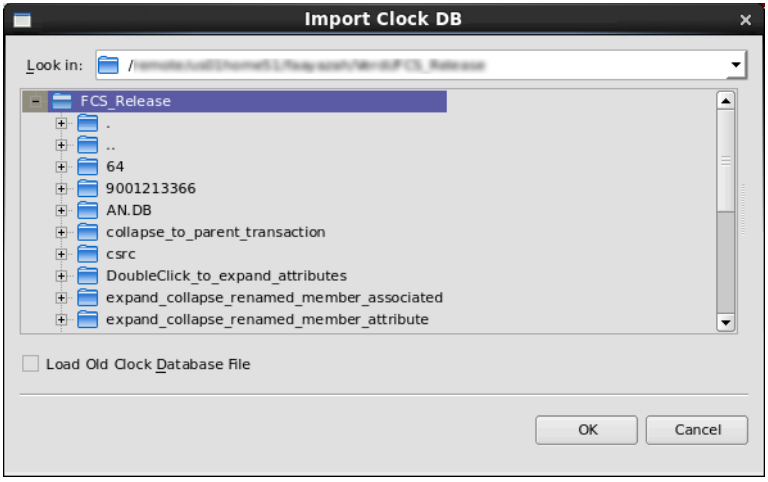


Figure: Import Clock DB Form

If the loading is successful, *nSchema* opens the *Clock Domains* window listing all clock domains. Refer to the *Clock Domains Window* section in the *Clock Analyzer* chapter for details.

Load Clock Tree Database - nAnalyzer

Menu Bar: File -> Load Clock Tree Database

This command opens the *Load As* form where the clock tree extraction results file can be loaded. If the loading is successful, the *Clock Tree Browser* window opens with the saved clock tree extraction results displayed. Refer to the *Clock Tree Browser Window* section of the *Clock Analyzer* chapter for details.

Print

Menu Bar: File -> Print

This command specifies the print options in the *nTrace Print* form which includes three tabs: **Scope**, **Basic**, and **Advanced**.

Scope Tab

The **Scope** tab includes the **Print Scope** section and the **Description** section.

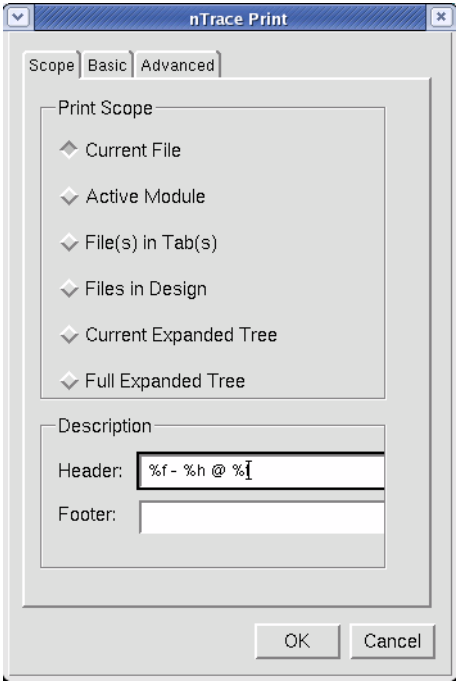


Figure: nTrace Print Form - Scope Tab

The **Print Scope** section includes the following options. One of the following options must be selected.

- **Current File:** Prints the current file.
- **Active Module:** Prints the current module only.
- **File(s) in Tab(s):** Prints the files displayed in all the source code tabs.
- **Files in Design:** Prints all the imported files.
- **Current Expanded Tree:** Prints the current expanded tree in the design browser frame.

- **Full Expanded Tree:** Prints the fully expanded tree in the design browser frame.

The **Description** section includes the **Header** and **Footer** options.

- **Header:** Defines the header on the top of a printout. By default, the header includes the file name (%f), which is currently viewed in the *nEditor* window, user name and host name (%h), and date and time (%t).
- **Footer:** Defines the footer on the bottom of a printout. Specify the text that you want to appear on the footer.

Basic Tab

The **Basic** tab includes the **Paper** section, the **Description** section and the **Other Symbol** section.

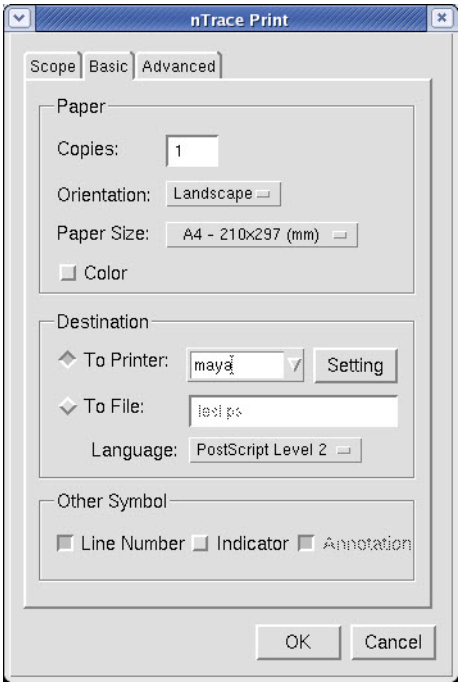


Figure: nTrace Print Form - Basic Tab

The **Paper** section includes following options:

- **Copies:** Enter the number of copies to print.
- **Orientation:** Specify the page orientation by selecting either **Landscape** or **Portrait**. The default value of this option is *Landscape*.

- **Paper Size:** Specify the supported paper size by selecting one of the following: **Letter, A4, A3, A2, A1, A0, B, C, D**, or **E**. The default value of this option is **A4**.
- **Color:** Turn this option *on* to print in color on a color printer. Otherwise, the printout is black and white when you print to a color printer.

The **Destination** section includes following options:

- **To Printer:** When this option is selected, the printed file goes to the printer specified in the text field.

  Setting: Click this button to configure the print options. First, enter the name of the printer in the **Printer Name** field. Next, select the output formats that the printer support from the **Postscript Level 1**, **Postscript Level 2**, or **HP GL/2** selections. Next, select the **Paper Size** supported by your printer. If **User-defined** is selected under **Paper Size**, the width and height of the paper must be specified.

  In the **Print Command and Options** section, the default **Command** is *lpr*, the default **Destination** is *-P*, and the default **Copies** is *-#*. Use these entries for printing your file, configure them for your own print command.

- **To File:** When this option is selected, the print file is saved to the working directory in postscript format. The file name in the text field must be specified. If this option is enabled, a section among **PostScript Level 2**, **PostScript Level 1,** or **GL2** must be specified in the **Language** selection field.

The **Other Symbol** section includes the following options:

- **Line Number:** Turn this option *on* to show the line numbers.
- **Indictor:** Turn this option *on* to show the indictor, such as trace results or bookmarks.
- **Annotation:** Turn this option *on* to show annotation on the printout.

Advanced Tab

The **Advanced** tab includes the **Advanced Option** section.

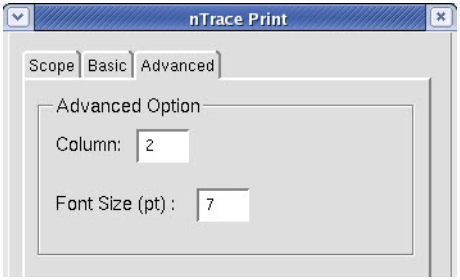


Figure: nTrace Print Form - Advanced Tab

- **Column:** Specify the number of columns (from 1 to 5) on a page.
- **Font Size (pt):** Specify the font size (from 4 to 20) to be used for the printout. If the **Annotation** option is *on*, the available font size is from 6 to 20.

Capture Window

Menu Bar: File -> Capture Window

This command is used to output an image in PNG (Portable Network Graphic format, a GNU standard similar to GIF). Initially, this command opens the *Capture Window - Select* form where the desired frame can be selected.

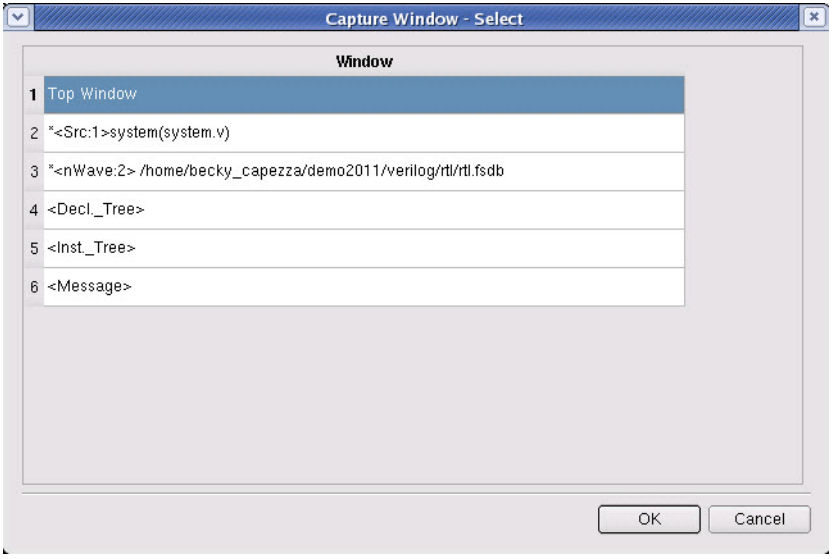


Figure: Capture Window - Select Form

After selecting the desired frame, click the **OK** button to open the *Capture Window - Preview* form.

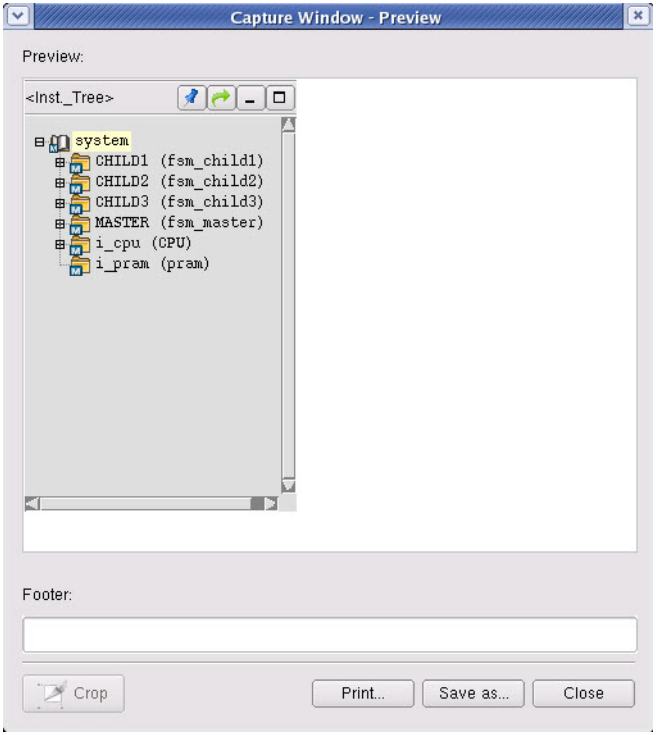


Figure: Capture Window - Preview Form

The following fields and buttons are available:

- **Footer:** Fill in the appropriate text to show on the footer.
- **Crop:** Drag the left mouse button to select the desired area and then click the **Crop** button to update the display in the *Capture Window - Preview* form.
- **Print:** This button opens a *Print* form where the desired print options for creating a hard copy can be specified.
- **Save as:** This button opens a *Save As* form where the directory structure can be viewed and a file name for the PNG file can be specified.

Restore Session

Menu Bar: File -> Restore Session

Bind Key R

This command opens a *Restore Session* form where the previous working environment, including all window locations and sizes, the debugging scope, the bookmarks set in the source code and design browser frames, the displayed signals and their order in the waveform window, the generated schematics, the testbench source code, the power design information, and some displayed information (for example, trace path of the assertion, options status, and so on) for assertions can be restored. This feature continues a previous debugging session without executing the same steps again.

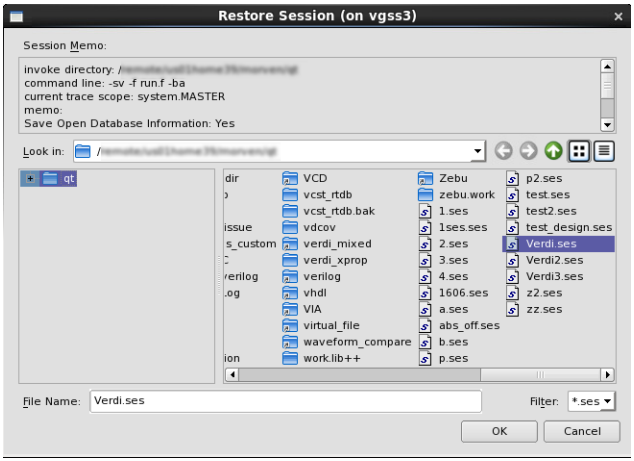


Figure: Restore Session Form

NOTE: In the **Session Memo** section, the save options information of the session file is listed as follows:
- **Save Open Database Information:** Yes or No
- **Path Option:** Absolute or Relative Paths
- **Windows Option:** All Windows or Single Waveform

Save Session

Menu Bar: File -> Save Session

This command opens the *Save Session* form where the directory structure can be viewed and the current debug session can be saved to a given session name. By default, in the **File Name** text field, the file is saved with a default name *Verdi* with the "*.ses*" file extension. The saved information includes the currently opened windows, what is displayed in a file with pointers to the design and simulation results, and the debug session enabled with the **-fastGate** option on the Verdi command line. The saved session files are shown automatically when the **Restore Session** command is invoked.

When the **Save Simulation State** option is turned *on*, the simulation state (including the time scale), debug process, and opened simulation windows are also saved.

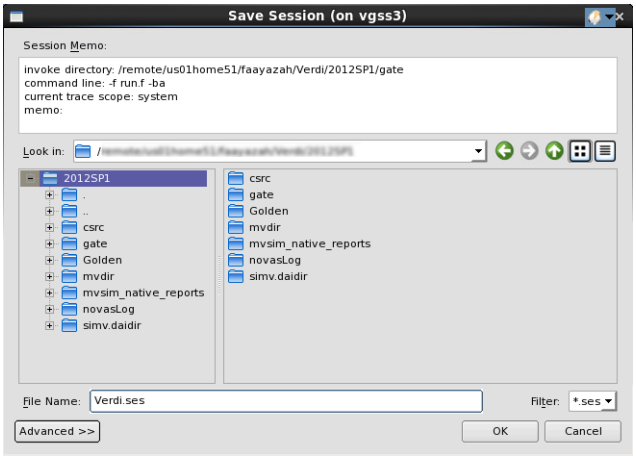


Figure: Save Session Form

Click **Advanced** to view the advanced settings as illustrated in the following figure:

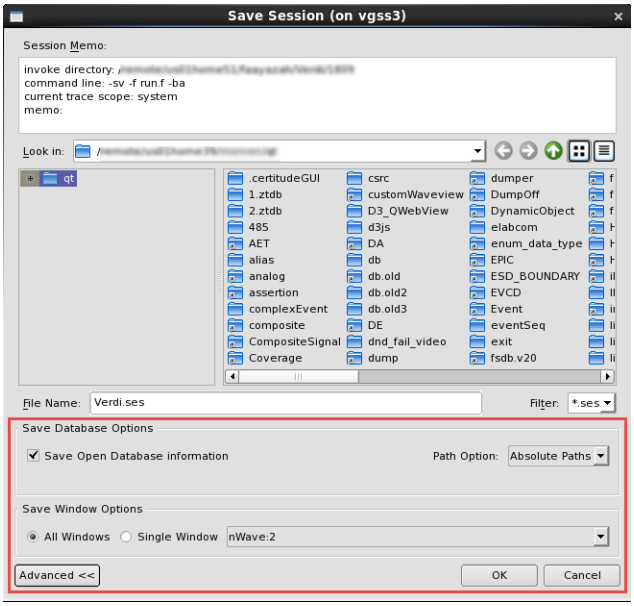


Figure: Save Session Advanced Options

For information regarding the options in the **Advanced** settings, see the *Saving a Session or Layout* section in the *Verdi SVTB Interactive Simulation Debug User Guide*.

NOTE: The current debug session is not automatically saved to the *lastSession.ses* file in the *<working_dir>/VerdiLog* or *<working_dir>/verdiLog* directory when the Verdi platform is exited unless the environment variable *NOVAS_DEB_AUTO_SESSION* is set to *1*. The name of the *VerdiLog* or *verdiLog* directory depends on whether the Verdi platform is invoked with the command *Verdi* or *verdi*, respectively. If creating a named session file in a specific location is preferred, use the **Save Session** command.

Save Design Snapshot

Menu Bar: File -> Save Design Snapshot

This command opens the *Save Design Snapshot* form where the directory structure (bottom pane) can be viewed and a directory (top text field) can be specified to save a design snapshot to. The default directory is *_SNAPSHOT*. The design snapshot includes the design, FSDB files, command line options, symbol

libraries, *novas.rc* resource file, a session file and a restore script. Click **OK** to generate a snapshot directory under the specified path. The design, FSDB files, command line options, symbol libraries, and *novas.rc* resource file are all copied and saved in the snapshot directory and a session file is created. The directory can then be used for further debug analysis and data integrity maintenance.

Creating a hard copy of the debug session is also useful when several essential signal FSDB files are generated but users cannot debug the failures before the design data base is updated. Users can use save session to create a copy of the pertinent files (including a snapshot of the design) and debug the failures at a later time.

NOTE: The design must be loaded from a compiled library; otherwise, nothing is saved.

To restore the design snapshot, change to the specified directory and execute the RESTORE script. The design, FSDB files, and window information is restored and debug can begin again.

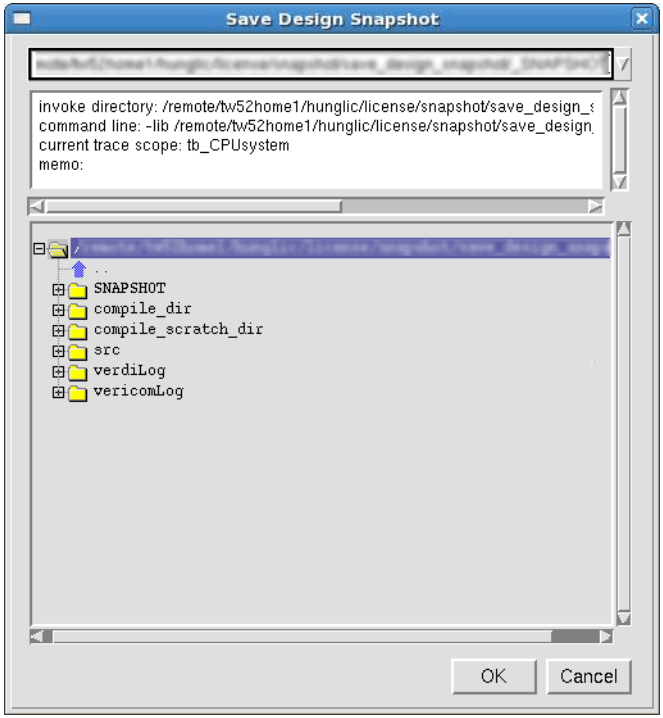


Figure: Save Design Snapshot Form

ECO

Save ECO Netlist - nECO

Menu Bar: File -> ECO -> Save ECO Netlist

Refer to the **File -> Save ECO Netlist - nECO** command description in the *nECO* chapter for details.

ECO Report - nECO

Menu Bar: File -> ECO -> ECO Report

Refer to the **File -> ECO Report - nECO** command description in the *nECO* chapter for details.

Save ECO Script - nECO

Menu Bar: File -> ECO -> Save ECO Script

Refer to the **File -> Save ECO Script - nECO** command description in the *nECO* chapter for details.

Load ECO Script - nECO

Menu Bar: File -> ECO -> Load ECO Script

An ECO script file can be loaded to see changes made to the netlist. Refer to *Appendix A: Novas ECO Script* in the *nECO User's Guide and Tutorial* for details.

Save Spare Cell Script - nECO

Menu Bar: File -> ECO -> Save Spare Cell Script

Refer to the **File -> Save Spare Cell Script - nECO** command description in the *nECO* chapter for details.

Exit

Menu Bar: File -> Exit

This command opens a *Question* form which verifies the request of closing all windows and exit the Verdi platform. Click the **Don't show this message again** option to not see this form in the future.

If the **Don't show this message again** option is selected, but you like to see this form again, click the **Enable** button in the **Disabled Messages** section in the **General** page of the *Preferences* form invoked with the **Tools -> Preferences** command.

View Commands

Source Tab

New Source Tab

Menu Bar: View -> Source Tab -> New Source Tab

Toolbar Icon: 

This command opens a blank source code tab. This tab becomes the active tab.The default name is **<Src:n>** where *n* is the next available number based on previously opened source code tabs.

Close Source Tab

Menu Bar: View -> Source Tab -> Close Source Tab

This command closes the selected source code tab.

Close Inactive Tabs

Menu Bar: View -> Source Tab -> Close Inactive Tabs

This command closes all inactive source code tabs.

Show Active Scope Tab

Menu Bar: View -> Source Tab -> Show Active Scope Tab

This command brings the active source code tab forward.

Signal List

Menu Bar: View -> Signal List

Toolbar Icon: 

This toggle command enables/disables the *Signal List* pane where the signals and signal types in each scope and variables are displayed. The icons are illustrated in the following table. The *Signal List* pane is docked to the middle pane as another tab. This is the same pane location as the *Constraint*, *Inheritance* and *FSDB_Msg* panes for testbench code. The default option is disabled.

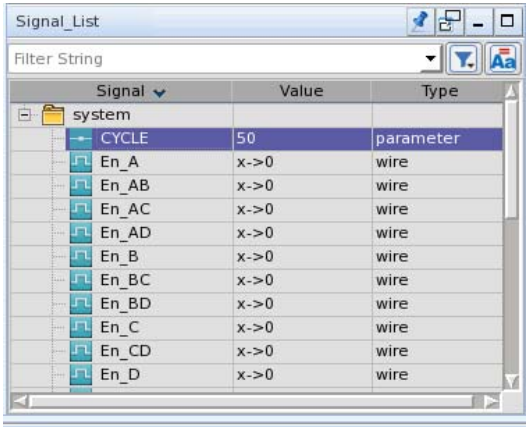


Figure: Signal List Pane

NOTE: The **Value** column is always enabled when FSDB file is loaded. It is also enabled if constant data types are available, even if the FSDB file is not loaded.

When the Active Annotation value changes (compared to the value from the previous time), the value is displayed in red.

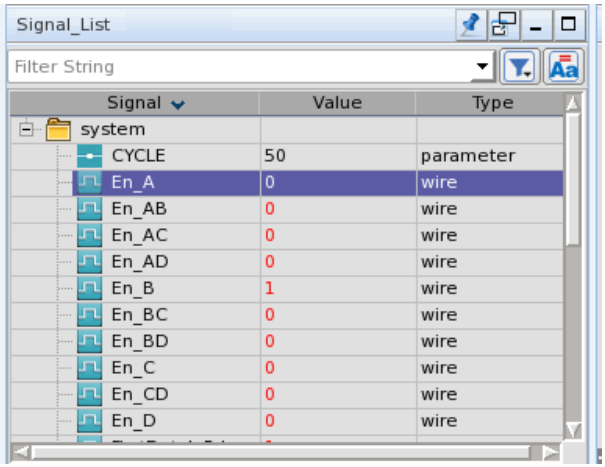


Figure: Active Annotation Displayed in Signal List Pane

When a scope is double-clicked in the design browser frame, all signals and variables in the scope is listed in the *Signal List* pane. A signal name can be typed in the text field to narrow down the filtering. The wildcard character (* and ?) are supported for signal name search.

Sorting

Each column in the *Signal List* pane can be sorted by clicking the *Signal* heading column. After sorting, the *Signal List* can be sorted as follows:

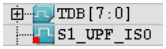
- **Ascending:** Sort signal names alphabetically by the ascending order.
- **Descending:** Sort signal names alphabetically by the descending order.
- **Declaration:** Sort signals by the definition order.

NOTE: The **Value** column cannot be sorted.
The Signal column can be sorted in ascending, descending, and declaration order.
All other columns in the *Signal List* can be sorted in ascending and descending order.

Filtering

Clicking the  icon to open the **Signal Type Filter** menu where signal type(s) can be selected for display in the *Signal List* pane. The available types include:

- **Input:** Displays input ports.
- **Output:** Displays output ports.

- **Inout:** Displays inout ports.
- **Wire/Signal:** Displays wire type signals.
- **Register:** Displays register data type signals.
- **Parameter:** Displays parameter signals.
- **Constant:** Displays constant signals.
- **Assertion:** Displays the assertion signals.
- **Buffer:** Displays the buffer signals.
- **Linkage:** Displays the linkage signals.
- **$unit:** Displays all the global signals supported in the Signal list.
- **Interface/Modport:** Displays the Interface/Modport signals.
- **Event:** Displays the event signals.
- **Isolation:** Displays Isolation signals in power designs. This type is available when a CPF/UPF file is loaded.
- **Retention:** Displays Retention signals in power designs. This type is available when a CPF/UPF file is loaded.
- **Level Shifter:** Displays Level-shifted signals in power designs. This type is available when a CPF/UPF file is loaded.
- **SRSN/SPA:** Displays SRSN/SPA signals in power designs. This type is available when a UPF file is loaded.
- **Repeater:** Displays the repeater signals. This type is available when a UPF file is loaded.
- **CSN:** Displays the CSN signals. This type is available when a UPF file is loaded.
- **ACK**: Displays the ACK signals. This type is available when a UPF file is loaded.
- **Simulation Only:** Displays simulation information in power designs. This type is available when a UPF file is loaded.
- **UPF Inserted:** Displays UPF inserted signals in the HDL design. This type is available when a UPF file is loaded. The icon of UPF inserted signal is shown with a red dot 

.
- **Other:** Displays other signal types.
- **All:** Displays all signal types.

The following table contains the icons for the signal list filter type:

Icon	Signal List Filter Type
	Input

Icon	Signal List Filter Type
	Output
	Inout
	Wire/Signal
	Register
	Parameter
	Constant
	Assertion
	Buffer
	Linkage
	Unit
	Interface/Modport
	Event
	Isolation
	Retention
	Level Shifter
	SRSN/SPA
	Repeater
	CSN
	ACK
	Simulation Only
	UPF Inserted
	Other
	All

You can also use the **Case sensitive**  icon to enable or disable case-sensitivity for the filter string and the search string entered in the **Filter/ Search** string text fields.

If the selected signal is a bus or structure signal (including SystemVerilog structs, Multi-Dimensional Arrays, and VHDL record type) and the signal is expanded, the element or partial-range signal is displayed as the child node of the signal node. For further debugging, signals can be dragged from the *Signal List* pane and dropped to the *nWave, nSchema,* or source code frames. Alternatively, double-click the signal to select the signal in the source code frame and add the signal to the *nWave* window.

Tooltip Information

To enable tooltip for the signals appearing in the *Signal List* pane, right-click a signal and select **Show Tip** option. The tooltip displays signal name, value, and type information, as illustrated in the following figure:

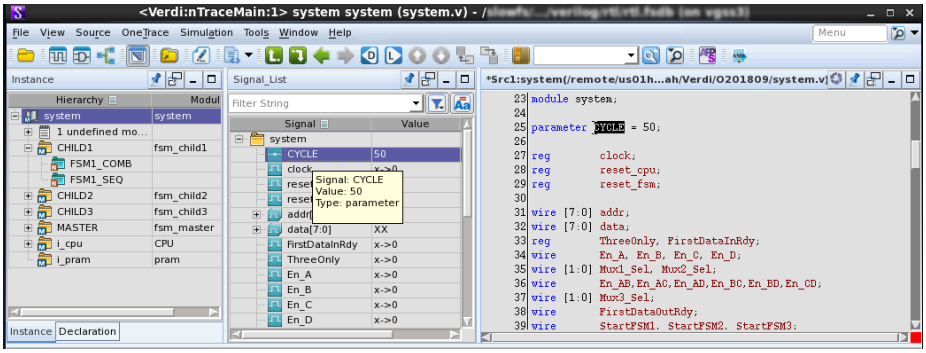


Figure: Tooltip for Signals in Signal List Pane

Maximum Number of Signals to Display in One Page

The maximum number of signals can be specified in the **Maximum Signals to Display in One Page** preference option which is available under the **Tools -> Preferences -> Source Code -> Signal List** page.

Power Information

If a CPF/UPF file is loaded, the **Power Info** column appears showing the power attributes for each signal. If an FSDB file is loaded and the **Active Annotation** command is turned *on*, the **Value** column also appears. The **Power Info** column can be sorted by clicking the triangle or upside-down triangle on the column headers.

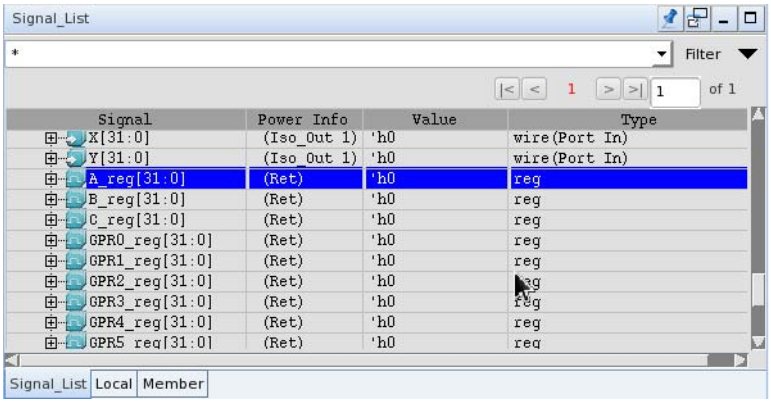


Figure: Signal List Pane With Power Information

Signal List Frame Right-Click Options

Following is the list of right-click menu commands available for the *Signal List* pane.

Show Filter

This command displays the filter text box to narrow down the searching for the signals.

Bind Key: Ctrl+F

Show Tip

This command turns the tip display *on* or *off*. When this option is turned *on* and the cursor is placed over an signal, the tip shows the signal details, that is, **Signal**, **Value**, and **Type** of the signal.

Drag

Refer to the Drag command description in the right-click command menu of the source code frame for details.

Drop

Refer to the Drop command description in the right-click command menu of the source code frame for details.

Copy Full Path

Bind Key: Ctrl+H

This command copies the full hierarchical name(s) of the selected signal(s) into the clipboard buffer.

OneTrace

Refer to the descriptions of **Value Change, Driver**, **Load**, **Connectivity**, **Chain Driver** commands under the **OneTrace** command for details.

Highlight

This command opens the *Highlight* form which enables you to highlight signals with the specified color. Refer to the **Signal -> Highlight** command description for details.

Add to Waveform

Refer to the **Add Instrumented Signals to Waveform** command description in the right-click command menu of the source code frame for details.

Copy Signal's Full Path

This command copies the full hierarchical name(s) of the selected signal(s) into the clipboard buffer.

New Schematic

Refer to descriptions of the **Fan-in**, **Fan-out**, **Driver**, **Load**, **Connectivity** commands under the **Tools -> New Schematic from Source** command for details.

NOTE: This command is disabled if the parameter type signals are selected in the **Signal Type Filter** menu where parameter signal type(s) are selected for display in the *Signal List* pane.

Temporal Flow View

Refer to descriptions of the **Auto Trace**, **Trace Glitch**, **New Temporal Flow View**, and **Add Reference Signals** commands under the **Tools -> Temporal Flow View** command for details.

Trace X

Refer to the **Trace X** command description in the under the **Tools** command for details.

Save Selected Signals to File

This command opens the *Save Selected Signals to File* form where the directory structure can be viewed and a filename can be specified. The signal(s) selected in the *Signal List* frame of *nTrace* is saved to this file.

Select All in Current Page

Bind Key: Ctrl+A

This command selects all the signals in the *Signal List* frame.

Constraint View

Menu Bar: View -> Constraint View

This toggle command enables/disables the *Constraint* frame. When testbench code is loaded, the default is enabled. Refer to the *Constraint Pane* section in the *Testbench Debug* chapter for details about this view.

Inheritance View

Menu Bar: View -> Inheritance View

This toggle command enables/disables the *Inheritance* frame. When testbench code is loaded, the default is enabled. Refer to the *Inheritance Pane* section in the *Testbench Debug* chapter for details about this view.

FSDB Message View

Menu Bar: View -> FSDB Message View

This toggle command enables/disables the *FSDB_Msg* frame. When testbench code is loaded, the default is enabled. Refer to the *FSDB Message Pane* section in the *Testbench Debug* chapter for details about this view.

Line Number

Menu Bar: View -> Line Number

This toggle command turns the display of line numbers in the source code frame *on* or *off*. The default value of this option is *on*.

Auto Wrap

Menu Bar: View -> Auto Wrap

This toggle command enables/disables the option to move the source code to the next line based on the view of the *Source Code* frame. By default, the option is enabled.

X-propagation Instrument

Menu Bar: View -> X-propagation Instrument

NOTE: This command is available when the X-propagation design and the log file have already been loaded with the **-xprop** and **-xproplog** options in the *verdi* utility.

This toggle command turns the display of the marked information for the X-propagation file in the source code frame *on* and *off*. The default value of this option is *off*.

Enable Syntax Color

Menu Bar: View -> Enable Syntax Color

This toggle command turns the display of the syntax color in the source code frame *on* and *off*. The syntax color is disabled (the command turned *off*) automatically the first time if the **Apply on Source Code Window** option is turned *on* in the form opened by invoking the **Clock Domain -> Highlight Clock Domains** command. The default value of this option is *on*.

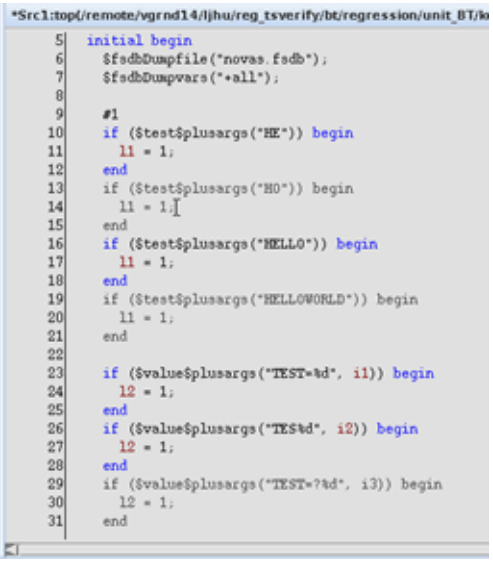
Identify False Logic

Menu Bar: View -> Identify False Logic

This toggle command turns the display of false logic in the source code frame *on* and *off*. When this command is turned *on*, false logic is displayed in gray in the source code frame after calculating expression values in the control logic. When the design is loaded again or the cursor time is changed, expression values is calculated again and new false logic is displayed in gray. When an FSDB file is loaded, *nTrace* calculates expression values, signals, function calls, and constants, and then display false logic in gray; otherwise, only constants and constant expressions are included. When this command is turned *off*, false logic is not highlighted in gray in the source code frame. The default value of this option is *off*.

NOTE: Identify false logic feature is not supported for VHDL and Verilog generate block.

NOTE: The *$test$plusargs* and *$value$plusargs* functions are supported for the **Identify False Logic** command when the FSDB file is loaded and the function argument is specified for the simulation run as shown in the figure (that is, the strings *+HELLO* and *+TEST=2* are set when running the simulation).



Source Code Folder

The **Source Code Folder** command and its sub-commands appear only when one of the **Automatic Source Code Folding** options on the **Source Code -> Code Folding** page of the *Preferences* form (invoked with **Tools -> Preferences**) is turned *on*.

Expand All in File

Menu Bar: View -> Source Code Folder -> Expand All in File

When this command is invoked, all statement, comment, and port list folders in the current file are expanded.

Collapse All in File

Menu Bar: View -> Source Code Folder -> Collapse All in File

When this command is invoked, all statement, comment, and port list folders in the current file as well as line numbers are folded, and the folded line number is marked in red.

Expand All in Design

Menu Bar: View -> Source Code Folder -> Expand All in Design

When this command is invoked, all statement, comment, and port list folders in the whole design are expanded.

Collapse All in Design

Menu Bar: View -> Source Code Folder -> Collapse All in Design

When this command is invoked, all statement, comment, and port list folders in the whole design as well as line numbers are folded, and the folded line number is marked in red.

NOTE:

1. Alternatively, click the plus (+) or minus (-) symbols to expand or collect the source code folders.
2. The folding column width is adjusted based on the maximum folding width of the current file. For example, the folding column must contain 4 folding marks if the current file has an always block that includes 3 level of statements (3+1 = 4).
3. When the start or end point of the folder cannot be seen in the present value window, a left-click on the black straight line scrolls to the out-of-sight start or end point. If both the start and end points of the folder cannot be seen, *nTrace* scrolls to start point only.
4. The folded line number color can be specified in the **Preference -> Source Code -> Color** page.

Statement Folder

The **Statement Folder** command and its sub-commands appear only when the **Automatic Statement Folding** option on the **Source Code -> Code Folding** page of the *Preferences* form (invoked with **Tools -> Preferences**) is turned *on*.

Expand All in File

Menu Bar: View -> Source Code Folder -> Statement Folder -> Expand All in File

When this command is invoked, all statement folders containing certain code statements (including if, else, case, always, for, while, loop, and process) in the current file are expanded.

Collapse All in File

Menu Bar: View -> Source Code Folder -> Statement Folder -> Collapse All in File

When this command is invoked, all statement folders containing certain code statements (including if, else, case, always, for, while, loop, and process) in the current file as well as line numbers are folded, and the folded line number is marked in red.

Expand All in Design

Menu Bar: View -> Source Code Folder -> Statement Folder -> Expand All in Design

When this command is invoked, all statement folders containing certain code statements (including if, else, case, always, for, while, loop, and process) in the whole design are expanded.

Collapse All in Design

Menu Bar: View -> Source Code Folder -> Statement Folder -> Collapse All in Design

When this command is invoked, all statement folders containing certain code statements (including if, else, case, always, for, while, loop, and process) in the whole design as well as line numbers are folded, and the folded line number is marked in red.

Comment Folder

The **Comment Folder** command and its sub-commands appear only when the **Automatic Comment Folding** option on the **Source Code -> Code Folding** page of the *Preferences* form (invoked with **Tools -> Preferences**) is turned *on*.

Expand All in File

Menu Bar: View -> Source Code Folder -> Comment Folder -> Expand All in File

When this command is invoked, all empty lines/comment lines/inactive block lines in the current file are expanded.

Collapse All in File

Menu Bar: View -> Source Code Folder -> Comment Folder -> Collapse All in File

When this command is invoked, all comment folders (including empty lines/ comment lines/inactive block lines) in the current file as well as line numbers are folded, and the folded line number is marked in red.

Expand All in Design

Menu Bar: View -> Source Code Folder -> Comment Folder -> Expand All in Design

When this command is invoked, all comment folders (including empty lines/ comment lines/inactive block lines) in the whole design are expanded.

Collapse All in Design

Menu Bar: View -> Source Code Folder -> Comment Folder -> Collapse All in Design

When this command is invoked, all comment folders (including empty lines/ comment lines/inactive block lines) in the whole design as well as line numbers are folded, and the folded line number is marked in red.

Port List Folder

The **Port List Folder** command and its sub-commands appear only when the **Automatic Port List Folding** option on the **Source Code -> Code Folding** page of the *Preferences* form (invoked with **Tools -> Preferences**) is turned *on*.

Expand All in File

Menu Bar: View -> Source Code Folder -> Port List Folder -> Expand All in File

When this command is invoked, all port/port instance list folders in the current file are expanded.

Collapse All in File

Menu Bar: View -> Source Code Folder -> Port List Folder -> Collapse All in File

When this command is invoked, all port/port instance list folders in the current file as well as line numbers are folded, and the folded line number is marked in red.

Expand All in Design

Menu Bar: View -> Source Code Folder -> Port List Folder -> Expand All in Design

When this command is invoked, all port/port instance list folders in the whole design are expanded.

Collapse All in Design

Menu Bar: View -> Source Code Folder -> Port List Folder -> Collapse All in Design

When this command is invoked, all port/port instance list folders in the whole design as well as line numbers are folded, and the folded line number is marked in red.

Sub-program Folder

The **Sub-program Folder** command and its sub-commands appear only when the **Automatic Sub-program Folding** option on the **Source Code -> Code Folding** page of the *Preferences* form (invoked with **Tools -> Preferences**) is turned *on*. Sub-program folder indicates folders for functions, tasks, or procedures.

Expand All in File

Menu Bar: View -> Source Code Folder -> Sub-program Folder -> Expand All in File

When this command is invoked, all sub-program folders in the current file are expanded.

Collapse All in File

Menu Bar: View -> Source Code Folder -> Sub-program Folder -> Collapse All in File

When this command is invoked, all sub-program folders in the current file are collapsed.

Expand All in Design

Menu Bar: View -> Source Code Folder -> Sub-program Folder -> Expand All in Design

When this command is invoked, all sub-program folders in the whole design are expanded.

Collapse All in Design

Menu Bar: View -> Source Code Folder -> Sub-program Folder -> Collapse All in Design

When this command is invoked, all sub-program folders in the whole design are collapsed.

Compiler Directive Folder

The **Compiler Directive Folder** command and its sub-commands appear only when the **Automatic Compiler Directive Folding** option on the **Source Code -> Code Folding** page of the *Preferences* form (invoked with **Tools -> Preferences**) is enabled.

NOTE: Only the ’ifdef, ’else, ’elsif, ’endif, and ’ifndef condition compiler directives are supported.

Expand All in File

Menu Bar: View -> Source Code Folder -> Compiler Directive Folder -> Expand All in File

When this command is invoked, all the unmatched compiler directive branches with multiple lines in the file are expanded.

Collapse All in File

Menu Bar: View -> Source Code Folder -> Compiler Directive Folder -> Collapse All in File

When this command is invoked, all the unmatched compiler directive branches with multiple lines in the file are collapsed.

Expand All in Design

Menu Bar: View -> Source Code Folder -> Compiler Directive Folder -> Expand All in Design

When this command is invoked, all the unmatched compiler directive branches in the design are expanded.

Collapse All in Design

Menu Bar: View -> Source Code Folder -> Compiler Directive Folder -> Collapse All in Design

When this command is invoked, all the unmatched compiler directive branches in the design are collapsed.

False Logic Folder

The **False Logic Folder** command and its sub-commands appear in the **View -> Source Code Folder** menu only when the **Automatic False Logic Folding** option in the **Source Code -> Code Folding** page of the *Preferences* form (invoked using **Tools -> Preferences**) and **Identify False Logic** (invoked using **View -> Identify False Logic** command) is enabled and then the source code folding occurs for false logic in *nTrace Source Code* frame. The **Automatic False Logic Folding** option is enabled by default and **Identify False Logic** option is disabled by default.

Expand All in File

Menu Bar: View -> Source Code Folder -> False Logic Folder -> Expand All in File

When this command is invoked, all the false logic folders in the file are expanded.

Collapse All in File

Menu Bar: View -> Source Code Folder -> False Logic Folder -> Collapse All in File

When this command is invoked, all the false logic folders in the file are collapsed.

Expand All in Design

Menu Bar: View -> Source Code Folder -> False Logic Folder -> Expand All in Design

When this command is invoked, all the false logic folders in the design are expanded.

Collapse All in Design

Menu Bar: View -> Source Code Folder -> False Logic Folder -> Collapse All in Design

When this command is invoked, all the false logic folders in the design are collapsed.

Macro Folder

The following sub-commands only appear when the **Source -> Expand Macro** command is turned *on*.

Expand All in File

Menu Bar: View -> Macro Folder -> Expand All in File

When this command is invoked, all macro folders in the current file are expanded.

Collapse All in File

Menu Bar: View -> Macro Folder -> Collapse All in File

When this command is invoked, all macro folders in the current file are folded.

Expand All in Design

Menu Bar: View -> Macro Folder -> Expand All in Design

When this command is invoked, all macro folders in the whole design are expanded.

Collapse All in Design

Menu Bar: View -> Macro Folder -> Collapse All in Design

When this command is invoked, all macro folders in the whole design are folded.

Implicit Port Folder

The following sub-commands appear only when the **Source -> Expand Implicit Port** command is turned *on*.

Expand All in File

Menu Bar: View -> Implicit Port Folder -> Expand All in File

When this command is invoked, all implicit port folders in the current file are expanded.

Collapse All in File

Menu Bar: View -> Implicit Port Folder -> Collapse All in File

When this command is invoked, all implicit port folders in the current file are folded.

Expand All in Design

Menu Bar: View -> Implicit Port Folder -> Expand All in Design

When this command is invoked, all implicit port folders in the whole design are expanded.

Collapse All in Design

Menu Bar: View -> Implicit Port Folder -> Collapse All in Design

When this command is invoked, all implicit port folders in the whole design are folded.

Hierarchy Tree by Level

Menu Bar: View -> Hierarchy Tree by Level

This command expands the selected instance in the design browser frame to the specified level: **All, 2 Levels, 3 Levels, 4 Levels** or **5 Levels**.

Sync. All Signal Selection

Menu Bar: View -> Sync. All Signal Selection

Toolbar Icon:  This icon indicates the current mode is disabled.

This command enables the global synchronization of all active frames for the selected signal or scope in any active frame by automatically enabling the **Sync. Signal Selection** icon in all frames. While this command is active, all new frames also are automatically synchronized.

NOTE: This is a switch command. After the **View -> Sync. All Signal Selection** command is activated, the next time the **View** pull-down menu is accessed, this command are replaced with its **View -> Release Signal Selection Sync.** counterpart.

Release Signal Selection Sync.

Menu Bar: View -> Release Signal Selection Sync.

Toolbar Icon:  This icon indicates the current mode is enabled.

This command disables the global synchronization of the selected signal by automatically disabling the **Sync. Signal Selection** icon in all frames.

NOTE: This is a switch command. After the **View -> Release Signal Selection Sync.** command is activated, the next time the **View** pull-down menu is accessed, this command are replaced with its **View -> Sync. All Signal Selection** counterpart.

Source Commands

Find Scope

Menu Bar:	Source -> Find Scope
Bind Key	Shift+S

This command opens the *Find Scope* form where all imported files can be viewed by type. The files can be viewed by sorting alphabetically or by call level.

The section below **Scope Type** lists the results matching the specified type and sort method. After selecting a result in the **Scope Type** section (only one **Scope Type** option can be selected at a time), the section below **Instance List** shows the instance (when the selection in the **Scope Type** is **File**) or hierarchical path (when the selection in the **Scope Type** is **Function**, **Module** or **Task**). If multiple scopes are selected in the **Scope Type** section, instances of the selected modules are listed scope by scope in the **Instance List** section.

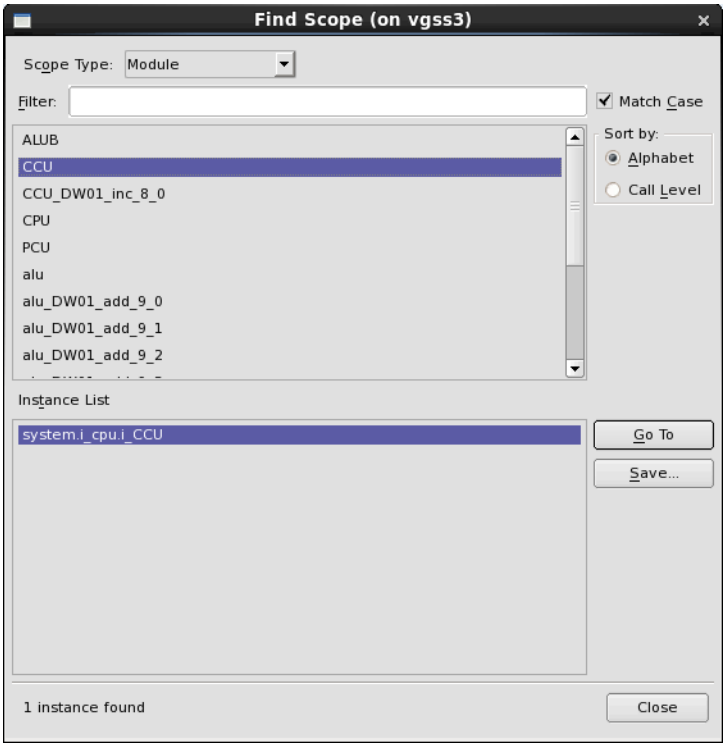


Figure: Find Scope Form

For Verilog designs, the **Scope Type** section includes the following options. The default type is **Module**.

- **File:** List by Verilog/VHDL file name including full directory path.
- **Module:** List by Verilog module name.
- **Task:** List by Verilog task name.
- **Function:** List by Verilog/VHDL function name.

For VHDL designs, the **Scope Type** section includes the following options. The default type is **Entity (Architecture)**.

- **Entity (Architecture):** List by VHDL entity (architecture) name.
- **File**: List by Verilog/VHDL file name including full directory path.
- **Function:** List by Verilog/VHDL function name.
- **Package**: List by VHDL package name.
- **Procedure**: List by VHDL procedure name.

For mixed Verilog/VHDL designs, all scope types are available.

The **Sort by** section includes the following sorting methods. Only one sorting method can be selected at a time. The default value of this option is *Alphabet*.

- **Alphabet:** List the selected type alphabetically.
- **Call Level:** List the selected type by the call level as displayed in the design browser.

When the **Match Case** option is turned *on*, the search with the string set in the **Filter** text field is case-sensitive. The default value of this option is *on*.

In the **Filter** text field, enter a string to search for source types matching the string. Wildcard characters are supported.

Click the **Save** button to open the *Save Result* form where you can view the directory structure and specify a file name to save the search results to. The saved file is in ASCII format.

Click the **Go To** button to highlight the selected item in the right column of the *nTrace* source code frame. You can also double-click the result in the **Instance List** section.

Click the **Close** button to close the *Find Scope* form without making any changes.

Find Signal/Instance/Instport

Menu Bar:	Source -> Find Signal/Instance/Instport
Bind Key	Shift+A

This command finds a signal, instance or instance port (Instport) in the source code by the signal, instance or instance port's hierarchical name.

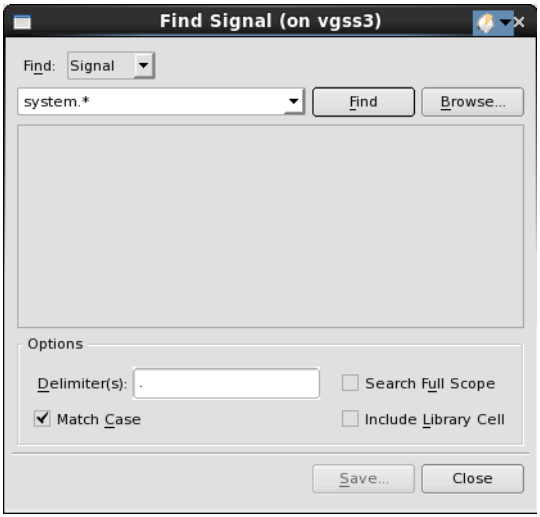


Figure: Find Signals Form

The *Find Signal* form includes the following options and fields:

- **Find:** Select **Signal**, **Instance** or **Instport** from the **Find** selection field. In the table below the **Find** selection field, enter the hierarchical name of desired the signal, instance or instance port. Then click the **Find** button.

  Select **Ctrl+A** key to select all the signals in the *Find* Form. Select **Ctrl+W** key to add all the selected signals at once in the *Find Signal* form. The signals are then added to the *nWave* window. The minimum signals which can be added is *1000* which is the default value. The default value for the signals which can be added in the *Find Signal* form can be changed using the **Get Signals -> Options form -> Others tab -> Ask if Signals Over** option.

  The **Find Signal**/Instance/**Instport** text field supports the following regular expressions:

Syntax	**Expression**	**Description**
? (question mark)	Any character	Matches any one character.
* (asterisk)	Zero or more	Finds zero or more occurrences of the preceding expression.
| (pipe)	OR	Matches the expression before or after the pipe sign “|”. Mostly used within a group. For example, “[sponge]|[mud] bath” matches “sponge bath” and “mud bath”.

+ (plus)	One or more	Matches at least one occurrence of the preceding expression. For example, “ba+” matches “ba”, “baa”, “baaa” and so on.
- (dash)	Prevent match	An example of “-Y” means to prevent a match when Y appears at this point in the expression.

- **Browse:** Click this button to browse signals of the current scope in the *Browse Signal* form. In the *Browse Signal* form, click any of the scopes displayed in the **Scope** list and the corresponding signals are listed in the **Signal** list. Select a desired signal from the **Signal** list or select **Ctrl+A** key to select all the signals and click **OK**. The *Browse Signal* form is closed and this signal's hierarchical name is then copied to the **Signal/Instance/ Instport** text field of the *Find Signals* form.

  Select **Ctrl+W** key to add all the selected signals at once. The signals are then added to the *nWave* window. The minimum signals which can be added is *1000* which is the default value. The default value for the signals which can be added in the *Find Signal* form can be changed using the **Get Signals -> Options form -> Others tab -> Ask if Signals Over** option.

  The *Browse Instport* form has similar behavior and functions the same.

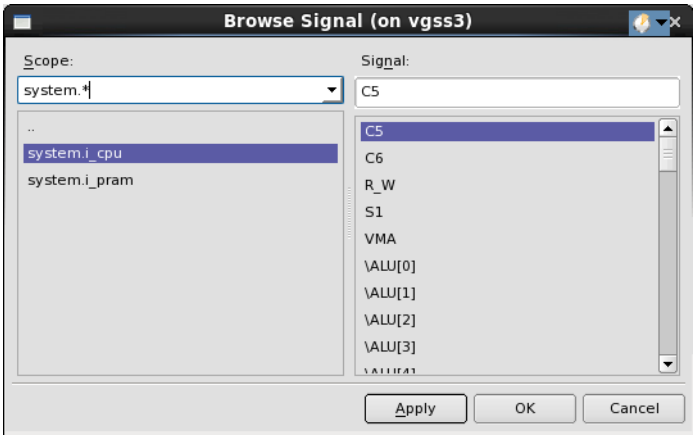


Figure: Browse Signal Form

The **Options** section has the following field and options:

- **Delimiter(s):** Enter the delimiter in the **Delimiter(s)** text field and press the **Enter** key on the keyboard, the delimiter in the **Find** text field and the search result area changes accordingly. Specifying multiple delimiters (with or without space) is also supported. After specifying multiple delimiters and pressing the **Enter** key, the delimiter in the **Find** text field and the search

result area changes based on the first specified delimiter. The default delimiter is the period (.) character.

- **Match Case:** When this option is turned *on*, *nTrace* finds only those instances in which the capitalization matches the text you typed in the **Signal/Instance/Instport** text field.
- **Search Full Scope:** When this option is turned *off*, the search is only performed in the scope defined in the **Find Signal/ Instance/Instport** text field of the *Find* form. When the **Search Full Scope** option is turned *on*, the signal search is performed through the entire hierarchy regardless of the scope (hierarchy) defined in the **Find Signal/Instance/Instport** text field.
- **Include Library Cell:** When this option is turned *on*, the search includes library cells.

NOTE: The found signal/instance/instport can be dragged to the *nTrace*/*nWave*/*nSchema* window.

Find String

Menu Bar: Source -> Find String

Toolbar Icon:  : Find String

Bind Key /

This command opens the *Find String* form where a specific string can be searched in the *nTrace* source code frame.

NOTE: After the target string is found in the source code frame and the **Set as Active Scope** command is invoked in the right-click menu options, the scope where the string is located is set as an active scope. The instance list of current scope is shown in the message form.

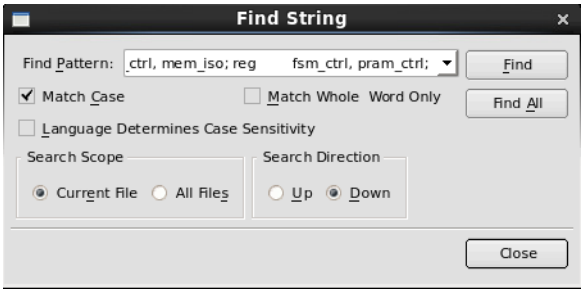


Figure: Find String Form

The *Find String* form includes the following options and fields:

- **Find Pattern:** Specify a search pattern in the **Find Pattern** text field.
- **Match Case:** When this option is turned *on*, instances matching the capitalization of the text entered in the **Find Pattern** text field is found. The default value of this option is *on*.
- **Match Whole Word Only:** When this option is turned *on*, the identifiers matching the complete word you type in the **Find Pattern** text field are searched. The default value of this option is *off*.
- **Language Determines Case Sensitivity:** When this option is turned *on*, case-sensitivity is determined by the design program language. For Verilog and mixed design, it is case-sensitive. For VHDL design, it is not case-sensitive. The default value of this option is *off*.
- **Search Scope:** Specify whether to find the specified pattern in the current file or in all files by selecting either the **Current File** option or the **All Files** option. The default value of this option is *Current File*.
  - When the **All Files** option is turned *on*, the **File Filter** field is shown where a string to filter files can be specified. Wildcard characters are supported for filtering. For multiple strings, a semicolon (;) character or a blank space, can be used as a separator. After clicking the **Find All** button, file names matching the string is listed in the **Search** tab of the *Message* frame.
  - When the **All Files** option is turned *on*, the **Skip ovm/uvm/vmm Predefined Library** option is shown. After enabling the **Skip ovm/uvm/vmm Predefined Library** option, search results excludes the

ovm/uvm/vmm predefined library files. The default value of this option is *off*.

NOTE: The user-specified files and the files directly included in the user-specified files are not treated as the predefined library file. When the **Skip ovm/uvm/vmm Predefined Library** option is turned *on*, the search results still show the user-specified files and files included in the user-specified files.

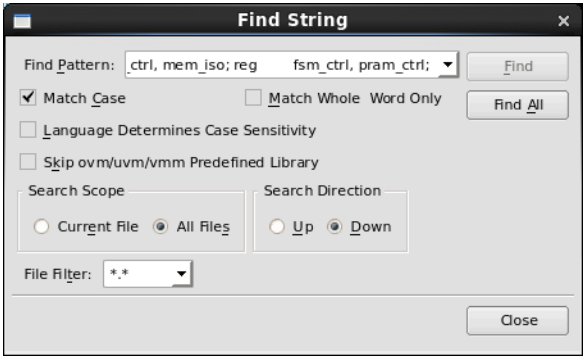


Figure: Find String Form with All Files option Enabled

- **Search Direction:** Specify the search direction from the current position; turn the **Up** option *on* to search backward or turn the **Down** option *on* to search forward. The default value of this option is *Down*.
- **Find:** Click this button and instances matching the specified string is highlighted on the source code frame. The matching string is shown one by one whenever this button is clicked.
- **Find All:** Click this button and all instances matching the specified string is shown on the **Search** tab of the *Message* frame. Double-click the associated string line on the **Search** tab of the *Message* frame to quickly and easily locate the specified string in the source code frame.

NOTE: **Ctrl+C** and **Ctrl+V** are only used for drag-and-drop. They are not used for copy-and-paste.

Go To

The **Go To** command includes three sub-commands: **Line**, **1st Executable**, and **Current Statement**.

Line

Menu Bar:	Source -> Go To -> Line
Bind Key	G

The **Go To Line** command opens the *Go to Line* form where the line number can be entered in the **Line Number** text field. Click the **OK** button to place the cursor at the specified line of the source code frame indicator area.

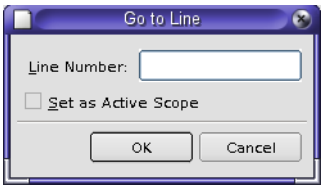


Figure: Go To Line Form

When the **Set as Active Scope** option is turned *on* and the **OK** button on the *Go to Line* form is clicked, the scope where the line jumped to is set as an active scope. When the **Set as Active Scope** option is turned *off*, clicking the **OK** button on the *Go to Line* form jumps to the specified line without changing the active scope. The default value of this option is *off*.

1st Executable

Menu Bar:	Source -> Go To -> 1st Executable
Bind Key	1 (one)

The **Go To 1st Executable** command highlights the first executable statement in the source code frame.

Current Statement

Menu Bar:	Source -> Go To -> Current Statement

The **Go To Current Statement** command goes to the statement that the simulator is currently stopped on.

Show

Calling

Menu Bar:	Source -> Show -> Calling
Toolbar Icon:	
Bind Key	Ctrl+Alt+A

This command brings up the statement that called (shown in the source code frame) the current active scope. Two alternatives are available to determine the calling statement:

1. Double-clicking on a module in the source code frame locates the calling statement (shown in the source code frame) of that module.
2. Right-click the instance in the design browser frame to invoke the same command through the **Scope** right-click menu.

Definition

Menu Bar:	Source -> Show -> Definition
Toolbar Icon:	
Bind Key	Ctrl+Alt+N

This command brings up the definition statement (shown in the source code frame) of the selected instance in the source code frame. An alternative method for determining the definition statement is to double-click an instance name in the source code frame or an instance node in the design browser frame. This locates the definition statement (shown in the source code frame) of the selected instance.

Entity

Menu Bar:	Source -> Show -> Entity

This command brings up the entity statement (shown in the source code frame) of the selected architecture in the design browser frame.

NOTE: This command is for VHDL only.

Reference

Menu Bar: Source -> Show -> Reference

Refer to the **Source -> Show -> Reference** command description in the *Testbench Debug* chapter for details.

NOTE: This command is for SystemVerilog Testbench only.

Edit Source File

Menu Bar: Source -> Edit Source File

Toolbar Icon: 

NOTE: This command is enabled when the design is loaded from source files. If the design is loaded from a precompiled library, this command is disabled.

This command invokes the text editor and displays the text for the source currently shown in the source code pane. The text editor to use is specified on the **Editor** page of the *Preferences* form (invoked with the **Tools -> Preferences** command). When the design is imported from the source files directly, invoke the **File -> Reload Design** command to recompile the design after editing a file.

NOTE: When the design is imported from a precompiled library and one or more source files are edited, the design must be recompiled outside the Verdi platform (using the *vericom/vhdlcom* utilities) before it can be reloaded.

Parameter Annotation

Menu Bar: Source -> Parameter Annotation

This toggle command turns the display of parameter values in the source code frame *on* and *off*. The default value of this option is *off*. To set the color used for parameter annotation, select the **Tools -> Preferences -> Source Code** page-> **Color** page. In the **Type** list, click **Parameter Annotation**. The font and color used for the **Parameter Annotation** type can now be changed.

Active Annotation

Menu Bar: Source -> Active Annotation

Bind Key X

After the simulation results are loaded from the **File ->** Open Waveform File command, the **Active Annotation** toggle command can be activated to display the simulation results (the signal/bus values) in the source code frame for all types of HDL signals (for example, wire, reg, task, and so on), SystemVerilog signals (for example, name, interface, and so on), and SystemVerilog Assertions (for example, cover, assume, assert, property, sequence, local variables, and so on). *nTrace* provides cursor time information in the toolbar, and the signal values can be updated by changing the cursor time.

Objects are annotated as follows:

Object Type	Symbol	Meaning
All	NF	The object is not found in the simulation dumping.
	NV	No value is dumped in current time.
	SE	Start Evaluation.
	UE	Under evaluation. The evaluation of the object is started but is not completed at current time.
Assert		The assertion succeeds.
		The assertion fails.
	incomplete	The assertion is still under evaluation, but the simulation terminated.
End Point	True/False	Annotates the value of end points with string.
Falling Edge		The signal is transitioning from 1 to 0.
Local Variables	NV*n	Multiple SVA local variables with no value dumped (for example, {NV*4}).
Rising Edge		The signal is transitioning from 0 to 1.
Signals	Val	Current value (for example, 0 or 1) for single bit signals or 55 for buses.
	Val1 -> Val2	Signal changes from value 1 to value 2 (for example, 0 -> 1) for single bit signals or aa -> 55 for buses.

Refer to the **Active Annotation** command of the *Interactive Simulation Debug* chapter for details about the active annotation support in Interactive Simulation Debug mode.

Function Annotation

Menu Bar:	Source -> Function Annotation
Bind Key	Y

This toggle command is activated when the **Active Annotation** toggle command is *on*. When **Function Annotation** is turned *on*, the return values of a given function is annotated in blue under where it is called in the *nTrace* source code frame.

NOTE: The function value is not annotated in the testbench scope or in the class function of any scope.

If the **Function Annotation** toggle command is turned *on* and you double-click a function calling statement in the source code frame, the actual path of the function is determined and the values inside the function is calculated dynamically. At the same time, *nTrace's* design browser frame is changed to a **Function Stack** frame for viewing the calling function's stack information.

Refer to the **Function Step In** and **Function Step Out** commands for details on how to trace into function stacks and step out to the previous stack.

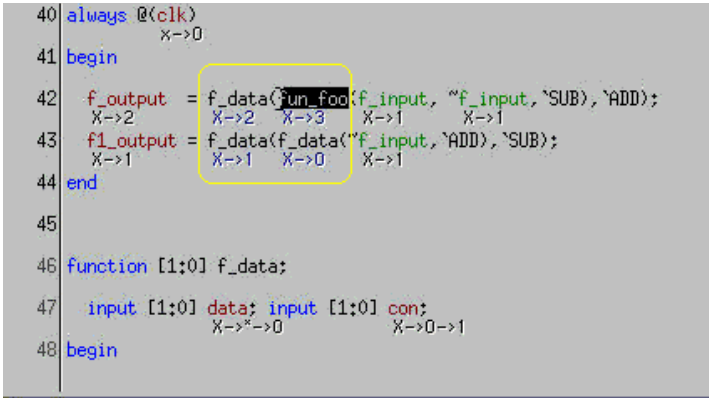


Figure: Function Calling with Annotation

To set the color used for function annotation, select the **Tools -> Preferences -> Source Code** page **-> Color** page. In the **Type** list, click **Function Annotation**. You can now change the font and color used for the **Function Annotation** type.

Leading Zeros

Menu Bar: Source -> Leading Zeros

This toggle command is activated when the **Active Annotation** toggle command is *on*. After the simulation results are loaded and displayed in the source code frame, invoke this toggle command to display leading zeros for signal values in the source code frame. The number of leading zeros depends on the format set for the signal value. To set the signal format, select any signal and invoke the **Format** command from the right-click menu, the desired format for the signal is displayed in the source code frame.

To always enable leading zeros in the future, turn *on* the **Leading Zero** option (located on the **View Options** page under the **Source Code** page on the *Preferences* form invoked by the **Tools -> Preferences** command).

Expand Macro

Menu Bar: Source -> Expand Macro

Bind Key: M

When this command is turned *on*, macro statements in the source code frame is expanded, and the background color of macro statements is highlighted in white. Source code operations, such as **Trace Driver**, **Trace Load**, and **Active Annotation**, are available for the expanded macro.

Click the plus (+) or minus (-) symbols in the line number pane to expand or collect the macro block folders. Alternatively, use the **View -> Macro Folder -> Collapse All in File/Expand All in File/Collapse All in Design/Expand All in Design** commands to collect or expand all macro folders in the current file or the entire design. When this command is *off*, moving the cursor over a macro displays a yellow tip containing the macro details. The default value of this option is *on*.

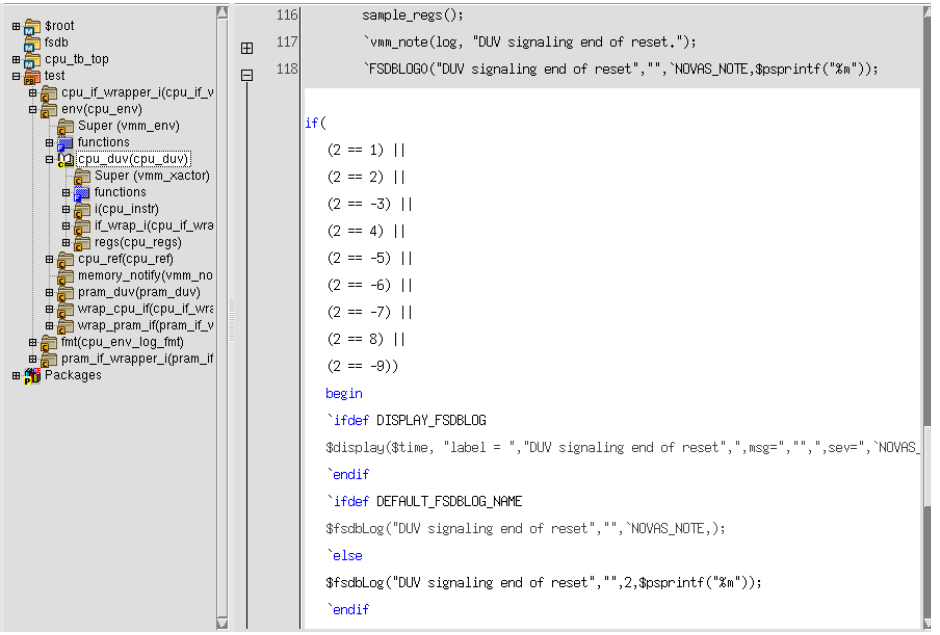


Figure: Expanded Macro in nTrace

When a tree node with class/function/task/module/coverage/constraint type in the **Declaration** frame is clicked and the definition is inside a macro, only the first line of the expanded macro is highlighted.

Expand Decrypted Code

Menu Bar: Source -> Expand Decrypted Code

Bind Key: D

In the *nTrace* main window, decrypted code is treated like a normal macro except for the copying, pasting, dumping, printing and editing functions. When a SystemVerilog design with protected code is imported and this command is turned *on*, the plus (+) symbol is shown in the line number pane for protected code. Click any plus (+) symbol to expand the decrypted code and the decrypted code with the background color highlighted in white is displayed immediately below the protected code line. Click any minus (-) symbol in the line number pane to collect the expanded decrypted code folder. When this command is

turned *off*, the plus (+) symbol is not shown in the line number pane. The default value of this option is *on*.

NOTE: The plus (+) and minus (-) symbols only appear when the **Automatic Encrypted Folding** option on the **Source Code -> Code Folding** page of the *Preferences* form (invoked with the **Tools -> Preferences** command) is turned *on*.

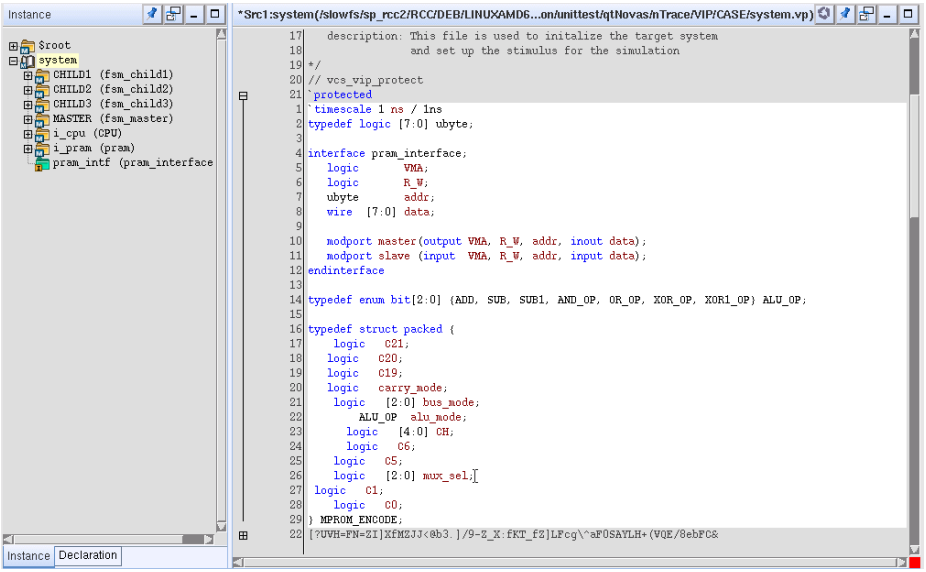


Figure: Expanded Decrypted Code in nTrace

NOTE: When entering Interactive Debug mode, the **Expand Decrypted Code** command is turned *on* and then disabled. The setting for this command cannot be changed in Interactive Debug mode.

It is supported to set breakpoints for Interactive Debug inside the decrypted protected code and execute line by line, like macro-debugging.

Expand Implicit Port

Menu Bar: Source -> Expand Implicit Port

When a SystemVerilog design with implicit port connections is imported and this command is *on*, implicit port statements in the source code frame is expanded in

a single line following the explicit port statements and the background color of the statements is highlighted in white. Source code operations, such as **Trace Driver**, **Trace Load**, and **Active Annotation**, are available for the expanded implicit ports. Clicking on the plus (+) or minus (-) symbols to expand or collect the implicit port folder. Alternatively, use the **View -> Implicit Port Folder -> Collapse All in File/Expand All in File/Collapse All in Design/Expand All in Design** commands to collect or expand all implicit port folders in the current file or whole design. The default value of this option is *off*.

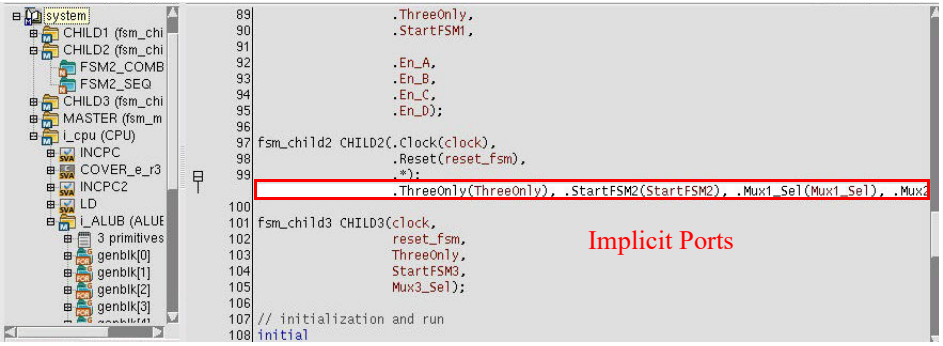


Figure: Expanded Implicit Port in nTrace

Signal Value Radix

After a desired signal is selected (multiple selections are supported) in the source code frame, invoke this command to show the value radix of the selected signal.

NOTE: After a simulation results file is loaded from the **File -> Open Waveform File** command and the **Source -> Active Annotation** command is invoked, the sub-commands of the **Signal Value Radix** command can be activated.

Binary

Menu Bar: Source -> Signal Value Radix -> Binary

This command displays the signal value in **Binary** format.

Octal

Menu Bar: Source -> Signal Value Radix -> Octal

This command displays the signal value in **Octal** format.

Hexadecimal

Menu Bar: Source -> Signal Value Radix -> Hexadecimal

This command displays the signal value in **Hexadecimal** format.

Decimal

Menu Bar: Source -> Signal Value Radix -> Decimal

This command displays the signal value in **Decimal** format.

ASCII

Menu Bar: Source -> Signal Value Radix -> ASCII

This command displays the signal value in **ASCII** format.

Enumerated Literal

Menu Bar: Source -> Signal Value Radix -> Enumerated Literal

This command displays the signal value in **Enumerated Literal** format.

Add Alias from File

Menu Bar: Source -> Signal Value Radix -> Add Alias from File

Refer to the **Waveform -> Signal Value Radix -> Add Alias From File** command of the *nWave* chapter for details.

Add Alias from Program

Menu Bar: Source -> Signal Value Radix -> Add Alias from Program

Refer to the **Waveform -> Signal Value Radix -> Add Alias From Program** command of the *nWave* chapter for details.

Remove Alias

Menu Bar: Source -> Signal Value Radix -> Remove Alias

Refer to the **Waveform -> Signal Value Radix -> Remove Alias** command of the *nWave* chapter for details.

Signal Value Notation

After a desired signal is selected (multiple selections are allowed) in the source code frame, invoke this command for the signal value notation of the selected signal.

NOTE: After a simulation results file is loaded from the **File -> Open Waveform File** command and the **Source -> Active Annotation** command is invoked, the sub-commands of the **Signal Value Notation** command can be activated.

Unsigned

Menu Bar: Source -> Signal Value Notation -> Unsigned

Refer to the **Waveform -> Signal Value Notation -> Unsigned** command description in the *nWave* chapter for details.

Signed 2's Complement

Menu Bar: Source -> Signal Value Notation -> Signed 2's Complement

Refer to the **Waveform -> Signal Value Notation -> Signed 2's Complement** command description in the *nWave* chapter for details.

Signed 1's Complement

Menu Bar: Source -> Signal Value Notation -> Signed 1's Complement

Refer to the **Waveform -> Signal Value Notation -> Signed 1's Complement** command description in the *nWave* chapter for details.

Signed Magnitude

Menu Bar: Source -> Signal Value Notation -> Signed Magnitude

Refer to the **Waveform -> Signal Value Notation -> Signed Magnitude** command description in the *nWave* chapter for details.

Signal Current Value Type

NOTE: This command is available only when the **Current** option is selected in the **Show Spice Value** section of the **Preferences -> Spice Debug -> Miscellaneous Page**.

This command includes the two subcommands: **Cursor Value** and **Average Value**. Only one value can be active at a time. The default value of this option is *Cursor Value*.

Signal Current Value Type and **Signal Voltage Value Type** are similar. Refer to the **Signal Voltage Value Type** command descriptions for details.

Cursor Value

Menu Bar: Source -> Signal Current Value Type -> Cursor Value

This command shows the cursor value of the selected signal.

Average Value

Menu Bar: Source -> Signal Current Value Type -> Average Value

This command shows the average value of the annotation time range.

Signal Voltage Value Type

This command includes the two subcommands: **Cursor Value** and **Average Value**. Only one value can be active at a time. The default value of this option is *Cursor Value*.

Cursor Value

Menu Bar: Source -> Signal Voltage Value Type -> Cursor Value

NOTE: This subcommand is enabled only when the selected signal is an analog signal and the annotation time range is specified in the **Set Annotation Time Range** command.

After a signal is selected, this command shows the cursor value of the selected signal. This command is also supported in the right-click menu option of the source code pane.

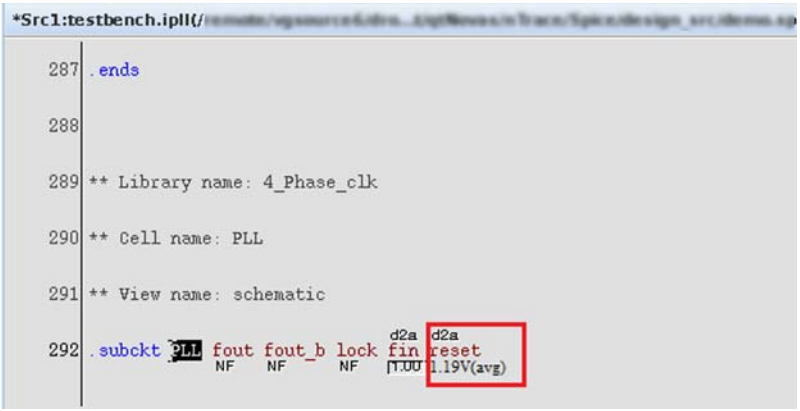


Figure: Display Format of Annotation

Average Value

Menu Bar: Source -> Signal Voltage Value Type -> Average Value

NOTE: This subcommand is enabled only when the selected signal is an analog signal and the annotation time range is specified in the **Set Annotation Time Range** command.

After a signal is selected, this command shows the average value of the selected signal. This command is also supported in the right-click menu option of the source code pane.

Set Annotation Time Range

Menu Bar: Source -> Set Annotation Time Range

NOTE: This command is available after an AMS design is loaded and the **Source -> Active Annotation** toggle command is turned *on*.

This command opens the *Set Annotation Time Range* form where the **From Time** and **End Time** fields can be set for the annotation time range.

When the **Sync with Cursor/Marker** option is enabled, the **From Time** and **End Time** fields is disabled and Verdi uses the cursor or marker of the primary waveform as the from time and end time. If the waveform window is closed after the option is enabled, Verdi uses the latest cursor or marker value of waveform as the time range.

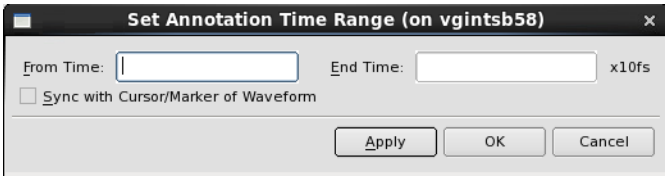


Figure: Set Annotation Time Range Form

Set/Unset Bookmark

Menu Bar: Source -> Set/Unset Bookmark

Toolbar Icon: 

Bind Key: Ctrl+F2

This command sets or clears a bookmark at the current line of the source code. Bookmarks return the source code to the bookmarked area by clicking the **Go To** option in the *Manage Bookmarks* form (through the **Source -> Manage Bookmarks** command) or using the bookmark cache. One usage of bookmarks is to keep record of important decision branches during the back-trace process.

Previous Bookmark

Menu Bar: Source -> Previous Bookmark

Bind Key: Shift+F2

This command goes to the previously set bookmark in the same scope.

Next Bookmark

Menu Bar: Source -> Next Bookmark

Bind Key: F2

This command goes to the next bookmark that you set in the current scope.

Manage Bookmarks

Edit Bookmarks

Menu Bar:	Source -> Manage Bookmarks -> Edit Bookmarks
Related Bind Keys:	Ctrl+F2: Toggle Bookmark
	F2: Go To Next Bookmark
	Shift+F2: Go To Previous Bookmark

This command opens the *Manage Bookmarks* form where the set bookmark(s) can be deleted, renamed or go to.

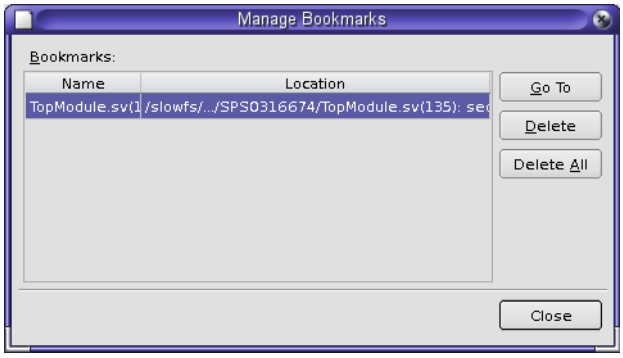


Figure: Manage Bookmarks Form

To rename bookmarks, select the editable cell (under the **Name** column) and press **Insert** on your keyboard. The background of the cell turns gray, which indicates that the bookmark can be renamed.

The bookmarks also appear in the **Source -> Manage Bookmarks** submenu as a bookmark cache that can be used to go to the bookmark in the source code without accessing the *Manage Bookmarks* form.

Manage User-defined Folders

Menu Bar:	Source -> Manage User-defined Folders

This command opens the *Manage User-defined Folders* form where user-defined folders can be added, deleted and edited. The user-defined folder is shown as a collapsed folder by default.

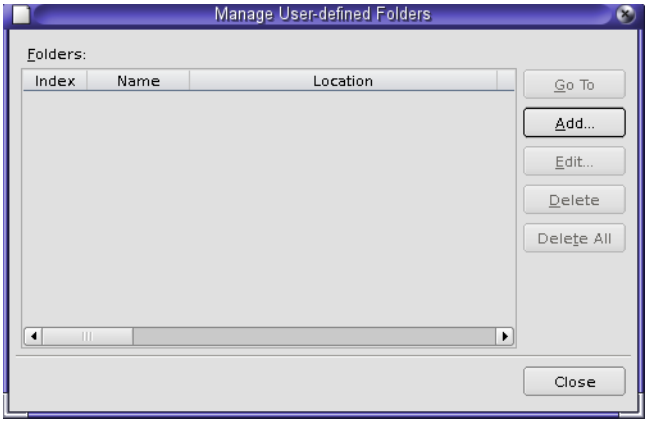


Figure: Manage User-defined Folders Form

The *Manage User-defined Folders* form includes the following options:

- **Go To:** Click this button to go to the corresponding source code of the selected folder and the caret cursor is placed at the first line of the folder. The current scope is also changed and highlighted in the source code.
- **Add:** Click this button to open the *Add Folder* form where new folders can be added.

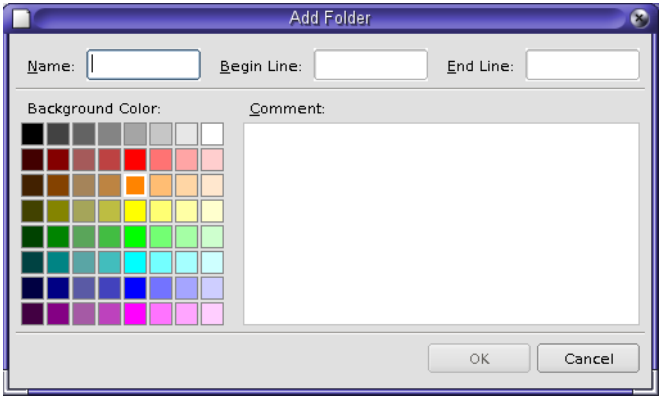


Figure: Add Folder Form

The *Add Folder* form includes the following fields:

- **Name:** Enter the folder name.
- **Begin Line:** Specify the begin line of the folder.
- **End Line:** Specify the end line of the folder.

  - **Background Color:** Specify the background color for the folder.
  - **Comment:** Add a comment for the folder. The comment is shown as a tip when the mouse moves across line range of the folder.
- **Edit:** Click this button to change the settings of the existing folder.
- **Delete:** Click this button to delete the selected folder.
- **Delete All:** Click this button to delete all folders.
- **Close:** Click this button to close the *Manage User-defined Folders* form.

Recent Files

Menu Bar: Source -> Recent Files

This cache stores the last ten viewed source code scopes/files in the source code frame.

OneTrace Commands

These commands trace the signal in the source code frame, the schematic frame/window, and the waveform window. The signal selection among windows can be synchronized by enabling the **View -> Sync. All Signal Selection** command. If any signals in the waveform, source, or schematic windows are selected, the associated signal in the other frames is also selected.

A closed-folder icon with red dot in the design browser frame indicates that the traced signal passes through the corresponding instance.

Value Change

Menu Bar:	OneTrace -> Value Change
Toolbar Icon:	
Bind Key	Alt+Shift+V

This command backtraces the selected signal's value at the current cursor time by finding an active statement driver for a signal, tracing its driving signal, and then repeating this on the active driving signal until more than one active driving signal is encountered. The results of backtracing are shown in the source view and the trace view and it displays the last driver encountered.

Driver

Menu Bar:	OneTrace -> Driver
Toolbar Icon:	
Bind Key	Alt+Shift+D
Mouse Action:	Double-click a signal in the source code frame to trace the drivers of that signal.

This command statically traces all possible drivers that exist in the design for the selected signal (multiple drivers in different instances and in many designs are allowed). The trace results are shown in two places:

1. In the source code frame indicator area with semicircles indicating driver results. The current selected result is highlighted with a color-filled semicircle  and additional results (if multiple results exist) are identified with a semicircle symbol  . The driver results that may be impacted by

power states are highlighted with the power impacted symbol (this only appears after a CPF/UPF file is loaded).

2. On the **OneTrace** tab in the *Message* frame which reports the name, instance, and location of each match. By double-clicking on the line of a match, you can easily locate that match in the source code frame.

Alternatively, a signal's driver can be traced by dragging the signal from an open schematic window to the **Trace Driver** icon on the *nTrace* toolbar.

NOTE: The **Driver** command name is changed to **Trace HDL Driver** when CPF/UPF files are loaded into the Verdi platform.

Load

Menu Bar:	OneTrace -> Load
Toolbar Icon:	
Bind Key	Alt+Shift+L

This command statically traces all possible loads that exist in the design for the selected signal (multiple loads in different instances and in many designs are allowed). The trace results are shown in two places:

1. In the source code frame indicator area with semicircles indicating load results. The current selected result is highlighted with a color-filled semicircle and additional results (if multiple results exist) are identified with a semicircle symbol . The load results that may be impacted by power states are highlighted with the power impacted symbol (this only appears after a CPF/UPF file is loaded).
2. On the **OneTrace** tab in the *Message* frame which reports the name, instance, and location of each match. By double-clicking on the text line of a match, you can easily locate that match in the source code frame.

Alternatively, a signal's load can be traced by dragging the signal from an open schematic window to the **Trace Load** icon on the *nTrace* toolbar.

NOTE: The **Load** command name is changed to **Trace HDL Load** when CPF/UPF files are loaded into the Verdi platform.

Connectivity

Menu Bar: OneTrace -> Connectivity

Toolbar Icon: 

Bind Key Alt+Shift+C

This command simultaneously gives the results of both the **Driver** and **Load** commands with a single command. The results are displayed in the same manner as described above.

NOTE: This command only traces within the HDL code.

Chain Driver

Menu Bar: OneTrace -> Chain Driver

This command traces through buffer/inverter chains and stops at the first non-single input driver. The driver chain results is printed to the *Message* frame as a numbered list of tracing history with the driver highlighted. The format is shown as *<D> <*number*> (*scope name*) *file name*(*line number*): *source_code contents* (*<D> designates driver; <D> designates pass-through, and <*number*> designates descending driver sequence number). Following is an example.

```
<1> top.e /* result of trace chain driver */
  *Total : 2 driver(s)
   [+]*<D> <5> (top) top.sv(6): and a1(a, a_l, a_r);
       *<D> <4> (top) top.sv(7): buf b1(b, a);
       *<D> <3> (top) top.sv(8): buf b2(c, b);
       *<D> <2> (top) top.sv(9): buf b3(d, c);
       *<D> <1> (top) top.sv(10): buf b4(e, e2, d);
   [+]*<D> <5> (top) top.sv(12): A aa(e, clk);
        <D> <4> (top.aa) top.sv(16): B bb(oa, ia);
       *<D> <3> (top.aa.bb) top.sv(20): buf bb1(ob, ib);
        <D> <2> (top.aa) top.sv(16): B bb(oa, ia);
        <D> <1> (top) top.sv(12): A aa(e, clk);
```

NOTE: The **Chain Driver** command is synchronized with the **OneTrace -> Trace across Hierarchy** toggle command. When the **Trace across Hierarchy** command is turned *off*, the **Chain Driver** command only shows the tracing results in the current scope. When the **Trace across Hierarchy** command is turned *on*, the **Chain Driver** command traces across all scopes.

Fan-in

Menu Bar: OneTrace -> Fan-in

Bind Key Alt+Shift+I

This command displays the fan-in register trace results for the selected signal in the source code frame and the **OneTrace** tab of the *Message* frame.

Fan-out

Menu Bar: OneTrace -> Fan-out

Bind Key Alt+Shift+F

This command displays the fan-out register trace results for the selected signal in the source code frame and the **OneTrace** tab of the *Message* frame.

Mapped Signal

Menu Bar: OneTrace -> Mapped Signal

When this command is turned *on*, the number of mapped drivers or sequential outputs in a driver result is showed in the *nTrace* message pane.

Trace across Hierarchy

Menu Bar: OneTrace -> Trace across Hierarchy

When this command is turned *on*, a signal is traced through the design hierarchy level by level and the **Trace Driver** and **Trace Load** commands behave exactly the same in *nTrace*. When this command is turned *off*, the **Trace Driver** and **Trace Load** commands traces a signal level by level. The default value of this option is *on*.

Stop on Complex Exp Instport

Menu Bar: OneTrace -> Stop on Complex Exp Instport

When this command is turned *on*, complex expressions (that is, all expressions except slice, index, and concatenate) is regarded as a driver/load and tracing stops on the instance port. For slice, index, and concatenate expressions, tracing passes through the instance port. The default value of this option is *off*.

Trace Including Top Level Port

Menu Bar: OneTrace -> Trace Including Top Level Port

When this command is turned *on*, the input port of the top level is regarded as a driver and the output port as a load. The “no drivers” messages is not reported when executing the **Trace Driver** command for a net that is a primary input (a declared input port of the top module), and the “no loads” messages is not reported when executing the **Trace Load** command for a net that is a primary output (a declared output port of the top module). The default value of this option is *off*.

Trace across Power Design

Menu Bar: OneTrace -> Trace across Power Design

NOTE: This command is available when a CPF/UPF file is loaded.

Refer to the **Trace across Power Design** command description in the *Power Aware Debug* chapter for details.

Bus Driver

Menu Bar: OneTrace -> Bus Driver

This command extracts the drivers from the index range of the selected bus and shows the results in the current *nTrace* window. After selecting a bus in the source code frame and choosing this command, the *Specify Range* form is displayed where the index range of the selected bus can be specified. If the selected signal is not a bus, a warning form is displayed.

Bus Load

Menu Bar: OneTrace -> Bus Load

This command extracts the loads from the index range of the selected bus and shows the results in the current *nTrace* window. After selecting a bus in the source code frame and choosing this command, the *Specify Range* form is displayed where the index range of the selected bus can be specified. If the selected signal is not a bus, a warning form is displayed.

Bus Connectivity

Menu Bar: OneTrace -> Bus Connectivity

This command extracts the connectivity from the index range of the selected bus and shows the results in the current *nTrace* window. After selecting a bus in the source code frame and choosing this command, the *Specify Range* form is displayed where the index range of the selected bus can be specified. If the selected signal is not a bus, a warning form is displayed.

Pop View Up from Port

Menu Bar: OneTrace -> Pop View Up from Port

This command goes to the upper hierarchy of the port after a port is selected from the source code frame.

Push View In from Port

Menu Bar: OneTrace -> Push View In from Port

This command goes to the entity or definition module of the port after a port is selected from the source code frame.

Show Previous

Menu Bar: OneTrace -> Show Previous

Toolbar Icon: 

Bind Key: P

This command jumps to the previous trace result within the instance by highlighting the corresponding semicircle symbol in the indicator area with a color-filled semicircle. The command is disabled if no further trace results are found within the current instance.

Show Next

Menu Bar: OneTrace -> Show Next

Toolbar Icon: 

Bind Key: N

This command jumps to the next trace result within the current instance by highlighting the corresponding semicircle symbol in the indicator area with a color- filled semicircle. This command is disabled if no further trace results are found within the current instance.

Show Previous in Hierarchy

Menu Bar: OneTrace -> Show Previous in Hierarchy

Toolbar Icon: 

This button provides a quick way to jump among instances of the traced results. If the results of performing a **Trace Driver** or **Trace Load** command on a signal are scattered across multiple instances, and each of the instances pertains to a non-primitive module, use the **Show Previous in Hierarchy** command to step directly to the previous instance across the design hierarchy that contains the traced results.

Show Next in Hierarchy

Menu Bar: OneTrace -> Show Next in Hierarchy

Toolbar Icon: 

This button provides a quick way to jump among instances of the traced results. If the results of performing a **Trace Driver** or **Trace Load** command on a signal are scattered across multiple instances, and each of the instances pertains to a non-primitive module, use the **Show Next in Hierarchy** command to step directly to the next instance across the design hierarchy that contains the traced results.

Reset Traced Signal's Color

Menu Bar: OneTrace -> Reset Traced Signal's Color

This command resets all the traced signals' colors in the source code frame to their default settings.

Forward History

Menu Bar: OneTrace -> Forward History

Toolbar Icon: 

Bind Key: Ctrl+Alt+F

This command traces forward through the 32 most recent trace results.

Backward History

Menu Bar: OneTrace -> Backward History

Toolbar Icon: 

Bind Key: Ctrl+Alt+B

This command traces back through the 32 most recent trace results. This command (or toolbar icon) can also be used to trace back when using the **Show Calling** or **Show Definition** command in *nTrace*.

Reset History

Menu Bar: OneTrace -> Reset History

This command clears the traced buffer and all of the displayed indicators in the source code frame.

Interface Mapping

Menu Bar: OneTrace -> Interface Mapping

Refer to the **Interface Mapping** command description in the *Testbench Debug* chapter for details.

From HDL to HVL Interface

Menu Bar: OneTrace -> From HDL to HVL Interface

Refer to the **From HDL to HVL Interface** command description in the *Testbench Debug* chapter for details.

From HVL to HDL Interface

Menu Bar: OneTrace -> From HVL to HDL Interface

Refer to the **From HVL to HDL Interface** command description in the *Testbench Debug* chapter for details.

Simulation Commands

Refer to the *Simulation Options* section of the *Interactive Simulation Debug* chapter for details.

Debug Commands

Refer to the *Interactive Debug Panes* section of the *Interactive Simulation Debug* chapter for details.

Tools Commands

Visibility

Menu Bar: Tools -> Visibility

Refer to the *Visibility* chapter for details.

VC Apps Toolbox

VC Apps Toolbox

Menu Bar: Tools -> VC Apps Toolbox

Toolbar Icon: 

This command opens the *VC Apps Toolbox* form where the VC Apps of the following categories can be invoked:

- Design Exploration
- Design Manipulation
- Design Validation
- FSDB Investigation
- Utility

New SmartLog

Menu Bar: Tools -> New SmartLog

Toolbar Icon: 

This command opens the **Smart Log** pane where the details of the Smart Log can be viewed.

New Waveform

Menu Bar: Tools -> New Waveform

Toolbar Icon: 

This command opens a new *nWave* window.

New Schematic from Source

New Schematic

Menu Bar: Tools -> New Schematic from Source -> New Schematic

Toolbar Icon: 

This command generates a hierarchical view in the *nSchema* window based on the selected scope (active scope) in the design browser frame.

NOTE: If the selected scope does not include any netlists, the *nSchema* window is not created.

Browser Window

Menu Bar: Tools -> New Schematic from Source -> Browser Window

This command creates a partial hierarchical view in *nSchema* that reflects the selected signals or instances in the *nTrace* or *nSchema* windows. The port is displayed by default and can be switched to the upper hierarchy.

NOTE: With this type of window, the scope can only be changed by the **View -> Pop View Up**, **View -> Push View In**, or **View -> Pop View Up from Port** commands, or by double-clicking on the port or the instance port.

Flattened Window

Menu Bar: Tools -> New Schematic from Source -> Flattened Window

This command creates a flattened schematic view in *nTrace* or *nSchema* after a primitive instance or signal is selected (multiple sections are supported) in *nTrace* or *nSchema*. In the new flattened schematic view, the primitive instance

and the connectivity of the signal is shown for the primitive instance and signal respectively.

Hierarchical Flattened View

Menu Bar: Tools -> New Schematic from Source -> Hierarchical Flattened View

This command invoke the *Hierarchical Flattened View* window that shows the hierarchical boundaries in the flattened schematic window.

ECO Window for Selected

Menu Bar: Tools -> New Schematic from Source -> ECO Window for Selected

This command opens an ECO window for the selected instances.

Editable Window for Selected

Menu Bar: Tools -> New Schematic from Source -> Editable Window for Selected

This command opens an *Editable Schematic* window for selected instances in the current window.

Fan-in

Menu Bar: Tools -> New Schematic from Source -> Fan-in

This command creates a new partial flattened view in the *nSchema* window based on the results from tracing the fan-in of the selected signal.

Fan-out

Menu Bar: Tools -> New Schematic from Source -> Fan-out

This command creates a new partial flattened view in the *nSchema* window based on the results from tracing the fan-out of the selected signal.

Driver

Menu Bar: Tools -> New Schematic from Source -> Driver

This command creates a new partial flattened view in the *nSchema* window based on the results from tracing the drivers of the selected signal or instance.

Load

Menu Bar: Tools -> New Schematic from Source -> Load

This command creates a new partial flattened view in the *nSchema* window based on the results from tracing the loads of the selected signal or instance.

Connectivity

Menu Bar: Tools -> New Schematic from Source -> Connectivity

This command creates a new partial flattened schematic window based on the results from tracing the connectivity of the selected signal or instance.

Clock Tree

Menu Bar: Tools -> New Schematic from Source -> Clock Tree

This command creates a new partial flattened view in the *nSchema* window based on the results from creating the clock tree for the selected signal or instance. After tracing a clock tree and creating a flattened schematic window, the flip-flops, latches, and macros for a clock domain is separated and displayed in different compressed view (black box) symbols.

If the compressed view is not a flip-flop, its type is identified (for example, latch or macro); no type identification refers to a flip-flop. In the compressed view, '+' indicates posedge-triggered; '-' indicates negedge-triggered; 'x' indicates unknown, and the number indicates the number of instances included. See the following figure as an example:

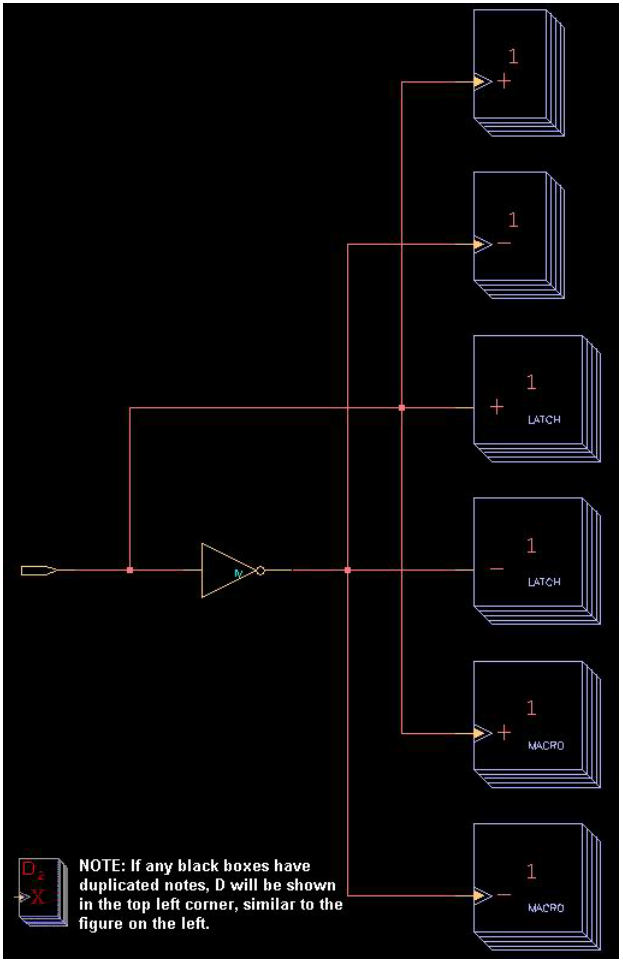


Figure: Example Clock Tree Results

When the **Clock Tree** command is invoked, the *Clock Extraction Settings* form is opened to the clock tree parameters. Refer to the *Clock Extraction Settings Form* section of the *Clock Analyzer* chapter for details.

Reset Tree

Menu Bar: Tools -> New Schematic from Source -> Reset Tree

A primitive instance or net can be selected to create a new flattened schematic (*nSchema*) window containing the extracted reset tree results.

Temporal Flow View

Auto Trace 

Menu Bar:	Tools -> Temporal Flow View -> Auto Trace
Mouse Action	Right-click

After a signal is selected, click this command to trace the source in the *Flow View* for the selected signal. The trace mode is based on the cycle-based or transition-based setting in the **Default Trace Method** section on the **Temporal Flow View ->Trace** folder **-> Trace** page of the *Preferences* form.

On the cycle-based/transition-based setting, if a signal containing an unknown "X" value is selected, click the **Trace X** command without opening the *Temporal Flow View* window. If a signal containing other values except X is selected, click **Trace This Value** command with a *Temporal Flow View* window opened.

Refer to the **Trace X** command descriptions under the *nTrace Source Code Frame* section for details.

Trace Glitch

Menu Bar:	Tools -> Temporal Flow View -> Trace Glitch
Mouse Action	Right-click

NOTE To dump the glitch information during simv runtime, add the **+fsdb+glitch=0 +fsdb+sequential** option.

This command enables debugging a glitch in *Temporal Flow View* by identifying annotated glitches and locating the root source of the glitch. This command does not create a report but only helps trace signals with glitches.

When you invoke the **Trace Glitch** command from the *nTrace* menu option, or from the right-click menu option on the *Temporal Flow View*, the **Display Glitch** option is automatically turned *on*.

If a reference signal in the *Temporal Flow View* contains a glitch, it is marked with a * symbol and the complete transition value is displayed, even if it is on the input side of the function block. The GUI command is available only when the selected signal contains glitches in its value.

New Temporal Flow View

Menu Bar: Tools -> Temporal Flow View -> New Temporal Flow View

This command creates a *Flow View* window using the settings specified in the *Trace Method Setting* form, as illustrated in the following figure:

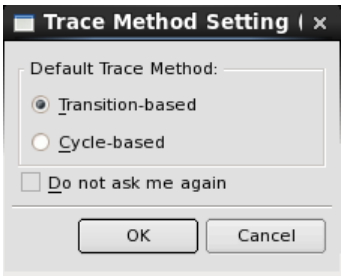


Figure: Trace Method Setting Form

NOTE: This form appears only if the **Show Default Trace Method Dialog** option invoked using **Temporal Flow View ->Trace** folder **-> Trace** page of the *Preferences* form is turned *on*.

You can also set the Flow View settings using the setting in the **Default Trace Method** section on the **Temporal Flow View ->Trace** folder **-> Trace** page of the *Preferences* form.

Add Reference Signals

Refer to the **Trace -> Add Reference Signal** command description in the *Flow View* chapter for details.

Trace X

Menu Bar: Tools -> Temporal Flow View -> Trace X

Toolbar Icon: 

The **Trace X** option opens the *Trace X Settings* form.

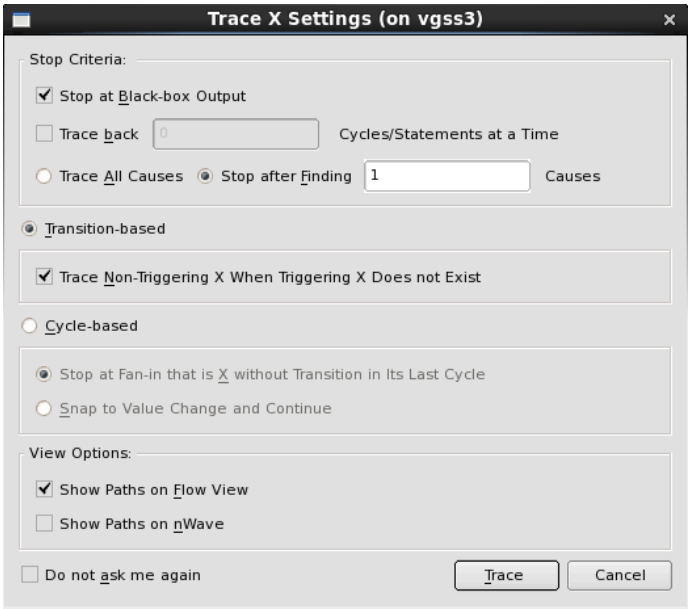


Figure: Trace X Settings

The following options are available in the **Stop Criteria** section.

- **Stop at Black Box Output:** During the trace back process, when this option is turned *on*, stop at signals that are outputs of black boxes (normally testbench, monitor, system tasks, or behavior models). When this option is turned *off*, continue tracing from black box inputs with unknown values. The default is *on*.
- **Trace Back 'n' Cycles/Statements at a Time:** The value in this option determines the number of cycles or statements to display for each trace. The default is *0*. If the number of cycles to display is smaller than the number of cycles to the root cause of the unknown, the *Trace Triggering X Results* window do not open, but the traced cycles are displayed on the flow view.
- **Trace All Causes** or **Stop after Finding 'n' Cause(s)**: The functionality works in a depth first search manner. The algorithm can be set to **Stop after Finding 'n' Cause(s)** or **Trace All Causes**. Only one option can be selected at a time. The default is to stop tracing after the first cause is found.
- **Transition-Based:** By default, the option is *on*.
  - **Trace Non-triggering X When Triggering X Does not Exist**: When this option is *on*, the unknown value on the non-triggering fan-in signals is traced when the triggering inputs have an unknown value. When this

option is *off*, only the triggering input with a transition from a known value to an unknown value is traced. By default, the option is *on*.

- **Cycle-Based:**
  - **Stop at Fan-in that is X without Transition in its Last Cycle:** When this option is *on*, the tracing is stopped for the unknown transition when no fan-in signals exist with a transition from a known value to unknown value during the last cycle. When this option is *off*, the tracing is continued for the fan-in signals that are unknown and do not have any transition (or value change) in the last cycle of fan-in signal's clock domain. By default, this option is *on*.
  - **Snap to Value Change and Continue:** When this option is *on* and an unknown transition exists for multiple cycles, then the time snaps to the point where the unknown transition begins and then continues the trace process. When this option is *off*, the time does not snap to where the transition to unknown transition begins. By default, this option is *off*.

The following options are available in the **View Options** section.

- **Show Paths on Flow View:** The results are displayed on the **Trace Triggering X Results** window. When the **Show Paths on Flow View** option is *on*, the results are also displayed on the flow view. Showing the results on the flow view may slow down the search process. By default the option is *on*. You must enable this option only when you want to display the trace X results on the flow view.
- **Show Paths on nWave:** The results are displayed on the **Trace Triggering X Results** window. When the **Show Paths on nWave** option is enabled, the results are also displayed in the *nWave* window as waveforms. The default is *off*. You must enable this option only when you want to display the trace X results on the *nWave* window.

- Click **Trace** to open the **Trace Triggering X Results** window. For more information see the *Trace Triggering X Results Window* section in the *Verdi and Siloti Command Reference Manual*.
- If the **Do not ask me again** option is selected in the *Trace X Setting* form before hiding it, then it does not appear the next time the Verdi platform is invoked.

Behavior Analysis

Menu Bar: Tools -> Behavior Analysis

This command invokes the *Behavior Analysis* form which enables the Behavior Analysis engine. The design must be loaded into the Verdi platform before the **Behavior Analysis** command is enabled. The simulation results (FSDB files) are optional although further analysis requires them. Refer to the *Behavior Analysis Folder* section in the *Preferences* chapter for additional options for setting up the Behavior Analysis engine.

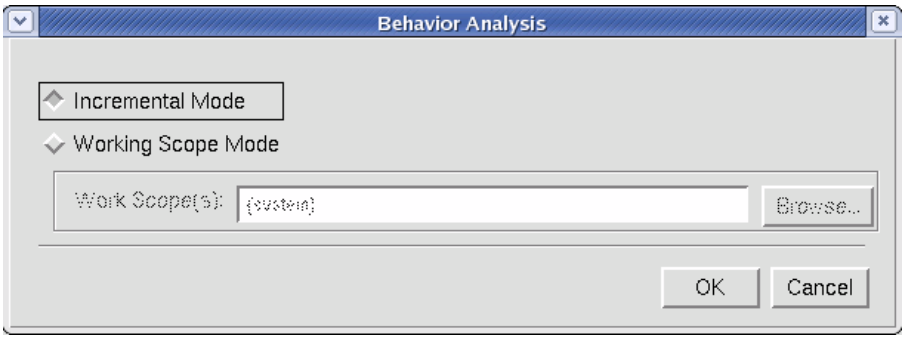


Figure: Behavior Analysis Form

The following options and field are available:

- **Incremental Mode:** When this option selected, the Behavior Analysis engine infers the design incrementally. The default is selected.
- **Working Scope Mode:** When this option is selected and one or more scopes are specified in the **Working Scope** text field, the Behavior Analysis engine analyzes the design starting with the specified scope(s), including everything below the scope(s) and then incrementally inference scopes outside the previously specified scope(s) as needed.

  To change the scope, either type in the text field or click the **Browse** button to open the *Add Scope* form where the design hierarchy can be browsed and one or more scopes can be selected/deselected using the , , or buttons. When the **OK** button is clicked, the selected scope(s) is added to the text field with {} around each scope.

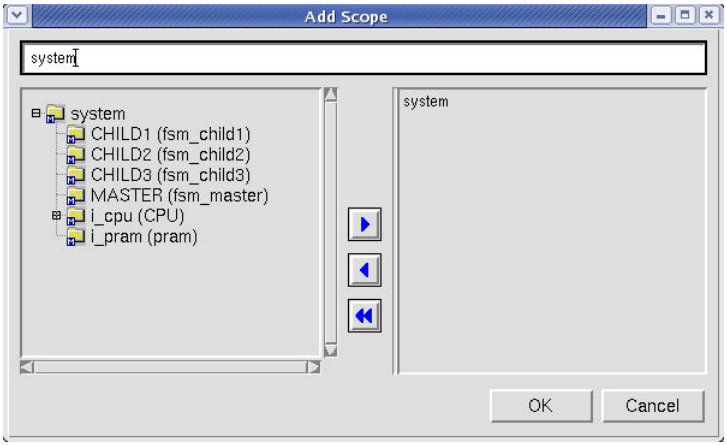


Figure: Add Scope Form

Extract Interactive FSM

Menu Bar: Tools -> Extract Interactive FSM

After the **FSM Recognition** option is turned *on* in the **Schematics -> RTL** page of the *Preferences* form (invoked with the **Tools -> Preferences** command), invoke this command to open the *Extract Interactive FSM* form where any interactive finite state machines can be extracted from the design.

NOTE: A design scope containing an FSM must be selected before invoking this command.

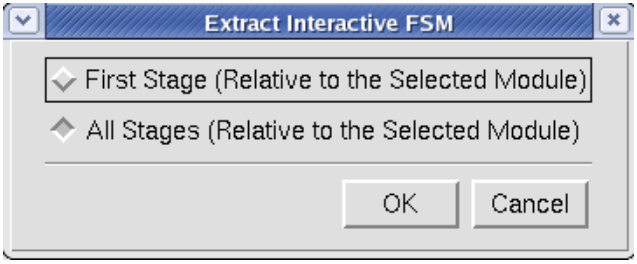


Figure: Extract Interactive FSM Form

First Stage (Relative to the Selected Module): This option acts as a filter that instructs the Verdi platform to only extract FSMs located within the current module, or in the parent module, or in direct child modules. This is in contrast to invoking the command with **First Stage** turned *off*, which extracts all FSMs in the design.

All Stages (Relative to the Selected Module): This options extracts all FSMs in the current scope and all FSMs that have direct or indirect relationships with the current scope's FSMs.

Browse/Watch/List

Interface Browser

Menu Bar: Tools -> Browse/Watch/List -> Interface Browser

NOTE: A SystemVerilog design must be loaded for this command to be enabled.

This command opens the *Interface Browser* form.

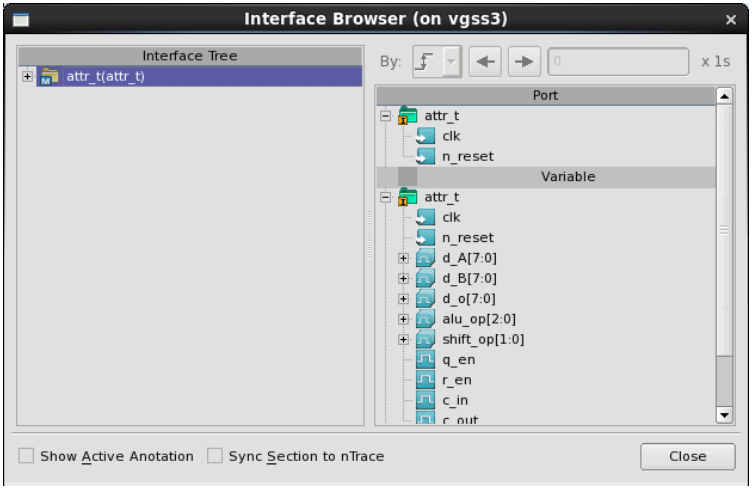


Figure: Interface Browser Form

The *Interface Browser* form includes the following panes, options, buttons and field.

- **Interface Tree**: This pane (on the left) lists the interface definitions, interface instances, modports, and HDL instances in a tree node format. Each interface definition is shown as a top node with the associated instances listed as underlying nodes. The modports and HDL instances are listed under the interface instance. The interface instance, modport and HDL instance nodes may be dragged to the source code frame to show the

associated code, the *nSchema* window to show the schematic, or the *nWave* window to show the waveform.

- **Interface Summary**: This pane (on the right) lists the ports, variables, and subroutines of the interface. The contents in this pane are automatically updated when a node in the **Interface Tree** pane is double-clicked. Signal node types can be dragged to other windows.
- **Show Active Annotation**: This option is enabled when an FSDB file is loaded. When this option is turned *on*, the simulation values is displayed in the right-most column for the variables or ports in the **Interface Summary** pane. The default value of this option is *off*.
- **Sync Selection to nTrace**: When this option is turned *on*, double-clicking an interface instance, or HDL instance in the **Interface Tree** pane displays the associated code in the source code frame. When this option is turned *off*, double-clicking an interface instance, or HDL instance in the **Interface Tree** pane only updates the **Interface Summary** pane. The default value of this option is *off*.
- **By:** Click this selection button to specify the search criteria for the value change of a signal.This button is enabled when an FSDB file is loaded.
- **Search Backward:** Click this button to search backward in time and locate the previous value change. This button is enabled when an FSDB file is loaded.
- **Search Forward:** Click this button to search forward in time and locate the next value change. This button is enabled when an FSDB file is loaded.
- **Cursor Time:** This text field displays the current cursor time and also sets the cursor time. This button is enabled when an FSDB file is loaded.

Browse Cell Summary

Menu Bar: Tools -> Browse/Watch/List -> Browse Cell Summary

This command extracts all of the FFs, latches, FSMs, tri-state buffers, and specified cells in a design. The result summary can be saved into a file.

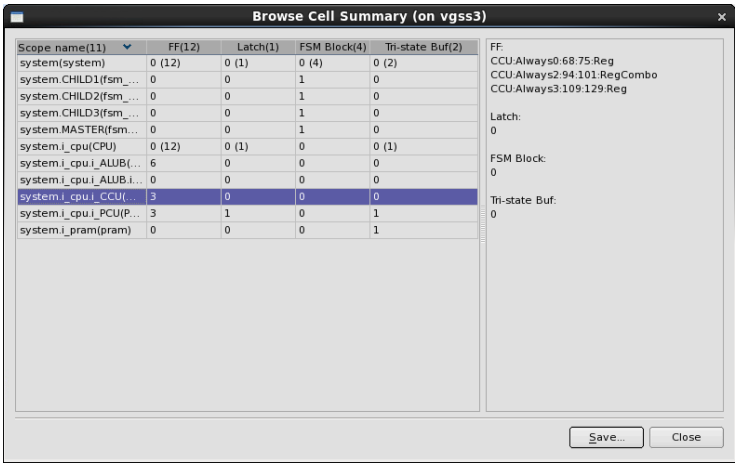


Figure: Browse Cell Summary Form

The left side of the form lists the quantity of **FF**, **Latch**, **FSM Block**, and **Tri-state Buf** for each scope. The total for the **Scope Name**, **FF, Latch**, **FSM Block**, and **Tri-state Buf** cell types are displayed in parentheses in the respective column headings. The total cell number for the scopes which have cells in sub-scopes are also displayed in parentheses. The results vary depending on the preference setting at that time. Choose the column to use for sorting the list by clicking the heading row. The right side of the form lists the detailed instance names in alphabetical order for each category.

The **Save** button opens the *Browse Cell Summary* form where the content can be saved to a file in ASCII format.

Watch Signals

Menu Bar: Tools -> Browse/Watch/List -> Watch Signals

This command opens the *Watch Signals* window where the values of specific signals, transactions, or messages during simulation runs can be displayed. Signal(s) can be dragged from the source code frame or a signal's hierarchical name can be typed in the **Signal** column. The value for each signal in the *Watch Signals* pane is updated interactively and value changes are marked in red. For transaction and message type signals, the label is displayed in the **Value** column.

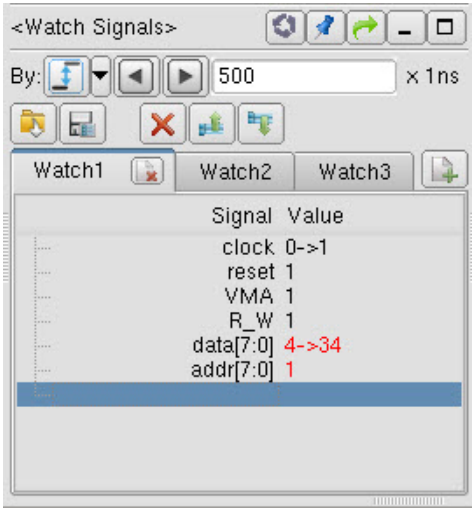


Figure: Watch Frame

The following column headings are available in the *Watch Signals* window:

- **Signal:** The signals in the **Signal** column are the signals to watch. Drag and drop signals from the source code frame or type a signal name with its hierarchy manually.
- **Value:** This column shows the signal values during a simulation run. When the value changes, it is marked in red.

Watch Signals Frame Toolbar Icons and Fields

The following fields and icons are available in the *Watch Signals* window:

- **Search By**  **, Search Backward**  **, Search Forward**  **:** The search criterion for the value change of a signal (**Search By**) can be chosen as **Any Change** and **Falling**. Click **Search Forward** or **Search Backward** to find the next/previous change based on the specified search criteria. Select the signal(s) in the list before invoking these commands. These commands are synchronized with the source code frame.
- **Cursor Time** 

: This text field shows the current cursor time. A cursor time can be typed into this text field to see the simulation results at that time. The **Cursor Time** text field is synchronized with the source code frame.
- **Load**  **:** Click this icon to open the *Load the Signal File* form where the previously saved signal(s) can be loaded into the current **Watch** tab.
- **Save**  **:** Click this icon to open the *Save the Signal File* form where the selected signal(s) from the current **Watch** tab can be saved.

- **Delete**  **:** Click this icon to delete the selected signal from the **Watch** tab.
- **Move Up**  **:** Click this icon to move the selected signal up one row.
- **Move Down**  **:** Click this icon to move the selected signal down one row.

The following icons are associated with the watch tabs:

- **Add Watch Page**  **:** Click this icon to add a new **Watch** tab. The default name is the next WatchN where N is the next numeric value. Double-click the default name to enter edit mode and specify a new name.
- **Delete Current Page**  **:** Click this icon to delete the selected **Watch** tab.

Watch Signals Main Area Right-Click Options

Following is the list of right-click menu options available for the watch signals frame in non-interactive mode.

Format

This command changes the format of the current watch signals tab. The following options are available:

- Binary
- Octal
- Hexadecimal
- ASCII
- Enumerated Literal
- Unsigned Decimal
- Signed 2’s Complement
- Signed 1’s Complement
- Signed Magnitude

Show Full Hierarchy

When this command is turned *on* (checked), the complete hierarchical path for all signals is displayed. When this command is turned *off* (unchecked), only the signal name is displayed.

Delete All

This command removes all signals from the current watch signals tab.

Move Up

This command moves the selected signal up in the list one row.

Move Down

This command moves the selected signal down in the list one row.

Delete

This command deletes the selected signal.

Change Value

NOTE: This command is for Mentor only and is available in interactive mode only.

This command changes the value of a signal.

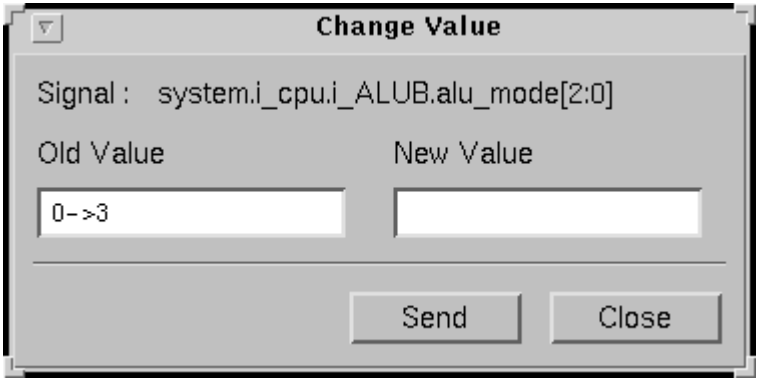


Figure: Change Value Form

When you fill in the form with value and click **Send**, it passes the signal name, the new value, and the old value to the simulator.

Watch Signals Frame Right-Click Options

Following is the list of right-click menu options available for the watch signals window in non-interactive mode.

Add Watch Page

This command adds a new **Watch** tab. The default name is the next *WatchN* where N is the next numeric value.

Delete Current Page

This command deletes the selected **Watch** tab.

Delete Others

This command deletes all **Watch** tabs except for the selected one.

Rename Current Page

This command enters edit mode where a new name can be specified for the selected **Watch** tab.

Watch Expressions

Menu Bar: Tools -> Browse/Watch/List -> Watch Expressions

This command is enabled when an FSDB file is loaded and one term of an expression in the *nTrace* source code frame is selected. For example, assuming the following assignment in the *nTrace* source code frame:

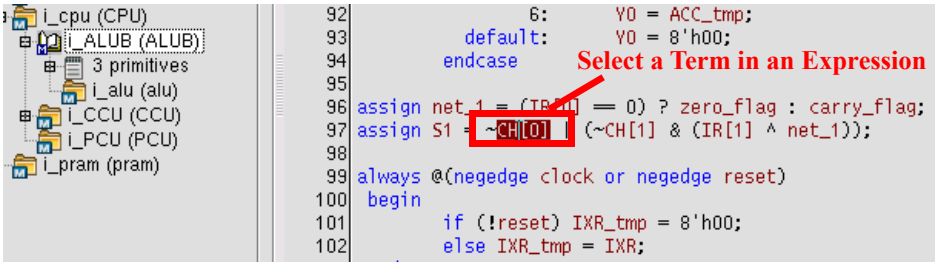


Figure: Assignment in the nTrace Source Code Frame

When one term of an expression in the *nTrace* source code frame is selected and this command is invoked, the *Watch Expressions* window is opened. The *Watch Expressions* window is added as a new tab is the same frame location as the *Message/nWave* windows.

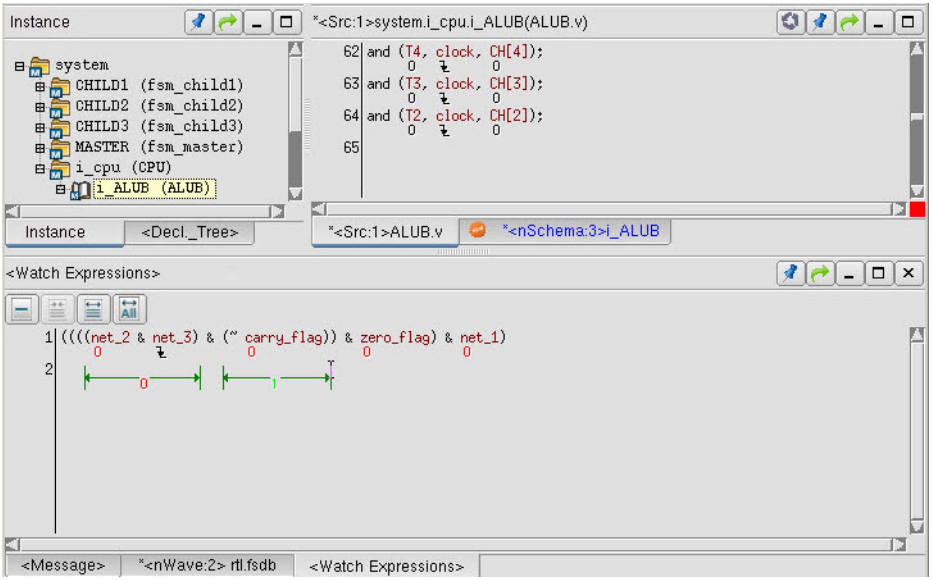


Figure: Watch Expressions Window

The entire combinational logic statement is displayed, along with its active annotation and a horizontal line representing the range of text for the first operation of the statement. The result for the operation is also displayed under the horizontal line.

The following toolbar icons are available:

- **Shrink** : This toolbar icon clears the last step operation result including the horizontal line and its active annotation.
- **Expand :** This toolbar icon displays one more operation. A new horizontal line of the range and the new operation result are appended below the last value annotation.

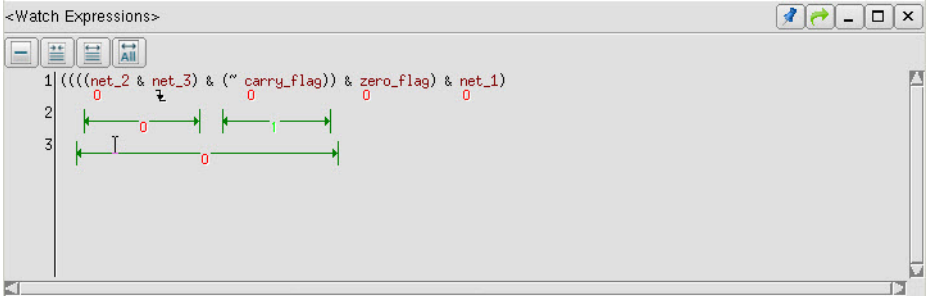


Figure: Watch Expressions Form after Clicking Expand Button Once

- **Expand All** : This toolbar icon displays all results of the operation.

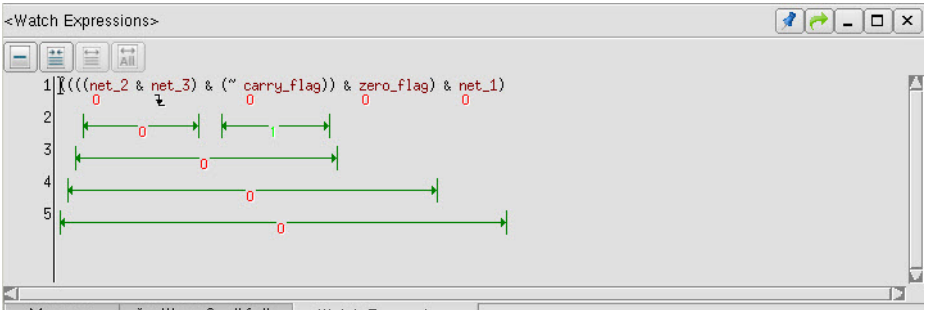


Figure: Watch Expressions Form after Clicking Expand All

- **Show One Level** : When this toolbar icon is turned *on*, only display the results for the current operation step instead of all previous steps.

List X

Menu Bar: Tools -> Browse/Watch/List -> List X

- Refer to the **Tools -> Browse/Watch/List** -> **List X** command in the *nWave* chapter for details.

Browse SDF Path

Menu Bar: Tools -> Browse/Watch/List -> Browse SDF Path

NOTE: An SDF file containing PATHCONSTRAINT information must be loaded for this command to be enabled.

This command generates a partial schematic view that is loaded by the **File -> Load SDF Files** command. SDF is used to exchange a circuit's timing delay and constraint data between different tools. The SDF file contains server paths in SDF format and its reasonable maximum path number can be set to 256. The figure shown reflects the result after specifying an SDF file and clicking the **Add Select Delay Path** button . Double-clicking on the constraint path in the **Path Constraint Browser** highlights the instance in the *Schematic Preview* form. Refer to the *nAnalyzer User's Guide and Tutorial* for details.

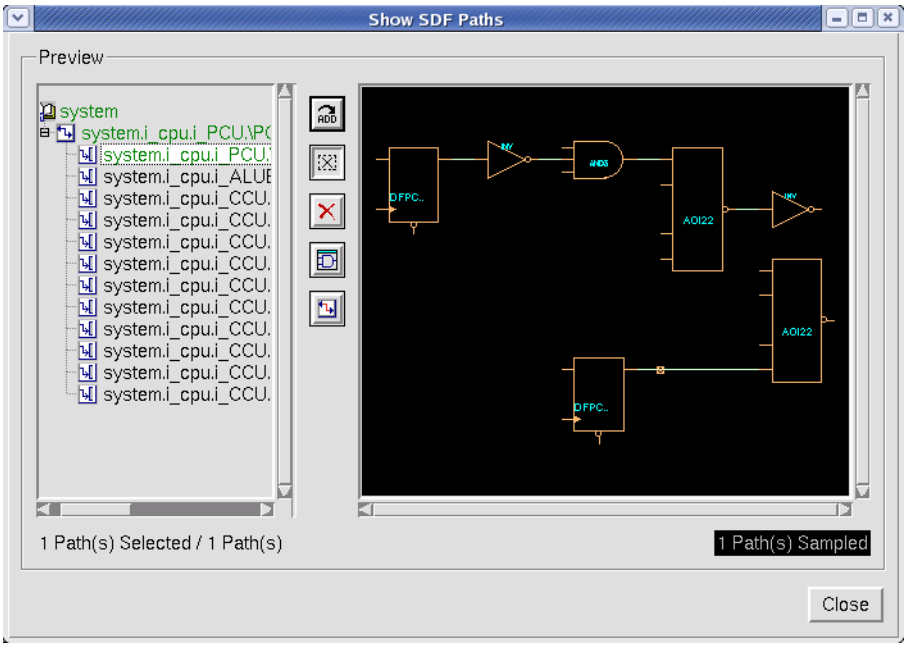


Figure: Schematic Preview Form

Add Selected Delay Path  : Use this button to add the selected path in the **Path Constraint Browser** to the *Schematic Preview* form. The selected constraint path changes to green.

Select All  : This button selects all of the constraint paths in the **Path Constraint Browser**.

Delete All in Preview  : This button deletes all of the schematic views in the *Schematic Preview* form.

New Schematic  : This button opens a schematic window and displays the partial schematic view according to the constraint paths selected in the **Path Constraint Browser**. In the *Schematic Preview* form, you have only viewing and finding capabilities (such as **Zoom In**, **Zoom Out**, **Zoom All**, **Pan Left**, **Pan Right**, **Pan Up**, **Pan Down**, **Last View** and **Find**), while in the schematic window you have the full capabilities of *nSchema*. You can expand a circuit path in the *Schematic Preview* form by double-clicking on the node in which you are interested.

Display Hit Rate  : This button opens the *SDF Hit Rate List* form. It lists the signal path hit rate for the selected sampled path(s). Make changes to the **Selected Set** or **Hit Rate** by selecting a color box under the **Color Preference Setting** section.

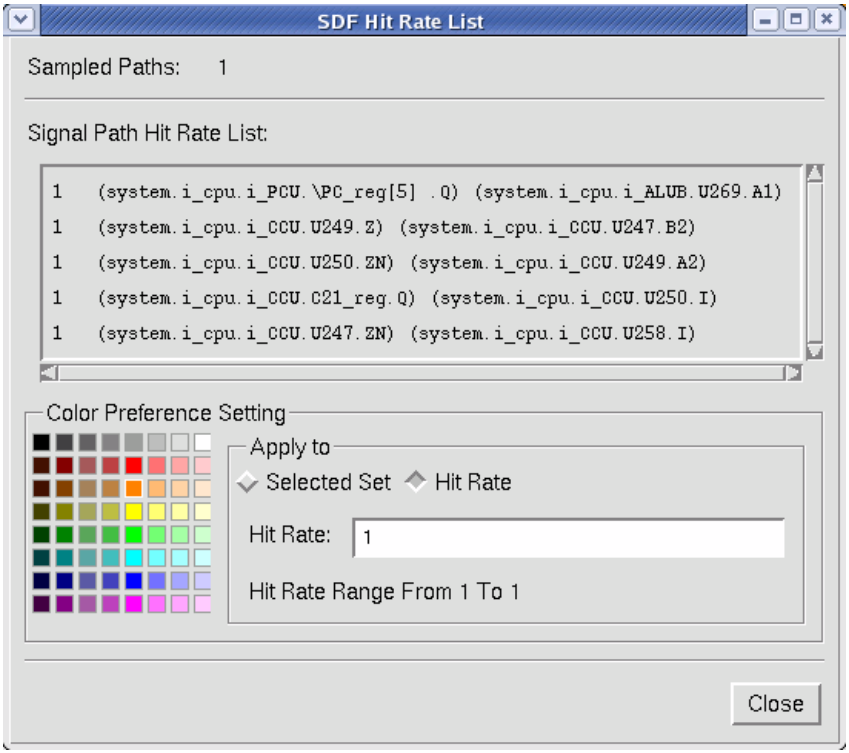


Figure: SDF Hit Rate List

Each PATHCONSTRAINT contains a list of INTERCONNECT and each INTERCONNECT specifies the from/to instport pair. The **Hit Rate** is the number of PATHCONSTRAINT which passes through the Interconnect and indicates where a bottleneck may exist in the design. For example, consider the following:

```
PATHCONSTRAINT 1 : (a.o, b.i) (b.o, c.i1) (c.o, d.i)
PATHCONSTRAINT 2 : (x.o, y.i) (y.o, c.i2) (c.o, d.i)
```

In this case, the path hit associated with *(a.o, b.i)*, *(b.o, c.i1)*, *(x.o, y.i)* and *(y.o, c.i2)* is 1, and the path hit associated with *(c.o, d.i)* is 2. If you change the color of **Hit Rate** 2 to red, the net between *c.o* and *d.i* is then set to red.

Register

Menu Bar: Tools -> Register

This command opens a *Register Window* where signals can be added and edited and then saved to a register file. Refer to the *Register Window* chapter for details.

Memory

Memory Definition Table

Menu Bar: Tools -> Memory -> Memory Definition Table

This command opens the *Memory Definition Table* form where conditions to control a memory can be defined. Each module/array pair that is defined can have one or more initial, write, or read operations. The same operation type can be used multiple times for the same array. The default form contains information if one or more memory definitions are preloaded; otherwise, the form is in blank.

There are two methods available to preload a memory definition file:

1. From the Verdi command line using the **-mdt** option. This option accepts a single memory definition file or a text file that contains a list of memory definition files and paths (each file name must be on a new line).
2. Using the *NOVAS_MDT_LIBS* and *NOVAS_MDT_LIBPATHS* environment variables. *NOVAS_MDT_LIBS* is used to specify one or more memory definition file and *NOVAS_MDT_LIBPATHS* is used to specify the directory location(s) of these files.

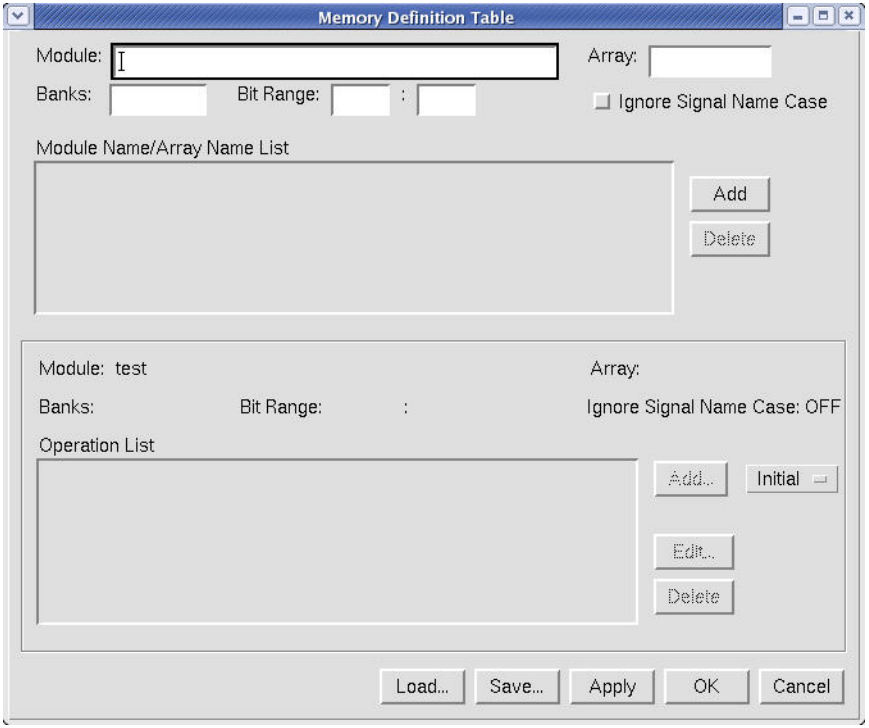


Figure: Memory Definition Table Form

The following fields, options and buttons are available on the *Memory Definition Table* form:

- **Module:** Specify the memory in this text field. The memory is either a Verilog module or a VHDL entity(architecture) pair. It is recommended that the module name matches an existing memory model. The module text field accepts regular expressions.
- **Array:** Specify a unique memory identifier. Although it is recommended that the array name matches an existing memory definition, it is not required. The memory definition table can be used to create descriptions for memories that do not exist in the current design.
- **Banks:** Specify a numeric value for the number of banks available in the memory. The application for the banks field is memories with mask-able writes or byte enables. This field is optional. If no value is entered, the default is *1*.

  If a value greater than 1 is entered in this field, the corresponding signal must be entered in the **Bank Selection** text field in the Write and Read operation definitions, and the bit range must be specified. When banks is

assigned to '*', the bit width of the bank selection signal is used to determine the number of banks.

- **Bit Range:** Specify the word size for the memory. The value can be entered as most significant/least significant (for example, 31:0) or least significant/most significant (0:31). If no value is entered, the Verdi platform attempts to extract the range from the design or the read/write expression; otherwise, (31:0) is used by default.
- **Ignore Signal Name Case:** When this option is turned *on*, the case in all signal names associated with the memory in the FSDB file is matched using one of following variations: no case change (for example, Addr), all uppercase (for example, ADDR) or all lowercase (for example, addr). When this option is turned *off*, the case in all signal names associated with the memory in the FSDB file is matched (for example, addr=addr and ADDR=ADDR only). The default value of this option is *off*. This option must be specified before the memory is created for it to be associated with that memory.
- **Load:** This command invokes the *File Manager* form for you to select a memory definition file for loading. Refer to the *File Manager Form* section for details.
- **Save:** This button opens the *File Manager* form where the directory structure can be viewed and a file name for saving the defined memories can be specified.
- **Apply:** This button applies changes to the operation list of the selected module/array and changes to the list of memory arrays and leaves the *Memory Definition Table* form open.
- **OK:** This button applies changes to the operation list of the selected module/array and changes to the list of memory arrays and closes the *Memory Definition Table* form.
- **Cancel:** This button cancels changes to the operation list of the selected module/array and changes to the list of memory arrays and closes the *Memory Definition Table* form.

The following buttons are associated with the **Module Name/Array Name List** field:

- **Add:** Click this button to add the specified module/array name to the list in the **Module Name/Array Name List** text field.
- **Delete:** Click this button to delete the selected (left-click) module/array name from the **Module Name/Array Name List** text field. Only one item can be selected at a time.

The following buttons are associated with the **Operation List** field:

- **Add:** Click this lower **Add** button to add a new operation to the list based on the selection of the menu located on the right of this **Add** button. The available selection operations include **Initial**, **Write**, **Read** and **Bypass**. Select the preferred operation and click the **Add** button to open an edit form. Refer to the *Initial Operation Form*, *Write Operation Form*, *Read Operation Form*, and *Bypass Operation Form* for details about each operation in the *Memory Definition Table Editing Window*.
- **Edit:** Click this button to edit the selected operation. The associated operation form opens when clicking the **Edit** button. Only one operation can be selected at a time.
- **Delete:** Click this button to remove the selected operation from the list.

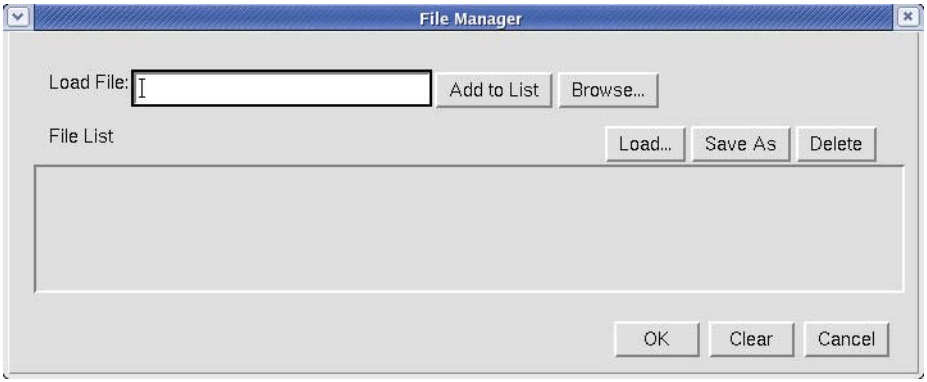


Figure: File Manager Form

The *File Manager* form includes the following text fields and buttons.

- **Load File:** Specify the complete directory path and file name for the memory definition file. The information can be entered manually or click the **Browse** button to open the *File Manager* where the directory structure can be viewed and a previously saved file can be loaded. After the memory definition file information is entered, click the **Add to List** button to add the file to the **File List** text field.
- **File List:** This text field contains a list of the memory definition files that have been loaded.
- **Load:** This button opens the *File Manager* form where the directory structure can be viewed and a previously saved file list can be loaded. This file contains a list of memory definition files. See the following for an example file:

```
mem.def
/home/proj/mem/ram.def
```

- **Save As:** This button opens the *File Manager* form where the directory structure can be viewed and a file name to save the memory definition file list to can be designated. The list is saved exactly as it appears in the text field.
- **Delete:** The delete button removes the selected memory definition files from the **File List** text field. Selection occurs with a left-click on the file name. Multiple files can be selected.
- **OK:** This button closes the window and compile/load the memory definition files. The details in the file list text field are saved.
- **Clear:** This button clears the entire file list. The form remains open.
- **Cancel:** This button closes the window without loading the files. The details in the file list text field are saved.

Initial Operation Form

Use the following form to specify the various conditions that define the initialization operation for the array. This form is opened when clicking the **Add** button while **Initial** is selected.

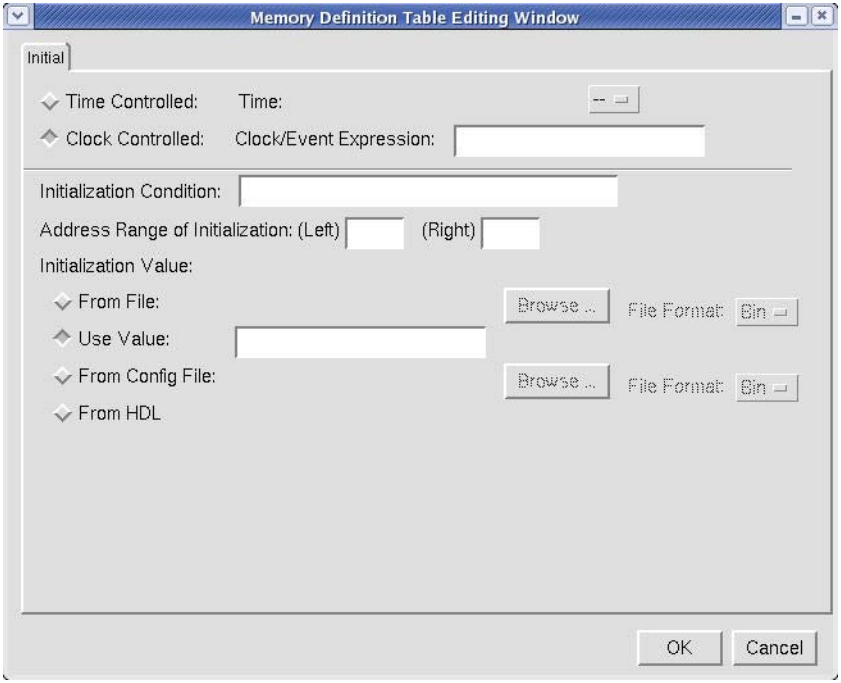


Figure: Memory Definition Table Editing Window - Initial Operation

The following fields and options are available:

- **Time Controlled:** Specify a time for the array initialization. When this option is turned *on*, a value must be specified in the time text field. The time units can be toggled between mili-seconds (ms), nano-seconds (ns), or micro-seconds (us). If an FSDB file is loaded, the default time units matches the units in the *nWave* window. If the time controlled option is selected, the clock controlled option cannot be selected.
- **Clock Controlled:** Specify an expression that indicates the clock edge for the array initialization. When this option is turned *on*, a Verilog expression (for example, @(posedge clk)) must be entered in the text field for the trigger clock or event. If the clock controlled option is selected, the time controlled option cannot be selected.
- **Initialization Condition:** Specify an expression that enables the array initialization. A signal or Verilog expression must be entered in the text field (for example, init==1). This field is optional.
- **Address Range of Initialization:** Specify the address range of the array to be initialized. If both fields are blank, the entire address range is assumed. When this option is used, a value must be entered in both fields.
- **Initialization Value:** Three methods for initializing the array are supported, from a file, from a configuration, and set to a value. Only one of these options can be selected at a time.
  - **From File:** Specify the complete directory path and file name for the memory initialization file. The initialization file is a list of values in either binary or hexidecimal format. The path and the file name can be entered manually or click the **Browse** button to open the *File Manager* where the directory structure can be viewed and a file can be loaded. The file format can be toggled between binary (**Bin**) and hexidecimal (**Hex**). Select the format that matches the specified file.
  - **Use Value:** Specify the initialization value. The bit width should match the width of the array element. If the value is short, leading zeros is added. If the value is long, it is truncated from the most significant bit(s). This value is applied to the entire array.

NOTE: A predefined system function -- $size_of(expression) can be used in the **Use Value** field to define a constant of dynamic bit width. For example: {$size_of(INIT){1'b0}}
This statement gets the bit width of signal INIT and generate a constant '0' of equal bit width.

  - **From Config File:** Specify the complete directory path and file name for the memory configuration file. The information can be entered manually or the **Browse** button can be clicked to open the *File Manager*

form where the directory structure can be viewed and a file can be loaded. The configuration file is used to specify different initialization files for different instantiations of the same memory. The format is:

```
instName1  file1
instName2  file2
...
    You can use regular expressions to specify instance names.
     For example:
Top.u0.u1  u0Mem.init
Top.u2.*   u2Mem.init
*          otherMem.init
```

NOTE: If multiple rules match for an instance name, the first rule in the list that matches is used and subsequent matches is ignored.

- **From HDL:** Use the initialization value defined in HDL. This includes initialization by $readmemb or $readmemh if it is specified in HDL.

Write Operation Form

Use the following form to specify the various conditions that define the write operation for the array. This form is opened when clicking the **Add** button while **Write** is selected.

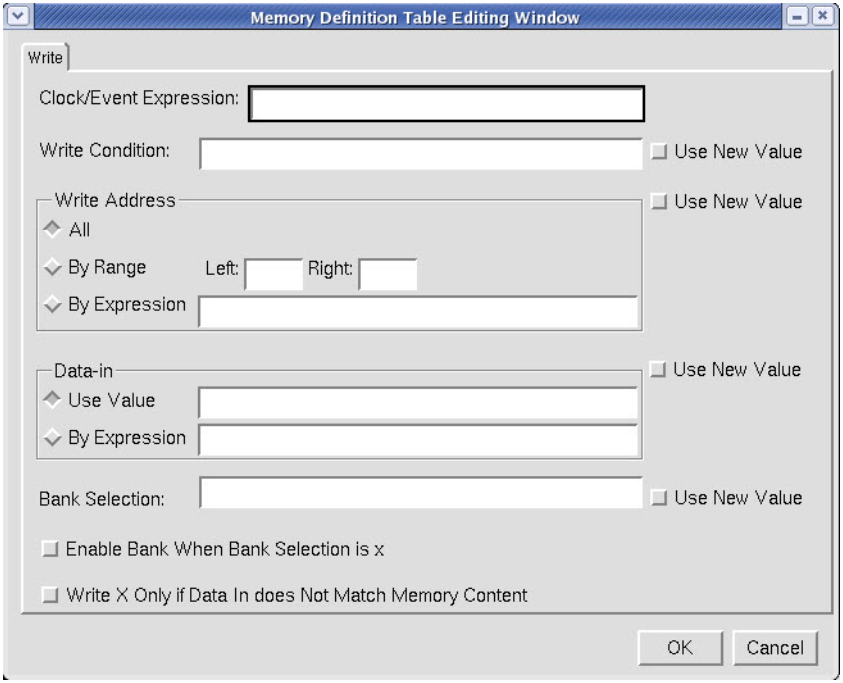


Figure: Memory Definition Table Editing Window - Write Operation

NOTE: The **Write Condition**, **Write Address**, **Data In**, and **Bank Selection** fields accept two predefined preprocessor operators: "'<+>" and "'<->"
"'<+> NAME" is used to indicate the concatenation of NAME0, NAME1, ... , NAMEn-1, NAMEn
"'<-> NAME" is used to indicate the concatenation of NAMEn, NAMEn-1, ... NAME1, NAME0
where 'NAME' is the prefix of the signal name and 'n' can vary.
The Verdi platform collects all port signal names (signals defined inside modules or architectures are ignored) with the pattern of NAME plus a number and concatenate the signals as increasing (+) or decreasing (-) order. The number suffix does not need to be continual that is, {NAME1, NAME3, NAME5} is recognized.

NOTE: A predefined system function -- $size_of(expression) can be used in any field where you want to define a constant of dynamic bit width. For example, if you specify {$size_of(DIN){1'bx}}in the **Data In** field, the bit width of signal DIN is used to generate a constant 'X' of equal bit width.

- **Clock/Event Expression:** Specify the clock expression for the array write operation. The text field accepts Verilog expressions of the form

@(event_expression). For example, @(posedge signal), @(negedge signal) @(signal), or some combination of signals or events. This field is mandatory.

- **Write Condition:** Specify an expression that enables the array write. The field accepts a signal or Verilog expression (for example, strb & ~rw). This field is optional.
- **Use New Value** (following **Write Condition** field): When this option is turned *on*, the write condition is evaluated after the clock edge. When this option is turned *off*, the write condition is evaluated before the clock edge.
- **Write Address:** This section specifies the memory address from one of the following options: **All**, **By Range**, and **By Expression**.
  - **All:** Enables all bits of each dimension to be used as write address.
  - **By Range:** Specify the range of the memory address. When this option is turned *on*, the **Left:** and **Right:** fields is enabled and the left and right address values can be entered to specify the range of the memory write address.
  - **By Expression:** Specify the write address signal for the array. This field accepts a signal or Verilog expression (for example, addr). To indicate the entire range, assign the write address to '*'. This field is mandatory.
- **Use New Value** (following **Write Address** section): When this option is turned *on*, the write address is evaluated after the clock edge. When this option is turned *off*, the write address is evaluated before the clock edge.

NOTE: Two predefined system functions -- $encode_left() and $encode_right() can be used in the address field to convert a decoded address into normal form. For example, consider:
reg [0:7] decoded_address = 8'b10000000
$encode_left(decoded_address) = 7. ('1' appears at bit location 7).
or when decoded_address = 8'b00000010,
$encode_left(decoded_address) = 1.

Usually, a decoded address is one-hot, which means only one bit of the vector is '1'. Therefore, the results of $encode_left() and $encode_right() is same. However, if more than one bit in a vector is '1', $encode_left searches for '1' from the left while $encode_right searches for '1' from the right. For example, when:
decoded_address = 8'b10000010,
$encode_left(decoded_address) = 7 and
$encode_right(decoded_address) = 1.

When none of the bits in a vector is '1', both $encode_left() and $encode_right() results in 'x' when applied to the vector.

- **Data In:** This section specifies the memory address from one of the following options: **Use Value** and **By Expression**.
  - **Use Value:** Specify the input data value for each dimension.
  - **By Expression:** Specify the input data signal for the array. This field accepts a signal or Verilog expression (for example, temp_data). This field is mandatory.
- **Use New Value** (following **Data In** section): When this option is turned *on*, the data is evaluated after the clock edge. When this option is turned *off*, the data is evaluated before the clock edge.
- **Bank Selection:** Specify the memory bank selection signal or expression for the array. When the bank selection is not specified, the entire word is written. When bank selection is specified, the write operation consists of multiple bank-operations which performs one write operation on each bank according to the bank select condition. For example:

```
always @(clock_expression)
begin
    if (cond_expression) begin
       for (i=0; i<#banks; i++) // i is inline to a CONSTANT
          if (bsExp[i]) Mem[address][i] = dataExp[i];
    end
end
```

Each bank operation is constructed from one control bit of bsExp, one bank of the memory and one bank of dataExp. The data banks are divided into the same width as the memory bank. This field is optional.

- **Use New Value** (following **Bank Selection** field): When this option is turned *on*, the bank selection condition is evaluated after the clock edge.When this option is turned *off*, the bank selection condition is evaluated before the clock edge.
- **Enable Bank When Bank Selection is x**: When this option is turned *on*, the bank with the selected signal “x” can be enabled. The default value of this option is *off*.
- **Write X Only if Data In does Not Match Memory Content:** When this option is turned *on*, “X” shows when the data does not match the memory content. The default value of this option is *off*.

Read Operation Form

Use the following form to specify the various conditions that define the read operation for the array. This form is opened when clicking the **Add** button while **Read** is selected.

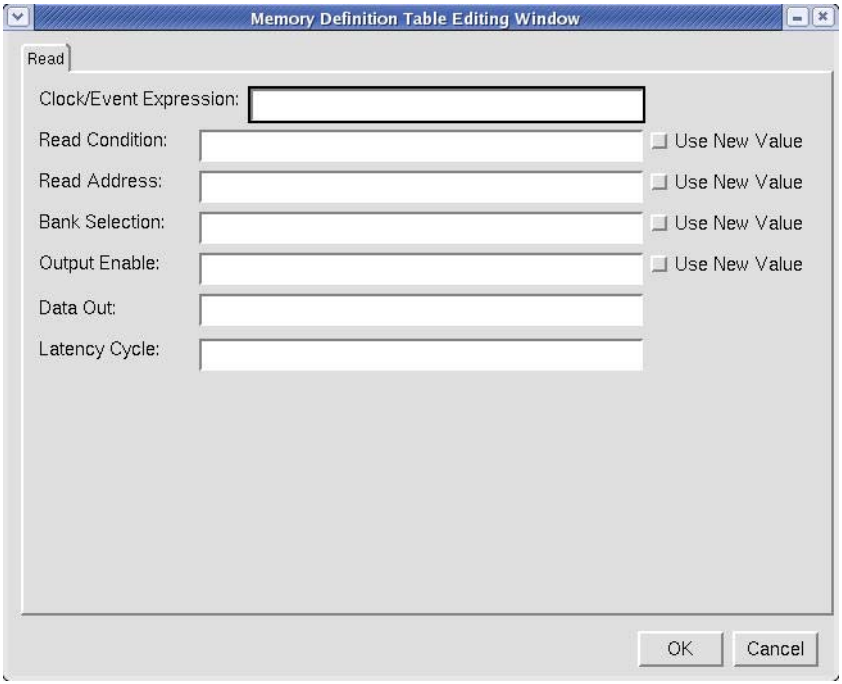


Figure: Memory Definition Table Editing Window - Read Operation

NOTE: The **Read Condition**, **Read Address**, **Bank Selection** and **Data Out** fields accept two predefined preprocessor operators:“'<+>” and “'<->”
“<+> NAME” is used to indicate the concatenation of NAME0, NAME1, ... , NAMEn-1, NAMEn
“'<-> NAME” is used to indicate the concatenation of NAMEn, NAMEn-1, ... NAME1, NAME0
where 'NAME' is the prefix of the signal name and 'n' can vary.
The Verdi platform collects all port signal names (signals defined inside modules or architectures are ignored) with the pattern of NAME plus a number and concatenate the signals as increasing (+) or decreasing (-) order. The number suffix does not need to be continual for example,{NAME1, NAME3, NAME5} is recognized.

The following fields and options are available:

- **Clock/Event Expression:** Specify the clock expression for the array read operation. The text field accepts Verilog expressions of the form @(event_expression). For example, @(posedge signal), @(negedge signal) @(signal), or some combination of the previous list. This field is mandatory.

- **Read Condition:** Specify an expression that enables the array read. The text field accepts a signal or Verilog expression (for example, strb & rw). This field is optional.

  If the **Use New Value** option is turned *on*, the read condition is evaluated after the clock edge. If the **Use New Value** option is turned *off*, the read condition is evaluated before the clock edge.

- **Read Address:** Specify the read address signal for the array. The text field accepts a signal or Verilog expression (for example, addr). This field is mandatory.

  If the **Use New Value** option is turned *on*, the read address is evaluated after the clock edge. If the **Use New Value** option is turned *off*, the read address is evaluated before the clock edge.

NOTE: Two predefined system functions -- $encode_left() and $encode_right() can be used in the address field to convert a decoded address into normal form. For example, consider:
reg [0:7] decoded_address = 8'b10000000
$encode_left(decoded_address) = 7. ('1' appears at bit location 7).
or when decoded_address = 8'b00000010,
$encode_left(decoded_address) = 1.

Usually, a decoded address is one-hot, which means only one bit of the vector is '1'. Therefore, the results of $encode_left() and $encode_right() is the same. However, if more than one bit in a vector is '1', $encode_left searches for '1' from the left while $encode_right searches for '1' from the right. For example, when:
decoded_address = 8'b10000010,
$encode_left(decoded_address) = 7 and
$encode_right(decoded_address) = 1.

When none of the bits in a vector is '1', both $encode_left() and $encode_right() results in 'x' when applied to the vector.

- **Bank Selection:** Specify the memory bank selection signal or expression for the array. This field is optional. When the bank selection is not specified, the entire word is read.

  When bank selection is specified, it means the read operation consists of multiple bank-operations which performs one read operation on each bank according to the bank select condition. For example:

```
always @(clock_expression)
begin
    if (cond_expression) begin
       for (i=0; i<#banks; i++) // i is inline to a CONSTANT
          if (bsExp[i]) dataout[i] = Mem[address][i];
```

```
    end
end
```

Each bank operation is constructed from one control bit of bsExp, one bank of the memory and one bank of dataout. The data banks are divided into the same slice width as a memory bank.

If the **Use New Value** option is turned *on*, the bank selection condition is evaluated after the clock edge. If the **Use New Value** option is turned *off*, the bank selection condition is evaluated before the clock edge.

- **Output Enable:** Specify an expression for the output enable of the array. The text field accepts a signal or Verilog expression (for example, oe). This field is optional.

  If the **Use New Value** option is turned *on*, the output enable is evaluated after the clock edge. If the **Use New Value** option is turned *off*, the output enable is evaluated before the clock edge.

- **Data Out:** Specify the output data signal for the array. The text field accepts a signal (for example, dataout) or a concatenated Verilog expression (for example, {signal1,signal2, signal3}). This field is mandatory.
- **Latency Cycle:** Specify the number of cycles to wait after a read operation is initiated before sampling the output data. The text field accepts a numeric value greater than or equal to 1. This field is optional.

Bypass Operation Form

Use the following form to specify the various conditions that define the bypass read operation for the array. This form is opened when clicking the **Add** button while **Bypass** is selected.

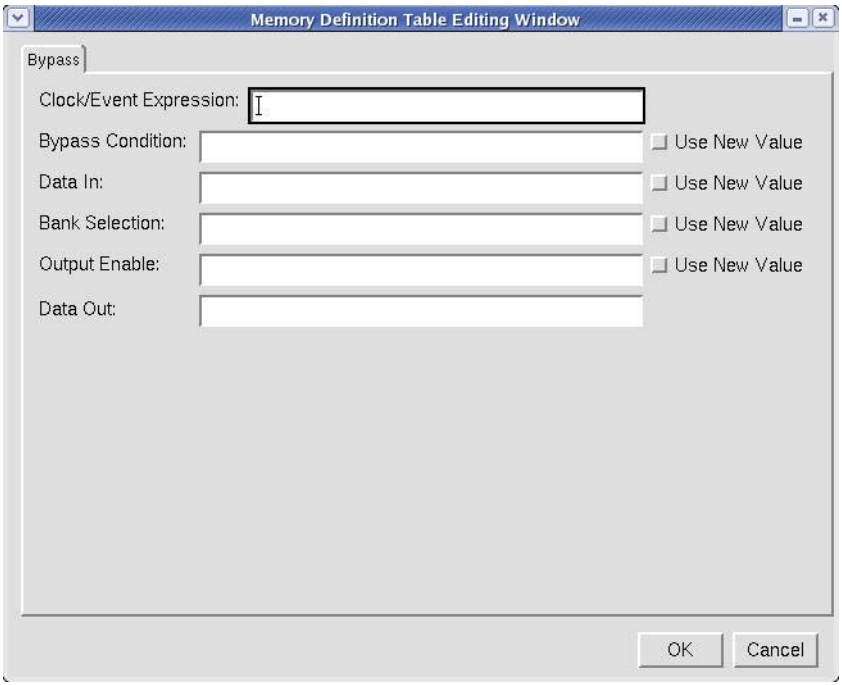


Figure: Memory Definition Table Editing Window - Bypass Operation

NOTE: The **Bypass Condition**, **Data In**, **Bank Selection**, **Output Enable**, and **Data Out** fields accept two predefined preprocessor operators: “'<+>” and “'<->”
“'<+> NAME” is used to indicate the concatenation of NAME0, NAME1, ... , NAMEn-1, NAMEn
“'<-> NAME” is used to indicate the concatenation of NAMEn, NAMEn-1, ... NAME1, NAME0
where 'NAME' is the prefix of the signal name and 'n' can vary.
The Verdi platform collects all port signal names (signals defined inside modules or architectures are ignored) with the pattern of NAME plus a number and concatenate the signals as increasing (+) or decreasing (-) order. The number suffix does not need to be continual that is, {NAME1, NAME3, NAME5} is recognized.

NOTE: A predefined system function -- $size_of(expression) can be used in any field where you want to define a constant of dynamic bit width. For example, if you specify {$size_of(DIN){1'bx}}in the **Data In** field, the bit width of signal DIN is used to generate a constant 'X' of equal bit width.

The following fields and options are available:

- **Clock/Event Expression:** Specify the clock expression for the bypass operation. The text field accepts Verilog expressions of the form @(event_expression). For example, @(posedge signal), @(negedge signal) @(signal), or some combination of the previous list. This field is mandatory.
- **Bypass Condition:** Specify an expression that enables the bypass condition for the array read. The text field accepts a signal or Verilog expression (for example, strb & rw). This field is optional.

  If the **Use New Value** option is turned *on*, the bypass condition is evaluated after the clock edge. If the **Use New Value** option is turned *off*, the bypass condition is evaluated before the clock edge.
- **Data In:** Specify the data input signal for the array. The text field accepts a signal or Verilog expression (for example, data). This field is mandatory.

  If the **Use New Value** option is turned *on*, the data in is evaluated after the clock edge. If the **Use New Value** option is turned *off*, the data in is evaluated before the clock edge.
- **Bank Selection:** Specify the memory bank selection signal or expression for the array. This field is optional. When the bank selection is not specified, the entire word is read.

  When bank selection is specified, it means the bypass read operation consists of multiple bank-operations which performs one read operation on each bank according to the bank select condition. For example:

```
always @(clock_expression)
begin
    if (cond_expression) begin
       for (i=0; i<#banks; i++) // i is inline to a CONSTANT
          if (bsExp[i]) dataout[i] = datain[i];
    end
end
```

  Each bank operation is constructed from one control bit of bsExp, one bank of the datain and one bank of dataout.

  If the **Use New Value** option is turned *on*, the bank selection condition is evaluated after the clock edge. If the **Use New Value** option is turned *off*, the bank selection condition is evaluated before the clock edge.
- **Output Enable:** Specify an expression for the output enable of the array. The text field accepts a signal or Verilog expression (for example, oe). This field is optional.

If the **Use New Value** option is turned *on*, the output enable is evaluated after the clock edge. If the **Use New Value** option is turned *off*, the output enable is evaluated before the clock edge.

- **Data Out:** Specify the output data signal for the array. The text field accepts a signal (for example, dataout) or a concatenated Verilog expression (for example, {signal1,signal2, signal3}). This field is mandatory.

Memory/MDA

Menu Bar: Tools -> Memory -> Memory/MDA

NOTE: To enable this command, an FSDB file must be loaded into the Verdi platform.

This command opens the *nMemory* window. Refer to the *Memory/MDA Pane* chapter for details.

Property Tools

Evaluator

Menu Bar: Tools -> Property Tools -> Evaluator

Bind Key: Shift+E

NOTE: A design with SystemVerilog Assertions (SVA) code and a corresponding FSDB file must be loaded into the Verdi platform to enable this command.

This command opens the *Evaluate Properties* form where all SystemVerilog Assertions are listed and the assertions to be evaluated can be selected. Refer to the *Evaluate Properties Form* section in the *Assertion Debug* chapter for details.

Statistics

Menu Bar: Tools -> Property Tools -> Statistics

This command opens the *Statistics* window. Refer to the *Statistics Pane* section in the *Assertion Debug* chapter for details.

Analyzer

Menu Bar: Tools -> Property Tools -> Analyzer

This command opens the *Analyzer* window. Refer to the *Analyzer Pane* section in the *Assertion Debug* chapter for details.

Add Temporary Assertions

Menu Bar: Tools -> Property Tools -> Add Temporary Assertions

This command opens the *Add Temporary Assertions* form. Refer to the *Add Temporary Assertions Form* section in the *Assertion Debug* chapter for details.

Transaction Debug

Transaction and Protocol Analyzer

Menu Bar: Tools -> Transaction Debug -> Transaction and Protocol Analyzer

This command opens the *Transaction and Protocol Analyzer* window that provides a software-oriented waveform view (compared to the hardware hierarchical view in the *nTrace* window) for testbench debugging. The waveform view supports new transaction types and the virtual streams that are recognized transactions from different streams. The *Transaction and Protocol Analyzer* window can be used to debug and analyze transactions and their temporal activity by the time and stream axes.

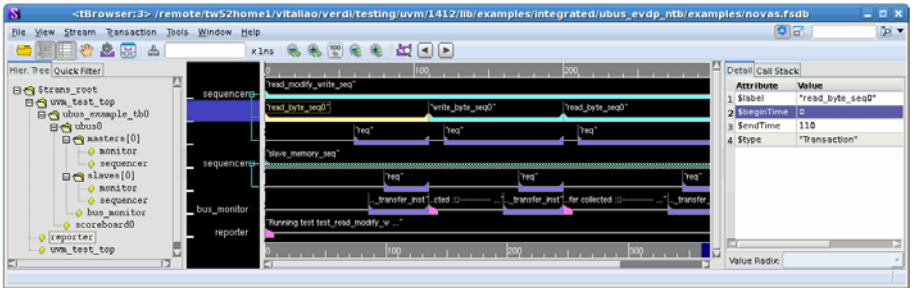


Figure: Transaction and Protocol Analyzer Window

Transaction Table View

Menu Bar: Tools -> Transaction Debug -> Transaction Table View

This command opens the *Transaction Table View* window that displays streams in the FSDB file as a spreadsheet view of transaction types. The contents are based on data-mining (during loading) or other operations (for example, drag and drop streams from other windows or created virtual streams). Read and analyze the streams by the tabular highlight form. The relationship between transactions can also be shown in the *Transaction Table View* window.

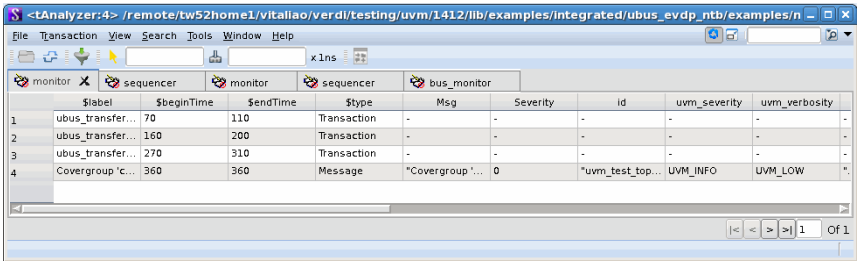


Figure: Transaction Table View Window

Classic Transaction

The **Classic Transaction** command includes two sub-commands: **Evaluator** and **Analysis Window**.

Evaluator

Menu Bar: Tools -> Transaction Debug -> Classic Transaction -> Evaluator

The **Evaluator** command is enabled when a design with SystemVerilog Assertions (SVA) code and a corresponding FSDB file are loaded into the Verdi platform. The command opens the *Transaction Evaluator* form where all SystemVerilog Assertions are listed, the assertions to be evaluated as transactions can be selected, and the results file name specified. Assertions can also be dragged to the *nTrace* source code frame to see the related code.

Refer to the *Verdi User Guide and Tutorial* for details on writing assertions to be used specifically for transaction extraction.

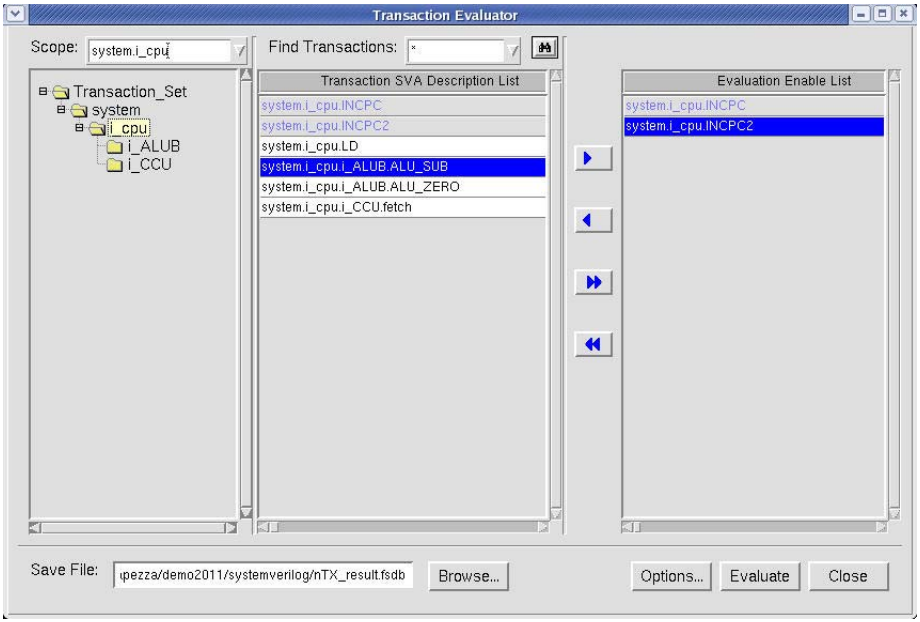


Figure: Transaction Evaluator Form

In the *Transaction Evaluation* form, the left pane displays the design hierarchies that have underlying SVA, the center pane lists all transactions for the current scope and the right pane displays the transactions selected for evaluation.

The following fields and buttons are available:

- **Scope:** This text field displays the scope selected in the **Hierarchical Scope Tree** pane. A partial scope name can be entered in the **Scope** text field, and the matched scope(s) is highlighted. It is case-insensitive.
- **Find Transactions:** This text field is used to filter the transactions to display for the current selected scope. It is case-insensitive. For example, if "C*" is typed in the **Find Transactions** text field and the **Find** icon clicked (or the **Enter** key on the keyboard pressed), the **Transaction SVA Description List** is updated to show all the transactions that start with C or c.
- **Transaction SVA Description List:** This section lists all of the transaction SVA descriptions in the selected scope. If the **Search in Sub-scope** option in the *Options* form is turned *off*, this pane only lists the transaction SVA descriptions in the selected sub-scope. Left-click to select one row at a time, or Shift-left-click or drag-left to select a range, or Ctrl-left-click to select non-continuous options. Click the header to sort by the SVA transaction descriptions.

- **Evaluation Enable List:** This section lists all the transactions added from the **Transaction SVA Description List**. These are the transactions that is evaluated when the **Evaluate** button is clicked. Left-click to select a row at a time, or Shift-left-click or drag-left to select a range, or Ctrl-left-click to select non-continuous options. Click the header to sort by the **Evaluation Enable List**.

  After selecting the transactions of interest in the middle section (or **Transaction SVA Description** list), add the desired transaction to the **Evaluation Enable List** by drag-and-drop or clicking the **Add Selected Transaction**  icon. Click the **Add All Transactions**  icon to add all transactions. After adding transactions to the **Evaluation Enable List**, remove unwanted transactions by clicking the **Delete Selected Transaction**  icon (to remove a single selection) or the **Delete All Transactions**  icon (to remove all transactions). The transaction(s) that are already added to the **Evaluation Enable List** and selected for evaluation is marked in blue and the background turns gray.

- **Save File:** Specify the FSDB file name for saving the transaction evaluation results. To change the default, type directly in the text field or click the **Browse** button to open the *FSDB File* form where the directory structure can be viewed and an FSDB file name specified. The default FSDB file name is *nTX_result.fsdb*.
- **Options:** This button opens the *Options* form where the options to use in the *Transaction Evaluator* form can be configured after extraction. Refer to the *Options Form* section for details.
- **Evaluate:** Click this button to extract the transactions listed in the **Transaction Enable List** from the assertion code and save the results to the specified FSDB file.

Options Form

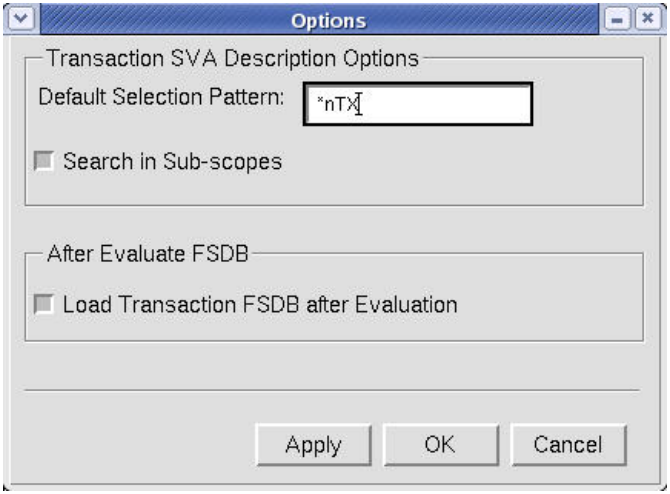


Figure: Options Form

This form has the following fields and options:

- **Default Selection Pattern:** Specify the default pattern to search for in the Transaction SVA Description List. The specified default pattern is entered in the **Find Transactions** text field. The default pattern is **nTX*.
- **Search in Sub-scopes:** When this option is turned *on*, all transactions under the scope and its sub-scopes is shown in the Transaction SVA Description List. When this option is turned *off*, the transactions under the selected sub-scope is shown in the Transaction SVA Description List. The default value of this option is *on*.
- **Load Transaction FSDB after Evaluation:** When this option is turned *on*, the transaction FSDB is automatically loaded into the Verdi platform after evaluation. When this option is turned *off*, the transaction FSDB is not loaded into the Verdi platform after evaluation. The default value of this option is *on*.

Analysis Window

Menu Bar:	Tools -> Transaction Debug -> Classic Transaction -> Analysis Window

The **Analysis Window** command opens the *Transaction/Message Analyzer* window. Refer to the *Transaction/Message Analyzer* chapter for details.

Classic Message

The **Classic Message** command includes one sub-command: **Analysis Window**.

Analysis Window

Menu Bar: Tools -> Transaction Debug -> Classic Message -> Analysis Window

This command opens the *Transaction/Message Analyzer* window. Refer to the *Transaction/Message Analyzer* chapter for details.

UVM/OVM/VMM Debug

Menu Bar: Tools -> UVM/OVM/VMM Debug

The UVM/OVM/VMM Debug command includes six sub-commands: **All Views**, **Resource/Config View**, **Factory View**, **Phase View**, **Sequence View**, and **Register View**.

All Views

Menu Bar: Tools -> UVM/OVM/VMM Debug -> All Views

This command opens the *Resource View*, *Factory View*, *Phase View*, *Sequence View*, and *Register View* windows for the UVM Interactive Simulation Debug support.

Resource/Config View

Menu Bar: Tools -> UVM/OVM/VMM Debug -> Resource/Config View

This command opens the *Resource View* window. Refer to the **Resource View Pane** section in the *UVM Interactive Simulation Debug* chapter for details.

Factory View

Menu Bar: Tools -> UVM/OVM/VMM Debug -> Factory View

This command opens the *Factory View* window. Refer to the **Factory View Pane** section in the *UVM Interactive Simulation Debug* chapter for details.

Phase View

Menu Bar: Tools -> UVM/OVM/VMM Debug -> Phase View

This command opens the *Phase View* window. Refer to the **Phase View Pane** section in the *UVM Interactive Simulation Debug* chapter for details.

Sequence View

Menu Bar: Tools -> UVM/OVM/VMM Debug -> Sequence View

This command opens the *Sequence View* window. Refer to the **Sequence View Pane** section in the *UVM Interactive Simulation Debug* chapter for details.

Register View

Menu Bar: Tools -> UVM/OVM/VMM Debug -> Register View

This command opens the *Register View* window. Refer to the **Register View Pane** section in the *UVM Interactive Simulation Debug* chapter for details.

Clock Analyzer

Extract Clock Information

Menu Bar: Tools -> Clock Analyzer -> Extract Clock Information

This command opens the *Clock Extraction Settings* form. Refer to the *Clock Extraction Settings Form* section in the *Clock Analyzer* chapter for details.

List Clock Domains

Menu Bar: Tools -> Clock Analyzer -> List Clock Domains

This command opens the *Clock Domains* window listing all current loaded or extracted clock domains. Refer to the *Clock Domains Window* section in the *Clock Analyzer* chapter for details.

List Specified Clock Domains

Menu Bar: Tools -> Clock Analyzer -> List Specified Clock Domains

This command opens the *Clock Tree Browser* window. Refer to the *Clock Tree Browser Window* section in the *Clock Analyzer* chapter for details.

Highlight Clock Domains

Menu Bar: Tools -> Clock Analyzer -> Highlight Clock Domains

This command opens the *Highlight Clock Domains* form. Refer to the *Highlight Clock Domains Form* section in the *Clock Analyzer* chapter for details.

Check Crossing Paths

Menu Bar: Tools -> Clock Analyzer -> Check Crossing Paths

This command opens the *Check Crossing Paths* form. Refer to the *Check Crossing Paths Form* section in the *Clock Analyzer* chapter for details.

List Crossing Paths

Menu Bar: Tools -> Clock Analyzer -> List Crossing Paths

This command opens the *Crossing Paths* window where the crossing path results are listed by displaying CDC correlations and register pair details. Refer to the *Crossing Paths Window* section in the *Clock Analyzer* chapter for details.

Switching Analysis

Switching analysis uses the design and FSDB file to determine which nets have the most transitions over a specified time range (switching activity) or which time has the most transitions (peak activity).

New Query

Menu Bar: Tools -> Switching Analysis -> New Query

This command invokes the *Switching Analysis* form where the parameters for switching analysis on the current design and FSDB file can be specified. Both the

design and the simulation results (FSDB file) must be loaded into the Verdi platform before the **New Query** command is enabled.

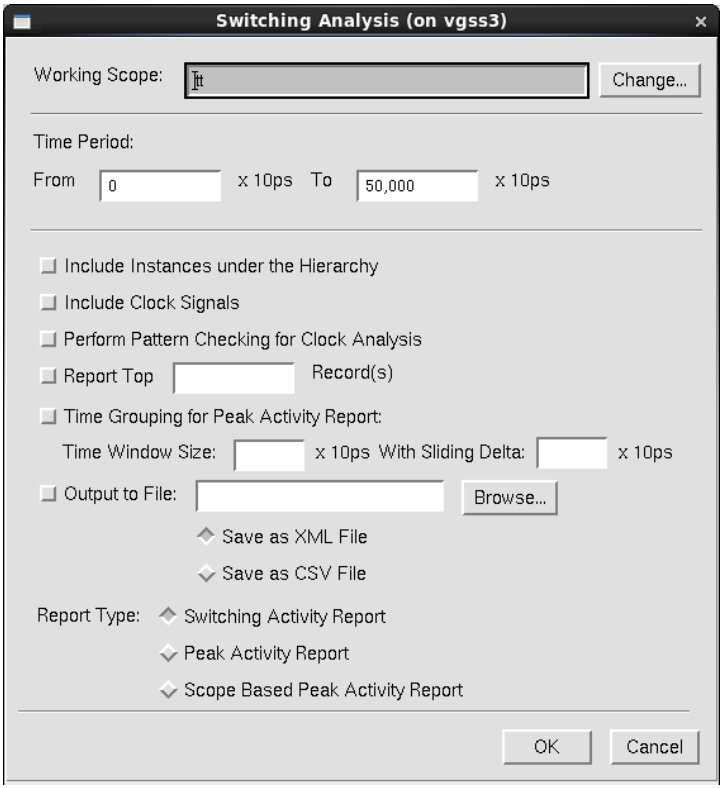


Figure: Switching Analysis Form

The *Switching Analysis* form includes the following options and fields:

- **Working Scope:** When a working scope is specified, the *Behavior Analysis* engine analyzes the design starting with this scope and everything below this scope. The first time the Verdi platform is invoked it has a different behavior than subsequent invocations in the same directory. On the first invocation of the Verdi platform, the working scope is blank.
- **Change:** To change the working scope, click the **Change** button to open the *Behavior Analysis* form. After the scope is changed, click **OK** to perform Behavior Analysis and update the working scope in the *Switching Analysis* form.
- **Time Period:** Specify the time range to use for switching analysis.

    - **From:** Specify the start time for analysis. The default is the simulation start time of the loaded FSDB file.
    - **To:** Specify the end time for analysis. The default is the simulation end time of the loaded FSDB file.
- **Include Instances under the Hierarchy:** When this option is turned *on*, instances under the current working scope is included in the analysis. When this option is turned *off*, only signals at the current working scope is analyzed. The default value of this option is *off*. The more instances that are included the longer the analysis takes.
- **Include Clock Signals:** When this option is turned *on*, clock signals under the current working scope is included in the analysis. When this option is turned *off*, only non-clock signals at the current working scope is analyzed. The default value of this option is *off*.
- **Perform Pattern Checking for Clock Analysis:** When this option is turned *on*, the waveforms is checked for patterns to identify clock signals under the current working scope. When this option is turned *off*, pattern checking is not used to identify clock signals. The default value of this option is *off*.
- **Report Top 'n' Record(s):** When this option is turned *on*, specify a positive value for the maximum number of results to report. When this option is turned *off*, up to 1000 results is reported. The default value of this option is *off*.
- **Time Grouping for Peak Activity Report:** When this option is turned *off*, peak switching activity is reported for individual time stamps. This is similar to a histogram. When this option is turned *on*, the peak switching activity is reported for a time window of a specific size with a sliding delta. This is similar to a histogram where there are classes of time stamps and the classes can be overlapped. The default value of this option is *off*.
    - **Time Window Size: 'n':** Specify the size of the time stamp for calculation of peak signal activity. The value for the time window can be between 2 and the value specified in the **Time Period To** field. The default time units match the units in the FSDB file.
    - **With Sliding Delta: 'n':** Specify the interval for the start of the next time stamp class. The sliding delta value must be between 1 and the value for the time window size. If a value is not specified, the default is *1*. The default time units match the units in the FSDB file.
      For example, if 10 is specified for the time window and 5 for the delta, overlapping classes for 0-10, 5-15, 10-20 is shown.
- **Output to File:** When this option is turned *on*, the report file name specified in the text field is used for saving switching analysis results. To

change the file name, type directly in the text field or click the **Browse** button to open the *File Manager* form where the directory structure can be viewed and a report file name specified. When this option is turned *off*, the results are not saved to a file. The default value of this option is *off*.

- **Save as XML File:** When this option is selected, save the report as a *.xml* file in the directory specified in the **Output to File** field.
- **Save as CSV File:** When this option is selected, save the report as a.*csv* file in the directory specified in the **Output to File** field

- **Report Type:** Two report options are available (only one option can be selected at a time):

NOTE: When the signal has more than one bit, the total transition count is accumulated through all its sub bits.

A glitch contributes two transition counts to both the switching and peak activity reports.

  - **Switching Activity Report:** Report the switching activity based on nets with the most transitions. Refer to the *Switching Activity Report* section for details.
  - **Peak Activity Report:** Report the switching activity based on the times with the most transitions. Refer to the *Peak Activity Report* section for details.
  - **Scope Based Peak Activity Report:** Report the switching activity based on the scope with the most transitions. Refer to the *Scope Based Peak Activity Report* section for details.
- **OK:** After the options are selected, click the **OK** button to close the window and complete the analysis process. After the design and FSDB file are analyzed, a new report frame based on the selected report type is displayed in the same frame location as the source code frame as a new tab.
- **Cancel:** This button closes the form without starting the analysis. The selected options is saved.

Manage Report

Menu Bar: Tools -> Switching Analysis -> Manage Report

This command opens the *Switching Analysis Report* window as a new tab in the same frame location as the source code frame. The form is empty until a switching activity or peak activity report file is loaded. Refer to the *Switching Activity Report* or *Peak Activity Report* sections for details.

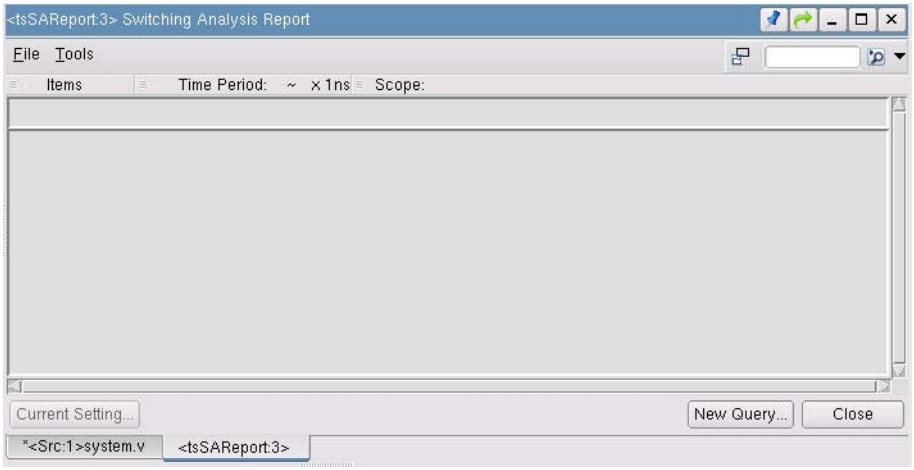


Figure: Switching Analysis Report Frame - No Reports Loaded

The **File** command has three commands and two sub-commands:

Load from XML File

Menu Bar: *Switching Analysis Report*, File -> Load from XML File

This command opens the *Load from XML File* form where a report file to load can be entered or the **Browse** button clicked to view the directory structure to locate and load a report file. After the file name is entered in the text field, click the **OK** button to close the form and load the results or the **Cancel** button to close the form without loading the results.

Save As -> XML File

Menu Bar: *Switching Analysis Report*, File -> Save As -> XML File

This command opens the *Save as XML File* form where the directory structure can be viewed to save the currently loaded results to a file in the XML file format.

Save As -> CSV File

Menu Bar: *Switching Analysis Report*, File -> Save As -> CSV File

This command opens the *Save as CSV File* form where the directory structure can be viewed to save the currently loaded results to a file in the CSV file format.

Close Window

Menu Bar: *Switching Analysis Report*, File -> Close Window

This command closes the current report frame. This is the same as clicking the **Close** button.

The **Tools** command has one command:

Customize Menu/Toolbar

Menu Bar: *Switching Analysis Report*, Tools -> Customize Menu/Toolbar

Refer to the **Tools -> Customize Menu/Toolbar** command in the main *nTrace* menu for details.

Current Setting: This button is enabled when a report file is loaded. This button opens a *Current Report Settings* information form displaying the settings used to generate the report. Click the **OK** button to close the window.

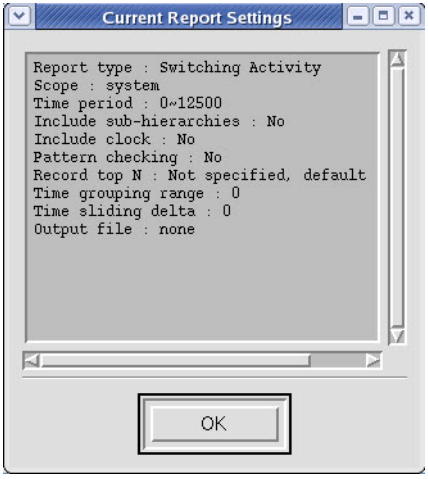


Figure: Current Report Settings Form

New Query: This button opens the *Switching Analysis* form (the same as the form opened by the **Exploration -> Switching Analysis -> New Query** command from the *nTrace* window). A new report frame is opened each time this button is clicked and a query is executed.

Close: This button closes the current report frame (the same as the window opened by the **File -> Close Window** command).

Switching Activity Report

The *Switching Analysis Report* window displays the switching activity based on signals with the most transitions. The results are viewed in a tabular format and can be sorted in a variety of ways.

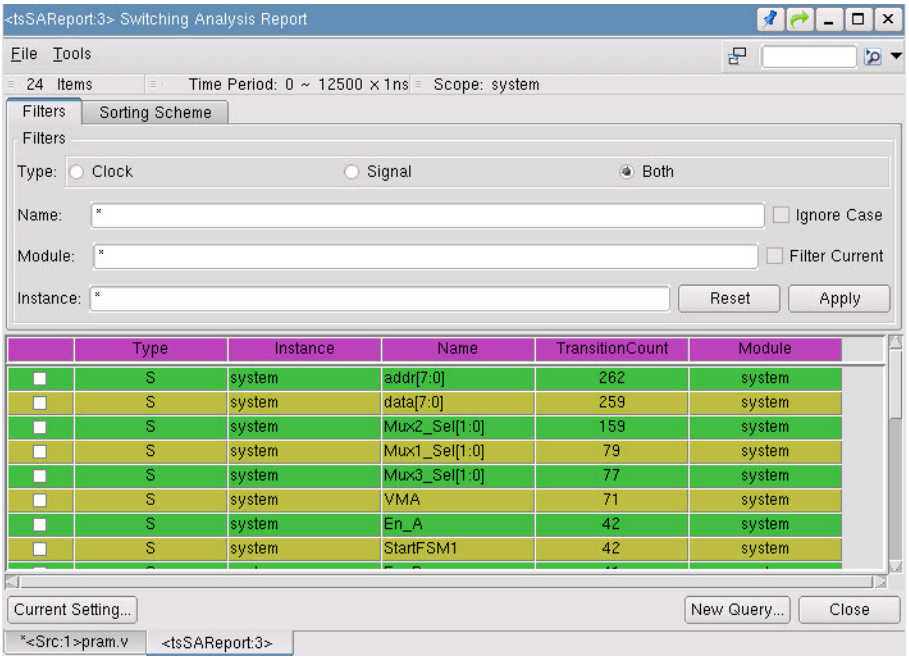


Figure: Switching Analysis Report Window - Switching Activity Report Loaded

The *Switching Analysis Report* window includes the **Filters** tab and the **Sorting Scheme** tab:

Filters Tab

The **Filters** tab can specify the filters for listing the results. Regular expressions using the asterisk (*) wildcard character are accepted in all fields.

Type: Filter the results list by the signal type.

- **Clock:** Display only clock signals (denoted with a C in the Type column).
- **Signal:** Display only non-clock signals (denoted with a S in the Type column).
- **Both:** Display all signal types.

Name: Filter the result list based on the signal name.

Module: Filter the result list based on the module name.

Instance: Filter the result list based on the instance name. The full hierarchical path can be searched.

Ignore Case: When this option is turned *on*, the case is ignored (that is, ‘a’ finds ‘a’ and ‘A’) in all the filter fields. When this option is turned *off*, the case is considered (that is, ‘b’ finds ‘b’ only) in all the filter fields. The default value of this option is *off*.

Filter Current: When this option is turned *on*, the filter is based on the current display list. When this option is turned *off*, the filter is based on the entire list of results. The default value of this option *off*.

Reset: Click this button to reset all the filter fields to list everything (all text fields contains an ‘*’). The displayed results is not updated until the **Apply** button is clicked.

Apply: Click this button to apply the current filter selections and update the results list display. The same effect is achieved by pressing **Enter** after changing a filter value.

Sorting Scheme Tab

The **Sorting Scheme** tab has fields to specify the criteria for sorting results.

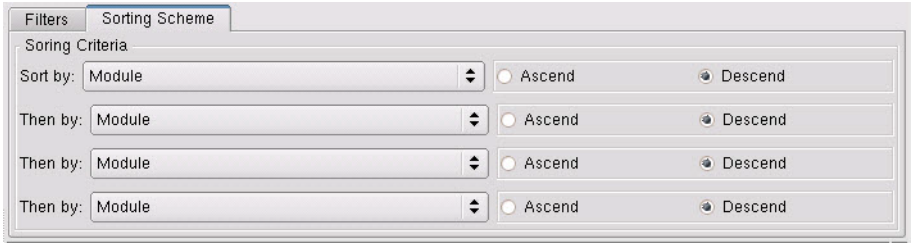


Figure: Sorting Criteria

Multiple selection fields are used to specify which fields (**Name**, **Instance**, **Type**, **TransitionCount**, **Module**) for sorting. The sorting order is determined by selecting either the **Ascend** or **Descend** options. Multiple sorts are supported by entering other fields in the **Then By** columns, with the selection of either the **Ascend** or **Descend** options.

Sorting can also be achieved with a left-click on the column header. Clicking on the column header of the table toggles the sorting order and promotes that heading as the first sorting criteria.

Main Report Display Area
Five columns are available in the report display. The width of each column can be adjusted by placing the cursor over the vertical bar in the heading row and then pressing and holding the left mouse button. The column headings are summarized as follows:

- **Type:** The type of signal: clock (C) or non-clock (S).
- **Instance:** The full hierarchical path to the instance.
- **Name:** The signal name.
- **TransitionCount:** The number of transitions on the signal during the specified time range. When the signal in question has more than one bit (for example, a bus), the total transition count is accumulated through all its sub bits. For any signal, a glitch counts as two transitions.
- **Module:** The module name.

One or more result rows can be selected with a left-click in the white box in the first column. After one or more rows are selected, use the middle mouse button to drag and drop to the *nWave* window to add to the waveforms. A single row can be dropped to the *nTrace* source code frame to trace the signal's connectivity.

Select one or more signals in the *nWave* or *nTrace* windows and use the middle mouse button to drag and drop to the *Switching Analysis Report* window. If the signals exist in the report, only those results is displayed.

In the main display area, the following commands are available in the right-click context menu:

- **Select All Rows:** Selects all rows currently being displayed.
- **Deselect All Rows:** Deselects all rows that are currently selected.
- **Remove Selections:** Removes the selected rows from the display.
- **Recover All Rows**: Displays all available result rows regardless of the filter selection.
- **Expand Selections:** Displays the individual bits of any selected rows that contain buses. After this command is invoked, all selected rows are deselected.
- **Collapse Selections:** Bundles individual bits back into buses of any selected rows that contain buses. After this command is invoked, all selected rows are deselected.
- **Toggle Expansion:** Toggles between an expanded or collapsed view for any selected rows that contain buses. After this command is invoked, all selected rows are deselected.

- **Column Arrangement:** Changes the display order of the columns. After invoking this command, click the column header to be moved and then the desired column to move it to. After this command is invoked, all selected rows are deselected.
- **Peak Time Activity:** The result form changes to display the results for peak activity.

Peak Activity Report

The *Switching Analysis Report* window is used to display the peak activity based on the times with the most transitions in a tabular format.

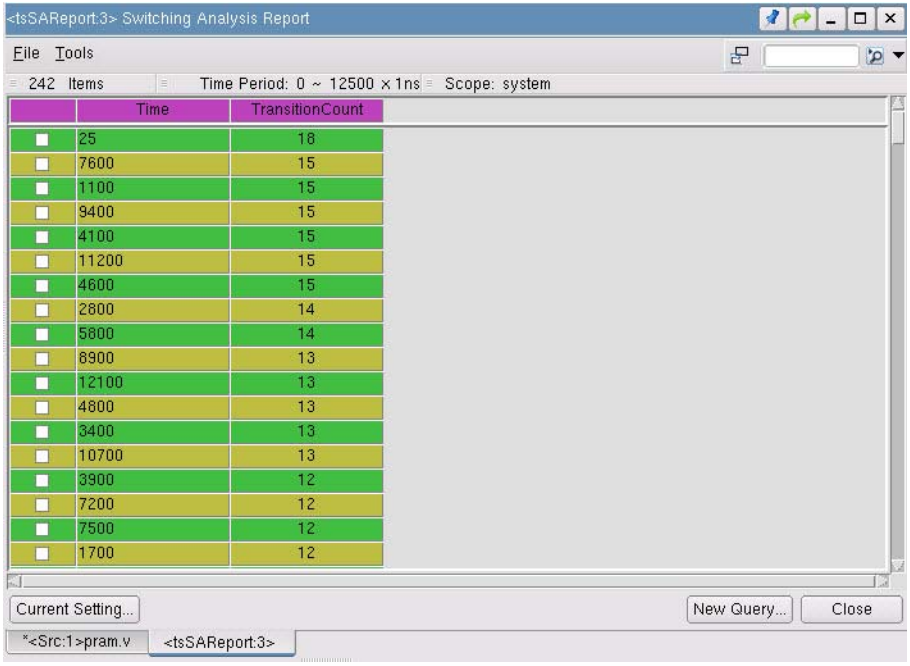


Figure: Switching Analysis Report Window - Peak Activity Report Loaded

The width of each column can be adjusted by placing the cursor over the vertical bar in the heading row and then pressing and holding the left mouse button. The column headings are summarized as follows:

- **Time:** The time where peak activity exists.
- **TransitionCount:** The number of transitions at a specific time.

One or more result rows can be selected with a left-click in the white box in the first column. After one or more rows are selected, use the right mouse button context menu to remove the selections from the display.

Select one or more signals in the *nWave* or *nTrace* windows and use the middle mouse button to drag and drop to the *Switching Analysis Report* window. The peak activity report automatically changes to a switching activity report and display the results for the signals.

In the main display area, the following commands are available in the right-click context menu:

- **Select All Rows:** Selects all rows currently being displayed.
- **Deselect All Rows:** De-selects all rows that are currently selected.
- **Remove Selections:** Removes the selected rows from the display.
- **Recover All Rows**: Displays all available result rows regardless of the filter selection.
- **Expand Selections:** Lists the individual signals that contributed to the peak activity time of any selected rows. After this command is executed, all selected rows are deselected.
- **Collapse Selections:** Removes the individual signals that contributed to the peak activity time from the display for any selected rows. After this command is executed, all selected rows are deselected.
- **Toggle Expansion:** Toggles between an expanded or collapsed view for any selected rows. After this command is executed, all selected rows are deselected.
- **Column Arrangement:** Change the display order of the columns. After invoking this command, click the column header to be moved and then the desired column to move it to. After this command is executed, all selected rows are deselected.
- **Switching Activity:** The result form changes to display the results for switching activity.

Scope Based Peak Activity Report

The *Switching Analysis Report* window is used to display the peak activity based on the scope with the most transitions in a tabular format.

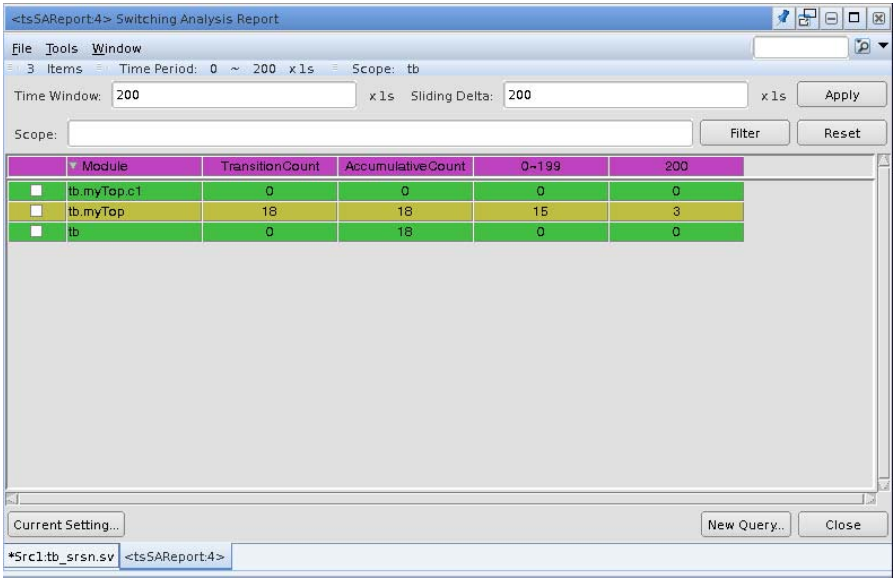


Figure: Switching Analysis Report Window - Scope Based Peak Activity Report Loaded

The width of each column can be adjusted by placing the cursor over the vertical bar in the heading row and then pressing and holding the left mouse button. The column headings are summarized as follows:

- **Time Window:** Specify the size of the time stamp for calculation of peak signal activity. The default time units match the units in the FSDB file.
- **Sliding Delta:** Specify the interval for the start of the next time stamp class. The sliding delta value must be between 1 and the value for the time window size. If a value is not specified, the default value of this option is *1*. The default time units match the units in the FSDB file.
  For example, if 10 is specified for the time window and 5 for the delta, overlapping classes for 0-10, 5-15, 10-20 is shown.
- **Scope:** Specify the search string for the scope. Only scopes that match the search string is displayed.
- **Filter:** Click this button to display the scopes that match the search string specified in the **Scope** filed.
- **Reset:** Click this button to reset the **Scope** filed. The displayed results is not updated until the **Apply** button is clicked.

- **Apply:** Click this button to apply the current filter selections and update the results list display. The same effect is achieved by pressing **Enter** after changing a filter value.
- **Module:** The names of all scopes of the selected module.
- **Transition Count:** The number of transitions on the signal of the scope during the specified time range. When the signal in question has more than one bit (for example, a bus), the total transition count is accumulated through all its sub bits. For any signal, a glitch counts as two transitions.
- **Accumulative Count:** The number of transitions on the signal accumulated from children scopes during the specified time range.
- **Time Widow Size:** The size of the time stamp for calculation of peak signal activity.
- **New Query:** Click this button to invoke the *Switching Analysis* form.
- **Close:** Click this button to close the current report. One or more result rows can be selected with a left-click in the white box in the first column. After one or more rows are selected, use the right mouse button context menu to remove the selections from the display.
- **Select All Rows:** Selects all rows currently being displayed.
- **Deselect All Rows:** Deselects all rows that are currently selected.
- **Remove Selections:** Removes the selected rows from the display.
- **Recover All Rows**: Displays all available result rows regardless of the filter selection.
- **Add Selections to Waveform:** Add the selected transition to a newly invoked waveform. In the waveform, a report ID is added to each hierarchy name of the signal for the user to identify which report the signal was added from.
- **Switching Activity:** The result form changes to display the results for switching activity.
- **Peak Time Activity:** The result form changes to display the results for peak time activity.

Invoke HW/SW Debug

Menu Bar: Tools -> Invoke HW/SW Debug

This command opens the *Debug Configurations* form where HW/SW debug settings can be configured.

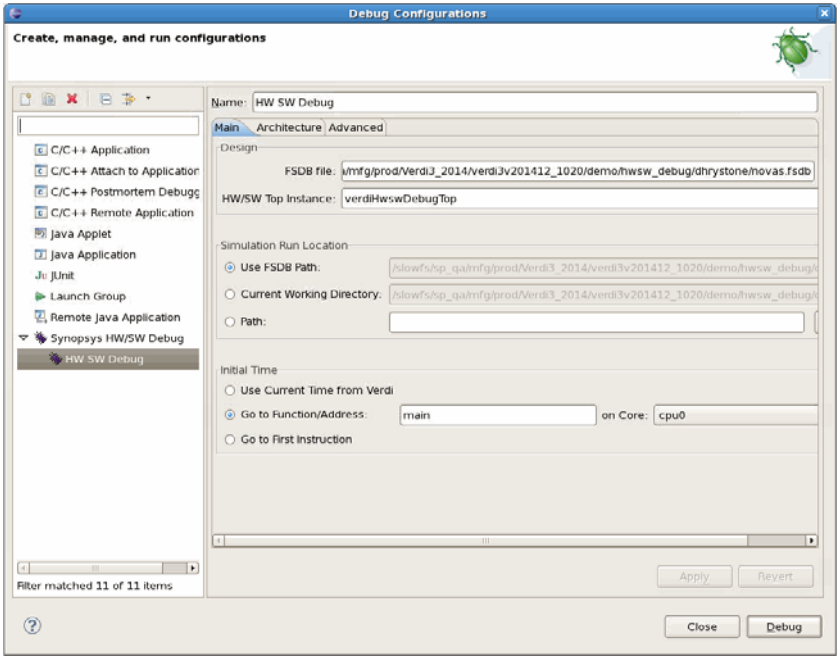


Figure: Debug Configurations Form

After specifying the configuration details, click **Debug** to close the form and launch HW/SW Debug.

Highlight

Menu Bar: Tools -> Highlight

This command opens the *Highlight* form which enables you to highlight signals with the specified color.

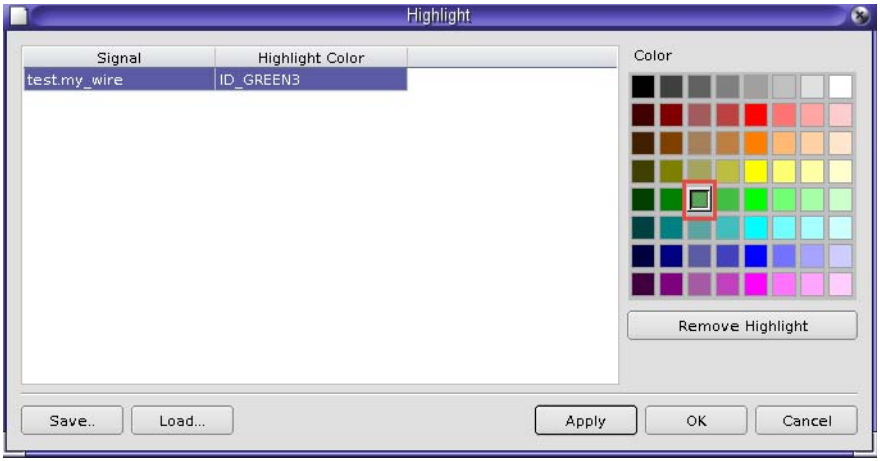


Figure: Highlight Form

The Highlight form includes the following options, fields, and buttons:

- **Signal:** Displays the complete hierarchy name of the selected signal.
- **Highlight Color:** Displays the highlighted color selected for the signal.
- **Color:** Displays the available colors that you can select to highlight the signal. Select the preferred color for the signal from the color palette.
- **Remove Highlight:** Click to remove the color from the highlighted signal. The **Highlight Color** is **N/A** for the signal for which the highlighted color is removed.
- **Save:** Saves the highlighted color settings to the file.
- **Load:** Loads the highlighted color settings from the file.

NOTE: This command is not supported for the following signals.

- Signals in testbench
- Power related signals.
- Node current or voltage in the format of vn(…) or in(….) in AMS debug, for example i1(mn1).
- Simulation signals, for example logical expression.
- Property signals, for example assert and cover signals.

Preferences

Menu Bar: Tools -> Preferences

The preference settings are saved in the *novas.rc* resource file and is loaded each time the Verdi platform is invoked. Refer to the *General Folder* section of the *Preferences* chapter for details.

Customize Menu/Toolbar

Menu Bar: Tools -> Customize Menu/Toolbar

The **Tools -> Customize Menu/Toolbar** command opens the *Customize Menu/Toolbar* form in which custom commands can be created, bind keys for menu commands can be modified, and toolbar categories can be added, removed, or changed.

After making the desired modifications, click **OK** to save the changes and close the form. The changes are saved to the *novas.conf* file and the *novas.rc* resource file. The *novas.rc* resource file takes precedence when the Verdi platform is invoked. New toolbar categories automatically is added below the existing toolbar icons. Click **Cancel** to close the form without making any changes.

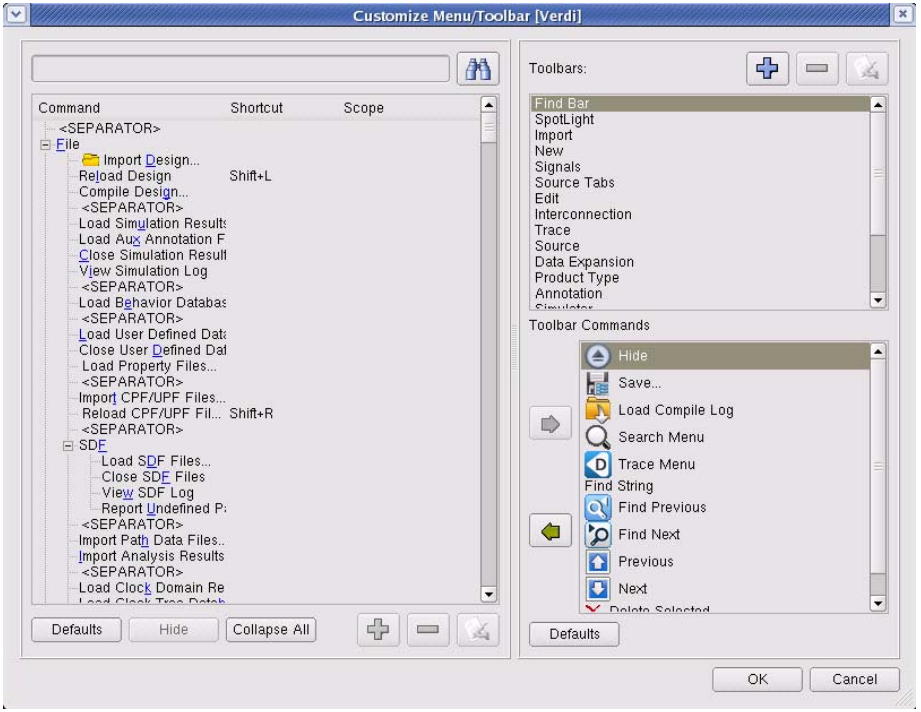


Figure: Customize Menu/Toolbar Form

On the left pane of the form, all menu command types (drop-down and right-click) and non-menu command types (invoked with mouse clicks or shortcuts) are hierarchically listed. In each row, the command name, mnemonics key (highlighted in blue), bind (shortcut) key, and scope are displayed.

The bind key for a command (for example, Reload Design) but not a command group (for example, File) may be modified by double-clicking the associated **Shortcut** cell to enter edit mode. Then press the desired bind key combination. Alphanumeric and normal modifier keys (**Ctrl**, **Alt**, **Shift**) are supported. For example, to specify **Ctrl+C**, press the **Ctrl** key followed by the **C** key, and keep both pressed. If the bind key is already used, the cell is highlighted in red. Press the **Backspace** key to enter a different bind key value or press the **Enter** key to display an *Error* message. Click **Yes** to delete the original assignment or click **No** to keep the original assignment.

The *Customize Menu/Toolbar* form includes the following fields and buttons on the left pane of the form:

- **Search** text field and button 

Enter the string to be searched for in the text field to dynamically search for the first command that matches. Then press **Enter** or **Search** to recursively locate the next command matching the search string. A partial string is valid and searching is not case-sensitive.

- **Defaults**

  Change only the settings of all commands of the selected menu back to their default values. Settings of other menus are not changed.

- **Hide/Show**

  Hide the selected command. The command is disabled and the button changes to **Show**. Click **Show** button to make the selected command visible again.

- **Collapse All**

  Collapse the command tree to the top level node.

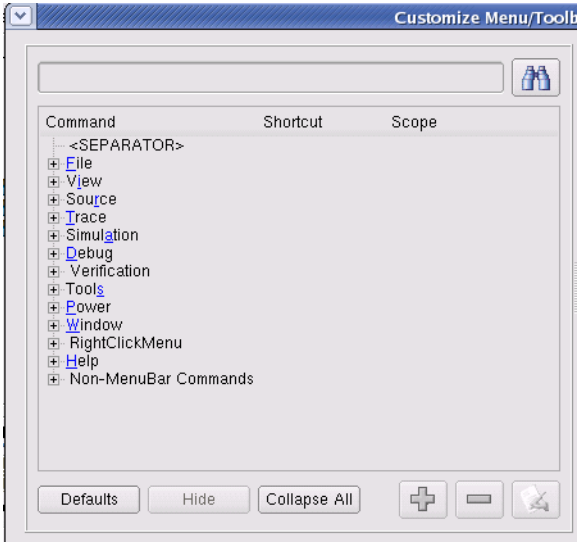


Figure: Collapsed Command Tree

- **Add Custom Command** 

  Add a custom command below the current selection. This button opens the *Custom Command* form for defining the new command, sub-menu, or separator. Refer to the *Custom Command Form* section for details.

- **Delete Custom Command** 

  Delete the selected custom command. This button is available for custom commands and is not activated on tool-provided commands.

- **Edit Custom Command** 

  Edit the selected custom command. This button opens the *Custom Command* form for editing. Refer to the *Custom Command Form* section for details. This button is available for custom commands and is not activated on tool-provided commands.

The right pane of the *Customize Menu/Toolbar* form is for modifying toolbar categories. All available toolbar categories of the sub-window/window are listed in the **Toolbars** field with the commands belonging to the selected toolbar category listed in the **Toolbar Commands** field.

The following buttons are available on the right pane of the form:

- **Add Custom Toolbar** 

  Add a custom toolbar to the end of the list. The default name is ‘new toolbar’ and it can be immediately modified by typing the preferred name.

- **Remove Custom Toolbar** 

  Delete the selected custom toolbar. This button is available for custom toolbars and is not activated on tool-provided toolbars.

- **Rename Custom Toolbar** 

  Edit the name of the selected custom toolbar. This button is available for custom toolbars and is not activated on tool-provided toolbars.

- **Add Selected Command To Toolbar** 

  Add the selected command from the left pane to the current toolbar category in the **Toolbar Commands** field. The command is added above the currently selected toolbar command.

- **Remove Selected Command From Toolbar** 

  Remove the selected command from the **Toolbar Commands** field in the current toolbar category.

- **Defaults**

  Reset the toolbar category settings back to their defaults. All new categories is removed and any additions/subtractions to/from tool-provided categories is reset.

Custom Command Form

The *Custom Command* form is used to add user-specified commands, sub-menus, and separators.

After making the required modifications, click **OK** to save the changes and close the form. Click **Cancel** to close the form without making any changes.

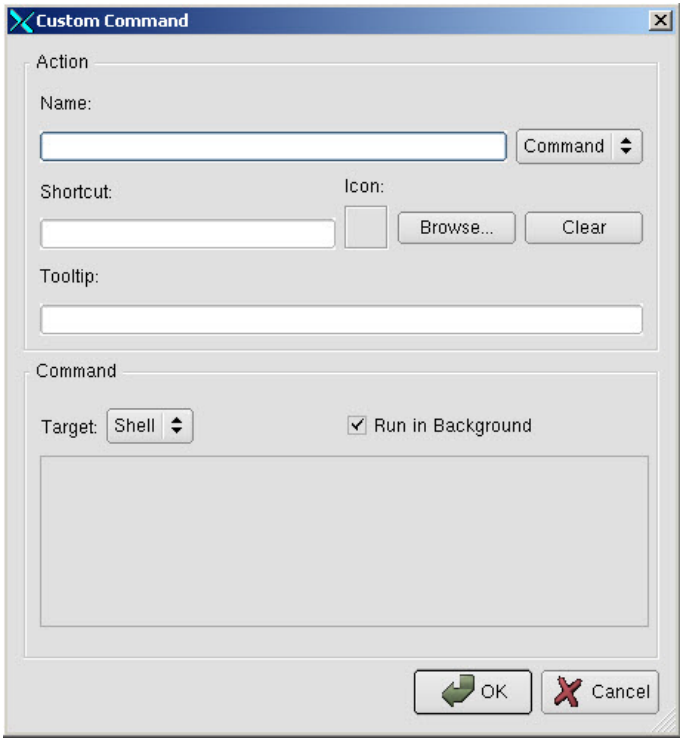


Figure: Custom Command Form

The *Custom Command* form includes the following fields and buttons in the **Action** section:

- **Name**: Specify the name for the command, sub-menu, or separator. You can also specify the command's shortcut key by adding '&' in front of the character. For example, enter **My&Command** in the **Name** field. The **MyCommand** is added with the shortcut key as **C**.The name type is determined by the option selection menu (containing the **Command**, **Submenu**, and **Separator** options) to the right of the field.

NOTE: After new sub-menu and sub-command has been created, an existing command may be dragged to the default sub-command which then takes the properties of the dragged command.

- **Shortcut**: Specify the shortcut keys.

- **Icon**: Specify the icon file. Click **Browse** to display the *Open File* form and select the required icon from the specified directory. Click **Clear** to remove the icon.
- **Tooltip**: Specify the text for the tooltip.

The following fields are available in the **Command** section:

- **Target**: Select the target script type from the selection menu. The options are **Shell** or **Tcl** and the default is *Shell*. When **Shell** is selected, enter the path to the shell script in the text field. When **Tcl** is selected, enter one or more Tcl commands to be executed in the text field.
- **Run in Background**: When this option is *on*, the command automatically runs in the background. When this option is *off*, the command is run in the foreground. The default value of this option is *on*.

Emulation

Summary View

Menu Bar: Tools -> Emulation -> Summary View

Toolbar Icon: 

Refer to the *Summary View* chapter for details.

Run Simulation

Menu Bar: Tools -> Run Simulation

Refer to the **Run/Continue** command description in the *Interactive Simulation Debug* chapter for details.

Attach to Simulation

Menu Bar: Tools -> Attach to Simulation

NOTE: The required settings for this feature are as follows:
1. The simulation must start with the **-ucli2Proc** VCS runtime option.
2. The **-debug_access/-debug_pp/-debug/-debug_all** VCS elaboration options must be added to enable the VCS debug features.
3. The design must be loaded in Verdi before attaching a simulation.

NOTE: If the simulation is running in GUI mode (DVE or Verdi) or specman, the process cannot be attached.

NOTE: The **Restart** and **Restore Session** commands are not supported in Verdi after a simulation is attached.

This command opens the *Attach to Simulation* form where all running simulation processes are listed. After a simulation process is selected (only one simulation process can be attached at one time), click **Attach** and then in the UCLI terminal, press **Enter** on the keyboard to accept the attached simulation.

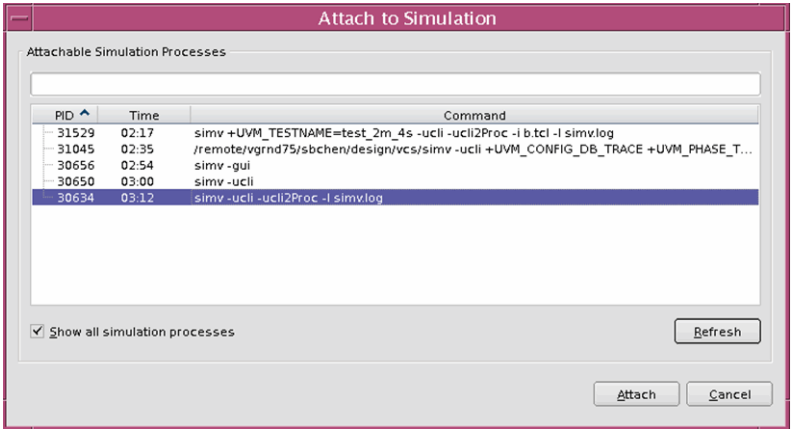


Figure: Attach to Simulation Form

The *Attach to Simulation* form includes the following field and options:

- **Attachable Simulation Process** text field: Enter the text in the text field to filter simulation process. The wildcard characters are supported.
- **Show all simulation processes:** When this option is turned *on*, all running processes are displayed. The default value of this option is *off*.
- **Refresh:** Click this button to refresh processes in the current machine.

Detach Simulation

Menu Bar: Tools -> Detach Simulation

This command opens the *Attach to Simulation* form where the attached process is marked with the icon. Select the connected process and click the **Detach** button to detach the simulation.

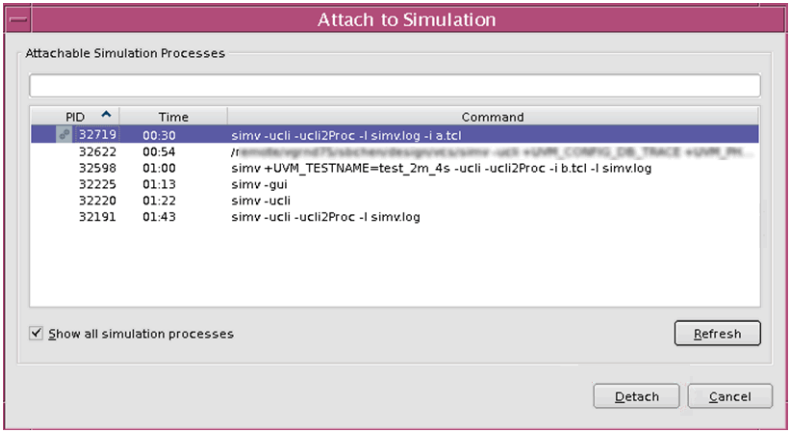


Figure: Attach to Simulation Form for Detaching

Stop Interactive Debug

Menu Bar: Tools -> Stop Interactive Debug

Refer to the **Stop** command description in the *Interactive Simulation Debug* chapter for details.

Power Commands

Refer to the *Power Options* section in the *Power Aware Debug* chapter for details.

Window Commands

Hardware Debug Mode

Menu Bar: Window -> Hardware Debug Mode

This command opens the relevant frames for hardware debug. The supported frames include:

- Instance tree frame (**Instance** tab of the design browser frame)
- Declaration tree frame (**Declaration** tab of the design browser frame)
- Source code frame
- *Message* frame
- *nWave* window
- *nSchema* window

The supported frames are displayed in the Message Bar.

Testbench Debug Mode

Menu Bar: Window -> Testbench Debug Mode

This command opens the relevant frames for testbench debug. The supported frames include:

- *Constraint* frame
- *Inheritance* frame
- *FSDB_Msg* frame
- Instance tree frame (**Instance** tab of the design browser frame)
- Declaration tree frame (**Declaration** tab of the design browser frame)
- Source code frame
- *Message* frame
- *nWave* window
- *nSchema* window

The supported frames are displayed in the Message Bar.

Power Debug Mode

Menu Bar: Window -> Power Debug Mode

This command opens the relevant frames for power debug. The supported frames include:

- *Flatten Power Domain/Hierarchical Power Domain* frame (**Flatten Power Domain/Hierarchical Power Domain** tab of the design browser frame)
- UPF/CPF command source code frame
- Instance tree frame (**Instance** tab of the design browser frame)
- Declaration tree frame (**Declaration** tab of the design browser frame)
- HDL design source code frame
- *Message* frame
- *nWave* window
- *nSchema* window

The supported frames are displayed in the Message Bar.

Assertion Debug Mode

Menu Bar: Window -> Assertion Debug Mode

This command opens the relevant frames for assertion debug. The supported frames include:

- *Statistics* frame
- *Analyzer* frame
- Instance tree frame (**Instance** tab of the design browser frame)
- Declaration tree frame (**Declaration** tab of the design browser frame)
- Source code frame
- *Message* frame
- *nWave* window
- *nSchema* window

The supported frames are shown in the Message Bar.

Transaction Debug Mode

Menu Bar: Window -> Transaction Debug Mode

This command opens the relevant frames for transaction debug. The supported frames include:

- *Transaction and Protocol Analyzer* frame (tProtocolAnalyzer)
- *Transaction Table View* frame (tTableView)
- *Transaction Relation Navigator* frame (tRelation)

The supported frames are displayed in the Message Bar.

Interactive Debug Mode

Menu Bar: Window -> Interactive Debug Mode

This command opens the relevant frames for Interactive Simulation Debug. The supported frames include:

- Instance tree frame (**Instance** tab of the design browser frame)
- Declaration tree frame (**Declaration** tab of the design browser frame)
- *Class* frame (default location is a tab of the same name in the design browser frame)
- *Object* frame (default location is a tab of the same name in the design browser frame)
- *Stack* frame (default location is a tab of the same name in the design browser frame)
- Source code frame
- *Local* frame
- *Member* frame
- *Interactive Console* frame
- *Message* frame
- *Watch* frame

The supported frames are displayed in the Message Bar.

Preserve Visible Windows/Widgets

Menu Bar: Window -> Preserve Visible Windows/Widgets

This toggle command turns the display of current windows/widgets *on* and *off*. When this command is turned *on*, current frames is kept when changing to different work modes. When this command is turned *off*, default frames for the selected work mode is displayed. The default value of this option is *off*.

Previous Layout

Menu Bar: Window -> Previous Layout

Bind Key: Ctrl+Alt+R

This command returns the main window to the previous layout. The 30 previous layouts are saved. The layout can be restored for any of the following actions:

- Dock/undock a frame using the toolbar icon
- Change a docked frame to a top-level window or vice versa
- Move a docked frame to another dock location
- Resize the top window or any inner frames
- Hide a frame
- Change layout modes
- Maximize or restore a frame

NOTE: If the layout is manually changed after executing either of the **Previous**/**Next Layout** commands, the layout history is updated. For example, assume *Layout_1* to *Layout_5* is available and the following steps are performed:
1. The current state is *Layout_5*.
2. The **Previous Layout** command is selected 4 times to return to *Layout_1*.
3. The **Next Layout** command is selected 2 times to return to *Layout_3*
4. The layout is manually changed. The layout is saved to *Layout_4* and *Layout_5* is cleaned up. *Layout_1*, *Layout_2*, and *Layout_3* are still available.
5. The **Previous Layout** command is selected 2 times to return to *Layout_2*.
6. The layout is manually changed. The layout is saved to *Layout_3* and *Layout_4* is cleaned up. *Layout_1* and *Layout_2* are still available.

Next Layout

Menu Bar: Window -> Next Layout

Bind Key: Ctrl+Alt+E

This command returns the main window to the next layout. The 30 previous layouts are saved.

Save/Restore User Layout

Menu Bar: Window -> Save/Restore User Layout

This command opens the *Save/Restore User Layout* form where the current configuration can be saved or a previously saved configuration can be restored. The layouts are saved to the *novas.conf* file.

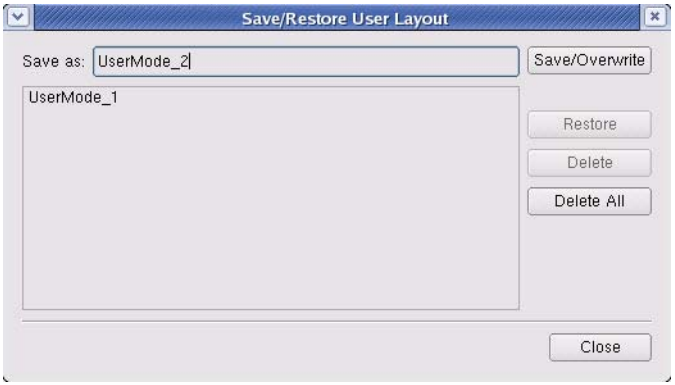


Figure: Save/Restore User Mode Form

The *Save/Restore User Mode* form includes the following fields and options:

- **Save as:** In this text field, specify the name of the user mode. The default name is UserMode_# where # is the next consecutive number.
- **Save/Overwrite:** Click this button to add the specified mode to the list or overwrite the selected mode in the list.
- **Restore:** Click this button to apply the configuration.
- **Delete:** Click this button to delete the selected mode from the list.
- **Delete All:** Click this button to remove all modes.
- **Close:** Click this button to close the form.

Switch Full Screen

Menu Bar: Window -> Switch Full Screen

Bind Key: Alt+Enter

This command expands the main window to its maximum size and hides all menus and dock banners by toggling on the **Hide Menus** icon and **Hide Banners** icons in the message bar. You can also click the **Switch Full Screen** icon in the message bar of the main window to switch back to the full screen mode.

Run the command again to switch back to the normal window size mode, and **Hide Menus** and **Hide Banners** icons also appear as the same status before the switch full screen mode.

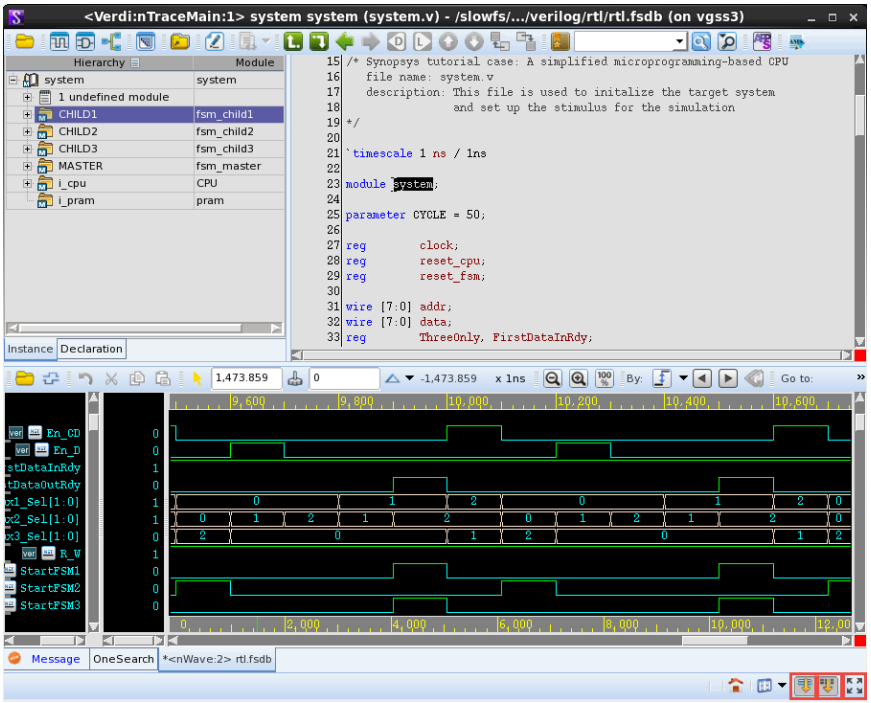


Figure: Example of Expanded Main Window

Click **Alt** key to quickly popup the mouse hovered window's menu. When you click the **Switch Full Screen** icon again, then the window switches back to the normal window size.

Refer to the *Welcome Page* section in the *Introduction* chapter for more details regarding the icons.

Window Manager

Menu Bar: Window -> Window Manager

This command opens the *Window Manager* form where current or previously opened frames from this session are displayed.

In the **Window Name** column, the available frames are listed. In addition, the current content displayed in the frame (where applicable) is included. The **Displayed** column indicates whether the frame is currently displayed (**Yes**) or was displayed previously (**No**).

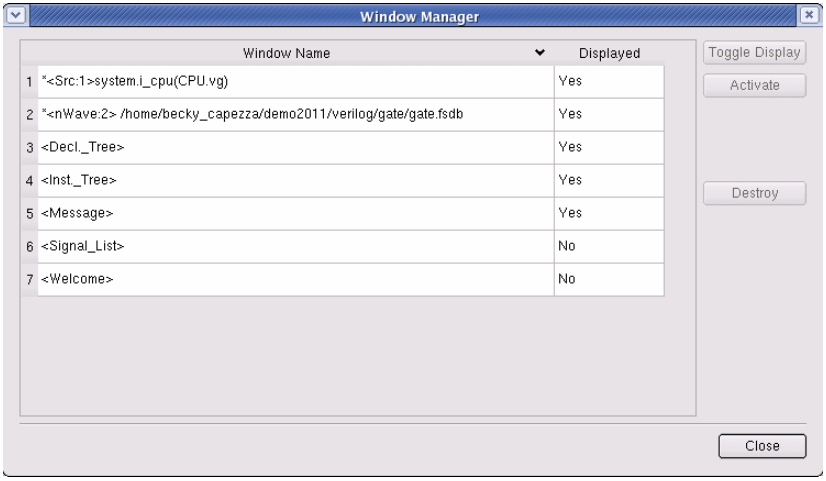


Figure: Window Manager Form

The *Window Manager* form includes the following options:

- **Toggle Display:** This button toggles the display for the selected row. Alternatively, right-click the main menu bar to display the list of available frames and check to show or uncheck to hide.
- **Activate:** This button makes the selected frame the active frame.
- **Destroy:** This button completely removes the selected frame and its contents from the main framework.

Help

Welcome

This command displays the **Welcome** page as a new tab on top of the source code/schematic frame. This is equivalent to clicking the **Welcome** icon in the message bar of the main window. Refer to the *Welcome Page* section in the *Introduction* chapter for details.

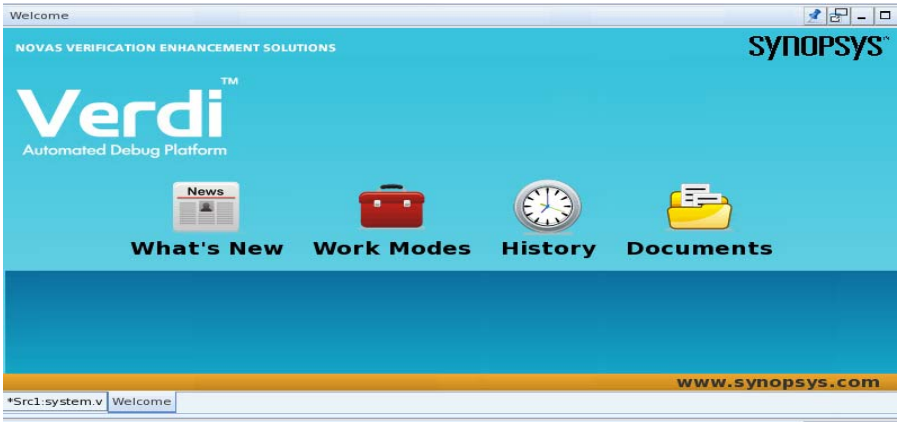


Figure: Welcome Tab

Bookshelf

This menu selection brings up the *Verdi Bookshelf*, listing all other documents for the Verdi and Siloti systems and associated modules, in PDF format.

Legend

This menu selection opens the *Legend* form. The form displays the icons available in the *Design Browser* frame on the **Design** tab, the *nWave* window on the **nWave** tab, the *Temporal Flow View* window on the **TFV** tab, and the *Power Map* window on the **Power Manager** tab.

About FSDB/About Verdi

These commands display the version information for the FSDB file and the GUI respectively.

Message Frame

The *Message* frame is located at the bottom of the main framework. The *Message* frame includes the **General**, **Compile**, **OneTrace**, **Search**, **Interconnection** and **X-propagation** tabs. For messages that reference design files, design objects, or power objects, the text in the *Message* frame can be double-clicked to find the source code associated with the result.

The **General** tab reports information when the following commands are executed.

- **File -> Import Design**
- **File -> Compile Design**

The **Compile** tab reports information when there are compilation errors. When there are more than 100 compile errors and warnings, the first 99 errors are listed. Double-click **More** at the end of the list to see all of the results. Warning messages can be filtered from the list by turning *on* the **Filter Compile Warning(s)** option on the **Import Design** page of the **Source Code** folder in the *Preferences* form (invoked with the **Tools -> Preferences** command).

The **OneTrace** tab reports information including a sequence order, a trace prefix, and summary line when the following commands are executed:

- **OneTrace -> Driver**
- **OneTrace -> Load**
- **OneTrace -> Fan-in**
- **New Schematic-> Fan-in**
- **OneTrace -> Connectivity**
- **OneTrace -> Mapped Signal**

You can trace drivers and loads of a signal at any time to view the drivers and loads that caused a value change and also view all the drivers and loads that possibly contributed to a signal value.

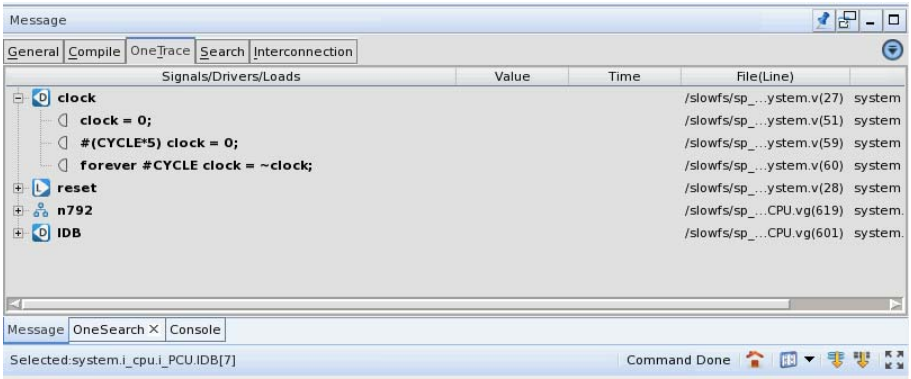


Figure: Trace Results

Following fields are available on the *OneTrace* tab:

- **Signals/Drivers/Loads**: Provides the short name of the traced signal or contributor and the driver/load design statement.
- **Value**: Provides the value of signal from the FSDB file and the **Time** field.
- **Time**: Provides the FSDB cursor time.
- **File(Line)**: Provides the position of the signal.
- **Scope**: Provides the complete hierarchical scope.

NOTE: Verdi issues an error message in the *General* tab if the traced results do not appear when tracing.

Double-click the signal to view the code in the *Source Code* pane, as illustrated in the following figure:

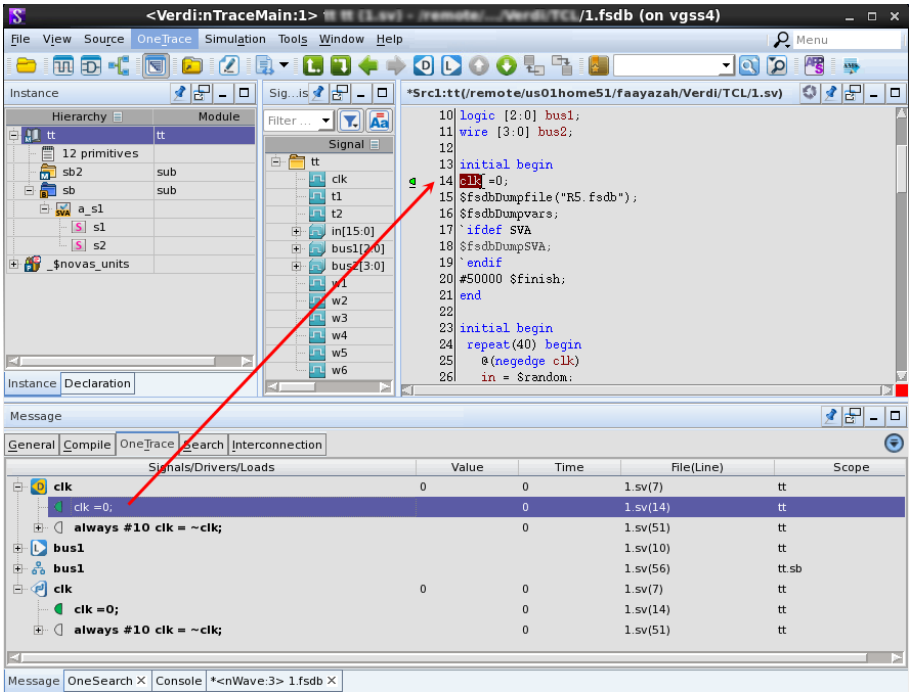


Figure: Source Code Pane

If you hover the mouse pointer on an icon in the *Message* pane, the tooltip appears providing information of the traced signal, as illustrated in the following figure:

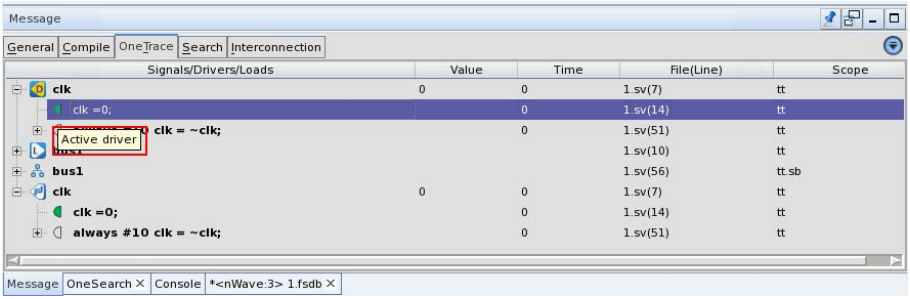


Figure: Tooltip Information

When you trace or continue to trace a signal, Verdi displays a statement result in the source window automatically. You can also double-click in the **OneTrace** tab to display the statement results scope.

When you display the statement results scope, Verdi displays and highlights the statement in the *nTrace* source window. However, the *Signal List* pane is not updated.

Points to Note:

- UCLI driver command may be different with Verdi, this is because the data source is different. Verdi uses static data, but UCLI uses dynamic data.
- The limitations for **Go To Previous Value Change** option (GTPVC) are as follows:
  - The reverse/interactive mode cannot function accurately (for example, undo/redo, jump to correct source location, and so on) in VHDL scope. Hence, for mixed design, some hardware/testbench targets may become untraceable.
  - Support for partial bit tracing is not available. If partial bit tracing is specified, then the complete tracing is specified.
  - Local signals/variables of the function/task cannot be traced.
- For the testbench signals/variables, if the signal is not dumped and if the previous values are not available then the signals cannot be analyzed by the current value from simulators.
- For VHDL scope, Verdi does not provide support for expanded contributors.

The **Search** tab reports information when the following command is executed:

- **Source -> Find String**

The *OneTrace* tab provides information that includes signal name, value, time, file (line), and scope fields of the traced signal. The following figure illustrates an example of the trace results that appears in the *OneTrace* tab of the *Message* pane:

The **Interconnection** tab reports information when the following commands are executed.

- **Trace -> Interface Mapping**
- **Trace -> From HDL to HVL Interface**
- **Show Reference** (source code frame right-click menu option)
- **Show File Information** (right-click menu option of the **Instance** and **Declaration** tabs in the design browser frame)
- **Show Instantiation** (right-click menu option of the **Declaration** tab in the design browser frame)

The **X-propagation** tab displays the X-propagation log file read by the Verdi platform with the **-xproplog** command line option. Depending on the size of the log file, it may take some time for the log file to be displayed. After the log file is displayed, click a message to jump to the corresponding file and line in the *Source Code* frame.

NOTE: The **X-propagation** tab is available when the X-propagation design and the log file are already loaded with the **-xprop** and **-xproplog** options in the *verdi* utility.

Click the **Show Toolbar**  toolbar icon to display the **Find String** text field and the **Find Previous/Find Next** icons to locate file instrument information.

Message Frame OneTrace Prefix Legend

The following is a list of prefixes and identifiers that are added to the results in the **OneTrace** tab of the *Message* frame. Multiple prefixes may be combined.

Following is a list of prefixes and identifiers that appear with the results in the One*Trace* tab of the *Message* pane:

Identifier	Description
	Signifies the driver traced result.
	Signifies the active driver traced result.
	Signifies the Driver and Load results with a single command.
	Signifies the trace value change traced result.
	Signifies the load traced result.

Following is a list of prefixes and identifiers that appear with the results in the *OneTrace* tab of the *Message* pane for the contributing signals:

Identifier	Description
	Signifies the contributor is completely analyzed and detected as an active driving signal.
	Signifies the contributor time is accurate but it is unable to determine if the signal caused a value change of the traced signal.
	Signifies the traced signal is not able to analyze the contributor.

Following is a list of prefixes and identifiers that appear with the results in the *OneTrace* tab of the *Message* pane for the active analyzed driver traced results:

Identifier	Description
	Signifies the active analyzed traced result.
	Signifies the possible analyzed traced result.
	Signifies the PLI_UCLI driver traced result.
	Signifies the X-Merge/T-Merge driver traced result.
	Signifies the inactive driver traced result.

Following is a list of prefixes and identifiers that hat appear with the results in the *OneTrace* tab of the *Message* pane for the static/inactive driver, load, and connectivity traced results:

Identifier	Description
	Signifies the driver traced results.
	Signifies the load traced results.
	Signifies the connectivity traced results.

Following is a list of prefixes and identifiers that appear with the results in the *OneTrace* tab of the *Message* pane for the pass-through static/inactive driver, load, and connectivity traced results:

Identifier	Description
	Signifies the pass-through driver traced results.
	Signifies the pass-through load traced results.

Identifier	Description
	Signifies the pass-through connectivity traced results.

Message Frame Right-Click Options

Following is the list of right-click menu options available for the **General**, **Compile**, **OneTrace**, **Search,** and **Interconnection** tabs of the *Message* frame.

Show Toolbar

This command displays the *Message* frame toolbar.

Trace Value Change

This command backtraces the selected signal's value at the current cursor time and displays the last driver encountered results in the source view and the trace view.

Trace Driver

This command traces all the possible drivers of the selected signal.

Trace Load

This command traces all the possible loads of the selected signal.

Show Bit Info

Displays the bit driver information for a single-dimension packed/unpacked traced signal, as illustrated in the following figure:

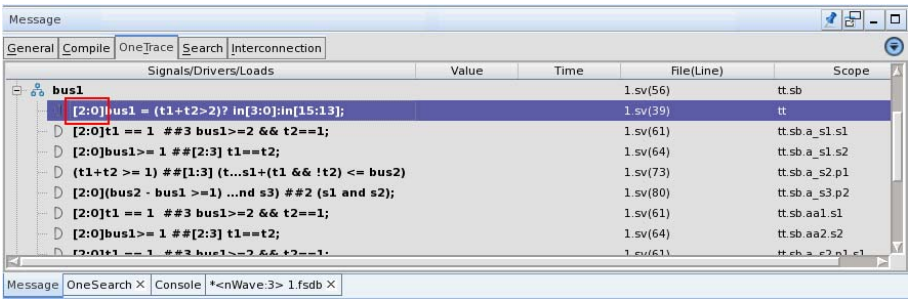


Figure: Show Bit Information

Highlight

Opens the *Highlight* form that enables you to highlight signals with the specified color, as illustrated in the following figure:

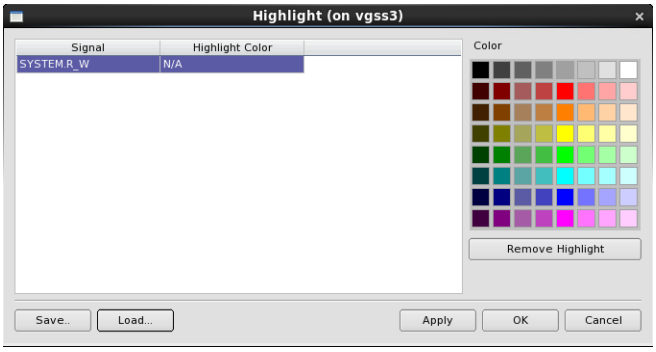


Figure: Highlight Form

The following options, fields, and buttons are available on the *Highlight* form:

- **Signal:** Displays the complete hierarchy name of the selected signal.
- **Highlight Color:** Displays the highlighted color selected for the signal.
- **Color:** Displays the available colors that you can select to highlight the signal. Select the preferred color for the signal from the color palette.
- **Remove Highlight:** Click to remove the color from the highlighted signal. The **Highlight Color** is N/A for the signal for which the highlighted color is removed.
- **Save:** Saves the highlighted color settings to the file.

- **Load:** Loads the highlighted color settings from the file.

NOTE: This command is not supported for the following signals.

- Signals in testbench.
- Power related signals.
- Node current or voltage in the format of vn(…) or in(….) in AMS debug, for example i1(mn1).
- Simulation signals, for example, logical expression.
- Property signals, for example, assert and cover signals.

Delete All

This command removes all results from the tab.

Delete Selected

This command removes the selected result and its underlying nodes from the tab. Multiple results may be selected for deletion.

After a line is selected, click the line to activate the current line so the text of interest can be selected by dragging the mouse. To select multiple lines, select a line, press and hold the **Shift** key and drag it to the left to select continuous multiple lines; press and hold the **Ctrl** key and drag it to the left to select one-item-by-one-item for multiple lines.

The selected text can then be added (copied) into the clipboard where it can be pasted into other applications or the **Delete Selected** command can be invoked to remove the selection from the display.

Save

This command opens a *Save Message* form where the results in the current tab can be saved to the specified text file.

Expand All

This command expands all results on the selected tab. This command is not available on the **General** tab.

Collapse All

This command collapses all results on the selected tab. This command is not available on the **General** tab

Show Both Active and Inactive Drivers

Displays both active and inactive driver trace result in *nWave* and *nTrace* windows. By default, the option is *on*. You can also select the **Show Both Active and Inactive Drivers** preference option in the **Trace** page (**Trace -> Active Driver -> Show Both Active and Inactive Drivers**) of the *Preferences* form.

Show Intermediate Drivers

Displays intermediate drivers between a signal and the last driver encountered. By default, this option is *off*. You can also invoke this option in the S **Trace** page (**Trace -> Active Driver -> Show Intermediate Drivers Found when Tracing Value Change**) of the *Preferences* form.

Show Passthroughs

Displays the passthroughs by selecting from one of the All or By Port Direction settings. You can also enable this option in the **Trace** page (**Trace -> General -> Show Passthroughs**) of the *Preferences* form.

Show Duplicate Results

Displays the duplicate trace results. That is, the results that point to the same source line. You can also enable this option in the **Trace** page (**Trace -> General -> Show Duplicate Results**) of the *Preferences* form.

Show Top Level Ports

Displays the top level ports of the trace results. You can also enable this option in the **Trace** page (**Trace -> General -> Show Top Level Ports**) of the *Preferences* form.

Show Columns

Displays the columns that are selected in the *OneTrace* tab, as illustrated in the following figure. Uncheck the columns to hide the columns.

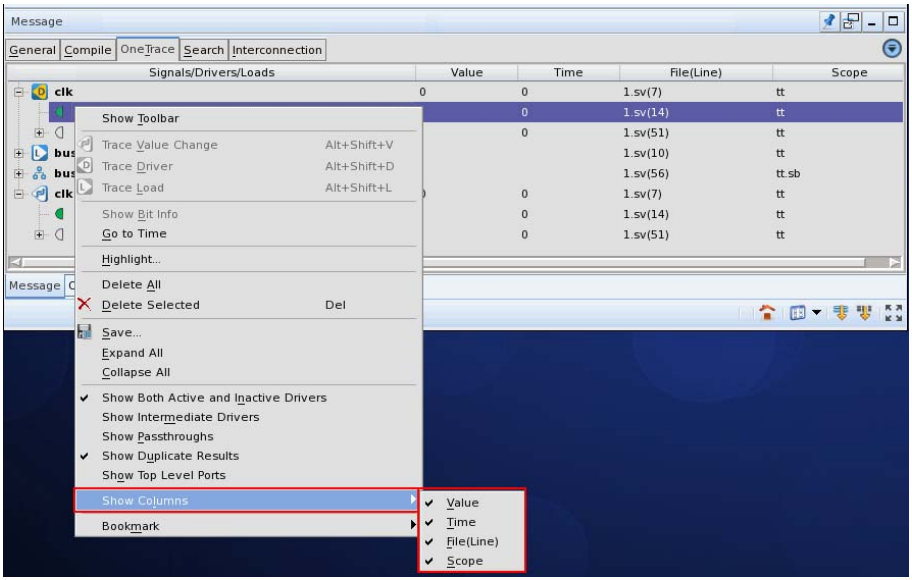


Figure: Show Columns Option

Bookmark

Bookmarks provides quick access to the saved views and enables you to easily toggle between them.

Set/Unset Bookmark

Bind Key: Ctrl+F2

This command sets or clears a bookmark at the current line of the traced signal as illustrated in the following figure:

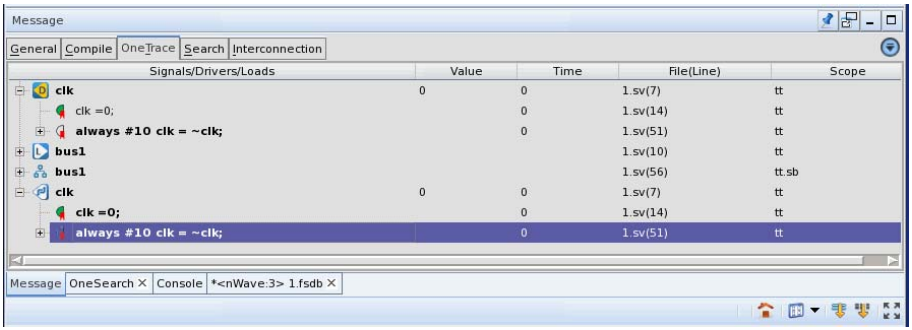


Figure: Set/Unset Bookmark

If there are more than 17 bookmarks sets, then the **More** option is available as illustrated in the following figure.

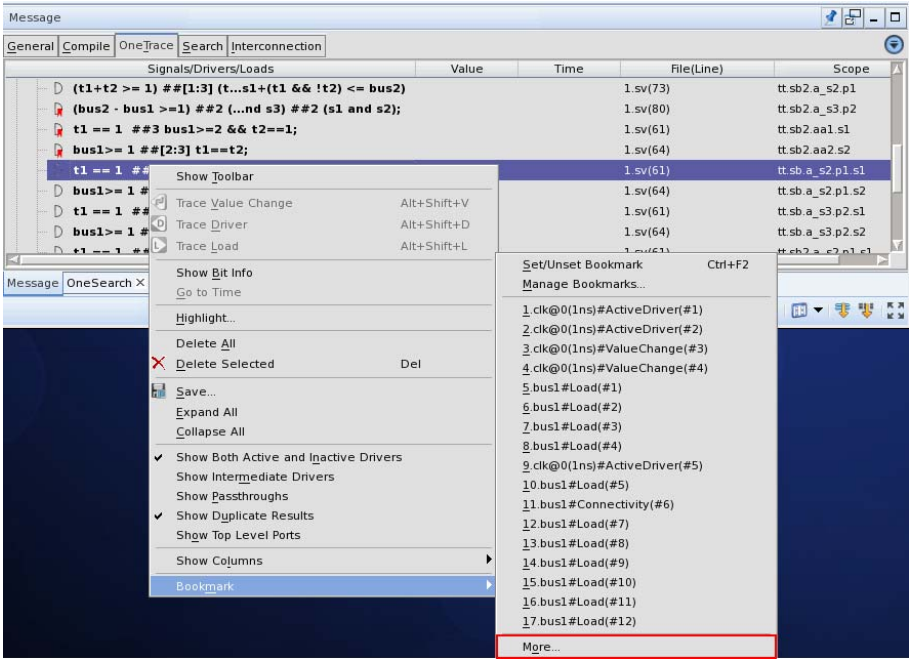


Figure: More Option

Click the **More** option or use the **o** shortcut key to open the *Manage Bookmarks* form.

For more information regarding the *Manage Bookmarks* form, see the Manage Bookmarks section.

Manage Bookmarks

This command opens the *Manage Bookmarks* form where the set bookmark(s) can be deleted, renamed, or located.

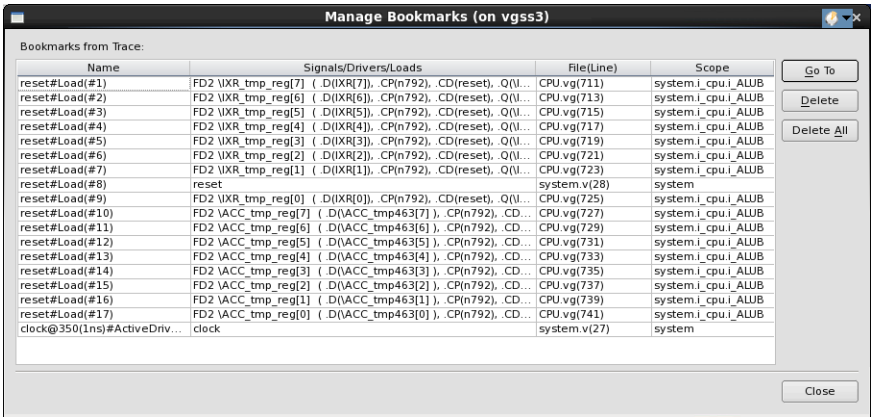


Figure: Manage Bookmarks Form

To rename bookmarks, select the editable cell (under the **Name** column). The background of the cell turns gray and can be edited, which indicates that the bookmark can be renamed.

The *Manage Bookmarks* Form includes the following fields:

- **Go To:** Click this button to open the selected bookmark in the traced signals and set it as active. Double-click the bookmark to set the scope as active. This is the default button.
- **Delete:** Click this button to delete the selected bookmark from the list.
- **Delete All:** Click this button to remove all the bookmarks.
- **Close:** Click this button to close the form.

Message Frame Toolbar Icons

The message frame toolbar icons are displayed when the **Show Toolbar** icon is clicked on the tab bar.

Figure: Message Frame - Find Toolbar - Compile Tab

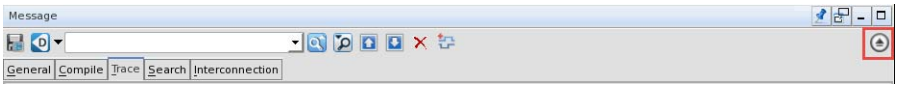


Figure: Message Frame - Find Toolbar - OneTrace Tab

Figure: Message Frame - Find Toolbar - Search Tab

Figure: Message Frame - Find Toolbar - General and Interconnection Tabs

Save

Click this toolbar icon to save the information in the current message tab to a file.

Load Compile Log

Click this toolbar icon to open the *Load Compile Log* form and select a compile log to load.

Trace Results

Click this toolbar icon to trace the driver, load, connectivity or active trace the selected signal on the **OneTrace** tab.

Find String

In this text field, enter the string to search on the **OneTrace** tab only.

Find Previous

Click this toolbar icon to find the previous string on the **OneTrace** tab.

Find Next

Click this toolbar icon to find the next string on the **OneTrace** tab.

Previous Trace Result

Click this toolbar icon to jump to the previous trace result on the **OneTrace** tab.

Next Trace Result 

Click this toolbar icon to jump to the next trace result on the **OneTrace** tab.

Add Trace to Waveform: 

Click this toolbar icon to add all the selected signals to the synchronized *nWave* window. You can also enable this option in the **Source Code - > Trace** page of the *Preferences* form (invoked with the **Tools -> Preferences** command).

Delete Selected 

Click this toolbar icon to remove the selected trace results from the **OneTrace** tab.

Search in All Files  /Search in Current File 

Click this toolbar icon to toggle the search range between all files or the current file. The default is to search all files.

Hide Toolbar  /Show Toolbar 

Click this toolbar icon to turn the display of the *Message* frame toolbar *on* or *off*. The default is to hide the toolbar.

Text Viewer Frame

The *Text Viewer* frame is located at the same frame location as the *Source Code* frame as a new tab. A *Text Viewer* tab can become a standalone window by clicking the **Undock** icon on the toolbar.

The menu bars for the *Text Viewer* frame are illustrated as follows:

Figure: Text Viewer Menus

The drop-down menus for the *Text Viewer* frame are summarized and explained in the following sections. Refer to the *Icons for Dockable Panes* section for details on the menu bar icons.

File commands

Save As

This command opens a *Save As* form where the directory structure can be viewed and a text file name specified for saving the compile log file contents.

Print

This command opens the *Print* form where print options for creating a hard copy can be specified. Refer to the **File** -> **Print** command description in the *nEditor* chapter for details.

Close Window

This command closes the current window.

Tools commands

Customize Menu/Toolbar

Refer to the **Tools -> Customize Menu/Toolbar** command in the main *nTrace* menu for details.

Text Viewer Frame Right-Click Options

When the right mouse button is clicked anywhere in the menu, toolbar icon, or frame banner area of the *Text Viewer* frame or standalone window, a configuration option menu is displayed. Use this menu to configure the available icons and frames.

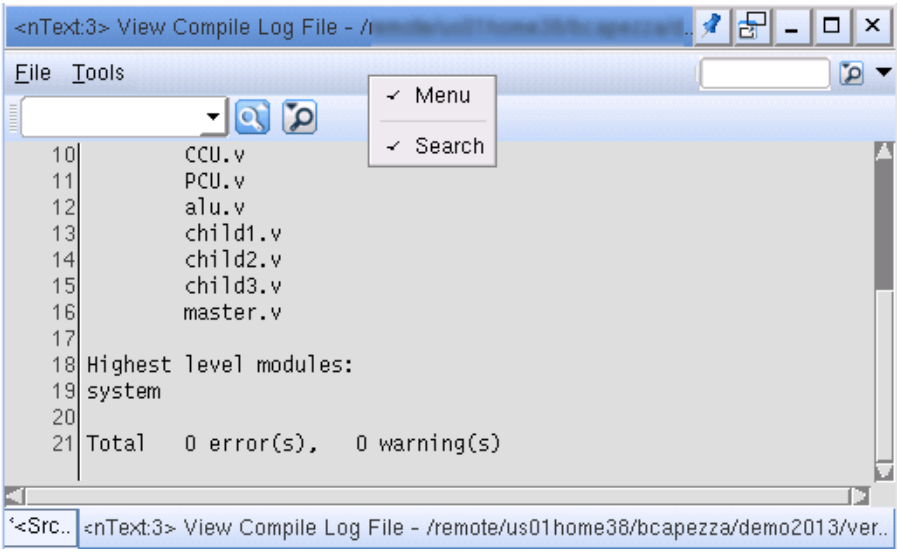


Figure: Configuration Option Men

The **Menu** option enables/disables the command menu bar. The options in the bottom section enable/disable the toolbar icons for different functions.

Text Viewer Frame Toolbar Icons

Figure: Toolbar Used in the Text Viewer Frame

The available toolbar icons may be modified. Refer to the *Toolbars* section of the *User Interface* chapter in the *Verdi User Guide and Tutorial* manual for details.

The different toolbar categories and available icons are described as follows:

Search Category

Find String 
In this text field, enter the string to search.

Find Previous 
Click this toolbar icon to find the previous string specified in the **Find String** text field.

Find Next 
Click this toolbar icon to find the next string specified in the **Find String** text field.

Right-Click Options

Many of the previously described commands can also be selected from the right-click menu option.

Main Window Right-Click Options

After the right mouse button is clicked anywhere in the menu, toolbar icon, or frame banner area of the main window, a configuration option menu is displayed.This menu can be used to configure the available icons and frames.

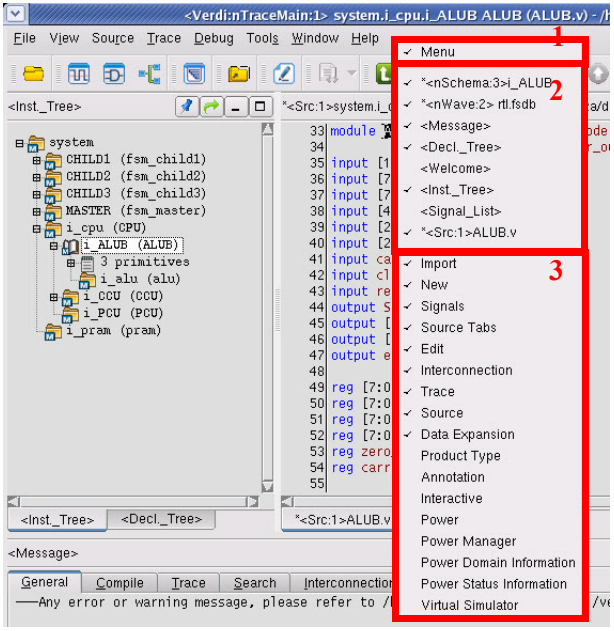


Figure: Configuration Option Menu

The **Menu** option (labeled 1 in the figure above) enables/disables the command menu bar. The options in the middle section (labeled 2 in the figure above) enable/disable the frames that have been previously opened. The options in the bottom section (labeled 3 in the figure above) enable/disable the toolbar icons for different functions.

After the menu bar is hidden (unchecked) with the **Menu** option, the Alt key can be pressed to display the menu again and then press the **Alt** key again or anywhere inside the docked window to hide it. The Alt key does not hide the menu bar if the **Menu** option is checked.

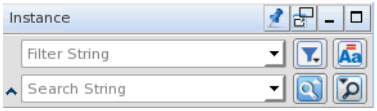
Design Browser Frame Right-Click Options

Following is the list of right-click menu options available for the design browser frame. Some selections are available from the right mouse button menu. Other selections have equivalent selections available from the toolbar menus.

Show Search/Filter

Bind Key: Ctrl+F

Toolbar Icon:



This right-click option allows you to filter and search the *Instance* tab for a specific node. This option allows you to display or hide the **Filter/Search** string text fields in the *Instance* tab. By default, the **Filter/Search** string text fields are hidden. You can also change this default setting using the **Instance Tree** preference option in the **Search/Filter** page (**General -> Search/Filter -> Enable Search/Filter on -> Instance Tree**) of the *Preferences* form. The Instance Tree preference option is disabled by default.

Enter a string in the **Filter** field, select the filter type by clicking the **Filter**  icon on the right, and press the **Enter** key.Tree nodes (including their parent nodes) matching the input string and the filter type are displayed in the *Design Browser* pane. Child nodes of the matched tree nodes are included and collapsed. The child nodes can be expanded as needed. Leaf nodes can also be filtered and searched in the *Design Browser* pane using the filters Leaf Verilog Cell, Leaf VHDL Cell, Leaf SystemC Cell, and Leaf Spice Cell.

Following are the available filter types. Multiple selections of the filter types are supported.

- Class
- Assertion
- Used Function
- Unused Function
- Used Task
- Unused Task
- Interface
- Named Block
- Unnamed Block

- Power Switch Cell (available when power files are loaded)
- Power Isolation Cell (available when power files are loaded)
- Power SPA Cell (available when power files are loaded)
- Power SRSN Cell (available when power files are loaded)
- Boundary Mux Cell (available when power files are loaded)
- Package
- Leaf Verilog Cell
- Leaf VHDL Cell
- Leaf SystemC Cell
- Leaf Spice Cell
- All

Enter a search string in the **Search String** text field and press the **Enter** key to locate the first match. Click the **Next Node**  icon to find the next matching tree node or click the **Previous Node**  icon to find the previous tree node.

In the **Filter/Search** string text fields, the wildcard characters (* or ?) are supported to specify the pattern-matching string. For example, if *CH** is entered, *CHILD1*, *CHILD2*, and *CHILD3* are all found.

You can also use the **Case sensitive**  icon to enable or disable case-sensitivity for the string entered in the **Filter** and **Search** fields. By default, the searching and filtering mechanisms are case-sensitive. You can change this default setting using the **Match case** preference option in the **Search/Filter** page (**General -> Search/Filter -> Search/Filter Options -> Match case**) of the *Preferences* form.

An example of the **Filter/Search** string text fields with the icons are illustrated in the following figure:

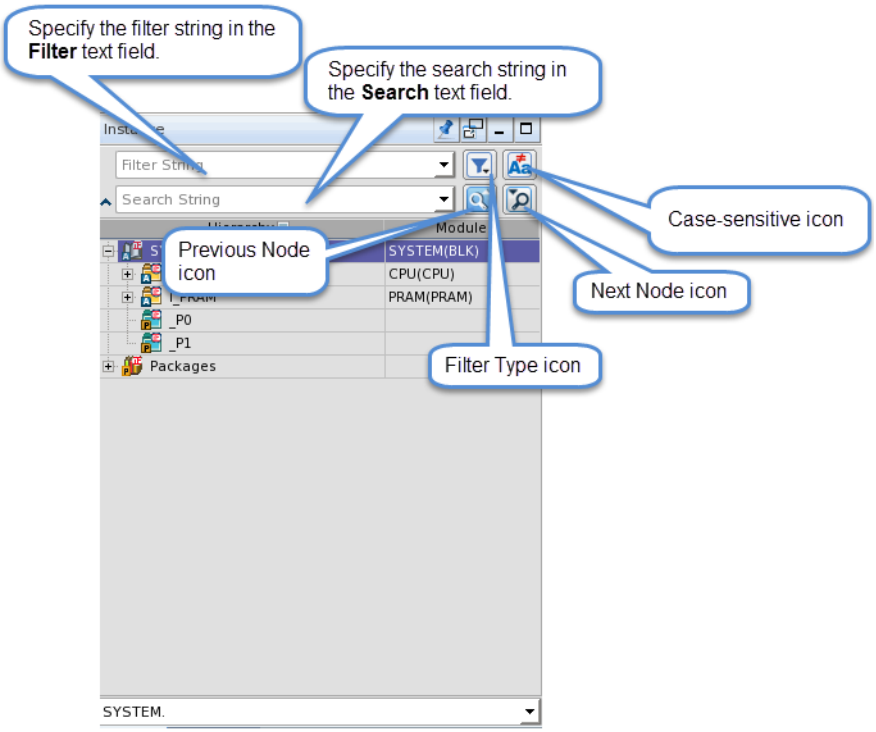


Figure: Example of the Filter/Search String Text Fields With the Icons

In the Verdi platform, the edit box, which is displayed at the bottom of the *Design Browser* pane displays the full hierarchy name of the selected node, as illustrated in the following figure. Also, you can enter a hierarchy name in the edit box to select the corresponding tree node.

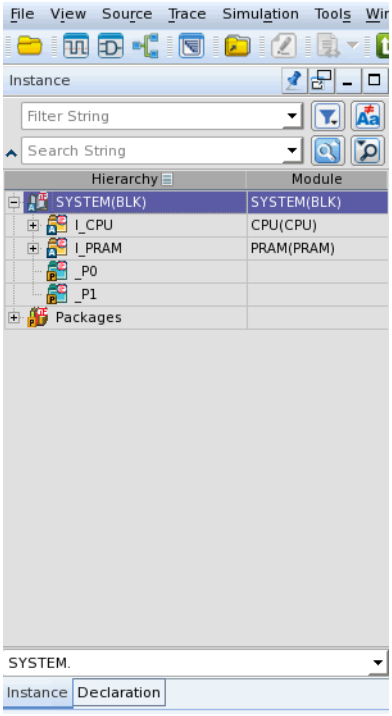


Figure: Search by Complete Name

Exclude Matched Nodes

Bind Key: Ctrl+E

This command is available in the design browser frame to exclude the tree nodes from the instance tree that match the specified criteria. This command opens the *Exclude Matched Nodes* form where the condition can be set/specified to exclude tree nodes from the instance tree. A progress bar is also displayed in *nTrace* to allow you to stop the exclusion progress.

NOTE: This command does not support clear single filtration. After reload or restore session, the *Exclude Matched Nodes* form is closed and the instance tree excludes the tree nodes from the instance tree that are specified. However, after you import the design, the *Exclude Matched Nodes* form is closed and the instance tree is restored to the original tree.

The *Exclude Matched Nodes* form is illustrated in *Figure: Exclude Matched Nodes Form*.

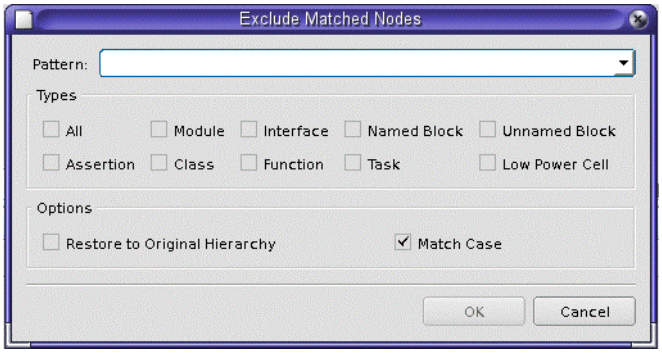


Figure: Exclude Matched Nodes Form

The *Exclude Matched Nodes* form includes the following fields:

- **Pattern:** This field (combo box) allows you to exclude tree nodes by specifying a pattern. It supports wildcard characters, and does not support a regular expression.
- **Types:** This field allows you to select a certain type of tree nodes for exclusion from the instance tree. It allows you to toggle the corresponding check boxes and also selects multiple types at the same time. By default, all the check boxes are unchecked.
- **Options:** This field provides the following two options:
  - **Restore to Original Hierarchy:** This option is unchecked by default. If this option is checked, *nTrace* restores to the original instance, and then excludes the matched nodes.
  - ***Match Case:*** This option is case sensitive and is checked by default. When the design is pure VHDL, this check box is always unchecked and disabled.

Restore to Original Instance

This command is available in the design browser frame to restore the original instance tree. The instance tree remains in the current expanded level after clearing.

Copy Full Path

Bind Key: Ctrl+H

This command copies the complete hierarchical name of the select scope into the clipboard buffer.

Show Tip

This command turns the tip display *on* or *off*. When this option is turned *on*, and the cursor is placed over an instance, the hierarchy name is displayed in the tip. When a low power design is loaded and you place the cursor on an instance, then you can also view tooltip with relevant power domain information. By default, the option is *on*. You can also select the **Ctrl+T** shortcut key to save the hierarchy name in the clipboard.

The following is an example of the *Show Tip* right-click option.

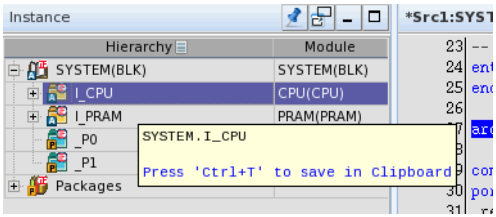


Figure: Example of Show Tip Option

Show Module Name

This command displays the *Module* column as shown in the following figure if the **Show Module Name** right-click option is turned *on.* By default, the option is *on.*

You can also select the **Display Module Name** preference option in the **Design Tree** page (**Source Code -> Design Tree -> Display Module Name**) of the *Preferences* form. This option is enabled by default.

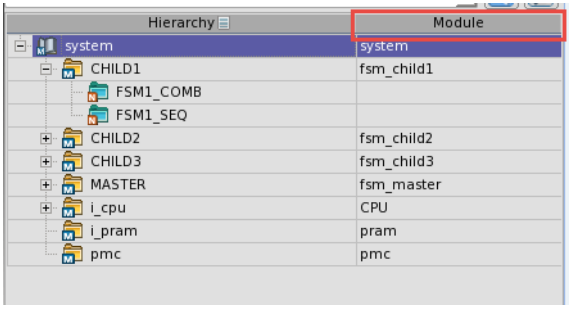


Figure: Module Name Column

Show Definition

The detailed description for this command is available under the **Source -> Show -> Definition** menu.

Show Calling

The detailed description for this command is available under the **Source -> Show -> Calling** menu.

Show Entity

The detailed description for this command is available under the **Source -> Show -> Entity** menu.

Show File Information

This command displays the definition and instantiation information for the selected scope on the **Interconnection** tab in the *Message* frame.

Show Analog Scope Only

This command displays only the analog instances in the hierarchy tree of the *Designer Browser* pane if the option is turned *on*. By default, the option is *off*.

NOTE: This command is available only after the analog design is loaded.

Expand Tree by Level

The detailed descriptions for the **Expand Tree by Level** command can be found in the **Hierarchy Tree by Level** command under the **View** menu.

- Expand Tree by Level-> All
- Expand Tree by Level-> 2 Levels
- Expand Tree by Level-> 3 Levels
- Expand Tree by Level-> 4 Levels
- Expand Tree by Level-> 5 Levels

Collapse All Scopes

This command collapses all the tree nodes in the design browser frame.

Add IE-Nets to Waveform

This command adds the IE-net types from the *nTrace* designer browser pane to the waveform.

The following are sub-commands of the **Add IE-Nets to Waveform** command.

- **All:** This command enables you to add all IEs (no through-net) in the design to the current waveform file.
- **Current Scope:** This command enables you to add all IEs (no through-net) in the current scope to the current waveform file.
- **Current Scope and Below:** This command enables you to add all IEs (no through-net) in the current and the below scope to the current waveform file.

NOTE: This command is available only if the Spice design and the FSDB file are loaded.

Display Source Code in New Tab

This command displays the source code for the selected scope in a new tab in the *Source Code* frame. The new tab and its associated scope become active.

Allow Multiple Selection

This command turns multiple selections in the design browser frame *on* and *off*. The selected item(s) can be dragged and dropped to other windows. Multiple items can be selected by holding the **Shift** key (selects a range) or the **Ctrl** key (selects non-contiguous items) and a left-click or drag-left.

Export Hierarchy

This command dumps the design hierarchy tree to an ASCII file.

Behavior Analysis

Refer to the **Tools -> Temporal Flow View** -> **Behavior Analysis** command description for details.

Interface Browser

Refer to the **Tools -> Browse/Watch/List** -> **Interface Browser** command description for details.

Set/Unset Bookmark

This command sets (a red bookmark symbol  is added to the scope) or clears a bookmark on the selected node in the design browser frame. The selected node must be a scope. Bookmarks return the design browser to the bookmarked area by clicking the **Go To** button in the *Scope Bookmarks* form (through the right-click menu option **Manage Bookmarks** command) or using the bookmark cache. One usage of bookmarks is to keep records of important decision branches during the back-trace process.

Go to Bookmark

This command displays the bookmark cache. The bookmarks are displayed with a number (reflecting the order they are set) and the full scope for the bookmark. The bookmark is truncated when the length exceeds 40 characters. The design browser opens the scope for the selected bookmark and is set as active.

Manage Bookmarks

This command opens the *Manage Bookmarks* form where the set bookmark(s) can be deleted or opened (go to).

The bookmarks in the **Marked Scopes** list also appear in the right-click menu option **Go to Bookmark** command as a bookmark cache, which can be used to go to the bookmark in the design browser without accessing the *Manage Bookmarks* form.

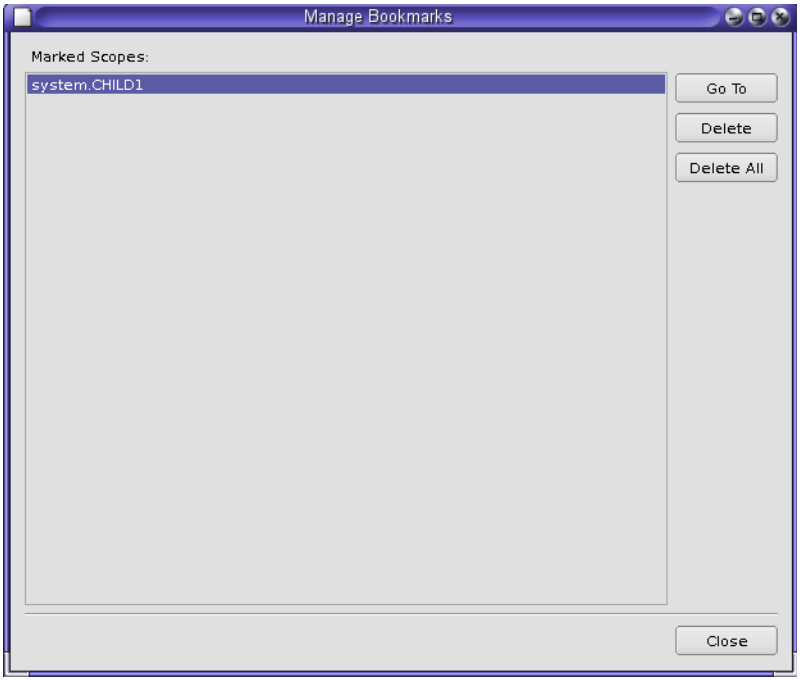


Figure: Manage Bookmarks Form

The *Manage Bookmarks Form* includes the following fields:

- **Go To:** Click this button to open the selected bookmark in the design browser frame and set it as active. Double-click the bookmark to set the scope as active.
- **Delete:** Click this button to delete the selected bookmark from the list.
- **Delete All:** Click this button to remove all the bookmarks.
- **Close:** Click this button to close the form.

Source Code Frame Right-Click Options

Following is the list of right-click menu options available for the *Source Code* frame. Some selections are available from the right mouse button menu. Other selections have equivalent selections available from the toolbar menus.

Set as Active Scope

This command moves the selected object to the top of the *Source Code* frame.

Drag

Bind Key: Ctrl+C

The selected signal can be dragged to another location and dropped. This is similar to using the middle mouse button to select and drag.

Drop

Bind Key: Ctrl+V

The selected signal dragged from another location can be dropped in this window. This is similar to releasing the middle mouse button for a drop.

Copy Full Path

Bind Key: Ctrl+H

This command copies the complete hierarchical name(s) of the selected signal(s) and instance(s) into the clipboard buffer. It only applies to the *Source Code* frame and not the *Message* frame.

Function Step In

NOTE: This command is available when the **Source -> Function Annotation** command is *on* and a function calling statement is right-clicked.

When the **Function Step In** command is invoked, the calling function's stack information is listed in the **Function Stack** pane. Double-click or right-click a nested function and invoke this command to continue with the function tracing.

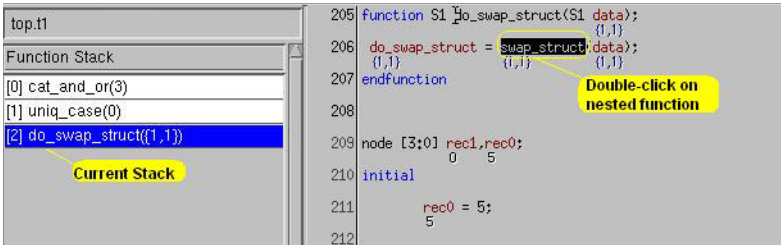


Figure: Trace into Function Stacks

Function Step Out

NOTE: This command is available when the **Function Step In** command is enabled.

To trace into a function, right-click the function calling statement and invoke the **Function Step Out** command to step out to the calling position. Alternatively, double-click the function calling statement in the **Function Stack** pane or *Source Code* frame to step out to the calling position.

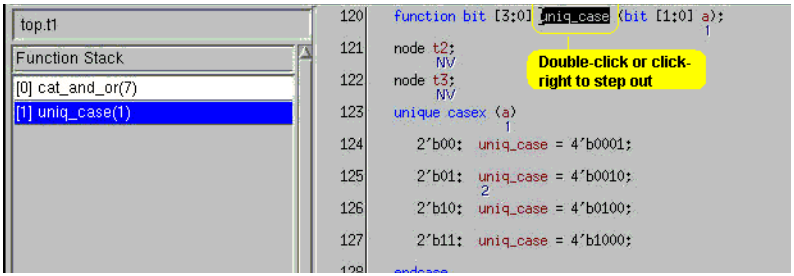


Figure: Step Out to Function Calling

Show Calling

Refer to the **Source -> Show ->** **Calling** command description for details.

Show Definition

Refer to the **Source -> Show ->** **Definition** command description for details.

Display Liberty Definition

Bind Key: Ctrl+L

NOTE: This command is enabled when a cell is selected. To obtain the cell definition, a liberty file should be loaded first.

This command displays the definition of liberty file for the selected cell.

Show Reference

NOTE: This command is available when a design object is selected.

Refer to the **Source -> Show -> Reference** command description in the *Testbench Debug* chapter for details.

Jump to Report File

NOTE: This command is available when an interface element is selected.

This command opens a window to display the interface element report file, and jumps to the line in which the selected interface element is defined.

OneTrace

The following are the sub-commands of the **OneTrace** command.

Value Change

Refer to the **Trace -> Value Change** command description for details.

Driver

Refer to the **Trace -> Driver** command description for details.

Load

Refer to the **Trace -> Load** command description for details.

Connectivity

Refer to the **Trace -> Connectivity** command description for details.

Chain Driver

Refer to the **Trace -> Chain Driver** command description for details.

Mapped Signal

Refer to the **Trace -> Mapped Signal** command description for details.

Active Trace

Bind Key: Ctrl+T

NOTE: This command is enabled when a simulation results file is loaded (use the **File -> Open Waveform File** command). This command locates the active driver in the *Source Code* frame for the selected signal.

Trace Power

Refer to the *Design Source Code Pane Right-click Options* section in the *Power Aware Debug* chapter for details.

Watch Expressions

NOTE: This command is available when a the **Source -> Active Annotation** command is turned on.

Refer to the **Tools -> Browse/Watch/List** -> **Watch Expressions** command description for details.

Specify Array Index

NOTE: This command is available when a simulation results file is loaded, the **Source -> Active Annotation** command is turned *on*, and an array signal under the for-loop or genvar/generate-for statement in the *Source Code* frame is selected.

This command opens the *Specify Index Value for Array* form in which the for-loop or genvar/generate index value(s) can be set. The for-loop or genvar/generate index value is applied to the entire loop block, including nested for-loops, and not only the selected line.

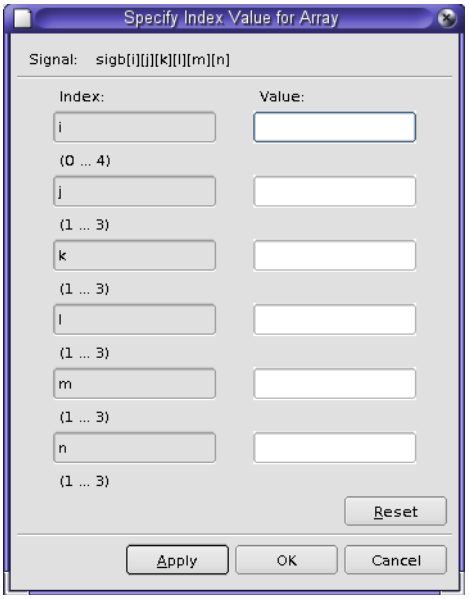


Figure: Specify Index Value for Array Form

If an input index value is out of bound, a warning message requesting for a valid input, is reported. However, only signals dumped in the FSDB file, constants or parameters can be applied to the above rule.

NOTE: The array index that is not within a bracket '[ ]' can also be specified.

Assertion Analyzer

NOTE: This command is available when the **Source -> Active Annotation** command is turned on, and an assertion object in the *Source Code* frame is selected.

This command can be enabled after a design with SystemVerilog Assertions (SVA) and the FSDB file with assertion results are imported. Select the failure/success point of an assertion and invoke the **Assertion Analyzer** command. If the assertion has more than one time range, a *Begin Time List* form appears displaying all time ranges.

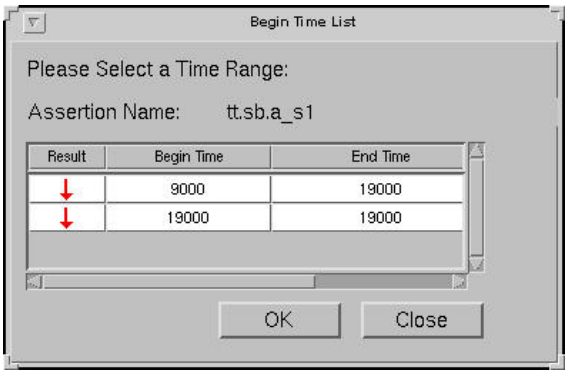


Figure: Begin Time List Form

After selecting a time range in the *Begin Time List* form click **OK** to open the *Analyzer* frame to trace the active driver of assertions. The *nWave* window is also available with the signals related to the assertion displayed.

NOTE: The *Analyzer* frame is directly invoked after clicking the **Assertion Analyzer** command if the assertion only has a single time range (this includes assertions which begin/end point are at the same time).

If more than two properties or sequences exist in the same scope, **Assertion Analyzer** stops at this scope.

For details on the *Analyzer* window, refer to the *Analyzer Pane* section in the *Assertion Debug* chapter. For details on the assertion comments, refer to the *Assertion Comments* section of the *nWave* chapter.

Navigator

NOTE: This command is available when an assertion object or a signal in the *Source Code* frame is selected.

This command launches VC Formal for further analysis on assertions and properties. For details, refer to the VC Formal documentation.

Highlight

Refer to the **Signal -> Highlight** command description for details.

Add to Waveform

The following are sub-commands of the **Add to Waveform** command.

New Waveform

This command enables you to create a new waveform window, open an annotation file, and add signals.

Add to Wave[n]

Bind Key Ctrl+4

This command enables you to add selected objects to the last adding signal waveform or the new created waveform.

Add to New Group

This command creates a new group in the waveform and adds selected objects to this group.

*Wave[k]

Bind Key Ctrl+W

This command adds selected objects to the waveform window. For example, *Wave[k], k is the primary waveform window Id.

Wave[m]

This command adds selected objects to waveform window. For example, Wave[m], m is the window id.

Wave[m]

This command adds selected objects to waveform window. For example, Wave[m], m is the window id.

New Schematic

Following are the sub-commands of the **New Schematic** command.

Fan-in

Refer to the **Fan-in** command for details.

Fan-out

Refer to the **Fan-out** command for details.

Driver

Refer to the **Driver** command for details.

Load

Refer to the **Load** command for details.

Connectivity

Refer to the **Connectivity** command for details.

Flattened Window

Refer to the **Flattened Window** command for details.

Hierarchical Flattened View

Refer to the **Hierarchical Flattened View** command for details.

Temporal Flow View

NOTE: This command is available when a signal is selected.

The **Temporal Flow View** menu includes the following sub-commands:

Auto Trace

Refer to the **Tools -> Temporal Flow View -> Auto Trace** command description for details.

Trace Glitch

Refer to the **Tools -> Temporal Flow View -> Trace Glitch** command description for details.

New Temporal Flow View

Refer to the **Tools -> Temporal Flow View -> New Temporal Flow View** command description for details.

Add Reference Signals

Refer to the **Tools -> Temporal Flow View -> Add Reference Signals** command description in the *FlowView* chapter for details.

Trace X

Refer to the **Trace X** command for details.

Follow Signal

NOTE: This command is available when a signal in the *Source Code* frame is selected.

When a signal is selected and this command is invoked, the results (driver/load/connectivity of the selected signal in its upper/lower hierarchy) is displayed in the **Follow Signal** sub-command menu with a line separating the results in the upper hierarchy and in the lower hierarchy. The format of the follow signal is *scope(signal)* or *scope(.port(portInst))* (if the follow signal is a port). Selecting one of the displayed follow signals changes to the selected follow signal's scope and the signal's string is displayed in the **Find String** text field.

An example of the **Follow Signal** sub-command menu is illustrated in the following figure:

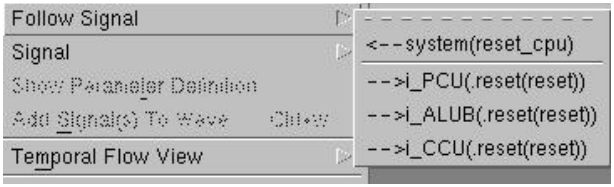


Figure: Follow Signal Sub-command Menu

The label <-- indicates that the selected signal is a load of the follow signal; --> indicates that the selected signal is a driver of the follow signal; <-> indicates that the selected signal is both a load and a driver; --- indicates an unknown direction between the selected signals and the follow signal. 'No result' is displayed if the selected signal does not have follow the signal in the upper/lower hierarchy.

Signal

The **Signal** menu includes the following sub-commands: **Snip Signal(s), Unsnip Signal(s)**, **Show Equivalent Signal, Show Signal Definition, Show Signal Type, Signal Value Radix,** and **Signal Value Notation**.

Snip Signal(s)

This command snips the selected signal(s). If a signal is snipped, the driver/load tracing stops at this signal.

Unsnip Signal(s)

This command unsets the snipped signals.

Show Equivalent Signal

This command opens the *Equivalent Signal* window which displays the selected signal and its equivalent signals. You can also save the results to a file. For example, if you select *system.i_cpu.clock*, it provides the equivalent signals *system.CHILD1.Clock*, *system.CHILD2.Clock*, *system.CHILD3.Clock*, *system.MASTER.Clock*, *system.clock*, *system.i_cpu.i_ALUB.clock*, *system.i_cpu.i_CCU.clock*, and *system.i_pram.clock* in the *Equivalent Signal* form.

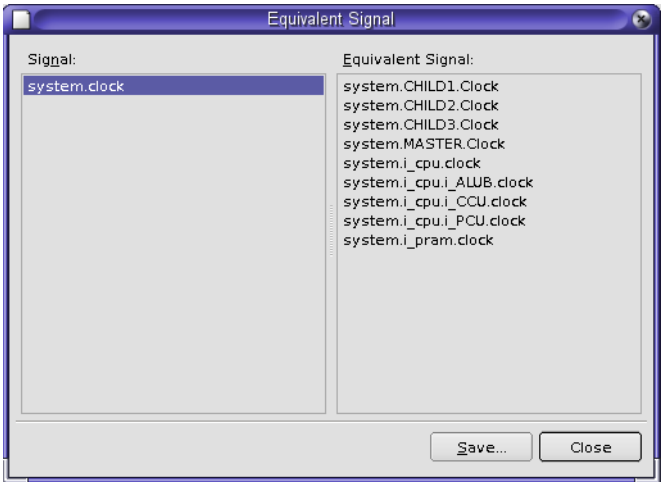


Figure: Equivalent Signal

Show Signal Definition

This command jumps to signal declaration line of the selected signal.

Show Signal Type

This command jumps to signal type declaration line of the selected signal.

Signal Value Radix

This command changes the value radix of the selected signal from the **Binary, Octal, Hexadecimal, Decimal, ASCII** and **Enumerated Literal** sub-commands.

Refer to the **Waveform -> Signal Value Radix** command description for details.

NOTE: The **Signal Value Radix** command can be activated after a simulation results file is loaded from the **File ->Open Waveform File** command and the **Source -> Active Annotation** command is invoked.

Signal Value Notation

This command displays the value radix of the selected signal from the **Unsigned, Signed 2's Complement, Signed 1's Complement**, and **Signed Magnitude** sub-commands.

Refer to the **Waveform -> Signal Value Notation** command description for details.

NOTE: The **Signal Value Notation** command can be activated after a simulation results file is loaded from the **File -> Open Waveform File** command and the **Source -> Active Annotation** command is invoked.

Parameter

NOTE: This command is available when a parameter is selected.

The following are sub-commands of the **Parameter** command.

Show Parameter Definition

This command jumps to parameter declaration line of the selected signal.

Value Radix

NOTE: This command is available when the parameter annotation is turned on and a parameter is selected.

This command changes the value radix of the selected parameter. The options are **Binary**, **Octal**, **Hexadecimal**, and **Decimal**.

Value Notation

NOTE: This command is available when the parameter annotation is turned *on* and a parameter is selected.

This command displays the value radix of the selected parameter. The options are **Unsigned**, **Signed 2's Complement**, **Signed 1's Complement** and **Signed Magnitude**

Variable

The following are sub-commands of the **Variables** command.

Dump Variable Value

To enable this command, import a mixed language or VHDL design and load the simulation results first. Then, select a VHDL variable and invoke this command

to calculate the variable value from the simulation begin to end time into the data agent.

NOTE: This value is not saved in the FSDB file and disappears when you exit the Verdi platform.

Dump Variable Snapshot

To enable this command, import a mixed language or VHDL design and load the simulation results first. Then, select a VHDL variable and invoke this command to calculate the variable value at the current time into the data agent.

NOTE: This value is not saved in the FSDB file and disappears when you exit the Verdi platform.

Signal

This command adds all the selected signals to the synchronized *nWave* window.

Add Instrumented Signals to Waveform

This command adds all the instrumented signals to the synchronized *nWave* window.

Add Signal(s) in Block to Waveform

NOTE: These commands are available when an FSDB file is loaded.

The **Add Signal(s) in Block to Waveform** menu includes three sub-commands: **All**, **Sensitivity List**, and **Contents**.

The block is defined as a block with `begin` and `end` keywords.

The following code is used to illustrate the signals that are recognized for each command.

```
module top;
  logic signal_a, signal_b;
  logic clk, is_enabled, condition;

  always @(posedge clk & is_enabled)
  begin
    if (condition)
      signal_a = '1;
    else
      signal_b = '1;
  end
```

```
endmodule
```

All

After selecting the beginning line of a block (that is, the line with the begin keyword), this command adds all signals in the selected block to the synchronized *nWave* window.

Sensitivity List

After selecting the beginning line of a block (that is, the line with the `begin` keyword), this command adds signals in the sensitivity list of the selected block to the synchronized *nWave* window.

In the example above, `top.clk` and `top.is_enabled` are added as signals from the sensitivity list.

Contents

After selecting the beginning line of a block (that is, the line with the `begin` keyword), this command adds all signals in the contents (located between the `being` and `end` keywords) to the synchronized *nWave* window.

In the example above, contents includes the signals between the `begin` and `end` keywords (that is, `top.condition`, `top.signal_a`, and `top.signal_b`).

Interface Browser

NOTE: This commands is available when an interface object is selected.

Refer to the **Tools -> Browse/Watch/List -> Interface Browser** command description for details.

Temporal Flow View

NOTE: This command is available when a signal is selected.

The **Temporal Flow View** menu includes the following sub-commands:

Auto Trace

Refer to the **Tools -> Temporal Flow View -> Auto Trace** command description for details.

Trace Glitch

Refer to the **Tools -> Temporal Flow View -> Trace Glitch** command description for details.

New Temporal Flow View

Refer to the **Tools -> Temporal Flow View -> New Temporal Flow View** command description for details.

Add Reference Signals

Refer to the **Tools -> Temporal Flow View -> Add Reference Signals** command description in the *Flow View* chapter for details.

Trace X

Refer to the **Tools -> Trace X** command description in the *nTrace* chapter for details.

Debug Memory

The **Debug Memory** menu includes four sub-commands: **Dump Memory Waveform to FSDB, Show Memory Contents, Trace Memory Write,** and **Memory Definition Table**.

Dump Memory Waveform to FSDB

Refer to the **Tools -> Dump Memory Waveform to FSDB** command description in the *Temporal Flow View* chapter for details.

Show Memory Contents

This command opens the **Dumped by Simulator** or **Calculated by Verdi** tab of the *Get Memory Variable* form and the *nMemory* window. The complete name of the selected memory variable is displayed in the *Get Memory Variable* form. Refer to the **Get Memory Variable** command description in the *Memory/MDA* chapter for details.

Trace Memory Write

This command traces the previous memory write on the selected memory net. The address, data, and control for the memory net are used to determine when the selected location are written previously. If the address location is initialized to the current value and writes do not occur previously, an information form opens stating “This array element is assigned during initialization (value=*n*),” the cursor jumps to time 0 and the initialization statement is highlighted in the source

code. If write occurs previously, the information form opens stating when the array element is assigned, the global cursor jumps to the write time and the write equation is highlighted in the source code. Any additional messages is printed to *nTrace*'s message form.

Memory Definition Table

Refer to the **Tools -> Memory -> Memory Definition Table** command description for details.

Correlate to Gate Design/Correlate to RTL Design

NOTE: This command is available when a signal or an instance is selected.

Source Code Line Number Area Right-Click Options

Following is the list of right-click menu options available for the line number area which is on the left side of the hypertext *Source Code* frame.

Manage Breakpoints

Refer to the **Simulation -> Manage Breakpoints** command description in the *Interactive Simulation Debug* chapter for details.

Set/Unset Bookmark

Refer to the **Source -> Set/Unset Bookmark** command description for details.

Previous Bookmark

Refer to the **Source -> Previous Bookmark** command description for details.

Next Bookmark

Refer to the **Source -> Next Bookmark** command description for details.

Edit Bookmarks

Refer to the **Source -> Manage Bookmarks** command description for details.

Toolbar Icons and Fields

Figure: Default Toolbar Used in Main Window for Source Code Frame

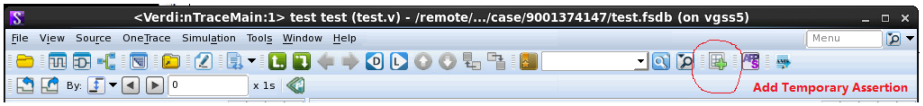


Figure: Toolbar in Main Window with Add Temporary Assertion Icon

By default the *Add Temporary Assertions* icon is hidden in the toolbar of main window. But when you switch to *Assertion Debug Mode*, this toolbar is displayed.

The available toolbar icons may be modified. Refer to the *Toolbars* section of the *User Interface* chapter in the *Verdi User Guide and Tutorial* manual for details.

The different toolbar categories and available icons are described as follows:

Import Category

Import Design

Click this toolbar icon to import designs from either a pre-compiled library or from a design file. Refer to the **File -> Import Design** command description for details.

New Category

New Waveform

Click this toolbar icon to open a new *nWave* window. Refer to the **Tools -> New Waveform** command in the *nWave* chapter for details.

New Schematic

Click this toolbar icon to open a new schematic window. Refer to the **Tools -> New Schematic From Source -> New Schematic** command in the *nSchema* chapter for details.

Temporal Flow View 

Click this toolbar icon to open a new *Flow View* window. Refer to the **Tools -> Temporal Flow View** command for details.

Signals Category

Signal List 

Click this toolbar icon to turn *on* and *off* the display of signals in each scope in the *Signal List* frame in *nTrace*. Refer to the **View -> Signal List** command description for details.

Source Tabs Category

New Source Tab 

Click this toolbar icon to open a blank source code tab in the *Source Code* frame. Refer to the **View -> Source Tab -> New Source Tab** command for details.

Edit Category

Edit Source File 

Click this toolbar icon to edit the source code. Refer to the *nEditor* chapter for details.

Interconnection Category

Show Reference 

NOTE: This toolbar icon is activated when SystemVerilog code is loaded.

Click this toolbar icon to show references. Click the arrow to change the reference type. Refer to the **Source -> Show -> Reference** command description in the *Testbench Debug* chapter for details.

Trace Category

Calling 

Click this toolbar icon to go to the statement that called (displayed in the *Source Code* frame) the current active scope. Refer to the **Source -> Show -> Calling** command description for details.

Definition 

Click this toolbar icon to go to the definition statement (displayed in the *Source Code* frame) of the selected instance in the *Source Code* frame. Refer to the **Source -> Show -> Definition** command description for details.

Backward History 

Click this toolbar icon to trace back through the 32 most recent trace results. You can also use this command (toolbar icon) to trace back when using the **Show Calling** or **Show Definition** command in *nTrace*.

Forward History 

Click this toolbar icon to trace forward through the 32 most recent trace results.

Trace Driver 

Click this toolbar icon to trace all the possible drivers of the selected signal. Refer to the **OneTrace -> Driver** command description for details.

Trace Load 

Click this toolbar icon to trace all the possible loads of the selected signal. Refer to the **OneTrace -> Load** command description for details.

Show Previous 

Click this toolbar icon to jump to the previous trace result in the instance by highlighting the corresponding semicircle symbol in the indicator area with a color-filled semicircle. Refer to the **OneTrace -> Show Previous** command description for details.

Show Next 

Click this toolbar icon to jump to the next trace result in the current instance by highlighting the corresponding semicircle symbol in the indicator area with a

color- filled semicircle. Refer to the **OneTrace -> Show Next** command description for details.

Show Previous in Hierarchy 

Click this toolbar icon to jump in the instances of the traced results. Refer to the **OneTrace -> Show Previous in Hierarchy** command description for details.

Show Next in Hierarchy 

Click this toolbar icon to jump in the instances of the traced results. Refer to **OneTrace -> Show Next in Hierarchy** command description for details.

Source Category

Set/Unset Bookmark 

Click this toolbar icon to set or clear a bookmark at the current line in the source code. Refer to the **Source -> Set/Unset Bookmark** command description for details.

Find String 

In this text field, enter the string to search in the current file. Refer to the **Source -> Find String** command description for details.

Find Previous 

Click this toolbar icon to find the previous string specified in the **Search for** text field.

Find Next 

Click this toolbar icon to find the next string specified in the **Search for** text field.

VIA Tool Bar Category

Click the upside-down triangle on the right of the  icon to display the **VIA Tool Box** and **VIA Log Viewer** commands.

VIA Tool Box 

Click this toolbar icon to access the VIA tool box.

VIA Log Viewer

Click this toolbar icon to open the *VIA Log Viewer* window where the execution log of VIA application is displayed and contents in the frame can be linked to the *nTrace* main window.

Annotation Category

Figure: Default Toolbar Used in Main Window with Active Annotation

Previous Change in Current Statement

This command finds the previous change, either rising or falling edge, for all signals in the source code statement for the following four types of source code: Instance instantiation, Continuous assignment, Initial statement, and Always block.

NOTE: Remember to move the cursor to any position in the statement before invoking these commands.

The driving signal that causes a transition may be difficult to identify if the statement has delay information (for example, explicit delay or gate-level library cell). The **Previous Change in Current Statement** function can therefore prove useful for signal delay verification in your source code, especially if you are not using a waveform display. Using this function, you can easily locate the previous transition of all the signals in the current statement.

Next Change in Current Statement

This command finds the next change, either rising or falling edge, for all signals in the source code statement for the following four types of source code: Instance instantiation, Continuous assignment, Initial statement, and Always block.

NOTE: Remember to move the cursor to any position in the statement before invoking these commands.

The driving signal that causes a transition may be difficult to identify if the statement has delay information (for example, explicit delay or gate-level library cell). The **Next Change in Current Statement** function can prove useful for signal delay verification in your source code, if you are not using a waveform display. Using this function, you can easily locate the next transition of all the signals in the current statement.

Search By 

Click this toolbar icon to specify the search criteria for the value change of a signal.

Search Backward 

Click this toolbar icon to search backwards in time and locate the value in the waveform window. This icon is active only when an FSDB file is loaded and the **Source -> Active Annotation** command is enabled.

Search Forward 

Click this toolbar icon to search forwards in time and locate the value in the waveform window. This icon is active only when an FSDB file is loaded and the **Source -> Active Annotation** command is enabled.

Cursor Time Entry/Display 

Specify the cursor time in the text field. Refer to the **Waveform -> Waveform Time ->** **Set Window Time Unit** command in *nWave* for details.

Auto Trace 

NOTE: This toolbar icon is available when the **Active Annotation** command is turned *on*.

After a signal is selected, click this icon to trace the source in the *Flow View* for the selected signal. The trace mode is based on the cycle-based or transition-based setting in the **Default Trace Method** section on the **Temporal Flow View -> Trace** page of the *Preferences* form.

On the cycle-based setting, if a signal containing an unknown “X” value is selected, click this toolbar icon to invoke the **Trace X** command without opening the *Temporal Flow View* window. If a signal containing other values except X is selected, click this toolbar icon to invoke the **Trace This Value** command with a *Temporal Flow View* window opened.

On the transition-based setting, if a signal containing an unknown “X” value is selected, click this toolbar icon to invoke the **Trace X** command without opening the *Temporal Flow View* window. If a signal containing other values except X is selected, click this toolbar icon to invoke the **Trace Triggering Path** command with a *Temporal Flow View* window opened.

Interactive Category

Refer to the *Interactive Simulation Debug Toolbar Icons and Fields* section of the Interactive Simulation Debug chapter for details.

Power Category

Refer to the *Power Category* section in the *Power Aware Debug* chapter for details.

Power Manager Category

Refer to the *Power Manager Category* section in the *Power Aware Debug* chapter for details.

Toggle Category

Data Expansion 

Both the design and simulation results (FSDB file) must be loaded for this toolbar icon to be activated. Refer to the **Tools -> Visibility -> Data Expansion -> Enable Data Expansion** or **Disable Data Expansion** command descriptions in the *Visibility* chapter for details.

Data Expansion Refresh 

Both the design and simulation results (FSDB file) must be loaded for this toolbar icon to be activated. Refer to the **Tools -> Visibility -> Data Expansion -> Refresh Signal Values** command description in the *Visibility* chapter for details.

Power Domain Information Category

Refer to the *Power Domain Information Category* section in the *Power Aware Debug* chapter for details.

Power Status Information Category

Refer to the *Power Status Information Category* section in the *Power Aware Debug* chapter for details.

nSchema

Overview

The *nSchema* window is displayed when the **Tools -> New Schematic from Source -> Current Scope** command is invoked from the *nTrace* menu bar or the **New Schematic** icon. The *nSchema* window is docked in the same area as the *Source Code* frame as a new tab.

Click the **Undock** icon on the toolbar to make it a standalone window.

Three types of *nSchema* views are available as follows:

- **Full Hierarchical:** Displays the complete design in a specific scope. The design can be traversed hierarchically by pushing the view into a lower level for the hierarchical instance, or popping the view up to an upper level.

  The *Full Hierarchy* window can be invoked using one of the following methods:

  - In the *nTrace* window, select the scope in the design browser frame and click **Tools -> New Schematic from Source -> Current Scope** command.
  - In the *nTrace* window, click the **Schematic** icon on the toolbar to display the active scope from the design browser frame in a new tab.
- **Browser:** Displays a partial schematic view in a specific scope. The partial design can be traversed hierarchically. Double-click the symbol ports to push in, pop up, or expand the logic in the *Browser* window.

  The *Browser* window can be invoked using one of the following methods:

  - In the *nTrace* window, click **Tools -> New Schematic from Source -> Browser Window** command.
  - In the *nSchema* window, click **Tools -> New Schematic -> Browser Window** command.
- **Flattened:** Displays a partial schematic in a flat view. The hierarchy is flattened and schematics cross all scopes in the *Flattened* window. A hierarchical connector is added when crossing the hierarchy boundary. Objects can be added into the *Flattened* window by double-clicking on the instance pins or dragging-and-dropping objects from other windows.

  The *Flattened* window can be invoked using any of the following methods:

- In the *nTrace* window, select the primitive instance and invoke the **Tools -> New Schematic from Source -> Flattened Window** command.
- In the *nSchema* window, select the primitive instance and invoke the **Tools -> New Schematic -> Flattened Window** command.
- In the *nTrace* window, select the primitive instance and invoke the **Tools -> New Schematic from Source** -> **Hierarchical Flattened View** command. This command opens the *Hierarchical Flattened View* frame that displays the hierarchical boundaries in the *Flattened Schematic* window.
- In the *nSchema* window, select the primitive instance and invoke the **Tools -> New Schematic** -> **Hierarchical Flattened View** command. This command opens the *Hierarchical Flattened View* frame that displays the hierarchical boundaries in the flattened schematic window.
- In the *nTrace* window, select a signal/instance and invoke any one of the following commands: **Tools -> New Schematic from Source -> Fan-in**, **Fan-out**, **Driver**, **Load**, or **Connectivity**.
- In the *nSchema* window, select a signal/instance and invoke one of the following commands: **Tools -> New Schematic -> Fan-in**, **Fan-out**, **Driver**, **Load**, or **Connectivity**.
- In the *nTrace* window, invoke one of the following commands: **Tools -> New Schematic from Source -> Clock Tree** or **Tools -> New Schematic from Source -> Reset Tree**. These commands open a special clock/reset tree flattened window.
- In the *nSchema* window, invoke one of the following commands: **Tools -> New Schematic -> Clock Tree** or **Tools -> New Schematic -> Reset Tree**. These commands open a special clock/reset tree flattened window.

NOTE: Several sub-menus exist under the **Tools -> New Schematic** command. With the exception of the **Current Scope** option, *nSchema* windows generated by the others sub-menus are known as partial schematic windows.

The menu bars for the *nSchema* window are as follows:

Figure: nSchema Menu Bar - Full Hierarchical

Figure: nSchema Menu Bar - Flatten and Browser

When an area is selected in *nSchema*, if a full instance is not included in the selection, the pin(s) are kept in the selection. If any full instances are included in the selection, the instance(s) are kept in the selection.

The pull-down menus for *nSchema* are summarized and explained on the following sections. Refer to the *Icons for Dockable Panes* section for details on the menu bar icons.

Menu Summary

The *nSchema* menu options are summarized below. All commands can be double-clicked to jump to the corresponding command description. For right-click commands, refer to the *Right-Click Commands* section for details.

File Commands

Save in HDL
Import Path Data File
Load Clock Domain Results
Load Clock Tree Database
Load SDC File (Flattened View Only)
Print
Capture Window
Close Window

Edit Commands

Undo
Redo
Viewing Objects
Remove from Viewing Objects
Clear All Viewing Objects

View Commands

Instport Name
IO Port Name
Hier. Connector Port Name (Flattened View Only)
Module Name
Entity Name
Arch Name
Instance Name
Net Name
Complete Name
Power/Ground Pin
Locator
Align Locator Right
Symbol Annotation
Short Name (Flattened View Only)
SDC Annotation (Flattened View Only)
Highlight Power Aware Object ->
  Power Domain Color
  Power Aware Object Color
Preselect
Parameter List
High Contrast
Detailed RTL
Hierarchical Flattened View (Flattened View Only)
Instrumented Cell (Flattened View Only)
Zoom ->
  Zoom In
  Zoom Out
  Zoom All
  Minimap
  Fit Select Set
Pan ->
  Pan Left
  Pan Right
  Pan Up
  Pan Down
Pop View Up from Port
Pop View Up
Push View In
Last View

Schematic Commands

Find Signal/Instance/Instport
Find in Current Scope
Find Locator
Auto Fit Found Object(s)
Selection ->
  Select All Objects
  Select Instances/Signals
  Select All Instances
  Select All Signals
  Select All Pins
  Deselect All
Focus Connection
Add to Waveform
Color Instances by Scope (Flattened View Only)
Add Locator
Change Color
All Objects to Default Color
Active Annotation
Annotate in Color
Leading Zeros
Signal Value Radix
Signal Value Notation
Propagated Value
Constant Net
Set Annotation Time Range
Signal Current Value Type
Signal Voltage Value Type
SDF Annotation - nAnalyzer
Cell Delay - nAnalyzer
Delay Scale - nAnalyzer
Delay Type - nAnalyzer
Delay Precision - nAnalyzer
ViewMark (Hierarchical View Only)
Sync. Active Scope
Recent Schematics ->
  [Filename]

Trace Commands

Driver
Load
Connectivity
Fan-in
Fan-out
Active Fan-in
Advanced Trace
Two Points
By Level
Shortest/Longest Path
Show Trace Report
Value Propagation ->
  Value Propagation
  Conflict Report
Clock Analyzer ->
  Find Clock Source - nAnalyzer
  Extract Clock Information - nAnalyzer
  List Clock Domains - nAnalyzer
  List Specified Clock Domains - nAnalyzer
  Highlight Clock Domain - nAnalyzer
  Check Crossing Paths - nAnalyzer
  List Crossing Paths - nAnalyzer
Clock Tree ->
  Highlight Clock Tree - nAnalyzer
  Unhighlight Clock Tree - nAnalyzer
  Save Clock Tree - nAnalyzer
  Clock Skew - nAnalyzer
Reset Tree ->
  Highlight Reset Tree - nAnalyzer
  Unhighlight Reset Tree- nAnalyzer
  Save Reset Tree - nAnalyzer
  Reset Skew - nAnalyzer
Add Results to Waveform (Hierarchical View Only)
Backward History
Forward History
Reset History
Options ->
  Ignore Scopes during Tracing
  Auto Merge Setting - nAnalyzer
  Trace Stop on Module Boundaries
  Stop Cone Tracing on FSM
  Stop Cone Tracing on MOS Cell
  Hide Extraneous Bus (Flattened View Only)
  Auto Group Buffer/Inverter Cell (Flattened View Only)
  Auto Merge Cell - nAnalyzer

Tools Commands

New Schematic ->
  Current Scope
  Browser Window
  Flattened Window
  Hierarchical Flattened View
  ECO Window for Selected Instance(s) - nECO
  ECO Window for All Instances - nECO
  Editable Window for Selected
  Editable Window for All
  From Trace Results
  Fan-in
  Fan-out
  Driver
  Load
  Connectivity
  Clock Tree
  Reset Tree
Rearrange Schematic (Flattened View Only)
ECO Utility ->
  Clone Logic to ECO Window - nECO
  Clone Module - nECO
  Change Instance Scope - nECO
  Create Port - nECO
  Cell Type Replacement - nECO
  Spare Cell - nECO
Extract Interactive FSM
List User-defined Delay
Netlistcom Information (Full Hierarchical View Only)
Preferences
Customize Menu/Toolbar

Bind Keys

For a complete list of bind keys used in *nSchema*, refer to the *nSchema* section in the *Bind Key Summary* for details.

File Commands

Save in HDL

Menu Bar: File -> Save in HDL

This command writes the source code corresponding to the instances to a specified output file, sorted by scope, source file name, begin line, and end line. An output file name and keynote text string must be specified (the keynote text string appears as the first line of the output file).

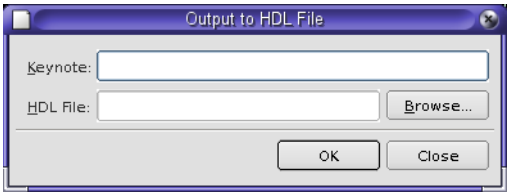


Figure: Output to HDL File Form

Import Path Data File

Menu Bar: File -> Import Path Data File

Refer to the **File -> Import Path Data Files - nAnalyzer** command description in the *nTrace* chapter for details.

Load Clock Domain Results

Menu Bar: File -> Load Clock Domain Results

Refer to the **File -> Load Clock Domain Results - nAnalyzer** command description in the *nTrace* chapter for details.

Load Clock Tree Database

Menu Bar: File -> Load Clock Tree Database

Refer to the **File -> Load Clock Tree Database - nAnalyzer** command description in the *nTrace* chapter for details.

Load SDC File (Flattened View Only)

Menu Bar: File -> Load SDC File

This command opens the *Load SDC* form in which the specified SDC file can be loaded.

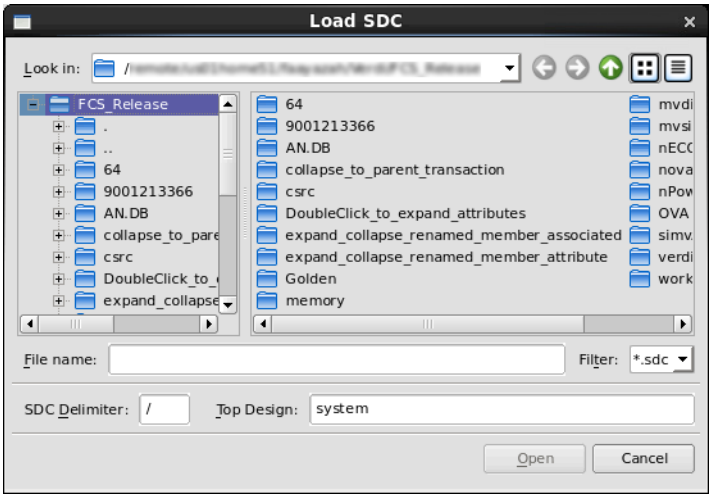


Figure: Load SDC Form

NOTE: The SDC file can be loaded multiple times, but only the latest SDC file takes effect. The displayed SDC annotations are based on the latest SDC file imported.

Print

Menu Bar: File -> Print

This command prints the current view from the schematic window. Before the view is printed, different print settings can be specified on the *nSchema Print* form using the **Basic**, **Page Mode**, **Signature,** and **Color** tabs.

Basic Tab

The **Basic** tab specifies general printing settings.

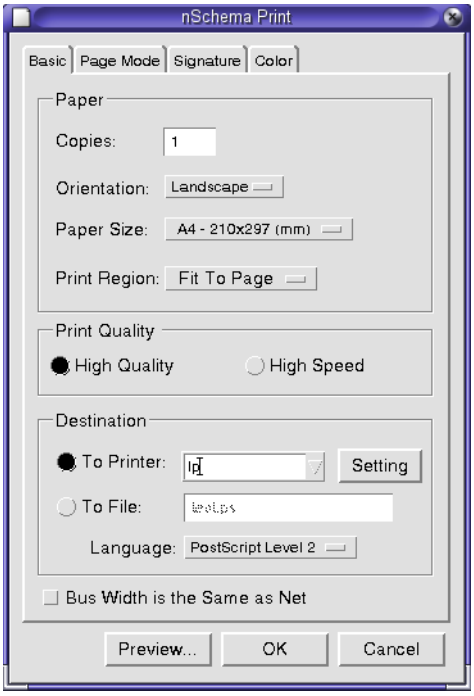


Figure: nSchema Print - Basic Tab

The **Paper** section includes the following options and fields:

- **Copies**: Enter the number of copies to be printed.
- **Orientation**: Specify the orientation for the printed page by selecting either **Landscape** or **Portrait**.
- **Paper Size**: Select the paper size of the printed page from the list.

- **Print Region**: Specify the printing region by selecting one of the following: **All**, **Current View**, or **Fit To Page**. The default option is *Current View*.

The **Print Quality** section includes two options. Select either **High Quality** or **High Speed** for the print quality preference.

The **Destination** section includes the following options and fields:

- **To Printer**: When this option is turned *on*, the file is printed from the printer specified in the text field.
- **Setting**: Click this option to configure the print options as follows:
  - Enter the name of the required printer in the **Printer Name** field.
  - Select one of the following output formats this printer supports: **PostScript Level 2**, **PostScript Level 1**, **MIF**, or **GL/2**.
  - Select the **Paper Size** supported by the printer. Multiple selection of paper sizes is allowed. If **User-defined** is specified under **Paper Size**, then the width and height of the paper must be specified.
  - In the **Print Command** section, the default **Command** is *lpr*, the default **Destination** is *-P*, and the default **Copies** is *-#*. These entries can be used for printing the file, and can be configured if a personal print command exists.
- **To File**: When this option is turned *on* and the file name is specified in the **To File** text field, the print file is saved in the working directory in the postscript format. When the **To File** option is turned *on*, an option must be selected from **PostScript Level 2** or **PostScript Level 1** in the **Language** selection field.

When the **Bus Width is the Same as Net** option is turned *on*, the printed bus width is the same as the net.

Page Mode Tab

The **Page Mode** tab specifies page settings for printing.

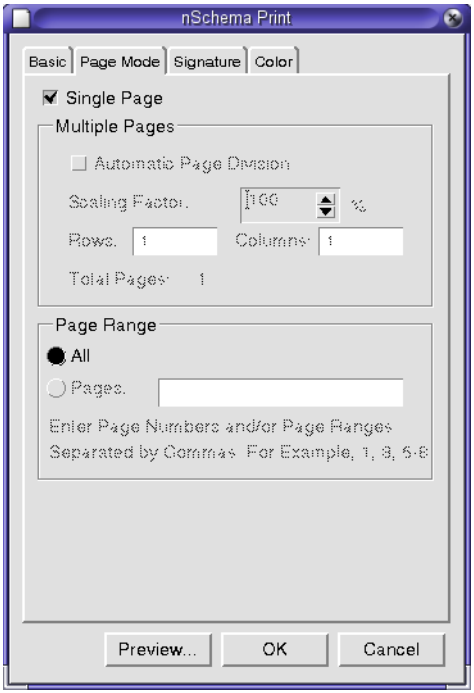


Figure: nSchema Print - Page Mode Tab

This tab provides two page mode sections: **Single Page** and **Multiple Pages**. The default option is *Single Page*.

- **Single Page:** Turn this option *on* to print a schematic diagram on one page. When this option is turned *off*, options related to the **Multiple Pages** mode are enabled automatically. **All** is the default value for the **Page Range** section when the **Single Page** option is turned *on*.

  The **Multiple Pages** section distributes a schematic diagram across multiple pages. To see how the schematic diagram is spread, click **Preview**.

  The **Multiple Pages** section includes the following fields:

  - **Automatic Page Division:** Turn this option *on* to calculate the page number automatically based on the **Scaling Factor**. The number of rows, columns, and total pages are automatically displayed. When this option is turned *off*, the pages must be divided manually by adjusting the **Rows** and **Columns**.

- **Scaling Factor:** Specify the printing ratio. The default ratio is *100%*. The scale setting automatically changes the number of **Columns**, **Rows**, and **Total Pages**.
- **Total Pages:** Displays the page number based on the size of x (**Columns**) and y (**Rows**) that the printout occupies. These figures are recalculated automatically according to the **Scaling Factor**.

The **Page Range** section includes the following options:

- **All:** Turn this option *on* to print all pages. The default option is *All* for the **Page Range** section.
- **Pages:** When this option is selected, you can specify the pages to be printed. A comma is used to separate pages, such as enter2, 4, 5 to print pages 2, 4, and 5. A hyphen is used for a range, such as enter 4-6 to print pages 4, 5, and 6.

Signature Tab

The **Signature** tab configures the signature in the printout.

Figure: nSchema Print - Signature Tab

The *nSchema Print - Signature Tab* form includes the following fields, options, and buttons:

- **Location:** Specify the signature location by selecting one of the following options: **None**, **Auto**, **Top**, **Bottom**, **Upper Left**, **Upper Right**, **Lower Left**, and **Lower Right**. The default option is *Lower Right*.
- **Design Name:** Enter the design name. The default design name is the module name.
- **Designer:** Enter the designer's name. The default designer name is the user's login name.
- **Description:** Enter the description. The default description is blank.
- **Date:** Specify the date. The default date is the system time.
- **Version:** Enter the version. The default version is blank.
- **Disable Boarder:** When the signature information is disabled by selecting **None** in the **Location** option, the border is removed from the printout.

Color Tab

The **Color** tab specifies the color for the selected item.

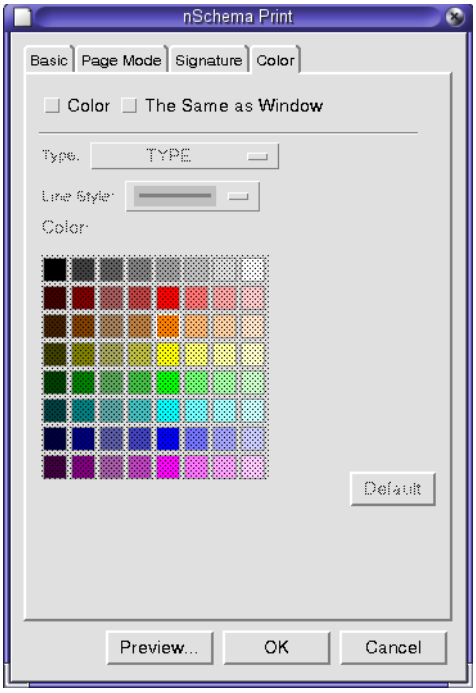


Figure: nSchema Print - Color Tab

The *nSchema Print - Color Tab* includes the following options and fields:

- **Color:** Turn this option *on* to specify the preferred color of the selected item in **Type**. When this option is turned *on*, the printout includes multiple colors; if not, the printout is black and white on a color printer.
- **The Same as Window:** Turn this option *on* to make the preview and the print colors appear the same as they are in the window.

  When the **Color** option is turned *on*, the preferred color settings can be specified in the **Type** and **Line Style** selection fields. The new color preference is saved and applied when you click **OK**.
- **Default:** Click **Default** to restore the default settings.

NOTE: The color preference setting is for printouts only. The color for the schematic window can be set by invoking the **Tools -> Preferences** command in the *nSchema* window.

Preview: Click **Preview** to display the *Print Preview* form, as illustrated in the following figure:

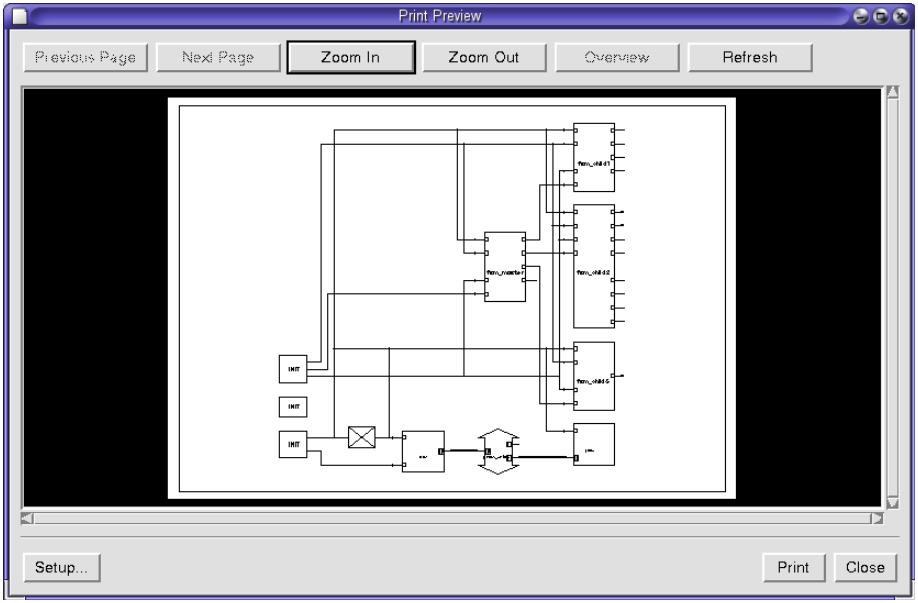


Figure: Print Preview Form

- In the *Print Preview* form, select the following options:
  - **Zoom In/Zoom Out:** Click **Zoom In** or **Zoom Out**, or their bind keys **Shift+Z** or **Z**, respectively, to control the schematic view. Press the bind

key **F** to fit the image to the size of the preview window. When the printout consists of multiple pages, double-click one of the segments to get the comp page.

- **Previous Page/Next Page:** Click **Previous Page** or **Next Page** to view the contents of other pages.
- **Overview:** Click **Overview** to get the overview of the schematic image, as illustrated above.
- **Refresh:** If the *nSchematic* window is changed, click **Refresh** to view an updated print preview image.
- **Setup:** Click **Setup** to open the *nSchema Print* form to display the print settings.

Capture Window

Menu Bar: File -> Capture Window

This command opens the *Capture Window - Preview* form where an image in Portable Network Graphic (PNG) format, a GNU standard similar to GIF can be the output.

The following fields and buttons are available in the *Capture Window - Preview* form:

- **Footer:** Enter the appropriate text which needs to be displayed on the footer.
- **Crop:** Drag the left mouse button to select the required area and then click **Crop** to update the display in the *Capture Window - Preview* form.
- **Print:** This button opens a *Print* form where the required print options for creating a hard copy can be specified.
- **Save as:** This button opens a *Save As* form where the directory structure can be viewed and a file name for the PNG file can be specified.

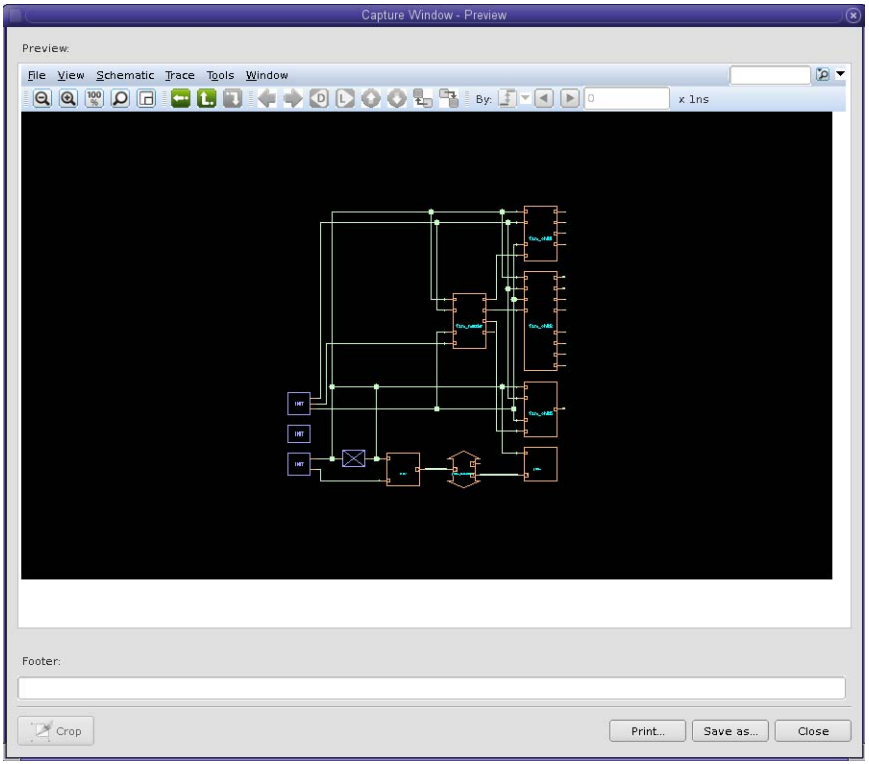


Figure: Capture Window - Preview Form

Close Window

Menu Bar: File -> Close Window

This command closes the current *nSchematic* window. Alternatively, you can close the window using the window manager control button.

Edit Commands

NOTE: The **Edit** commands are available in the *Browser* and *Flattened* schematic windows.

Undo

Menu Bar: Edit -> Undo

Toolbar Icon: 

Bind Key: Ctrl+Z

This command reverses the previous action and removes the content that is added from the view.

Redo

Menu Bar: Edit -> Redo

Toolbar Icon: 

Bind Key: Ctrl+Y

This command repeats the previous action and adds the content that is removed back to the view.

Viewing Objects

Menu Bar: Edit -> Viewing Objects

NOTE: This command is available in a *Browser* window.

This command opens the *nSchema Viewing Objects* form where the signals and instances in the current view are listed and can be selected. Select either the **Instance** or **Signal** options to specify the desired object type The default option is *Signal*.

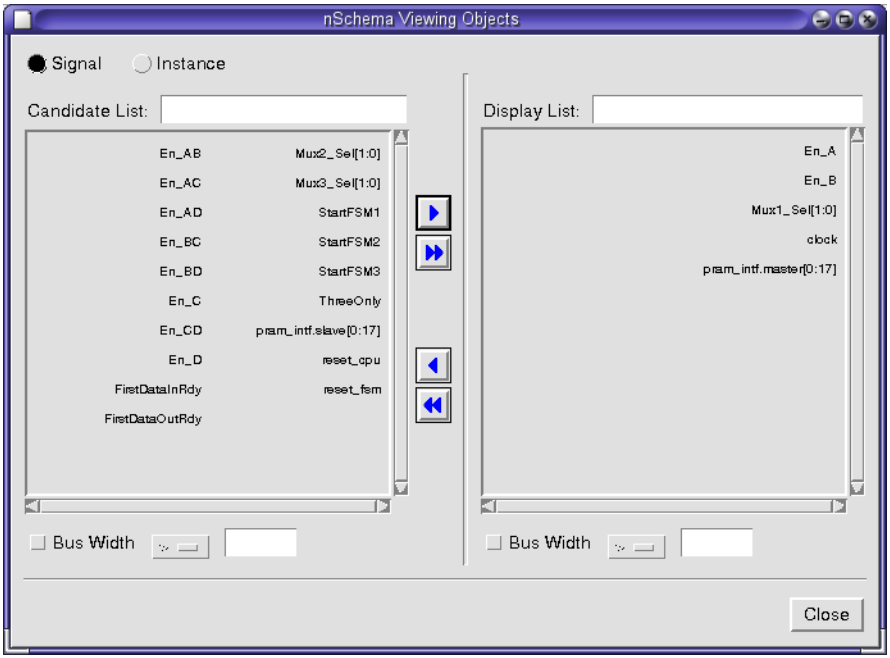


Figure: nSchema Viewing Objects Form

The *nSchema Viewing Objects* form includes the following fields, options, and buttons:

- **Add Object** : Click this option to move the selected object from the **Candidate List** to the **Display List**.
- **Add All** : Click this option to move all objects in the **Candidate List** to the **Display List**.
- **Remove Object** : Click this option to move the selected object from the **Display List** to the **Candidate List**.
- **Remove All** Click this option to move all objects in the **Display List** to the **Candidate List**.
- **Bus Width**: When this option is turned on, select from one of the conditional operators: greater than (>), greater than or equal to (>=), equal to (=), less than or equal to (<=), or less than (<) to specify the bus width criteria. This conditional operator is combined with the number of bits specified. Press **Enter**, to display the buses meeting the criteria in the left pane.

Remove from Viewing Objects

Menu Bar:	Edit -> Remove from Viewing Objects
Toolbar Icon:	
Bind Key	Del

This command removes the selected object from the current view. Hold the **Ctrl** key and left-click the object to select and remove multiple objects.

Clear All Viewing Objects

Menu Bar:	Edit -> Clear All Viewing Objects
Bind Key	Shift+C

NOTE: This command is available in a *Flattened* window.

This command removes the complete object (schema) from the current view.

View Commands

Instport Name

Menu Bar: View -> Instport Name

This toggle command turns the display of instance port names *on* or *off*.

IO Port Name

Menu Bar: View -> IO Port Name

This toggle command turns the display of Input/Output (I/O) port names *on* or *off*.

Hier. Connector Port Name (Flattened View Only)

Menu Bar: View -> Hier. Connector Port Name

NOTE: This command is available in the *Flatten*ed window.

This toggle command turns the display of module port names for hierarchical connectors in flattened schematic windows *on* or *off*.

Module Name

Menu Bar: View -> Module Name

This toggle command turns the display of module names *on* or *off*.

Entity Name

Menu Bar: View -> Entity Name

This toggle command turns the display of entity names inside the modules *on* or *off*.

Arch Name

Menu Bar: View -> Arch Name

This toggle command turns the display of architecture names *on* or *off*.

Instance Name

Menu Bar: View -> Instance Name

This toggle command turns the display of instance names *on* or *off*.

Net Name

Menu Bar: View -> Net Name

This toggle command turns the display of local net names *on* or *off*.

Complete Name

Menu Bar: View -> Complete Name

This toggle command turns the display of complete net/module names *on* or *off*. If the net/module names are too long and this option is turned *off*, only a partial name can be viewed.

Power/Ground Pin

Menu Bar: View -> Power/Ground Pin

This toggle command turns the display of power/ground pins *on* or *off*.

Locator

Menu Bar: View -> Locator

Bind Key: W

This toggle command turns the display of the locator on or off.

Align Locator Right

Menu Bar: View -> Align Locator Right

This toggle command aligns the locator to the right.

Symbol Annotation

Menu Bar: View -> Symbol Annotation

NOTE: This command is available for UDDB files. A UDDB file is generated by the *txt2uddb* utility.

After a UDDB file is loaded or the *schAddAnnotation* tcl command is used to add the symbol annotation then this toggle command in the active and the new *nSchema* window is turned *on*. The default option is *off*.

Short Name (Flattened View Only)

Menu Bar: View -> Short Name

NOTE: This command is available in the *Flattened* window.

This toggle command turns the display of short names *on* or *off*.

SDC Annotation (Flattened View Only)

Menu Bar: View -> SDC Annotation

NOTE: This command is available after an SDC file is loaded in the *Flattened* window.

This toggle command turns the display of SDC annotations *on* or *off*. When this command is turned *on*, information of the supported SDC constraints is annotated on the defined objects in the flattened schematic window and the **Schematic -> Propagated Value** command is disabled. The default option is *off*.

Options for the following SDC commands are supported for annotation:

SDC Type	Annotation Color	Supported Annotation Options
create_clock	Red	*source_objects* [-name *clock_name*] [-add] [-period *period_value*] [-waveform *edge_list*]
create_generated_clock	Green	*source_objects* -source *master_pin* [-name *clock_name*] [-add] [-master_clock *clock*] [-divide_by *divide_factor* | -multiply_by *multiply_factor*]
set_case_analysis	Yellow	value *port_or_pin_list*
set_disable_timing	Purple	*object_list* [-from *from_pin_name* -to *to_pin_name*]

Highlight Power Aware Object

NOTE: These commands are available when CPF/UPF files are loaded.

Power Domain Color

Menu Bar: View -> Highlight Power Aware Object -> Power Domain Color

Bind Key: Ctrl+D

After the CPF/UPF file is loaded and the **Highlight Power Domain** option is turned *on* (available on the **General -> Power** page of the *Preferences* form invoked with **Tools -> Preferences** command), invoke this toggle command to highlight the power domain color for instances and nets.

To change the highlight color, select the **Power Domain Instance** item in the **Power Aware Object** section and specify a preferred color on the **General -> Power** page of the *Preferences* form (invoked with **Tools -> Preferences** command).

Power Aware Object Color

Menu Bar: View -> Highlight Power Aware Object -> Power Aware Object Color

Bind Key: Ctrl+O

After the CPF/UPF file is loaded and the **Highlight Power Aware Design Object(s)** option is turned *on* (available on the **General -> Power** page of the *Preferences* form), invoke this toggle command to highlight the power aware color for the following four signal types:

- Retention signals for ports/nets
- Isolation signals for ports/nets
- Level shifted signal nets
- Boundary port signals

To change the highlight color, select the **Retention Object**, **Isolation Object**, **Level Shifted Object**, or **Boundary Port Object** item in the **Power Aware Object** section and specify a preferred color on the **General -> Power** page of the *Preferences* form (invoked with **Tools -> Preferences** command).

NOTE: Isolation signals' clamp values (for UPF) or output values (for CPF) are displayed on pins if the Isolation signals have clamp/output values. The clamp/output value format is as follows:
1. Single net which has single clamp/output value: (clamp/output value)
2. Single net which has multiple clamp/output values: (...)
3. Concatenated net's clamp/output values: (clamp/output value1, clamp/output value2, ...)

If the sub-net do not have any clamp/output value, “NF” is displayed in the corresponding position. If the **Active Annotation** or **Propagate Value** commands are *on*, an Isolation signal's clamp/output value are appended after the active annotation or propagation value. The format is: Active Annotation / Propagate Value / Clamp/Output Value.

NOTE: If the clamp value (for UPF) or output value (for CPF) is not specified in an Isolation rule command, the default Isolation signal value is displayed on the pins. The default value is UPF: *(0)* and CPF: *(low)*.

NOTE: An Isolation signal's clamp value (for UPF) or output value (for CPF) is also displayed in a tip when the **Show Tip** and **Show Power Information** options are both turned *on* in the **Schematics -> Display Options -> Select** page of the *Preferences* form and the cursor is placed over the Isolation signal.

Preselect

Menu Bar: View -> Preselect

When this command is *on*, signals or blocks are highlighted and a tip is opened when the cursor moves over the objects in the *nSchema* window. The frame banner at the top of the *nSchema* window displays the name of the selected signal or blocks even if the **Preselect** option is turned *on* or *off*. By default, this option is turned on.

Parameter List

Menu Bar: View -> Parameter List

When this toggle command is *on*, the parameter list is displayed on the module block (the parameter list must be defined in the design file).

High Contrast

Menu Bar: View -> High Contrast

When this toggle command is *on*, the schematic view is blurred except for the selected and traced set.

NOTE: The highlight color specified with the **nSchema** -> **View** -> **High Contrast** command does not overwrite the highlight color specified with the **nSchema** -> **Schematic** -> **Propagated Value** and **nSchema** -> **Schematic** -> **Constant Net** commands.

Detailed RTL

Menu Bar: View -> Detailed RTL

When this toggle command is *on*, the circuit is displayed as a boolean equivalent logic. When this toggle command is *off*, the circuit is displayed as a function symbol.

Hierarchical Flattened View (Flattened View Only)

Menu Bar: View -> Hierarchical Flattened View

When this toggle command is *on*, it displays the hierarchical boundaries in the flattened schematic window. When this toggle command is *off*, it does not display the hierarchical boundaries in the flattened schematic window.

NOTE: The **Hierarchical Flattened View** command creates a schematic view in *nSchema* after an instance or signal is selected.

Instrumented Cell (Flattened View Only)

NOTE: This command is available in the *Flatten*ed window after a UPF design is loaded.

When this toggle command is *on*, Isolation cells are displayed in the current flattened schematic window. When this toggle command is *off*, Isolation cells are not displayed.

Zoom

Zoom In

Menu Bar: View -> Zoom -> Zoom In

Toolbar Icon: 

Bind Key: Shift+Z

This command provides a close view of the content in the *nSchema* window. The magnification of the viewing area is changed to half the magnification of the previous view.

NOTE: A specific area can be zoomed by dragging-left to form a rectangle around the area to be zoomed.

Zoom Out

Menu Bar: View -> Zoom -> Zoom Out

Toolbar Icon: 

Bind Key: Z

This command enables more of the contents to be viewed in the *nSchema* window at a reduced size. The magnification of the viewing area is changed to two times the magnification of the previous view from the center point in both the horizontal and vertical directions.

Zoom All

Menu Bar: View -> Zoom -> Zoom All

Toolbar Icon: 

Bind Key: F

This command displays all the contents of the schematic.

Magnifying Glass

Menu Bar: View -> Zoom -> Magnifying Glass

Toolbar Icon: 

Bind Key: M

Select this command or click the toolbar icon and then move the mouse in the drawing area. A squared window (magnifying window) appears with the schematics magnified inside the window, as illustrated in the following figure:

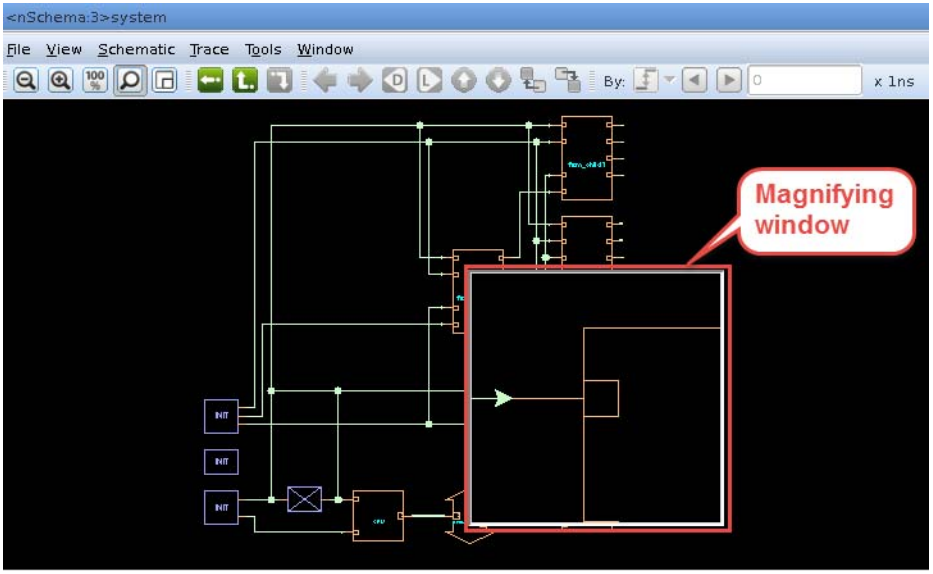


Figure: Magnifying Window

The magnifying window moves with the mouse. The size and ratio of the magnifying window can be modified by invoking the T**ools - > Preferences -> Schematics -> Display Options -> View** page. For more information, refer to the *View Page* section.

Minimap

Menu Bar: View -> Zoom -> Minimap

Toolbar Icon: 

This command provides a concise view of the complete schematic in the small window created for minimap. The minimap window is docked by default in the active *nSchema* window, as illustrated in the following figure:

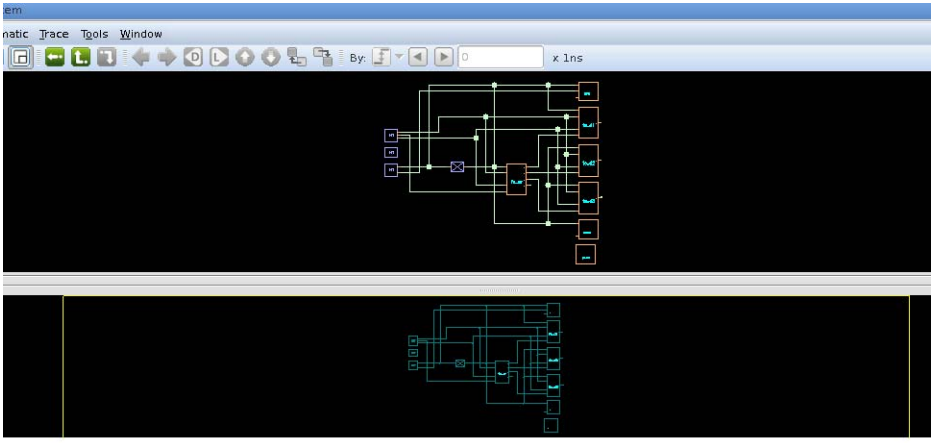


Figure: Minimap Window

Click the **Undock** icon to display the minimap window as a standalone window, as illustrated in the following figure:

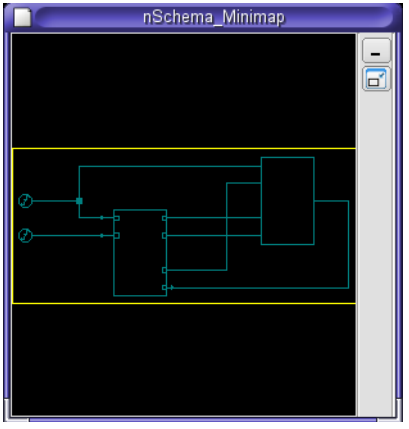


Figure: Undocked Minimap Window

In the minimap window, a bounding box is available to change the scale and position of the minimap window, as illustrated in the following figure.

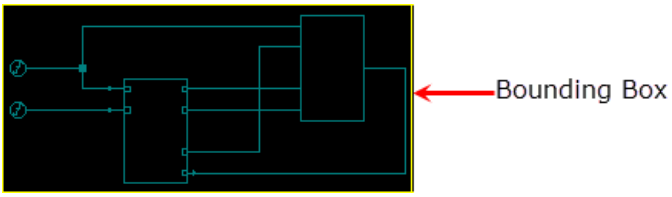


Figure: Bounding Box

If any change occurs in the active *nSchema* window, for example, zoom in or zoom out, then the minimap bounding box updates the scale to display the current scale. If you drag or move the bounding box in the minimap window, then *nSchema* window view is updated accordingly to match the minimap window view. By default, the **Minimap** option is turned off

---

NOTE: Only one minimap window appears for one active *nSchema* window.

Fit Select Set

Menu Bar: View -> Zoom -> Fit Select Set

Bind Key: Ctrl+F

This command fits the selected objects in the schematic window.

Pan

Toolbar Icon: 

Click this toolbar icon to pan the *nEco* pane to the left, right, up, or down.

Pan Left

Menu Bar: View -> Pan -> Pan Left

Bind Key: Left Arrow

This command pans the schematic window to the left.

Pan Right

Menu Bar: View -> Pan -> Pan Right

Bind Key: Right Arrow

This command pans the schematic window to the right.

NOTE: Panning left and right can also be achieved by moving the horizontal scroll bar in the *nSchema* window.

Pan Up

Menu Bar: View -> Pan -> Pan Up

Bind Key: Up Arrow

This command pans the schematic window up.

Pan Down

Menu Bar: View -> Pan -> Pan Down

Bind Key: Down Arrow

This command pans the schematic window down.

NOTE: Panning up and down can also be achieved by moving the vertical scroll bar in the *nSchema* window.

Pop View Up from Port

Menu Bar: View -> Pop View Up from Port

NOTE: This command is not available in a *Flattened* window.

This command opens the *Pop View Up from Port* form, as illustrated in the following figure. Select a port or instance port from the list to switch to the upper hierarchy.

Figure: Pop View Up From Port Form

Pop View Up

Menu Bar: View -> Pop View Up

Toolbar Icon: 

Bind Key: Shift +P

NOTE: This command is not available in a *Flattened* window.

This command generates the parent hierarchical level schematic.

Push View In

Menu Bar: View -> Push View In

Toolbar Icon: 

Bind Key: L

NOTE: This command is not available in a *Flattened* window.

After selecting a module, invoke this command to generate the child hierarchical level schematic.

Last View

Menu Bar: View -> Last View

Toolbar Icon: 

Bind Key: L

This command returns the schematic to the view associated with the last invoked viewing command and only saves the last view. When this command is invoked more than once, it toggles between the current and last views.

Schematic Commands

Find Signal/Instance/Instport

Menu Bar: Schematic -> Find Signal/Instance/Instport

Bind Key: Shift+A

This command opens the *Find Signals* form where a signal, instance, or instance port can be located in the *nSchema* window.

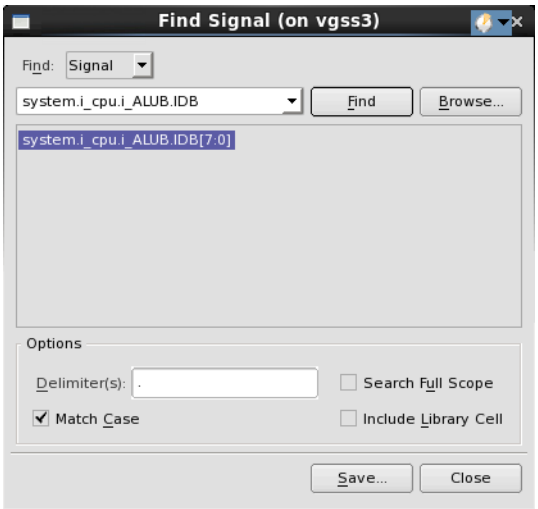


Figure: Find Signals Form

The *Find Signals* form includes the following fields, options, and buttons:

- **Find**: Specify the type of object to search by selecting one of the following options: **Signal**, **Instance**, or **Instport**. The default option is *Signal*.

  Enter the hierarchical name of the signal or the instance in the associated **Find** text field.

  Select **Ctrl+A** key to select all the signals in the *Find* Form. Select **Ctrl+W** key to add all the selected signals at once in the *Find Signal* form. The signals are then added to the *nWave* window. The minimum signals which can be added is *1000* which is the default value. The default value for the signals which can be added in the *Find Signal* form can be changed using

the **Get Signals -> Options form -> Others tab -> Ask if Signals Over** option.

This text field supports the following regular expressions:

Syntax	Expression	Description
? (question mark)	Any character	Matches any one character.
* (asterisk)	Zero or more	Finds zero or more occurrences of the preceding expression.
| (pipe)	OR	Matches the expression before or after the pipe sign "|". It is mostly used in a group. For example, "[sponge]|[mud] bath" matches "sponge bath", and "mud bath".
+ (plus)	One or more	Matches at least one occurrence of the preceding expression. For example, "ba+" matches "ba", "baa", "baaa" and so on.
~ (tilde)	Prevent match	Prevents a match on 'X' at the point it appears in the expression. For example, "real~[ity]" matches the "real" in "really", but not the "real" in "reality".

- Click **Browse** to browse the signals in the current scope and open the *Browse Signal* form as illustrated in the following figure:

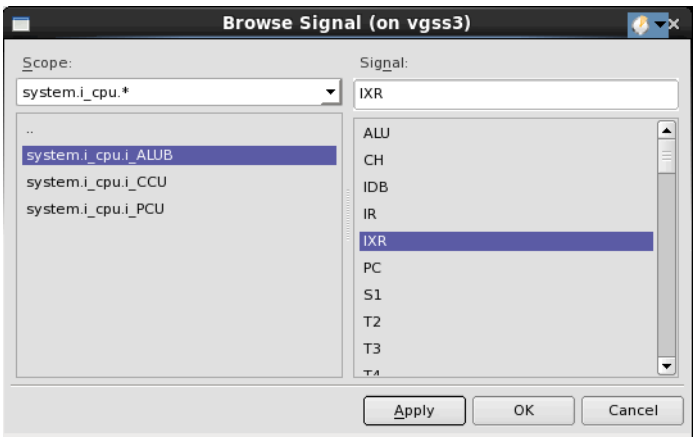


Figure: Browse Signal Form

Click any of the scopes in the **Scope** list and the corresponding signals are listed in the **Signal** list. Select a required signal or select **Ctrl+A** key to select all the signals from the **Signal** list and then click **OK**. This signal's

hierarchical name is copied into the **Find** text field on the *Find Signals* form.

Select **Ctrl+W** key to add all the selected signals at once. The signals are then added to the *nWave* window. The minimum signals which can be added is *1000* which is the default value. The default value for the signals which can be added in the *Find Signal* form can be changed using the **Get Signals -> Options form -> Others tab -> Ask if Signals Over** option.

Click **Find** to display the results of the defined object in the schematic window.

- **Delimiter**: Enter the delimiter value in this text field. The default option is the period (.) character.
- **Match Case**: When this option is turned *on*, *nSchema* finds only the instances in which the capitalization matches the text entered in the **Find** text field.
- **Search Full Scope**: When this option is turned *on*, the signal search is performed throughout the hierarchy regardless of the scope (hierarchy) defined in the **Signal** text field. When this option is turned *off*, the search is only performed in the scope defined in the **Signal** text field on the *Find Signals* form. The default value is *off*.
- **Include Library Cell**: When this option is turned *on*, *nSchema* includes library cells in the search.The default value is *off*.

NOTE: The found signals/instances/instports can be dragged to the *nTrace*/*nWave*/*nSchema* window.

Find in Current Scope

Menu Bar: Schematic -> Find in Current Scope

Bind Key: A

This command opens the *Find* form which lists all signals, instances, instance ports (instport), ports, or modules in the *nSchema* window. The signal, instance, instport, port, or module selected in the *Find* form are also highlighted in the *nSchema* window.

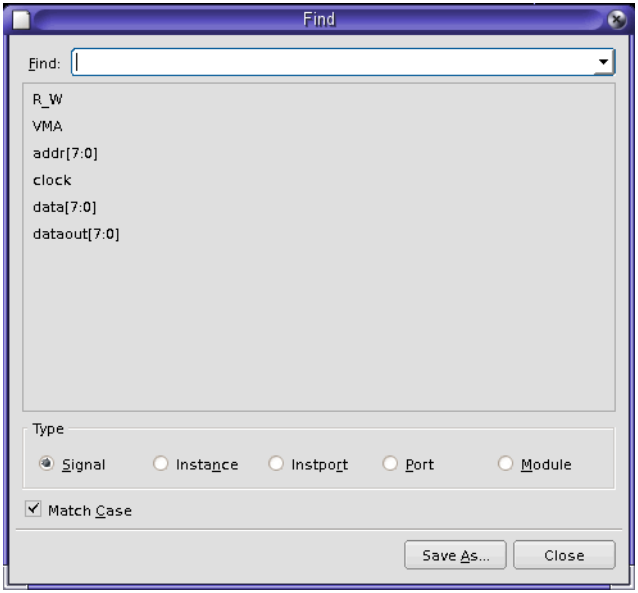


Figure: Find Form

The *Find* form includes the following fields and options:

- **Find**: In this text field, enter the name of the desired signal, instance, instport, port, or module to locate the signals.
- **Type**: Select one of the following options to specify the object type of interest **Signal**, **Instance**, **Instport**, **Port**, or **Module**. The default option is *Signal*.
- **Match Case**: When this option is turned *on*, search is case-sensitive. The default value is *on*.

Find Locator

Menu Bar: Schematic -> Find Locator

This command opens the *Find Locator* form where the signal, instance, instance ports (instport), or ports, can be searched. The signal, instance, instports, or ports selected in the *Find Locator* form are also highlighted in the *nSchema* window.

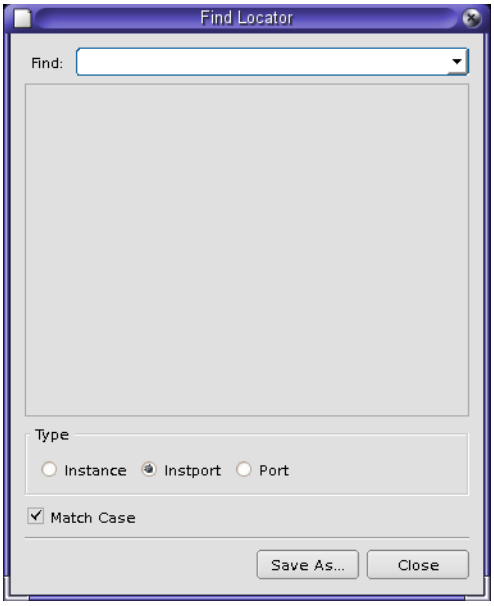


Figure: Find Locator

The Find Locator form includes the following fields and options:

- **Find:** In this text field, enter the name of the required instance, instport, or port, to search the locator.
- **Type:** Select one of the following options to specify the object type of interest, **Signal**, **Instance**, **Instport**, or **Port**. The default option is *Insport*.
- **Match Case:** When this option is turned on, search is case-sensitive. The default value is *on*.

Auto Fit Found Object(s)

Menu Bar: Schematic -> Auto Fit Found Object(s)

This command fits the selected set in the *nSchema* window automatically. This option applies to the **Drag-and-Drop**, **Schematic -> Find**, and **Tools -> Options -> Sync. Signal Selection** commands.

Selection

Select All Objects

Menu Bar: Schematic -> Selection -> Select All Objects

Bind Key: Ctrl+A

This command selects all the objects in the schematic window.

Select Instances/Signals

Menu Bar: Schematic -> Selection -> Select Instances/Signals

This command opens the *Select Instances/Signals* form in which instance(s) and/or signal(s) can be selected in the schematic window. The form lists all of the instances or signals in the current module in alphabetical order.

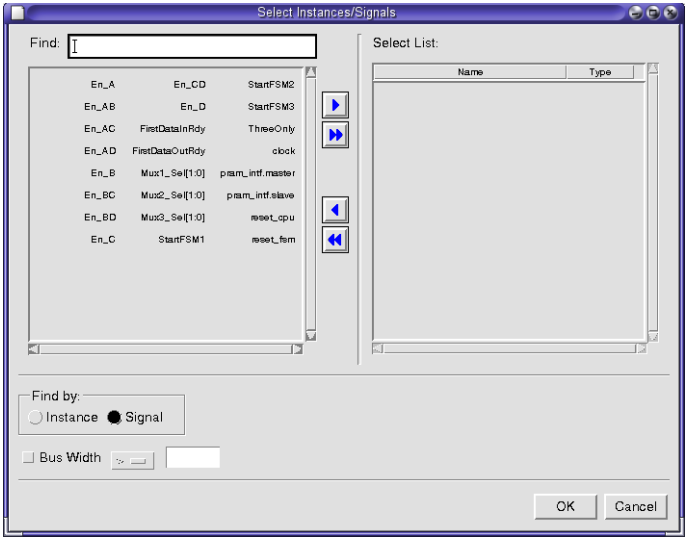


Figure: Select Instances/Signals Form

This *Select Instances/Signals* form includes the following fields, options, and buttons:

- **Find by**: Select either the **Instance** or **Signal** options to specify the required object type. The default option is *Signal*.

- **Find**: Type an instance or signal name in this text field and press enter to refine the object list. The wildcard character (*) is supported.
- **Add Object** : Click this button to move the selected object from the left pane to the **Select List**.
- **Add All** : Click this button to move all objects in the left pane to the **Select List**.
- **Remove Object** : Click this button to move the selected object from the **Select List** to the left pane.
- **Remove All** Click this button to move all objects in the **Select List** to the left pane.
- **Bus Width**: When this option is turned on, specify the bus width criteria by selecting from one of the conditional operators: greater than (>), greater than or equal to (>=), equal to (=), less than or equal to (<=), or less than (<). This conditional operator is combined with the number of bits specified. When **Enter** is pressed, the buses meeting the criteria are displayed in the left pane.

Select All Instances

Menu Bar:	Schematic -> Selection -> Select All Instances
Bind Key:	I

This command selects all instances in the schematic window.

Select All Signals

Menu Bar:	Schematic -> Selection -> Select All Signals
Bind Key:	S

This command selects all signals in the schematic window.

Select All Pins

Menu Bar:	Schematic -> Selection -> Select All Pins

This command selects all pins in the schematic window.

Deselect All

Menu Bar: Schematic -> Selection -> Deselect All

This command disables all objects in the schematic window.

Focus Connection

Menu Bar: Schematic -> Focus Connection

Mouse Action: Double-click the signal in the schematic window

This command highlights the connection (driving instance to loading instances(s) with the connecting net) for the selected signal in the current schematic window. Use *n* (previous) or *N* (next) to browse the connecting instances and net. This command is available in all schematic window types (full hierarchical, partial hierarchical, and flattened).

Highlight Incomplete Net

Menu Bar: Schematic -> Highlight Incomplete Net

Bind Key: I

This option enables the display of all the incomplete nets in the current window by highlighting them as dash-lines.

Add to Waveform

Menu Bar: Schematic -> Add to Waveform

This command adds the selected signal(s) in the *nSchema* window to the signal pane in *nWave* (the signals are added at the current cursor position). Alternatively, the selected signals can be dragged from the *nSchema* window to the *nWave* window.

If the *nSchema* window is displaying an RTL module, the nets associated with the IO of the instance are used in *nWave*. If the *nSchema* window is displaying a gate module and the **Show Port Name for Dropped Instance** option under **Tools -> Preferences -> Waveform** page **-> View Options** page **-> Miscellaneous Page** is turned *on*, the pin names for the instance IO are used in

nWave. If a pin name cannot be found or the **Show Port Name for Dropped Instance** option is turned *off*, the associated net is used.

NOTE: An FSDB file must be loaded into the Verdi platform (for example, with the **File -> Open Waveform File** command in *nTrace*) before invoking this command.

The following are the sub-commands of the **Add to Waveform** command.

New Waveform

This command enables you to create a new waveform window, open an annotation file, and add signals.

Add to Wave[n]

Bind Key Ctrl+4

This command enables you to add selected objects to the last adding signal waveform or the new created waveform.

Add to New Group

This command creates a new group in the waveform and adds selected objects to this group.

*Wave[k]

Bind Key Ctrl+W

This command adds selected objects to the waveform window. For example, *Wave[k], k is the primary waveform window Id.

Wave[m]

This command adds selected objects to waveform window. For example, Wave[m], m is the window id.

Add Locator

Menu Bar: Schematic -> Add Locator

Bind Key: Ctrl+Shift+O

This command opens the *Add Locator* form in which a locator can be added for any instance, instports, or ports in the *nSchema* window for the selected object as illustrated in the following figure. You can also select the icons available in the **Icon Type** drop-down list to add the icon to the locator.

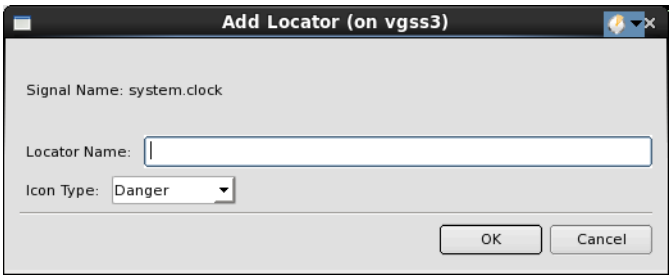


Figure: Add Locator

Change Color

Menu Bar: Schematic -> Change Color

Bind Key: C

This command opens the *Change Selection Color* form in which the color of the selected signal can be changed.

This form displays the selected signal/instance with its current color in the color plate. The color can be changed instantly by clicking another color on the color plate. Set the selected signal color as the default by enabling the **Default** option. The new color is inherited when these signals are dropped in the *nWave* window.

NOTE: A quick way to change a signal's color within a predefined color set is to select the signal, then continue pressing the (**T**) key to sequence through the color set until the preferred color is reached.

Figure: Change Selection Color Form

Color Instances by Scope (Flattened View Only)

Menu Bar: Schematic -> Color Instances by Scope

NOTE: This command is available in a *Flattened* window.

This command opens the *Auto Highlight Instances* form in which instance colors can be customized by scope in a *Flattened* window. All scopes are displayed in the **Scopes** table of the *Auto Highlight Instances* form. After selecting the preferred scope, change the color for any of the listed scopes.

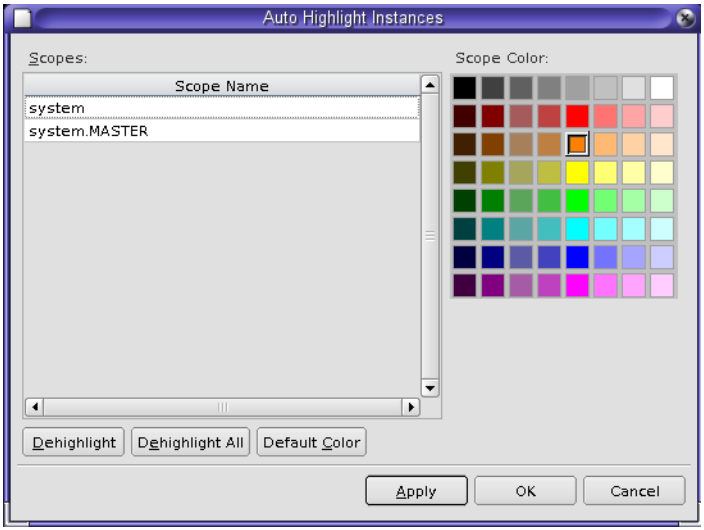


Figure: Auto Highlight Instances Form

The *Auto Highlight Instances* form includes the following buttons and field:

- **Scopes:** This table displays all scopes traversed in the current *Flattened* window.
- **Scope Color:** Specify the desired color for each scope in this color plate.
- **Dehighlight:** Click this button to remove the highlight from the selected scope. The change does not take effect until **Apply** or **OK** is also clicked.
- **Dehighlight All:** Click this button to remove the highlights from all scopes in all schematic windows. The change does not take effect until **Apply** or **OK** is also clicked.
- **Default Color:** Click this button to automatically assign a different color to each scope. 16 colors are available (ID_GRAY3, ID_RED3, ID_ORANGE3, ID_YELLOW3, ID_GREEN3, ID_CYAN3, ID_BLUE2, ID_PURPLE2, ID_GRAY6, ID_RED6, ID_ORANGE6, ID_YELLOW6, ID_GREEN6, ID_CYAN6, ID_BLUE6, ID_PURPLE6, ID_UNDEFINED_COLOR) for automatic assignment. If more than 16 scopes exist, the colors are repeated. Any previously assigned colors are overwritten by clicking this button. The change does not take effect until the **Apply** or **OK** is also clicked.

All Objects to Default Color

Menu Bar: Schematic -> All Objects to Default Color

This command returns all signals to their default colors.

Active Annotation

Menu Bar: Schematic -> Active Annotation

Bind Key: X

This toggle command is enabled after a simulation results file is loaded either from *nTrace* or *nWave*. When **Active Annotation** is *on*, the signal values appear in *nSchema* and change simultaneously when the current simulation time changes. Refer to the **Active Annotation** command in the *nTrace* chapter for details.

In Power Aware mode (when CPF/UPF files are loaded) with the **Annotate Power Aware Design Object(s)** option turned *on* (located on the **General -> Power** page of the *Preferences* form), turn this toggle command *on* to annotate the power state on instances. The power state changes simultaneously when the current simulation time changes. If an object state is *off*, the object is disabled.

Annotate in Color

Menu Bar: Schematic -> Annotate in Color

This toggle command turns the annotation style and line coloring *on* or *off*.

The definitions of the value types are as follows:

- 1: logic high
- 0: logic low
- z: high impedance
- x: unknown

To change the color or the line style associated with annotations, invoke the **Tools -> Preferences** command in the *nTrace* window to open the *Preferences* form, then select the **Schematics** page -> **Color** page. In **Type** drop-down list, select the list of objects to specify the color and line style, including the annotation line coloring for *1*, *0*, *x*, and *z*.

Leading Zeros

Menu Bar: Schematic -> Leading Zeros

This toggle command turns the leading zero display *on* or *off*. When this toggle command is *on*, leading zeros are displayed for signal values in the schematic window when **Active Annotation** is enabled. The number of leading zeros depends on the format set for the signal value.

To ensure leading zeros are always enabled in the future, turn *on* the **Leading Zeros** option on the **Schematics -> Viewing** page of the *Preferences* form.

Signal Value Radix

Menu Bar: Schematic -> Signal Value Radix

After the required signal is selected (multiple selections are supported) in the *nSchema* window, invoke the command to display the value radix of the selected signal.

NOTE: This command is available after a simulation results file is loaded in the *nTrace* window using the **File -> Open Waveform File** command and the **Schematic -> Active Annotation** command is invoked in the *nSchema* window.

Binary

Menu Bar: Schematic -> Signal Value Radix -> Binary

This command displays the signal value in **Binary** format.

Octal

Menu Bar: Schematic -> Signal Value Radix -> Octal

This command displays the signal value in **Octal** format.

Hexadecimal

Menu Bar: Schematic -> Signal Value Radix -> Hexadecimal

This command displays the signal value in **Hexadecimal** format.

Decimal

Menu Bar: Schematic -> Signal Value Radix -> Decimal

This command displays the signal value in **Decimal** format.

ASCII

Menu Bar: Schematic -> Signal Value Radix -> ASCII

This command displays the signal value in **ASCII** format.

Add Alias from File

Menu Bar: Schematic -> Signal Value Radix -> Add Alias from File

Refer to the **Waveform -> Signal Value Radix -> Add Alias From File** command of the *nWave* chapter for details.

Add Alias from Program

Menu Bar: Schematic -> Signal Value Radix -> Add Alias from Program

Refer to the **Waveform -> Signal Value Radix -> Add Alias From Program** command of the *nWave* chapter for details.

Remove Alias

Menu Bar: Schematic -> Signal Value Radix -> Remove Alias

Refer to the **Waveform -> Signal Value Radix -> Remove Alias** command of the *nWave* chapter for details.

Signal Value Notation

After a desired signal is selected (multiple selections are allowed) in the *nSchema*, invoke this command for the signal value notation of the selected signal.

NOTE: This command is available after a simulation results file is loaded in the *nTrace* window using the **File -> Open Waveform File** command and the **Schematic -> Active Annotation** command is invoked in the *nSchema* window.

Unsigned

Menu Bar: Schematic -> Signal Value Notation -> Unsigned

Refer to the **Waveform -> Signal Value Notation -> Unsigned** command description in the *nWave* chapter for details.

Signed 2's Complement

Menu Bar: Schematic -> Signal Value Notation -> Signed 2's Complement

Refer to the **Waveform -> Signal Value Notation -> Signed 2's Complement** command description in the *nWave* chapter for details.

Signed 1's Complement

Menu Bar: Schematic -> Signal Value Notation -> Signed 1's Complement

Refer to the **Waveform -> Signal Value Notation -> Signed 1's Complement** command description in the *nWave* chapter for details.

Signed Magnitude

Menu Bar: Schematic -> Signal Value Notation -> Signed Magnitude

Refer to the **Waveform -> Signal Value Notation -> Signed Magnitude** command description in the *nWave* chapter for details.

Propagated Value

Menu Bar: Schematic -> Propagated Value

NOTE: This command is automatically enabled after value propagation is performed with the **Trace -> Value Propagation -> Value Propagation** command.

When the **Propagated Value** toggle command is turned *on*, propagated values are displayed according to the value propagation settings in the **Trace -> Value Propagation -> Value Propagation** command. If the **Schematic -> Active Annotation** command is *on*, the value display is $FSDBValue / PV : $PropagatedValue; if the **Schematic -> Active Annotation** command is *off*, the value display is PV : $PropagatedValue. If propagated values do not exist for the signal, the displayed value is "NF".

Constant Net

Menu Bar: Schematic -> Constant Net

NOTE: This command is automatically enabled after value propagation is performed from the **Trace -> Value Propagation -> Value Propagation** command.

When the **Constant Net** command is turned *on*, nets with propagated value as 0 or 1 are highlighted. The default for nets with 1 have the value is *red*; for nets with 0 have the value is *blue*.

The net color can be set in **Constant Net 0**/**Constant Net 1** from **Type** on the **Tools -> Preferences -> Schematics** page **-> Color/Font Page**.

Set Annotation Time Range

Menu Bar: Schematic -> Set Annotation Time Range

NOTE: This command is available after an AMS design is loaded and the **Schematic -> Active Annotation** toggle command is turned *on*.

This command opens the *Set Annotation Time Range* form where the **From Time** and **End Time** fields can be set for the annotation time range.

When the **Sync with Cursor/Marker** option is enabled, the **From Time** and **End Time** fields are disabled and Verdi uses the cursor or marker of the primary waveform as the from time and end time. If the waveform window is closed after the option is enabled, Verdi uses the latest cursor or marker value of waveform as the time range.

Figure: Set Annotation Time Range Form

Signal Current Value Type

Menu Bar: Schematic -> Signal Current Value Type

This command displays different values for current based on the defining value selected. Only one of the defining values can be active at a time. You can select from the following values:

- Cursor Value
- Average Value

The display format of annotation is: **value(format)**, as illustrated in the following figure:

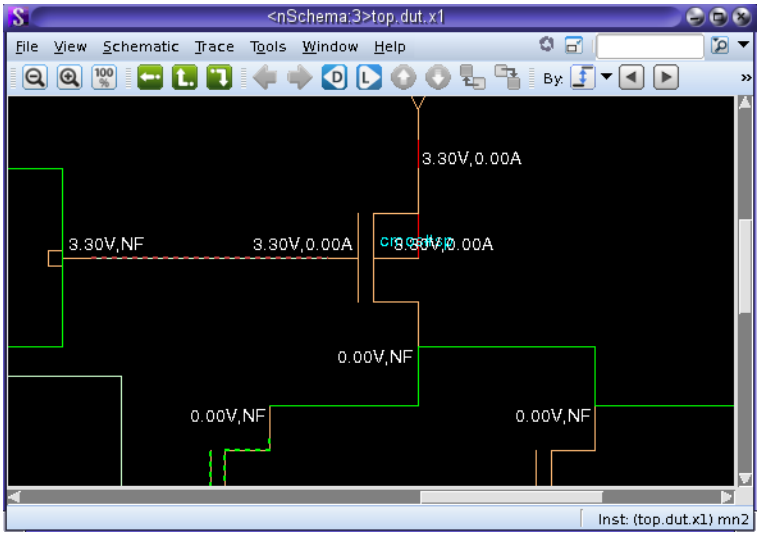


Figure: Display Format of Annotation

Both right-click menu options and pull-down menu status are updated to reflect the value type of selected signal. If there is no signal selected or the selected signal is digital scope signals, both menus are disabled.

If the time range is not defined, a warning message appears when you click on any defining value type (except cursor value).

This command is only enabled when the selected signal is in analog scope.

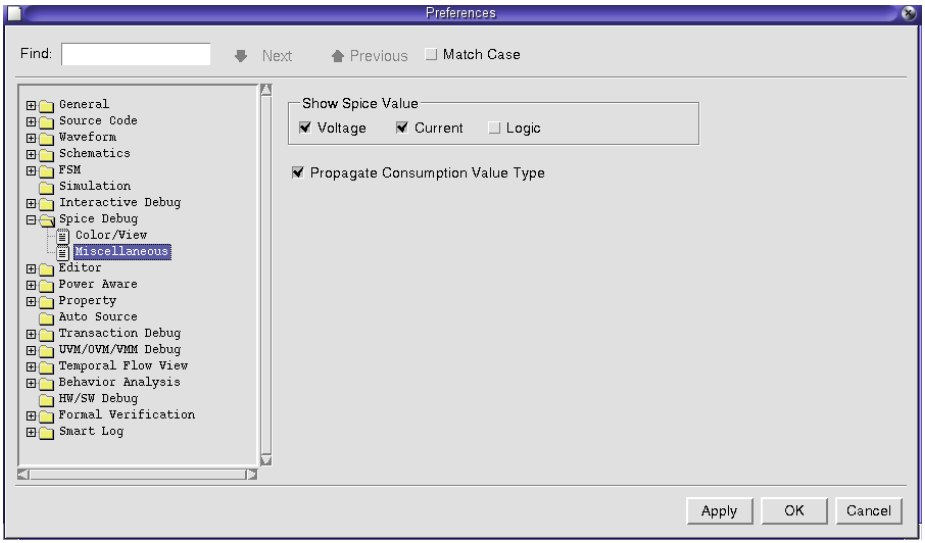


Figure: Preference Setting for Signal Current

If **Voltage** is deselected in the preference setting, the **Signal Voltage Value Type** option is disabled; If **Current** is deselected in the preference setting, the **Signal Current Value Type** is disabled.

If the option **Propagate Consumption Value Type** is toggled, the selected signal's value type propagates to all its analog scope same nets and a2a through nets in digital scope. Otherwise, only the selected signal's value type changes. The option is toggled by default.

Cursor Value

Menu Bar: Schematic -> Signal Current Value Type -> Cursor Value

Displays the cursor value of the selected signal. This is default defining value.

Average Value

Menu Bar: Schematic -> Signal Current Value Type -> Average Value

Displays the average value of the selected signal.

Signal Voltage Value Type

Menu Bar: Schematic -> Signal Voltage Value Type

Displays different values for voltage based on the defining value selected. Only one of the defining values can be active at a time. You can select from the following values:

- Cursor Value
- Average Value

The display format of annotation is: **value(format),** as illustrated in the following figure:

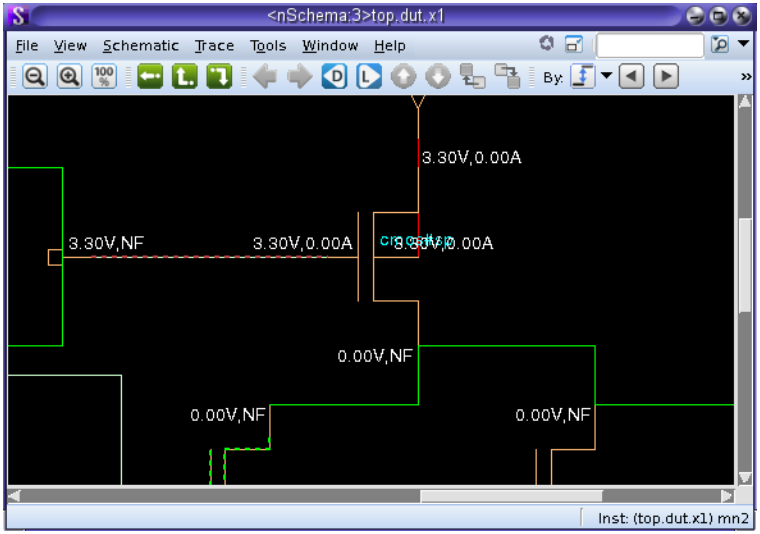


Figure: Display Format for Annotation

Both right-click menu options and pull-down menu status are updated to reflect the value type of the selected signal. If the signal is not selected or the selected signal is digital scope signals, both menus are disabled.

If the time range is not defined, then a warning message appears when you click on any defining value type (except cursor value).

This command is only enabled when the selected signal is in analog scope.

If **Voltage** is disabled in the preference setting, the **Signal Voltage Value Type** option is disabled; If **Current** is disabled in the preference setting, the **Signal Current Value Type** is disabled.

If the option **Propagate Consumption Value** Type is toggled, the selected signal's value type propagates to all its analog scope same nets and a2a through nets in digital scope. Otherwise, only the selected signal's value type changes. The option is toggled by default.

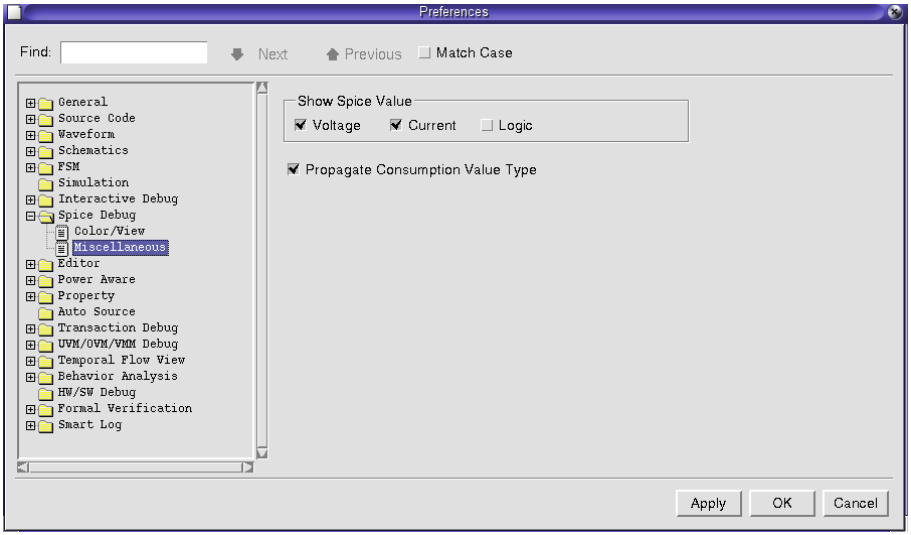


Figure: Preference Setting for Voltage Current

Cursor Value

Menu Bar: Schematic -> Signal Voltage Value Type -> Cursor Value

This command displays the cursor value of the selected signal. This is the default defining value.

Average Value

Menu Bar: Schematic -> Signal Voltage Value Type -> Average Value

This command displays the average value of the selected signal.

SDF Annotation - nAnalyzer

Menu Bar: Schematic -> SDF Annotation

When this command is enabled and an SDF file contains INTERCONNECT delay statements, SDF annotation is displayed in the schematic window. An SDF file must be loaded from the **File -> SDF -> Load SDF Files - nAnalyzer** command in the *nTrace* window before this command is activated. The insertion delay is annotated on each net in which the number before the slash is rising interconnect delay and the number after the slash is falling interconnect delay.

NOTE: The top level of the design must match the top level of the SDF file for annotation to be displayed.

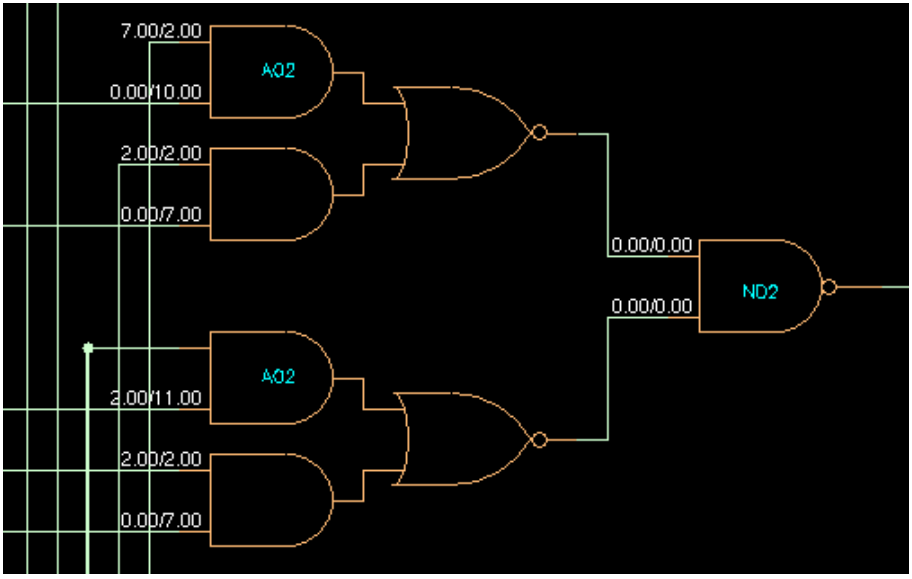


Figure: SDF Annotation Displayed in nSchema

Cell Delay - nAnalyzer

Menu Bar: Schematic -> Cell Delay

To view cell delays (IO Path delays), select a cell, invoke this command, and *nSchema* displays a message listing all IO Path_delays associated with this cell.

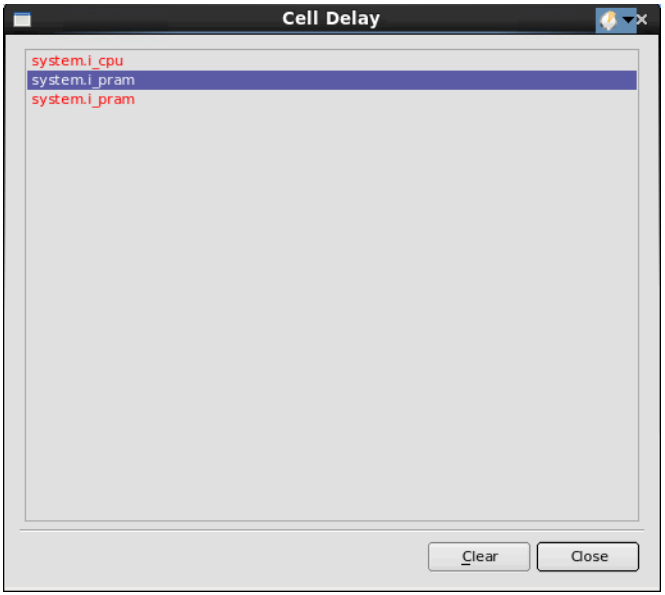


Figure: Cell Delay Window

Click **Clear** to clear the contents in this window. When **Show Cell Delay** is executed again, the results are appended following the current contents, if the window is not cleared.

Delay Scale - nAnalyzer

Menu Bar: Schematic -> Delay Scale

Specify the SDF **Delay Scale** by selecting **Default, 10fs, 100fs, 1ps, 10ps, 100ps, 1ns,** or **10ns** option. The selected scale is displayed in the menu command. The default setting is **10ps**.

Delay Type - nAnalyzer

Menu Bar: Schematic -> Delay Type

Specify the SDF **Delay Type** by selecting **Minimum**, **Typical**, or **Maximum** option**.** The selected type is displayed in the menu command. The default setting is **Typical**.

Delay Precision - nAnalyzer

Menu Bar: Schematic -> Delay Precision

Specify the SDF **Delay Precision** by selecting **0.1**, **0.01**, or **0.001** option. The selected precision is displayed in the menu command. The default setting is **0.01**.

ViewMark (Hierarchical View Only)

Menu Bar: Schematic -> ViewMark

NOTE: This command is available in a *Hierarchical View* window.

This command marks the view of a schematic to quickly access the marked viewing area. The scope and viewing area are kept in the **Schematic -> Recent Schematics** command. The three most recent ViewMarks are cached by the **ViewMark** command.

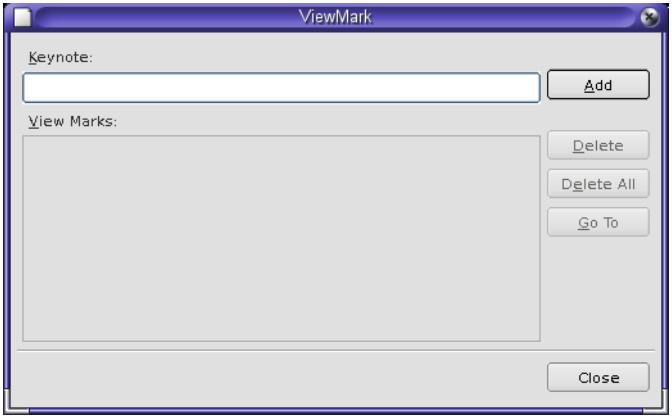


Figure: ViewMark Form

Sync. Active Scope

Menu Bar: Schematic -> Sync. Active Scope

This toggle command enables/disables the synchronization between the active scope in the design browser frame and current *nSchema* window. When this command is enabled (checked), the active scope in the full hierarchical *nSchema*

window is synchronized with the design browser frame. By default, it is disabled (unchecked). Only one *nSchema* window can be actively synchronized at a time.

Recent Schematics

Menu Bar: Schematic -> Recent Schematics

NOTE: This command is available in a *Hierarchical View* window.

This command displays the list of recently viewed schematics in the *nSchema* window.

Trace Commands

Signal selection in the windows can be synchronized by enabling the **View -> Sync. All Signal Selection** command in *nTrace*. If a signal in a waveform, source code, or schematic frame is selected, the corresponding signals in the other two frames are also selected.

The *nSchema* window supports a number of different views: full hierarchy, partial flattened, and partial hierarchy. The trace capability differs in these views. The following commands explains how to invoke different views in the *nSchema* window and the trace capabilities that are supported in each view.

NOTE: The **Trace Driver** command is not bound to the double-click mouse action on a net.

These commands are not available in a *Browser* window.

Driver

Menu Bar:	Trace -> Driver
Toolbar Icon:	
Bind Key	Alt+Shift+D

This command traces all possible drivers of a selected signal and displays the results in the current *nSchema* window.

Load

Menu Bar:	Trace -> Load
Toolbar Icon:	
Bind Key	Alt+Shift+L

This command traces all possible loads of a selected signal and displays the results in the current *nSchema* window.

Connectivity

Menu Bar: Trace -> Connectivity

Toolbar Icon: 

Bind Key Alt+Shift+C

This command traces all possible loads and drivers of a selected signal and displays the results in the current *nSchema* window. This command combines the functions of the **Trace Driver** and **Trace Load** commands

Fan-in

Menu Bar: Trace -> Fan-in

Bind Key Alt+Shift+I

This command extracts the fan-in (that is, trace in the driver's direction until an IO, storage element, or FSM is reached) of the selected signal and appends the result to the schematic window according to the settings defined by the **Trace -> Options -> Trace Stop on Module Boundaries** or **Trace -> Options -> Stop Cone Tracing on FSM** commands.

Fan-out

Menu Bar: Trace -> Fan-out

This command extracts the fan-out (that is, trace in the load's direction until an IO, storage element, or FSM is reached) of the selected signal and appends the result to the schematic window according to the settings defined by the **Trace -> Options -> Trace Stop on Module Boundaries** or **Trace -> Options -> Stop Cone Tracing on FSM** commands.

Active Fan-in

Menu Bar: Trace -> Active Fan-in

This command extracts the active fan-in of the selected signal and appends the result to a flat partial schematic window. An FSDB file must be loaded into the

Verdi platform (for example, with the **File -> Load Simulation Results** command in *nTrace*) before invoking this command.

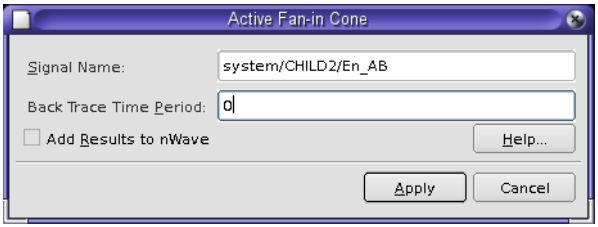


Figure: Active Fan-in Cone Form

The *Active Fan-in Cone* form includes the following fields, options, and buttons:

- **Signal Name:** Displays the name of the signal selected in the *nSchema* window.
- **Back Trace Time Period:** Specify a time period for back trace time.
- **Add Results To nWave:** When this option is turned *on*, the results are added to the *nWave* window. This is equivalent to **Tools -> New Schematic -> Active Fan-in** command in *nWave*.
- **Apply:** Click **Apply** to display the portion of the fan-in cone that is active (those elements/signals in the fan-in cone that changed value) during the **Back Trace Time Period**. The results are a subset of the full fan-in cone that are viewed if the **Tools -> New Schematic from Source -> Fan-in** command is used. The resulting schematic displays the logic pertinent to the transition selected, as opposed to displaying the entire fan-in cone.

Bus Trace

The following are the sub-commands of the **Bus Trace** command.

Driver

Menu Bar: Trace -> Driver

NOTE: A bus signal must be selected to enable this command.

This command extracts the drivers from the index range of the selected bus and displays the results in the current *nSchema* window. The index range of the selected bus is specified in the *Specify Range* form, which is displayed after

selecting a bus in the schematic window and selecting the **Bus Driver** command. If the selected signal is not a bus, a warning message is displayed.

Load

Menu Bar: Trace -> Load

NOTE: A bus signal must be selected to enable this command.

This command extracts the loads from the index range of the selected bus and displays the results in the current *nSchema* window. The index range of the selected bus is specified in the *Specify Range* form, which is displayed after selecting a bus in the schematic window and invoking the **Bus Load** command. If the selected signal is not a bus, a warning message is displayed.

Connectivity

Menu Bar: Trace -> Connectivity

NOTE: A bus signal must be selected to enable this command.

This command extracts the connectivity from the index range of the selected bus and displays the results in the current *nSchema* window. The index range of the selected bus is specified in the *Specify Range* form, which is displayed after selecting a bus in the schematic window and selecting the **Bus Connectivity** command. If the selected signal is not a bus, a warning message is displayed.

Fan-in

Menu Bar: Trace -> Fan-in

NOTE: A bus signal must be selected to enable this command.

This command is used to select a bus signal, specify the bit range of interest in the form provided, extract the fan-in cone (that is, trace in the driver's direction until an IO, storage element, or FSM is reached), and append the results to the schematic window according to the settings of the **Trace -> Options -> Trace Stop on Module Boundaries** or **Trace -> Options -> Stop Cone Tracing on FSM** commands.

Fan-out

Menu Bar: Trace -> Fan-out

NOTE: A bus signal must be selected to enable this command.

This command is used to select a bus signal, specify the bit range of interest in the form provided, extract the fan-out cone (that is, trace in the load's direction until an IO, storage element, or FSM is reached), and append the results to the schematic window according to the settings of the **Trace -> Options ->** **Trace Stop on Module Boundaries** or **Trace -> Options ->** **Stop Cone Tracing on FSM** commands.

Advanced Trace

Menu Bar: Trace -> Advanced Trace

NOTE: A bus signal/pin must be selected to enable this command.

This command opens the *Advanced Trace Bus* form which can be used to select incontinuous bits to do tracing by using the <**CTRL**> key.

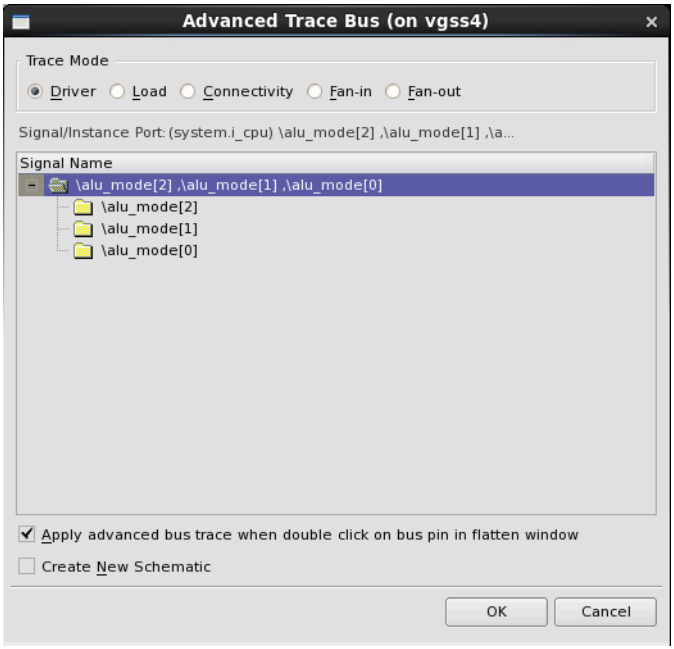


Figure: Advanced Trace Bus Form

The *Advanced Trace Bus* form includes the following options and fields:

- **Trace Mode:** Specify the signal/instance trace type by selecting one of the following options: Driver, Load, Connectivity, Fan-in, or Fan-out. Only one type can be selected at a time.
- **Signal/Instance Port:** This field displays the name of the selected signal/instance port to be traced. To trace another signal/instance, select the signals from the **Signal Name** list to replace the existing signal/instance port.
- **Apply advance bus trace when double click on bus in flatten window:** When you double-click on the input pin in the *Flatten Window* or *Hierarchical Flattened View* window, the trace type is set as **Driver** automatically. When the **Apply advanced bus trace when double click on bus pin in flatten window** is turned *on*, and you double-click on the output pin in the *Flatten Window* or *Hierarchical Flattened View* window, the trace type is set as **Load** automatically.
- **Create Window:** This option opens a new schematic window to display the trace result when the trace is completed. The default option is *off*.

Two Points

Menu Bar: Trace -> Two Points

This command opens the *Trace Two Points* form where criteria to trace between two points can be specified. The points can be either instance ports or signals.

Instance Port Based Trace Mode

You can select **Instance Port Based** for instance ports. The default mode is **Instance Port Based**, as illustrated in the following figure:

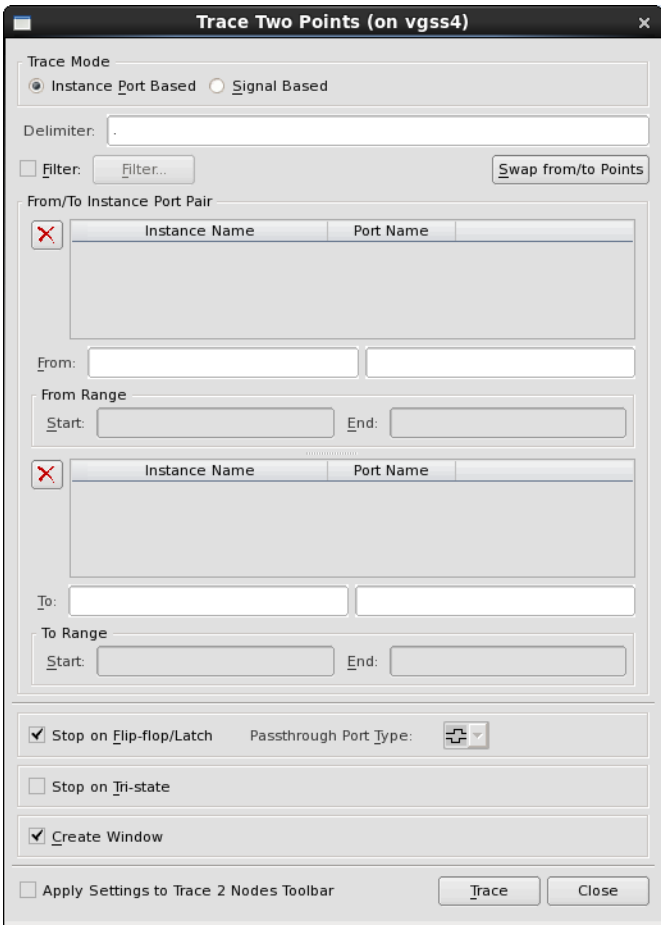


Figure: Trace Two Points Form (Instance Based Port)

The *Trace Two Points* form includes the following fields, options, and buttons:

- **Trace Mode**: Specify the trace mode by selecting from the **Instance Port Based** and **Signal Based** options. The default mode is **Instance Port Based**.
  - **Instance Port Based:** After an instance port name is entered in the **From** and **To** text fields (an object in the *nSchema* or *nTrace* windows can be dragged and then the selected object can be dropped in the **From** and **To** text fields where the instance ports associated with the selected object are listed), click **Trace** and the tracing results are highlighted in the current schematic window.

    If a signal is dragged into either the **From** or **To** text fields, *nSchema* lists all the instance ports in the **From Instance Port** list and automatically selects one of the input instance ports in the **From** text field.
- **Delimiter:** Enter the delimiter in this text field. The default is the period (.) character.
- **Filter:** When this option is turned *on* and the corresponding **Filter** button is clicked, the *Trace Two Points* filter form is opened. Refer to the *Trace Two Points Filter* section for details.
- **Swap from/to Points:** Click this button to switch the **From** details, that is instance ports or signal names to the **To** text field and vice versa.
- **From/To Instance Port Pair** or **From/To Signal Pair:** Display the instance port for tracing. The  icon is used to clear the signal(s) selected in the **From** and **To** text fields.
- **From Range:** Specify the range information (start range and end range) for the start of the instance port or the signal pair for tracing. By default, the option is disabled. If you specify the instance port/signal as a bus signal or connected to a bus signal instance port, then the option is enabled and you can specify the range information.
- **To Range:** Specify the range information (start range and end range) for the end of the instance port or the signal pair for tracing. By default, the option is disabled. If you specify the instance port as a bus signal or connected to a bus signal instance port, then the option is enabled and you can specify the range information.
- **Stop on Flip-flop/Latch:** When this option is turned *on*, tracing stops at a flip-flop boundary and/or a latch when tracing between two points. When this option is turned *off*, tracing passes through data ports. The default option is *on*.

  When the **Stop on Flip-flop/Latch** option is turned *off*, the **Passthrough Port Type**  selection field is activated. Click the upside-down triangle

to turn *on* the selected port type from the **Data**, **Clock**, **control**, **Set/Reset**, and **Other** options (multiple selections are supported). When tracing between two points, the selected port type is passed through and port types that are not selected is stopped. When the **Stop on Flip-flop/Latch** option is turned *on*, the **Passthrough Port Type** selection field is disabled. The default port type is **Data**.

- **Stop on Tri-state**: When this option is turned *on*, tracing stops at a tri-state when tracing between two points. The default option is *on*.
- **Create Window**: When this option is turned *on*, the tracing results is displayed in a new *nSchema* window. The default option is *off*.
- **Apply Settings to Trace 2 Nodes Toolbar:** When this option is turned *on*, the settings in *Trace Two Points* form is automatically applied to **Trace 2 Nodes** toolbar.

Trace Two Points Filter

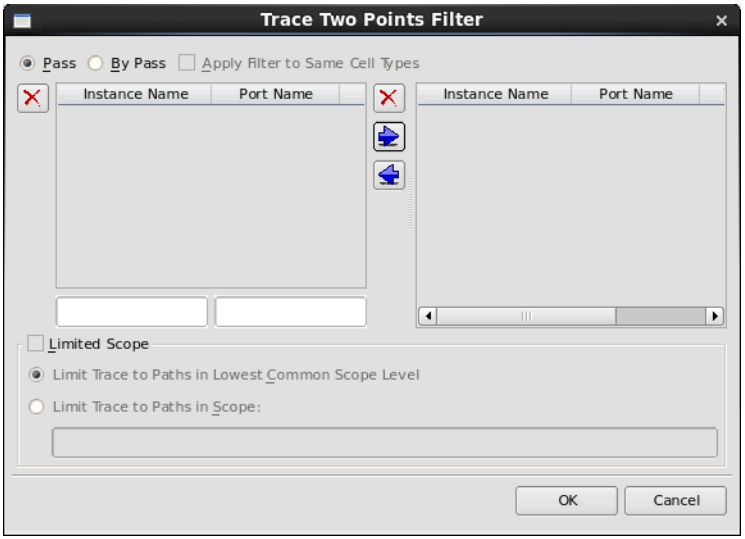


Figure: Trace Two Points Filter Form

After the **Pass** or **By Pass** options are selected, enter the instance port or signal name (depending on which mode, **Instance Port Based** or **Signal Based**, is used) in the text field directly below the left table. For example, to search paths that must skip the output port of *system.i_cpu.i_ALUB.i_alu.U110*, drag the instance from the *nSchema* window to the text field directly below the left table and all ports of *system.i_cpu.i_ALUB.i_alu.U110* are displayed in the left table.

Click **Add**, **Remove**, or **Clear** icons to determine the items which need to be available in the right table for **Pass** or **By Pass**, and then click **OK**.

- **Pass:** Pass through the specified instance port or signal.
- **By Pass:** Do not pass through the specified instance port or signal.
  - **Apply Filter to Same Cell Types:** If this option is *on*, the trace ignores all the instances with the same cell type as the instances specified in this dialog box.
- **Limit Trace to Paths in Lowest Common Scope Level:** Limit trace paths to the lowest common scope between from-point and to-point. For example, when the from-point is *Top.A.B.C.D.portA* and the to-point is *Top.A.B.E.F.G.portB*, *Top.A.B* is the lowest common scope.
- **Limit Trace to Paths in Scope:** Limit the trace to paths in the specified scope. The scope name can be entered manually by entering values in the text field or by dragging a scope from the *nSchema* window and dropping it in the text field.

Signal Based Trace Mode

You can select **Signal Based** for signals, as illustrated in the following figure:

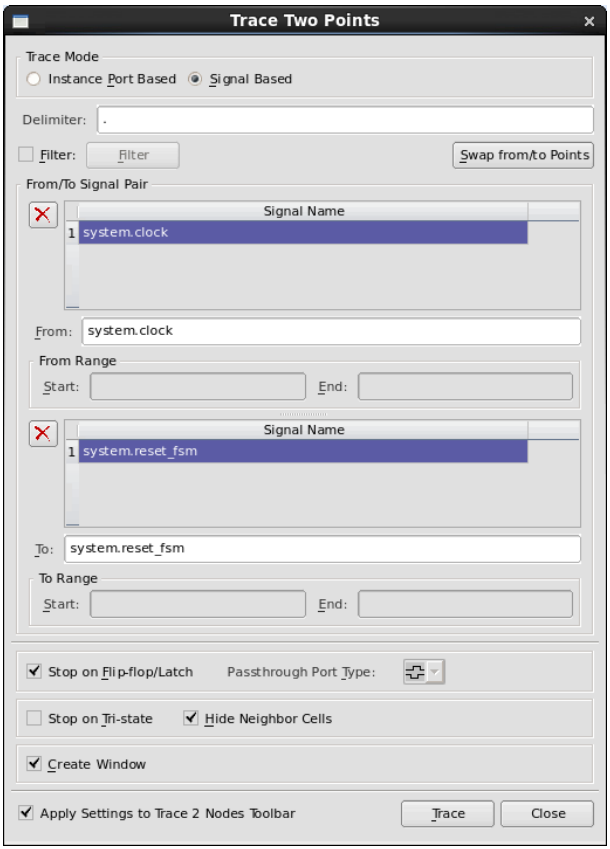


Figure: Trace Two Points (Signal Based)

The *Trace Two Points* form includes the following fields, options, and buttons:

- **Trace Mode**: Specify the trace mode by selecting **Signal Based** option.
  - **Signal Based:** After a signal name is entered in the **From** and **To** text fields (an object in the *nSchema* or *nTrace* windows can be dragged and then the selected object can be dropped in the **From** and **To** text fields where the signals associated with the selected object are listed). Click **Trace** and the tracing results are displayed in a message window. The **Create Window** option can also be enabled to view the trace results in a schematic window.
- **Delimiter:** Enter the delimiter in this text field. The default is the period (.) character.

- **Filter:** When this option is turned *on* and the corresponding **Filter** button is clicked, the *Trace Two Points* filter form is opened. Refer to the *Trace Two Points Filter* section for details.
- **Swap from/to Points:** Click this button to switch the **From** details, that is instance ports or signal names to the **To** text field and vice versa.
- **From/To Signal Pair** or **From/To Signal Pair:** Display the signal pair for tracing. The  icon is used to clear the signal(s) selected in the **From** text field.
- **From Range:** Specify the range information (start range and end range) for the start of the instance port or the signal pair for tracing. By default, the option is disabled. If you specify the signal as a bus signal or connected to a bus signal instance port, then the option is enabled and you can specify the range information.
- **To Range:** Specify the range information (start range and end range) for the end of the instance port or the signal pair for tracing. By default, the option is disabled. If you specify the instance port/signal as a bus signal or connected to a bus signal instance port, then the option is enabled and you can specify the range information.
- **Stop on Flip-flop/Latch:** When this option is turned *on*, tracing stops at a flip-flop boundary and/or a latch when tracing between two points. When this option is turned *off*, tracing passes through data ports. The default option is *on*.

  When the **Stop on Flip-flop/Latch** option is turned *off*, the **Passthrough Port Type**  selection field is activated. Click the upside-down triangle to turn *on* the selected port type from the **Data**, **Clock**, **control**, **Set/Reset**, and **Other** options (multiple selections are supported). When tracing between two points, the selected port type is passed through and port types that are not selected is stopped. When the **Stop on Flip-flop/Latch** option is turned *on*, the **Passthrough Port Type**  selection field is disabled. The default port type is **Data**.
- **Stop on Tri-state**: When this option is turned *on*, tracing stops at a tri-state when tracing between two points. The default option is *on*.
- **Hide Neighbor Cells**: When this option is turned on, neighbor cells of the traced signal is not displayed. The default option is *on*.
- **Create Window**: When this option is turned *on*, the tracing results is displayed in a new *nSchema* window. The default option is *off*.
- **Apply Settings to Trace 2 Nodes Toolbar:** When this option is turned *on*, the settings in *Trace Two Points* form is automatically applied to **Trace 2 Nodes** toolbar.

Trace Two Points Filter

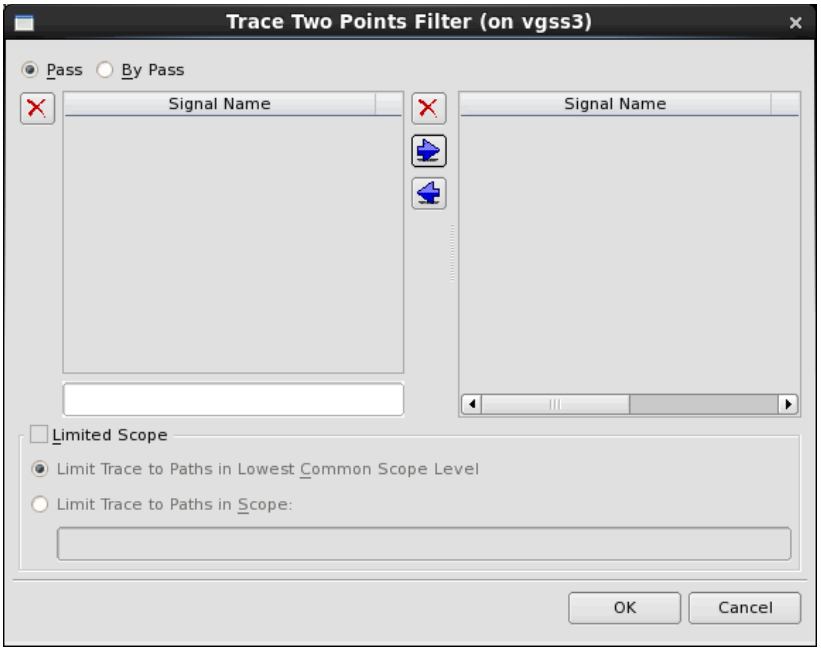


Figure: Trace Two Points Filter Form

After the **Pass** or **By Pass** options are selected, enter the signal name in the text field directly below the left table. Click **Add**, **Remove**, or **Clear** icons to determine the items which need to be available in the right table for **Pass** or **By Pass**, and then click **OK**.

- **Pass:** Pass through the specified signal.
- **By Pass:** Do not pass through the specified signal.
- **Limit Trace to Paths in Lowest Common Scope Level:** Limit trace paths to the lowest common scope between from-point and to-point. For example, when the from-point is *Top.A.B.C.D.portA* and the to-point is *Top.A.B.E.F.G.portB*, *Top.A.B* is the lowest common scope.
- **Limit Trace to Paths in Scope:** Limit the trace to paths in the specified scope. The scope name can be entered manually by entering values in the text field or by dragging a scope from the *nSchema* window and dropping it in the text field.

By Level

Menu Bar: Trace -> By Level

This command opens the *Trace by Level* form which can be used to set the trace type and level, display the signal/instance to be traced, and open a new schematic window to display the trace result.

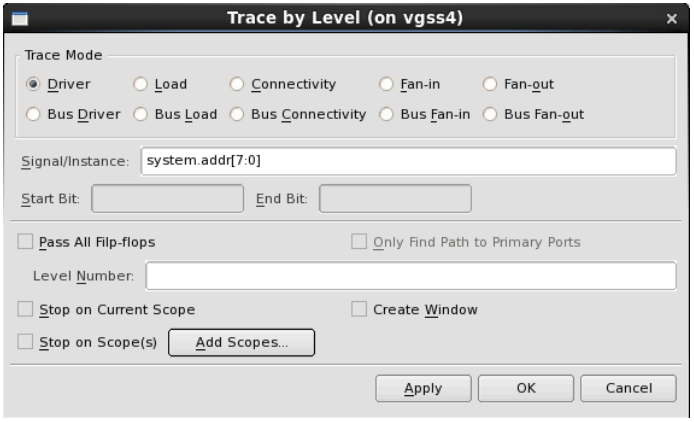


Figure: Trace by Level Form

The *Trace by Level* form includes the following options and fields:

- **Trace Mode:** Specify the signal/instance trace type by selecting one of the following options: **Driver, Load, Connectivity, Bus Driver, Bus Load, Bus Connectivity, Fan-in, Fan-out, Bus Fan-in**, and **Bus Fan-out**. Only one type may be selected at a time.
- **Signal/Instance:** This text field displays the name of the selected signal/instance to be traced. To trace another signal/instance, drag it from the schematic and drop it to this field to replace the existing signal/instance.
- **Start Bit:** When the signal type is a bus, the start bit of the bus to be traced can be entered in this text field. When the signal type is not a bus, this field is disabled. The default is the start bit of the full bus bit range.
- **End Bit:** When the signal type is a bus, the end bit of the bus to be traced can be entered in this text field. When the signal type is not a bus, this field is disabled. The default is the end bit of the full bus bit range.
- **Level Number:** This field is activated when the **Pass All Devices** option is turned *off*. The level number to be traced can be specified and entered in this field.

- **Stop on Module Boundary:** When this option is turned *on*, the trace stops on the module boundary without continuing.
- **Create Window:** This option opens a new schematic window to display the trace result when the trace is completed. The default option is *off*.

NOTE: The options specified here are NOT saved to the *novas.rc* resource file.

Shortest/Longest Path

Menu Bar: Trace -> Shortest/Longest Path

NOTE: An SDF file must be loaded to enable this command.

This command opens the *Shortest/Longest Path* form in which tracing for the shortest/longest path between two points, fan-ins, or fan-outs in the selected instance port can be set, based on the path delay information (both INTERCONNECT and CELL DELAY) in the SDF file.

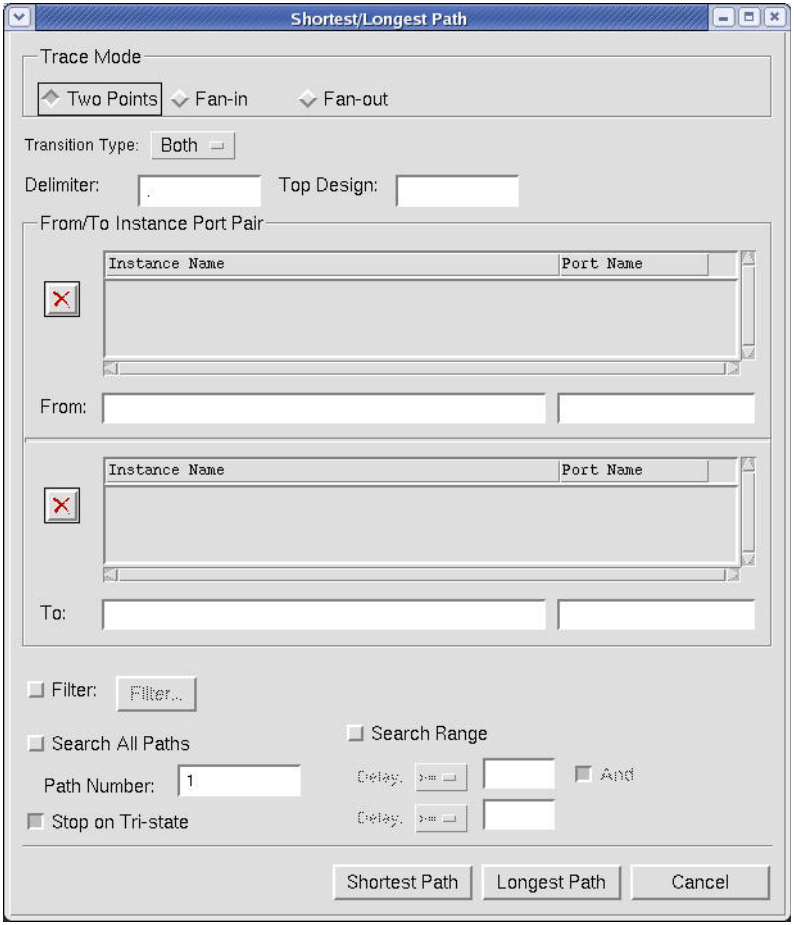


Figure: Shortest/Longest Path Form

The *Shortest/Longest Path* form includes the following options and fields:

- **Trace Mode:** Specify the trace mode by selecting one of the following mode options (only one may be selected at a time):
  - **Two Points:** To trace paths between two points, enter an instance port in the **From** and **To** text fields. The port may be entered by entering the options manually or by dragging an object from the *nSchema* or *nTrace* windows and dropping the selection in the **From** and **To** text fields.

    If the selected object is an instance, all port(s) in the instance are listed in the **From** and **To** table and the desired port must be selected.
  - **Fan-in:** To trace fan-in registers from the specified instance port, enter an instance port in the **From** text field. The port may be entered by entering the options manually or by dragging an object from the

nSchema or *nTrace* windows and dropping the selection to the **From** text field.

If the selected object is an instance, all port(s) in the instance are listed in the **From** table and the desired port must be selected.

- **Fan-out**: To trace fan-out registers from the specified instance port, enter an instance port in the **From** text field. The port may be entered by entering the options manually or by dragging an object from the *nSchema* or *nTrace* windows and dropping the selection to the **From** text field.

  If the selected object is an instance, all port(s) in the instance are listed in the **From** table and the desired port must be selected.

- **Transition Type:** Specify a transition type from the following three selections: **Both**, **Rising**, and **Falling**.
- **Delimiter:** Enter the delimiter in this text field. The default option is the period (.) character.
- **Top Design:** Enter part of the scope name in this text field and the top design name are added to the complete the scope name.
- **From/To Instance Port Pair:** Display the instance port pairs for tracing. Select the **Clear** icon to clear the selected signal(s) in the **From** and **To** text fields.
- **Filter:** This option is only activated when the **Two Points** option is selected.
- **Search All Paths:** When this option is turned *on*, all paths are searched. When this option is turned *off*, a number must be entered in the **Path Number** text field.
- **Search Range:** When this option is turned *on,* the search range can be specified.
- **Stop on Tri-state:** When this option is turned *on*, tracing stops at a tri-state devices when tracing between two points. The default option is *on*.
- **Shortest Path**: After specifying the settings in the *Shortest/Longest List* form, click this button to view the results in the *Path List* form in which the paths are listed from the shortest delay time to the longest delay time.
- **Longest Path:** After specifying the settings in the *Shortest/Longest List* form, click this button to view the results in the *Path List* form in which paths are listed from the longest delay time to the shortest delay time.

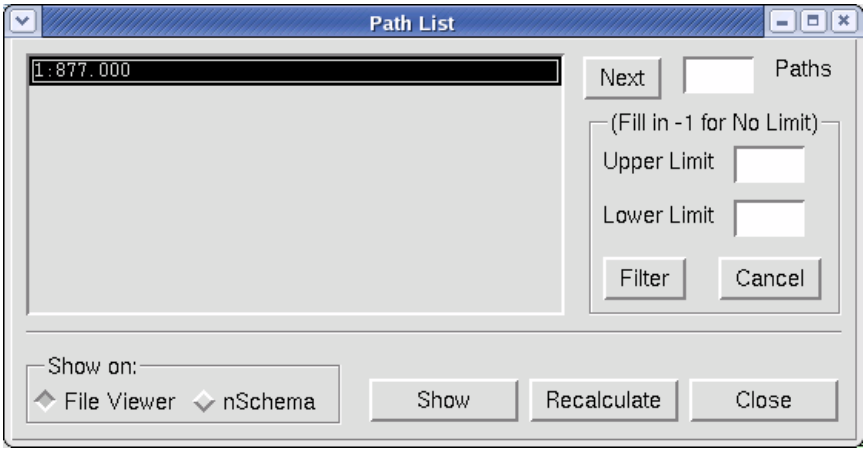


Figure: Path List Form

The next shortest/longest path are displayed by clicking the **Next __ Paths** button in the *Path List* form. The **Upper Limit** and **Lower Limit** fields are filters to set SDF path delay. In the **Show On** section, enable the **File Viewer** option to display the path result in ASCII format in a *Text Viewer* frame; enable the **nSchema** option to display the path results in a flattened schematic window with SDF annotation displayed. Click **Show** to view the results based on specifications in the **Show On** section.

NOTE: To generate correct results, the symbol library must be generated by the most updated *syn2SymDB utility*. The *syn2SymDB* utility extracts the timing information from the Synopsys.*lib* file which the **Longest/ Shortest Path** command requires to calculate the path.

Show Trace Report

Menu Bar: Trace -> Show Trace Report

This command is used to generate a report for the **2 Points**, **Fan In**, **Fan Out,** or **Clock Tree** schematics. When an SDF is loaded, the report can also display the delay of each path.

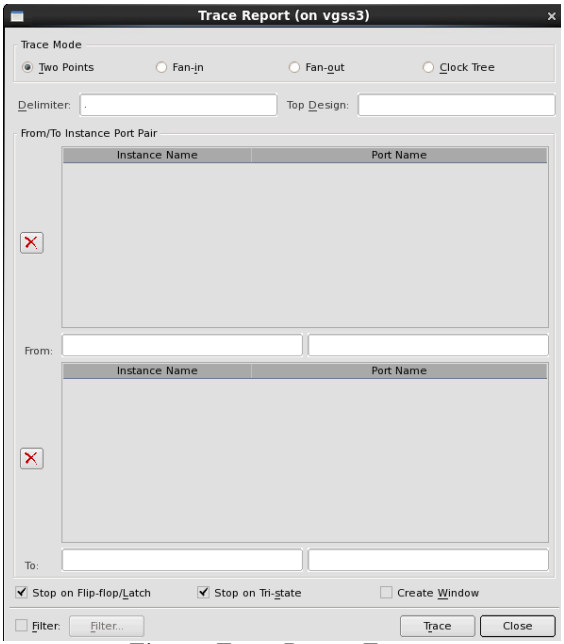


Figure: Trace Report Form

The *Trace Report* form includes the following options, fields, and buttons:

- **Trace Mode:** Specify a trace mode by selecting one of the following options: **2 Points**, **Fan In**, **Fan Out,** or **Clock Tree**.
- **Delimiter:** Enter the delimiter in this text field. The default value is the period (.) character.
- **Top Design:** Enter the top design unit.
- **From/To Instance Port Pair:** Specify one instance loading port in the **From** text field and another instance driving port in the **To** text field. An instance port can be dragged from the schematic window to both the **From** and **To** text fields. This command only traces at the bottom level.
- **Clear** : Clear the signal(s) deleted in the **From** and **To** text fields.
- **Stop On FF/Latch:** When this option is turned *on*, tracing stops at a FF boundary and/or a latch when tracing between two points. The default option is *on*.
- **Stop On TriState:** When this option is turned *on*, tracing stops at a TriState device when tracing between two points. The default option is *on*.
- **Create Window:** When this option is turned *on*, a new schematic window is created when tracing between two points. The default option is *off*.

- **Filter (only available for 2 Points):** When this option is turned *on* and the corresponding **Filter** is clicked, the *Trace Two Points* filter form is opened. Refer to the *Trace Two Points Filter* section for details.
- **Trace:** Click this button to open the *Trace Report Result* form in which the trace results are available.

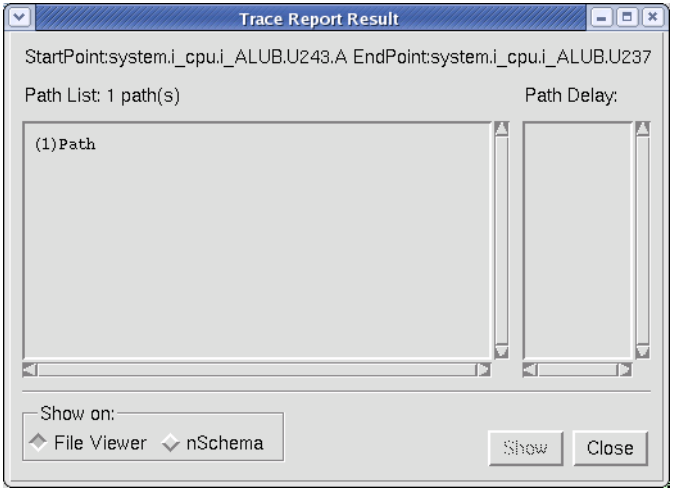


Figure: Trace Report Result Form

The *Trace Report Result* form displays the **Path List** and **Path Delay**. In the **Show on** section, two display formats, **File Viewer** or **nSchema**, can be selected to display the trace results. **File Viewer** displays the path in ASCII format in a *Text Viewer* frame. **nSchema** displays the path with the path annotation in a flattened window. After selecting either the **File Viewer** or **nSchema** option, click **Show** to view the results in the specified format.

Value Propagation

Value Propagation

Menu Bar: Trace -> Value Propagation -> Value Propagation

This command opens the *Propagation Settings* form in which propagation values for the desired signal/cell can be set under the **Signal** and **Cell Port** tabs in the **Value Propagation Settings** section.

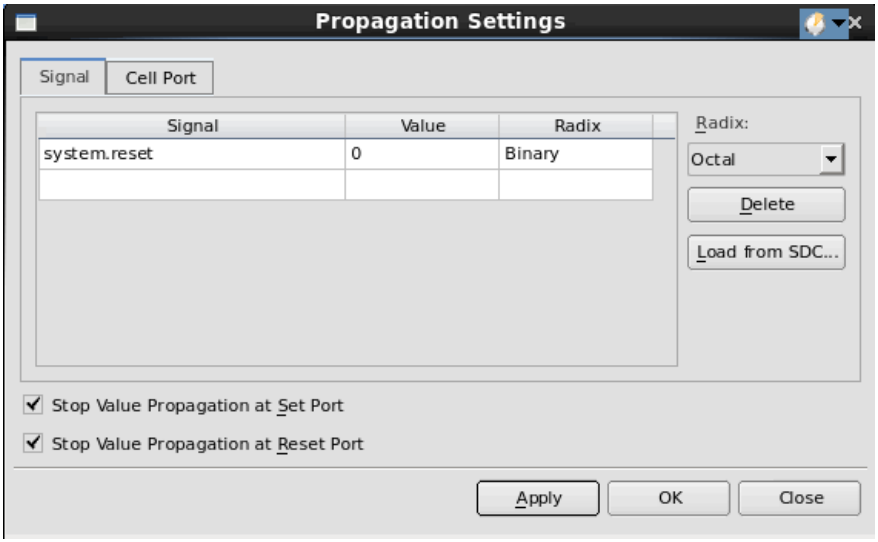


Figure: Propagation Settings Form

The *Propagation Settings Form* includes the following fields, options, and buttons:

- **Value Propagation Settings:** In this section, enter a signal name in the **Signal** column of the **Signal** tab or a cell/port name in the **Cell.port** column of the **Cell Port** tab. Alternatively, a signal/cell can be dragged from the *nSchema* window and dropped to the desired column. After a signal/cell name is added to the **Signal** or **Cell Port** tabs, enter a value in the **Value** column and select the appropriate radix in the **Radix** selection field.
- **Radix:** Specify the radix value by selecting one of the following options: **Decimal**, **Hex**, **Octal**, or **Binary**. The default option is *Binary*.
- **Delete:** Click this button to remove the selected signal or the port from the table.
- **Load from SDC:** Click this button to load the SDC file contents to the *Propagation Settings* form.
- **Stop Value Propagation at Set Port:** When this option is turned *on*, propagation is stopped when a set port is reached.
- **Stop Value Propagation at Reset Port:** When this option is turned *on*, propagation is stopped when a reset port is reached.

Conflict Report

Menu Bar: Trace -> Value Propagation -> Conflict Report

After the value propagation is completed and an FSDB file is loaded in the *nTrace* window, this command is enabled. Invoke this command to open the *Value Propagation Report* form in which differences between the FSDB value and the propagated value are displayed.

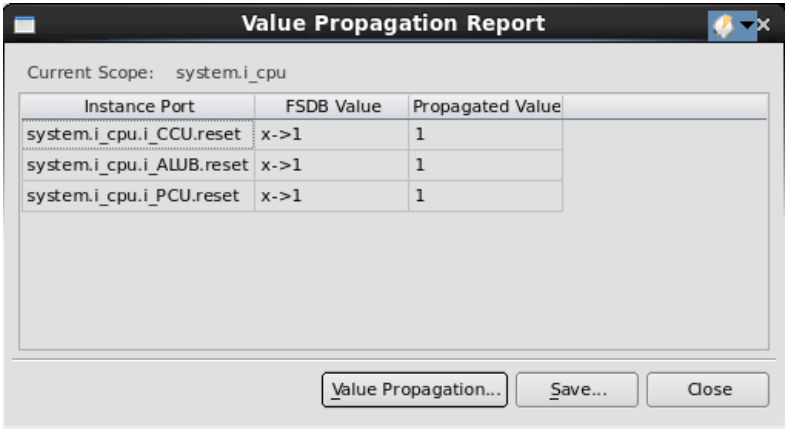


Figure: Value Propagation Report Form

The *Value Propagation Report* form includes the following fields and options:

- **Instance Port, FSDB Value, Propagated Value:** In the value propagation report, the **Instance Port** column displays the instance port name, the **FSDB Value** column displays the FSDB value at the current time, and the **Propagated Value** column displays the propagated value.
- **Value Propagation:** Click this option to open the *Propagation Settings* form in which new propagation values can be set or previous propagation settings edited and propagated again.
- **Save:** Click this button to save the current report to a file as follows:

  ***** Value Propagation Confliction Report *****

  Instance Port : system.i_cpu.i_ALUB.U237.B

  FSDB Value : NF

  Propagated Value : 0

  Instance Port : system.i_cpu.i_ALUB.U237.Z

  FSDB Value : NF

  Propagated Value: 1
- **Close:** Click this button to close the *Value Propagation Report* form.

Clock Analyzer

Find Clock Source - nAnalyzer

Menu Bar: Trace -> Clock Analyzer -> Find Clock Source

This command is enabled when a signal (net or port), a flip-flop, or a latch is selected. This command opens the *Find Clock Source* form where the clock source for a clock net can be found. It can also be applied to a normal signal to find its possible drivers. Constants can be specified for certain signals to guide the analysis and obtain the real clock source. The constant signals can be entered in the grid window or a signal (from schematic window, source window, and so on.) can be dragged to the grid window. An instance port or an instance can also be entered/dragged into the grid window. This function only supports library cells.

Here, disable indicates the constant forces the gates to be stuck at a value. For example, set 0 to one input of a 2-input AND gate, and the 2-input AND gate is disabled since the output of the gate is stuck at 0. Another example, set 0 to one input of a 2-input NAND gate, and the 2-input NAND gate is disabled since the output of the gate is stuck at 1. The clock source is not found from the 2-input AND gate.

Click **Find** to find the clock source, and select it automatically. If the real clock source is invoked from the detailed-RTL schematic window, *nSchema* selects the new clock source. If not, *nSchema* opens a new detailed RTL window, and selects the clock source in the new schematic window.

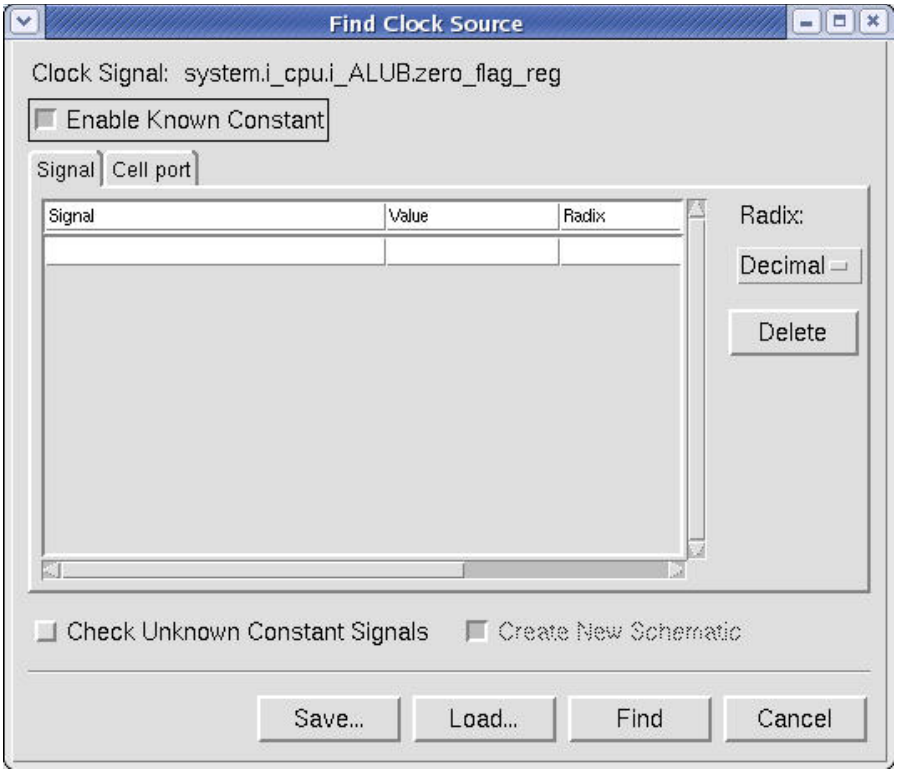


Figure: Find Clock Source Form

The *Find Clock Source* form includes the following fields, options, and buttons:

- **Enable Known Constant:** When this option is turned *off*, the grid window, the buttons on the right side, **Save**, and **Load** are disabled.
- **Radix:** Specify the radix value by selecting one of the following options: **Decimal, Hex, Octal,** or **Binary**.
- **Delete:** Click this option to delete the selected signal from the **Known Constant Signals** list.
- **Save:** Click this option to save the constant signals to a file.
- **Load:** Click this option to load the constant signals from a file.
- **Check Unknown Constant Signals:** When this option is turned *on*, *nSchema* opens a form so the value of the constant signal can be set when an undetermined gate is encountered.
- **Create New Schematic:** When this option is turned *off*, the results are displayed on the same window instead of a new schematic window. When *nSchema* displays the results on the same window, the newly added results are highlighted with the “trace color” (default is yellow) similar to tracing a

driver or the load in the schematic window. If the selected signal for **Find Clock Source** is in a hierarchical schematic view, this option is turned *on* and it cannot be turned *off*.

Extract Clock Information - nAnalyzer

Menu Bar: Trace -> Clock Analyzer -> Extract Clock Information

This command opens the *Clock Extraction Settings* form. Refer to the *Clock Extraction Settings Form* section in the *Clock Analyzer* chapter for details.

List Clock Domains - nAnalyzer

Menu Bar: Trace -> Clock Analyzer -> List Clock Domains

This command opens the *Clock Domains* window. Refer to the *Clock Domains Window* section in the *Clock Analyzer* chapter for details.

List Specified Clock Domains - nAnalyzer

Menu Bar: Trace -> Clock Analyzer -> List Specified Clock Domains

This command opens the *Clock Tree Browser* window. Refer to the *Clock Tree Browser Window* section in the *Clock Analyzer* chapter for details.

Highlight Clock Domain - nAnalyzer

Menu Bar: Trace -> Clock Analyzer -> Highlight Clock Domain

This command opens the *Highlight Clock Domains* form. Refer to the *Highlight Clock Domains Form* section in the *Clock Analyzer* chapter for details.

Check Crossing Paths - nAnalyzer

Menu Bar: Trace -> Clock Analyzer -> Check Crossing Paths

This command opens the *Check Crossing Paths* form. Refer to the *Check Crossing Paths Form* section in the *Clock Analyzer* chapter for details.

List Crossing Paths - nAnalyzer

Menu Bar: Trace -> Clock Analyzer -> List Crossing Paths

This command opens the *Crossing Paths* window. Refer to the *Crossing Paths Window* section in the *Clock Analyzer* chapter for details.

Clock Tree

Highlight Clock Tree - nAnalyzer

Menu Bar: Trace -> Clock Tree -> Highlight Clock Tree

After a net is selected, this command opens the *Highlight Clock Tree* form displaying the selected net with its current color in the color plate. Click the color on the color plate to highlight the clock tree with a different color.The default option is *red*.

Unhighlight Clock Tree - nAnalyzer

Menu Bar: Trace -> Clock Tree -> Unhighlight Clock Tree

This command displays a form listing all the highlighted clock trees. This form can be used to select the clock tree which must be unhighlighted.

Save Clock Tree - nAnalyzer

Menu Bar: Trace -> Clock Tree -> Save Clock Tree

After a clock tree is created or highlighted, use this command to save the trace results, including path type attributes (for example, Data Pin of Register, Duplication, Loop, GATECLK, CTS Stop Pin (which includes Primary Outputs), EXOR, and Pin Name). This command opens a *Save As* form where the directory and file name for the trace results can be specified.

Clock Skew - nAnalyzer

Menu Bar: Trace -> Clock Tree -> Clock Skew

This command checks the clock skew for a clock tree based on the shortest/ longest path. Select a clock source and check its clock skew.

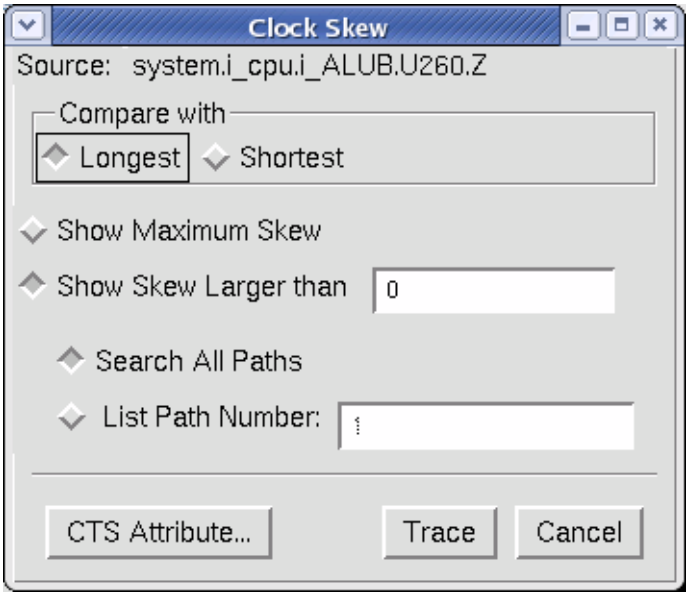


Figure: Clock Skew Form

The *Clock Skew* form includes the following options and fields:

- **Compare With:** Specify if the clock skew with either the **Longest** or **Shortest** path must be compared.
- **Show Maximum Skew:** When this option is selected, it calculates and displays the results based on the maximum skew.
- **Show Skew Larger than:** When this option is selected, it calculates and displays the results for skews larger than the specified value.
- **Search All Paths:** When this option is selected, the report skews for all paths.
- **List Path Number:** When this option is selected, the report skews for the specified number of paths.
- **CTS Attribute:** Click this option to open a form where CTS pin attribute can be input. Refer to the **CTS Attributes Tab** description of the *Clock Extraction Settings Form* form in the *Clock Analyzer* chapter for details.
- **Trace:** Click **Trace** to open the *Clock Skew List* form. Refer to the *Clock Skew List Form* section for details.

Clock Skew List Form

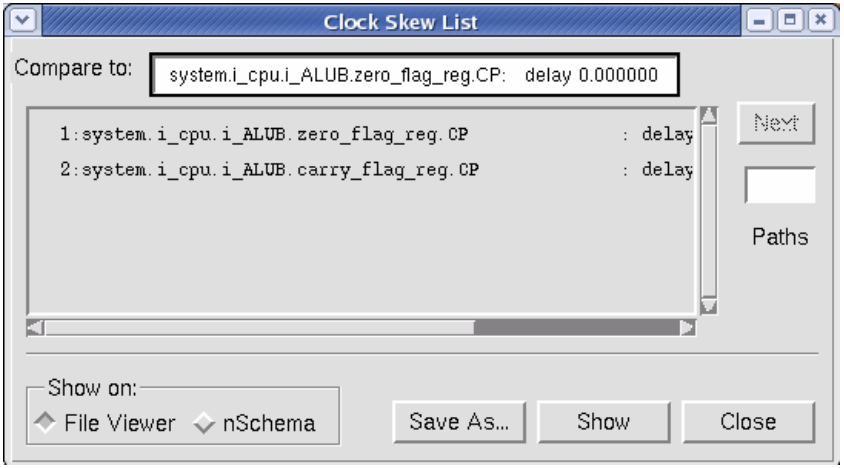


Figure: Clock Skew List Form

The *Clock Skew List* form includes the following options and fields:

- **Compare To:** Select the clock source to compare it.
- **Next Paths:** If the number *y* entered in the **List Path Number** text field on the *Clock Skew* form is greater than 1 and more than 1 path exists in the *Clock Skew List* form, enter a number *x* in the text field and click **Next** to see the next *x* paths instead of all of the *y* paths.
- **Show On:** Specify the output format by selecting either the **File Viewer** or **nSchema** option. A single or multiple clock domain(s) can be selected from the *Clock Skew List* form. To select multiple clock skews, drag-left or ctrl-left-click. In the **Show On** section, select if the **Clock Skew List** must be displayed in either **File Viewer** format or **nSchema** format. The **File Viewer** displays the domain in ASCII format in a *Text Viewer* frame and **nSchema** displays the skew in a flattened window. After selecting the option, click **Show** to display the results in the format defined by the selection in the **Show On** section.
- **Save as:** Click this button to save the results as a text (*.txt*) file.
- **Show:** After the **File Viewer** or **nSchema** option is selected in the **Show On** section, click **Show** to view the results in the format defined in the **Show On** section.

Reset Tree

Highlight Reset Tree - nAnalyzer

Menu Bar: Trace -> Reset Tree -> Highlight Reset Tree

This command opens the *Highlight Reset Tree* form that displays the highlight color for a selected primitive instance or net in the current reset tree results. Click a color on the color plate to highlight the reset tree with a different color. The default option is *green*.

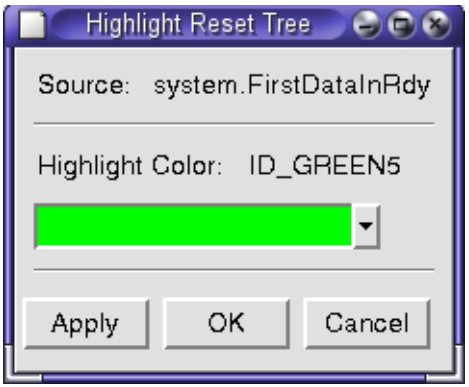


Figure: Highlight Reset Tree Form

Unhighlight Reset Tree- nAnalyzer

Menu Bar: Trace -> Reset Tree -> Unhighlight Reset Tree

This command is only active after a reset tree is highlighted. This command opens the *Unhighlight Reset Tree* form that lists the highlight colors for the current extracted reset tree.

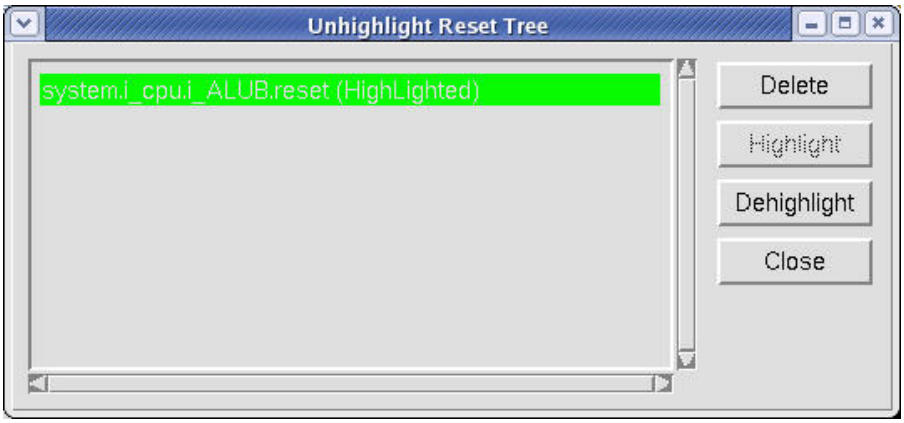


Figure: Unhighlight Reset Tree Form

The *Unhighlight Reset Tree* form includes the following options and fields:

- **Delete**: Removes the selected reset net from the list.
- **Highlight**: Highlights the selected reset net.
- **Dehighlight**: Removes the current highlight from the selected reset net and leaves the net in the list to be highlighted again later.

Save Reset Tree - nAnalyzer

Menu Bar: Trace -> Reset Tree -> Save Reset Tree

After the **Tools -> New Schematic -> Reset Tree** command is invoked to create a new flattened schematic window containing the extracted reset tree, use this command to save the extracted reset tree results. A form opens in which the directory and file name to be used for the extracted reset tree results can be specified.

Reset Skew - nAnalyzer

Menu Bar: Trace -> Reset Tree -> Reset Skew

When the SDF file is loaded, this command checks the reset skew for a selected reset source in the current reset tree.

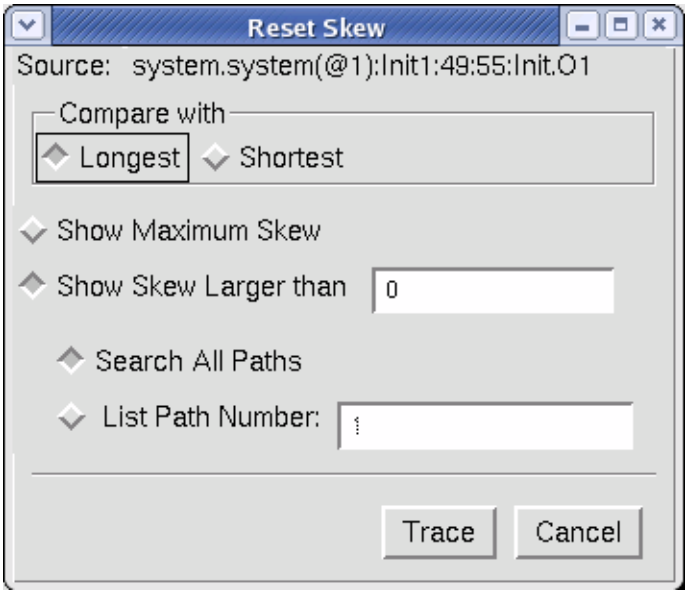


Figure: Reset Skew Form

The *Reset Skew* form includes the following options, fields, and buttons:

- **Compare With:** Specify if the reset skew must be compared with either the **Longest** or **Shortest** path.
- **Show Maximum Skew:** When this option is selected, it calculates and displays the results based on the maximum skew.
- **Show Skew Larger than:** When this option is selected, it calculates and displays the results for skews larger than the specified value.
- **Search All Paths:** When this option is selected, the report skews for all paths.
- **List Path Number:** When this option is selected, the report skews for the specified number of paths.
- **Trace:** Click this button to open the *Choose* form if the **Rising** or **Falling** edge options for the reset source must be selected.

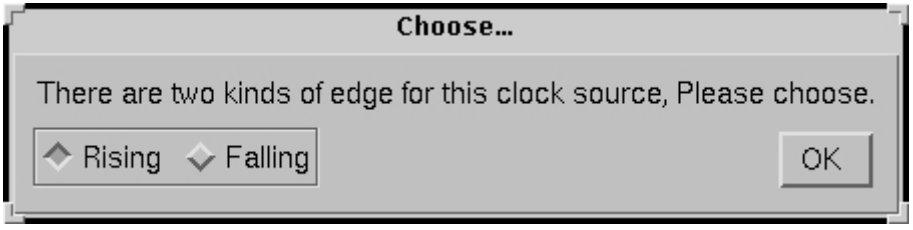


Figure: Choose Either Rising or Falling for the Reset Source

- Click **OK** on the *Choose* form and the *Reset Skew List* form displays the results. Refer to the *Reset Skew List Form* section for details.

Reset Skew List Form

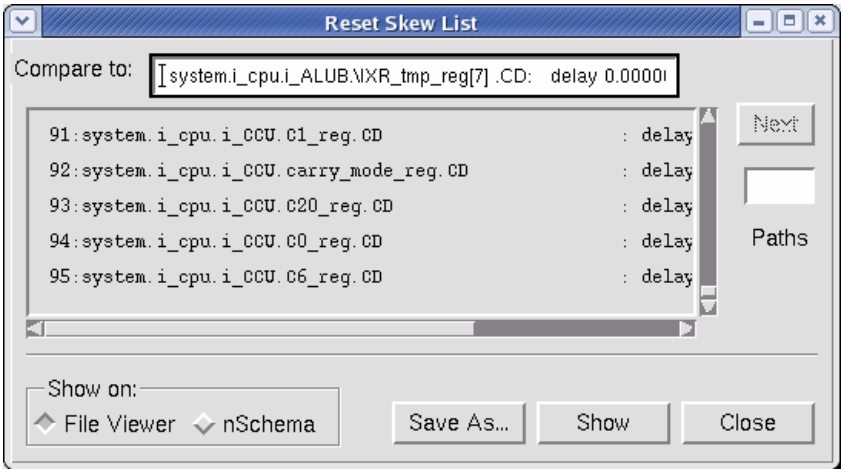


Figure: Reset Skew List Form

The *Reset Skew List* form includes the following options, fields, and buttons:

- **Compare To:** Lists the reset port that is selected for the results.
- **Next Paths:** If the number *y* entered in the **List Path Number** text field on the *Reset Skew* form is greater than 1 and more than 1 path exists in the **Reset Skew List**, enter a number *x* in the text field and click **Next** to view the next *x* paths instead of all the *y* paths.
- **Show On:** Select either the **File Viewer** or **nSchema** option to specify the output format. A single or multiple clock domain(s) can be selected from the *Reset Skew List* form. To select multiple reset skews, drag-left or ctrl-left-click. In the **Show On** section, select to display the **Reset Skew List** in either **File Viewer** format or **nSchema** format. The **File Viewer** displays the domain in ASCII format in a *Text Viewer* frame and **nSchema** displays the skew in a flattened window. After selecting an option, click **Show** to display the results in the format defined by the selection in the **Show On** section.
- **Save as:** Click this button to save the results as a text (*.txt*) file.
- **Show:** After selecting either the **File Viewer** or **nSchema** option, click **Show** to view the results in the specified format.

Add Results to Waveform (Hierarchical View Only)

Menu Bar: Trace -> Add Results to Waveform

NOTE: This command is available in the *Full Hierarchical* window.

This command is enabled when an *nWave* window is open, simulation results are loaded, and the trace results exist in the schematic view. This command adds the trace results to the *nWave* window (the results are added to the location of the cursor bar).

Backward History

Menu Bar: Trace -> Backward History

Toolbar Icon: 

This command can trace back through the most recent 32 trace results.

Forward History

Menu Bar: Trace -> Forward History

Toolbar Icon: 

This command can trace forward through the most recent 32 trace results.

Reset History

Menu Bar: Trace -> Reset History

This command cleans up the trace buffer, which records the most recent 32 trace results.

Options

Ignore Scopes during Tracing

Menu Bar: Trace -> Options -> Ignore Scopes during Tracing

This command opens the *Trace Ignore Scopes* form in which the scopes to ignore during schematic can be specified. After setting the scopes, the preference must be enabled.

Drag the module instance from *nTrace* (both the design browser frame and the source code frame), *nSchema*, or type in the module name in the **Scope** text field and click **Add**. This adds the ignored module to the table. The scopes can be toggled *on* and *off* in this window without deleting them. Only the enabled scopes are applied to the *Trace Ignore Scopes* window when tracing schematics. Each scope is toggled *on* when it is added first. Multiple selections are supported in the *Trace Ignore Scopes* window.

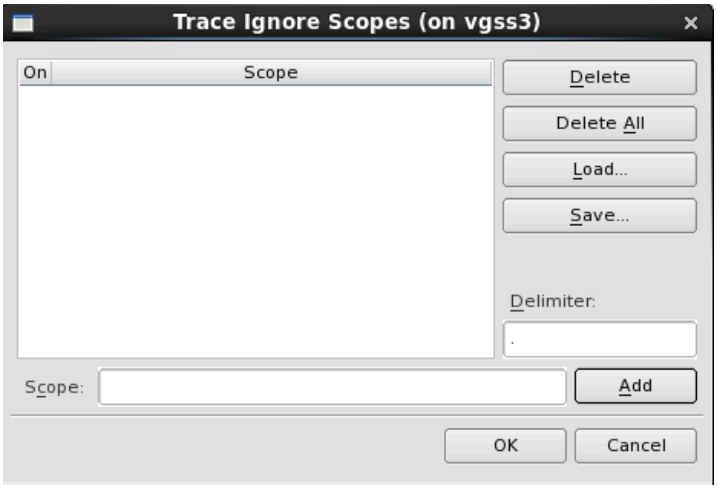


Figure: Trace Ignore Scopes Form

The *Trace Ignore Scopes Form* form includes the following buttons and fields:

- **Delete:** After the required module from the list is selected, click this button to set the specified module as the ignored module.
- **Delete All:** Click this button to delete all the ignored modules.
- **Load:** Click this button to open the *Open File* form where the directory structure can be viewed to load a previously saved filter file.
- **Save:** Click this button to open the *Save File* form in which the directory structure can be viewed to specify a file to save the filter settings.
- **Delimiter:** Enter the delimiter in this text field. The default option is the period (.) character.

Auto Merge Setting - nAnalyzer

Menu Bar: Trace -> Options -> Auto Merge Setting

NOTE: This command is enabled in a schematic window opened with the **Tools -> New Schematic** command in the *Clock Tree Browser* window and the *Clock Domains* window.

This command opens the *Auto Merge Cell Settings* form in which cells can be set to merge.

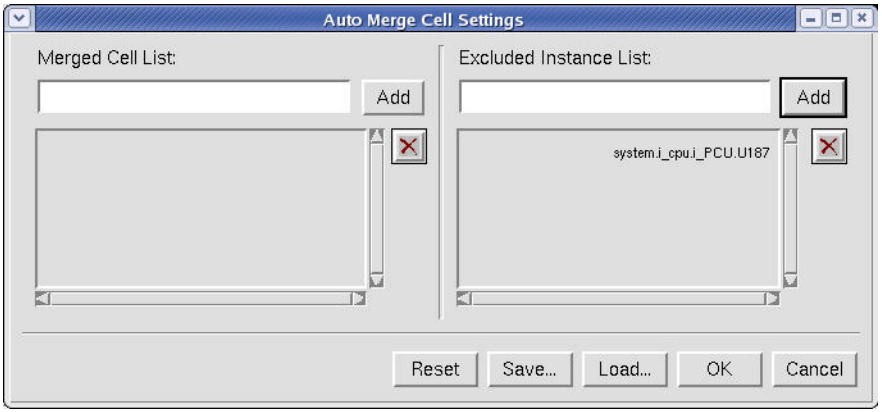


Figure: Auto Merge Cell Settings Form

- **Merged Cell List:** The required cell may be entered by entering manually (the wildcard character * is supported) or by dragging a cell from the *nSchema* or *nTrace* windows and dropping the selection to the **Merged Cell List** text field. Click **Add** to display the selected cell in the **Merged Cell List** table. Click icon to remove the selection.
- **Excluded Instance List:** The instance to exclude may be entered by entering manually (the wildcard character * is supported) or by dragging an instance from the *nSchema* or *nTrace* windows and dropping the selection to the **Excluded Instance List** text field. Click **Add** to display the selected instance in the **Excluded Instance List** table. Click icon to remove the selection.

Trace Stop on Module Boundaries

Menu Bar: Trace -> Options -> Trace Stop on Module Boundaries

This toggle command indicates if the cone trace stops at the current module boundary. It passes through the boundary of the sub-modules. The default option is *off*.

When the **Stop on Module Boundary** option is turned *on* in the *Preferences* form (invoked with the **Tools -> Preferences -> Schematic -> Trace** page), then this option is enabled by default for the new *nSchema* windows.

Stop Cone Tracing on FSM

Menu Bar: Trace -> Options -> Stop Cone Tracing on FSM

This toggle command indicates if the cone trace stops at an FSM boundary. The default option is *on*.

You can also disable the command for the new *nSchema* windows using the **Stop on FSM** option (invoked with the **Tools -> Preferences -> Schematic -> Trace** page).

Stop Cone Tracing on MOS Cell

Menu Bar: Trace -> Options -> Stop Cone Tracing on MOS Cell

This toggle command indicates if the cone trace stops at the MOS cell. The default option is *off*.

When the **Stop on MOS Cell** option is turned *on* in the *Preferences* form (invoked with the **Tools -> Preferences -> Schematic -> Trace** page), then this option is enabled by default for the new *nSchema* windows.

Hide Extraneous Bus (Flattened View Only)

Menu Bar: Trace -> Options -> Hide Extraneous Bus

NOTE: This command is available in the *Flattened* window.

This toggle command controls the bus place and route (P&R) style. When this command is enabled, all bus members are treated as individual signals and are not displayed as a bus. If a bus is dragged and then dropped into an *nSchema* window with the **Hide Extraneous Bus** command enabled, *nSchema* highlights all the nets belonging to the bus. *nSchema* always keeps the selected set when the **Hide Extraneous Bus** command is invoked.

When a flattened schematic window or an ECO window is created from another flattened schematic window, the **Hide Extraneous Bus** option inherits its parent.

Refer to the *NON_SPLIT_BUS* environment variable for details.

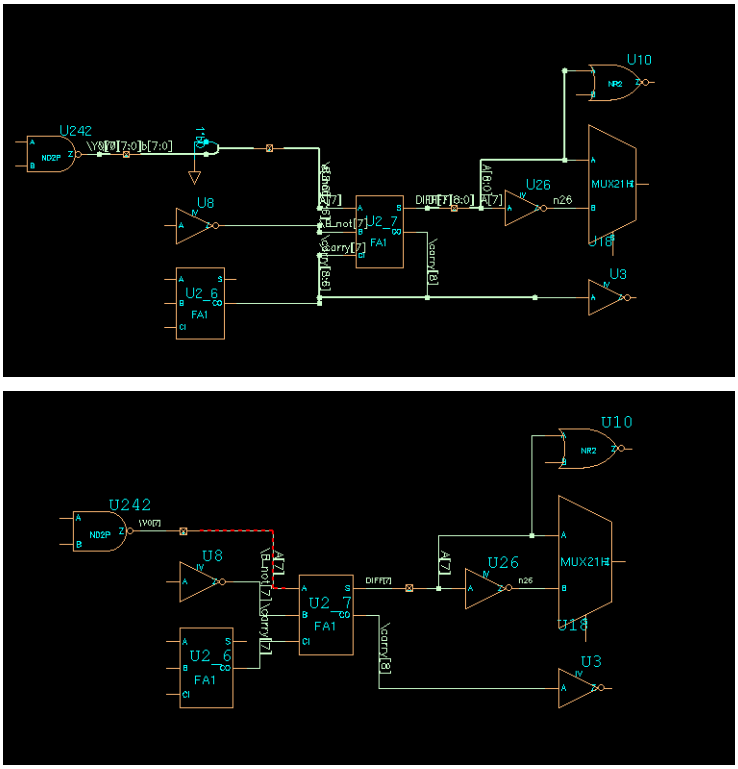


Figure: Before and After Turning the Hide Extraneous Bus Command On

Auto Group Buffer/Inverter Cell (Flattened View Only)

Menu Bar: Trace -> Options -> Auto Group Buffer/Inverter Cell

NOTE: This command is available in the *Flattened* window.

If the clock trees in the design include many buffers and inverters, enable this toggle command to automatically group all buffers and inverters together. The number of grouped buffers and inverters is displayed on the triangle symbol, as illustrated in the following images for an example.

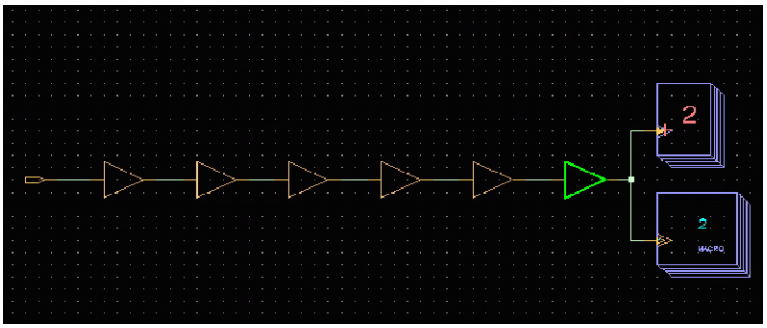


Figure: Before Enabling the Auto Group Buffer/Inverter Cell Command

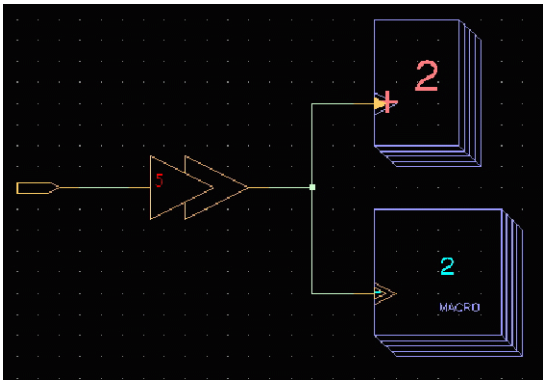


Figure: After Enabling the Auto Group Buffer/Inverter Cell Command

To edit the group instance, select the required instance and then use the two right-click menu options as follows:

- **Group Instance Information:** This command opens the *Group Instance Information* form which displays the instance names of the group instance.
- **Delete Group Instance**: This command removes the group instance. The individual instances in the deleted group instance are returned to the current schematic view. If a group instance is deleted when the **Auto Group Buffer/Inverter Cell** option is turned *on*, the **Auto Group Buffer/Inverter Cell** option is turned *off* automatically.

To create a group instance, select the desired instances in the flattened schematic window and then invoke the **Create Group Instance** command in the right-click menu option.

Auto Merge Cell - nAnalyzer

Menu Bar: Trace -> Options -> Auto Merge Cell

This toggle command turns the display of merged cells *on* or *off*. When **Auto Merge Cell** is turned *on*, the merged cell are highlighted and the name of the merged cell are displayed as *MergedGroupNumber*. The default option is *off*.

Tools Commands

Refer to the *Overview* section for a description of the different types of *nSchema* windows.

New Schematic

Current Scope

Menu Bar: Tools -> New Schematic -> Current Scope

Bind Key: Shift+S

This command generates a hierarchical view in *nSchema* that reflects the selected scope (active scope) in the design browser frame.

Browser Window

Menu Bar: Tools -> New Schematic -> Browser Window

This command creates a partial hierarchical view in *nSchema* that reflects the selected signals or instances in the *nTrace* or *nSchema* windows. The port is displayed by default to switch to the upper hierarchy.

NOTE: The **Browser Window** command automatically enables active annotation in the new *nSchema* window.

NOTE: With this type of window, the scope can only be changed with the **View -> Pop View Up**, **View -> Push View In**, or **View -> Pop View Up from Port** commands, or you can double-click on the port or instance port.

NOTE: A new browser window inherits the **View -> Detailed RTL** toggle command setting of the parent schematic window. If the **View -> Detailed RTL** toggle command is enabled in the schematic window in which the new browser window is created, the **View -> Detailed RTL** toggle command is enabled in this new browser window.

Flattened Window

Menu Bar: Tools -> New Schematic -> Flattened Window

Bind Key: Shift+F

This command creates a flattened schematic window that reflects the selected primitive instances in the *nTrace* or *nSchema* windows.

Hierarchical Flattened View

Menu Bar: Tools -> New Schematic -> Hierarchical Flattened View

Bind Key: Shift+H

This command opens the *Hierarchical Flattened View* window that displays the hierarchical boundaries in the flattened schematic window.

ECO Window for Selected Instance(s) - nECO

Menu Bar: Tools -> New Schematic -> ECO Window for Selected Instance(s)

This command opens an *nECO* window for selected instances.

ECO Window for All Instances - nECO

Menu Bar: Tools -> New Schematic -> ECO Window for All Instances

This command opens an *nECO* window for all instances for the invoked *nSchema* hierarchical window, flattened window, or browser window. All the view attributes such as instance names, ports, and nets names are carried over automatically.

Editable Window for Selected

Menu Bar: Tools -> New Schematic -> Editable Window for Selected

This command opens an *Editable Schematic* window for selected instances in the current window. When an area is selected in *nSchema*, if a full instance is not included in the selection, only pins are kept in the selection; if any full instances are included in the selection, instances are kept in the selection. Refer to the *Editable Schematic* chapter for details.

Editable Window for All

Menu Bar: Tools -> New Schematic -> Editable Window for All

This command opens an *Editable Schematic* form for all instances in the current window. Refer to the *Editable Schematic* chapter for details.

From Trace Results

Menu Bar: Tools -> New Schematic -> From Trace Results

Bind Key: O

This command generates a new schematic from the trace results. This command narrows the debugging scope by generating a partial schematic containing only the trace results. The fan-in/fan-out of this partial schematic can be continuously expanded or collapsed.

Fan-in

Menu Bar: Tools -> New Schematic -> Fan-in

This command creates a new partial flattened schematic window based on the results from tracing the fan-in cone of the selected signal.

Fan-out

Menu Bar: Tools -> New Schematic -> Fan-out

This command creates a new partial flattened schematic window based on the results from tracing the fan-out cone of the selected signal.

Driver

Menu Bar: Tools -> New Schematic -> Driver

This command creates a new partial flattened schematic window based on the results from tracing the drivers of the selected signal or instance.

Load

Menu Bar: Tools -> New Schematic -> Load

This command creates a new partial flattened schematic window based on the results from tracing the loads of the selected signal or instance.

Connectivity

Menu Bar: Tools -> New Schematic -> Connectivity

This command creates a new partial flattened schematic window based on the results from tracing the connectivity of the selected signal or instance.

Clock Tree

Menu Bar: Tools -> New Schematic -> Clock Tree

Refer to the **Tools -> New Schematic from Source ->** **Clock Tree** command description in the *nTrace* chapter.

Reset Tree

Menu Bar: Tools -> New Schematic -> Reset Tree

A primitive instance or net can be selected to create a new flattened schematic window containing the extracted reset tree results.

Rearrange Schematic (Flattened View Only)

Menu Bar: Tools -> Rearrange Schematic

This command automatically optimizes the size and placement of all instances in the current view and adjusts the placement of pins, ports, and net routing.

ECO Utility

Clone Logic to ECO Window - nECO

Menu Bar: Tools -> ECO Utility -> Clone Logic to ECO Window

This command duplicates selected logic in Non-Freeze Silicon mode. The duplicated logic retains the connectivity between each object. You can select to retain connectivity for the inputs/outputs of these logic.

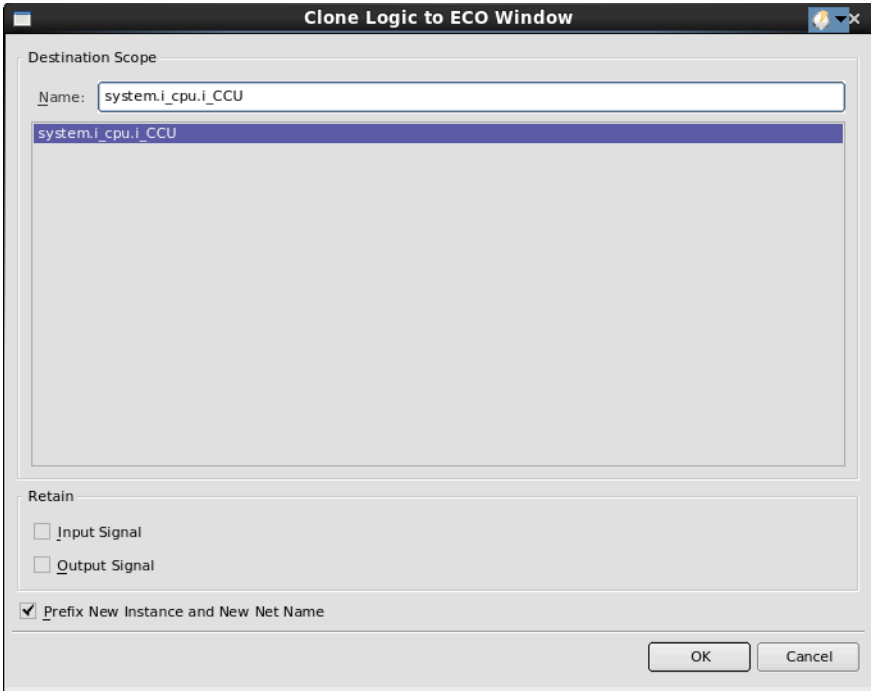


Figure: Clone Logic to ECO Window Form

The *Clone Logic to ECO Window* form includes the following options and fields:

- **Destination Scope:** Select the desired scope in the **Name** text field.
- **Input Signal**: When this option is turned *on*, the connections between input signals are retained. The default option is *off*.
- **Output Signal**: When this option is turned *on*, the connections between output signals are retained. The default option is *off*.

- **Prefix New Instance and New Net Name:** When this option is turned *on*, the name format is *prefix+"_"+original name+"_"+serial number*, that is: *eco_i_I1_7*. The default option is *on*.

Clone Module - nECO

Menu Bar: Tools -> ECO Utility -> Clone Module

This command opens the *Clone Module* form which lists all instance names of the **Original Module** to be cloned.

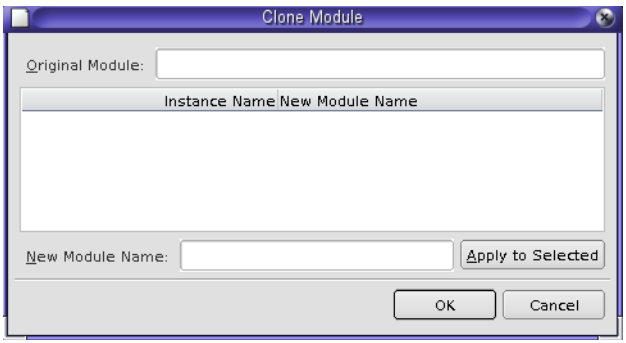


Figure: Clone Module Form - Clone the Module "ALUB" under nSchema

Click the instances under the **Instance Name** heading of the module to see if any upper scopes need to be cloned first. Select one scope segment, and the grid on the right displays the mapping module's instantiations. Input a new module name in **New Module Name** text field and enable the **Apply to Selected** option to change the name in the **New Module Name** column for the module to be cloned.

Change Instance Scope - nECO

Menu Bar: Tools -> ECO Utility -> Change Instance Scope

NOTE: The **Freeze Silicon** option on the **Schematics -> ECO -> Freeze Silicon** page of the *Preferences* form (invoked with the **Tools -> Preferences** command) must be turned *off* to enable this command.

This command opens the *Change Instance Scope* form in which the selected instance scope in a *Browser* window can be changed when keeping the instance scope connections. The selected scope can either contain single or multiple

instantiations. Enter the new scope for the selected instance by in the **New Scope** text field.

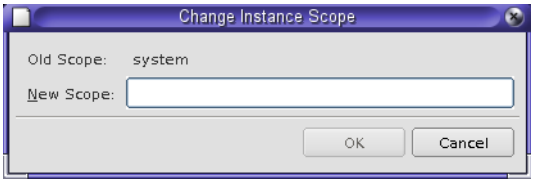


Figure: Change Instance Scope Form

Create Port - nECO

Menu Bar: Tools -> ECO Utility -> Create Port

This command creates a port after the **Module Name**, the **Port Name**, and the **Port Direction** are entered in the *Create Module Port* form.

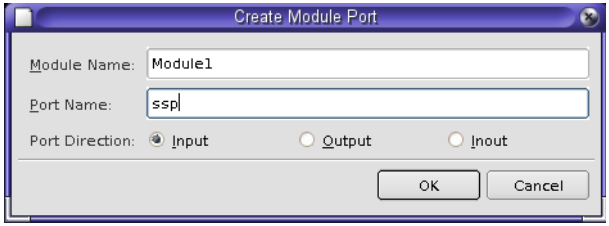


Figure: Create Module Port Form

If the specified module contains multiple instantiations, the new port is created for all the instantiations. An additional *Add Port* form is opened for specifying the new ports of the instances to be displayed in the ECO window.

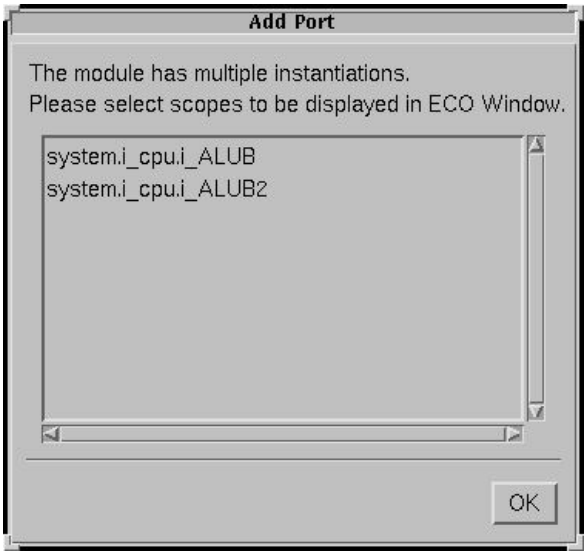


Figure: Add Port Form - Displays When Creating Port for Multiple Instantiations Scope

Cell Type Replacement - nECO

Menu Bar: Tools -> ECO Utility -> Cell Type Replacement

NOTE: The **Freeze Silicon** option on the **Schematics -> ECO -> Freeze Silicon** page of the *Preferences* form (invoked with the **Tools -> Preferences** command) must be turned *off* to enable this command.

This command replaces all the cells simultaneously.

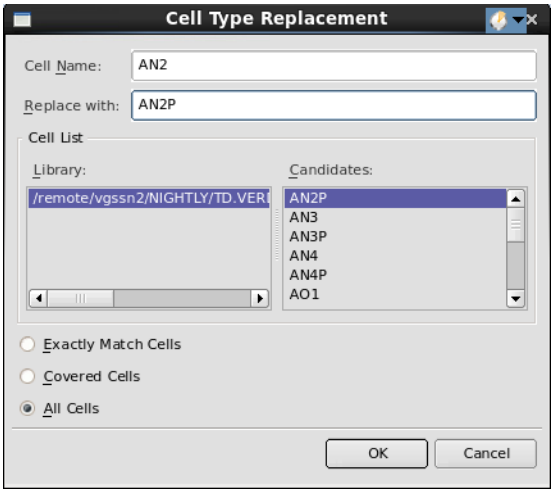


Figure: Cell Type Replacement Form

The *Cell Type Replacement* form includes the following options and fields:

- **Cell Name:** Specify the old cell name.
- **Replace with**: Browse the cell to replace from the **Cell List**.

Select one of the following options listed below to list the **Candidates** from the **Cell List**:

- **Exactly Match Cells:** Only lists the cells with port numbers, names, and direction of each port that are matched exactly with the old cell.
- **Covered Cells:** Lists the cells which (1) their port numbers are larger than or equal to the old cell and (2) each of their ports contain the same name and direction as the old cell.
- **All Cells:** Lists all cells from the library.

If the new selected cell do not exactly match the old cell after clicking **OK**, the *Port Mapping* form opens for port mapping configuration, as illustrated in the following figure:

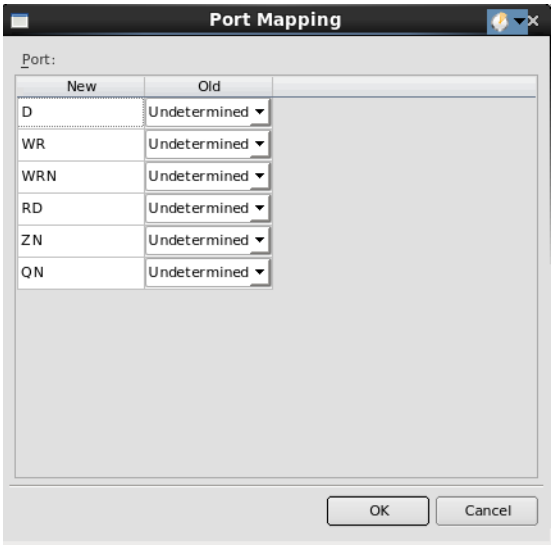


Figure: Port Mapping Form

The ports are matched in the following order:

1. Matches a port for signal input or signal output even if the name is the same.
2. Matches ports with the same name.
3. Marks unmatched ports as **Undetermined**.

The matched results can be changed. When any cell in the **Old** column is clicked, a selection field with connecting options opens for selection.

When you click **OK** again, the *Commit* form opens as follows:

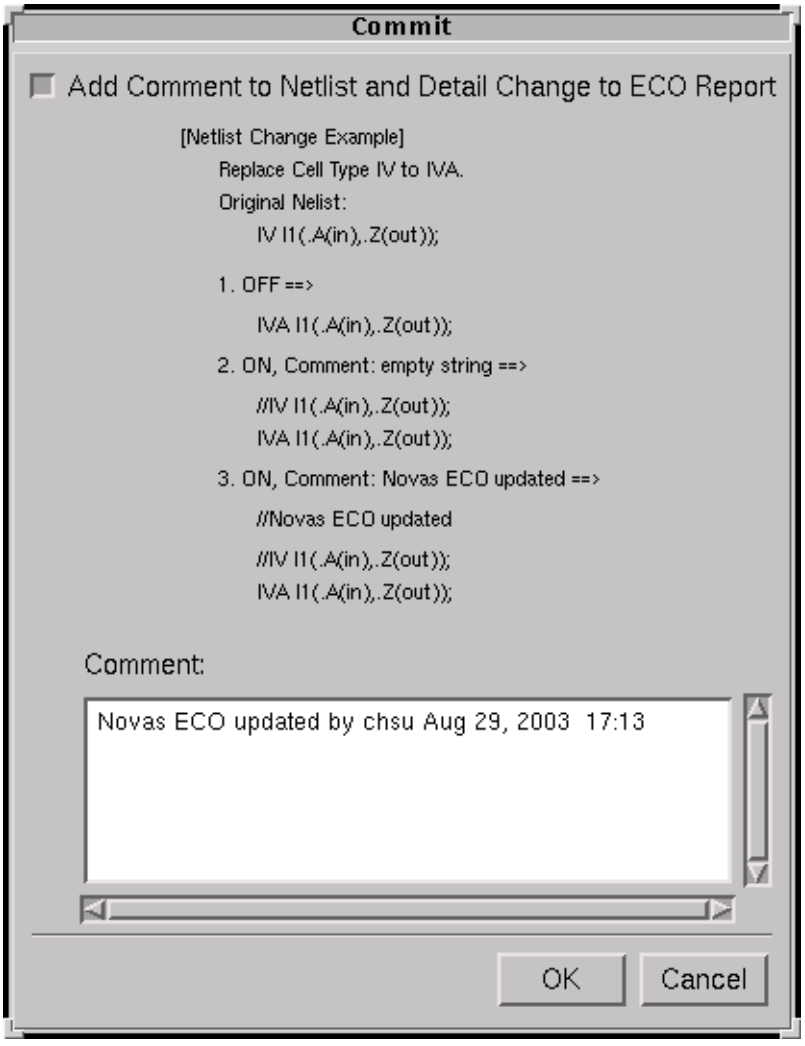


Figure: Commit Form

When the **Add Comment to Netlist and Detail Change to ECO Report** option is turned *on*, the **Comment** text field is enabled with the following features:

- Comments can be added next to netlist changes.
- Detailed changes can be logged to the **ECO Report**.

When the **Add Comment to Netlist and Detail Change to ECO Report** option is turned *off*, then the comment is not added to the netlist. In addition, ECO only logs the **Cell Type Replace** action to the **ECO Report** without logging every single line change in the netlist.

Spare Cell - nECO

Menu Bar: Tools -> ECO Utility -> Spare Cell

NOTE: The **Freeze Silicon** option on the **Schematics -> ECO -> Freeze Silicon** page of the *Preferences* form (invoked with the **Tools -> Preferences** command) must be turned *on* to enable this command.

This command specifies the spare cell instances of the design. Refer to the **Edit -> Spare Cell** command in the *nECO* window for details.

Extract Interactive FSM

Menu Bar: Tools -> Extract Interactive FSM

Refer to the **Tools -> Clock Analyzer** command in the *nTrace* chapter for details.

List User-defined Delay

Menu Bar: Tools -> List User-defined Delay

This command opens the *Edit User Define SDF Value* form where delay to nets and gates can be set and the path delay can be calculated according to the current delay value.

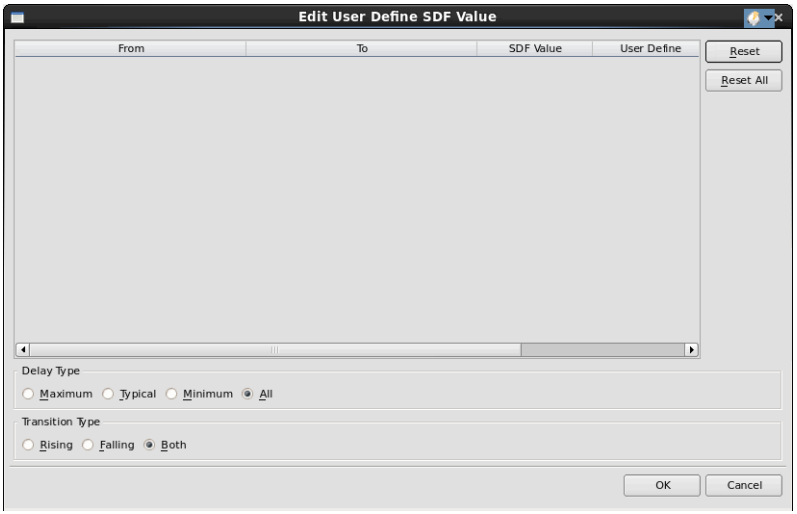


Figure: Edit User Define SDF Value Form

The *Edit User Define SDF Value* form includes the following fields:

- **Delay Type:** Specify a delay type by selecting one of the following options: **Maximum**, **Typical**, **Minimum**, or **All**.
- **Transition Type:** Specify a transition type by selection one of the following options: **Rising**, **Falling**, or **Both**.

Netlistcom Information (Full Hierarchical View Only)

Menu Bar: Tools -> Netlistcom Information

This command displays the netlistcom information to confirm if the netlistcom works accurately when starting the Verdi platform. The message which opens indicates if *nSchema* uses netlistcom data and the *on/off* status of RTL options.

Figure: Netlistcom Information Form

Preferences

Menu Bar: Tools -> Preferences

Refer to the *Schematics Folder* section of the *Preferences* chapter for details.

Customize Menu/Toolbar

Refer to the **Tools -> Customize Menu/Toolbar** command in the *nTrace* chapter for details.

Right-Click Commands

Many of the previously described commands can also be selected from the right-click menu options in *nSchema*.

nSchema Frame Right-Click Menu Options

After the right mouse button is clicked anywhere in the menu, toolbar icon, or frame banner area of the *nSchema* window or standalone window, a configuration option menu is displayed, as illustrated in the following figure. This menu can be used to configure the available icons.

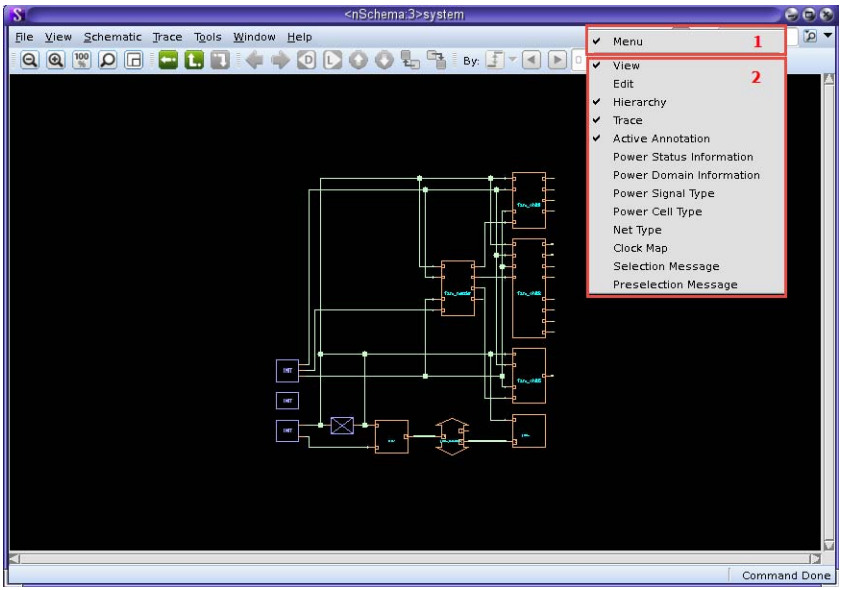


Figure: Configuration Option Menu

The **Menu** option (labeled 1 in the figure above) enables/disables the command menu options. The options in the bottom section (labeled 2 in the figure above) enable/disable the toolbar icons for different functions.

Full Hierarchical Window Right-Click Commands

The following list of right-click menu options are available for the *Full Hierarchical* view of *nSchema*. Some selections are available from the right mouse button menu options. Other selections have equivalent selections available from the toolbar menus.

Properties

This command opens the *Properties* form in which the information about the selected object is displayed.

Drag

Bind Key: Ctrl+C

This command helps to drag the selected signal to another location and dropped. This is similar to using the middle mouse button to select and drag.

Drop

Bind Key: Ctrl+V

This command helps to drop the selected signal dragged from another location in this window. This is similar to releasing the middle mouse button for a drop.

Copy Full Path

Bind Key: Ctrl+H

This command copies the complete hierarchical name(s) of the selected signal(s) into the clipboard buffer.

New ECO

The **New ECO** menu has the following two sub-commands: **ECO Window for Selected Instance(s)** and **ECO Window for All Instances** which are described as follows:

ECO Window for Selected Instance(s)

Refer to the **Tools -> ECO Window for Selected Instance(s) - nECO** command description for details.

ECO Window for All Instances

Refer to the **Tools -> ECO Window for All Instances - nECO** command description for details.

Highlight

Refer to the **Signal -> Highlight** command description for details.

Sync. Active Scope

Refer to the **Schematic -> Sync. Active Scope** command description for details.

Display Liberty Definition

NOTE: This command is hidden when the selected object is not an instance or the selected instance does not have liberty file information.

Refer to the **Display Liberty Definition** command description in the *nTrace* chapter for details.

Trace

The Trace command has the following three sub-commands: Driver, Load, and Connectivity.

Driver

Refer to the **Trace -> Driver** command description for details.

Load

Refer to the **Trace -> Load** command description for details.

Connectivity

Refer to the **Trace -> Connectivity** command description for details.

Focus Connection

Refer to the **Schematic -> Focus Connection** command description for details.

Show Focus Connection

NOTE: This command is available in full hierarchical and flattened schematic windows. It is not available in partial hierarchical windows.

This command traces the connections (driving instance to loading instance(s) with connecting net) of the selected net, port, or instance pin and summarizes the results in a tabular form. This command is similar to **Focus Connection** except that the results are displayed in a form instead of being highlighted in the schematic window.

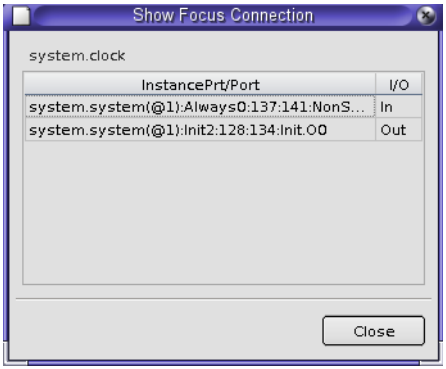


Figure: Show Focus Connection Form

If another net, port, or instance pin is selected in the current schematic window, the *Show Focus Connection* form automatically is updated to display the new connection results. If a net, port, or instance pin is selected in another schematic window, the form is not updated automatically and the command must be invoked again.

NOTE: If the **Focus Connection** command is invoked when the *Show Focus Connection* form is open, the form is cleared and the results are highlighted only in the schematic window.

Show Number of Driver/Load

These commands display the number of drivers/loads on a node. They are valid only if a net, an instance port, or an input/output port are right-clicked. This command can be used as a quick check to determine if the driver/load in *nTrace* or *nSchema* are traced.

If the total number of drivers/loads are very large, it may take a long time to generate the schematic view. When the schematic view is complex, the cause/effect relationships are not easily observed, which makes it difficult to debug. In this situation, the recommendation is to trace all the drivers/loads in *nTrace*. If the total number of drivers/loads is small, the source/destination can easily be observed in *nSchema*.

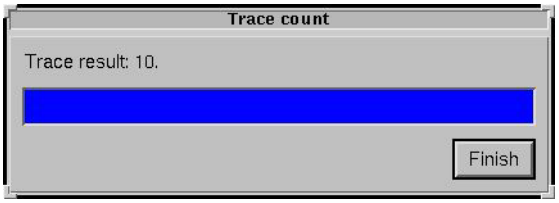


Figure: Trace Count for Number of Drivers/Loads

Add to Waveform

NOTE: This command is enabled when an FSDB is loaded and *nWave* is opened.

Refer to the **Schematic -> Add to Waveform** command description for details.

Add Locator

Refer to the **Schematic** -> **Add Locator** command description for details.

Pop View Up

Refer to the **View -> Pop View Up** command description for details.

Push View In

Refer to the **View -> Push View In** command description for details.

Cell Delay

NOTE: This command is enabled when a gate-level netlist and an SDF file are loaded.

Refer to the **Schematic -> Cell Delay - nAnalyzer** command description for details.

Change Delay

NOTE: This command is enabled when a gate-level netlist and an SDF file are loaded.

Refer to the **Tools -> List User-defined Delay** command description for details.

Display Source Code

This command opens a new *Source Code* frame to display the source code of the selected scope. The associated code is displayed at the top of the frame.

Display Detailed RTL Block

NOTE: This command is enabled when the selected RTL block is right-clicked in a schematic window and the **View -> Detailed RTL** toggle command is turned *off*.

This command extracts the detailed RTL for the selected RTL functional block and displays the results in a new *nSchema* window.

List Ports

This command displays the **Instance Port List** of a specific net or instance. After the list is opened, select any port and click **Zoom** to view the list more clearly.

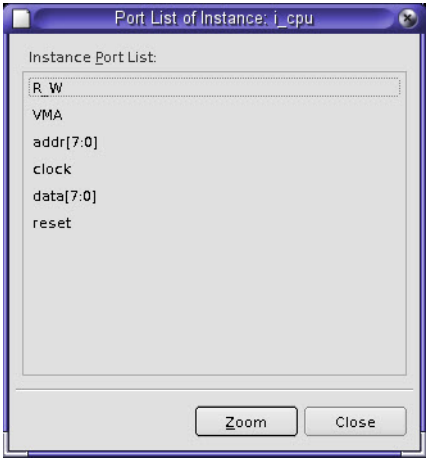


Figure: Instance Port List

Show Detailed Connection

This command displays the exact connection of all bits of a bus connected to a black box or a hierarchy module. After selecting one port, invoke this command to open the *Bus Connection* form listing all connections of the bus, bit by bit. This command supports multiple selections. If more than one port is selected, the *Bus Connection* form displays the selected ports and their connected buses.

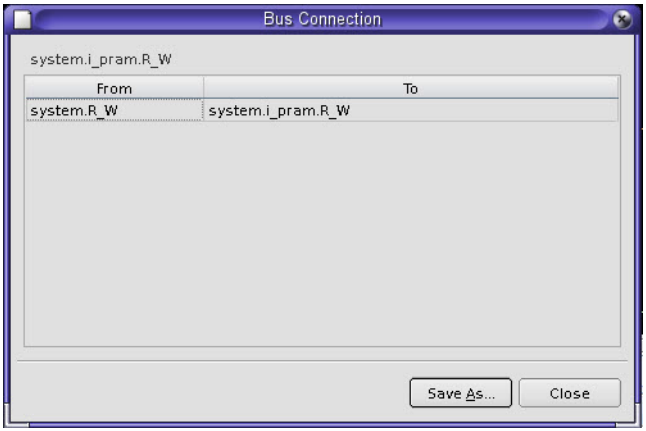


Figure: Bus Connection Form

Click **Save As** to open the *Save File* form where the connection information can be saved as a text (*.txt*) file.

This command is valid for the following four conditions:

1. A port connected to another module that contains multi-bit range.
2. The instance port of a continuous assignment in a RTL design that contains multi-bit range.
3. The instance port of the schematic block in a *Browser* window that contains multi-bit range.
4. The instance port of the connection in a *Flattened* window that contains multi-bit range.

The following figures illustrate the four conditions:

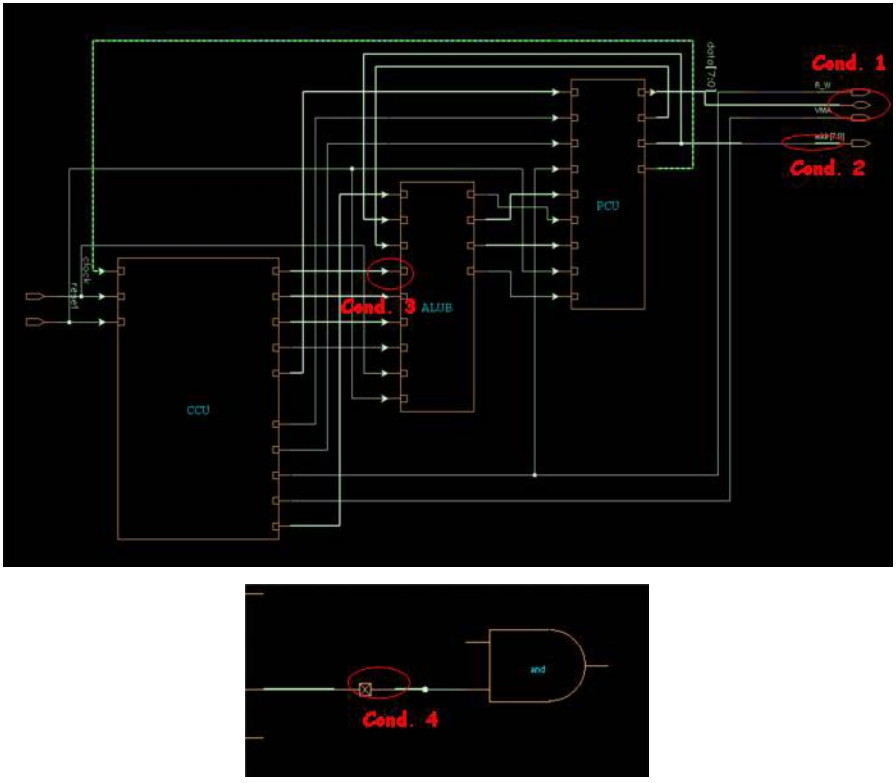


Figure: Valid Conditions for the Show Detailed Connection Command

Memory

This command opens a *Memory* window for viewing memory data stored in a database.

NOTE: This database is created by the *fsdbDumpMem* and *fsdbDumpVars* commands (in VHDL), which dump the memory data.

This command is enabled when:

1. An FSDB file containing memory data is loaded.
2. **Interactive Mode** command is on.

The command is hidden if the condition is not met.

See the *Memory/MDA Pane* chapter for details.

NOTE: If the FSDB file is imported:
1. In the case of a Verilog design, if memory components are detected, the (*nTrace*) **Tools -> Memory** and (*nWave*) **Tools -> Memory** commands are enabled.
2. In the case of a VHDL and mixed-language designs, the **Tools -> Memory** command is always enabled since (a) if a memory variable exists in the FSDB file cannot be distinguished, (b) the **nMemory -> Get Memory Signal** command cannot retrieve a VHDL format memory signal, and (c) dragging a memory signal from *nTrace*, *nSchema*, or *nWave* and dropping in *nMemory* is supported.

Set CTS Attribute Constant -> Stop/Ignore/Passthrough

These commands facilitate setting CTS attribute constants for clock domain extraction. After a signal, instance, or pin is selected and the **Set CTS Attribute Constant -> Stop, Ignore,** or **Passthrough** commands are invoked in the right-click menu option, the *Clock Extraction Settings* form opens. Refer to the *Clock Extraction Settings* form description in the *Clock Analyzer* chapter for details.

Clock Domain Known Constant -> Signal/Cell Port

These commands facilitate setting known constants for clock domain extraction. After a signal, instance or pin is selected, invoke the **Set Known Constant -> Signal** or **Cell Port** commands in the right-click menu option to open the *Extract Clock Domain* form. The selected signal, instance, or pin in *nSchema* is displayed in the form under the **Signal** or **Cell port** tabs. These commands reduce the time required to drag a signal, instance, or pin from *nSchema* and drop it in the *Extract Clock Domain* form under the **Enable Known Constant** option to the appropriate **Signal** or **Cell Port** tab.

Browser Window Right-Click Commands

The following list of right-click menu options are available for the *Browser* window (partial hierarchical view of *nSchema*). Some selections are available from the right mouse button menu options. Other selections have equivalent selections available from the toolbar menus.

Properties

Refer to the **Properties** command description for details.

Drag

Refer to the **Drag** command description for details.

Drop

Refer to the **Drop** command description for details.

Copy Full Path

Refer to the **Copy Full Path** command description for details.

New ECO

Refer to the **New ECO** command description for details.

Focus Connection

Refer to the **Schematic -> Focus Connection** command description for details.

Expand

This command expands the logic attached to the selected port.

Add to Waveform

Refer to the **Schematic -> Add to Waveform** command description for details.

Add Locator

Refer to the **Schematic -> Add Locator** command description for details.

Pop View Up

Refer to the **View -> Pop View Up** command description for details.

Push View In

Refer to the **View -> Push View In** command description for details.

Cell Delay

Refer to the **Cell Delay** command description for details.

Change Delay

Refer to the **Change Delay** command description for details.

Display Source Code

This command opens a new *Source Code* frame to display the source code of the selected scope.

Show Detailed Connection

Refer to the **Show Detailed Connection** command description for details.

Memory

Refer to the **Memory** command description for details.

Clock Domain Known Constant -> Signal/Cell Port

Refer to the **Clock Domain Known Constant -> Signal/Cell Port** command description for details.

Flattened Window Right-Click Commands

The following list of right-click menu options is available for the *Flattened* window (partial flattened schematic window). Some selections are available from the right mouse button menu options. Other selections have equivalent selections available from the toolbar menus.

Properties

Refer to the **Properties** command description for details.

Drag

Refer to the **Drag** command description for details.

Drop

Refer to the **Drop** command description for details.

Copy Full Path

Refer to the **Copy Full Path** command description for details.

New ECO

Refer to the **New ECO** command description for details.

Trace

The Trace command has the following three sub-commands: Driver, Load, and Connectivity.

Driver

Refer to the **Trace -> Driver** command description for details.

Load

Refer to the **Trace -> Load** command description for details.

Connectivity

Refer to the **Trace -> Connectivity** command description for details.

Focus Connection

Refer to the **Schematic -> Focus Connection** command description for details.

Expand Clock Tree

This command expands the clock tree. The command is available only if a pin is selected in a flattened schematic window first. Before the clock tree result is displayed, the *Clock Tree Setting* form where the Clock Tree Synthesis (CTS) pin attribute can be set is opened. If a clock tree is not found, a warning message is displayed indicating that the storage is not found.

Expand Reset Tree

This command views the results of a reset tree after a pin or instance port is selected in a flattened schematic window. The command is available only if a pin is selected in a flattened schematic window first.

If a reset tree is not found, a warning message is displayed indicating that no storage is found.

Show Focus Connection

Refer to the **Show Focus Connection** command description for details.

Show Detailed Connection

Refer to the **Show Detailed Connection** command description for details.

Show Number of Driver/Load

Refer to the **Show Number of Driver/Load** command description for details.

Collapse

This command collapses the logic associated with the selected instance pin. The command is available only if a pin is selected in a flattened *nSchema* window first.

Create Group Instance

After selecting at least two buffer or inverter instances, this command opens the *Create Group Instance* form. The selected instances are listed in the summary field and a default **Group Instance** name is created. Click **OK** to create the group (a new symbol is added to the *nSchema* window).

Remove Group Instance

After selecting a group instance symbol, this command removes the group instance and displays the individual instances.

Group Instance Information

After selecting a group instance symbol, this command opens the *Group Instance Information* form which lists the instances included in the group.

Add to Waveform

Refer to the **Schematic -> Add to Waveform** command description for details. This command is available when the FSDB is loaded and *nWave* is opened.

Add Locator

Refer to the **Schematic -> Add Locator** command description for details.

Cell Delay

Refer to the **Cell Delay** command description for details. This command is available only when an SDF file is loaded.

Change Delay

Refer to the **Change Delay** command description for details. This command is available only when an SDF file is loaded.

Delay/Level Information

After an instance is selected, this command displays the instance delay and level information in the clock tree. The information includes **Number of Instances to Leaves (Total)**, **Number of Instances to Leaves (BUF, MUX, Gated Clock Cell, Others), Current Level (Total), Current Level (BUF, MUX, Gated Clock Cell, Others), Min Level to Leaves (Total), Min Level to Leaves (BUF, MUX, Gated Clock Cell, Others), Max Level to Leaves (Total),** and **Max Level to Leaves (BUF, MUX, Gated Clock Cell, Others)**.

NOTE: This command is available in a clock tree schematic window.

Display Source Code

Refer to the **Display Source Code** command description for details.

Display Detailed RTL Block

Refer to the **Display Detailed RTL Block** command description for details.

Collect Flip-Flop

NOTE: This command is available in a clock tree or reset tree flattened schematic view; otherwise, the command is hidden.

This command groups the selected flip-flop(s) as a black box instance (compressed register symbol) in a clock/reset tree window.

Black Box Information

NOTE: This command is available in a clock tree or reset tree flattened schematic view; otherwise, the command is hidden

This command displays the detailed flip-flop list for the selected black box instance (compressed register symbol) in the *Flip Flop List* form.

Show Detailed Connection

Refer to the **Show Detailed Connection** command description for details.

Memory

Refer to the **Memory** command description for details.

Set CTS Attribute Constant -> Stop/Ignore/Passthrough

Refer to the **Set CTS Attribute Constant -> Stop/Ignore/Passthrough** command description for details.

Clock Domain Known Constant -> Signal/Cell Port

Refer to the **Clock Domain Known Constant -> Signal/Cell Port** command description for details.

Toolbar Icons and Fields

Figure: Toolbar Used in nSchema Window

The available toolbar icons may be modified. Refer to the *Toolbars* section of the *User Interface* chapter in the *Verdi User Guide and Tutorial* manual for details.

The different toolbar categories and available icons are described below.

View Category

Zoom Out

Refer to the **View -> Zoom -> Zoom Out** command description for details.

Zoom In

Refer to the **View -> Zoom -> Zoom In** command description for details.

Zoom All

Refer to the **View -> Zoom -> Zoom All** command description for details.

Minimap

Refer to the **View -> Zoom -> Minimap** command description for details.

Pan

Click this toolbar icon to pan the schematic window to the left, right, up, or down. Refer the **View -> Pan** command description for more details.

Edit Category

The icons in this category are available in flattened schematic windows.

Undo

Refer to the **Edit -> Undo** command description for details.

Redo 

Refer to the **Edit ->** **Redo** command description for details.

Remove from Viewing Objects 

Refer to the **Edit ->** **Remove from Viewing Objects** command description for details.

Hierarchy Category

Last View 

Refer to the **View ->** **Last View** command description for details.

Pop View Up 

Click this toolbar icon to generate the parent hierarchical level schematic.

Push View In 

Click this toolbar icon to generate the child hierarchical level schematic of the selected module.

Trace Category

Backward History 

Click this toolbar icon to trace back the most recent 32 trace results.

Forward History 

Click this toolbar icon to trace forward the most recent 32 trace results.

Trace Driver 

Click this toolbar icon to trace all possible drivers of a selected signal and display the results in the current *nSchema* window.

Trace Load 

Click this toolbar icon to trace all possible loads of a selected signal and display the results in the current *nSchema* window.

Trace 2 Nodes

Click this toolbar icon to open the **Trace 2 Nodes** toolbar as illustrated in the following figure in which you can specify the criteria to trace between two points. The points can be either instance ports or signals and the results are displayed in the current *nSchema* window.

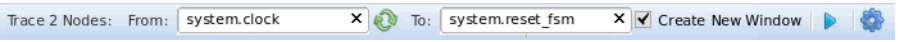


Figure: Trace 2 Nodes Toolbar

The **Trace 2 Nodes** toolbar includes the following fields and icons:

- **From:** Enter the start trace point to trace the instance port or signal.
- **To:** Enter the stop trace point to trace the instance port or signal.
- **Switch From/To Node** : Click this icon to switch the **From** details, that is instance ports or signal names to the **To** text field and vice versa.
- **Create New Window:** When this option is turned *on*, the tracing results is displayed in a new *nSchema* window. The default option is *on*.
- **Trace** : Click this icon to trace between two points specified.
- **Two Points** : Click this icon to open the *Trace Two Points* form. For more information, see the *Two Points* section.

NOTE: You can also enable this option using the **Show Trace 2 Nodes Toolbar** option on the **Schematics -> Display Options -> Schematic** page of the *Preferences* form (invoked with the **Tools -> Preferences** command).

Show Previous

Click this toolbar icon to jump to the previous trace result in the instance by highlighting (with a dashed line) the corresponding symbol in the schematic window.

NOTE: This toolbar icon is also available in the *Flattened Window* and *Hierarchical Flattened View*.

Show Next 

Click this toolbar icon to jump to the next trace result in the current instance by highlighting (with a dashed line) the corresponding symbol in the schematic window.

NOTE: This toolbar icon is also available in the *Flattened Window* and *Hierarchical Flattened View*.

Show Previous in Hierarchy 

Click this toolbar icon to jump to the previous instances of the traced results in different hierarchies.

Show Next in Hierarchy 

Click this toolbar icon to jump to the next instances of the traced results in different hierarchies.

Active Annotation Category

Search By 

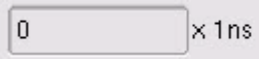
Click this toolbar icon to specify the search criteria for the value change of a signal.This icon is active only when the FSDB file is loaded and **Active Annotation** is enabled.

Search Backward 

Click this toolbar icon to search backwards in time and locate the value in the waveform window. This icon is active only when the FSDB file is loaded and the **Schematic -> Active Annotation** command is enabled.

Search Forward 

Click this toolbar icon to search forwards in time and locate the value in the waveform window. This icon is active only when the FSDB file is loaded and the **Schematic -> Active Annotation** command is enabled.

Cursor Time Entry/Display 

This text field displays the cursor time and also sets the cursor time. The field updates all back-annotated values at a specified cursor time. This field is active

only when the FSDB file is loaded and the **Schematic -> Active Annotation** command is enabled.

Power Status Information Category

Power State

This field is available when a CPF/UPF file is loaded.

Power Domain Information Category

Power Domain

This field is available when a CPF/UPF file is loaded and a node is clicked. This field displays the power domain name with the specified highlighted color.

Power Cell Type Category

Power Type

This field is available when a CPF/UPF file is loaded and a Retention/Isolation signal is selected. This field displays the power type as Retention signals (*ret)* and/or Isolation signals (*iso)*.

Selection Message Category

Selection Message

This field displays information related to the selected object. By default, the option is disabled.

NOTE: This option is enabled by default only if the **Selection Message** option in the *Preferences* form (invoked with the **Tools -> Preferences** command) is enabled.

An example of the **Selection Message** option is illustrated in the following figure:

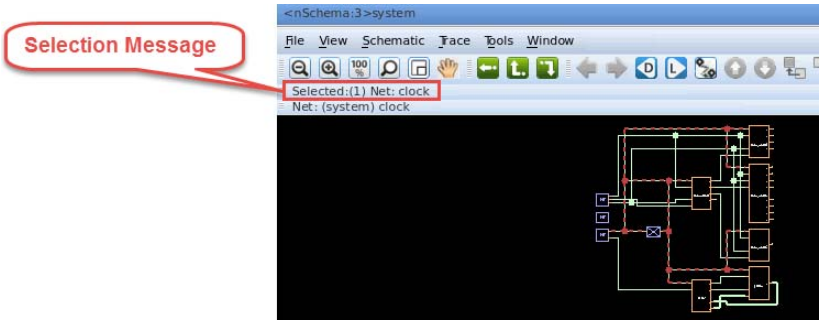


Figure: Example of the Selection Message Option

Preselection Message Category

Preselection Message

When the cursor is moved over an object, this field displays information related to the preselected object. By default, the option is disabled.

NOTE: This option is enabled by default only if the **Preselection Message** option in the *Preferences* form (invoked with the **Tools -> Preferences** command) is enabled.

An example of the **Preselection Message** option is illustrated in the following figure:

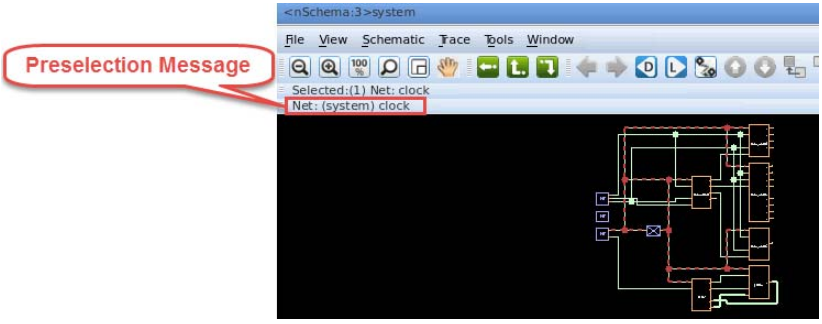


Figure: Example of the Preselection Message Option

nWave

Overview

The *nWave* window is displayed when the **Tools -> New Waveform** option is invoked from the *nTrace* menu bar or the **New Waveform** icon is clicked. The *nWave* window is docked to the same pane location as the *Message* pane as a new tab. You can make the *nWave* window a standalone window by clicking the **Undock** icon on the toolbar.

The menu bar for the *nWave* window is as shown in the following figure:

Figure: nWave Menu Bar

The pull-down menus for *nWave* are summarized and explained on the following pages. Refer to the *Icons for Dockable Panes* section for details on the menu bar icons.

Menu Option Summary

The *nWave* menu options are summarized below. All options can be double-clicked to jump to the corresponding menu option description. For right-click options, refer to the *Right-Click Options* section for details.

File Menu Options

Open
Set View Time Range
Reload
Set Active
Close
Save Signal
Restore Signal
Edit Virtual File
Recent Files ->
  [FSDB Filename]
Convert to Fast File
Extract Selected Signals
Report Selected Signals
Print
Capture Window
Close Window

Signal Menu Options

Undo/Redo
Get Signals
Get All Signals
Cut
Copy
Paste
Move
Overlay Operations
 Overlay
 Highlight Overlaid Signals
 Overlaid Signal Properties
 Unhighlight
Auto Fit
Auto Color/Pattern
Find Signal
Select ->
  Input Port
  Output Port
  Inout Port
  Net
  Register
  Error Signals
  Stuck Signals with Cursor/Marker
  Others
  All
Deselect All
Clear All
Sort Signals
Select Group Mode (Group/Signal)
Comment ->
  Insert
  Add Square Box
  Add Attached Square Box
  Add Period Box
  Add Arrow Period Box
Split Inout
Split Inout ->
  Expand as Sub-bus
  Edit Bus
  Create Bus
  Adjust Bus
  Reverse Bus
  Add Full Bus
Expand/Collapse
Collapse to Parent
Logical Operation
Event

View Menu Options

Hierarchical Name
Signal Type
Highlight Selected Signals
Values at Cursor/Marker
Leading Zeros
Display Glitch
Dense Block Drawing
All File Time Range
Compress Time Range
Clear Assertion Analysis Mask
Sort Signals by First Transition
Power ->
  Mask Power Off
  Mask Isolation
  Mask Retention
  Mask Driving Power Off
  Mask Toggle
Sort Signals by First Transition
Expand Delta ->
  Region Mode
  Expand/Collapse Time at Cursor
Grid Options
Zoom ->
  Zoom In
  Zoom Out
  Zoom All
  Zoom Cursor
  Zoom Window
Pan ->
  Pan Left
  Pan Right
  Pan Up
  Pan Down
Last View
Signal Event Report
Group Manager

Waveform Menu Options

- Auto Update
- Spacing
- Height
- Fit Waveform
- Color/Pattern
- Signal Value Radix ->
  - Binary
  - Octal
  - Hexadecimal
  - Decimal
  - ASCII
  - Enumerated Literal
  - IEEE-754 Floating Point
  - Add Alias From File
  - Add Alias From Program
  - Remove Alias
  - Edit Alias
- Signal Value Notation ->
  - Unsigned
  - Signed 2's Complement
  - Signed 1's Complement
  - Signed Magnitude
- Analog Waveform
- Digital Waveform
- Invert Waveform
- Property ->
  - Expand/Shrink Overlapping
- Classic Transaction ->
  - Expand/Shrink Overlapping
  - Display/Hide Attributes
  - Create Attribute Signals
  - Filter/Colorize
  - Clear Filtering Results
  - Clear Colorization Results
  - Keep Marker at Transaction End Time
- Classic Message ->
  - Expand/Shrink Overlapping
  - Display/Hide Attributes
  - Create Attribute Signals
  - Filter/Colorize
  - Clear Filtering Results
  - Clear Colorization Results
- Go To ->
  - Search Forward
  - Search Backward
  - Begin
  - End
  - Time
- Set Search Value
- Set Search Constraint
- Snap Cursor/Marker to Transitions
- Fix Cursor/Marker Delta Time
- Keep Cursor at Center

Waveform Menu Options (Continued)

- Waveform Time ->
  - Shift File Time
  - Shift Individual Signal Time
  - Set File Time Scale
  - Set Window Time Unit
- Marker

Analog Menu Options

Display Delta Y
Zoom Value
Vertical Fit
Auto 100% Vertical Fit
Ruler
Set Search Analog Value
Wave Slew
Drawing Style ->
  Piecewise Constant/Piecewise Linear
  Point Style
Select Analogs
Format & Precision
Average/Min/Max/RMS
Convert to Analog
Analog to Digital
Analog Expression
FFT

Tools Menu Options

New Waveform
Waveform Compare ->
  Options
  Compare 2 Signals
  Compare Selected Signals
  Compare Selected Signals to Another File
  Compare Displayed Signals
  Compare Two Groups
  Compare Signals from File
  Clear Comparison Results
Error Viewing ->
  Load File
  Select Error Types
  Display Error Mark
  Display Error Description
  Select Message-Attached Signals
Classic Transaction ->
  Evaluator
  Analysis Window
  Add Selected to Analysis Window
  Comparison Window
Register ->
  Analysis Window
  Add Selected to Analysis Window
  Comparison Window
New Schematic ->
  Trace X in Schematic
  Active Fan-in
  Bus Contention
  Contention by Time Range
Temporal Flow View
Trace X
Register
Memory/MDA
Event Sequence
List X
Property Tools ->
  Evaluator
  Statistics
  Analyzer
Toggle Coverage Report
Preferences
nCompare
Customize Menu/Toolbar

Window Menu Options

Sync All Waveforms by
  Cursor/Marker
  Horizontal Range
  Vertical Scrolling
  All
Change to Primary
Horizontal Split
Stop Split

Bind Keys

For a complete list of bind keys used in *nWave*, refer to the *nWave* section in the *Bind Key Summary* for details.

File Menu Options

Open

Menu Bar:	**File -> Open**
Toolbar Icon:	
Bind Key:	O

NOTE: The design hierarchy delimiter may be different in different simulation output files, so the correct hierarchical delimiter must be specified in the waveform preferences (specify the hierarchical delimiter in the **Hierarchical Delimiter** text field of the **Tools -> Preferences -> Waveform** page **-> Default Value** page **-> Conversion Attributes** page) before opening a file. The forward slash (/) character is the default hierarchical delimiter in *nWave*.
If the hierarchical delimiter is changed to a different value (such as "^"), any signal names containing the forward slash (`/`) character is replaced with a new character. For example, the signal name `di/dt(n14)` changes to `di^dt(n14)` in *nWave*.
This option must be set before opening the file, so the design browser pane can work properly with the **Signal -> Get Signals** option.

A waveform dump file must be opened before accessing and displaying signals. The **Open** option opens an *Open Dump File* form where one or more waveform dump files can be selected to be loaded in the *nWave* window.

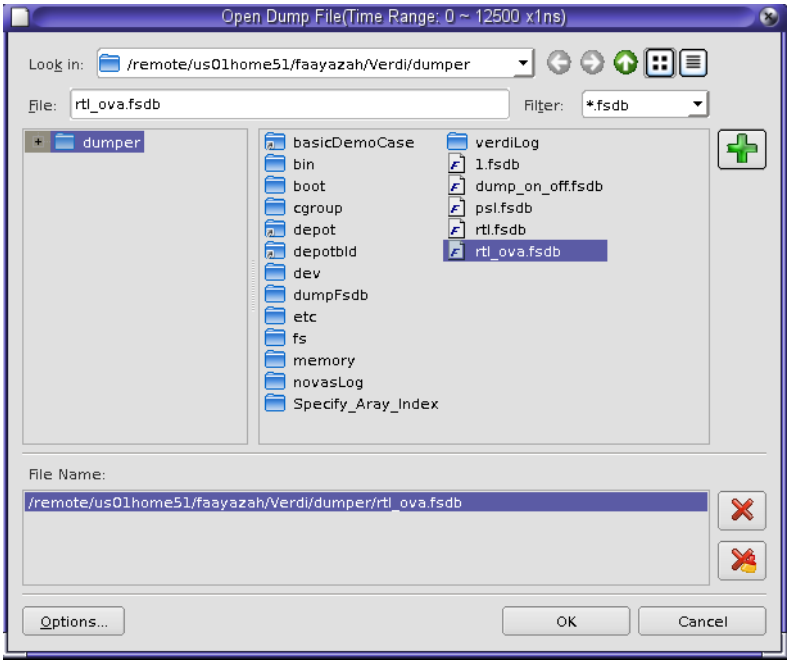


Figure: Open Dump File Form

The *Open Dump File* form includes the following fields, buttons, and options:

The *Open Dump File* form includes the following fields, buttons, and options:

- **File Name:** This field displays the FSDB files to be opened in *nWave*.

  You can select multiple files by holding the **Ctrl** key while left-clicking to select non-continuous files, or by holding the **Shift** key while left-clicking or dragging-left to select a range.

  After loading multiple waveform files in the same *nWave* window, invoke the **File -> Set Active** option to select one of the opened files as the current active file in the waveform window. The active file defines the source for the **Signal -> Get Signals** option in *nWave* and the **Active Annotation** option in *nTrace* and *nSchema*.

NOTE: Use the **Refresh** right-click option in both the directory browser and file browser to update directory and files respectively.

- **Add File:** Click this icon to add the selected FSDB file after a file is selected in the file browser.

- **Delete File:** Click this icon to delete the selected FSDB file after a file is selected in the **File Name** field.
- **Delete All Files:** Click this icon to remove all FSDB files in the **File Name** field.

NOTE: The **File -> Open** option converts the following file types to the FSDB format: SPICE Transient Analysis output files (`*.tr0`), Verilog dump files, ViewSim files, and gzipped VCD files.

- **Filter:** Select the file format filter for the file list. The default file extensions in the **Filter** field are `*.fsdb` and `*.fsdb`; `*.fsdb.gz`; `*.fsdb.bz2`; `*.vf`; `*.flst`. Additional filters can be added in this selection field using a semicolon (;) as the separator. The default filter options can be changed by changing the default in the **Open File Filter** options under the **Tools -> Preferences -> Waveform** page -> **Default Value** page -> **Display Signal** page.

  If a Verilog dump file is opened, an information form is displayed with details pertaining to the FSDB conversion. If a VCD/gzipped VCD file is opened, *nWave* converts it into FSDB format automatically and name the converted file by attaching the `.fsdb` file to the original VCD/gzipped VCD file name.

- **Options**: Click **Options** to open the *Open Dump File Options* form as illustrated in the following figure and select the following options.

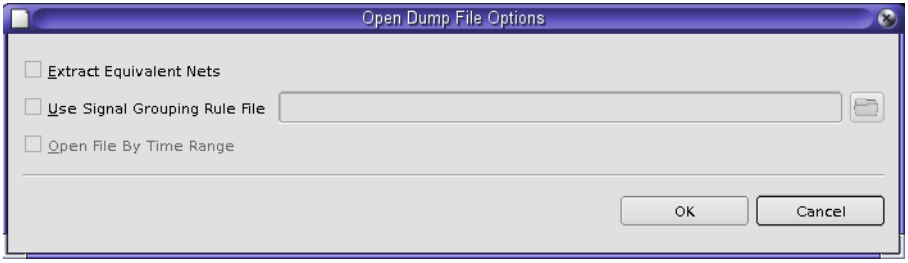


Figure: Open Dump File Options

- **Extract Equivalent Nets:** When this option is turned on, the equivalent signals of every signal stored in the FSDB file are found and added to the signal list. For example, if a signal *system.clock* is in the FSDB file, the equivalent signals `system.i_cpu.clock`, `system.i_cpu.i_ALUB.clock`, `system.i_cpu.i_CCU.clock`, and `system.i_pram.clock` are included. The waveforms associated with these signals are now available, since they are the same as *system.clock*. If this option is turned off and only *system.clock* was stored in

the FSDB file, the waveforms associated with the other signals are not available.

- **Use Signal Grouping Rule File:** When this option is turned on, you can specify a grouping rule file by either inputting the file name or by clicking the **Browse** button to view the directory structure and locating the grouping rule file. Automatic bus grouping is performed for the current session. If the option is turned on and no file is specified, no group rule is used.

  Refer to the *Specifying Signal Grouping Rules* section in *Appendix B: Customizing Verdi* for details on using the signal grouping rule file.

- **Open File by Time Range:** This option speeds up the loading process when loading a partial FSDB file. When this option is turned on, a *Time Range of Opened File* form is opened.

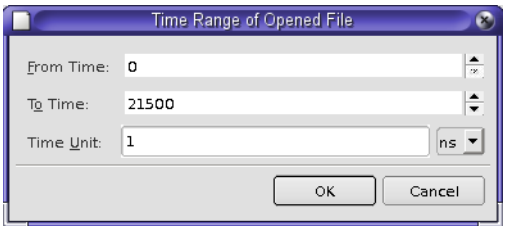


Figure: Time Range of Opened File Form

The time range of interest can be specified in the **From Time** and **To Time** text fields. Set the time unit by selecting one of the following options: fs, ps, ns, us, ms, or s. The time unit defaults to the opened file time scale. The time unit setting is not saved to the `novas.rc` resource file.

Click **OK** to import the FSDB file with the specified time range and unit, and click **Cancel** to import the FSDB file for the entire time range.

Open an SVA File

If an FSDB file with SVA signals is opened, *nWave* is able to display the property variables for them. Refer to the **Active Annotation** option description in the *nTrace* chapter for details on the property and assert objects.

For SVA, five types of property variables are available: **Cover**, **Assume**, **Assert**, **Property**, and **Sequence**. Each property variable contains a time period and a result. The time period is denoted by the begin time and the end time, and the result is denoted by the following patterns:

- **Success** (green upward arrow ): when assert condition is true or forbid condition is false.

- **Failure** (red downward arrow): when assert condition is false.
- **Match** (Vertical bar).
- **Explosion** (bomb): when forbid condition is true.

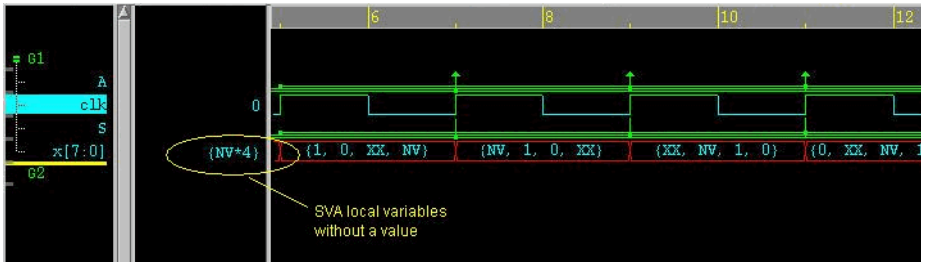


Figure: Open an SVA File in nWave

In the *nWave* waveform pane, if multiple SVA local variables without a value exist, the result is denoted as {NV*n}, where 'n' indicates the number of No Value (NV) changes. For example, if 2 non-values exist, the display shows {NV*2}.

The **Match/Success**, **No-Match/Failure**, and **Start** options are appended in the **Search By** list on the toolbar.

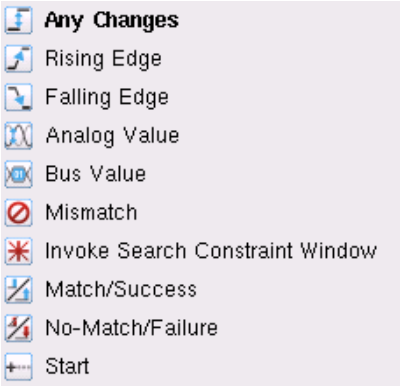


Figure: Search By on Toolbar

Search By Match/Success: Search for the **Success** (**Match**) result of **Assert**, **Forbid**, **Boolean**, **Event** and **Cover** signals.

Search By No-Match/Failure: Search for the **Failure** (**No-Match**) result of **Assert**, **Forbid**, **Boolean**, **Event**, and **Cover** signals.

Search By Start: Search for the beginning of the evaluation time for all types of property signals. **Search By Success** and **Search By Failure** are used to search for evaluation results for all types of property signals.

NOTE: The evaluation results of all property signals are also denoted as **Success** and **Failure** in the *nWave* value pane.

Open a File with Transaction or Message Information

If an FSDB file with transaction streams is opened, *nWave* is able to display the transactions as shown in the figure below.

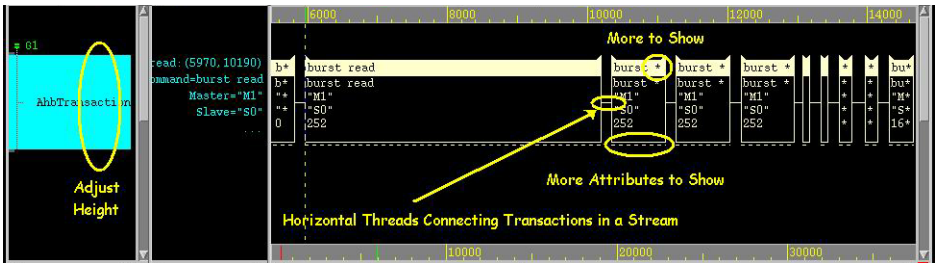


Figure: Open a File with Transaction Information in nWave

In the *nWave* waveform pane, the transaction information and the horizontal threads connecting transactions in a stream are displayed. The default number of attributes shown per transaction is *4*. If more attributes are available to be viewed, a line (**--------**) at the bottom of the transaction is shown. The line indicates that more attributes exist. The height can be adjusted accordingly. The attribute information is also displayed in the value pane. If the width of each transaction is not enough, an asterisk (*) is seen at the right end to indicate that more can be viewed.

If an FSDB file with message information is opened, *nWave* can display the messages as shown in the figure below.

Figure: Open a File with Message Information in nWave

In the waveform pane, several messages are displayed. The red tip indicates the message overlaps with other messages. The total number of overlapping messages and the order of current messages is shown as “order/total” (2/2). When a message tip is clicked, the selected message is highlighted in a specific color and the complete message contents are displayed. The color can be changed when the appropriate color attribute has been assigned and the **Tools -> Preferences -> View Options** page **-> Waveform Window** page **-> Draw Transaction/Message In Specific Color** option is turned on.

In addition, the transaction or message content can be seen on tips in both the value pane and the waveform pane.Use the **Zoom In** button to see details of the transaction or message content.

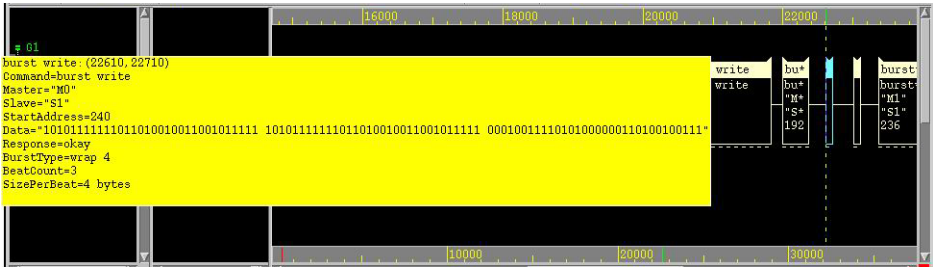


Figure: Transaction Label Information in the Value Window

The **Transaction/Message Attribute Values** options are appended in the **Search By** list on the toolbar.

Figure: Transaction/Message Search By on Toolbar

Search By Attribute Values: Search for the beginning time of the transactions or messages with one or two attribute combinations. Refer to the **Waveform** -> **Set Search Attributes** option description for details.

Transaction Synchronization

The selected transaction stream in the *nWave* window is synchronized with the *Transaction/Message Analyzer* window and vice versa when the **View -> Sync**

Active Transaction option is enabled in the *Transaction/Message Analyzer* window. Refer to the *Transaction/Message Analyzer* chapter for details.

Set View Time Range

Menu Bar: **File -> Set View Time Range**

This option is enabled when an FSDB file with a time range is opened. Use this option to specify another time range to view.

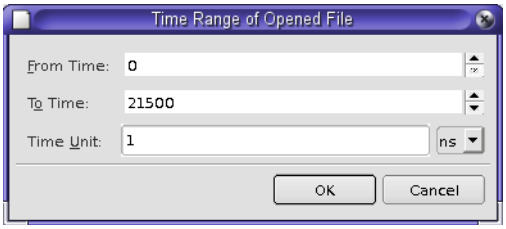


Figure: Time Range of Opened File Form

Reload

Menu Bar: **File -> Reload**

Bind Key: Shift+L

This option reloads up-to-date information from previously opened files. By default, the active FSDB file in all *nWave* frames is reloaded. Refer to the **File -> Set Active** option for details on setting the active FSDB file.

The **Reload Active File of Invoking Window Only** option under the **Waveform** folder -> **General** page in the *Preferences* form (invoked with **Tools -> Preferences**) can be enabled to only reload the active file of the invoking *nWave* window.

If the **Reload Active File of Invoking Window Only** option is turned on and the same active file is loaded in other *nWave* frames, it is reloaded in all windows even if it is not active.

For example, assume there are two *nWave* frames opened. The first *nWave* window (*nWave_1*) contains the following files: *file_A* (active), *file_B*, and *file_C*. The second *nWave* window (*nWave_2*) contains the following files: *file_B*

(active) and *file_C*. Then, the actions and associated results of using different options in *nWave* are summarized below:

Actions	Results
Invoking **File -> Reload** in *nWave_1* with the **Reload Active File of Invoking Window Only** option turned off	The active file `file_A` is reloaded in *`nWave_1`*; the active file `file_B` is reloaded in *`nWave_2`*.
Invoking **File -> Reload** in *nWave_2* with the **Reload Active File of Invoking Window Only** option turned off	The active file `file_A` is reloaded in *nWave_1*; the active file `file_B` is reloaded in *nWave_2*.
Invoking **File -> Reload** in *nWave_1* with the **Reload Active File of Invoking Window Only** option turned on	The active file `file_A` is reloaded in *nWave_1*; no file is reloaded in *nWave_2*.
Invoking **File -> Reload** in *nWave_2* with the **Reload Active File of Invoking Window Only** option turned on	The active file `file_B` is reloaded in both *nWave_1* and *nWave_2*.

NOTE: In a standalone *nWave* window (that is, opened by the **Verdi -nWave** or **nWave** option lines), the **Reload Active File of Invoking Window Only** option is located under the **General** page of the *Preferences* form (invoked with **Tools -> Preferences**).

Set Active

Menu Bar: **File -> Set Active**

Bind Key: A

NOTE: This option is enabled when at least one FSDB file is opened in *nWave*.

This option opens the *Active File* form where you can determine the active file by selecting a file name and by clicking **OK**. Alternatively, you can select the file name by double-clicking the file name. The last opened file is the active file by default.

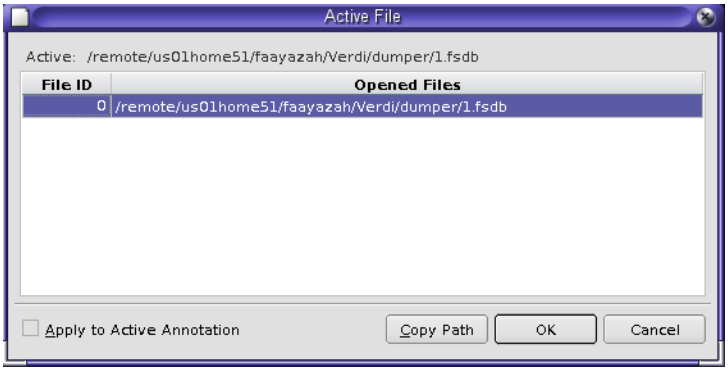


Figure: Active File Form

The active file name is shown on the banner of the *nWave* window. An active file defines the source for the **Get Signals** toolbar icon and the corresponding **Signal -> Get Signals** option. It does not affect the signals already in the display window. Each opened file is assigned a file ID as shown in the *Active File* form. This file ID is displayed in the signal pane when the **File Number** option is selected for the **Display Ahead of Signal Name** field in the **Waveform** -> **View Options** -> **Signal Pane** page of the *Preferences* form.

The *Active File* form includes the following option and button:

- **Apply to Active Annotation**: Turn this option on to allow *nTrace* or *nSchema* to synchronize the specified active file in *nWave*.
- **Copy Path**: Click this button to copy the full file path to the clipboard.

Close

Menu Bar: **File -> Close**

This option closes the selected FSDB file or all FSDB files.

From the sub-menu, choose which listed FSDB file to close or choose **All** to close all FSDB files.

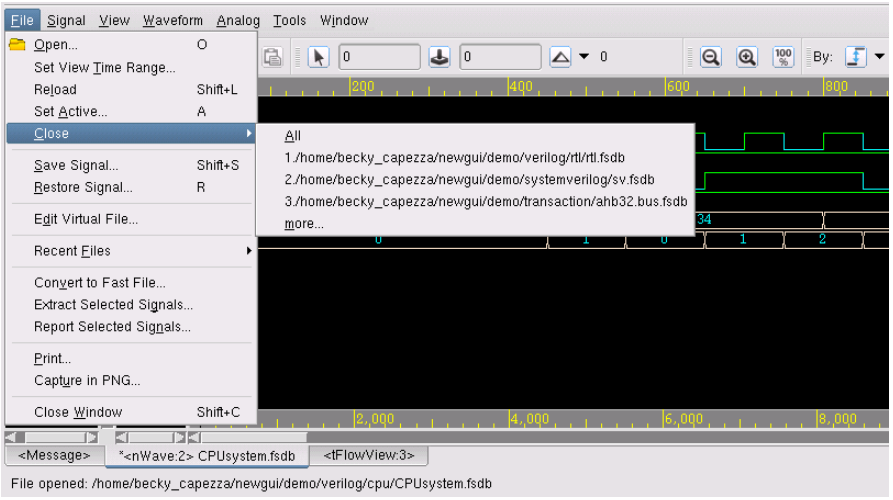


Figure: Close Option

Save Signal

Menu Bar: **File -> Save Signal**

Bind Key: Shift+S

The **Save Signal** option saves all the displayed signals and their signal attributes (such as color, height, and other information) to a file specified in the *Save Signal* form.

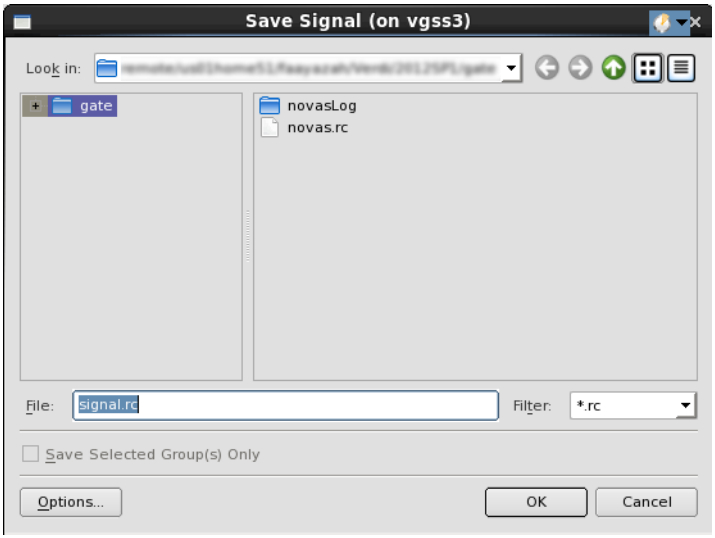


Figure: Save Signal Form

Enter the file name in the text field and click **OK**. By default, the file is saved with the default signal file name *signal* with the file extension as (`*.rc`). The signals being saved can be restored at a later time using the **File -> Restore Signal** option. If a signal file has been restored or previously saved in the same session, the save file location defaults to the last restore/save file name.

NOTE: The saved signal file must NOT be named *novas.rc* as the new name conflicts with the tool resource file.

NOTE: The default file extension (`*.rc`) is automatically added to the saved file if the extension is not specified.

Click **Options** to select the signal attributes to be saved to a signal file (`*.rc`).

Figure: Save Signal RC Attributes Form

The *Save Signal RC Attributes* form includes the following options:

- **Window Layout:** The window layout properties of the signals.
- **Cursor:** The cursor properties of the signals.
- **Marker:** The marker properties of the signals.
- **Save Alias Table:** The alias table properties of the signals. The alias table cab be saved either **By Alias Content** or **By Alias File Name**.
  - **By Alias Content:** The alias mapping is saved directly in a resource file. Modifying an alias file does not affect the alias when the resource file is restored.
  - **By Alias File Name:** The alias file name is saved and alias mapping refers to the alias file when the resource file is restored.
- **User-defined Marker:** The user-defined marker properties of the signals.
- **Digital Logical Operations Signals:** The digital logical operation signal properties of the signals.

- **Analog Logical Operations Signals:** The analog logical operation signal properties of the signals.
- **Events:** The event properties of the signals.
- **Created Buses:** The created buses properties of the signals.
- **Digital to Analog Signals:** The digital to analog signal properties of the signals.
- **Analog to Digital Signals:** The analog to digital signal properties of the signals.
- **Renamed Signals:** The renamed signal properties of the signals.
- **Comments:** The comment properties of the signals.
- **Get Signals Form with Scope Hierarchies:** The *Get Signals* form status and scope hierarchal properties in the *Get Signals* form.
- **Waveform File Information:** Includes waveform file information as comments. When this option is turned off, waveform file information is treated as comments (such as the Verdi platform only adds `;` before the keyword `activeDirFile` and `openDirFile`).

  For example, one line exists in *signal.rc*:

  ```
  activeDirFile "/" "/lx107a/RTL/verilog.dump.fsdb"
  ```

  It is treated as:

  ```
  ; activeDirFile "/" "/lx107a/RTL/verilog.dump.fsdb"
  ```

NOTE: If the `signal.rc` file is saved while the `-fsdbinfo` option is turned off and the same resource file is restored when a different waveform file is opened, a warning message may appear (stating that signals may not exist as they already existed in the original waveform files, not in the active waveform file).

- **Classic Transaction Attribute Information**: When this option is turned on, save the following transaction attribute information:
  - Comment - beginning with notation ";".
  - Active file full path - beginning with *transAttrFile*.
  - Transaction attribute information - beginning with *transAttrInfo*.
  - Attribute Name, Data_type, and Radix/Alias setting with *Radix* or *Alias*.

  The default is *off*.
- **Do Not Save Bit Range of Natural Bus**: When this option is turned on, only MDA and created/runtime buses are saved; the bit range of natural

buses (that is. signals with the same range dumped in the FSDB file) are not saved. When this option is turned off, the bit range of buses including natural buses are saved. The default is *off*.

- **Save Scope with Macro**: When this option is turned on, the scope in the *signal.rc* configuration file is replaced with the macro (virtual top) specified in the text field. A scope/signal can be dragged and dropped to the text field. The default is *off*.
- **Absolute Path:** When this option is turned on, save a signal file with the full file path related to the current directory. It is only effective when the **Waveform File Information** option is also turned on. The default is on.

  For example, under the */workdir* directory, the file `verilog.fsdb` -> `12345.fsdb` is loaded:

  a. If both **Resolve Symbolic Link** in **Tools-> Preference-> Waveform -> General** page and **Absolute Path** in the *Save Signal RC Attributes* form are turned *on*, the file name is resolved to "" */workdir/*`12345.fsdb`.

  b. If **Resolve Symbolic Link** is turned on and **Absolute Path** is turned off, the file name is resolved to `/workdir 12345.fsdb`.

  c. If **Resolve Symbolic Link** is turned off and **Absolute Path** is turned on, the file name is resolved to "" `/workdir/verilog.fsdb`.

  d. If both **Resolve Symbolic Link** and **Absolute Path** are turned off, the file name is resolved to `/workdir verilog.fsdb`.

If the **Select All** button in **Options** is clicked, all the attributes are turned on. To save only the specified groups when multiple groups exist in the signal pane, turn the **Save Selected Group(s) only** option on in the *Save Signal* form.

Refer to the *signal.rc* section in *Appendix B: Customizing Verdi* for details.

Restore Signal

Menu Bar:	**File -> Restore Signal**
Bind Key:	R

This option opens the *Restore Signal* form where signals that were previously saved using the **Save Signal** option can be restored. Select the file from the file name list (files usually have a *.rc* file extension) and click the **OK** button. One or more resource files can be selected in the form.

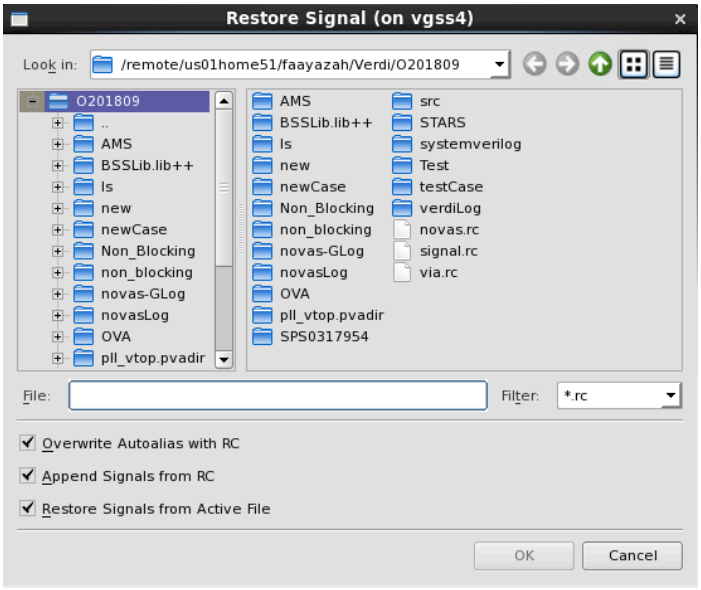


Figure: Restore Signal Form

nWave displays the signals saved in the file.

The following fields and options are available:

- **Overwrite Autoalias with RC:** When this option is turned on, it overwrites the autoalias of the restored signals included with the resource file (`*.rc`).
- **Append Signals from RC:** When this option is turned *on*, all the signals are added with the existing signals included with the resource file (`*.rc`). When this option is turned *off*, all the existing signals available in the *nWave* window are deleted, and then the signals included with the resource file (`*.rc`) are added.
- **Restore Signals from Active File:** When this option is turned on, it directly restores the signals saved in the resource file (`*.rc`) regardless of the currently opened dump file in *nWave*. In other words, the restored FSDB file is set as the active file instead of the currently opened FSDB file. Selecting one or more resource files in the form is allowed.

Refer to the *signal.rc* section in *Appendix B: Customizing Verdi* for details.

Edit Virtual File

Menu Bar: **File -> Edit Virtual File**

This option opens a *Virtual File Editor* form where the existing virtual FSDB file can be edited or a new virtual file (*.vf*) can be created. The virtual file is in text format.

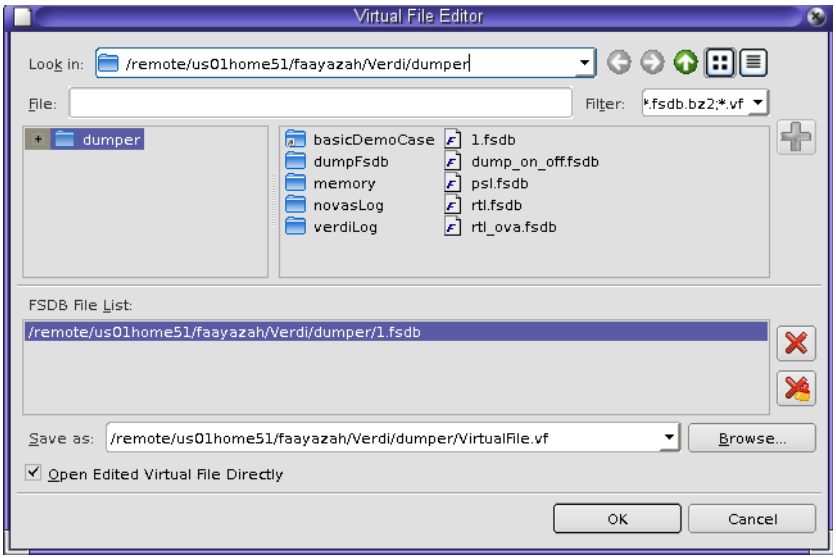


Figure: Virtual File Editor Form

The following fields and options are available:

- **FSDB File List:** This section lists the FSDB files to be included in the virtual file. FSDB files can be added by selecting the files from the file list and then click **Add File** icon. An FSDB file can be deleted by selecting a file in the **FSDB File Name** section and then click **Delete File** icon. The **Delete All Files** icon removes all listed FSDB files whether they are selected or not. When the **FSDB File List** section is empty, then **Delete File** icon, **Delete All Files** icon, and **OK** is disabled.
- **Filter:** This is the file format filter for the file list. The default file extension is `*.fsdb`; `*.fsdb.gz`; `*.fsdb.bz2`;`*.vf`. Additional filters can be added in this selection field using a semicolon (`;`) as the separator.
- **Save as:** Specify the name of the virtual file in which to save the FSDB files. The file name can be entered in the **Save as** field directly or click the **Browse** button to open the *Virtual File Browse* form for specifying a

different path and file name. Click **OK** to save the file name and close the *Virtual File Browse* form or click **Cancel** to exit without saving the file name.

- **Open Edited Virtual File Directly:** When this option is turned on, the specified virtual file is opened in *nWave* directly after you click **OK**. The default is *on*.

After the virtual file is specified, click **OK**. The virtual file is then created and the *Virtual File Editor* form is closed.

The created virtual file is displayed in purple on the *Open Dump File* form opened by invoking the **File -> Open** option in *nWave*. The created virtual file can be selected on the *Open Dump File* form. After you click **OK**, the file is opened in *nWave*.

The virtual FSDB file can also be loaded by invoking the **File** -> **Load Simulation Results** option in *nTrace*.

Virtual File Format

An example of the created virtual file is as follows:

```
@FSDB rc file Version 1.0
[VRTL_FILE_HEADER]
# !! DON'T EDIT [VRTL_FILE_HEADER] SESSION !!
Version = 1
[VRTL_FILE_SOURCE]
FileType = stitch
File1 = /verdi/home/verilog/rtl/rtl_0_100ns.fsdb
File2 = /verdi/home/verilog/rtl/rtl_100_200ns.fsdb
File3 = /verdi/home/verilog/rtl/rtl_200_300ns.fsdb
```

The FileType values include **stitch** and **split**.

- **stitch:** Merge waveforms for the same signal from the files.
- **split:** Do not merge waveforms even when the signal name is the same.

The created virtual file can be modified (such as modify the FileType field or modify the file list).

For cases where the composed files do not have the same top hierarchy, the hierarchy can be adjusted as in the following example:

`1.fsdb` -> contains hierarchy `top.sub_sys1.sig.sigc...`

`2.fsdb` -> contains hierarchy `sub_sys1.sig.sigc...`

`3.fsdb` -> contains hierarchy `sig.sigc...`

To allow all files to have the same top scope as `1.fsdb`:

`2.fsdb` -> `top.sub_sys1.sig.sigc...`

`top` scope needs to be added to `2.fsdb`

`3.fsdb` -> `top.sub_sys1.sig.sigc...`

`top.sub_sys1` scope needs to be added to `3.fsdb`

Then, `VRTL_FILE_ADD_PRESCOPE` can be added, similar to the following:

```
@FSDB rc file Version 1.0
[VRTL_FILE_HEADER]
# !! DON'T EDIT [VRTL_FILE_HEADER] SESSION !!
Version = 1
[VRTL_FILE_SOURCE]
FileType = split
File1 = ./1.fsdb
File2 = ./2.fsdb
File3 = ./3.fsdb
[VRTL_FILE_ADD_PRESCOPE]
ScopeSeparator = "/"
File2 = /top
File3 = /top/sub_sys1
```

Recent Files

Menu Bar: **File -> Recent Files**

This option displays the names of the most recently used files. Five or fewer file names can be displayed.

Convert to Fast File

Menu Bar: File -> Convert to Fast File

This option opens a *Convert to Fast File* form where a waveform dump file can be selected to convert into *nWave* FSDB format. The FSDB file is a compact binary file that can be accessed in the same way as other dump files. An FSDB file displays signals using the **Signal -> Get Signals** option.

To convert a dump file, select a file and then click **OK**. *nWave* submits a background process to perform the conversion task and informs users when the conversion is complete. By default, the dump file extensions in the **Filter** field are `*.vcd;*.out;*.tr0;*.xp;*.raw;*.wfm`. The name of the converted fast file is appended with `.fsdb`.

Extract Selected Signals

Menu Bar: **File -> Extract Selected Signals**

This option, which provides a GUI-based mechanism to perform *fsdbextract*, is available when one or more signals that belong to single or multiple FSDB files are selected in the signal pane. When signals of interest are selected in the signal pane and this option is invoked, the *Extract Selected Signal - fsdbextract* form opens. Type in the various text fields and select the appropriate options as described below, then click **OK** to extract the selected signals to the specified file and wait for the extraction to finish. If any errors or warnings are encountered during the extraction, the FSDB extractor displays messages on a message form. Refer to the *fsdbextract* section of the *Utilities* chapter for details.

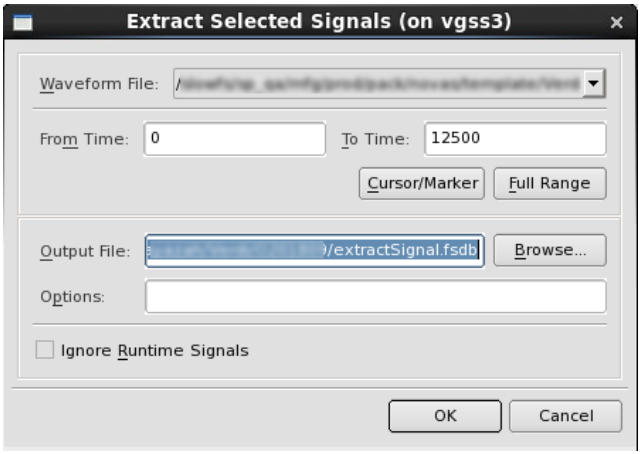


Figure: Extract Selected Signals - fsdbextract

The *Extract Selected Signal - fsdbextract* form includes the following fields and buttons:

- **Waveform File:** This field displays the waveform files of selected signals. When a waveform file is selected and the **OK** button is clicked, *nWave* skips all signals which do not belong to the selected file and display a warning message.
- **From Time/To Time:** These text fields define the time range. The default is the full range of the current form. The time range can be specified by one of the following methods:
  - Typing values directly into the **From Time** and **To Time** text fields.

    - Click the **Cursor/Marker** button to obtain the range between the **Cursor** and **Marker** in the current form.
    - Click the **Full Range** button to obtain the full range associated with the current form. This is the default option.
- **Cursor/Marker:** Click this button to obtain the **From Time** and **To Time** values from the **Cursor** and **Marker** in the current waveform window. *nWave* automatically assigns the smaller value to the **From Time** text field and the larger value to the **To Time** text field.
- **Full Range:** Specify the full range associated with the current form.
- **Sample Resolution:** This field is available when an HSpice format file (or an FSDB file generated from HSpice) is loaded. Specify the resolution of the HSpice format file in the text field and set the unit in *fs, ps, ns, us, ms,* or *s*. When extracted signals are in HSpice format, the **Options** field is disabled when the **Sample Resolution** field is enabled. The default is *1.0*, and the default unit is *ns*.
- **Output File:** Specify the target output FSDB file name or click the **Browse** button to locate an existing file. By default, the name of the output file is saved as *extractSignal.fsdb*.
- **Options:** Specify other options for extracting selected signals. Refer to the *fsdbextract* section of the *Utilities* chapter for all available options (all options are supported except the **-h** | **-help** option).
- **Ignore Runtime Signals:** When this option is turned on, runtime signals are ignored when extracting the selected signals. The default is *off*.

Click the **OK** button to perform the extraction and close the form (*nWave* waits for the extraction to finish).

NOTE:

1. If a selected signal is a member of a bus or a record, *nWave* extracts the entire bus or record from the FSDB file.
2. Runtime scalar and vector signals are supported.Data Expansion computed signals are not supported.

Report Selected Signals

Menu Bar: **File -> Report Selected Signals**

This option, which provides a GUI-based mechanism to perform *fsdbreport*, is available when one or more signals that belong to single or multiple FSDB files are selected in the signal pane. When this option is invoked, the *Report Selected*

Signals - fsdbreport form opens. Type in the various text fields and specify appropriate options as described below, then click **OK** to report the selected signals in the specified (*.frpt*) file. If any errors or warnings are encountered during the reporting, warning messages are displayed on a message form. Refer to the *fsdbreport* section of the *Utilities* chapter for details.

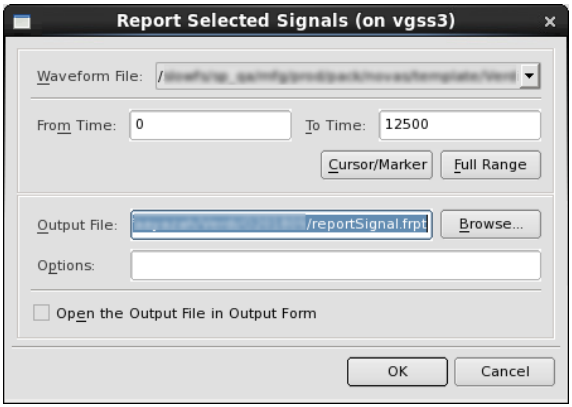


Figure: Report Selected Signals - fsdbreport

The *Selected Signals - fsdbreport* form includes the following fields and buttons:

- **Waveform File:** This field displays the waveform files of selected signals. When a waveform file is selected and the **OK** button is clicked, *nWave* skips all signals which do not belong to the selected file and display a warning message.
- **From Time / To Time:** These text fields define the time range. The default value is the full range of the current form. The time range can be specified by one of the following methods:
  - Typing values directly into the **From Time** and **To Time** text fields.
  - Click the **Cursor/Marker** button to obtain the range between the **Cursor** and **Marker** in the current form.
  - Click the **Full Range** button to obtain the full range associated with the current form. This is the default option.
- **Cursor/Marker:** Click this button to obtain the **From Time** and **To Time** values from the **Cursor** and **Marker** in the current waveform window. *nWave* automatically assigns the smaller value to the **From Time** text field and the larger value to the **To Time** text field.
- **Full Range:** Specify the full range associated with the current form.

- **Output File:** Specify the target output file name or click the **Browse** button to locate an existing file. By default, the name of the output file is saved as *reportSignal.frpt*.
- **Options:** Specify other options for reporting selected signals. Refer to the *fsdbreport* section of the *Utilities* chapter for all available options (all options are supported except **-h** | **-help**).

Click the **OK** button to generate an ASCII text file of value change lists for the specified signals.

Print

Menu Bar: **File -> Print**

This option prints the displayed signals to a printer or to a file. The option opens an *nWave Print* form which includes the **Basic**, **nWave Options**, and **Signature** tabs.

Basic Tab

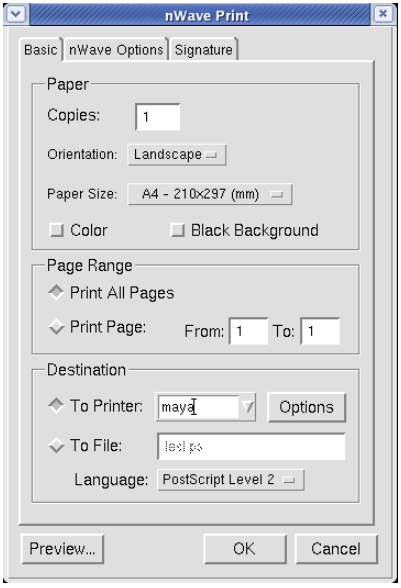


Figure: Print Form - Basic Tab

The **Paper** section of the **Basic** tab includes the following options:

- **Copies:** Enter the number of copies to be printed.

- **Orientation:** Specify the orientation for the printed page by selecting either **Landscape** or **Portrait**.
- **Paper Size:** Select the paper size of the printed page from the supplied list.
- **Color:** Select this option to print multiple colors on a color printer. Otherwise, the printout is black and white on a color printer.
- **Black Background:** When this option is turned off, the background color is white. When this option and the **Color** option are both turned on, the background color is black (this setting is generally used when viewing in the postscript file viewer). When this option is turned on and the **Color** option is turned off, the background color is gray.

The **Page Range** section of the **Basic** tab includes the following options:

- **Print All Pages:** Select this option to print the entire file.
- **Print Page:** Select this option to print a specific page range from the file.

The **Destination** section of the **Basic** tab includes the following options:

- **To Printer:** When this option is turned on, the file prints on the printer specified in the text field.
- **Options:** Click this button to configure print options. First, enter the name of the printer in the **Printer Name** field. Next, specify the formats supported by this printer by selecting one of the following: **Postscript Level 1**, **Postscript Level 2**, **HP GL/2**, or **MIF**. Next, select the **Paper Size** supported by the printer. If the **User-defined** option is chosen under **Paper Size**, the width and height of the paper must be specified.

  In the **Print Command and Options** section, the default **Command** is *lpr*, the default **Destination** is *-P*, and the default **Copies** is *-#*. Use these values for printing the file. Change them if a user-specific print option exists.
- **To File:** This option saves the print file to the working directory in postscript format. The file name must be specified in the text field.

Click the **Preview** button to display each page as it looks when printed. Use the **Prev Page** and **Next Page** buttons to traverse through the pages, and use the **Zoom In** and **Zoom Out** buttons to browse the current preview page. In addition, click the **Refresh** button to make the signals to be printed fit the page. That is, if only a few signals are selected to be printed, the heights of the signals are enlarged to fill the page.

nWave Options Tab

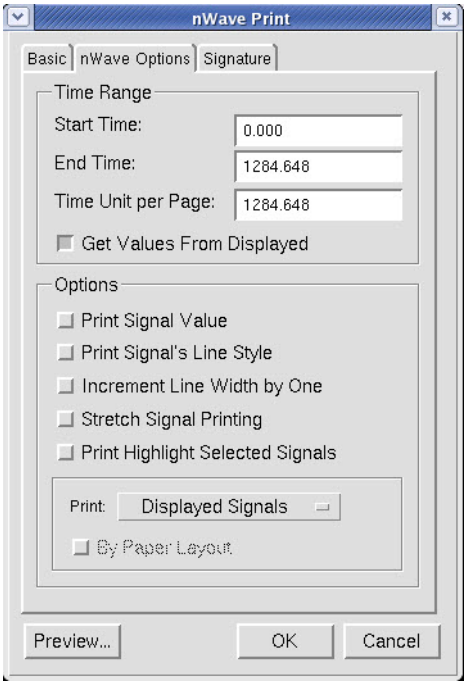


Figure: Print Form - nWave Options Tab

The **Time Range** section of the **nWave Options** tab includes the following fields and option:

- **Start Time:** Specify the beginning of the time range.
- **End Time:** Specify the end of the time range.
- **Time Unit Per Page:** Specify the time unit associated with every page.
- **Get Values From Displayed:** When this option is turned on, it is not necessary to specify values for the **Start Time**, **End Time**, or **Time Unit Per Page** fields as *nWave* automatically obtains these values from the displayed waveform.

The **Options** section the **nWave Options** tab includes the following options:

- **Print Signal Value:** When this option is turned on, the value pane is shown in the print output.
- **Print Signal's Line Style:** Each signal's line style is printed on the left of the signal pane. Before invoking the **Print** option, change the line style associated with a signal. Select the signal and invoke the **Change Color/**

Pattern option (press the bind key **C**) to bring up the *Change Color/Pattern* form to view and modify the line style.

- **Increment Line Width by One:** When this option is turned on, each signal's line width increases by one. For some printers (especially color or ink jet printers), the default line width may be too thin to print. In these cases, turning this option on results in a more legible document.
- **Stretch Signal Printing:** When this option is turned on, the signals to be printed are stretched out vertically to occupy the entire page where the waveforms can be seen more clearly.
- **Print Highlight Selected Signals**: When this option is turned on, the signals that are highlighted in the waveform pane are indicated as such in the print output.
- **Print:** Specify the signals to be printed by selecting one of the following options: **Displayed Signals**, **All Signals (Time First)**, **All Signals (Signal First)**, and **Selected Signals**.
  - **Displayed Signals**: Only prints the signals that are shown in the current window and the time range specified in the **Time Range** section.
  - **All Signals (Time First):** Print all signals in the time range specified in the **Time Range** section, prioritized according to the time range. This option prints a horizontal section first when the window is divided into sections.
  - **All Signals (Signal First):** Prints all signals in the time range specified in the **Time Range** section, prioritized according to signal order. This option prints a vertical section first when the window is divided into sections.
  - **Selected Signals:** Prints the selected signals only.

Signature Tab

The **Description** section of the **Signature** tab includes the following fields:

- **Header:** Specifies the information to be presented on the top of the printout. By default, the header includes the file name (`%f`) that is currently being viewed in the *nWave* window, the user name and host name (%h), and the date and time (%t).
- **Footer:** Specifies the information to be presented on the bottom of the printout.

Capture Window

Menu Bar: **File -> Capture Window**

This option opens the *Capture Window - Preview* form where an image in PNG (Portable Network Graphic format, a GNU standard similar to GIF) can be output.

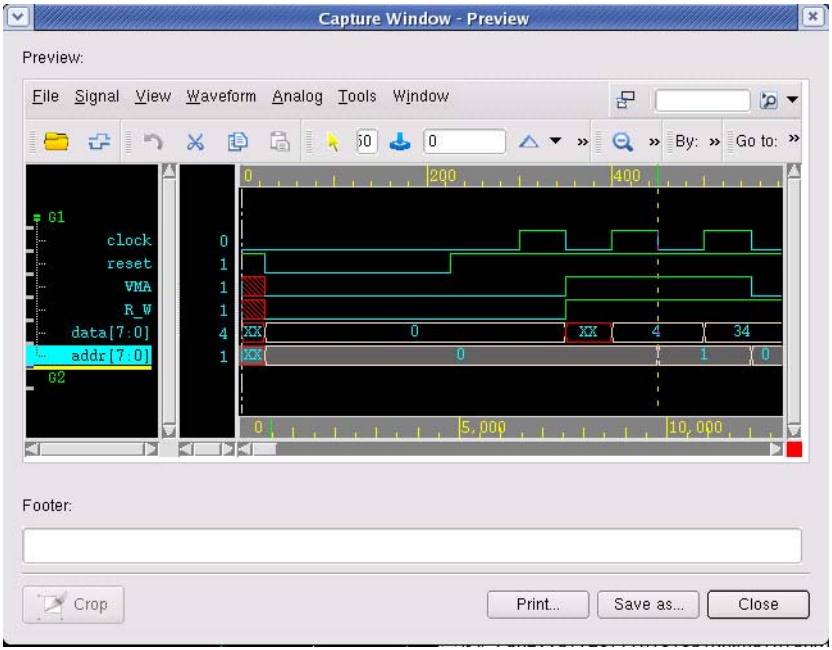


Figure: Capture Window - Preview Form

The *Capture Window - Preview* form includes the following fields and buttons:

- **Footer:** Fill in the appropriate text to be shown on the footer.
- **Crop:** Drag the left mouse button to select the desired area and then click the **Crop** button to update the display in the *Capture Window - Preview* form.
- **Print:** This button opens a *Print* form where the desired print options for creating a hard copy can be specified.
- **Save as:** This button opens a *Save As* form where the directory structure can be viewed and a file name for the PNG file can be specified.

Close Window

Menu Bar: **File -> Close Window**

Bind Key: Shift+C

This option closes the current *nWave* window.

Signal Menu Options

Undo/Redo

Menu Bar: **Signal -> Undo/Redo**

Toolbar Icon: 

This option toggles between undoing and redoing the previous action in the signal pane.

Get Signals

Menu Bar: **Signal -> Get Signals**

Toolbar Icon: 

Bind Key: G

This option opens the *Get Signals* form where signals to be added to the waveform view can be selected.

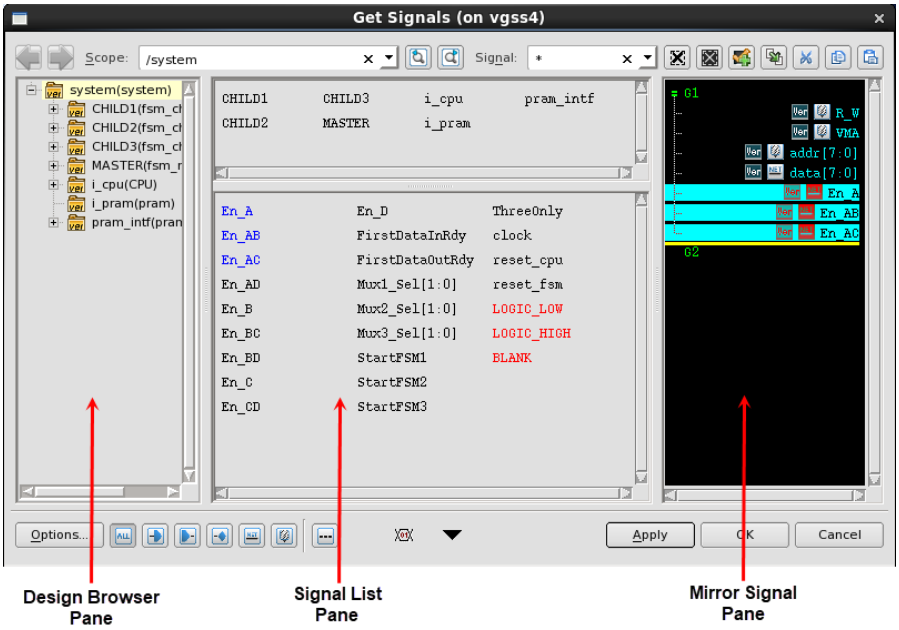


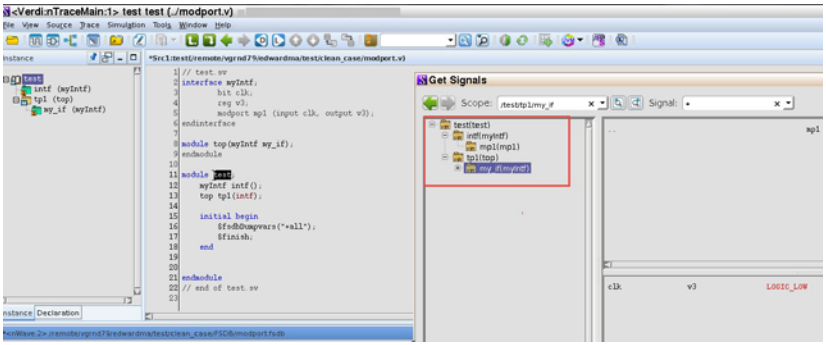
Figure: Get Signals Form

In this form, bus rules can be entered by selecting the drop-down arrow and entering the bus name in the text box. The user-defined buses are then immediately created and placed in *nWave*.

The *Get Signals* form includes the following: the **Design Browser** pane, the **Signal List** pane, the **Mirror Signal** pane, the **Scope** field and the **Signal** field.

NOTE: The interface port and modport is dumped to FSDB by default. In the hierarchy pane, the name of the interface port and modport is used as the name of scope. If the scope is selected, the signal under the interface

port and the port under the modport can be viewed.



Design Browser Pane
This pane shows the design hierarchy tree. A white highlighted rectangle indicates that the design scope has been selected and the scope's signals are currently being displayed in the signal list. The current scope is also shown in the **Scope** text field. To traverse to a different hierarchy, click the scope of interest in the design hierarchy section.

Signal List Pane
This pane displays all signals for the current design scope (highlighted with a white rectangle in the design hierarchy section). A subscope table is on the top. When you click a scope in the subscope table, it makes it easier to select the desired signals. Right-click in the subscope table to display the upperscope. To select signals in the signal list, left-click an individual signal or drag-left over a range of signals to select them. Alternatively, double-click the signal to immediately add it to the mirror signal pane. If too many signals exist in the current scope to be selected in the signal list, type the signal name in the **Signal** filter text field (the file extensions include "*" and "*.*nTX*"; the default file extension is "*"). The wildcard character is supported to filter unwanted signals. For example, type "*clk**" to select all signals beginning with *clk*.

The signals are displayed in different colors: black, blue, red, and brown.

- **Black:** not yet displayed in the *nWave* window.
- **Blue:** currently displayed in the *nWave* window.
- **Red:** constant; that is, predefined by *nWave*, such as LOGIC_LOW, LOGIC_HIGH, and BLANK (a dummy signal for separation).

- **Brown:** complex structure signals (or array of records, or MDAs).

NOTE: Signals that are expanded by Siloti's Data Expansion engine is also listed using a different color than signals that already exist in the FSDB. The expanded signal color can be changed with the `ComputedAnnotColor` key in the *novas.rc* resource file.

Interface ports and modports are dumped to FSDB by default. The dumped interface ports and modports can be viewed.

When adding signals, the view in the *nWave* window scrolls to yellow cursor line so that you need not reposition the *nWave* window view every time you add signals.

Mirror Signal Pane

Use this pane to arrange signals into groups and set the preferred signal display order before applying these signals to the waveform window for display. The signal cursor bar indicates the insertion point for adding new selected signals and/or the signals to be moved. Move the signal cursor bar with the middle mouse button.

Scope List Pane

The **Scope List** pane displays the scope highlighted in the design hierarchy section.

Enter the scope name in the **Scope** text field. The wildcard characters (* or ?) are supported to specify the pattern-matching scopes. For example, type “*sys1*” to select all the scopes which contains "sys1" in the scope name. The scopes which contain *sys1* are highlighted and the scope tree scrolls to the last matched scope, as illustrated in the following figure:

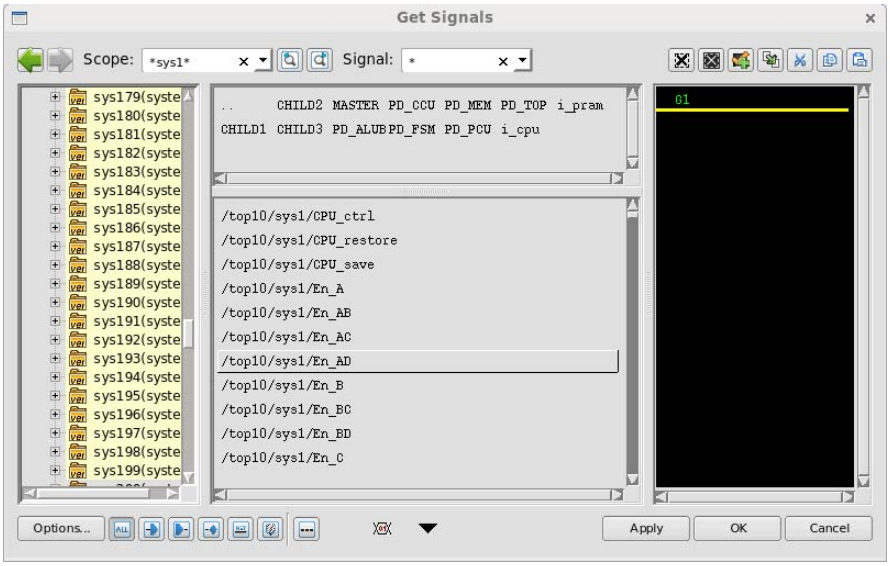


Figure: Example of Scope Filter Text Field

Click the **Search Previous**  icon to find the previous matching scope or click the **Search Next**  icon to find the next scope.

Find Signal Field

This field serves four functions: acts as a filter, performs structure expanding search, creates a bus, and extracts a partial bus.

Acting as a filter: For example, if “C*” is typed in the **Find Signal** field, the signal list shows all the signals that start with C. In addition, the **Find** engine is able to display their range, as well as their dimension delimiter “.”.

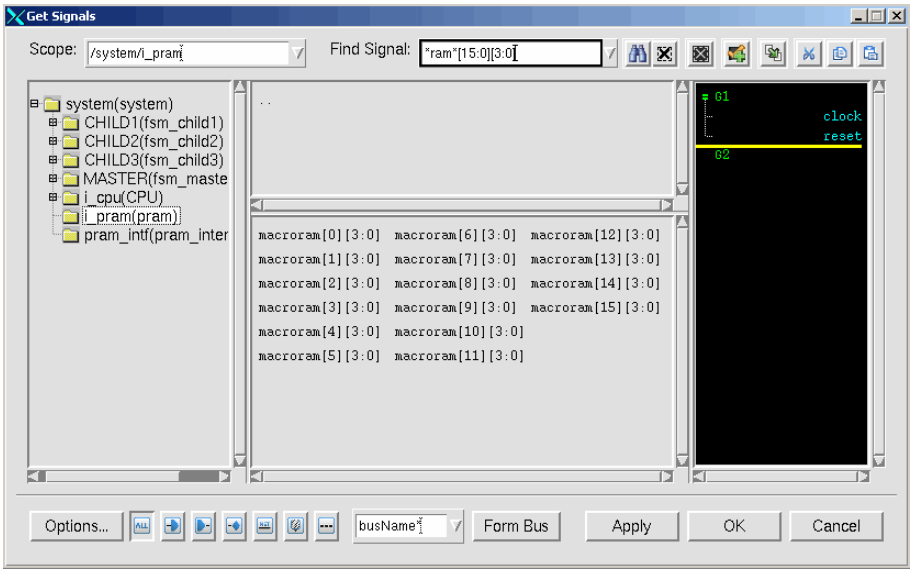


Figure: Expand and Match Fields by Specified Bit Range and Dimension Delimiter

Performing structure expanding search: Expand complex structures, array of records, or MDAs to match members using one or more asterisks and a structure delimiter in the **Find Signal** field. The following are some examples of different search strings (the search string is not case-sensitive) and their results.

a. `p*`

Matches all elements in the current level. For example:

```
Pipe1[10:0]
Pipe2[2:0][2:0]
Pinew2
...
```

b. `pi**.*`

Matches all lower members but does not include the top level member. For example:

```
Pipe1[10].pen
Pipe1[10].ops[7:0]
Pipe1[10].ops[0].src1[10:0]
...
```

c. `pipe2**.v`

Matches all elements with the lowest element “v”. For example:

```
Pipe2[2][2].ops[7].v
Pipe2[2][2].ops[7].mdst.v
```

```
...
```

d. `pipe*[1][2:1].*[6:4].v`

Matches bits in the specified range. For example:

```
Pipe2[1][2].ops[6].v
Pipe2[1][2].ops[5].v
Pipe2[1][2].ops[4].v
Pipe2[1][1].ops[6].v
Pipe2[1][1].ops[5].v
Pipe2[1][1].ops[4].v
...
```

e. `pipe2**.src?[2:0]`

Matches all the lower level's src1 and src2 fields. For example:

```
Pipe2[2][2].ops[7].src1[2:0]
Pipe2[2][2].ops[7].src2[2:0]
Pipe2[0][0].ops[7].src1[2:0]
...
```

f. `pipe2[**].ops*`

Expands MDA “pipe2” and obtains its one level members with prefix “ops”. For example:

```
Pipe2[2][2].ops[7:0]
Pipe2[2][1].ops[7:0]
Pipe2[2][0].ops[7:0]
...
```

g. `pipe2[**].ops[*]`

Expands MDA “pipe2” and expands its one level members “ops”. For example:

```
Pipe2[2][2].ops[7]
Pipe2[2][2].ops[6]
Pipe2[2][2].ops[5]
...
```

Creating a bus from the signals in the signal list: For example, if signals *C1*, *C2*, and *C3* exist in the signal list, typing *C[1:3]* in the **Find Signal** field creates a bus signal.

Extracting a partial bus for a bus signal: For example, if a bus signal *addr[7:0]* exists in the signal list, typing *addr[3:0]* in the **Find Signal** field extracts addr[3:0].

The following buttons are available on the *Get Signals* form:

- **Find Signals**  : This button locates the signals matching the pattern in the **Find Signals** text field.

- **Select/Deselect All Signals**  : This button selects or deselects all of the signals in the mirror signal pane.
- **Select/Deselect All Mirror Signals**  : This button selects or deselects all of the signals in the mirror signal pane.
- **Add Selected Signals**  : This button adds/inserts the signal selected in the signal list pane to the position of the signal cursor bar (the yellow bar) in the mirror signal pane.
- **Sync.**  : This button synchronizes the mirror signal pane with the contents of the signal list.
- **Editing: C**lick the **Cut**, **Copy**, or **Paste** buttons on the toolbar to cut, copy, or paste signals in the mirror signal pane.
- **Options:** Click the **Options** button to open the *Options* form. Refer to the *Options Form* section for details.
- **Signals Type**  : Without opening the *Options* form, these buttons can be used to choose the selected signal types displayed in the signal list of the *Get Signals* form. The available types are: **All, Input, Output, Inout, Net, Register,** and **Others** as displayed from left to right.
- **Form Bus Rule:** A temporary bus can be automatically created in the **Form Bus Rule** field by entering a user-defined bus rule. Examples are shown below.

Rules	Description
bus* buspre*buspos	'*' is integer, form bus%[msb:lsb] '*' is integer, form buspre%buspos[msb:lsb]
bus? buspre?buspos	'?' is integer, form bus%[msb:lsb] '?' is integer, form buspre%buspos[msb:lsb]

1. Assume that a design scope contains the following variables: my1bus, my2bus, my3bus, my4bus, my5bus, my6bus, my7bus, my8bus, my9bus, and my10bus. If “my*bus” or “my?bus” is typed in the **Form Bus Rule** field, the bus is created as my%bus[10:1] (MSB:my10bus, LSB:my1bus).
2. Assume that a design scope contains the following variables: mybus1, mybus2, mybus3, mybus4, mybus5, mybus7, mybus8, mybus9, and mybus10. If “mybus*” or “mybus?” is typed in the **Form Bus Rule** field,

the buses are created as mybus%[5:1] (MSB:mybus5, LSB:mybus1) and mybus%[10:7] (MSB:mybus10, LSB:mybus7).

NOTE: The form bus rule only works for single bit signals (such as If variables Mux1_sel[1:0] and Mux2_sel[1:0] exist, entering Mux?_sel or Mux*_sel do not automatically creates a new bus).

NOTE: Signal names without any numbers are not formed as a bus (such as If variables EnA, EnB, and EnC exist, entering En? or En* do not automatically creates a new bus).

The **Apply** button applies both the signals in the mirror signal pane and the selected signals in the signal list to the *nWave* window and leaves the *Get Signals* form open. The **OK** button applies both the signals in the mirror signal pane and the selected signals in the signal list to the *nWave* window and closes the *Get Signals* form.

Options Form

The *Options* form includes three tabs: **Display, Search,** and **Others**.

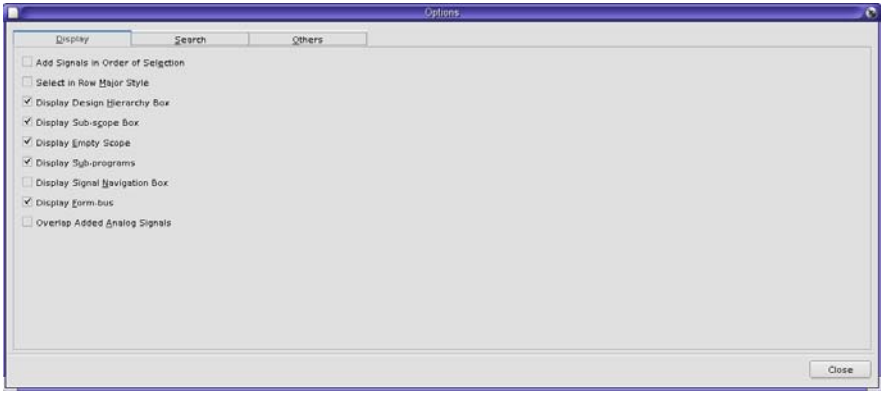


Figure: Options Form - Display Tab

The **Display** tab includes the following options:

- **Add Signals in Order of Selection:** Specify whether to add signals to the mirror signal pane in the order of their selection. When this option is turned on, the signals are displayed in the mirror window immediately when a signal name is selected; otherwise, signals are added to the mirror signal pane in alphabetical order.

- **Select in Row Major Style:** When this option is turned on, the display order of signals in the signal list pane is changed from the column major style to row major style. The default is *off*.
- **Display Design Hierarchy Box:** Specify whether to display the design hierarchy section.
- **Display Sub-scope Box:** Specify whether to display the instances under the selected instance in the subscope box.
- **Display Empty Scope**: Specify if the empty scope must be displayed.
- **Display Sub-programs**: Specify if the subprograms must be displayed.
- **Display Signal Navigation Box**: Specify if the Signal Navigation Box must be displayed.
- **Display Form-bus**: Specify if the Form Bus option on the form must be displayed.
- **Display Sub-programs:** Specify whether to display subprograms.
- **Overlap Added Analog Signals:** When this option is turned on, the signal with other sweep signals is renamed with "(sweeps)" in the tail. When this option is turned off, the signal with other sweep signals is not overlapped. When the current active file has multiple sweep signals, the option is disabled. The default is *on*.

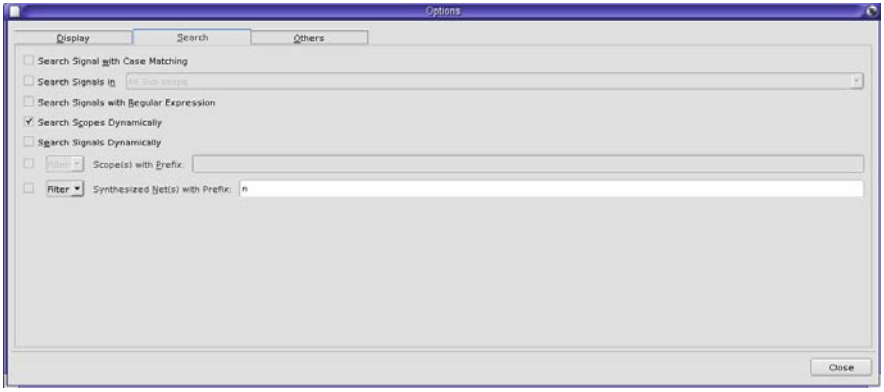


Figure: Options Form - Search Tab

The **Search** tab includes the following options:

- **Search Signals with Case Matching:** When this option is turned on, the case is matched (for example, Ab = Ab and ab = ab) during the signal list search. When this option is turned off, the case is not matched (for example, Ab=ab=AB=aB).

- **Search Signals in All Sub-scope/Interface Samenet Sub-scope:** When this option is enabled, you can select **All Sub-scope** option or **Interface Samenet Sub-scope** option from the drop-down list. By default, this option is disabled. When this option is disabled, only the current scope is searched.
  - **All Sub-scope**: When this option is selected, then the current scope and the sub-scopes are searched for the selected signal, as illustrated in the following figure:

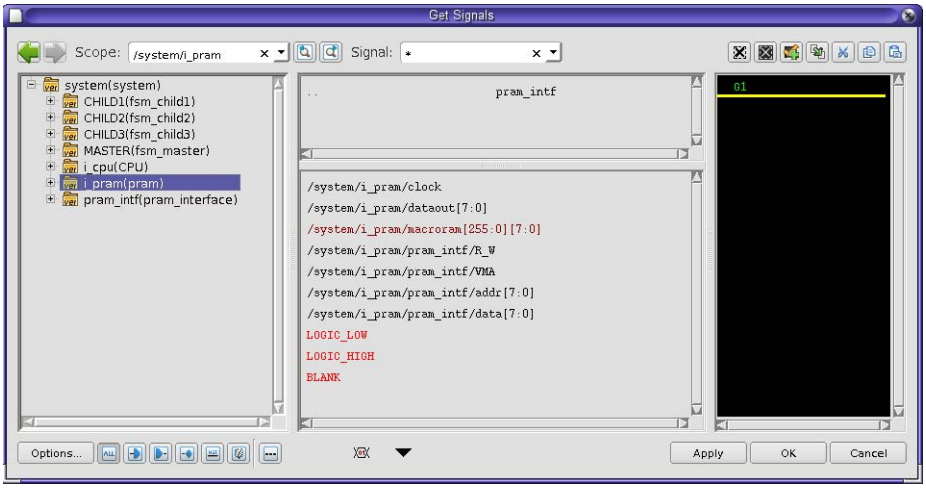


Figure: Example of All Sub-scope Option Search Results

  - **Interface Samenet Sub-scope**: When this option is selected, signals in the instantiated interface scope are searched. The search result is prefixed with the interface name, for example, `pram_intf`, as illustrated in the following figure:

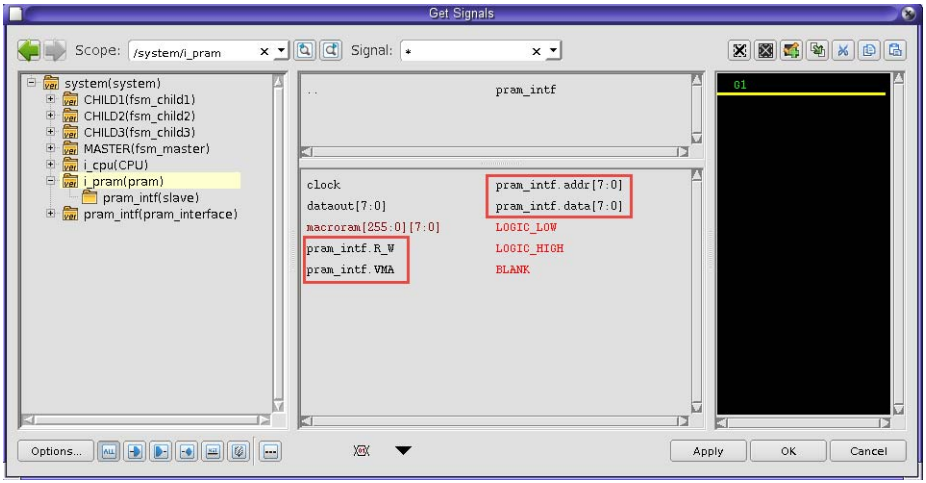


Figure: Example of Interface Samenet Sub-scope Option Search Results

- **Search Signals with Regular Expression:** When this option is turned on, regular expressions can be used when searching signals and scopes in the *Get Signals* form. This option works with the **Search Signals with Case Matching** option. For example, if `^data` is specified in the **Find Signal** text field and the **Search Signals with Case Matching** option is turned on, `/system/DATA[7:0]` cannot be found. If `^data` is specified in the **Find Signal** text field and the **Search Signals with Case Matching** option is turned off, `/system/DATA[7:0]` can be found. The **Search Signals Dynamically** option is disabled (since it is not suitable for regular expression mapping) when the **Search Signals with Regular Expression** option is turned on.
- **Search Scopes Dynamically:** Specify whether to search the matched scope name dynamically when a character is entered in the **Scope** text field. When this option is turned on, the Verdi platform attempts to match signals for each character typed which can slow down the search process. When this option is turned off, the Verdi platform waits until the Enter key is pressed to search for a match on the final entry.
- **Search Signals Dynamically:** Specify whether to search the matched signal name dynamically when a character is entered in the **Find Signal** text field. When this option is turned on, the Verdi platform attempts to match signals for each character typed which can slow down the search process. When this option is turned off, the Verdi platform waits until the Enter key is pressed to search for a match on the final entry.

- **Filter/Find Scopes with Prefix:** When this option is turned on, the scopes/instances with the prefix specified in the text field are filtered out by selecting **Filter** or shown in the *Get Signals* form by selecting **Find**. Multiple prefixes can be entered using a comma or space as the separator. When this option is turned off, all the scopes/instances are shown in the *Get Signals* form. The default is *on*. Select the **Filter** or **Find** option in the selection field:
  - **Filter Scopes with Prefix:** Filter the scopes/instances with the prefix specified in the text field.
  - **Find Scopes with Prefix:** Find the scopes/instances with the prefix specified in the text field and show these scopes/instances in the *Get Signals* form.
- **Filter/Find Synthesized Nets with Prefix:** When this option is turned on, the synthesized nets with the prefix specified in the text field are filtered out by selecting **Filter** or shown in the *Get Signals* form by selecting **Find**. Multiple prefixes can be entered using a comma or space as the separator. When this option is turned off, all the synthesized nets are shown in the *Get Signals* form. The default is *on*. Select the **Filter** or **Find** option in the selection field:
  - **Filter Synthesized Nets with Prefix:** Filter the synthesized nets with the prefix specified in the text field.
  - **Find Synthesized Nets with Prefix:** Find the synthesized nets with the prefix specified in the text field and show them in the *Get Signals* form.

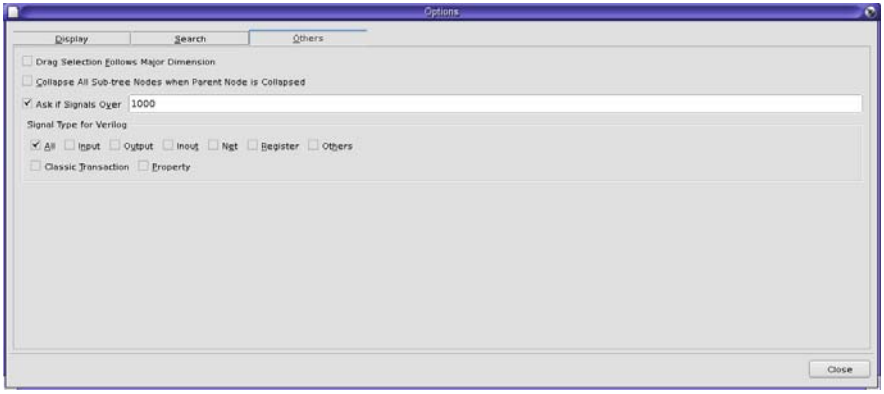


Figure: Options Form - Others Tab

The **Others** tab includes the following options:

- **Drag Selection Follows Major Dimension:** Specify whether drag selection follows the column. When this option is turned on, drag selection goes by column. When this option is turned off, drag selection goes by row.
- **Collapse All Sub-tree Nodes When Parent Node is Collapsed:** When this option is turned on, all subtree nodes are collapsed if their parent node is collapsed. When this option is turned off, the nodes are not collapsed.
- **Ask if Signals Over:** Specify whether to search over *n* matched signal names when a character is entered in the *Find Signal* text field (where *n* stands for the integer entered in the provided text field). The minimum value is 1000 and this is also the default. When the number of signals is more than the value entered, a message window opens to confirm whether or not to list the signals.

NOTE: The **Ask if Signals over** option does not work for FSDB files generated from VCD because not enough information is available in VCD files. It only works for FSDB files generated using the Novas object files.

In the **Signal Type for VHDL/Verilog** section, choose the signal types to be displayed in the signal list of the *Get Signals* form. The signal type depends on whether the imported simulation file is Verilog, VHDL, or none (such as the Analog simulation file in the demo case). The available types include:

- **All:** Displays all signal types.
- **Input:** Displays input ports.
- **Output:** Displays output ports.
- **Inout:** Displays inout ports.
- **Net:** Displays net data type signals.
- **Register:** Displays register data type signals.
- **Buffer:** Displays buffer ports in VHDL.
- **Linkage:** Displays linkage ports in VHDL.
- **Classic Transaction:** Displays transaction streams.
- **Property:** Displays assertion properties.
- **Retention:** Displays Retention signals in power designs.
- **Isolation:** Displays Isolation signals in power designs.
- **Level-shifted:** Displays Level-shifted signals in power designs.
- **Boundary Port**: Displays Boundary Port signals in power designs.
- **Others:** Displays other signal types.

Get All Signals

Menu Bar: **Signal -> Get All Signals**

This option obtains all signals in the active file and displays them. As this process can take quite some time, avoid using this option if a vast number of signals or dump data exists in the active file.

Cut

Menu Bar: **Signal -> Cut**

Toolbar Icon: 

Bind Key: Delete

This option cuts the selected signals from the signal pane into the clipboard buffer.

Copy

Menu Bar: **Signal -> Copy**

Toolbar Icon: 

Bind Key: Ctrl+P

This option copies the selected signals into the clipboard buffer.

Paste

Menu Bar: **Signal -> Paste**

Toolbar Icon: 

Bind Key: Insert

This option pastes the signals from the clipboard buffer to the mirror signal pane under the signal cursor (the yellow bar). Any previously selected signals are deselected and the newly pasted signals are selected. Before pasting signals, click the middle mouse button in the position where the pasted signals are to be located to set the signal cursor position.

Move

Menu Bar: **Signal -> Move**

Bind Key: M

This option moves the selected signals to the new destination immediately above the signal cursor (the yellow bar). By default, pressing the right mouse button in the signal pane invokes the **Signal -> Move** option. Before moving signals, click the middle mouse button in the position where the moved signals are to be located to set the signal cursor position.

Overlay Operations

The **Overlay, Highlight Overlay Signals,** and **Unhighlight** sub-commands are available in the **Overlay Operations** command.

Overlay

Menu Bar: **Signal -> Overlay Operations -> Overlay**

This option creates a new compound signal, which is overlaid by all the selected single bit digital/analog signals. Digital-digital, analog-analog, and digital-analog can be overlaid. However, if digital-analog are overlaid, the results are not normalized to the same level.

The created overlaid signal is added at the signal cursor bar position, and the waveform display is scrolled to the newly overlaid waveform.

Highlight Overlaid Signals

Menu Bar: **Signal -> Overlay Operations -> Highlight Overlaid Signals**

This option highlights the signals of a overlapped panel.

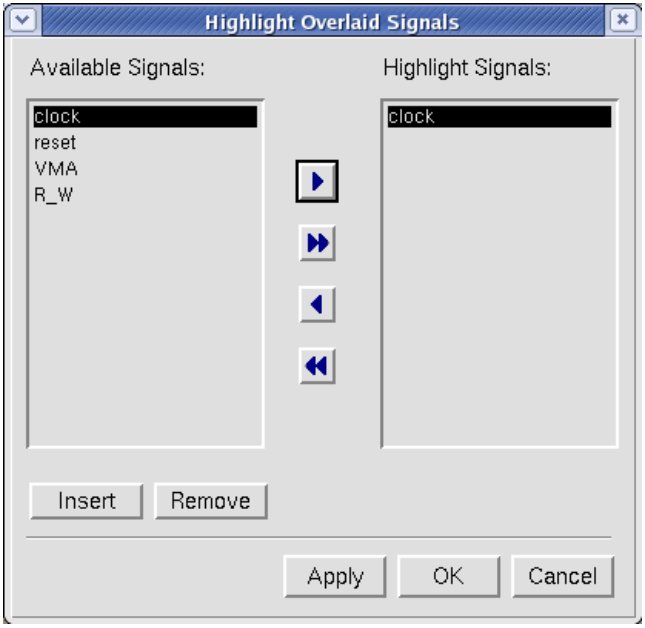


Figure: Highlight Overlaid Signals Form

The *Highlight Overlaid Signals* form lists all the signals under the current panel. After selecting the signals of interest, use the **Add Selected Signal to Highlight List** or **Add All Signals to Highlight List** icons from the **Available Signals** section to add the signals to the **Highlight Signals** section. Alternatively, remove unwanted signals using either the **Remove Selected Signal from Highlight List** or **Remove All Signals from Highlight List** icons.

Click the **Insert** button to add the selected signals in the *nWave* window to the overlapped panel. Click the **Remove** button to take off the selected signals in the **Available Signals** section from the overlapped panel.

Overlaid Signal Properties

Menu Bar:	**Signal -> Overlay Operations -> Overlaid Signal Properties**

This option provided the overlaid signal properties.

NOTE: Use the `setenv NOVAS_ENABLE_OVERLAP_ENHANCE 1` environment variable to turn on this option.

Unhighlight

Menu Bar: **Signal -> Overlay Operations -> Unhighlight**

This option disables the **Highlight Signals** option. All previously highlighted signals ate unhighlighted.

Auto Fit

Menu Bar: **Signal -> Auto Fit**

This toggle option is for the **Overlay Operations** option. When this toggle option is turned on, the newly overlaid signals automatically fit the whole window vertically.

Auto Color/Pattern

Menu Bar: **Signal -> Auto Color/Pattern**

This toggle option is for the **Overlay Operations** option. When this toggle option is turned on, the newly overlaid signals automatically have different colors and line patterns.

Find Signal

Menu Bar: **Signal -> Find Signal**

This option opens the *Find Signal* form where a signal in the signal pane can be searched using one of the following methods:

- Signal name only
- Signal name with hierarchical name

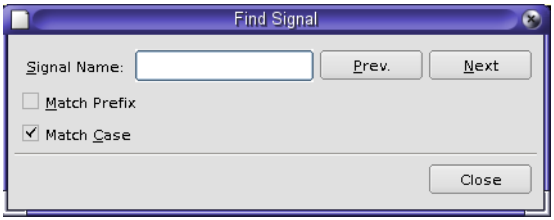


Figure: Find Signal Form

Signal Name Only
To use this method, an exact name for the signal must be specified. If the selection is a bus signal, the bit range does not need to be specified. For example, type the word *data* to search for the signal *data [7:0]*. This searching method looks for all signals named *data [7:0]* in the signal pane regardless of their locations.

Signal Name with Hierarchical Name
The full path of the signal can also be specified. For instance, type */system/reset* in the **Signal Name** text field. This search action finds only the specified location for that signal. To see the hierarchical name of each signal in the signal pane, enable the **View -> Hierarchical Name** option or press the bind key **H**.

The search scope of the signal can be specified. Turn the **Match Prefix** option on to find all signals with the specified string as the prefix. Turn the **Match Prefix** option off to find only signals that match the exact specified string. The default is *on*.

To find a signal, perform the following steps:

1. **Signal Name**: Enter the desired signal name (such as, `MyAbP`, and `My*`) in the text field.
2. **Prev**: Click **Prev** to search for the previous signal from the current signal selected in the signal pane.

   or

   Next: Click **Next** to search for the next signal in the signal pane. When the search reaches the end of the signal pane, it starts searching from the beginning.
3. **Match Prefix**: The search scope of the signal can be specified. Select the **Match Prefix** option to find all signals with the specified string as the prefix. Uncheck the **Match Prefix** option to find only signals that match the exact specified string.

4. **Match Case**: Select the **Match Case** option to perform a case-sensitive search. By default, this option is enabled.

NOTE: A signal can also be searched in a collapsed group and the collapsed group is expanded automatically if any specified signal is found.

A warning message appears if a signal is not found with the specified name.

Select

Select the signal type to highlight in the *nWave* signal pane.

Input Port

Menu Bar: **Signal -> Select -> Input Port**

This option selects and highlights input port type signals in the signal pane.

Output Port

Menu Bar: **Signal -> Select -> Output Port**

This option selects and highlights output port type signals in the signal pane.

Inout Port

Menu Bar: **Signal -> Select -> Inout Port**

This option selects and highlights inout port type signals in the signal pane.

Net

Menu Bar: **Signal -> Select -> Net**

This option selects and highlights net data type signals in the signal pane.

Register

Menu Bar: **Signal -> Select -> Register**

This option selects and highlights register data type signals in the signal pane.

Error Signals

Menu Bar: **Signal -> Select -> Error Signals**

This option highlights signals that have errors. A signal comparison must be performed before invoking the **Error Signals** option.

Stuck Signals with Cursor/Marker

Menu Bar: **Signal -> Select -> Stuck Signals with Cursor/Marker**

Bind Key: Ctrl+S

This option selects and highlights stuck signals (analog or digital signals without value changes from the cursor time to the specified marker time) in the signal pane. If the marker time is not specified, the end checkpoint is the end of the "available time range" (full time range when a file is opened directly or view time range when a file is opened with the **Open File by Time Range** option turned on). If the cursor time is the same as the marker time, no signal is selected as a stuck signal. To remove the highlighted stuck signals, press the **Delete** key or invoke the **Signal -> Cut** option in *nWave*.

NOTE: To avoid `x -> ?` glitches at time 0 which may occur with some simulators, place the cursor after the longest unknown value before invoking the **Stuck Signals with Cursor/Marker** option.

Others

Menu Bar: **Signal -> Select -> Others**

This option selects and highlights other types of signals in the signal pane.

All

Menu Bar: **Signal -> Select -> All**

Bind Key: Ctrl+A

This option selects and highlights all signals in the *nWave* window.

Deselect All

Menu Bar: **Signal -> Deselect All**

Bind Key: D

This option deselects all of the signals in the signal pane.

Clear All

Menu Bar: **Signal -> Clear All**

Bind Key: Ctrl+D

This option removes all signals and groups from *nWave* and adds group name *G1* automatically in the signal pane after invoking this option.

Sort Signals

This option sorts the signals globally in the *nWave Signal* pane either in ascending or descending order.

Ascending

Menu Bar: **Signal -> Sort Signals -> Ascending**

This option sorts all the signals and the sub-groups in the *nWave* window in the ascending order of the signal name in the *Signal* pane, as illustrated in the following figure:

Figure: Signals Sorted in Ascending Order

Descending

Menu Bar: **Signal -> Sort Signals -> Descending**

This option sorts all the signals and the sub-groups in the *nWave* window in the descending order of the signal name in the *Signal* pane, as illustrated in the following figure:

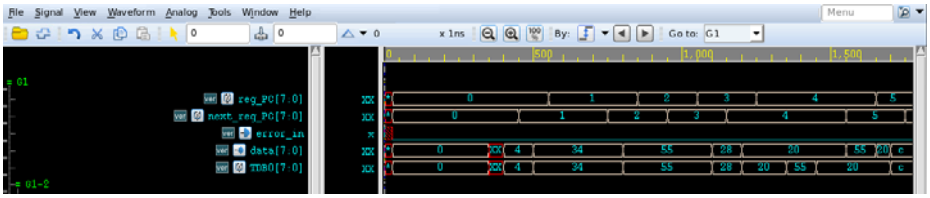


Figure: Signals Sorted in Descending Order

NOTE: Signals are sorted by the complete hierarchical name if the **View -> Hierarchical Name** option is enabled. If the option is disabled, signals are sorted by the signal name.

Sorting is also case-sensitive. That is, the signals starting with the upper-case letters are sorted first and then the lower-case letters.

Select Group Mode (Group/Signal)

Menu Bar: **Signal -> Select Group Mode (Group/Signal)**

This toggle option is used to control the selectability of a group or the signals in a group. When this option is turned on, the group is selected when the group name is selected. When this option is turned off, the signals in the group are selected when the group name is selected.

Comment

This option annotates additional feedback in the field about signals above or below them. A comment row can be inserted in the signal pane.

The height and color of the comment can be adjusted in the same way as for any other signal by selecting the signal (comment row in this case) and invoking the **Waveform -> Height** and **Waveform -> Color/Pattern** options respectively.

To delete a comment row, select it and invoke the **Cut** option (click the **Cut** button on the toolbar or invoke the **Signal -> Cut** option or press the bind key **T**).

Text Boxes

- Within a comment row, any number of the provided text boxes (that is, Square Box  , Attached Square Box  , Period Box  , and Arrow Period Box ) can be added.
- Each text box is hooked to a time point on the left and a time point on the right, therefore when users **Zoom In** or **Zoom Out** the waveform, a text box may appear squeezed or expanded.
- Any text within a text box is wrapped to its boundaries.
- Double-click the text box to edit the text inside.
- These text boxes are movable and resizable. To move a text box, drag it to a new location. To resize a text box, first select it (sizing handles appear at the corners and along the edges of the text box) then resize it by dragging its sizing handles.
- To delete a text box, select the text box, right-click to show the right-click option menu, and click **Delete**.

To save comment rows along with the signals:

Invoke the **File -> Save Signal** option to save the signals, including the comment rows, in the waveform. Other users can restore the signal file using the **File -> Restore Signal** option to see the comment text.

Insert

Menu Bar:	**Signal -> Comment -> Insert**
Right-click Menu:	Right-click the signal pane and invoke **Add Comment** on the right-click menu.

This option inserts a comment row below the signal cursor bar in the signal pane. Different text boxes (square box, attached square box, period box, and arrow period box) can be added in the comment row. Right-click the comment row in the signal pane, and then select the **Rename** option on the right-click menu to rename. Type the new name, and then click **Enter**.

Add Square Box

Menu Bar:	**Signal -> Comment -> Add Square Box**

Right-click Menu:	Right-click a comment row in the waveform window and click the **Square Box** icon  on the right-click menu.

This option adds a square box to the selected comment row. Invoke the **Signal -> Comment -> Add Square Box** option and click the position in a comment row in the waveform window where the square box is to be located.

Add Attached Square Box

Menu Bar:	**Signal -> Comment -> Add Attached Square Box**
Right-click Menu:	Right-click a comment row in the waveform window and click the **Attached Square Box** icon  on the right-click menu.

This option adds an attached square box to the selected comment row. Invoke the **Signal -> Comment -> Add Attached Square Box** option and click the position in a comment row in the waveform window where the attached square box is to be located.

Add Period Box

Menu Bar:	**Signal -> Comment -> Add Period Box**
Right-click Menu:	Right-click a comment row in the waveform window and click the **Period Box** icon  on the right-click menu.

This option adds a period box to the selected comment row. Invoke the **Signal -> Comment -> Add Period Box** option and click the position in a comment row in the waveform window where the period box is to be located.

Add Arrow Period Box

Menu Bar:	**Signal -> Comment -> Add Arrow Period Box**
Right-click Menu:	Right-click a comment row in the waveform window and click the **Arrow Period Box** icon  on the right-click menu.

This option adds an arrow period box to the selected comment row. Invoke the **Signal -> Comment -> Add Arrow Period Box** option and click the position in a comment row in the waveform window where the arrow period box is to be located.

Assertion Comments

In addition to the comment options described previously, special comments related to assertions can be inserted in the *nWave* window.

After invoking the **Assertion Analyzer** option at the success/failure point of an assertion, the *Analyzer* pane and the *nWave* window are displayed. In the *nWave* window, the signals related to the assertion are displayed. The observation points of the assertion and failure statements are also indicated as shown below.

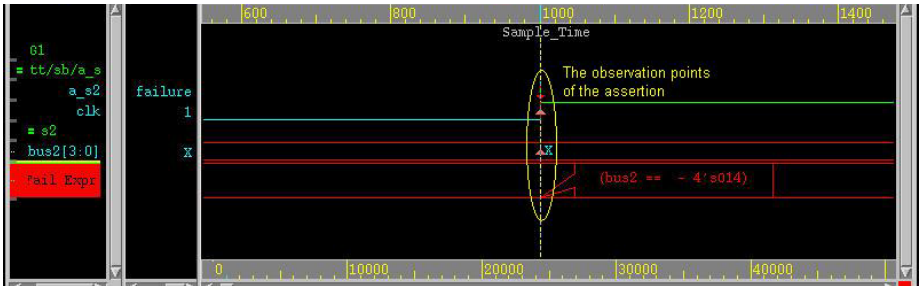


Figure: Assertion Comments in the nWave Window with Sample Time

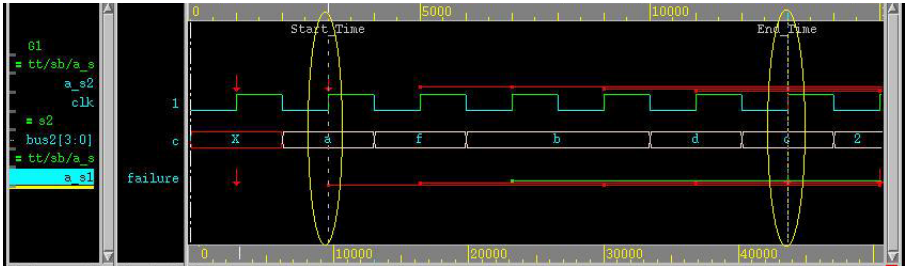


Figure: Assertion Comments in the nWave Window with Start and End Time

Refer to the *Analyzer Pane* section of the *Assertion Debug* chapter and the **Assertion Analyzer** option description in the *nTrace* chapter for details.

Split Inout

NOTE: This option is available when an EVCD FSDB file is loaded.

This option splits one or more EVCD signals that are selected in the signal pane into the input signals and output signals. The split signals are listed under the original signal.

Bus Operations

Six subcommands are available in the **Bus Operations** menu: **Expand as Sub-bus, Edit Bus, Create Bus, Adjust Bus, Reverse Bus,** and **Add Full Bus**.

Expand as Sub-bus

Menu Bar: **Signal -> Bus Operations -> Expand as Sub-bus**

This option opens the *Expand as Sub-bus* form where bus signals can be expanded in a specified range. To expand all of the bits in the bus, double-click the bus in the signal pane.

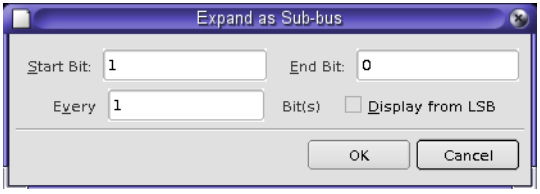


Figure: Expand as Sub-bus Form

The *Expand as Sub-bus* form includes the following option and fields:

- **Start Bit:** Specify the start bit of the selected bus.
- **End Bit:** Specify the end bit of the selected bus.
- **Every Bits:** Specify the bit size to form sub-buses. The default is *1*.
- **Display from LSB:** When this option is turned on, sub-buses are displayed from the LSB (Least Significant Bit).

Edit Bus

Menu Bar: **Signal -> Bus Operations -> Edit Bus**

This option edits natural buses and created buses. If a natural bus is selected in the *nWave* signal pane, invoking this option opens the *Edit Bus* form where bus bits can be modified or reversed. The created partial bus signals replace the original bus signals in the signal pane. If a created bus is selected in the *nWave* signal pane, invoking this option opens the *Create Bus* form (see the **Create Bus** option). Multiple natural buses can be edited at the same time, if they have the same bus range (same start and end bit).

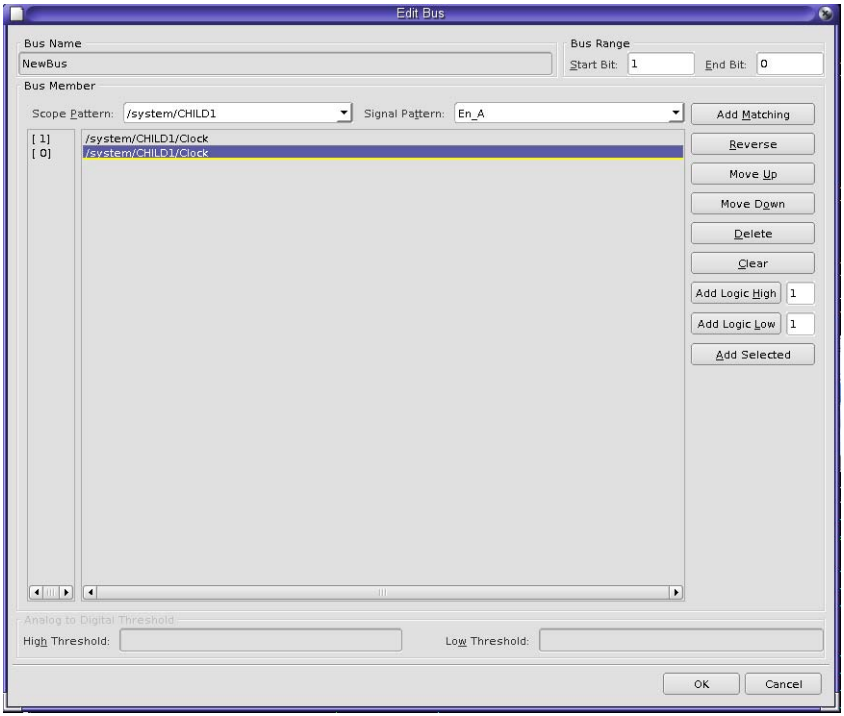


Figure: Edit Bus Form

The *Edit Bus* form includes the following fields and button:

- **Bus Name:** Displays the name of the selected bus.
- The **Bus Range** section creates partial bus signals after the start bit and end bit are specified.
  - **Start Bit:** Specify the start bit of the selected bus.
  - **End Bit:** Specify the end bit of the selected bus.
  - **Reverse:** Click this button to reverse the bit order of the selected bus.
- **OK:** Click this button to change the start bit and end bit of the selected bus.

Create Bus

Menu Bar:	**Signal -> Bus Operations -> Create Bus**
Bind Key:	K

After the signals are selected in the *nWave* signal pane, invoke this option to open the *Create Bus* form where a new bus signal can be created from existing signals,

MDA signals, buses, and VHDL integers. If the selected signal is a bus (multiple bus selection is supported), the buses are flattened to individual bus bits and these individual bits are used to form the new bus.

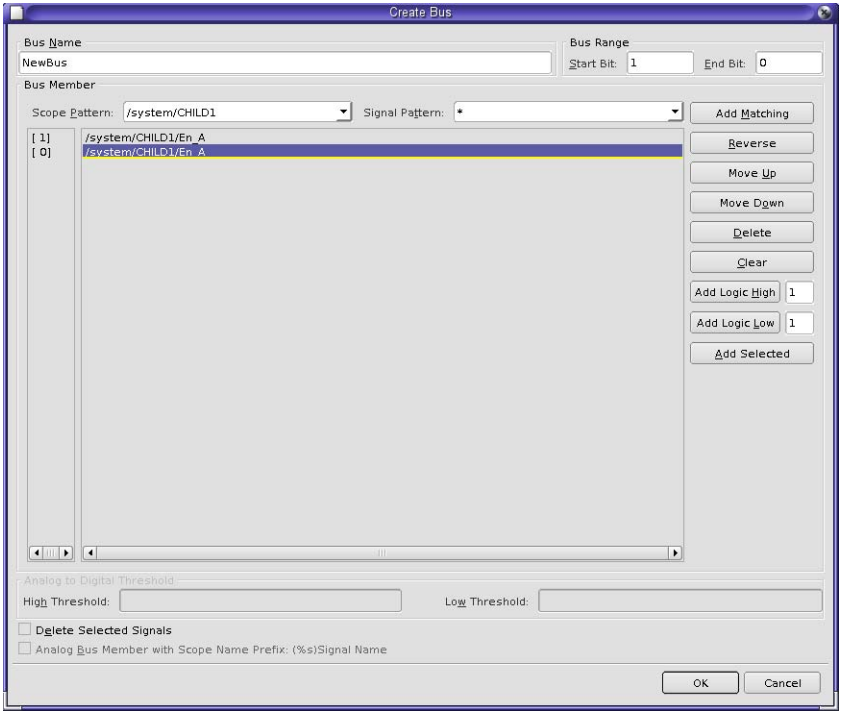


Figure: Create Bus Form

The *Create Bus* form includes the following options, fields, and buttons:

- **Bus Name:** Enter a name in this text field for the selected bus.
- **Bus Range:** This section creates partial bus signals after the start bit and end bit are specified.
  - **Start Bit:** Specify the start bit of the created bus.
  - **End Bit:** Specify the end bit of the created bus.

The **Bus Member** list shows the bus members of the selected signals which are within the specified range on the right side and the sequence on the left side. The numbers within the brackets indicate the individual bits of the bus.

- **Scope Pattern:** Specify the scope pattern to search for. Enter the string manually or drop a scope in this text field by dragging from *nTrace* or *nWave*.

- **Signal Pattern:** Specify the signal name pattern to search for. Enter the string manually or drop a signal in this text field by dragging from *nTrace* or *nWave*.
- **Add Matching:** Click this button and signals matching the specified patterns are shown in the **Bus Member** list.
- **Reverse:** Click this button to reverse the sequence of signals listed in the **Bus Member** list. The **Reverse** button only reverses selected member signals. If no member signals are selected, the **Reverse** button reverses all signals in the **Bus Member** list.
- **Move Up:** This button moves the selected bus member in the list one row up each time the button is clicked.
- **Move Down:** This button moves the selected bus member in the list one row down each time the button is clicked.
- **Delete:** This button deletes the selected signal in the **Bus Member** list.
- **Clear:** This button clears all of the signals in the **Bus Member** list.
- **Add Logic High**: This button adds the logic high signal (1) at the signal cursor bar (the yellow line) of the **Bus Member** list. The text field can also be used to specify the number of logic low signals to be added in the **Bus Member** list.
- **Add Logic Low:** This button adds the logic low signal (0) at the signal cursor (the yellow line) of the **Bus Member** list. The text field can also be used to specify the number of logic low signals to be added to the **Bus Member** list.
- **Add Selected:** Click this button to add the selected signals from the *nWave* signal pane to the **Bus Member** list.
- The **Analog to Digital Threshold** section specifies the high threshold and low threshold of an analog signal value.
  - **High Threshold:** Specify the high threshold of an analog signal value.
  - **Low Threshold:** Specify the low threshold of an analog signal value.
- **Delete Selected Signals:** When this option is turned on, the signals selected in the signal pane are deleted after the new bus signal is created. When this option is turned off, the selected signals are left in the signal pane. The default is *off*.
- **Analog Bus Member with Scope Name Prefix: (%s) Signal Name:** When this option is turned on, the full hierarchy scope name is added as a prefix to each analog signal to show the correct value of the newly created bus. This option is turned on if analog signals exist in the bus member list and at least two signal names in the bus member list are the same. This

option is turned off if no signals are in the bus member list, if none of the signals are analog signals, or if the name of each analog signal is different.

- **OK:** Click this button to add the new bus signal to the *nWave* signal pane.

Adjust Bus

Menu Bar: **Signal -> Bus Operations -> Adjust Bus**

This option adjusts multiple signals, especially bus signals.

NOTE: This option only applies to one-dimensional arrays or single-bit signals.

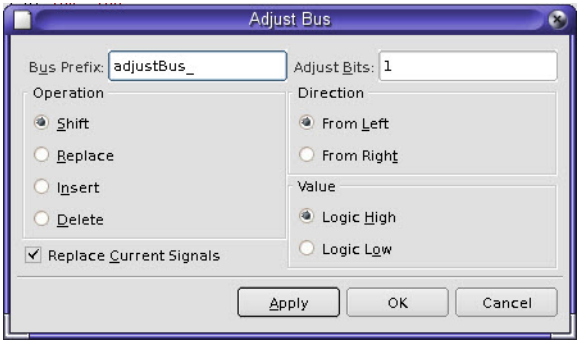


Figure: Adjust Bus Form

The *Adjust Bus* form includes the following options and fields:

- **Bus Prefix:** Enter the prefix for the signals of interest.
- **Adjust Bits:** Specify the total bits to adjust for the selected signals.

The **Operation** section changes the data shown in *nWave* for a particular bus by using one of the following options:

- **Shift**: Shift n bits from the left or right of the bus.
- **Replace**: Replace n bits from the left or right of the bus.
- **Insert**: Add n bits to the left or right of the bus.
- **Delete:** Delete n bits from the left or right of the bus.

The **Direction** section specifies the moving direction for the selected bus signals.

- **From Left**: Select this button to move the selected bus signals from the left.
- **From Right:** Select this button to move the selected bus signals from the right.

The **Value** section specifies the value of the new bits.

- **Logic High:** Specify the value of the new bits as logic high.
- **Logic Low:** Specify the value of the new bits as logic low.
- **Replace Current Signals:** When this option is turned on, it replaces the selected signals from the signal pane with the new bus signals.
- **Apply to Selected:** This button saves the settings and applies them to the selected signals.

  The following is an example of applying the adjust bus settings to a bus signal (data[7:0]) in *nWave*. The operations are based on the adjust bus settings below:

  Bus Prefix: adjustBus_
  Adjust Bits: 2
  Operation: Shift
  Direction: From Left
  Value: Logic High
  Replace Current Signals: Off

  In this example, the value of the original bus signal “data[7:0]” (0000_0000) is shifted 2 bits from the left with 2 new left-most bit = 1, and the new value of adjustBus_data[7:0] is 1100_0000, as shown in the following figure.

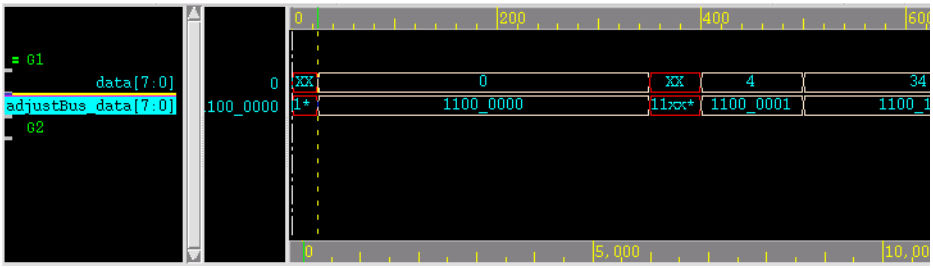


Figure: Apply the Adjust Bus Settings in nWave

Reverse Bus

Menu Bar: **Signal -> Bus Operations -> Reverse Bus**

This option can only be applied to bus signals. This option reverses the sequence of signals and the associated bits are annotated in the opposite direction. For example, after invoking the **Reverse Bus** option on the bus signal data[7:0], data[7:0] is displayed as data[0:7] in the signal pane.

Add Full Bus

Menu Bar: **Signal -> Bus Operations -> Add Full Bus**

This option is disabled if the selected signal is not a bus member or partial bus. This option adds and displays the entire bus for a bus member or a partial bus.

Expand/Collapse

Menu Bar: **Signal -> Expand/Collapse**

When a bus is selected, the **Expand** option expands the selected bus to display the bus members. When an assertion/property is selected, the **Expand** option expands the selected assertion/property to display its properties/signals. When a power domain signal is selected, the **Expand** option expands the three member signals to display value changes for power state, power nominal for a CPF file (power nominal is power alias when a UPF file is loaded), and power voltage.

The **Collapse** option folds the expanded bus members, assertion signals, or member signals.

NOTE: When the bus member signal is modified, for example the signal is renamed or cut and then the signal is collapsed, a message appears mentioning that the signals that are expanded are modified and are not the original signals when you expand the signals again.

Collapse to Parent

Menu Bar: **Signal -> Collapse to Parent**

When a bus is selected, the **Collapse to Parent** option collapses the expanded bus members, Multi-Dimensional Arrays (nMDA), transactions, and composite signals.

When this option is invoked for a bus member, it collapses the bus members and other sub-members associated with it.

If the signal selected is not the member of the bus selected or the signal cannot find its parent, then a warning message appears mentioning that the parent of the signal cannot be found.

This option is disabled when you select more than two bus members, group signals, and new compound signals (Overlay) signals.

Logical Operation

Menu Bar: **Signal -> Logical Operation**

This option opens the *Logical Operation* form where new signals can be created from a logical operation performed on existing signals.

Figure: Logical Operation Form

The *Logical Operation* form includes the following options, buttons, and fields:

- **Name:** This text field lists all existing expression names (and ranges). The expression names can be specified or modified manually. The default name is “logical_expression_<index>”.

NOTE: All bits of the bus must be specified in the **Name** text field if a bus is desired.

- **Expression:** Specify an expression in the **Expression** text field. The following methods are available to specify the expression:
  - Select signals from the signal pane and invoke the **Logical Operation** option. The expression of the selected signals is displayed in the **Expression** text field. The expression can be edited by adding operators

through the operator buttons in the *Logical Operation* form. During the first use of this field, the default operator is & (AND).

- Type the signal's full path and the operator in the **Expression** text field. Use the **Ctrl+A** key to select all text in the **Expression** text field.

- Invoke the **Logical Operation** option without selecting a signal from the signal pane and then either:
  - Click the operator buttons to drag signals from the signal pane to the **Expression** text field and add operators, or
  - Click the operator buttons to select signals from the signal pane and add operators. Then, click the **Add Signals** button. Click the **Clear** button to clear the **Expression** text field.
- Select **with Value** check box to display the value (with the radix prefix) on the **Expression** field at the current cursor time when signals are added, as illustrated in the following figure:

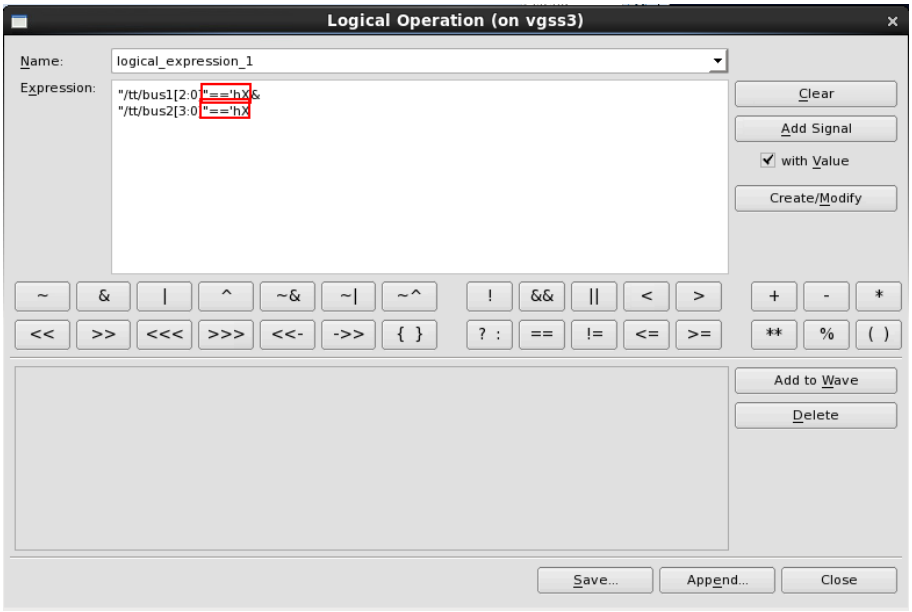


Figure: Logical Operation With Value

NOTE: If an alias has been applied to the specified signal, the alias string can be added after the signal name and operator (such as /`system/addr[7:0] + ONE`). If the alias string contains a space, a pair of double quotes must be added to the expression (such as `signal == Zero Value`).

If an ASCII value has been added after the specified signal name and operator, the ASCII value must begin with ‘a and must enclose the value with a pair of double quotes (such as `/system/i_cpu/check_alias[255:0] ==’a"jkl`).

The signal notation set by the **Waveform -> Signal Value Notation** option is considered during the expression evaluation.

Each signal added in the **Expression** field appears on separate lines.

- **Operator Buttons:** Three types of operators are available: Bit-wise operators, Logical operators, and Arithmetical operators. *nWave* provides the same operators used in Verilog/VHDL:

Bit-wise Operators		**Logical Operators**		**Arithmetical Operators**	
~	bit-wise negation	!	logical negation	+	addition
&	bit-wise and, reduction and	&&	logical and	-	subtraction
|	bit-wise or, reduction or	||	logical or	*	multiplier
^	bit-wise xor, reduction xor	<	less	**	power
~&	reduction nand	>	greater	%	modulus
~|	reduction nor	?:	conditional	( )	parentheses
~^	bit-wise xnor, reduction xnor	==	logical equality		
<<	logical left shift	!=	logical inequality		
>>	logical right shift	<=	less or equal		
<<<	arithmetic left shift	>=	greater or equal		
>>>	arithmetic right shift				
->>	signal right shift				
<<-	signal left shift				
{}	concate				

- **Expression List:** The list displays each expression after the information has been entered in the **Name** and **Expression** text fields and the **Create/Modify** button has been clicked.
- **Create/Modify:** Click this button to create or modify an expression. To modify an existing expression, double-click a selected expression in the **Expression** list, change the expression in the **Expression** text field, and then click the **Create/Modify** button. If the name is not changed (already

exists), the modified expression is updated in the **Expression** list. If the name is changed, a new expression is created.

- **Add to Wave:** Click this button to add the selected expression with the specified name to the signal pane in *nWave*.
- **Delete:** Click this button to delete a selected expression from the **Expression** list.
- **Save:** Click this button to open the *Save Logical Expression* form where the created/modified expression can be saved to a file (`*.exp`).
- **Append:** Click this button to open the *Append Logical Expression* form where the directory structure can be viewed and a previously saved logical expression can be loaded.

Event

Menu Bar: **Signal -> Event**

This option opens the *Event* form which includes the **Event** tab and the **Complex Event** tab. The following fields exist in both tabs.

- **Search by Event List:** The selected event is used by the **Search by Event** icon 

on the *nWave* toolbar.
- **Save:** This button saves the events and complex events defined in the **Event** window into an event resource file (saving the file with an `*.rc` extension is recommended).
- **Restore:** This button restores an event resource file (`*.rc`) to the *Event* form.

Event Tab

After an event has been created, it can be selected and searched in *nWave* just like a regular signal. Use the **Event** tab to define an event and then use the **Search by Event** icon from the *nWave* toolbar. The **Search By** option finds the time/position when an event occurred. The normal process to create an event is as follows:

1. Select one or more signals in the signal pane and place the cursor in the waveform pane on the values to use for the event.
2. In the *Event* form, click the **Insert** button to add the selected signals to an event in the form.
3. Select an event and click the **Edit** button to enter the edit mode. Then change the operator in the *Edit Event* form.

4. In the **Expression** text field of the *Edit Event* form, click the **Operators** list to enter operators and drag from other windows to enter signals.
5. Click **OK** to return to non-edit mode and perform expression syntax checking.

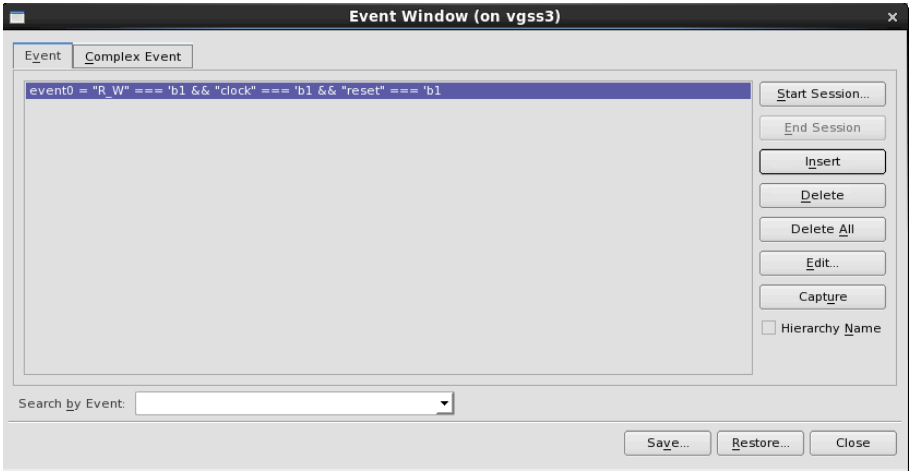


Figure: Event Window - Event Tab

The **Event** tab includes the following buttons and option:

- **Start Session:** Click this button to control the data capture of a simulator or emulator, such as to control which cycles become part of the FSDB for size or to facilitate the viewing of dependent events that span a large amount of simulation time.

  After you click the **Start Session button** and give the **Complex Event Name**, a message prompt appears, requesting further action. For example:

  ```
  Please select signals and double click specific transitions to
  create events, then click "End Session" button to create a
  complex event (a combination of created events).
  ```

- **End Session:** Click this button for the data capture of a simulator or emulator. Refer to the **Start Session** button described above for details.
- **Insert:** After selecting the signals in the signal pane of the *nWave* window, click the **Insert** button to add an event to the **Event** tab. The event's expression is automatically composed according to the selected signals. By default, a Case Equality (===) subexpression with the signal name on the left-hand side and the value on the right-hand side is automatically composed for each signal. The right-hand side value in a subexpression is

extracted from the cursor value in the waveform. Finally, the logical AND (&&) operator is inserted between subexpressions.

- **Delete:** Click this button to delete the event highlighted in the **Event** list.
- **Delete All:** Click this button to delete all of the events in the **Event** list.
- **Edit:** Click the **Edit** button to display the *Edit Event* form. Refer to the *Edit Event Form* for details.
- **Capture:** After selecting an event in the *Event* form, click the **Capture** button and the selected event is added to the signal pane of *nWave* at the current cursor position. The event's trigger time positions are displayed in the waveform pane.
- **Hierarchy Name:** When this option is turned on, each signal name appears with its full hierarchical path.

Edit Event Form

The name and expression of the event can be modified in the *Edit Event* form. Click the **Add Selected Signals** button to add a signal selected in the signal pane of *nWave*. By default, the logical AND (&&) operator is inserted between subexpressions.

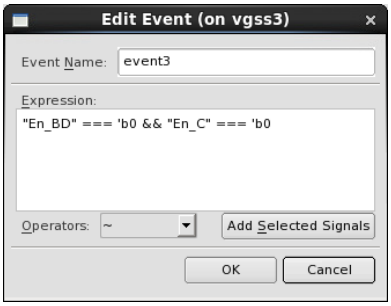


Figure: Edit Event Form

The *Edit Event* form includes the following fields:

- **Event Name:** Specify the event name. The event name must be unique.
- **Expression:** To obtain an event expression, select one or more signals in the signal pane and click **Insert** in the *Event* form. By default, a Case Equality (===) subexpression with the signal name on the left-hand side and the value on the right-hand side is automatically composed for each signal. The right-hand side value in a subexpression is extracted from the cursor value

in the waveform. Finally, the logical AND (&&) operator is inserted between subexpressions.

NOTE: Signal names in events should be surrounded by double quotes.

- **Operators:** *nWave* provides the following operators:

~	bit-wise negation	===	case-equality
!	logical negation	!==	case-inequality
+	addition	&	bit-wise and
-	subtraction	^	bit-wise xor
<<	left shift	~^	bit-wise xnor
>>	right shift	|	bit-wise or
<	less	&&	logical and
>	greater	||	logical or
<=	less or equal	or	event or
>=	greater or equal	posedge	positive edge
==	logical equality	negedge	negative edge
!=	logical inequality	anychange	edge

Complex Event Tab

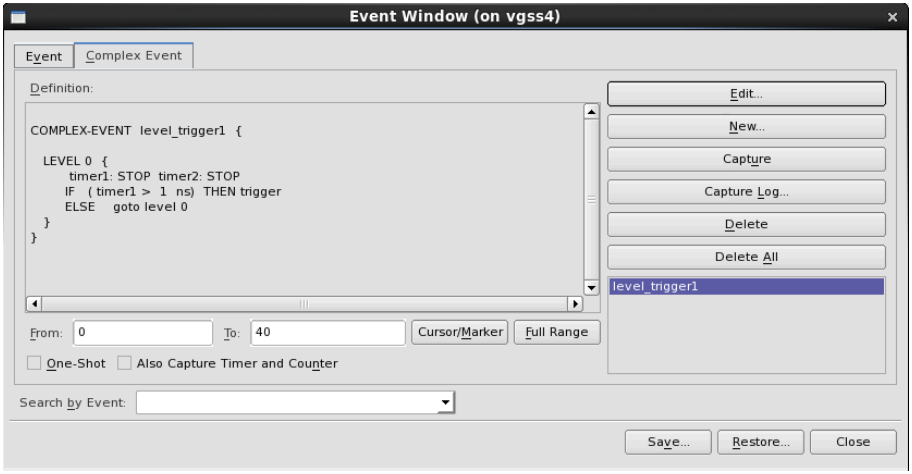


Figure: Event Window - Complex Event Tab

The **Complex Event** tab includes the following options and fields:

- **Edit:** Click this button to open the *Edit Complex Event* form where complex events can be edited. Refer to the *Edit Complex Event Form* form for details.
- **New:** Click this button to open the New Complex Event form where complex events can be created. Refer to the Edit Complex Event Form form for details.
- **Capture:** Select the complex event in the *Event* form and click the **Capture** button. The selected complex event is added to the signal pane of *nWave* at the current cursor position. The event's trigger times are displayed in the waveform pane.
- **Capture Log:** Select a complex event and click this button to view the detailed log in the *Complex Event Capture Log* form. In this form, the trace time (with the cursor time) can be changed by using the arrow buttons to check the complex event captured log at the trace time. Use the stack-like log to debug the selected complex event.

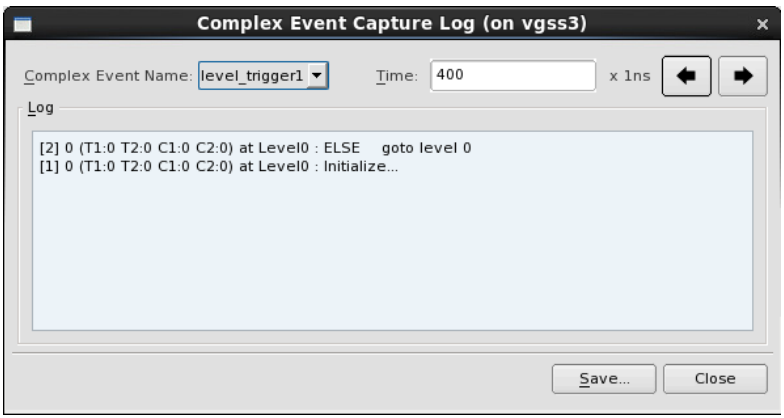


Figure: Complex Event Capture Log Form

- **Delete:** Click this button to delete the complex event highlighted in the **Complex Event** list.
- **Delete All:** Click this button to delete all complex events in the **Complex Event** list.
- **Complex Event List:** List all defined **Complex Events**. The definition of the selected complex event is shown in the definition table.
- **From:** Specify the start time for the event search.
- **To:** Specify the end time for the event search.
- **Cursor/Marker:** Click this button to synchronize the time range with the *nWave* cursor/marker times.

- **Full Range:** Click this button to set the range to the entire simulation time.
- **One-shot:** Determine the event search within the specified time range: stop at one-match (option is turned on) or continue repeatedly (option is turned off).
- **Also Capture Timer and Counter:** When this option is turned on and the **Capture** button is clicked, available timers and counters signals are also captured on the waveform window when capturing a complex event. Change the cursor time to check the current value of timers and counters.

Edit Complex Event Form

The *Edit Complex Event* form supports two timers and two counters for composing the event sequence. Multiple sequence levels are supported.

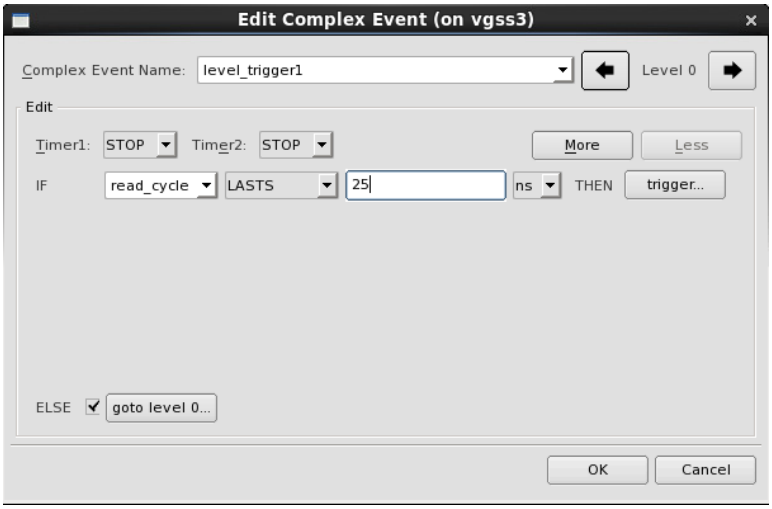


Figure: Edit Complex Event Form

The normal process to edit a complex event is as follows:

1. Enter the complex event name in the **Complex Event Name** text field.
2. Specify the timer state (**Timer1** and **Timer2**) for the level.
3. Edit the **If** branch *condition* portion of the complex event. First, select a term from a list comprising of timers, counters, or defined events. After a timer or counter term is selected, the selection field next to the term automatically provides available comparison operators (==, !=, >, >=, <, <=). Choose a comparison operator and edit the number field next to it. If a defined event term is selected, the selection field next to the term provides

OCCURS or **LASTS** for choice. Choose one option and edit the number field next to it.

4. Edit the **If** branch *action* portion of the complex event by selecting an action from a list that contains trigger, terminate, branch to, or other level actions and counter operations. To open the *Action* form, click the button with the default action of **Trigger**.

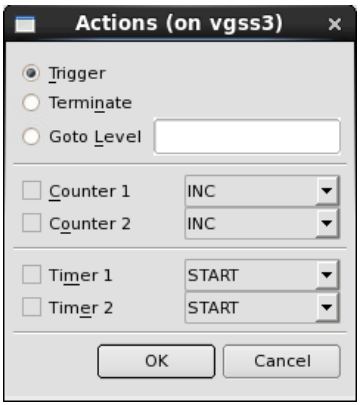


Figure: Actions Form

NOTE: During complex event capturing, the timer actions override the timer settings of the goto level.

5. Edit other branches. Click the **More** or **Less** buttons located under the **Level** arrows for additional counters.
6. Click the right arrow on the upper-right corner to display the next level.
7. Edit other levels following the steps described above.
8. At least one-trigger action must be specified in the last level of the complex event.
9. Click the **OK** button to finish editing or click the **Cancel** button to close the *Edit Complex Event* window without saving any changes.

Notes for Complex Event Rules

A complex event is evaluated according to the following rules:

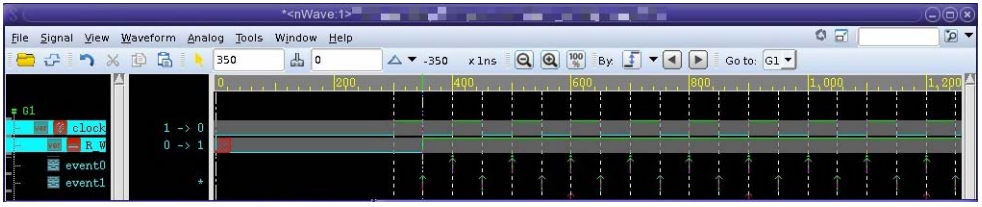
- A complex event is evaluated by the sampled time when all related signals change the values. See Case 1 as an example.
- In each level, when all branches of the if-else branch statements are evaluated, the complex event goes to the branch that takes its first true. See Case 2 as an example.

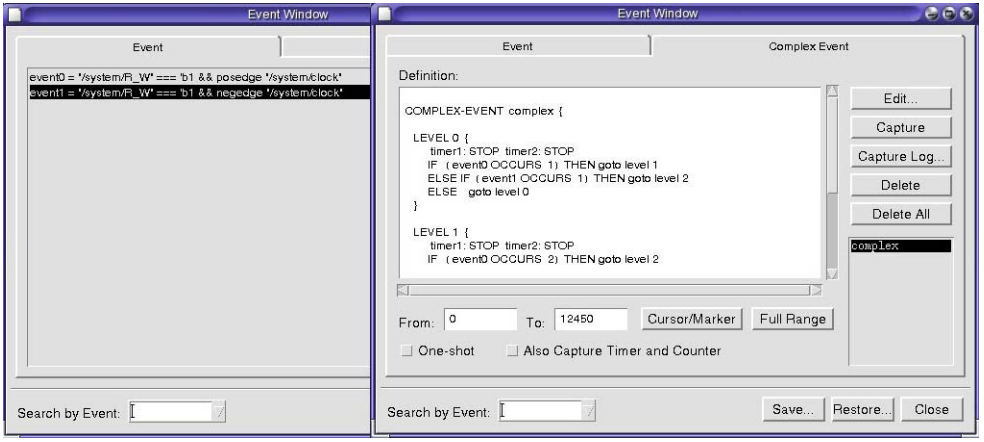
- The timer states specified in the beginning of each level impacts the evaluation in this level. See Case 3 as an example.
- The timer states in branch actions impacts the captured timer value in the waveform. If a timer state is not specified in a branch, the timer state in the current level is used. See Case 3 as an example.
- The timer/counter is updated when the complex event goes to a new level. When a trigger point is hit, the timer/counter is reset. See Case 3 as an example.
- After a trigger point is hit, the complex event is reset and go to level 0 in the next sampled time. See Case 3 as an example.

Case 1

The complex event contains two related events: `event0` and `event1`. The events are formed by the signal `/system/clock` and `/system/R_W` respectively. The complex event is evaluated by the sampled time when the related signals change values (at grid time shown in the waveform).

Following is an example of the complex event waveform with the complex event explained in detail:

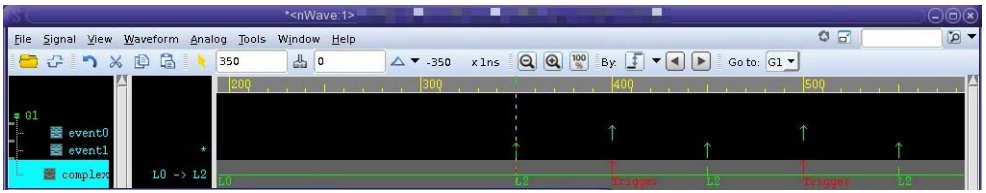


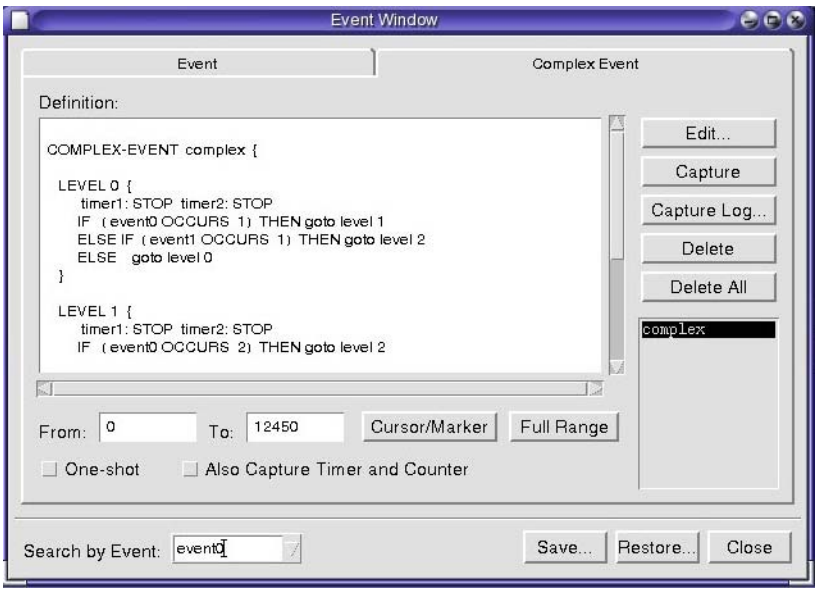


Case 2

In `LEVEL 0`, as `event1` occurs one time first, the evaluation takes the `ELSE IF` branch and the complex event goes to `level 2`.

Following is an example of the complex event waveform with the complex event explained in detail:





Case 3

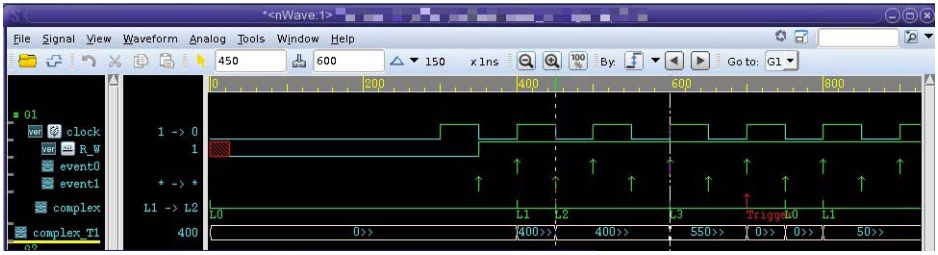
In `LEVEL 0`, `timer1` starts in this level. After `event 0` occurs one time, the complex event goes to `level 1`. The value of `timer1` captured is *400ns* after leaving this level.

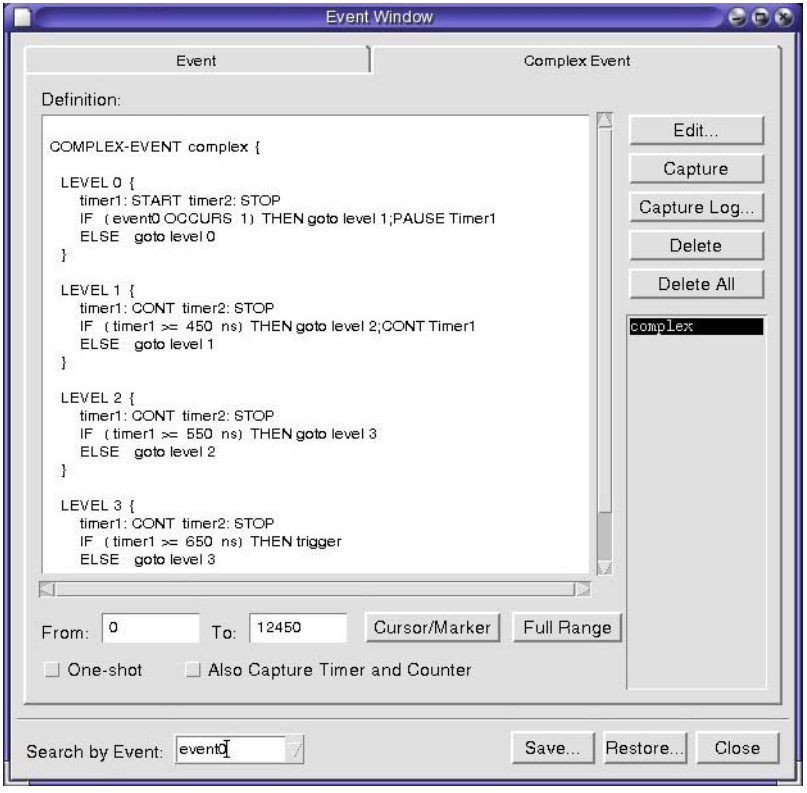
In `LEVEL 1`, as `timer1` continues to count, the complex event goes to `level 2`. However, as `timer1` pauses because of the branch action in `level 0`, the captured value of `timer1` still remains at *400ns* after the complex event leaves this level.

In `LEVEL 2`, as `timer1` continues to count, the complex event goes to `level 3`. As `timer1` continues because of the branch action in `level 1`, the captured value of `timer1` is updated to *550ns* after the complex event leaves this level.

In `LEVEL 3`, as `timer1` continues to count, the complex event finds a trigger point, and the captured value of `timer1` is reset to *0ns*. Then, the complex event is reset and goes to `level 0` in the next sampled time.

Following is an example of the complex event waveform with the complex event explained in detail:





Syntax of Complex and Simple Events

The following section includes a syntax of complex event program that includes simple event.

```
statement := COMPLEX_EVENT_BEGIN
                  <simple_event_list>
                  <complex_event_list>
                  COMPLEX_EVENT_END
simple_event_statement_list :=  <simple_event_statement> |
                                <simple_event_statement>
                                 <simple_event_statement_list>
complex_event_statement_list :=  <complex_event_statement> |
                                 <complex_event_statement>
                                  <complex_event_statement_list>
simple_event_statement := EVENT <event_name> <event_expression>
event_expression :=  <signal_condition> | <signal_condition>
                       operator <event_expression>
signal_condition := <signal_name> <operator> <value>
r_operator := '==' | '!=' | '>=' | '>' | '<=' | '<'
operator := '~' | '!' | '+' | '-' | <r_operator> | '>>' | '<<' |
            '===' | '!==' | '&' | '^' | '~^' | '|' | '&&' | '||'
|
            'or' | posedge | negedge | anychange
value := 'h[0-9a-f]+ | 'b[01]+
complex_event_statement := COMPLEX-EVENT <event_name> '{'
                           <level_statement_list> '}'
level_statement_list := <level_statement> |
                        <level_statement> <level_statement_list>
level_statement := LEVEL <number> '{' timer1: <t_act>
                    timer2:  <t_act> <branch_statement_list> '}'
branch_statement_list := <branch_statement> | <branch_statement>
                          <branch_statement_list>
branch_statement := <if_branch_list> <else_branch>
if_branch_list : = <if_branch> | <if_branch> <if_branch_list>
if_branch := IF '(' <condition> ')' THEN <branch_action>
             <timer_action> <counter_action> |
             ELSE IF '(' <condition> ')' THEN <branch_action>
             <timer_action> <counter_action> |
else_branch := ELSE <branch_action> <timer_action>
               <counter_action>
condition := <term_name> OCCURS <number> Times | <term_name>
LASTS
             <number> <time_unit> <counter_name> <r_operator>
             <number> Times | <timer_name> <r_operator> <number>
             <time_unit>
term_name :=  <event_name> | unused
timer_name := Timer1 | Timer2
counter_name := Counter1 | Counter2
time_unit := fs | ps | ns | us | ms | s
branch_action := trigger | terminate | goto level <number>
timer_action :=  | ';'<t_act> <timer_name>
```

```
t_act :=  START | STOP | PAUSE | CONT
counter_action :=  | ';'<c_act> <counter_name>
c_act :=  INC | DEC | RESET
event_name := [a-zA-Z_][a-zA-Z0-9_$]*
signal_name : = [a-zA-Z_][a-zA-Z0-9_$]*
number := [0-9]+
```

NOTE:

1. The complex event must have only one `Trigger` action in all levels.
2. The maximum branch number of each level is *4*.

View Menu Options

Hierarchical Name

Menu Bar: **View -> Hierarchical Name**

Bind Key: H

When this toggle option is turned on, each signal name appears with its full hierarchical path in the signal pane. For example, *CYCLE* is displayed as */system/CYCLE* after the **Hierarchical Name** option is turned *on*.

Signal Type

Menu Bar: **View -> Signal Type**

When this toggle option is turned on, the signal type of each signal name is displayed in the signal pane. The available types to display are: **Input, Output, Inout, Net, Register,** and **Others**. Alternatively, use the **Display Signal Type** option in the **Waveform** -> **View Options** -> **Signal Pane** page of the *Preferences* form to display the signal type ahead of each signal name.

The **Signal -> Select** options can also be used to select and highlight the signal type in the signal pane.

Highlight Selected Signals

Menu Bar: **View -> Highlight Selected Signals**

Bind Key: Shift+H

When this toggle option is turned on, selecting a signal in the signal pane highlights its waveform in the waveform pane. When this toggle option is turned off, the waveform of a selected signal is not highlighted.

Values at Cursor/Marker

Menu Bar: **View -> Values at Cursor/Marker**

The signal value displayed in the value pane is the value at the cursor position by default. When this toggle option is turned on, the value at the marker position is also shown next to (on the right of) the cursor value in the value pane.

Leading Zeros

Menu Bar: **View -> Leading Zeros**

When this toggle option is turned on, leading zeros are displayed for signal values (including signals within transactions) in both the value pane and waveform pane. The number of leading zeros depends on the format set for the signal value.

To always enable leading zeros in the future, turn the **Leading Zeros** option on in the **Waveform** -> **View Options** -> **Value Pane** page of the *Preferences* form.

Display Glitch

Menu Bar: **View -> Display Glitch**

Bind Key: Shift+G

When this toggle option is turned on, if any glitches (0 time events) exist, glitches are shown in the waveform pane. To activate this option, an FSDB file (generated with the *NOVAS_FSDB_ENV_MAX_GLITCH_NUM* environment variable set to a value of *0* or greater than *1*) must be loaded. Use the **Help -> About FSDB** option in the main window to check the status of this environment variable for

the currently loaded FSDB file. If the FSDB file contains glitches and the **Display Glitch** option in the **Waveform** -> **View Options** -> **Waveform Pane** page of the *Preferences* form is turned on, this toggle option is automatically turned on.

NOTE: If the value of `FSDB_ENV_MAX_GLITCH_NUM` is *0*, all glitches are dumped.

Dense Block Drawing

Menu Bar: **View -> Dense Block Drawing**

Bind Key: Shift+B

When this toggle option is turned on, dense block drawing signals for fast drawing of VHDL records are viewed in the following figure:

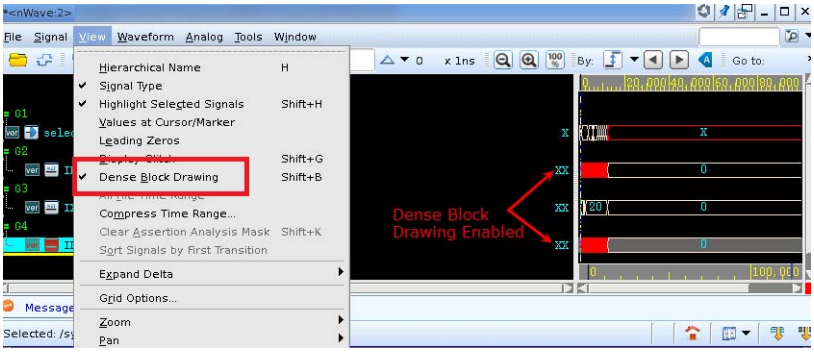


Figure: Example of Dense Block Drawing Turned On

When this toggle option is turned *on*, dense block drawing signals for analog signals are viewed.

NOTE: If the value of `WAVE_DISABLE_ANALOG_FAST_DRAW` is 1, then dense block drawings are not displayed for analog signals.

When this toggle option is turned off, the dense block is not colored, as illustrated in the following figure:

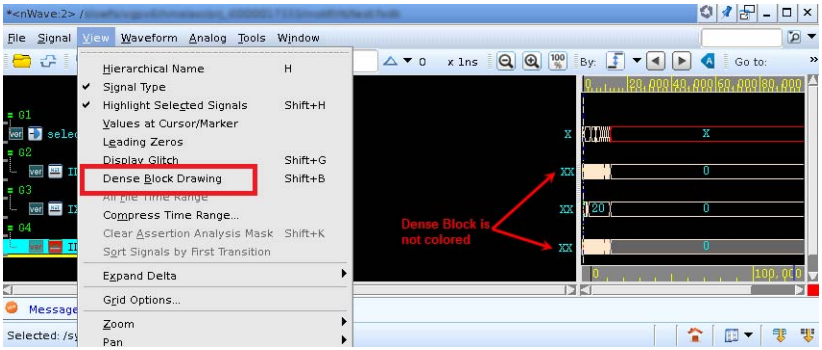


Figure: Example of Dense Block Turned Off

The composite (record, struct, and MDA) dense rectangle color can be configured at **Tools -> Preference -> Waveform** page **-> Color/Pattern -> Color -> Composite Dense Block Color**. The default is *on*.

Downsampling of Signals Drawn by Dense Block Drawing

Following are the identifications to distinguish if the current analog waveform is drawn by dense block drawing with downsampling as illustrated in the following figure:

1. A short-dashed purple line is drawn for analog dense waveform. The color of the line can be configured at **Tools -> Preference -> Waveform** page **-> Color/Pattern -> Color -> Analog Dense Block Color**.
2. The information icon is available on the left-top side of the waveform which provides information that the analog signals are drawn by dense block drawing with Downsampling.
3. Also a tooltip is shown when you hover cursor on the information icon that the analog signals are drawn by dense block drawing with Downsampling.

NOTE: If the value of `WAVE_DISABLE_ANALOG_DOWNSAMPLING_TIPS` is 1, then the information for downsampling for dense block drawings are not displayed for analog signals.

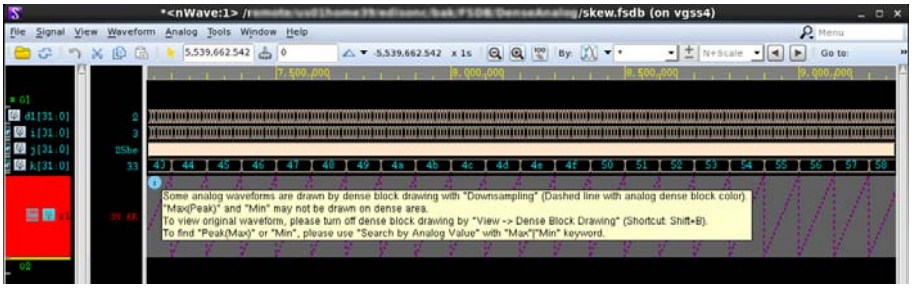


Figure: Example of Downsampling Analog Signals

All File Time Range

Menu Bar: **View -> All File Time Range**

When this toggle option is turned on, the maximum time range of all open FSDB files, instead of only the active file, is displayed.

Compress Time Range

Menu Bar: **View -> Compress Time Range**

This option draws grid lines in the current waveform pane using the time range (period) in the FSDB file without dumping or a specified time range. The compressed time range is the time period during which a waveform of a specified time range is compressed (or hidden).

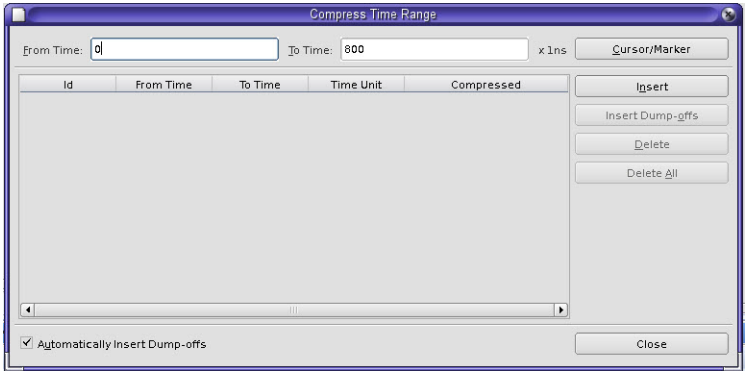


Figure: Compress Time Range Form

The **From Time/To Time** text fields define the time range during which grids are to be drawn. You can specify the time range by typing values directly into the **From Time** and **To Time** text fields, or by clicking the **Cursor/Marker** button to load the range defined by the **Cursor** and **Marker** positions in the current waveform pane.

After specifying the **From Time** and **To Time** values, click **Insert** to insert the time range specified into the compressed time range list. The specified time range is also displayed in the current waveform pane as a vertical bar. Inserting a compressed time range that overlaps with an existing time range is not allowed.

If the check box in the **Compressed** column is checked (enabled), the compressed time range displayed in the waveform pane is collapsed and a "+" mark is provided for expanding the ranges. If the **Compressed** check box is unchecked (disabled), the compressed time range displayed in the waveform pane is expanded with a yellow line between the "-" marks added. Click the left - mark to collapse the ranges to the left and click the right - mark to collapse the ranges to the right. The **Compressed** check box is checked by default.

An example of the compressed time ranges displayed in the waveform pane is shown below.

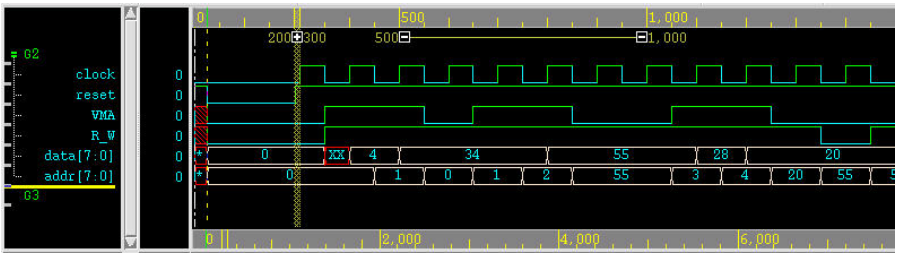


Figure: Compressed Time Range in the nWave Window

Click **Insert Dump-offs** to automatically insert the time ranges in the FSDB file without dump information (as specified by `fsdbDumpvarsOff`/ `fsdbDumpvarsOn` dumping options during simulation). Before compression, the time ranges without dumping are displayed in the current waveform pane as blue boxes.

Left-click to select a compressed time range (only one can be selected at a time) and then click **Delete** to remove it from the list. The selected compressed time range is also removed from the current waveform pane.

Click **Delete All** to remove all compressed time ranges from the compressed time range list. All the compressed time ranges in the current waveform pane are also removed.

When the **Automatically Insert Dump-offs** option is turned on, all dump-off regions are automatically collapsed and the compressed region is inserted into the *Compress Time Range* form. The default is *on*.

Clear Assertion Analysis Mask

Menu Bar: **View -> Clear Assertion Analysis Mask**

Bind Key: Shift+K

This option removes the mask drawn for the non-active time range in the waveform during assertion analysis (the mask is drawn when the **Mask Non-active Time Range in nWave** option in the *Assertion Analyzer Options* form is turned on). The option is disabled after the mask is removed.

Power

NOTE: The **Power** subcommands are available when a CPF/UPF file is loaded.

Mask Power Off

Menu Bar: **View -> Power -> Mask Power Off**

Bind Key: Ctrl+M

Refer to the **Mask Power Off** option description in the *Power Aware Debug* chapter for details.

Mask Isolation

Menu Bar: **View -> Power -> Mask Isolation**

Bind Key: Ctrl+I

Refer to the **Mask Isolation** option description in the *Power Aware Debug* chapter for details.

Mask Retention

Menu Bar: **View -> Power -> Mask Retention**

Refer to the **Mask Isolation** option description in the *Power Aware Debug* chapter for details.

Mask Driving Power Off

Menu Bar: **View -> Power -> Mask Driving Power Off**

Refer to the **Mask Driving Power Off** option description in the *Power Aware Debug* chapter for details.

Mask Toggle

Menu Bar: **View -> Power -> Mask Toggle**

Bind Key: Ctrl+G

Refer to the **Mask Toggle** option description in the *Power Aware Debug* chapter for details.

Sort Signals by First Transition

Menu Bar: **View -> Sort Signals by First Transition**

This option is activated when one or multiple groups in the signal pane are selected. When this option is invoked, all signals in the selected groups, including subgroup signals, are sorted by the first transition (value change). The initial unknown value or the glitch at the begin time is ignored during the sorting.

Expand Delta

Region Mode

Menu Bar: **View -> Expand Delta -> Region Mode**

NOTE: This option is enabled after an FSDB file with region information is loaded.

When this toggle option is turned on, region mode is enabled and a check mark is displayed in front of the option name. When this toggle option is turned off,

sequence mode is enabled and no check mark is displayed in front of the option name. The default is sequence mode.

Sequence mode shows the sequence of each value change for different signals in the same time slot. The time for the cursor/marker location are displayed in the **Time** field on the toolbar, and a *:N* value is appended after the **Time** field to indicate the sequence number for the value change. The sequence number label changes as the cursor/marker is moved within the expanded area.

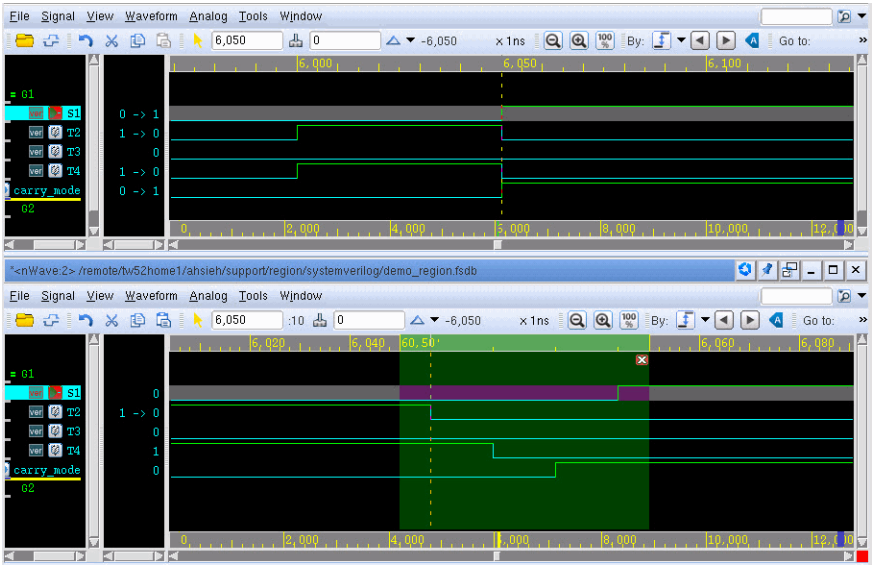


Figure: Sequence Region Expansion

Region mode shows the value changes in each region. The region change cycles (Active -> NBA -> Re-active -> Re-NBA is a cycle) is calculated for each signal, and the union region cycles for all signals are shown. The time for the cursor/marker location is displayed in the **Time** field on the toolbar and a *:region-type (cycle)* value is appended after the **Time** field to indicate the region type and cycle for the value change. The region type/cycle label changes as the cursor/marker is moved within the expanded area.

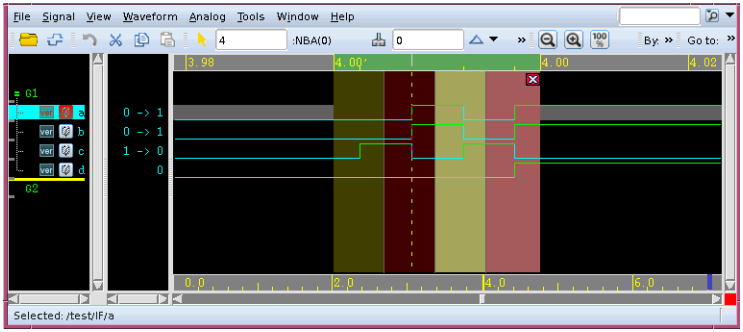


Figure: Region Mode Expansion

Display Dump-off Region in the Waveform

In the region mode, the dump-off region appears in gray and the cursor text displays as **Off**, as illustrated in the following figure:

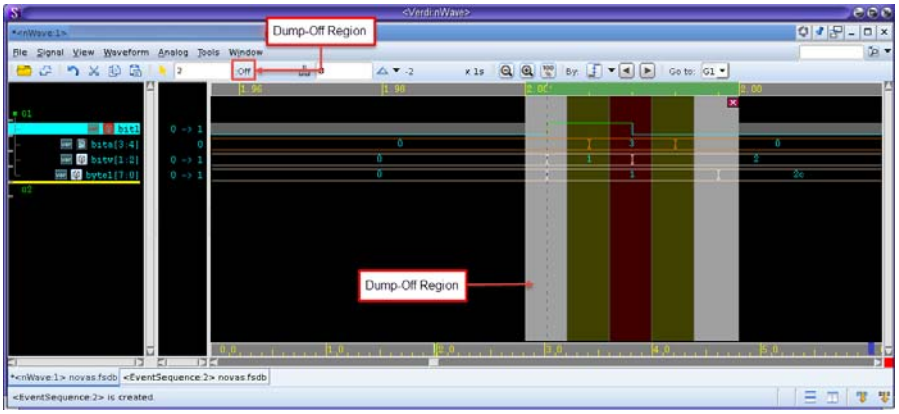


Figure: Example of Dump-off Region in Region Mode

Expand/Collapse Time at Cursor

Menu Bar:	**View -> Expand Delta -> Expand/Collapse Time at Cursor**
Bind Key:	W

NOTE: This option is enabled after an FSDB file with sequence or region information is loaded.

This toggle option expands or collapses an area with transition sequences or ordered regions. The default is collapsed.

Grid Options

Menu Bar: **View -> Grid Options**

This option opens the *Grid Options* form where the options for drawing grid lines in the waveform pane can be specified.

After making selections, click the **Apply** button to apply the grid option settings to the current *nWave* window and keep the form open.

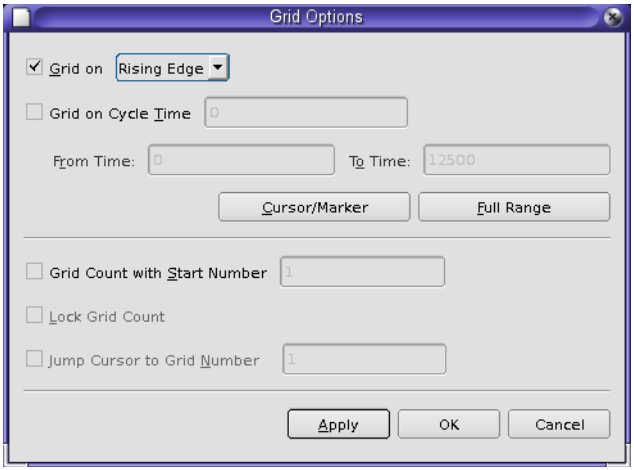


Figure: Grid Options Form

The *Grid Options* form includes the following options and fields:

- **Grid on:** When this option is turned on, the location for where to draw grid lines for the selected digital signal can be selected from the selection field. The options include **Rising Edge**, **Falling Edge**, and **Both Edges**.
- **Grid on Cycle Time:** When this option is turned on, enter a specific cycle time in the text field. The time value is based on the current window time unit.

NOTE: Only one of the **Grid on Rising Edge/Falling Edge/Both Edges** and **Grid on Cycle Time** options can be turned on at one time. The grids drawn in the waveform pane are removed when these two options are turned off.

- **From Time/To Time:** These text fields define the time range during which grid lines can be drawn. The default is the full range of the current window. The time range can be specified by following methods:

  - Type values directly into the **From Time** and **To Time** text fields.
  - Click the **Cursor/Marker** button to load the range defined by the **Cursor** and **Marker** positions in the current window.
  - Click the **Full Range** button to obtain the full range of the current window. This is the default option.
- **Cursor/Marker:** Click this button to load the **From Time** and **To Time** values based on the **Cursor** and **Marker** positions in the current waveform window. *nWave* automatically assigns the smaller value to the **From Time** text field and the larger value to the **To Time** text field.
- **Full Range:** Click this button to specify the full range of the current window.
- **Grid Count with Start Number:** This option is enabled when either the **Grid on Rising Edge**/**Falling Edge**/**Both Edges** or **Grid on Cycle Time** options are turned on. When this option is turned on, enter an integer in the text field to specify the starting index number for the grid line associated with the **Cursor** in the waveform pane. The default is *1*.
- **Lock Grid Count:** This option is enabled when the **Grid Count with Start Number** option is turned on with either of the **Grid on Rising Edge**/**Falling Edge**/**Both Edges** or **Grid on Cycle Time** options also turned *on*. When this option is turned on, the grid number is fixed. When this option is turned off, the grid number can be changed.
- **Jump Cursor to Grid Number:** This option is enabled only when the **Lock Grid Count** option is turned on. When this option is turned on, the cursor jumps to the index number for the grid line specified in the text field. The default is *1*.

Zoom

Zoom In

Menu Bar:	**View -> Zoom -> Zoom In**
Toolbar Icon:	
Bind Key:	Shift+Z

This option provides a close-up view of the content in the waveform pane. The magnification of the viewing area is changed to half the magnification of the previous view.

A full-scale ruler at the bottom of the waveform pane shows the full simulation time span of the opened dump file. Specify a time range to view with a left-click (to set the cursor) and a middle-click (to set the marker) at times of interest on this ruler, then click-right (zoom in on the time range between the cursor and the marker) to view all waveforms between the two points. Alternatively, perform a quick **Zoom In** by dragging-left (press and hold the left mouse button) along the full-scale ruler at the bottom of the waveform pane or between the full-scale ruler and the waveform pane.

Zoom Out

Menu Bar:	**View -> Zoom -> Zoom Out**
Toolbar Icon:	
Bind Key:	Z

This option enables more of the content to be seen in the waveform pane at a reduced size. The magnification of the viewing area is changed to two times the magnification of the previous view.

Zoom All

Menu Bar:	**View -> Zoom -> Zoom All**
Toolbar Icon:	
Bind Key:	F

This option displays waveforms for the entire simulation time range.

Zoom Cursor

Menu Bar:	**View -> Zoom -> Zoom Cursor**
Toolbar Icon:	

This option displays the time range between the cursor and the marker in more detail. Left-click to set the cursor position and middle-click to set the marker position in the waveform pane.

Zoom Window

Menu Bar: **View -> Zoom -> Zoom Window**

This option zooms to a precise time range by specifying the **Start Time** and **End Time** in the *Zoom Window* form.

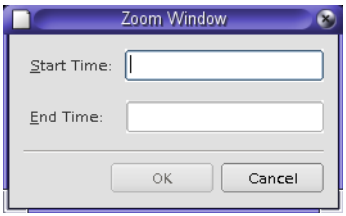


Figure: Zoom Window

Pan

Pan Left

Menu Bar: **View -> Pan -> Pan Left**

Bind Keys: Left Arrow (page), Ctrl+Left Arrow (small step)

This option pans the waveform pane to the left.

Pan Right

Menu Bar: **View -> Pan -> Pan Right**

Bind Keys: Right Arrow (page), Ctrl+Right Arrow (small step)

This option pans the waveform pane to the right.

NOTE: The horizontal scroll bar can be moved in the waveform pane to pan left and right.

Pan Up

Menu Bar: **View -> Pan -> Pan Up**

Bind Keys: Page Up, Up Arrow

This option pans the waveform pane up.

Pan Down

Menu Bar: **View -> Pan -> Pan Down**

Bind Keys: **Page Down, Down Arrow**

This option pans the waveform pane down.

NOTE: The vertical scroll bar can be moved in the waveform pane to pan up and down.

Last View

Menu Bar: **View -> Last View**

Bind Key: L

This option shows the previous view in the waveform pane. The **Last View** option only displays *one* previous view, so when this option is invoked more than once, it switches between the current and the previous view.

Signal Event Report

Menu Bar: **View -> Signal Event Report**

NOTE: After an FSDB file is opened in the *nWave* window, the cursor and marker should be set, and then the signals of interest selected before invoking this option.

This option opens the *Signal Event Report* form where the statistical data of an event is shown. A signal can also be added to the signal event table by dragging it from another window and dropping it to the signal event table. Multiple signal selection is supported.

The column headings of the signal event table are summarized as follows:

- **Signal Name:** Show the signal name.
- **Signal Type:** Show the signal type.

- **Rising#:** Show the number of rising edges of the signal during the specified time period.
- **Falling#:** Show the number of falling edges of the signal during the specified time period.
- **Transition Frequency:** Show the transition frequency of the selected signal.
- **VC#:** Show the number of value changes occurred during **From Time** and **To Time**. The **VC#/TC#** and **Trigger Time** columns, showing the number of triggers and trigger time respectively, only appear when one of selected signals is a complex event.
- **Duty Cycle:** Show the percentage of the period when the signal is active (that is, value = 1). The period is calculated from **From Time** to **To Time**. For unsupported signals, *N/A* is shown.
- **Longest Duration:** Show the longest period of which a signal is active (that is, value = 1) between **From Time** and **To Time**. For unsupported signals, *N/A* is shown.

NOTE: Changes in value and glitches stop the accumulation of the longest duration.

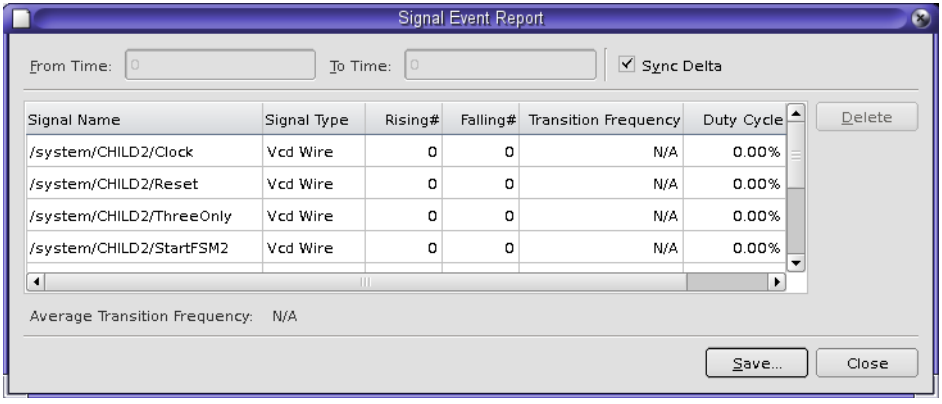


Figure: Signal Event Report Form

The *Signal Event Report* form includes the following option and fields:

- **From Time** and **To Time:** Specify the time range in these text fields.
- **Sync Delta:** When this option is turned on, the cursor time and marker time in the current waveform window are used as the **From Time** and **To Time** values. When the **Sync Delta** option is turned on, the **From Time** and **To Time** text fields are disabled. The default is *on*.

- **Transition Frequency:** The average transition frequency for the **Transition Frequency** column is shown.
- **Delete:** Delete the selected signal row from the list.
- **Save:** Save the *Signal Event Report* to a file.
- **Close:** Close the *Signal Event Report* form.

Group Manager

Menu Bar: **View -> Group Manager**

This option opens the *Group Manager* form where a new subgroup can be added to existing groups or the selected signals can be moved to a different group.

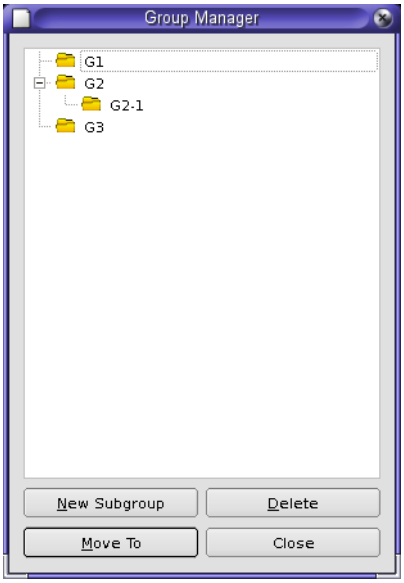


Figure: Group Manager Form

The *Group Manager* form includes the following buttons:

- **New Subgroup:** Create a subgroup under the selected group. The name of the new subgroup concatenates the name of the parent, with a dash and a serial number. For example, *G1-1* and *G1-2* are subgroups of *G1*.
- **Delete:** Delete the selected group/subgroup with its descendants.
- **Move To:** Click this button to move the selected signals in *nWave* to the selected group/subgroup in the *Group Manager* form.

Waveform Menu Options

Auto Update

Menu Bar: **Waveform -> Auto Update**

This toggle option automatically updates the simulation results being viewed while the simulation is running. The option is enabled if the simulation run for one of the opened FSDB files is not complete. When the simulation runs for all opened FSDB files are complete, this option is disabled.

The **Auto Update** option can be used in two ways:

- To update the display by continually monitoring the opened file.
- To integrate with the simulator and update the window on the fly.

nWave then turns on the **Auto Update** mode automatically. If the restore file is not specified in the configuration file, *nWave* opens the *Get Signals* form where the signals to display can be specified.

During the **Auto Update** operation, press **Ctrl+C** or **Escape** to turn the auto update mode off at any time, analyze the waveform, then invoke the **Waveform -> Auto Update** option to resume the screen updating. Alternatively, invoke any option from the menu using a bind key or a mouse click during the auto update period.

Spacing

Menu Bar: **Waveform -> Spacing**

This option defines the spacing between signals. A *Signal Spacing* form appears for specifying the spacing with the height measured in pixels. The **Default** button applies the default value specified in the *novas.rc* resource file.

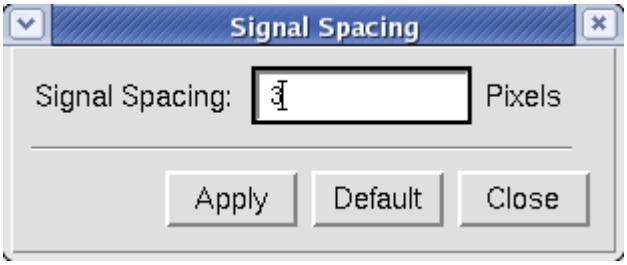


Figure: Signal Spacing Form

Height

Menu Bar: **Waveform -> Height**

This option sets the height of displayed signals but is enabled only when one or more signals are selected from the signal pane. After selecting the signals, the option opens the *Signal Height* form. The height is measured in pixels. The **Default** button applies the default value specified in the *novas.rc* resource file.

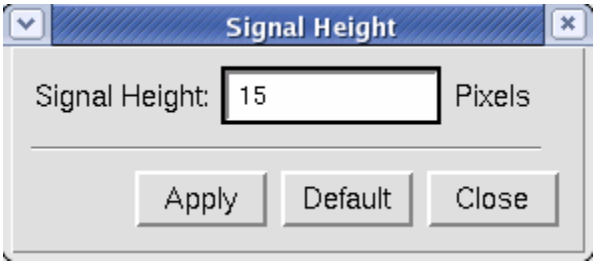


Figure: Signal Height Form

As a shortcut, to increase the height of a selected signal, you can also use **Alt** + mouse wheel up in the signal pane. Similarly, to decrease the height of the selected signal, you can use **Alt** + mouse wheel down in the signal pane.

To make scrolling faster, you can also use **Ctrl +** mouse wheel up in the signal pane.

Fit Waveform

Menu Bar: **Waveform -> Fit Waveform**

Bind Key: Ctrl+L

This option fits all the selected signals (including analog signals, if available) in the *nWave* frame.

The following figure illustrates an example of the selected waveforms that fit in the *nWave* window:

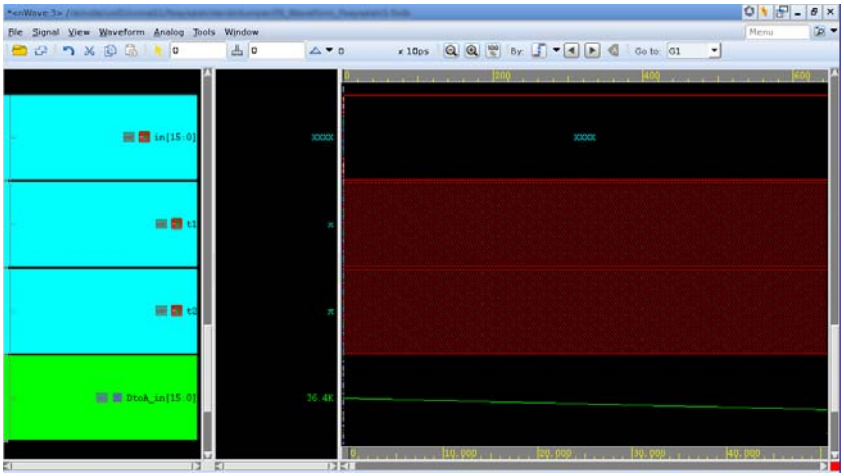


Figure: Example of Waveforms Fitted in the nWave Frame

NOTE: This option is disabled when you select the group name, for example G1, G2, and so on, or if you do not select any signal.

Color/Pattern

Menu Bar: **Waveform -> Color/Pattern**

Bind Key: C

This option invokes the *Change Color/Pattern* form where the **Color**, **Line Width**, and **Line Styles** for the selected signals can be specified. The new colors can be immediately viewed in the waveform pane. The signal names and values

displayed in the signal and value panes also change to the specified color. The default settings cannot be changed via the *novas.rc* resource file.

Figure: Change Color/Pattern Form

Signal Value Radix

Menu Bar: **Waveform -> Signal Value Radix**

NOTE: This option is available when the target signal is selected (multiple selections are supported). The selected signal must be a bus signal r a composite signal.
An alternative way to invoke the **Signal Value Radix** options is to right-click the desired signal (the right-click options only work for a single selected signal) in the value pane to open the **Format** right-click menu.

This option shows the value radix of the selected signal.

NOTE: If the format of one instance of a signal is changed, all other instances of that signal automatically adopts the same format.

Binary

Menu Bar: **Waveform -> Signal Value Radix -> Binary**

This option displays the signal value in **Binary** format in the value pane and the waveform pane.

Octal

Menu Bar: **Waveform -> Signal Value Radix -> Octal**

This option displays the signal value in **Octal** format in the value pane and the waveform pane.

Hexadecimal

Menu Bar: **Waveform -> Signal Value Radix -> Hexadecimal**

This option displays the signal value in **Hexadecimal** format in the value pane and the waveform pane. This option is set as the default when *nWave* is invoked for the first time.

Decimal

Menu Bar: **Waveform -> Signal Value Radix -> Decimal**

This option displays the signal value in **Decimal** format in the value pane and the waveform pane.

ASCII

Menu Bar: **Waveform -> Signal Value Radix -> ASCII**

This option displays the signal value in **ASCII** format in the value pane and the waveform pane. The **ASCII** format treats bytes as ASCII codes and displays the corresponding ASCII characters.

Enumerated Literal

Menu Bar: **Waveform -> Signal Value Radix -> Enumerated Literal**

This option displays the signal value in **Enumerated Literal** format in the value pane and the waveform pane.

NOTE: When radix is applied to the composite signals then all members of the signals apply the radix, expect the member signals which are of Enum type. However, you can change the format of the Enum type member signals by selecting the members directly and then applying radix.

IEEE-754 Floating Point

Menu Bar: **Waveform -> Signal Value Radix -> IEEE-754 Floating Point**

This option displays the signal value in **IEEE-754 Floating Point** format in the value pane and the waveform pane.

Add Alias From File

Menu Bar: **Waveform -> Signal Value Radix -> Add Alias from File**

This option associates an alias definition file with the selected signals. The aliases are displayed instead of numerical values when the signal value matches the predefined alias string.

Syntax of Alias Definition

```
alias alias_name
            alias_string   value|value1~value2   colorID
            alias_string   value|value1~value2   colorID
            . . .
endalias
slice slice_name
            bit_range   alias_name
            bit_range   alias_name
            . . .
endslice
condAlias conditional_alias_name
            conditional_signal1_name  alias_name|slice_name
            conditional_signal2_name  alias_name|slice_name
            . . .
endcondAlias
curAliasTable  alias_name|slice_name|conditional_alias_name
```

where the fields have the following meanings:

- `alias`: A keyword defining the start of a group of aliases

- `endalias`: A keyword defining the end of a group of aliases
- `alias_name`: A text string to use for naming a group of aliases
- `alias_string`: A text string to use for naming an alias (such as ZERO-to-FIVE).
- `value`: A single match value
- `value1~value2`: A range of values (the "~" character is used as a range delimiter indicating "*from value1 to value2*")
- `slice`: A keyword defining the start of a slice definition
- `endslice`: A keyword defining the end of a slice definition
- `slice_name`: A text string to use for naming a slice definition
- `bit_range`: A range of bits to be associated with a particular group of aliases
- `curAliasTable`: A keyword defining the first alias to apply. If not specified, the priority for the first applied alias name is searched by the following sequence:

  1. First-found conditional table
  2. First-found slice
  3. First-found alias

NOTE:

1. The keywords `alias`, `endalias`, `slice`, `endslice`, and ~ are keywords in the alias file. They are NOT case-sensitive.
2. If a word contains special characters or blank spaces, then that word must have double quotes around it (such as "ZERO to FIVE" or " ").
3. 'U' is not allowed in an alias file (such as 8'hUU).

It is recommended that the alias file is saved with the file extension "*.alias*". An example of a basic alias file (*cpu.alias*) appears below:

```
ADD 1
SUB 2
LOAD 3
TEST 8'b?0?0?0?0
SIX-to-TEN 8'h06~8'h0a
```

NOTE: `alias` and `endalias` are optional. They can be used for multi-alias map definitions.

According to this alias file, *ADD* is displayed when the signal value is 1, *SUB* for 2, and *LOAD* for 3. *TEST* is displayed for any byte whose 0^{th}, 2^{nd}, 4^{th}, and 6^{th} bits are 0s and SIX-to-TEN is displayed when a value is in range of 8'h06~8'h0a.

NOTE: As alias value matching is based on the variable value, the value format is important. When the value is displayed in hexadecimal format, 3 is mapped to LOAD. If the format is binary, 3 becomes 011 and is not mapped to LOAD.

Specify the value in a different format using Verilog conventions (such as decimal, binary, or octal).

The following example is for a more complex alias file:

```
alias alias_1
   0-5   4'h0~4'h5
  6-10   4'h6~4'h10
endalias

alias alias_2
  ONE     8'h01
  THREE   8'h03
  DEFAULT others
endalias

alias ranged_alias
  DEFAULT        others
  "SIX to FIFTY" 8'h06~8'h0a
  ZERO-to-FIVE   8'h00~8'h05
endalias

slice slice_1
  1:0   alias_1
  3:2   alias_2
  7:6   ranged_alias
  5:4
endslice
```

Assume there is an 8-bit bus called *control[7:0]*, and this bus is associated with the slice definition *slice_1* as described above. Further assume that from 100 ns to 150 ns this bus is carrying a value of 8'h08.

According to the *slice_1* slice definition, the bus is divided into four pieces: 1:0, 3:2, 7:6, and 5:4 (the slice definition has rearranged the default ordering):

- Bits 1:0 are associated with the alias group called *alias_1*. These bits are carrying a value of *0* and are therefore assigned the alias *0-5*.

- Bits 3:2 are associated with the alias group called *alias_2*. These bits are carrying a value of *2* and are therefore assigned the alias *DEFAULT*.
- Bits 7:6 are associated with the alias group called *ranged_alias*. These bits are carrying a value of *0* and are therefore assigned the alias *ZERO-to-FIVE*.
- Bits 5:4 are not associated with any alias group, and is therefore displayed using their numerical value, which is *0*.

Thus, the value of *control[7:0]* between 100ns and 150 ns is displayed as follows:

```
0-5 DEFAULT ZERO-to-FIVE 0
```

Add Alias From Program

Menu Bar: **Waveform -> Signal Value Radix -> Add Alias from Program**

This option maps a signal value to the symbol specified in a program. A program can be written that accepts a number (or a file containing a number) and prints out the string (or a list of strings) corresponding to that number. *nWave* calls this program to obtain the alias string for display. Examples of **Add Alias from Program** programs written in C and Perl are shown below.

When this option is invoked, the value of signals is sent to the alias program and the corresponding result is obtained from the alias program's output. For each input from the Verdi platform, the alias program should write its corresponding output to *std output* (the working display/screen). If the alias program (either in C or in Perl) generates more than one output, the Verdi platform receives more results for each value it sent to the alias program, and it applies the additional results to subsequent value changes. This indicates that the displayed waveform values are incorrect.

NOTE: The child process must explicitly set the standard output buffer to NULL. If the child program is written by C or C++, the following line must be added before any print statement:

setvbuf(stdout, NULL, _IONBF, 0);

If the child program is a Perl script, the following two lines must be added before any print statement:

select(STDOUT);
$| = 1;

Example: Add Alias from Program Written in C

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct alias {
  char* number;
  char* alias;
} alias_table[] ={
"0", "HLT",
"1", "SKZ",
"10", "ADD",
"11", "AND",
"100", "XOR",
"101", "LDA",
"110", "STO",
"111", "NOP",
"1000", "NOP",
"1001", "SL2",
"1010", "SR1",
"1011", "SR2",
"1100", "SUB",
"1101", "HALT",
0,0
};

main(int argc, char *argv[])
{
  char inbuff[1024];
  int index;
   long n;

  while (1)
  {
    clearerr(stdin);
    if (fgets(inbuff, 1024, stdin) == (char*) 0) break;
    n = strlen(inbuff);
    if (n > 0)
    {
      inbuff[n] = '\0';
      if (inbuff[n-1] == '\n') inbuff[n-1] = '\0';
      if (strcmp(inbuff, "exit") ==0) return 0;
      index =0;
      while(alias_table[index].number != 0)
      {
        if (strcmp(inbuff, alias_table[index].number) ==0)
        {
          printf("%s\n", alias_table[index].alias);
          break;
        }
        index++;
      }
      if (alias_table[index].number ==0) printf("\n");
      fflush(stdout);
    }
```

```
  }
}
```

Example: Add Alias from Program Written in Perl

```
#!/usr/local/bin/perl
select(STDOUT);
while (<>) {
if ($_ == 0){
  $|=1;
  print  "zero\n";
} elsif ($_ == 1) {
  $|=1;
 print  "one\n";
} elsif ($_ == 10) {
  $|=1;
 print  "two\n";
} elsif ($_ == 11) {
  $|=1;
 print  "three\n";
} elsif ($_ == 100) {
  $|=1;
 print "four\n";
}else {
 $|=1;
 print  "O.K\n";
 }
open(txt,">>aa");
$|=1;
print txt "$_";
close(txt);
}
```

Remove Alias

Menu Bar: **Waveform -> Signal Value Radix -> Remove Alias**

This option removes any aliases from the selected signals and reverts to the display of their numerical values.

Edit Alias

Menu Bar: **Waveform -> Signal Value Radix -> Edit Alias**

This option opens the *Alias Editor* form where the alias map file can be loaded or the alias map can be defined. Two methods can be used to apply the alias map to signals:

1. Select the signals in the signal pane, then select **Waveform -> Signal Value Radix -> [Alias Map Name]**. A maximum of 20 alias names can be listed in the **Waveform -> Signal Value Radix** pull-down menu. When more than 20 alias files exist, the **More Alias** option appears. Click the **More Alias** option to open the *Alias Editor* form for viewing all alias files and making a selection.
2. Select the signals or bus in the signal pane. Right-click the value pane and select **[Alias Map Name]** in the ensuing format right-click menu.

When a signal value matches a predefined alias string, alias names are displayed instead of numerical values. An alias with a range of values or a single value is allowed. For example, the alias *SIX-to-TEN* has the value in the range of 8'h06~8'h0a. The alias name appears both in the value pane and the waveform pane.

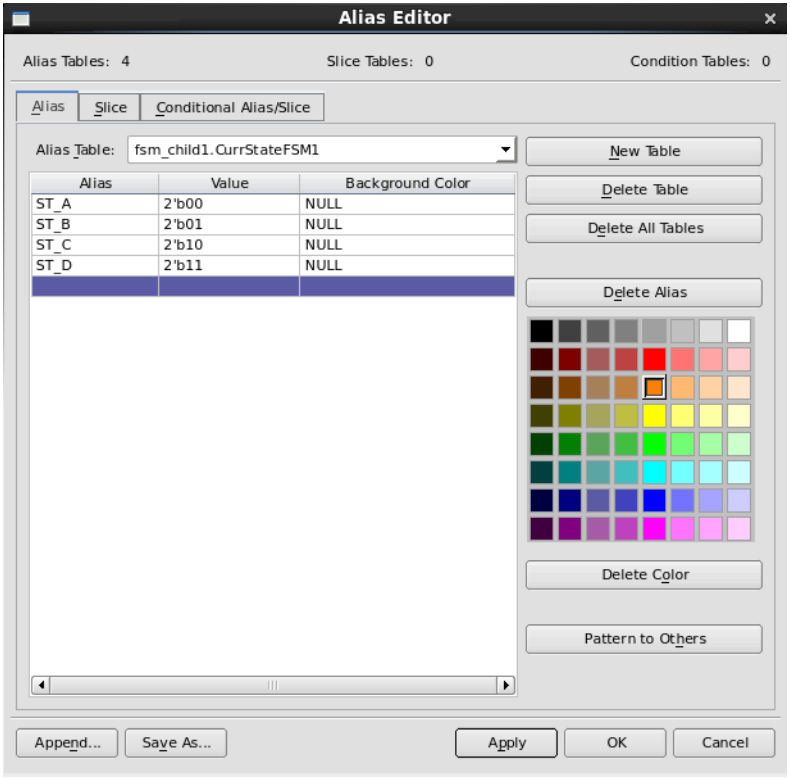


Figure: Alias Editor Form - Alias Tab

Alias Tab

The **Alias** tab can be used to modify or delete existing aliases and create new aliases. The following fields and buttons are available:

- **Alias Table:** This field contains a list of existing alias tables. You can enter or modify the alias table name by typing in the field directly and then by clicking the **Apply** button. Entering a string such as “Ranged” creates a list of alias table names beginning with “Ranged” (such as “Ranged_Alias”).
- **Alias:** Specify the alias or mnemonic name.
- **Value:** Specify the numerical value with the preferred radix type for the alias.
- **Background Color:** Specify the background color for the alias. A value can be entered manually or the preferred color can be selected from the color map.
- **New Table:** Click this button to create a new alias table. The **Alias Table** field and all columns are cleared to enable new values to be entered.
- **Delete Table:** Click this button to delete the current alias table.
- **Delete All Tables:** Click this button to delete all alias tables.
- **Delete Alias:** Click this button to delete the selected row of the current alias table.
- **Delete Color:** Click this button to delete the background color of the selected row of the current alias table.
- **Pattern to Others:** This field assigns a default alias for any unspecified values in the **Alias/Value** columns for the selected row. Double-click the **Value** column to edit the value if needed.

Slice Tab

The **Slice** tab can be used to create a slice for an alias. A bus slicing format can be defined after an alias file is loaded. If slice definitions exist in this alias file, *nWave* uses the first slice definition for the selected signal. Other slices can be selected later on.

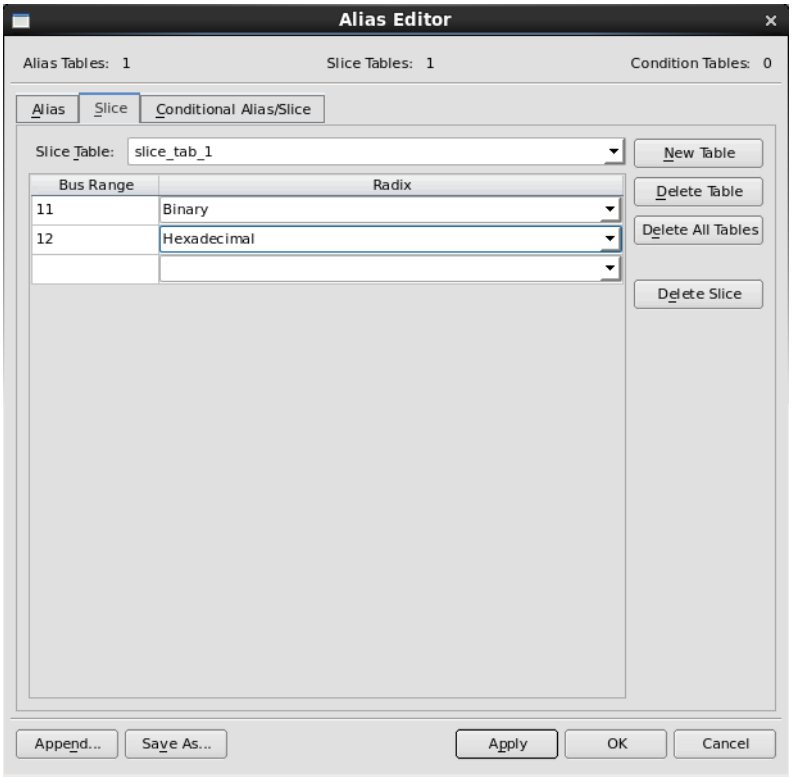


Figure: Alias Editor Form - Slice Tab

The **Slice** tab includes the following fields and buttons:

- **Slice Table**: This field contains a list of existing slice tables. You can enter or modify the slice table name by typing in the field directly and then by clicking the **Apply** button. Entering a string such as “slice” creates a list of alias table names beginning with “slice” (such as “slice_B”).
- **Bus Range**: Specify the numerical value for the slice.
- **Radix**: Indicate the preferred radix type for the slice.
- **New Table:** Click this button to create a new slice table.
- **Delete Table:** Click this button to delete the current slice table.
- **Delete All Tables:** Click this button to delete all slice tables.
- **Delete Slice:** Click this button to delete the selected row in the slice table.
- **Edit Radix:** Specify the radix for the **Radix** column of the slice list by selecting one of the following options: **Binary, Octal, Hexadecimal, Decimal, ASCII**, or the existing alias table.

Syntax of Slice Definition

```
slice slice_name
    range   alias_name
    range   alias_name
    . . .
endslice
range:= [ "a : b" |  "a"]/* bus range definition */
alias_name:= [    | alias_name]/* mapping alias */
```

`slice_name`: name of slicing map

`"a:b"`: bus range from a to b

`"a"`: single bit (bit a)

If the bus size is `n`, then `0 <= a, b <= n`.

`alias_name`: can be NULL or a defined alias map; if NULL, use the original value for display

---

NOTE:

1. `alias`, `endalias`, and ~ are keywords associated with the alias file. These keywords are not case-sensitive.
2. If a word contains special characters or blanks/spaces, then the word must have double-quotations “ ” around it.
3. Comment lines are allowed and prefixed by '#'.
4. The value display order follows the slice's definition order. If slices are defined up to down, then slice values are displayed left to right.
5. Consider a `bus[m : n]` which has `m>n`,. It is recommend that *a* is greater than `b` for each slice's bus-range `“a : b”`. Otherwise, this slice is displayed in reverse.
6. Assuming that the alias table for each slice is NULL or not available, then this slice is displayed using the current radix.

For example, assume a bus *A[7:0]* exists. Bit 0 is a parity bit; bits 1, 2, and 3 are address enable lines; and the top 4 bits act as pipeline control. The alias file *test.alias* associated with this bus is shown as follows:

```
#Old Alias Block Style
aliasA
   LOW   'h0
   HIGH   'h1

#New Alias Block Style
ALIAS ranged_alias
  "Hex Zero"    4'h0
   One2Three   "4'h1~   4'h3"
   Four2Seven   4'h4~4'h7
   Eight2Fifteen 4'h8~4'hF
ENDALIAS

SLICE slice_A
```

```
           7:4      ranged_alias
          "3:1"    ranged_alias
         0         aliasA
ENDSLICE

SLICE slice_B
           7:4
           3:1
           0
ENDSLICE

SLICE slice_C
           0     ranged_alias
           3:1
           7:4  aliasA
ENDSLICE

SLICE slice_D
           3:1   ranged_alias
ENDSLICE

SLICE slice_E
           0:3   ranged_alias
           4:7   ranged_alias
ENDSLICE
```

- If bus A[7:0] is *h' 3F (0011_1111)*:
  a. Map *slice_A* to bus A[7:0]: /* with a different alias map */

     The value of A[7:0] is *One2Three Four2Seven HIGH* (0011 111 1).

  b. Map *slice_B* to bus A[7:0]: /* no alias mapping, then show the value by the current radix */

     The value of A[7:0] is *3 7 1* (0011 111 1).

  c. Map *slice_C* to bus A[7:0]: /* slice order is relevant*/

     The value of A[7:0] is *One2Three 7 3* (1 111 0011).

  d. Map *slice_D* to bus A[7:0]: /* undefined slice, then no value is shown */

     The value of A[7:0] is *Four2Seven* (111).

  e. Map *slice_E* to bus A[7:0]: /* reverse slice's bus range, then the slice is displayed in reverse */

     The value of A[7:0] is *Eight2Fifteen Eight2Fifteen* (1111 1100).

- If bus A[7:0] is *h' 3F (0011_1x1z)*:

  Map *slice_B* to bus A[7:0]: /* no alias mapping, then show the value by the current radix. The alias is only shown if the bus value does not contain all 1's or 0's (such as the bus value includes an x, z, or u) */

The value of A[0:7] is 3 * * (0011 1x1 z) where "*" represents "not an available value".

Conditional Alias/Slice Tab

The **Conditional Alias/Slice** tab can be used to edit an alias/slice under a defined condition.

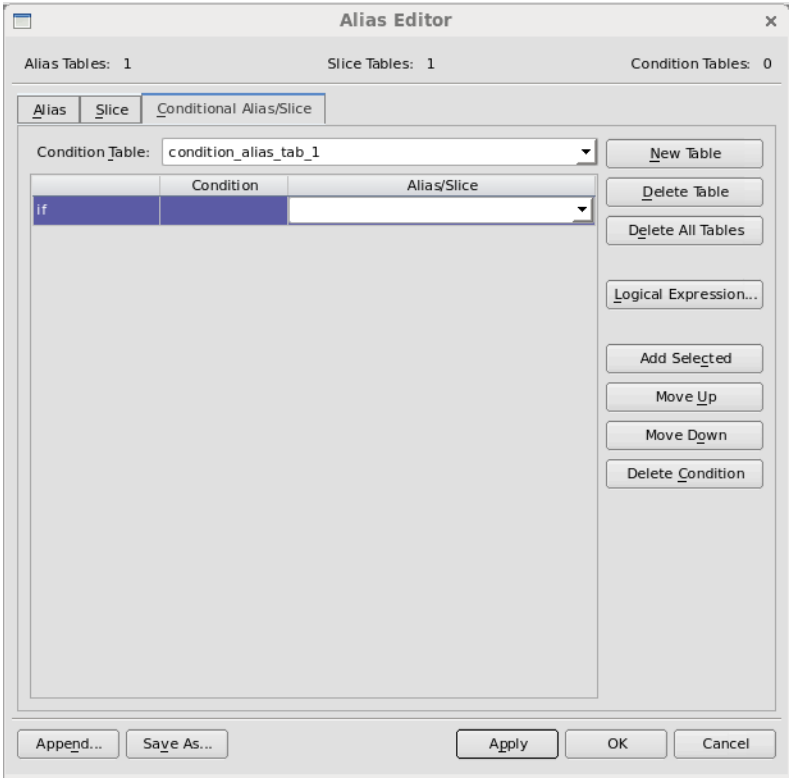


Figure: Alias Editor Form - Conditional Alias/Slice Tab

The **Conditional Alias/Slice** tab includes the following fields and buttons:

- **Condition Table**: This field contains a list of existing condition tables. You can enter the condition table name by typing in the field directly and then by clicking the **Apply** button. Entering a string such as "condition" creates a list of condition table names beginning with "condition" (such as "condition_ alias_tab").
- **Condition:** Specify the signal name with the full hierarchical path. Alternatively, dragging a signal from *nWave* and dropping it into the

condition list adds the signal to the list. Each condition is listed with the “if” or “else if” statement.

- **Alias/Slice**: Specify the alias/slice table name.
- **New Table**: Click this button to create a new condition table.
- **Delete Table**: Click this button to delete the current condition table.
- **Delete All Table**: Click this button to delete all condition tables.
- **Logical Expression**: Click this button to open the *Logical Expression* form where a logical expression can be created for the specified signals. The created logical expression can be dragged from *nWave* and dropped to the **Condition** column as the condition.
- **Add Selected**: Click this button to add the selected signal from the *nWave* window.
- **Move Up**: Click this button to move the selected condition in the list one row up.
- **Move Down**: Click this button to move the selected condition in the list one row down.
- **Delete Condition**: Click this button to remove the selected condition from the list.
- **Edit Alias/Slice**: Specify which alias/slice to edit in the **Alias/Slice** column of the conditional alias/slice list.

Append: Click this button to load the existing alias table files from the *Append Tables from Alias File* form.

Save As: Click this button to save the current alias/slice/conditional alias or slice tables to a new file.

Signal Value Notation

An alternative way to invoke the **Signal Value Notation** options is to place the cursor over the value pane in *nWave* and right-click to choose **Notation** from the ensuing **Format** right-click menu.

Unsigned

Menu Bar: **Waveform -> Signal Value Notation -> Unsigned**

This option displays the signal value in **Unsigned** format in the value pane and the waveform pane.

Signed 2's Complement

Menu Bar: **Waveform -> Signal Value Notation -> Signed 2's Complement**

This option displays the signal value in **Signed 2's Complement** format in the value pane and the waveform pane. The various signed binary number formats are shown below:

Decimal	Signed-2's complement	Signed-1's complement	Signed-magnitude
+7	0111	0111	0111
+6	0110	0110	0110
+5	0101	0101	0101
+4	0100	0100	0100
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
+0	0000	0000	0000
-0	-	1111	1000
-1	1111	1110	1001
-2	1110	1101	1010
-3	1101	1100	1011
-4	1100	1011	1100
-5	1011	1010	1101
-6	1010	1001	1110
-7	1001	1000	1111
-8	1000	-	-

Signed 1's Complement

Menu Bar: Waveform -> Signal Value Notation -> Signed 1's Complement

This option displays the signal value in **Signed 1's Complement** format in the value pane and the waveform pane. These values are listed above.

Signed Magnitude

Menu Bar: Waveform -> Signal Value Notation -> Signed Magnitude

This option displays the signal value in **Signed Magnitude** format in the value pane and the waveform pane. These values are listed above.

Analog Waveform

Menu Bar: **Waveform -> Analog Waveform**

This option has the following displays:

1. The selected bus(es). A histogram-like waveform is used to display the selected bus(es) in the waveform pane. Vertical zooming by the **Analog -> Zoom Value** option and the **Analog -> Vertical Fit** option are supported for an analog waveform from digital bus(es). An example of *addr[7:0]* displayed using the **Analog Waveform** option is shown below.

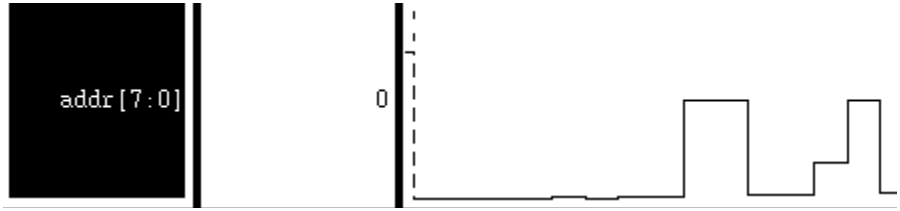


Figure: Analog Waveform of Bus

2. A single bit digital waveform. This digital to analog function can be applied to a single bit digital waveform. A sampling rate or a reference clock signal can be specified and then used to perform the D/A conversion. After the D/A conversion has been performed, all of the analog analysis functions from the Verdi platform can be applied to the new analog signal. An example of *DtoA_clock* displayed using the **Analog Waveform** option is shown below.

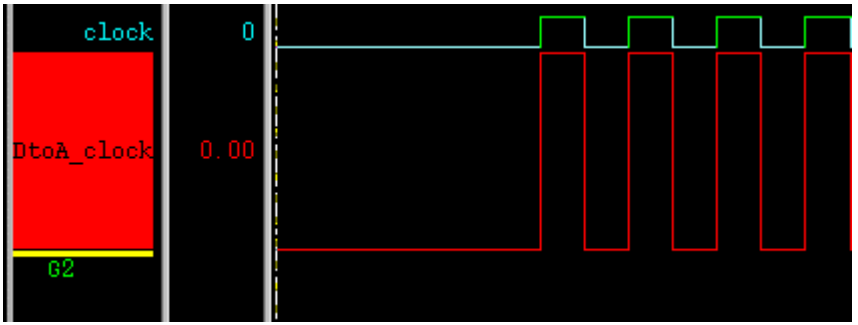


Figure: Analog Waveform of a Single Bit Digital Waveform

3. An integer signal in VHDL. *nWave* treats it as a scale signal in 2's complement. When applying, it always creates a new analog signal with the name *DtoA_{original_name}*.

Figure: Analog Waveform of an Integer Signal in VHDL

NOTE: For discrete values, use the **Piecewise Constant** or **Piecewise Linear** option on the converted analog signal.

Digital Waveform

Menu Bar: **Waveform -> Digital Waveform**

This option displays the selected signals as digital waveforms in the waveforms window. Values in the value pane show piecewise constant value for digital waveforms. The following figure shows an example of I(vdd) displayed after invoking the **Digital Waveform** option.

Figure: Digital Waveform

Invert Waveform

Menu Bar: **Waveform -> Invert Waveform**

This option can only be applied to bus signals. This option inverts all of the bit values in the value pane. The signal name is annotated with 'Invert' in the signal pane. For example, after invoking the **Invert Waveform** option on the bus signal *data[7:0]*, *data[7:0]* is displayed as *Invert(data[7:0])* in the signal pane. Also, the value of *data[7:0]* at time 0 is changed from '*00111010*' to '*11000101*' in the value pane.

Property

Expand/Shrink Overlapping

Menu Bar: **Waveform -> Property -> Expand/Shrink Overlapping**

After selecting one or more property signals and invoking this option, the un-expanded signals are expanded automatically into *n* layers where *n* is the maximum number of attempted overlapped (begin, end) time-spans. The next time this option is invoked, the selected expanded signals are shrunk to one layer.

When a session is saved, the information about the expanded property signals are saved in a signal file.

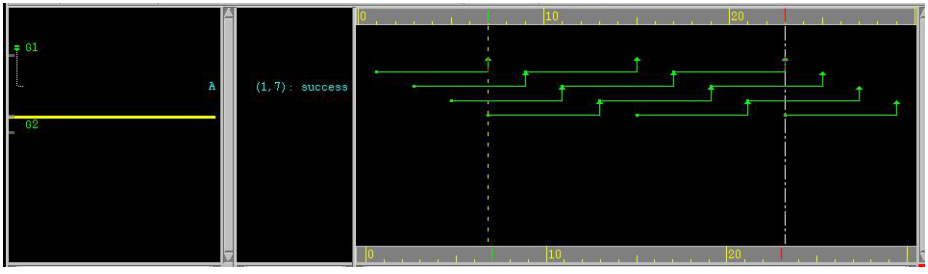


Figure: Expanded Property Signal A

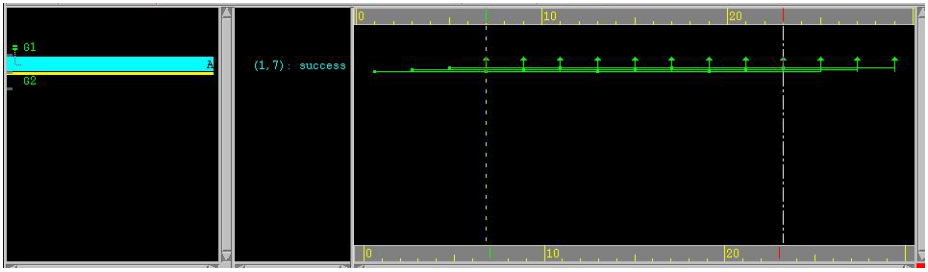


Figure: Un-expanded (Shrunk) Property Signal A

Classic Transaction

Expand/Shrink Overlapping

Menu Bar: **Waveform -> Classic Transaction -> Expand/Shrink Overlapping**

After selecting one or more transaction signals and invoking this option, the un-expanded signals are expanded automatically into *n* layers where *n* is the maximum number of attempted overlapped (begin, end) time-spans. The next time this option is invoked, the selected expanded signals are shrunk to one layer. The attribute information of the selected transaction is also displayed in the *nWave* value pane.

The maximum expanded layers of overlapped transactions can be changed by using the **Maximum Expanded Layer of Transactions** option on the **Display Signal** page under the **Waveform -> Default Value** folder of the *Preferences* form (invoked with **Tools -> Preferences**).

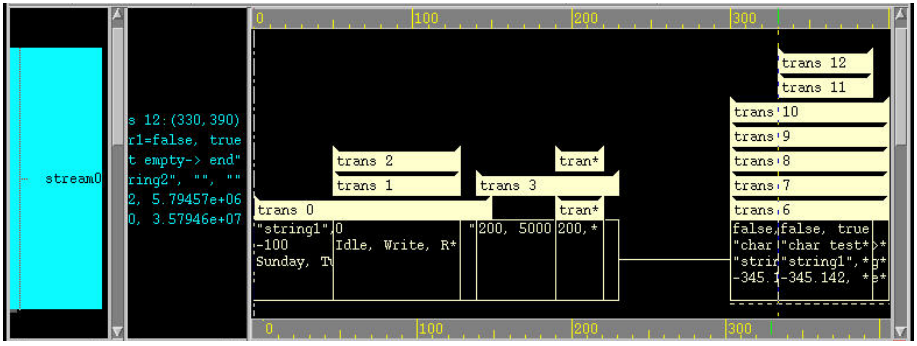


Figure: Expanded Transaction Signal

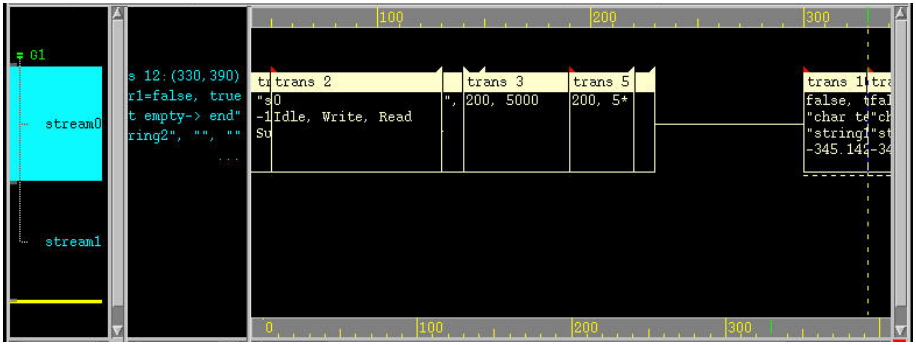


Figure: Un-expanded (Shrunk) Transaction Signal

Display/Hide Attributes

Menu Bar:	**Waveform -> Classic Transaction -> Display/Hide Attributes**

NOTE: This option is available when an FSDB file with transaction data is loaded.

This option displays the attributes of the selected signals in the waveform pane after one or more transaction signals have been selected. The next time this option is invoked, the attributes of selected signals are hidden in the waveform pane.

The **Hide Attributes** option on the **Transaction/Message** page under the **Waveform -> View Options -> Waveform Pane** folder of the *Preferences* form (invoked with **Tools -> Preferences**) can also be used to hide or display attributes of selected transaction signals in the waveform pane.

Create Attribute Signals

Menu Bar:	**Waveform -> Classic Transaction -> Create Attribute Signals**
Bind Key:	Ctrl+X

NOTE: This option is available when an FSDB file with transaction data is loaded.

This option opens the *Create Attribute Signals* form for the selected transaction stream. The *Create Attribute Signals* form displays the name and type of all transaction attributes.

In the attribute list, check (enable) or uncheck (disable) the check boxes to select or deselect the attributes. Check (enable) or uncheck (disable) the check box on the column header to select or deselect all attributes.

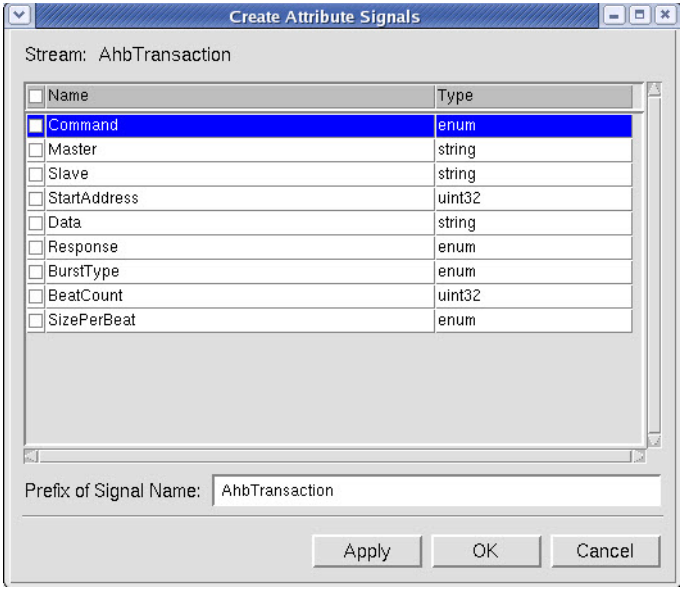


Figure: Create Attribute Signals Form

The *Create Attribute Signals* form includes the following fields:

- **Stream:** Displays the selected message stream name.
- **Prefix of Signal Name:** Specify the prefix used to create signals for selected attributes in the text field. The default prefix is the stream name. The format of the created signal name is *(prefix): attribute_name*.

When the **Apply** button is clicked, the selected attributes are expanded and newly created signals for the selected attributes are inserted below the selected transaction stream in the *nWave* window. The sample points (green triangles) shown in the waveform pane indicate the time spots where the attribute has a value. The sample points can be hidden by using the **Display Sample Point for Attribute Signals** option on the **Transaction/Message** page under the **Waveform -> View Options -> Waveform Pane** folder of the *Preferences* form (invoked with **Tools -> Preferences**).

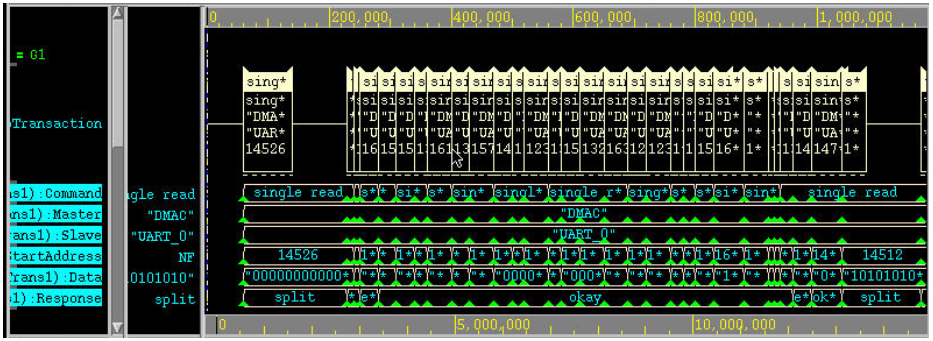


Figure: Expand Transaction Streams in nWave

Filter/Colorize

Menu Bar: **Waveform -> Classic Transaction -> Filter/Colorize**

NOTE: This option is available when an FSDB file with transaction data is loaded.

This option opens the *Filter/Colorize* form from the *Transaction/Message Analyzer* window. The filtered/colorized results are synchronized in the *Transaction/Message Analyzer* and *nWave* frames. Refer to the **Filter/Colorize** option description in the *Transaction/Message Analyzer* chapter for details.

Clear Filtering Results

Menu Bar: **Waveform -> Classic Transaction -> Clear Filtering Results**

NOTE: This option is available when an FSDB file with transaction data is loaded.

This option clears the filtered results for selected signals.

Clear Colorization Results

Menu Bar: **Waveform -> Classic Transaction -> Clear Colorization Results**

NOTE: This option is available when an FSDB file with transaction data is loaded.

This option clears the colorization results for selected signals.

Keep Marker at Transaction End Time

Menu Bar:	**Waveform -> Classic Transaction -> Keep Marker at Transaction End Time**

When this toggle option is turned on, the cursor is set to the start time of the selected transaction and the marker is set to the end time of the transaction.

Classic Message

Expand/Shrink Overlapping

Menu Bar:	**Waveform -> Classic Message -> Expand/Shrink Overlapping**
Bind Key:	Ctrl+O

After selecting one or more signals with messages and invoking this option, the un-expanded signals are expanded automatically into *n* layers where *n* is the maximum number of attempted overlapped (begin, end) time-spans. The next time this option is invoked, the selected expanded signals are shrunk to one layer. The attribute information of the messages is also displayed in the *nWave* value pane.

Display/Hide Attributes

Menu Bar:	**Waveform -> Classic Message -> Display/Hide Attributes**

NOTE: This option is available when an FSDB file with messages is loaded.

After selecting one or more signals with messages and invoking this option, the attributes of selected signals are displayed in the waveform pane. The next time this option is invoked, the attributes of selected signals with messages are hidden in the waveform pane.

The **Hide Attributes** option on the **Transaction/Message** page under the **Waveform -> View Options -> Waveform Pane** folder of the *Preferences* form (invoked with **Tools -> Preferences**) can also be used to hide or display attributes of selected signals in the waveform pane.

Create Attribute Signals

Menu Bar:	**Waveform -> Classic Message -> Create Attribute Signals**
Bind Key:	Ctrl+E

NOTE: This option is available when an FSDB file with messages is loaded.

This option opens the *Create Attribute Signals* form for the selected message stream. The *Create Attribute Signals* form displays the name and type of all message attributes.

In the attribute list, check (enable) or uncheck (disable) the check boxes to select or deselect the attributes. Check (enable) or uncheck (disable) the check box on the column header to select or deselect all attributes.

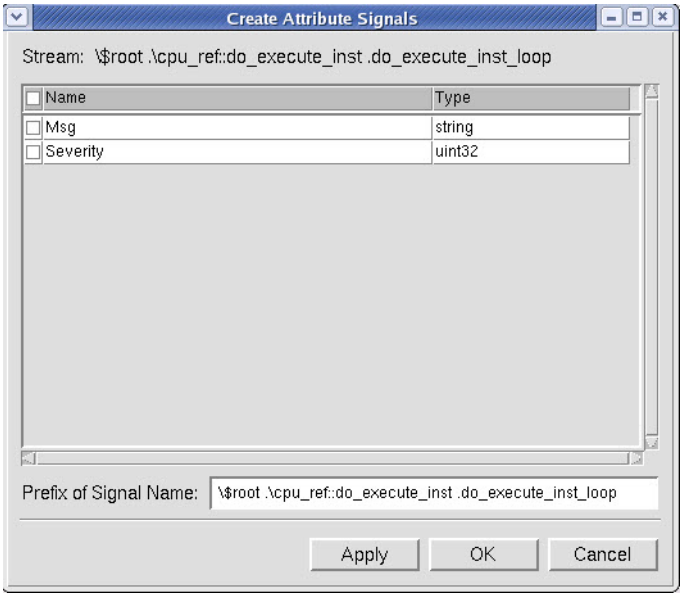


Figure: Create Attribute Signals Form

The *Create Attribute Signals* form includes the following fields:

- **Stream:** Displays the selected message stream name.
- **Prefix of Signal Name:** Specify the prefix used to create signals for selected attributes in the text field. The default prefix is the stream name. The format of the created signal name is *(prefix): attribute_name*.

When the **Apply** button is clicked, the selected attributes are expanded and newly created signals for the selected attributes are inserted below the specified

message stream in the *nWave* window. The sample points (green triangles) shown in the waveform pane indicate the time spots where the attribute has a value. The sample points can be hidden by using the **Display Sample Point for Attribute Signals** option on the **Transaction/Message** page under the **Waveform -> View Options -> Waveform Pane** folder of the *Preferences* form (invoked with **Tools -> Preferences**).

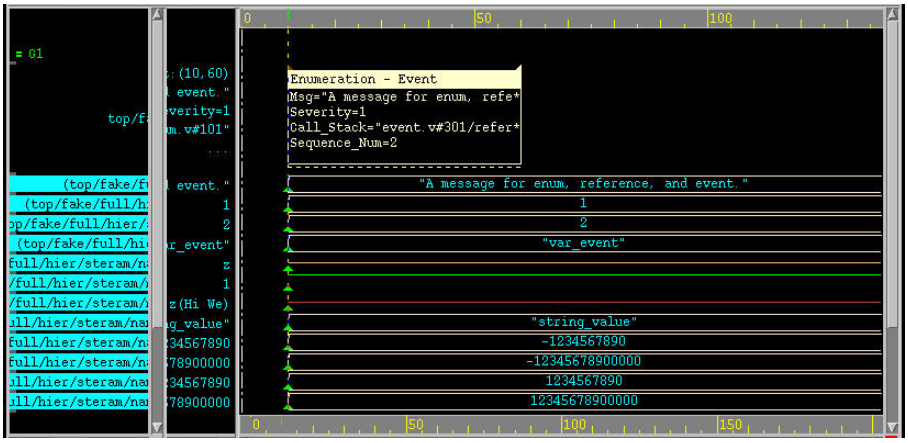


Figure: Expand Message Streams in nWave

Filter/Colorize

Menu Bar: **Waveform -> Classic Message -> Filter/Colorize**

NOTE: This option is available when an FSDB file with messages is loaded.

This option opens the *Filter/Colorize* form in the *Transaction/Message Analyzer* window. The filtered/colorized results in the *Transaction/Message Analyzer* and *nWave* window are synchronized. Refer to the **Filter/Colorize** option description in the *Transaction/Message Analyzer* chapter for details.

Clear Filtering Results

Menu Bar: **Waveform -> Classic Message -> Clear Filtering Results**

NOTE: This option is available when an FSDB file with messages is loaded.

This option clears the filtered results for selected signals.

Clear Colorization Results

Menu Bar:	**Waveform -> Classic Message -> Clear Colorization Results**

NOTE: This option is available when an FSDB file with messages is loaded.

This option clears the colorization results for selected signals.

Go To

Search Forward

Menu Bar:	**Waveform -> Go To -> Search Forward**
Bind Key	N
Toolbar Icon	

This option is used to search forward in time and locate the value in the waveform window.

Search Backward

Menu Bar:	**Waveform -> Go To -> Search Backward**
Bind Key	Shift+N
Toolbar Icon	

This option is used to search backward in time and locate the value in the waveform window.

Begin

Menu Bar:	**Waveform -> Go To -> Begin**
Bind Key	B

This option is used to set the cursor position to the beginning of the simulation.

End

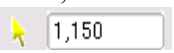
Menu Bar: **Waveform -> Go To -> End**

Bind Key E

This option is used to set the cursor position to the end of the simulation.

Time

Menu Bar: **Waveform -> Go To -> Time**

This option is used to set the precise cursor position in the waveform pane by specifying the time value in the *Search Time* form. The time specified here appears in the cursor time field on the toolbar. For instance, if the search time "600" is entered as shown below, the time appears as 

.

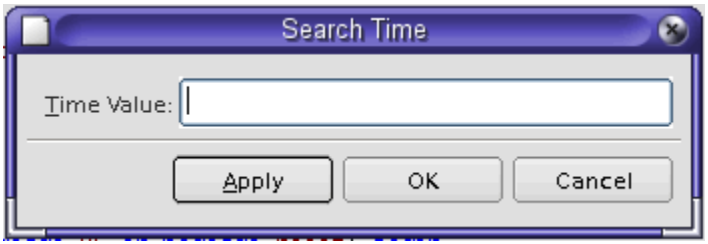


Figure: Search Time Form

Set Search Value

Menu Bar: **Waveform -> Set Search Value**

This option opens the *Search Value* form where the signal value to search for can be specified. The **Search By** icon on the toolbar automatically switches to **Bus Values** 

after a **Search Value** is defined.

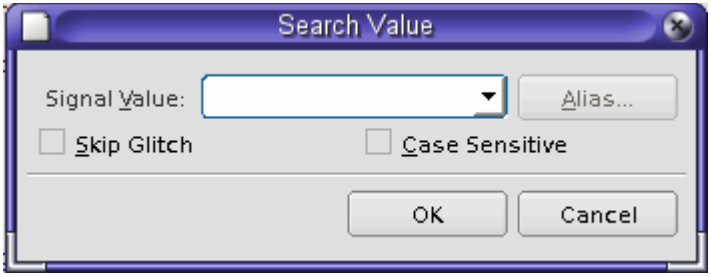


Figure: Search Value Form

The *Search Value* form includes the following field, button, and options:

- **Signal Value:** Before invoking the **Set Search Value** option, select the desired signals first, then specify the value, transition value (such as 0->55), or transition alias in the resulting *Search Value* form. The wildcard characters (* or ?) are supported. For example, using a string such as *ffa** finds the values of *ffa0*, *ffa1*, and *ffa2*.
- **Alias:** Enter the transition alias in the **Signal Value** text field directly or click the **Alias** button to obtain the desired aliases. When the **Alias** button is clicked, an *Alias Table* form appears for alias definition.

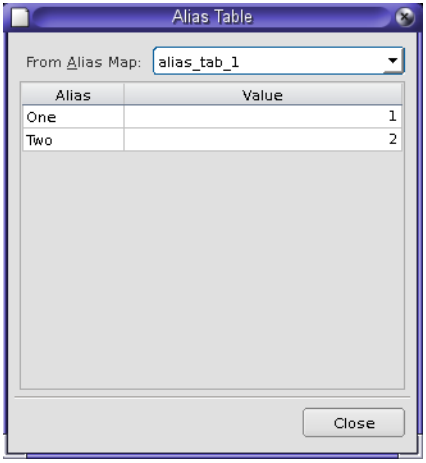


Figure: Alias Table

When an alias row is highlighted, that alias value is loaded automatically into the *Search Value Signal Value* form. The value for the selected signal is obtained from the current cursor position in the waveform pane. After **OK** is clicked in the *Search Value* form, a new text field is added to the *nWave* toolbar where a new bus value can be directly entered. If the search is

changed to another type, the text field disappears. When the search type is switched back to search bus values, the text field reappears.

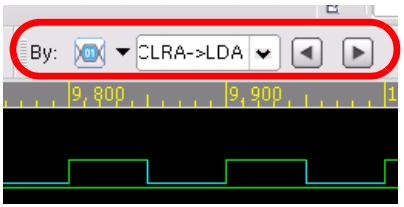


Figure: Search for Value Text Field is Added to the Toolbar

- **Case Sensitive**: When this option is turned on, the case in the signal value is matched (such as `fe = fe` and `FE = FE`). When this option is turned off, the case in the signal value is not matched (such as `fe = Fe = fE = FE`).
- **Skip Glitch:** When this option is turned on, glitches on signals are skipped (only the first/last transition is counted). When this option is turned off, glitches on signals are searched. The default is *off*.

The **Search Forward** and **Search Backward** buttons on the toolbar can be used to locate the value found in the waveform pane.

Set Search Attributes

Menu Bar: **Waveform -> Set Search Attributes**

NOTE: This option is available when an FSDB file with transaction/message data is loaded.

This option opens the *Set Search Attributes* form which is the same as the *Search* form invoked through the **View -> Search** option in the *Transaction Analysis* pane. Refer to the **Search** option description in the *Transaction/Message Analyzer* chapter for details.

Set Search Constraint

Menu Bar: **Waveform -> Set Search Constraint**

This option opens the *Set Search Constraint* form where the time interval or the n^{th} occurrence can be specified to constrain the search. This form is opened by invoking the **Waveform -> Set Search Constraint** option or by using the

Invoke Search Constraint Window icon  associated with the **Search By** option on the toolbar.

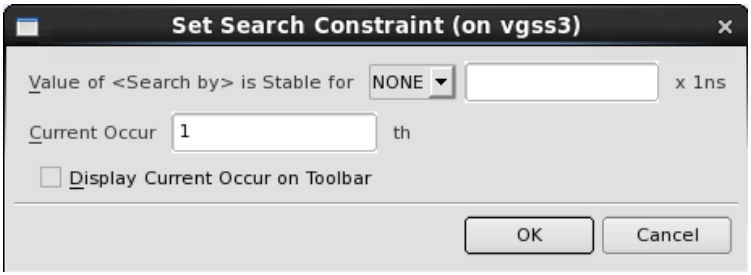


Figure: Set Search Constraint Form

The following figure depicts the icons associated with the **Search By** option without (None) and with (Constraint) the search constraint being set.

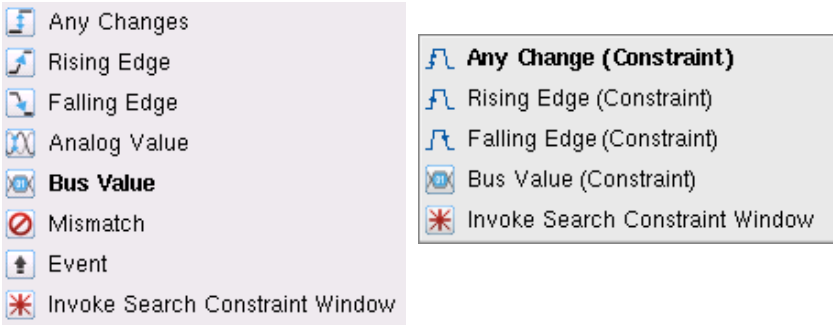


Figure: Set Search Constraint

In the *Set Search Constraint* form, includes the following fields and buttons:

- **Value of <Search By> is stable for:** This option locates the stable data or glitch data at the specified value. For example, select ">=" option from the drop-down list and type "*10*" in the x**1ns** text field. The constraint search can easily locate the stable data or glitch data at **10**ns.
- **Current Occur:** This option moves the constraint search to the stable data or glitch data at the specified occurrence at the selected signal. For example, if "100" is typed in the **Current Occur** text field, the constraint search moves to the stable data or glitch data at the 100th occurrence on the selected signal.
- **Display Current Occur at Toolbar:** When this option is selected, it displays the **Current Occur** valued specified in the form in the toolbar, as illustrated in following figure. Also, if the value is changed in the toolbar, the current occurred value is changed in the form accordingly.

- **Constraint Search By** icons: The icons include **Any Changes, Rising Edge, Falling Edge,** or **Bus Value**. Select one of the **Search By** icons and click the left or right arrow on the toolbar to move to the next matching transition for the selected signal in the backward or forward direction. To disable the constraint search, select **NONE** in the *Set Search Constraint* form.

Snap Cursor/Marker to Transitions

Menu Bar:	**Waveform -> Snap Cursor/Marker to Transitions**
Bind Key	S

When this toggle option is turned on and the cursor and/or marker are placed in the waveform pane, the cursor and/or marker automatically snaps to the nearest value change position. Otherwise, the cursor and/or marker moves to the nearest value change on the signal selected in the waveform pane. This option only affects the current window.

To snap a user-defined marker (set by the **Waveform -> Marker** option) to a transition, drag the user-defined marker and release it on a signal to snap. The user-defined marker snaps to the closest transition of the signal.

NOTE: When you click the **Zoom Scale Ruler** (at the top of the waveform pane) or **Full Scale Ruler** (at the bottom of the waveform pane), it does not affect the snap option.

Fix Cursor/Marker Delta Time

Menu Bar:	**Waveform -> Fix Cursor/Marker Delta Time**
Bind Key	X

When this toggle option is turned on, the delta time value is fixed. The delta value is the value associated with the marker minus the value associated with the cursor. That is, **Delta = Marker - Cursor**. Thus, when the cursor is moved in the waveform pane, the marker also moves to maintain the fixed delta value (the same behavior occurs when the marker is moved).

Keep Cursor at Center

Menu Bar:	**Waveform -> Keep Cursor at Center**
Bind Key	Y

When this toggle option is turned on, the cursor remains in the center of the *nWave* window when the **Search Backward** or **Search Forward** toolbar icons are clicked to search in time.

NOTE: When this toggle option is turned on, if the cursor is not seen in the waveform pane, the waveform is redrawn to keep the cursor in the center of the *nWave* window when the **Zoom In** and **Zoom Out** toolbar icons are clicked.

Waveform Time

Shift File Time

Menu Bar:	**Waveform -> Waveform Time -> Shift File Time**

This option adds an offset time to the active waveform in the *Shift Waveform* form. Use this option when two sets of simulation results have different start times and they need to be visually compared or verified.

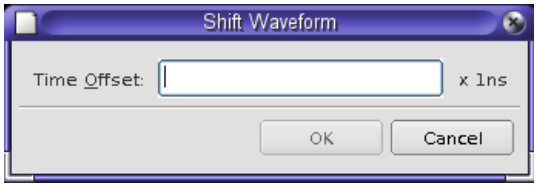


Figure: Shift Waveform Form

Shift Individual Signal Time

Menu Bar:	**Waveform -> Waveform Time -> Shift Individual Signal Time**

This option shifts the selected signals by the offset time specified in the *Shift Signal* form.

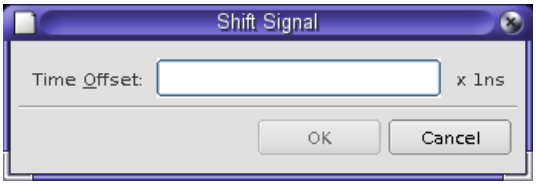


Figure: Shift Signal Form

Set File Time Scale

Menu Bar: **Waveform -> Waveform Time -> Set File Time Scale**

This option changes the time scale of the active waveform file. Enter the time scale in the **Time Scale** text field of the *Set File Time Scale* form and then click **OK**. This capability is useful when comparing two simulation runs with different simulation frequencies. After invoking this option, *nWave* ignores the original file's time scale and replaces the old value with the new one.

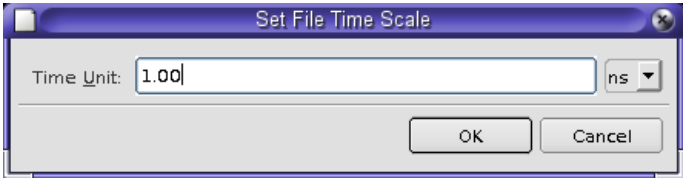


Figure: Set File Time Scale Form

The default is the time scale defined in the activate file. The time unit can be specified using the following options: *ps*, *ns*, *us*, *ms* and *s*. The definition of each of the units is listed below.

fs	fs = 10 ^ -15	femtosecond
ps = 1000 fs	ps = 10 ^ -12	picosecond
ns = 1000 ps	ns = 10 ^ -9	nanosecond
us = 1000 ns	us = 10 ^ -6	microsecond
ms = 1000 us	ms = 10 ^ -3	millisecond
s (sec)= 1000 ms	s = 1	second

Set Window Time Unit

Menu Bar: **Waveform -> Waveform Time -> Set Window Time Unit**

Toolbar Icon: 

This option uses the *Set Window Time Unit* form to specify the time unit in the waveform window as indicated on the toolbar. If the window time unit value is changed, the values shown in the **Full Scale Ruler**, **Zoom Scale Ruler**, **Cursor**, **Marker**, and **Delta** changes accordingly.

The difference between **Set Window Time Unit** and **Set File Time Scale** (**Waveform -> Set File Time Scale**) is that **Set Window Time Unit** does not change the time scale in the simulation results file, while the **Set File Time Scale** option replaces the time scale in the simulation results file.

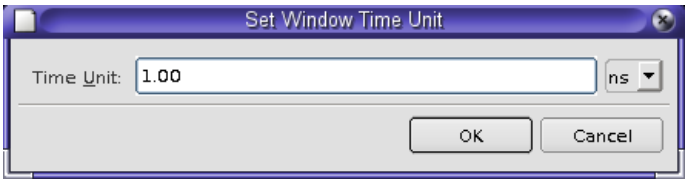


Figure: Set Window Time Unit Form

NOTE: The window time unit value always auto synchronizes between the primary *nWave, nTrace, and nSchema* windows when the annotation is turned *on*.

Marker

Create Marker

Menu Bar: **Waveform -> Marker -> Create Marker**

This option automatically creates new markers in the *nWave* window at the location of the cursor. By default, the marker is named as *M[n]* (where, *n* is the serial number), for example *M1*, *M2*, and so on as illustrated in the following figure:

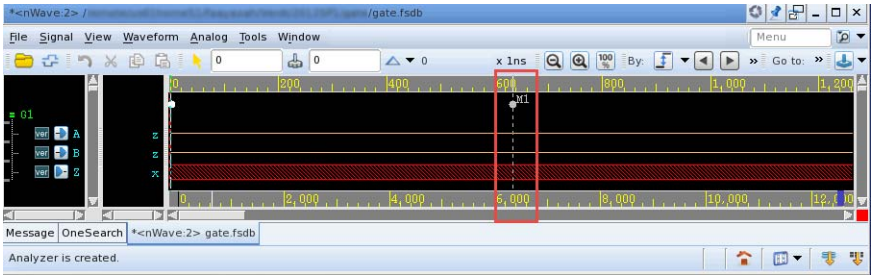


Figure: Create Marker

The default user-defined marker is a white dash line. You can specify width, style, and color for different line types by using the **Preference->Waveform->Color/Font/Pattern->Pattern** page.

Marker

Menu Bar:	**Waveform -> Marker -> Marker**
Bind Key:	Shift+M

This option opens the *Marker* form where markers can be set through the current cursor location, current marker location, or a specified time. This form can also be used to set or modify the time, color, or line style of the marker and cursor.

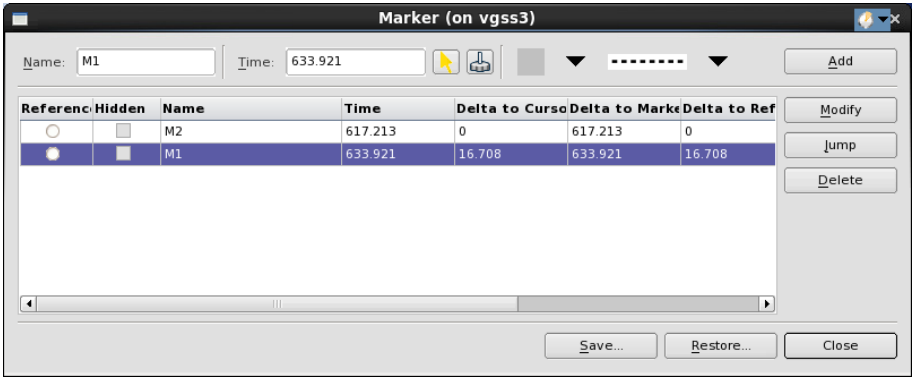


Figure: Marker Form

The *Marker* form includes the following options and fields:

- **Reference**: Select the user-defined marker as the reference marker.

- **Hidden**: Select the user-defined marker that must be hidden in the *Waveform* pane.
- **Name:** Enter a name in the text field for the marker name.
- **Time:** The time can be defined in three ways:
  - Enter a time in the **Time** text field and the specified time is added as a marker.
  - Click the cursor button and the time for the cursor location in the *Waveform* pane is added as a marker.
  - Click the marker button and the time for the marker location in the *Waveform* pane is added as a marker.
- **Delta to Cursor:** The relative time between user-defined marker and primary cursor.
- **Delta to Marker:** The relative time between user-defined marker and primary marker.
- **Delta to Ref:** The relative time between user-defined marker and the reference marker.
-  : Specify the color of the selected marker.
- : Specify the line style of the selected marker.
- **Add:** After the **Name** and the **Time** fields for the marker are defined, click this button to add the marker to the marker list. The related information including **Name, Time, Delta to Cursor, Delta to Marker, Color,** and **Line Style** of this marker is also added to the user marker list in the *Marker* form.
- **Modify:** Click this button to update the time, color, or line style of a selected marker in the user marker list after editing. This button is disabled when multiple markers are selected.
- **Jump:** Click this button to jump to the predefined marker time after a marker is selected in the marker list. This button is disabled when multiple markers are selected.
- **Delete:** Click this button to remove one or more selected markers from the user marker list. You can remove multiple markers by holding the **Shift** key (selects a range) or the **Ctrl** key (selects non-contiguous items) and by left-clicking.
- **Save:** Click this button to open the *Save Marker Report* form where the directory structure can be viewed and a file name specified to save the marker information to. The saved marker file (`*.rpt`) can be loaded by

restoring the *signal.rc* configuration file through the **Restore** button or the **File** -> **Restore Signal** option in *nWave*.

NOTE: The default file extension (`*.rpt`) is automatically added to the saved file if the extension is not specified.

- **Restore:** Click this button to open the *Restore Marker Report* form where the previously saved marker file (`*.rpt` extension) can be restored.

To customize the columns to be displayed on the *Marker* form, click the right-click menu option on the table header, as illustrated in the following figure:

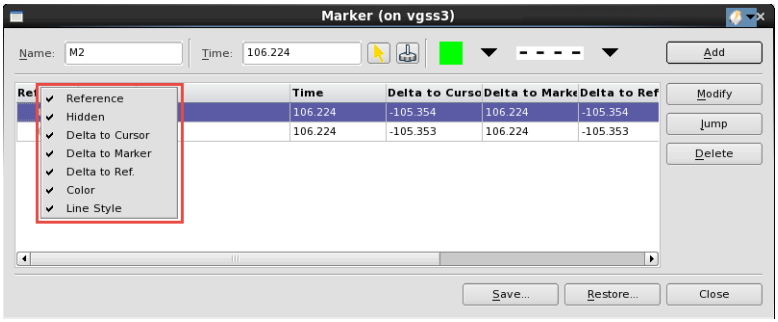


Figure: Right-Click Options

NOTE: The **Name** and **Time** columns cannot be removed from the *Marker* form.

After a marker is set successfully, double-click a marker name on the marker area to open a simplified marker form where you can change the specified marker name, time, color, and line style.

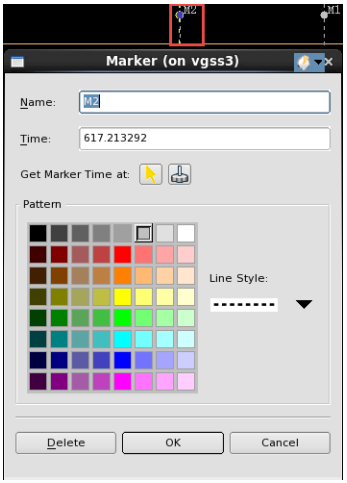


Figure: Simplified Marker Form

Goto Marker

This option allows you to select and go to the desired user-defined marker from the user marker list in the pull-down menu. The cursor moves to the time defined by the selected marker.

You can also select the **Goto Marker** toolbar icon to select and go to the desired user-defined marker.

Delete Marker

This option allows you to select and delete the desired user-defined marker from the user marker list in the pull-down menu. You can also select all the user-defined markers by selecting **All**.

Move Cursor/Marker

To move the user-defined cursor/markers, select the marker, cursor, or the user marker list from the pull-down menu, the marker/cursor gets selected and is displayed in white dash line along with the label, as illustrated in the following figure. Move the mouse to the location where you want the marker/cursor and press left mouse button to place the marker/cursor.

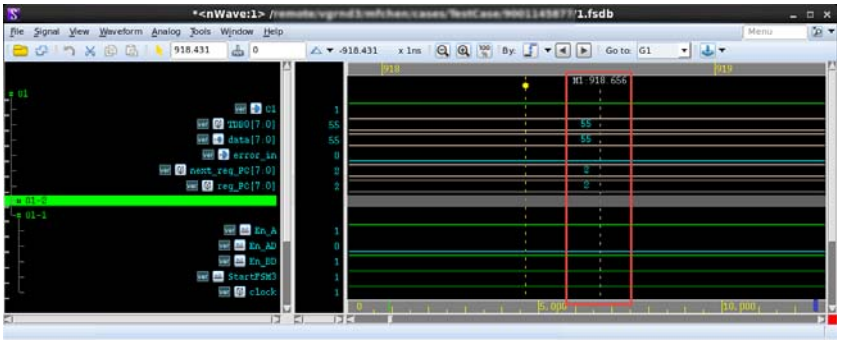


Figure: Example of Move Marker

You can also move the primary cursor and the primary marker.

Set Reference Marker

With this option you can select primary cursor, primary marker, user marker, or sim (the simulation end time, only for interactive mode) as the reference marker.

The default value of the **Set Reference Marker** option is **Cursor**.

When you click the **Waveform -> Show Cursor/Marker Values -> Relative** option, then in the *nWave* window **REF** is used to mark the reference marker, instead of relative time or frequency, as illustrated in the following figure:

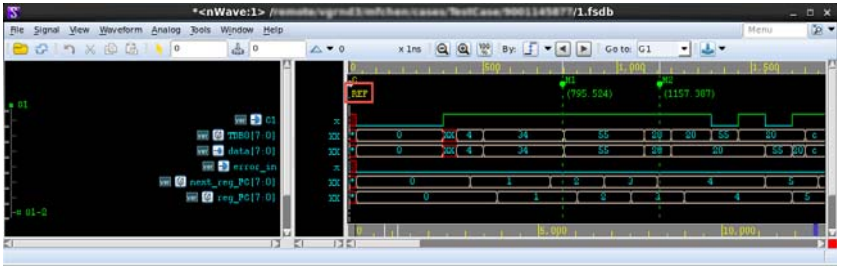


Figure: Example of REF Marker

Points to Note:

- You can select the simulation end time as the reference marker only in the interactive mode.
- If you select **Relative** sub-menu option as the **Show Cursor/Marker Values** menu option, *nWave* provides REF to mark the reference marker instead of the relative time or the frequency.

Show Cursor/Marker Values

This option provides information about the user marker (time, frequency, and so on) in the *Waveform* pane.

Absolute

Menu Bar: **Waveform -> Marker -> Show Marker Values -> Absolute**

This option displays the time of the marker in the *nWave* window next to the marker name.

Adjacent

Menu Bar: **Waveform -> Marker ->Show Marker Values -> Adjacent**

This option displays the difference between two adjacent markers (includes primary cursor, primary marker, and end time of signals in the interactive mode) in the *nWave* window in the middle of the marker area.

Relative

Menu Bar: **Waveform -> Marker -> Show Marker Values -> Relative**

This option displays the difference between user marker and reference marker in the *nWave* window below the marker name.

Difference as Frequency

Menu Bar: **Waveform -> Marker -> Show Marker Values -> Frequency**

This option displays the difference between user marker and reference marker as the frequency value. If it is not enabled, then the difference appears as the time value.

NOTE: When you click the **Absolute** option, the height of the marker area does not change.

When you click the **Relative** option or **Adjacent** option, the marker area height increases to twice the size than before.

When you click the **Absolute** option, Relative option, or Adjacent option, *nWave* displays the name for the primary cursor and the marker

in the *nWave* window. If the primary cursor name is C and the primary marker name is M, then you cannot use C or M as the user-defined marker names.

Stick Cursor/Marker on Waveform

When this toggle option is turned *on* the cursor/marker can only be dragged in the marker area. When this option is turned *off*, the cursor/marker can be dragged in both the waveform area and the marker area.

The default setting for the **Stick Cursor/Marker on Waveform** option can be set using the **Stick Cursor/Marker on Waveform** preference option in the **Waveform -> View Options -> Waveform Pane - > General** page of the *Preferences* form.

If the **Waveform->View Options->Waveform Pane->General->Stick Cursor/Marker on Waveform** preference option is *on*, then the default value of the **Stick Cursor/Marker on Waveform** option is *on*.

If the **Waveform->View Options->Waveform Pane->General->Stick Cursor/Marker on Waveform** preference option is *off*, then the default value of the **Stick Cursor/Marker on Waveform** option is *off*.

Marker Right-Click Options

When you right-click in the marker area of the *nWave* window, *nWave* displays the marker options that can be performed as illustrated in the following figure:

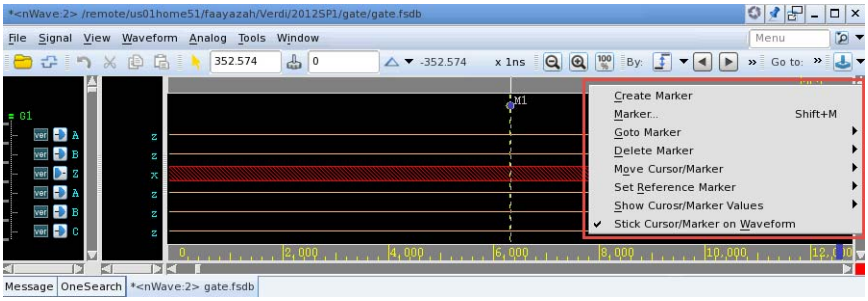


Figure: Marker Right-Click Options

For more information regarding the marker options, see the *Marker* section.

Error

nWave provides capability to show error indicators at which simulation error occurs as illustrated in the following figure:

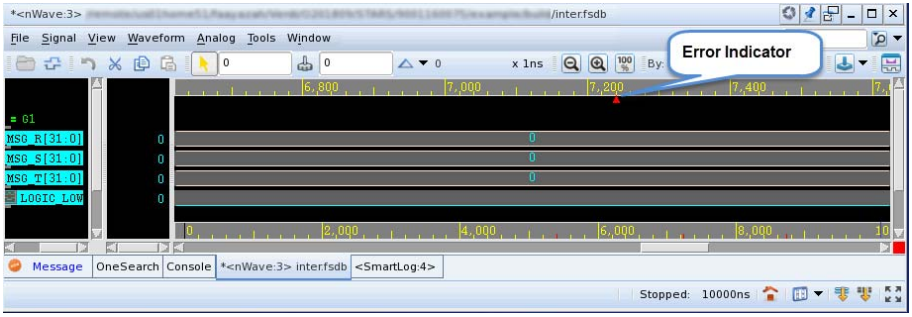


Figure: Error Indicator

When you double-click on an error indicator it invokes a SmartLog as illustrated in the following figure:

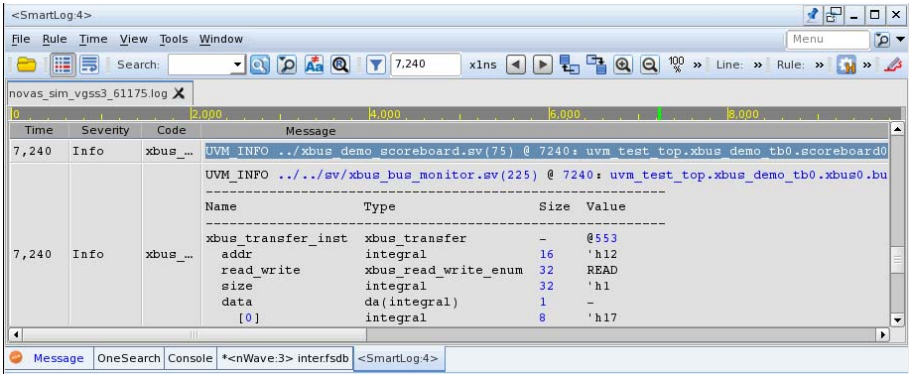


Figure: SmartLog Invoked for the Error Indicator

Search Previous

Menu Bar:	**Waveform -> Error -> Search Previous**
Bind Key	Ctrl+Shift+<

Finds the previous error indicator from the selected indicator. If there is no error indicator selected, then it points to the last indicator.

Search Next

Menu Bar:	**Waveform -> Error -> Search Next**
Bind Key	Ctrl+Shift+>

Finds the next error indicator from the selected indicator. If there is no error indicator selected, then it points to the first indicator.

Save as Markers

Menu Bar:	**Waveform -> Error -> Save as Markers**
Bind Key	Ctrl+Shift+M

Invokes the *Marker* form, and automatically creates new Markers with the marker name as (E[n]), time, color (red) and line style (solid line). You can modify the details as mentioned in the *Marker* section as illustrated in the following figure. If there is no selected error indicator, then this command is disabled.

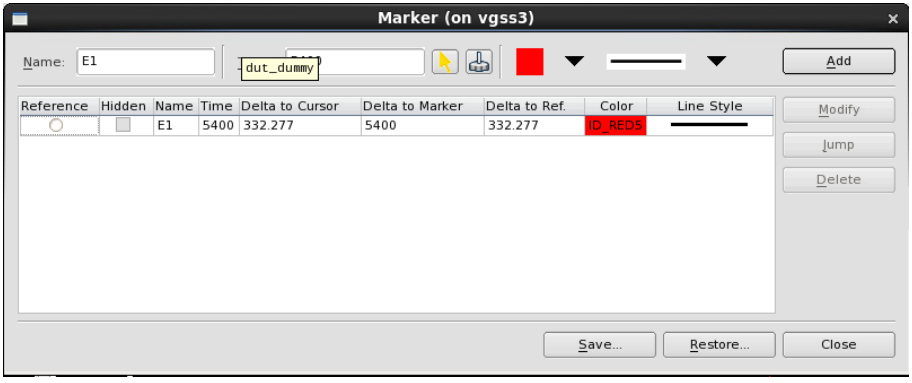


Figure: Save as Markers

Tooltip Information

When you hover the mouse on the error indicator, then the tooltip information is displayed as illustrated in the following figure:

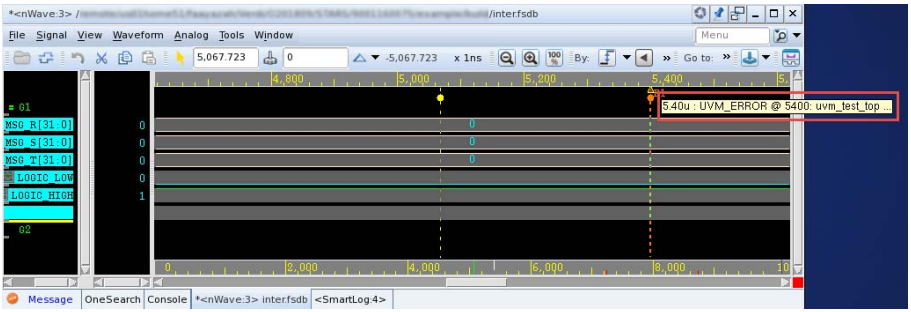


Figure: Tooltip Information

Analog Menu Options

NOTE: This menu is enabled after an Analog dump file is opened.

Display Delta Y

Menu Bar: **Analog -> Display Delta Y**

Delta Y is the vertical distance between the marker and cursor. When this toggle option is turned on, *nWave* displays the Delta Y value in the value pane.

NOTE: The display sequence in the value pane is cursor/marker/Delta Y if the following options are turned on: **Value at Cursor/Marker (View -> Value at Cursor/Marker)** and **Display Delta Y (Analog -> Display Delta Y)**.

Zoom Value

Menu Bar: **Analog -> Zoom Value**

This option opens the *Zoom Value* form where vertical zooming constraints for an analog signal can be specified. The default values for the upper and lower bounds are the full range of the selected signal.

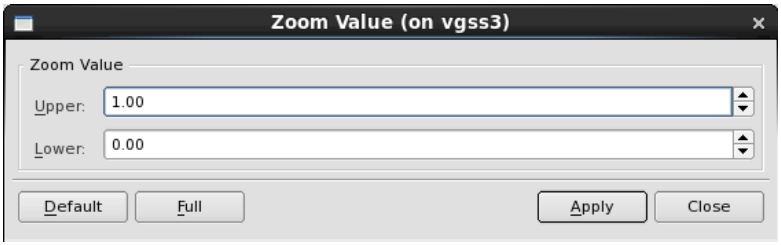


Figure: Zoom Value Form

The **Default** button resets the selected analog signals to their original waveforms. The **Full** button views the difference between two selected signals with respect to the signal height value. When values are specified in the **Upper** and **Lower** text fields, *nWave* zooms the selected signal to the values immediately. When the **Upper** and **Lower** bounds are set to the same value, positive one (+1) and negative one (-1) are automatically appended, respectively. If the lower bound is set higher than the upper bound, these two values are automatically swapped when the **Apply** button is clicked or the **Enter** key is pressed on the keyboard.

NOTE: Analog waveforms can be zoomed vertically by dragging-left vertically in the waveform pane.

Vertical Fit

Menu Bar: **Analog -> Vertical Fit**

Bind Key Shift+F

When this toggle option is turned on, the selected signals are vertically zoomed to their full range. You can achieve the same effect by clicking on the **Full** button in the *Analog Zoom* form (by invoking the **Analog -> Zoom Value**).

Auto 100% Vertical Fit

Menu Bar: **Analog -> Auto 100% Vertical Fit**

When this toggle option is turned on, the Y value view range is automatically fit to the full value range for the selected signals. When this option is turned on, click the **Zoom All** icon  on the toolbar to zoom the X and Y values to the full value range for all signals.

Ruler

Menu Bar: **Analog -> Ruler**

This option opens the *Analog Ruler* form where the analog ruler, grid, and reference line for the selected signals can be defined.

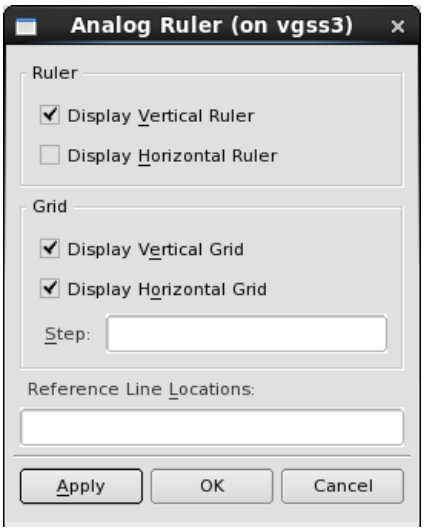


Figure: Analog Ruler Form

The *Analog Ruler* form includes the following options and fields:

- **Display Vertical Ruler:** When this option is turned on, *nWave* displays a vertical ruler (similar in function to the horizontal **Zoom Scale Ruler**, located at the top of the waveform pane) for the selected analog signals. When this option is turned off, *nWave* does not show a vertical ruler. If one exists, it is removed.
- **Display Horizontal Ruler:** When this option is turned on, *nWave* displays a horizontal ruler for the selected analog signals. When this option is turned off, *nWave* shows a horizontal ruler. If one exists, it is removed.
- **Display Vertical Grid:** When this option is turned on, *nWave* displays a vertical grid. When this option is turned off, *nWave* does not display a vertical grid. If one exists, it is removed.
- **Display Horizontal Grid:** When this option is turned on, *nWave* displays a horizontal grid. When this option is turned off, *nWave* does not display a horizontal grid. If one exists, it is removed. The grid distance can be

specified in the **Step** text field. When setting up the **Step** text field, the **Display Horizontal Ruler** option can also be enabled, so the step value can be adjusted to fit this ruler.

- **Reference Line Locations:** Specify one or more horizontal lines as reference lines within the analog range in the waveform pane.

Set Search Analog Value

Menu Bar: **Analog -> Set Search Analog Value**

This option sets the search value and error tolerance for an analog signal search. Select the analog signal in the signal pane and set the search value and error tolerance in the *Search Value* form.

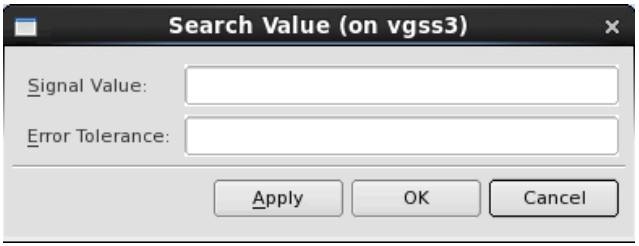


Figure: Search Value Form

After the values are set in the form, the **Search By** icon on the toolbar is changed to **Search By Analog Values** as shown below.

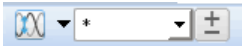


Figure: Search by Analog Values

A new text field is added to the *nWave* toolbar where a new analog value can be directly entered. If the search is changed to another type, the text field disappears. When the search type is switched back to search bus values, the text field appears again.

You can enter the number and scale value in the text field as illustrated in the following figure:

Figure: Number+Scale Text Field

Click the **Search Previous** or **Search Next** buttons on the toolbar to find the value. If the value is found, the cursor jumps to that time.

For example, assume that the **Signal Value** and **Error Tolerance** fields are set to 4.21 and 0.02, respectively, in the *Search Analog* form. Also, assume that the first fitted value is 4.23 at time 488 and the next is 4.22 at time 499. Click the **Search Next** button on the toolbar and the cursor finds the value of 4.22 at time 499 but not the value of 4.23 at time 488.

Wave Slew

Menu Bar: **Analog -> Wave Slew**

This option automatically measures the delta time for two specific values in an analog signal using the *Wave Slew* form. For example, assume the delta time between 10% from the cursor's right and 10% from the marker's left (or 90% of the full range) of an analog signal is desired:

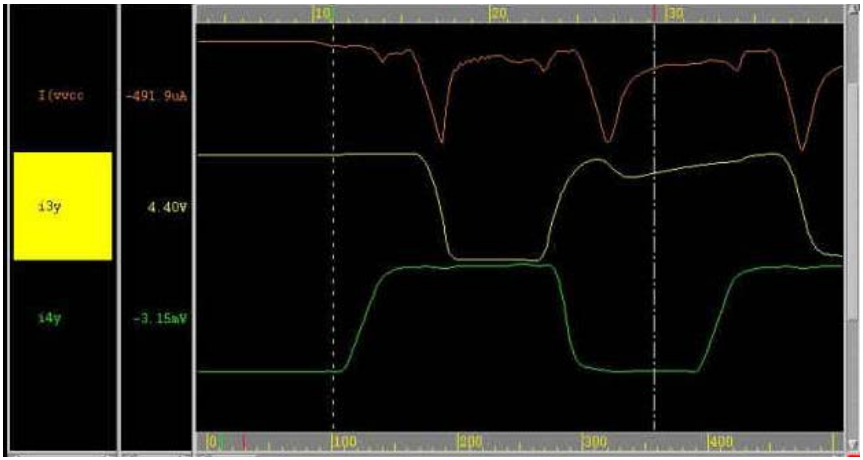


Figure: An Analog Signal

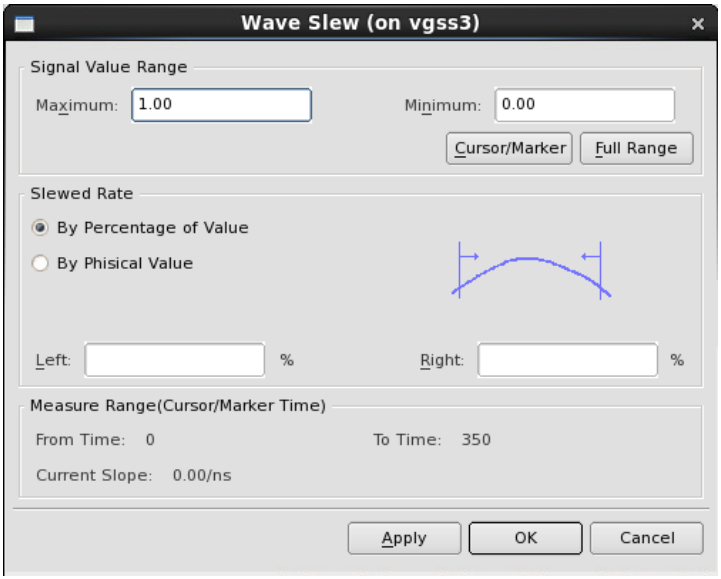


Figure: Wave Slew Form

The *Wave Slew* form includes the following options and fields:

- **Signal Value Range:** If the **Slewed Rate** is **By Percentage of Value**, the **Maximum** and **Minimum** values associated with this signal are displayed in the range of either between **Cursor/Marker** or in **Full Range** (by default); or the values can be specified. However, if the **Slewed Rate** is **By Physical Value**, only the signal value range is displayed in **Full Range** and values in the **Maximum** and **Minimum** text fields are not modified.

The **Slewed Rate** section includes the **By Percentage of Value** option and the **By Physical Value** option:

- **By Percentage of Value:** Select this option to input the left and right slew values by percentage of value.
- **By Physical Value**: Select this option to input the left and right slew values by physical value.
- **Left/Right:** If **By Percentage of Value** is chosen, numbers between 0 and 100 percent (**Left** is less than or equal to **Right**) must be entered. If **By Physical Value** is chosen, the physical numbers of the signal values (the **Minimum** signal value is less than or equal to **Left**; **Left** is less than or equal to **Right**; and **Right** is less than or equal to the **Maximum** signal value) must be entered. The units can be:

f: 1.0e-15
p: 1.0e-12
n: 1.0e-9
u: 1.0e-6
m: 1.0e-3
K: 1.0e3
M: 1.0e6
G: 1.0e9

Perform left slew only by inputing only the left value and leaving the right value empty, (and vice versa).

- **Measure Range (Cursor/Marker Time):** This section shows the cursor time (**From Time**) and marker time (**To Time**) as references, and also the **Current Slope**. **Current Slope** measures the slope of the analog signal between the values crossed by the **Cursor** and **Marker** lines.

After you click **Apply**, both cursor and marker snap to points matching the specifications. The delta time between the cursor and marker is displayed in the delta time field on the toolbar.

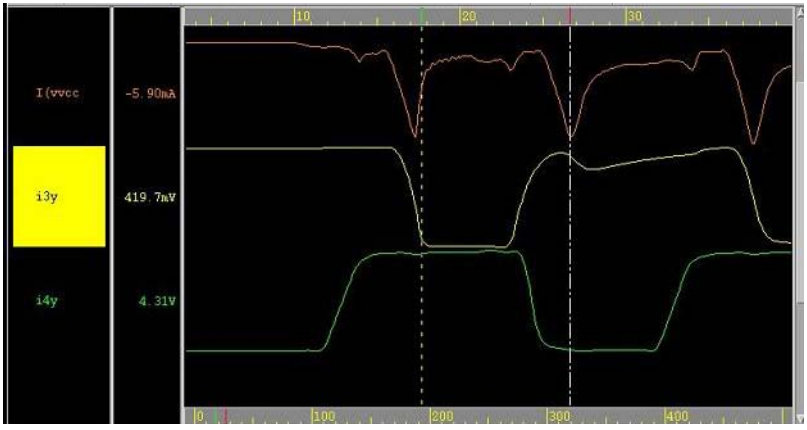


Figure: Analog Signal after Wave Slew

NOTE:

1. This option cannot calculate slew for overlapping signals.
2. This option cannot calculate slew for piecewise constant signals.
3. This option cannot calculate slew for multiple selected signals.
4. The left line and the right line may cross in some cases.
5. The signal value range is updated when the selected signal is changed.
6. The measure range is updated when changes occur on the cursor/marker times or the window time unit.

Drawing Style

These options specify the display method of the selected analog signal. Invoke either the **Piecewise Constant** or **Piecewise Linear** options to display an analog waveform using an interpolation technique.

Piecewise Constant/Piecewise Linear

Menu Bar:	**Analog -> Drawing Style -> Piecewise Constant/Piecewise Linear**

When a signal waveform that was drawn using a piecewise constant approach is selected and this option is invoked, the waveform display redraws the signal using a piecewise linear technique. Alternatively, if the signal was drawn using a piecewise linear technique, it is redrawn using a piecewise constant approach.

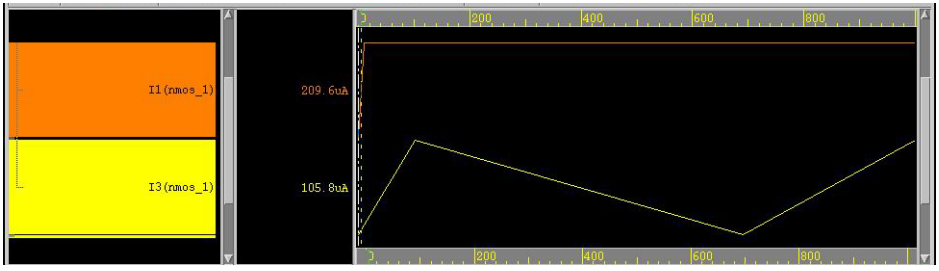


Figure: Drawing Style in Piecewise Linear

Point Style

Menu Bar:	**Analog -> Drawing Style -> Point Style**

This option redraws the selected analog signals in points.

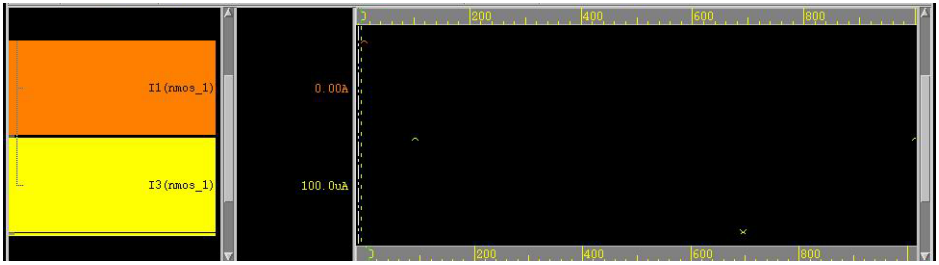


Figure: Drawing Style in Piecewise Point Style

NOTE: The **Drawing Style** options not only change the drawing style but also the value calculation for selected signals. For example, causes the signal drawing to be consistent with its value calculation.

Select Analogs

Menu Bar: **Analog -> Select Analogs**

This option selects all analog signals in the signal pane.

Format & Precision

Menu Bar: **Analog -> Format & Precision**

This option opens the *Format & Precision* form where the value displayed in the value pane can be defined.

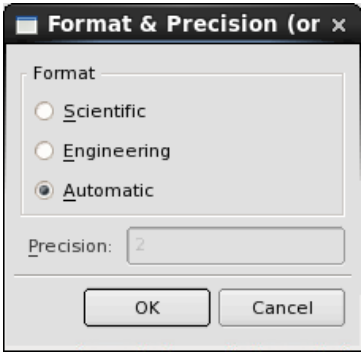


Figure: Format and Precision Form

The **Format** section defines the value format. This section includes three options. Only one option can be selected at a time.

- **Scientific:** When this option is turned on, *nWave* displays the value in **Scientific** format with the precision defined in the **Precision** text field. Scientific format uses a decimal notation in the form *[-]m.dddddde+/-xx*, where the number of *d*'s is specified by the precision. The default precision is *2*. For the decimal value of "0.0019884", the scientific representation is "1.98840e-03" with a default precision of 5.
- **Engineering:** When this option is turned on, *nWave* displays the value in **Engineering** format with the precision defined in the **Precision** text field. Engineering format uses a decimal notation in the form *[-]mmm.ddd*, where the number of d's is specified by the precision. The default precision is *2*. For the decimal value of "0.0019884", the Engineering representation is "0.00199" with a default precision of 5.
- **Automatic:** When this option is turned on, *nWave* determines which format is best suited for the signal, either **Scientific** or **Engineering**.

Average/Min/Max/RMS

Menu Bar: **Analog -> Average/Min/Max/RMS**

This option calculates the average, integration (a measure function of HSPICE with a formula of *integration = Average * (end_time - start_time)*), minimum, maximum, and RMS (root mean square) values of the selected signals for the range between the cursor and the marker. Select the signals first and set both the cursor and the marker in the waveform pane. This option opens the *Average/ Integration/RMS/Min/Max* form.

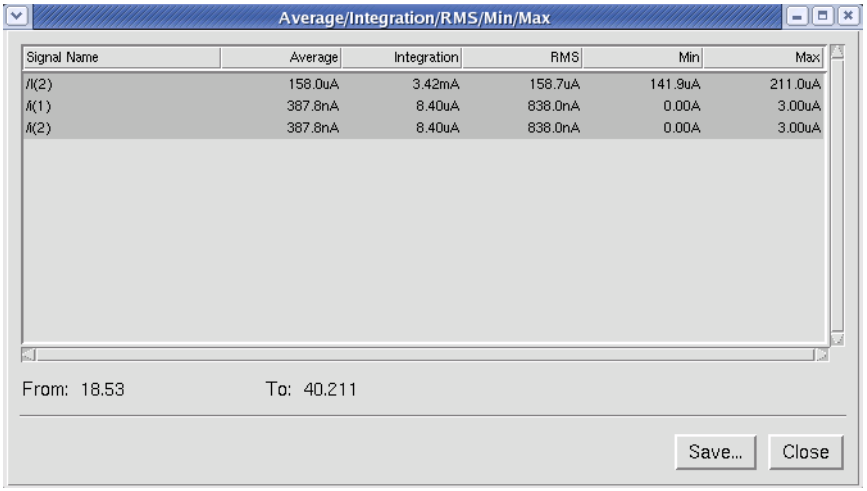


Figure: Average/Integration/RMS/Min/Max Form

In the *Average/Integration/RMS/Min/Max* form, the **From** text field displays the cursor value and the **To** text field displays the marker value. The selected signals and their average, integration, RMS, minimum, and maximum values are displayed in the table. When the *Average/Integration/RMS/Min/Max* form is open and the cursor and the marker are changed, the values in the form changes accordingly. You can save the results to a specified file by clicking the **Save** button.

Convert to Analog

Menu Bar: **Analog -> Convert to Analog**

This option converts a digital signal into an analog signal. When a digital waveform file is loaded into *nWave* and one (or more) of these signals is changed into an analog waveform, the vertical zoom-in and all analog options can be executed on the new analog signal. If an integer signal in VHDL is loaded, *nWave* treats it as a scale signal in 2's complement and creates a new analog signal with name *DtoA_{original_name}*.

Analog to Digital

Menu Bar: **Analog -> Analog to Digital**

This option opens the *Analogs to Digitals* form where one or more selected analog signals can be converted to a digital signal.

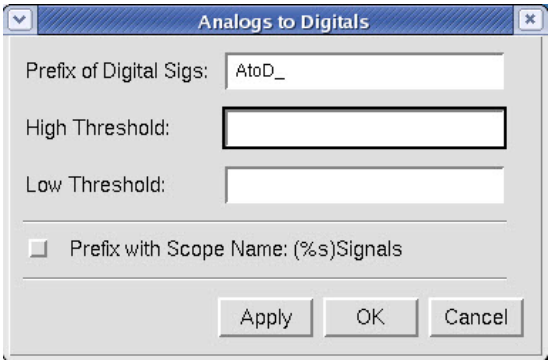


Figure: Analogs to Digitals Form

The *Analogs to Digitals* form includes the following fields and option:

- **Digital Signal Name (**for single analog signal) or **Prefix of Digital Sigs** (for multiple analog signals): Enter the name of the new digital signal here if only one signal is selected. If more than one signal is selected, the prefix for all the converted digital signals are set instead. The default prefix is *AtoD_*.
- **Low Threshold**: Specify the voltage of the low threshold.
- **High Threshold**: Specify the voltage of the high threshold.
- **Prefix with Scope Name: (%s)Sigs**: When this option is turned on, the scope name is added as a prefix for signals to avoid the same signal name occurring in the same scope. This option is turned on if the number of selected analog signals is greater than or equal to (>=) 2. The default is *on*.

nWave then creates a new signal with the following values:

0: if analog value < low threshold
1: if analog value > high threshold
X: otherwise

The value of the **High Threshold** must be greater than or equal to the value of the **Low Threshold** or a warning message appears. The same value for both **High Threshold** and **Low Threshold** can be defined to avoid an unknown signal.

Analog Expression

Menu Bar: **Analog -> Analog Expression**

This option creates a new analog signal after performing a logical operation on a signal.

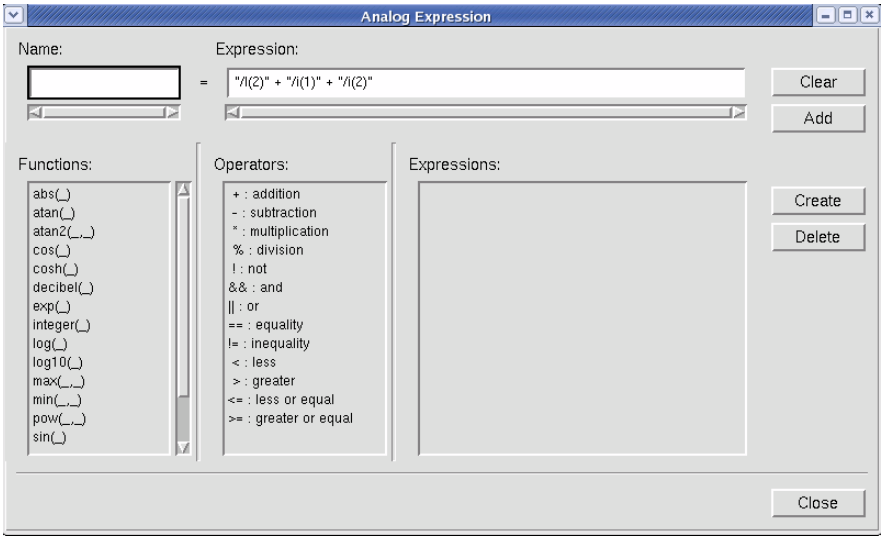


Figure: Analog Expression Form

Name: Enter a name for the analog signal in the **Name** text field.

Expression: The logical operation can be entered in several ways:

- Select the signal from the signal pane and invoke the **Analog Expression** option, then select an operator from the **Operators** list. The default operator is + (addition).
- Invoke the **Analog Expression** option without selecting a signal from the signal pane.
  a. Drag a signal from the signal pane to the **Expression** text field and double-click an operator from the **Operators** list.
  b. Select the signals from the signal pane and the operator from the **Operator** list, and then click the **Add** button. The **Clear** button clears the **Expression** text field.
- Type the signal's full path and the operator in the **Expression** text field.

Expressions List: This list displays the created **Logical Operation**. Each expression is displayed after the name and logical operation are entered in the

Name and **Expression** text fields respectively and the **Create** button is clicked. You can delete an expression listed in the **Expressions** list by selecting the item and clicking the **Delete** button.

Functions List: *nWave* supports the following C language functions:

abs(_) atan(_) atan2(_,_) cos(_) cosh(_) decibel(_)	exp(_) integer(_) integer(_) log(_) log10(_) max(_,_)	min(_,_) pow(_,_) sin(_) sinh(_) sqrt(_) tan(_)

Operators List: *nWave* supports the following C language operators:

+	addition	==	equality
-	subtraction	!=	inequality
*	multiplication	<	less
%	division	>	greater
!	not	<=	less or equal
&&	and	>=	greater or equal
||	or		

FFT

Menu Bar: **Analog -> FFT**

Refer to the *FFT Pane* chapter for details.

Tools Menu Options

New Waveform

Menu Bar: **Tools -> New Waveform**

This option opens a new *nWave* window where the same or different simulation result files can be displayed.

Waveform Compare

nWave provides a comprehensive comparison capability for automatically comparing results from different simulation runs. *nWave* graphically displays any mismatches in the waveform pane after the comparison is complete and then each mismatch can be stepped through to analyze the differences. When only one waveform window is open, only the **Compare 2 Signals** and **Compare Two Groups** options are enabled. It is recommended that the **View** -> **Hierarchical Name** option is enabled in *nWave* to obtain an accurate comparison of signals.

Options

Menu Bar: **Tools -> Waveform Compare -> Options**

This option opens the *Compare Options* form where the comparison criteria can be set. The *Compare Options* form include the **Digital** tab and **Analog** tab.

Digital Tab

The **Digital** tab specifies the comparison criteria to compare digital signals.

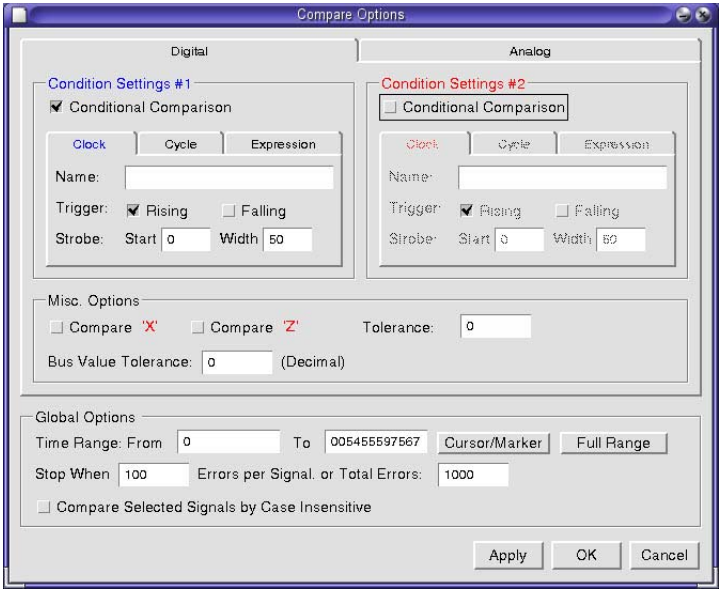


Figure: Digital Tab of Compare Options Form

The **Digital** tab includes **Condition Setting #1** and **Condition Setting #2** sections. When only one waveform window is open, the **First Signal** refers to the first signal (or the first set of signals) in the signal pane, and the **Second Signal** refers to the second signal (or the second set of signals) in the signal pane. The first and second signals correspond to the **Condition Settings #1** and **#2** respectively.

When two waveform windows are open, the **First Signal** refers to the signals in the master window (the window where the **Compare** options are invoked), while the **Second Signal** refers to the other waveform window (the slave window).

The following sections are available:

The **Condition Settings** sections include the following options and fields:

- **Conditional Comparison**: When this option is turned off, signal comparison is based on absolute time values. When this option is turned on, signal comparison is based on the definitions in the **Clock**, **Cycle**, and **Expression** tabs. If the clock signal file is closed, the **Conditional Comparison** option under the **Clock** tab is automatically disabled.
- **Clock Tab**
  - **Name:** Enter the hierarchical name of a signal or drag the signal from any *nWave*, *nTrace*, or *nSchema* window. This signal must be 1 bit and digital.

    - **Trigger:** Specify the trigger edge for comparison by turning *on* the **Rising** and/or **Falling** options.
    - **Strobe:** Specify the **Start** and **Width** of the strobe. The time unit is the **Window Time Unit** of the current waveform window. **Start** is an integer and **Width** is a natural number. The default is *0* for **Start** and *50* for **Width**.
- **Cycle Tab**

  This tab specifies the **Cycle Time**, **Delay**, and **Width of Strobe** using natural numbers. The default is *0* for **Delay** and *50* for **Width**.
- **Expression Tab**
    - **Expression**: Enter the hierarchical name of a signal or drag the signal from any *nWave*, *nTrace*, or *nSchema* window. The signal must be 1 bit and digital. A 1-bit expression signal can be created using the *Logical Operation* form associated with the **Signal -> Logical Operation** option.
    - **Status:** Select **True** or **False**.

The **Misc. Options** sections include the following options and fields:

- **Compare X:** X is the unknown value shown in the waveform window using X characters and a red box line by default (). When this option is turned on, the values signal comparison also compares these unknown values. The default is *off*.
- **Compare Z:** Z is the high-impedance value shown in the waveform window using yellow horizontal lines by default. When this option is turned on, the signal comparison also compares these high-impedance values. The default is *off*.
- **Tolerance:** Specify the mismatch tolerance time defined by the **Window Time Unit**. *nWave* filters out any mismatch that is less than this tolerance value. For example, if the **Tolerance** is specified as 1ns, all mismatches less than 1ns is filtered. The default is *0*.
- **Bus Value Tolerance:** Specify the bus value mismatch tolerance. *nWave* filters out any mismatch that is less than this tolerance value. For example, if the value is specified as 2, then buses with fewer than 2 value mismatches are filtered. The default is *0*.

The **Global Options** sections include the following buttons, options, and fields:

- **Time Range:** The **From** and **To** text fields define the comparison time range. The default is the full range of the current window. The **Time Range** can be specified by one of the following methods:
    - Enter values directly into the **From** and **To** text fields.

  - Click the **Cursor/Marker** button to obtain the range defined by the **Cursor** and **Marker** positions in the current window.
  - Click the **Full Range** button to obtain the full range of the current window. This is the default option.
- **Cursor/Marker:** Click this button to obtain the **From** and **To** values defined by the **Cursor** and **Marker** positions in the current waveform window. *nWave* automatically assigns the smaller value to the **From** text field and the larger value to the **To** text field.
- **Full Range:** Click this button to specify the full range of the current window.
- **Stop When 'n' Errors Per Signal or Total Errors 'n'**: Specify the values for the number of mismatches before stopping a comparison based on either errors per signal or total errors. If **signal error** is greater than **total errors**, *nWave* checks **signal error** first, then **total errors**.
- **Compare Selected Signals by Case Insensitive:** When this option is turned on, the comparison of selected signals is case-insensitive. When this option is turned off, the comparison of selected signals is case-sensitive. The default is *off*.

Analog Tab

The **Analog** tab specifies the comparison criteria to compare analog signals.

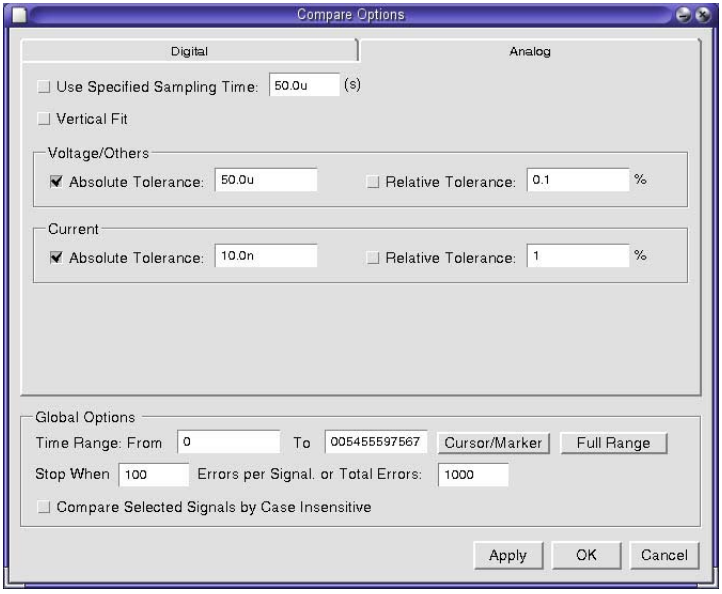


Figure: Analog Tab of Compare Options Form

The following options and sections are available:

- **Use Specified Sampling Time:** When this option is turned on, specify the sampling time by entering the time in the text field manually. The unit is *second*. When this option is turned off, the value change time dumped in the FSDB file is used as the sampling time. The default is *off*.
- **Vertical Fit:** When this option is turned on, the selected signals are vertically zoomed to their full range automatically after comparison. The default is *off*.

The **Voltage/Others** sections include the following options and fields:

- **Absolute Tolerance:** When this option is turned on, specify the absolute tolerance for the voltage or other objects by entering the value and unit of the tolerance in the text field manually. The default value is *50u*.The **Relative Tolerance** option is turned off. When this option is turned off, turn on the **Relative Tolerance** option and stop specifying the absolute tolerance. The default is *on*.
- **Relative Tolerance:** When this option is turned on, specify the relative tolerance for the voltage or other objects by entering the value of the tolerance in the text field manually. The default value is *0.1%*. The unit is %. The **Absolute Tolerance** option is turned off. When this option is turned off, turn on the **Absolute Tolerance** option and stop specifying the relative tolerance. The default is *off*.

The **Current** sections include the following options and fields:

- **Absolute Tolerance:** When this option is turned on, specify the absolute tolerance for the current by entering the value and unit of the tolerance in the text field manually. The default value is *1n*. The **Relative Tolerance** option is turned off. When this option is turned off, turn on the **Relative Tolerance** option and stop specifying the absolute tolerance. The default is *on*.
- **Relative Tolerance:** When this option is turned on, specify the relative tolerance for the current by entering the value of the tolerance in the text field manually. The unit is %. The default value is *1%*. The **Absolute Tolerance** option is turned off. When this option is turned off, turn on the **Absolute Tolerance** option and stop specifying the relative tolerance. The default is *off*.

The **Global Options** sections include the following buttons, options, and fields:

- **Time Range:** The **From** and **To** text fields define the comparison time range. The default is the full range of the current window. The **Time Range** can be specified by one of the following methods:

  - Enter values directly into the **From** and **To** text fields.
  - Click the **Cursor/Marker** button to obtain the range defined by the **Cursor** and **Marker** positions in the current window.
  - Click the **Full Range** button to obtain the full range of the current window. This is the default option.
- **Cursor/Marker:** Click this button to obtain the **From** and **To** values defined by the **Cursor** and **Marker** positions in the current waveform window. *nWave* automatically assigns the smaller value to the **From** text field and the larger value to the **To** text field.
- **Full Range:** Click this button to specify the full range of the current window.
- **Stop When 'n' Errors Per Signal or Total Errors 'n'**: Specify the values for the number of mismatches before stopping a comparison based on either errors per signal or total errors. If **signal error** is greater than **total errors**, *nWave* checks **signal error** first, then **total errors**.
- **Compare Selected Signals by Case Insensitive:** When this option is turned on, the comparison of selected signals is case-insensitive. When this option is turned off, the comparison of selected signals is case-sensitive. The default is *off*.

Compare 2 Signals

Menu Bar: **Tools -> Waveform Compare -> Compare 2 Signals**

This option compares two selected signals in one or two waveform panes. When only one waveform pane is open, select two different signals (they can be in the same group or different groups) and compare. When two waveform frames are open, a signal from each *nWave* window can be selected individually. A summary of the comparison results is presented in a comparison results form.

This option applies to transaction streams as well. The value change is marked as an error (red rectangle) on the first stream in the list when its begin time, end time, label, or value string (attributes) are not the same. If a transaction hides some of its attributes, then the comparison ignores the hidden attributes and only compares its begin time, end time, label, and other visible attributes. If the file time of a transaction FSDB is shifted, then this option compares the range of the new shifted time. If the value radix of a transaction signal is changed, then this option compares the new value string.

For example, take the two transaction streams shown in the following figure. Seven errors are found and they are marked as red rectangles.

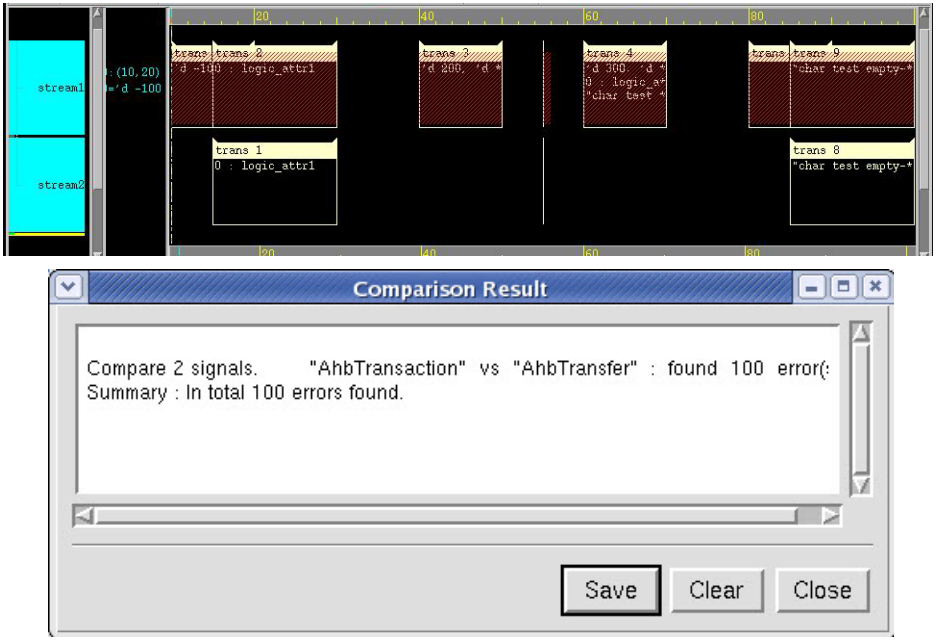


Figure: Results of Comparing Two Transaction Streams

Compare Selected Signals

Menu Bar: **Tools -> Waveform Compare -> Compare Selected Signals**

This option compares the selected signals. When two waveform frames are open, a signal can be selected from each pane. If only one signal is selected in each pane, *nWave* compares these signals only if the signal names are the same.

NOTE: This option is enabled when two waveform frames are open.

If any mismatches exist after the comparison, the **Search By** icon changes to the **Mismatches** icon . The **Search Backward/Forward/Upward/Downward** buttons on the toolbar can be used to locate the mismatches.

Compare Selected Signals to Another File

Menu Bar: **Tools -> Waveform Compare -> Compare Selected Signals to Another File**

This option compares the selected signals with the specified FSDB file in the same waveform pane. If any mismatches exist after the comparison, the mismatches are displayed as a red shadow on the waveform pane and the **Search**

By icon changes to the **Mismatches** icon  . The **Search Backward/Forward/Upward/Downward** buttons on the toolbar can be used to locate the mismatches.

Filter: This is the file format filter for the file list. Three default file extensions in the **Filter** field are 1) `*.fsdb`, 2) `*.fsdb`; `*.fsdb.gz`; `*.fsdb.bz2`, and 3) `*.*`. Additional filters can be added in this selection field using a semicolon (`;`) as the separator.

Compare Displayed Signals

Menu Bar: **Tools -> Waveform Compare -> Compare Displayed Signals**

This option compares a signal displayed in one waveform pane with the corresponding signal in a second waveform pane and presents a summary of the results in an *Information* form. If no corresponding signal exists, the *Information* form shows *"Could not find corresponding signal for..."*

NOTE: This option is enabled when two waveform frames are open.

Compare Two Groups

Menu Bar: **Tools -> Waveform Compare -> Compare Two Groups**

This option compares two groups of signals in the waveform frames. The comparison sequence is based on the signal sequence in each group (such as the first signal in the first group is compared with the first signal in the second group, and so on) These two compared signals must be of the same signal type. The pull-down menu can be used to select the comparison groups. The signal comparison results are displayed in an *Information* form. Click the **Save** button to save the error results to a file. Following are three common examples of possible error messages.

- "Signal A" vs. "Signal B": find X errors!
- "Signal C" vs. "Signal D": cannot be compared!
- Summary: In total Y errors found.

Since Signal C and D are not the same signal type, they cannot be compared.

NOTE: If the two groups contain a different number of signals, for example, five signals in group G1 and three signals in group G2, *nWave* only compares the first three signals.

If more than two *nWave* windows are opened, *nWave* always compares the signals in the current window with the “primary window” (that is, it has a red square in the lower-right corner). If the current window is the primary one, then *nWave* specifies any non-primary window as the window to be compared.

The primary window can be changed to specify the window to be compared; that is, *nWave* always compares the current window with the primary window if the current window is not primary.

Example

Suppose that three windows are opened, and *window_A* is the primary window. To compare the signals between *window_B* and *window_C*, *window_B* must be changed to the primary window and the compare action invoked from *window_C*'s main menu.

window_A (Primary) <== Second Window

window_B

window_C <== Golden Window

====>

window_A

window_B (Primary) <== Second Window

window_C <== Golden Window

In the *Compare Two Groups* form, the group comparison can be specified in the same window or different windows (if multiple *nWave* frames are open).

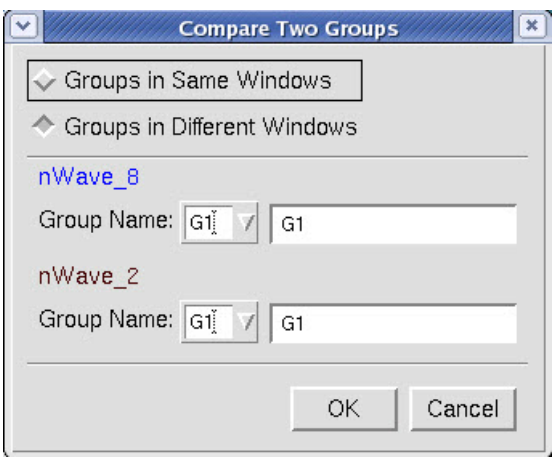


Figure: Compare Two Groups Form

Compare Signals from File

Menu Bar: **Tools -> Waveform Compare -> Compare Signals from File**

This option compares signals in two waveform windows against a file selected from the *Open Signal* list file form. Such a file contains only signal names with their full hierarchical paths. An example of this type of file is shown below:

```
/system/CYCLE
/system/R_W
```

Filter: This is the file format filter for the file list. Four default file extensions in the **Filter** field are 1) `*.err`, 2) `*.fsdb`, 3) `*.fsdb`; `*.fsdb.gz`; `*.fsdb.bz2`, and 4)`*.*`. Add another filter in this selection field using a semicolon (`;`) as the separator.

NOTE: This option is enabled when two waveform frames are open.

Clear Comparison Results

Menu Bar: **Tools -> Waveform Compare -> Clear Comparison Results**

This option clears all of the comparisons (red hatch marks) from the waveform.

Error Viewing

Load File

Menu Bar: **Tools -> Error Viewing -> Load File**

This option opens the *Load Message File* form where the message file can be loaded.

Select Error Types

Menu Bar: **Tools -> Error Viewing -> Select Error Types**

After a message file has been loaded through the **Tools -> ErrorViewing -> Load File** option, select the desired type of messages in the waveform window using the *Message Types* form.

Display Error Mark

Menu Bar: **Tools -> Error Viewing -> Display Error Mark**

When this toggle option is turned on, signals with messages are marked with red  in the waveform pane. Double-click a  to display the message in an *Information* form. This option also controls the display of error marks when comparing signal results.

Display Error Description

Menu Bar: **Tools -> Error Viewing -> Display Error Description**

When this toggle option is turned on, an *Information* form is displayed whenever the **Search By** option is invoked.

Select Message-Attached Signals

Menu Bar: **Tools -> Error Viewing -> Select Message-Attached Signals**

After a message file has been loaded, this option selects/highlights signals with messages attached.

Classic Transaction

Evaluator

Menu Bar: **Tools -> Classic Transaction -> Evaluator**

Refer to the **Evaluator** option in the *nTrace* chapter for details.

Analysis Window

Menu Bar: **Tools -> Classic Transaction -> Analysis Window**

Refer to the *Analysis Window* section in the *Transaction/Message Analyzer* chapter for details.

Add Selected to Analysis Window

Menu Bar: **Tools -> Classic Transaction -> Add Selected to Analysis Window**

This option opens the *Transaction/Message Analyzer* pane (if has not been opened previously) and adds the selected transaction streams to the *Transaction/Message Analyzer* pane.

Comparison Window

Menu Bar: **Tools -> Classic Transaction -> Comparison Window**

Refer to the **Comparison Window** option in the *Transaction/Message Analyzer* chapter for details.

Classic Message

Analysis Window

Menu Bar: **Tools -> Classic Message -> Analysis Window**

Refer to the *Analysis Window* section in the *Transaction/Message Analyzer* chapter for details.

Add Selected to Analysis Window

Menu Bar: **Tools -> Classic Message -> Add Selected to Analysis Window**

This option opens the *Transaction/Message Analyzer* pane (if it has not been opened previously) and adds the selected message streams to the *Transaction/Message Analyzer* pane.

Comparison Window

Menu Bar: **Tools -> Classic Message -> Comparison Window**

Refer to the **Comparison Window** option in the *Transaction/Message Analyzer* chapter for details.

New Schematic

Menu Bar: **Tools -> New Schematic**

Following are the sub-commands of the **New Schematic** command.

Trace X in Schematic

Menu Bar: **Tools -> New Schematic -> Trace X in Schematic**

This option opens a partial schematic view of the selected signal with the path to its drivers highlighted. If the selected signal's driver does not have a value X at the current time, a warning message is displayed and a partial schematic view is not displayed.

NOTE: If the opened FSDB file is EVCD, Trace X is not supported.

Active Fan-in

Menu Bar: **Tools -> New Schematic -> Active Fan-in**

This option can be invoked after a signal in *nWave* is selected and the cursor is placed over a transition. This option opens a form where a **Back Trace Time Period** of interest can be specified. Enter this time period and click the **OK** button. The result is a flat partial schematic showing the portion of the fan-in cone that was active (those elements/signals in the fan-in code that changed value) during the **Back Trace Time Period** prior to the transition. The results typically are a subset of the full fan-in that is displayed if the **Tools -> New Schematic -> Fan-in** option is used. Thus, the resulting schematic displays the logic pertinent to the transition selected, as opposed to displaying the entire fan-in cone.

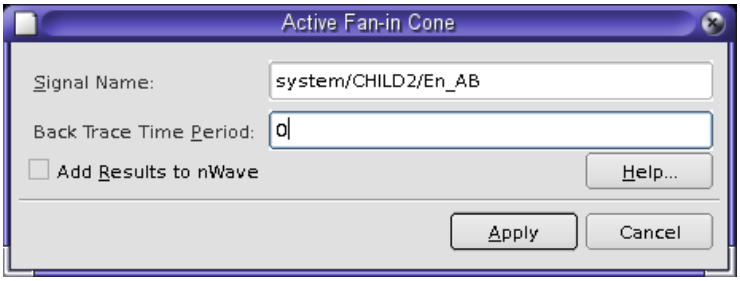


Figure: Active Fan-in Cone Form

Bus Contention

Menu Bar: **Tools -> New Schematic -> Bus Contention**

This option identifies bus contention (a net that is driven by more than one driver), and displays the results in *nSchema*. Select a signal in the signal pane and invoke this option. If the bus contention situation occurs on the selected signal at the current time, a partial schematic view is displayed and the active gates are also highlighted. If the incorrect value of the selected signal is not caused by bus contention, a warning message is displayed.

Contention by Time Range

Menu Bar: **Tools -> New Schematic -> Contention by Time Range**

This option is similar to the **Bus Contention** option. However, instead of using the current time, a bus contention situation occurring within a time range can be checked. If contention exists, then a *Bus Contention Report* form appears showing the times of any detected bus contentions.

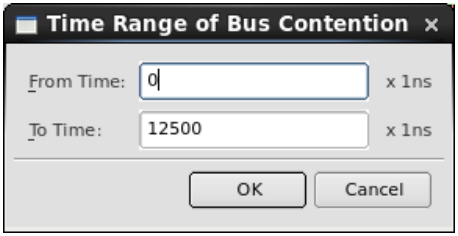


Figure: Bus Contention Report Form

This command opens the *Time Range of Bus Contention* form where the **From Time** and **To Time** fields can be set for the bus contention time range.

Temporal Flow View

Menu Bar: **Tools -> Temporal Flow View**

NOTE: This command is available when a signal is selected.

The **Temporal Flow View** menu includes the following sub-commands:

Auto Trace

Refer to the **Tools -> Temporal Flow View -> Auto Trace** command in the *nTrace* chapter for details.

Trace Glitch

Refer to the **Tools -> Temporal Flow View -> Trace Glitch** command in the *nTrace* chapter for details.

New Temporal Flow View

Refer to the **Tools -> Temporal Flow View -> New Temporal Flow View** command in the *nTrace* chapter for details.

Add Reference Signals

Refer to the **Tools -> Temporal Flow View -> Add Reference Signals** command description in the *Flow View* chapter for details.

Trace X

Menu Bar: **Tools -> Trace X**

This option opens the *Trace X Setting* form. Refer to the **Trace X** command description under the **Tools** command in the *nTrace* chapter for details.

Register

Menu Bar: **Tools -> Register**

This option opens an *nRegister* window. Refer to the *Register Window* chapter for details.

Memory/MDA

Menu Bar: **Tools -> Memory -> Memory/MDA**

Refer to the *Memory/MDA Pane* chapter for details.

Event Sequence

Menu Bar: **Tools -> Event Sequence**

Bind Key: Shift+E

This option opens an *Event Sequence* pane, which sequentially lists the transitions associated with the signals of interest at the current time spot. This option is activated when an FSDB file that was generated with the *NOVAS_FSDB_ENV_DUMP_SEQ_NUM* environment variable set to a non-null value is loaded and one or more signals are displayed in the *nWave* window. This option is also activated when a local variable, property, sequence, or assert is selected in the waveform regardless of the environment variable setting. This option also displays the sequential transitions of glitches if the *NOVAS_FSDB_ENV_MAX_GLITCH_NUM* environment variable was set to 0 (all glitches) or 2 or greater when the FSDB file was generated. Use the **Help -> About FSDB** option in the main toolbar to check the status of these environment variables for the current loaded FSDB file.

NOTE:

1. The following signal types are not displayed in the *Event Sequence* window:
   a. analog signal (real)
   b. comment signal
   c. overlap signal (analog overlap)
2. When the *nWave* main window reloads any FSDB files, the *Event Sequence* window created by this *nWave* window is refreshed automatically.

To use this option, select one or more signals of interest from the *nWave* signal pane, then invoke the **Tools -> Event Sequence** option. An *Event Sequence Window Options* form is opened. Some or all signals from a waveform window can be added to the *Event Sequence* pane by selecting either the **Selected** or **All** options.

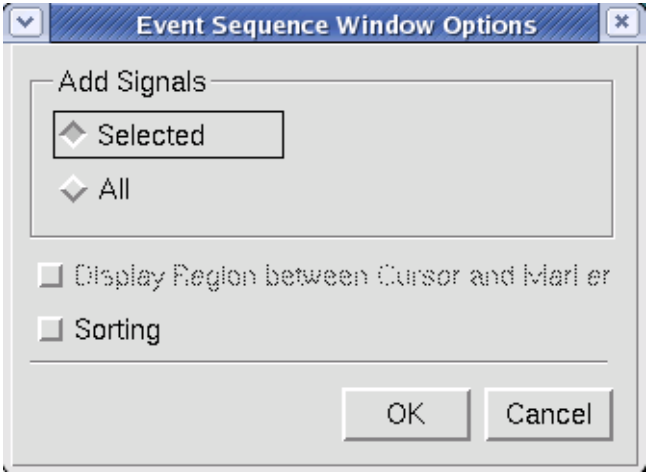


Figure: Event Sequence Window Options Form

The **Display Region Between Cursor And Marker** option enables viewing of the event range between cursor and marker.

The **Sorting** option sorts signals in the *Event Sequence* pane by value change sequence. This option is unavailable when the **Display Region Between Cursor And Marker** option is turned on.

After you set these options and click the **OK** button, the *Event Sequence* pane is added as a new tab in the same pane location as the source code pane. Add more signals to the *Event Sequence* pane by dragging them from the *nTrace* or *nWave* frames to the *Event Sequence* pane.

The *Event Sequence* pane has two display modes: **Waveform Mode** and **Grid Mode**. The default is **Waveform Mode**. Use the **Options -> Switch Displaying Mode** options (as described below) to switch the *Event Sequence* pane between the two modes. The setting of the last display view is saved into the *novas.rc* resource file and is effective the next time the Verdi platform is started. The options and usage of **Waveform Mode** and **Grid Mode** are similar, except a new column displaying the final stable value is present in the **Waveform Mode**.

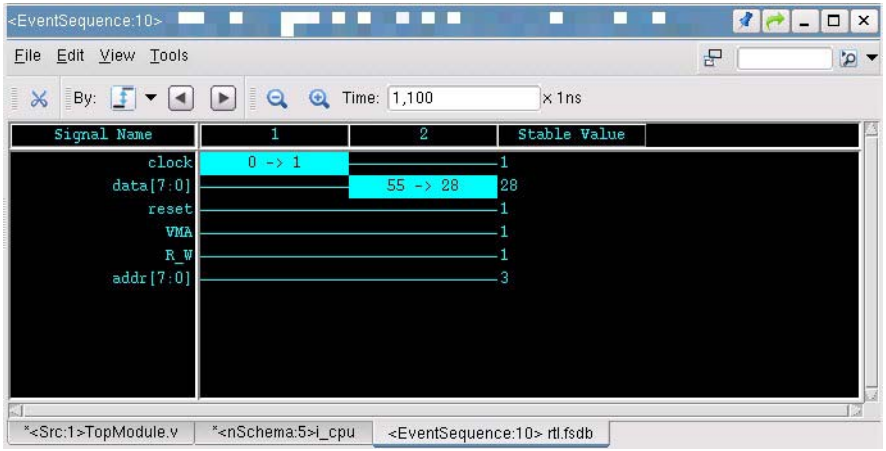


Figure: Event Sequence Window - Waveform Mode

All glitch occurrences associated with the selected signals at the current time spot are listed in sequence in the *Event Sequence* pane. Browse the glitch sequence to inspect circuit behavior at the time spot. Click the **Search Previous** or **Search Next** icons to jump to the previous or next time spot with any changes or glitches. The cursor time in the *Event Sequence* and *nWave* frames are synchronized when the **Options -> Sync Cursor Time** option is turned on.

Display Dump-off Sequence in the Waveform

In the event sequence window, for the dump-off sequence the column name appears as **Off**, as illustrated in the following figure:

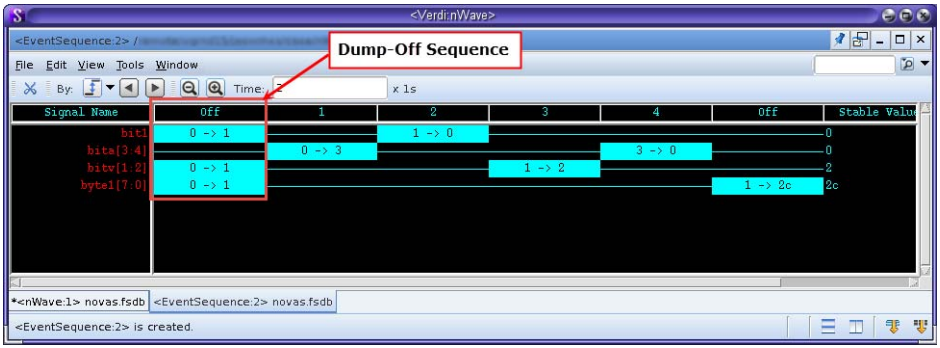


Figure: Example of Dump-off Sequence

File Menu Options

Add All Signals

Menu Bar: *Event Sequence*, **File -> Add All Signals**

This option adds all signals that are displayed in the waveform pane to the *Event Sequence* pane.

Save

Menu Bar: *Event Sequence*, **File -> Save**

This option saves the current *Event Sequence* pane to a text file.

Capture Window

Menu Bar: *Event Sequence*, **File -> Capture in PNG**

Refer to the **File -> Capture Window** option description for details.

Close Window

Menu Bar: *Event Sequence*, **File -> Close Window**

This option closes the *Event Sequence* pane.

Edit Menu Options

Select All

Menu Bar: *Event Sequence*, **Edit -> Select All**

This option selects all signals in the *Event Sequence* pane.

Cut

Menu Bar: *Event Sequence*, **Edit -> Cut**

Toolbar Icon: 

This option deletes the selected signals from the *Event Sequence* pane.

Set Search Value

Menu Bar: *Event Sequence*, **Edit -> Set Search Value**

This option opens the *Search Value* form where the signal value to be searched can be specified. When the **Case Sensitive** option is turned on, the case in the signal value is matched (such as `fe = fe` and `FE = FE`). When the **Case Sensitive** option is turned off, the case in the signal value is not matched (such as `fe = Fe = fE = FE`). The **Search Previous/Next** buttons can be used to search forward and backward in time to locate the different occurrences of the specified value.

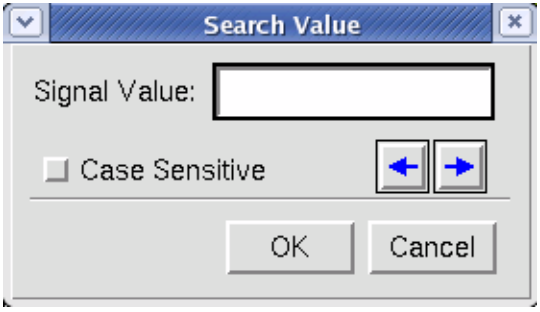


Figure: Search Value Form

Options Menu Options

Sync Cursor Time

Menu Bar: *Event Sequence*, **Options -> Sync Cursor Time**

This toggle option is turned on by default. It is used to synchronize the cursor time in the *Event Sequence* and *nWave* frames.

Sorting

Menu Bar: *Event Sequence*, **Options -> Sorting**

This option sorts the signals in the *Event Sequence* pane by value change sequence.

Remove Grid with Same-Value Cells

Menu Bar: *Event Sequence*, **Options -> Remove Grid with Same-Value Cells**

When this toggle option is turned on, continuous cells with the same value of the same row are merged.

NOTE: This option is always turned off while the contents of the *Event Sequence* pane are being saved to a text file.

Zoom

Zoom In

Menu Bar: *Event Sequence*, **Options -> Zoom -> Zoom In**

Toolbar Icon: 

Bind Key Shift+Z

This option provides a close-up view of the content in the *Event Sequence* pane. The magnification of the viewing area is changed to half the magnification of the previous view.

Zoom Out

Menu Bar: *Event Sequence*, **Options -> Zoom -> Zoom Out**

Toolbar Icon: 

Bind Key Z

This option enables more of the content to be seen in the *Event Sequence* pane at a reduced size. The magnification of the viewing area is changed to two times the magnification of the previous view.

Switch Displaying Mode

Menu Bar: *Event Sequence*, **Options -> Switch Displaying Mode**

This option switches the *Event Sequence* pane between **Display Grid Mode** and **Display Waveform Mode**. The **Waveform Mode** displays sequence time as waveforms. The **Grid Mode** displays sequence time as grid cells.

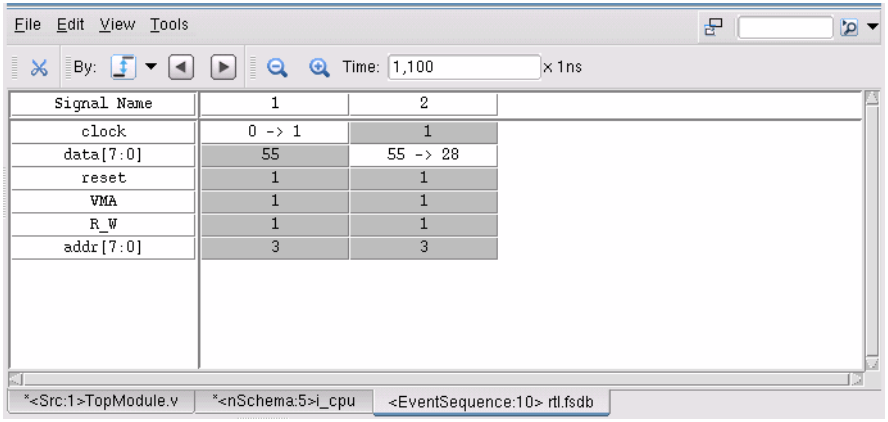


Figure: Event Sequence Window - Grid Mode

List X

Menu Bar: **Tools -> Browse/Watch/List -> List X**

NOTE: This option is enabled when a simulation results file has been loaded.

This option reports the variable types of storage elements and tri-states in the *List X* form. The *List X* form can be displayed by selecting one of the options from the *Source File to List X* form.

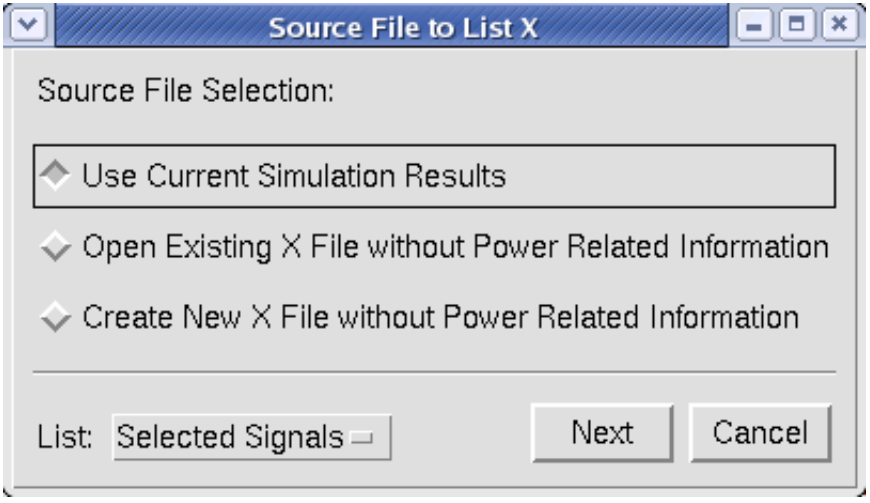


Figure: Source File to List X Form

The **Source File Selection** includes the following options. Only one option can be selected at a time.

- **Use Current Simulation Results**: When this option is selected, the current simulation results with power related information is used as the source file.
- **Open Existing X File without Power Related Information**: When this option is selected, specify the X file without power related information to use as the source file.
- **Create New X File without Power Related Information**: When this option is selected, create a new X file without power related information to use as the source file.
- **Next**: Displays different forms according to the selected options:
  - Display the *List X* form when the **Use Current Simulation Results** option is selected.
  - Display the *Create X File* form when the **Create New X File without Power Related Information** option is selected.
  - Display the *Create X File* form when the **Create New X File without Power Related Information** option is selected.

The **List** selection field includes the following options:

- **Selected Signals:** Display only the selected signals in the *List X* form.
- **All Signals:** Display all signals in the *List X* form.

The list columns in the *List X* form include: **Number (No.)**, **Name**, **Value**, **Time**, and **FFOut Type**. The **Power**, **Power Status**, **Driver Power**, and **Driver Power Status** columns are shown when the **Display Power Related Information** option is turned on.

NOTE: If the list includes new nodes, the numbers of these new nodes in the **Number (No.)** columns are shown in orange.

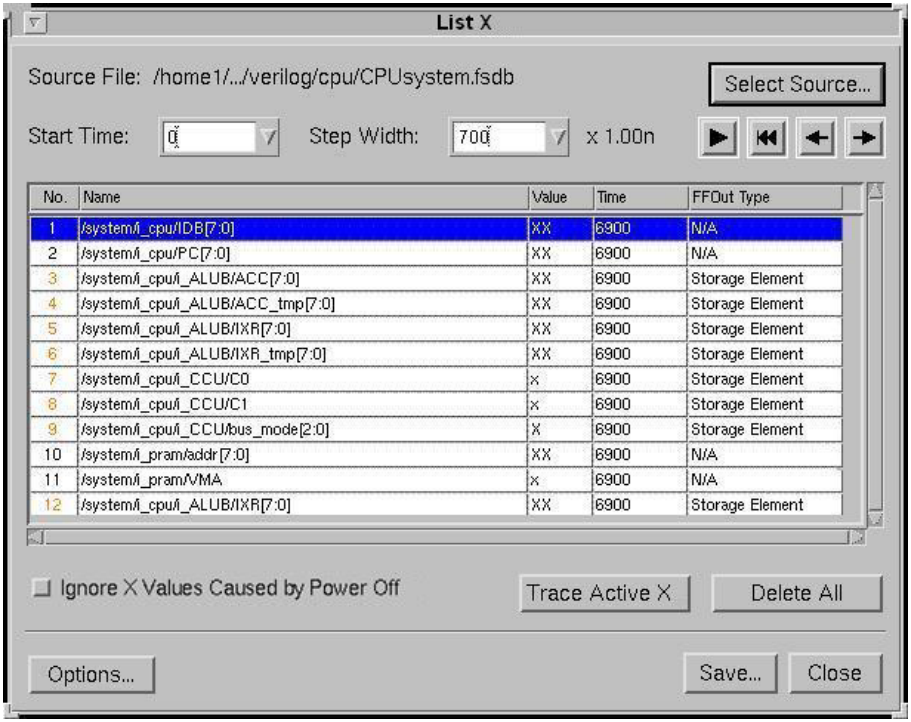


Figure: List X Form

The *List X* form includes the following fields, options, and buttons:

- **Source File:** Displays the source file to use for the List X function.
- **Start Time:** Specify the start time of the current viewing window. The default value is the source file begin time.
- **Step Width:** Specify the time width of each viewing window.
- **Select Source:** Opens the *Source File to List X* form to select the source file from.
- **Play** ▶**:** Applies the viewing settings to the List X value.
- **Fast Jump** ⏮**:** Jumps to the first time when the signals have unknown values and the time value is displayed in the **Start Time** text field.
- **Previous Search** ←**:** Searches backward for the previous time table.
- **Next Search** →**:** Searches forward for the next time table.
- **Ignore X Value Caused by Power Off:** When this option is turned on, the unknown (X) value is removed when the power domain is off or all driver power domains are off.

- **Trace X:** Tracks down the source of unknown (X) values. Refer to the **Trace X** option description under the right-click option menu of the *nWave* waveform pane for details.
- **Delete All:** Deletes all target signals in the list window.
- **Options:** Opens the *List X Options* form where the List X viewing settings can be specified.

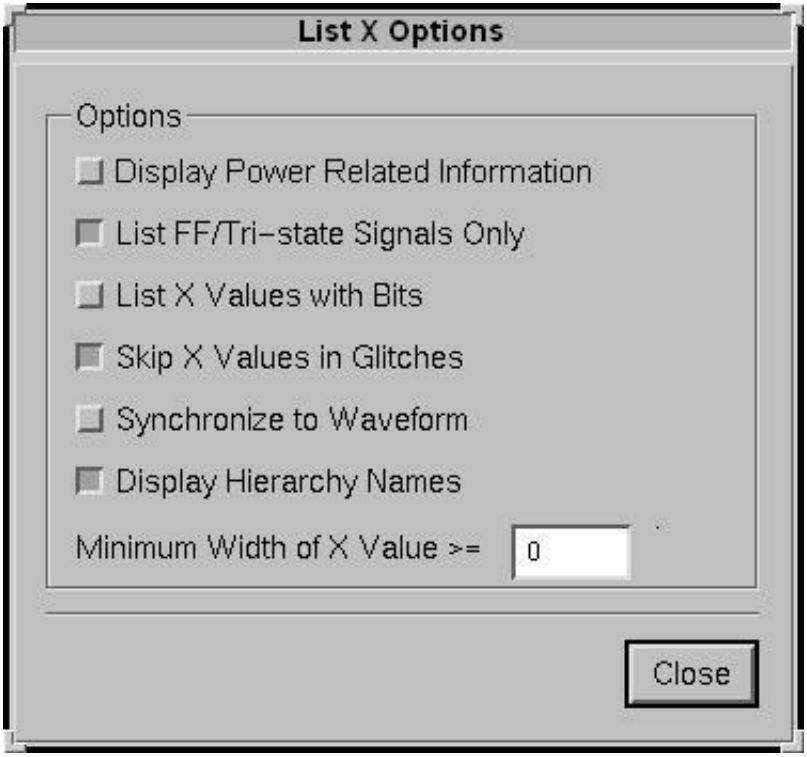


Figure: List X Options Form

The *List X Options* form includes the following options:

- **Display Power Related Information:** When this option is turned on, the power domain related information in the list window is displayed. The default is *off*.
- **List FF/Tri-state Signals Only**: When this option is turned on, only the flip-flop and Tri-state signals are displayed. The default is *on*.

- **List X Values with Bits:** When this option is turned on, only the bits of the specified signal with unknown (X) values are displayed. The default is *off*.

NOTE: If the bus signal is power related and has multiple power-domains, the bus is expanded whether this option is enabled or not.

- **Skip X Values in Glitches:** When this option is turned on, unknown (X) values caused by glitches are skipped. The default is *on*.
- **Synchronize to Waveform:** When this option is turned on, the signals in the selected row is synchronized with the signals in the primary *nWave* window while synchronizing the X time of the selected row with the cursor time in the primary *nWave* window. The default is *off*.
- **Display Hierarchy Names:** When this option is turned on, the full hierarchy names in the **Name** column is displayed. The default is *on*.
- **Minimum Width of X Value >=:** When this option is turned on, specify the minimum lasting width of the unknown X value. The default is *0*.

- **Save:** Opens the *Save List X* form and saves the listed X information to the specified text (*.txt*) file.
- **Close:** Closes the *List X* form.

Create X File

The *Create X File* form specifies the settings for the new X File.

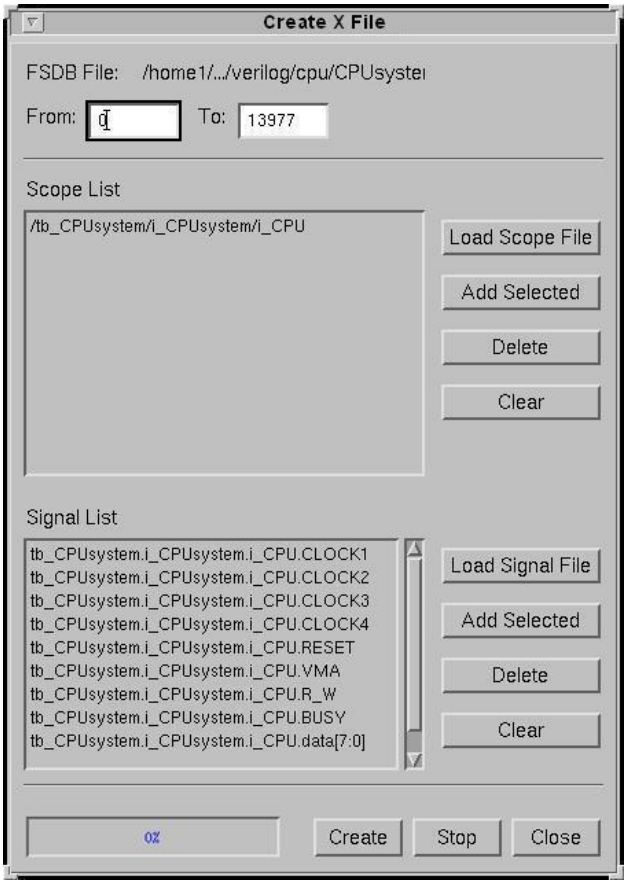


Figure: Create X File Form

The *Create X File* form includes the following fields and buttons:

- **FSDB File:** Displays the source file to use for generating the X file.
- **From:** Specify the start time for the generated X file.
- **To:** Specify the end time for the generated X file.
- The **Scope List** section specifies the scopes in the X file when the following buttons are clicked:
  - **Load Scope File:** Specify and load the scope files. Specify the number of scopes to be checked in the scope file. The scope file format is shown below.

```
scopeName {number_of_scope_to_be_checked}
```

The default number for *scopeName* is *1*. If the number is set to *0*, the List X function checks all of the signals under the specified scope and below.

Examples

```
/scope1/scope2 1
/scope3/scope4/scope5 0
/scope6/scope7
```

- **Add Selected:** Adds the scopes of the signals selected in the *nWave* window to the **Scope List** field.
- **Delete:** Removes the selected scopes.
- **Clear:** Removes all the scopes in the list.

- The **Signal List** section specifies the signals in the X file when the following buttons are clicked:
  - **Load Signal List:** Specify and load the signal files. The signal file format is shown below.

```
system.i_cpu.i_ALU.\ACC_tmp[7:0]
system.i_cpu.i_ALU.\IXR_tmp[7:0]
system.i_cpu.i_ALU.\X0[7:0]
system.i_cpu.i_ALU.\Y0[7:0]
system.i_cpu.i_ALU.alu_mode[2:0]
system.i_cpu.i_ALU.bus_mode[2:0]
system.i_cpu.i_ALU.carry
```

NOTE: A period (.) character is used as a delimiter.

  - **Add Selected:** Adds the signals selected in the *nWave* window to the **Scope List** field.
  - **Delete:** Removes the selected signals.
  - **Clear:** Removes all the signals in the list.
- **Create:** Generates a new X file following the settings in the *Create X File* form.

NOTE: If write permission does not exist for the FSDB file directory, the *Save As* form is displayed to specify the name and location to save the X file to.

- **Stop:** Terminates the X file process. The progress information is displayed at the bottom of the form.
- **Close:** Closes the *Create X File* form.

Property Tools

Evaluator

Menu Bar: **Tools -> Property Tools -> Evaluator**

Bind Key: Shift+E

NOTE: A design with SystemVerilog Assertions (SVA) code and a corresponding FSDB file must be loaded into the Verdi platform to enable this option.

This option opens the *Evaluate Properties* form where all SystemVerilog Assertions are listed and the assertions to be evaluated can be selected. Refer to the *Evaluate Properties Form* section in the *Assertion Debug* chapter for details.

Statistics

Menu Bar: **Tools -> Property Tools -> Statistics**

This option opens the *Statistics* pane. Refer to the *Statistics Pane* section in the *Assertion Debug* chapter for details.

Analyzer

Menu Bar: **Tools -> Property Tools -> Analyzer**

This option opens the *Analyzer* pane. Refer to the *Analyzer Pane* section in the *Assertion Debug* chapter for details.

Toggle Coverage Report

Menu Bar: **Tools -> Toggle Coverage Report**

This option opens the *Toggle Coverage* form where the conditions can be set to generate a report by analyzing the simulation results for signal transitions.

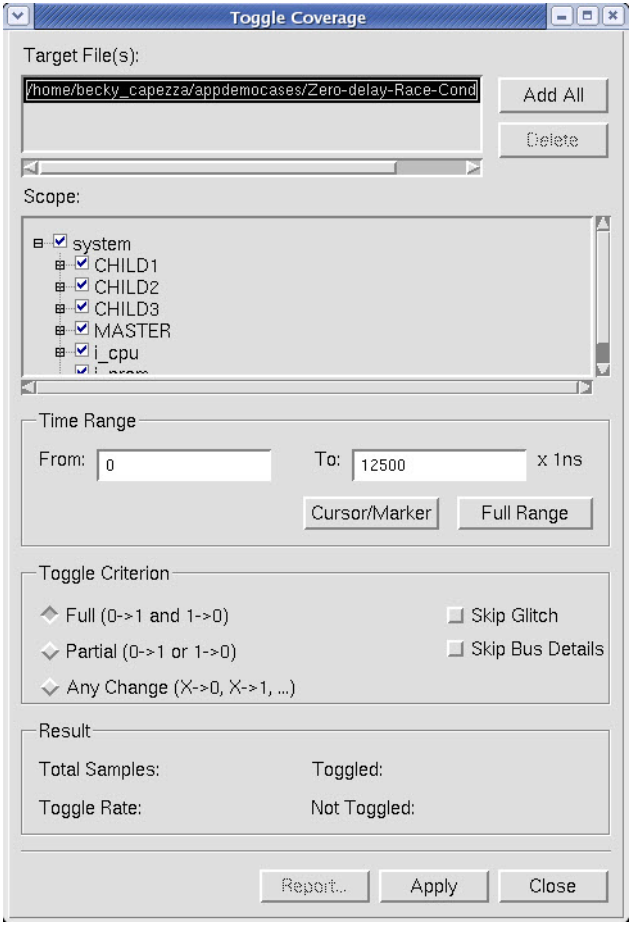


Figure: Toggle Coverage Form

The *Toggle Coverage* form includes the following sections, fields, options, and buttons.

- **Target Files:** This section lists the currently open FSDB files to use for calculating toggle coverage. When the *Toggle Coverage* form is first opened, the **Target Files** section contains one currently open FSDB file from the invoking *nWave* window. Click **Add All** to add all of the currently open FSDB files. Click **Delete** to delete the selected FSDB file from the list. Only one FSDB file can be selected at a time. If another FSDB file is loaded in *nWave*, the loaded file is automatically added to the **Target Files** list.

  Each file in the **Target Files** section can be selected individually to set the respective time range for the toggle coverage report.

- **Scope:** Specify the scopes for calculation of the toggle coverage report in this section. All the scopes are checked by default. If none of the scopes are checked, the **Apply** button is disabled. When the top scope is unchecked, all subscopes are unchecked.
- **Time Range:** Specify the time range for the toggle coverage report in this section.
  - **From**: Specify the start time.
  - **To**: Specify the end time.
  - **Cursor/Marker**: Click this button to synchronize the time range with the *nWave* cursor/marker times.
  - **Full Range**: Click this button to set the range to the entire simulation time.
- **Toggle Criterion:** Specify the criterion to be shown in the report in this section. Specify the type of transition to include by selecting one of the following options: **Full (0->1 and 1->0)**, **Partial (0->1 or 1->0)**, or **Any Change (X->0, X->1, ...)**.
  - **Skip Glitch:** When this option is turned on, glitches on signals are skipped (only the first/last transition is counted). When this option is turned off, glitches on signals are counted. The default is *off*.
  - **Skip Bus Details:** When this option is turned on, members of a bus is not expanded. When this option is turned off, members of a bus is expanded for single bit checking. The default is *off*.
- **Result:** After you set up the time range and the toggle criterion and click **Apply**, the results of all the FSDB files in the **Target Files** list are summarized as follows in this section:
  - **Total Samples:** total signals in the FSDB file
  - **Toggled:** total signals with transitions
  - **Not Toggled:** total signals without transitions
  - **Toggle Rate**: percent of toggled signals of the total available

Report: Click this button to open the *Toggle Coverage Report* form.

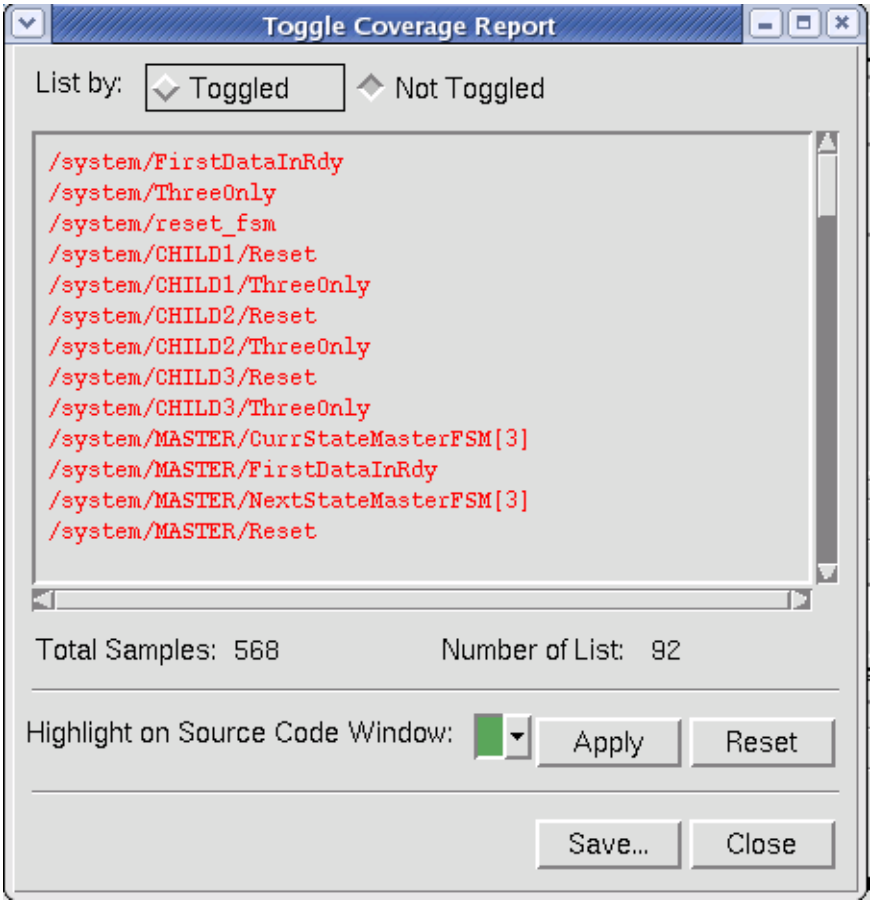


Figure: Toggle Coverage Report Form

The *Toggle Coverage Report* form includes the following options and fields:

- **Listed By:** Displays the **Toggled** or **Not Toggled** signals in the list. The **Total Samples** and **Number of List** (total for current selected display) are summarized at the bottom of the form.
- **Highlight on Source Code Window:** After selecting one or more signals (for multiple selections, press and hold the **Shift** key to select a range or the **Ctrl** key to select non-contiguous items) in the *Toggle Coverage Report* form, specify a color to highlight the selected signals in the source code pane.
  - Click the **Apply** button and the selected signals are highlighted in the specified color.

- Click the **Reset** button and the highlight color for the selected signals are reset to the default color. To reset just one signal, only select one signal; otherwise, all selected signals are reset. The default color is ID_GREEN3.

- **Save:** Save the toggle coverage report to a text file with the extension `.rpt`. A file browser form is opened.

Preferences

Menu Bar: **Tools -> Preferences**

Refer to the *Preferences* chapter for details.

nCompare

Menu Bar: **Tools -> nCompare**

Refer to the *nCompare* chapter for details.

Customize Menu/Toolbar

Refer to the **Tools -> Customize Menu/Toolbar** option in the *nTrace* chapter for details.

Window Menu Options

Tile Horizontally

Menu Bar: **Window -> Tile Horizontally**

Bind Key: `Ctrl+Alt+T`

NOTE: This option is available only in the standalone *nWave* window (that is, opened by the **verdi -nWave** or **nWave** option lines).

This option arranges multiple opened frames (such as *nRegister*, *nCompare*, and *Event Sequence*) in a horizontal tile pattern.

For frames opened in the standalone *nWave* window, the **Window -> Tile Horizontally** and **Window -> Tile Vertically** options are added automatically.

Tile Vertically

Menu Bar: **Window -> Tile Vertically**

Bind Key: `Ctrl+Alt+V`

NOTE: This option is available only in the standalone *nWave* window (that is, opened by the **verdi -nWave** or **nWave** option lines).

This option arranges multiple opened frames (such as *nRegister*, *nCompare*, and *Event Sequence*) in a vertical tile pattern.

For frames opened in the standalone *nWave* window, the **Window -> Tile Horizontally** and **Window -> Tile Vertically** options are added automatically.

Sync All Waveforms by

Menu Bar: **Window -> Sync All Waveforms by**

Cursor/Marker

Menu Bar: **Window -> Sync All Waveforms by -> Cursor/Marker**

This option synchronizes the views of multiple *nWave* frames. When this option is invoked in one window, this window's view and the **Cursor**/**Marker** movement are reflected in the other windows. This capability is helpful when multiple simulation results are compared and mismatches are verified.

This option synchronizes the waveform views based on the simulation time. Typically multiple windows are used to analyze several related simulation results. The synchronized items are:

- Cursor
- Marker

Consider, for example, that there are three open waveform windows: A, B, and C. When the **Cursor**/**Marker** option is invoked when viewing window A, window B and window C mirrors the cursor/marker in window A. When this option is enabled in all of the windows, they synchronize with each other.

Horizontal Range

Menu Bar: **Window -> Sync All Waveforms by -> Horizontal Range**

This option synchronizes the current visible range and the zoom ratio of multiple windows.

Vertical Scrolling

Menu Bar: **Window -> Sync All Waveforms by -> Vertical Scrolling**

This option synchronizes the vertical scrolling between the current window and other windows.

All

This option enables the **Cursor/Marker**, **Horizontal Range**, and V**ertical**

Menu Bar: **Window -> Sync All Waveforms by -> All**

Scrolling options to synchronize the views of multiple *nWave* windows.

Change to Primary

Menu Bar: **Window -> Change to Primary**

This option sets the current waveform window as the primary window. The primary waveform is annotated with a red square on the lower right corner of the waveform window  . By default, the first *nWave* window to be invoked is the primary window. All options and operations in the Verdi platform that communicate with *nWave* implicitly refer to the primary window. For example, time changes and the synchronization of signal selection (**Tools -> Options -> Sync. Signal Selection** in *nTrace*) are applied to the primary *nWave* window.

When you run the simulation, this command is enabled for non-primary *nWave* window which opens the annotation FSDB file individually.
When you run the simulation to close the primary *nWave* window, the annotation FSDB file should not be changed. Verdi tries to locate the other existing *nWave* windows with the annotation FSDB file opened individually.

If Verdi can locate the *nWave* window with the annotation FSDB file opened

individually, then the *Question* form appears mentioning that the located *nWave* window becomes the primary window.

If Verdi cannot locate the *nWave* window, then the *Question* form appears mentioning that all other *nWave* windows are closed automatically.

Horizontal Split

Menu Bar: **Window -> Horizontal Split**

This option divides the data representation area into upper and lower sections. Both sections contain the original signals displayed before splitting. The upper or lower section can be scrolled to compare the signals at the bottom of the upper section with those on top of the lower section.

Stop Split

Menu Bar: **Window -> Stop Split**

This option returns the display to a single window if the *nWave* window is split.

Right-Click Options

Many of the previously described options can also be selected from the right-click option in *nWave*.

nWave Window Right-Click Options

When the right mouse button is clicked anywhere in the menu, toolbar icon, or pane banner area of the *nWave* window or standalone window, a configuration option menu is displayed.This menu can be used to configure the available icons.

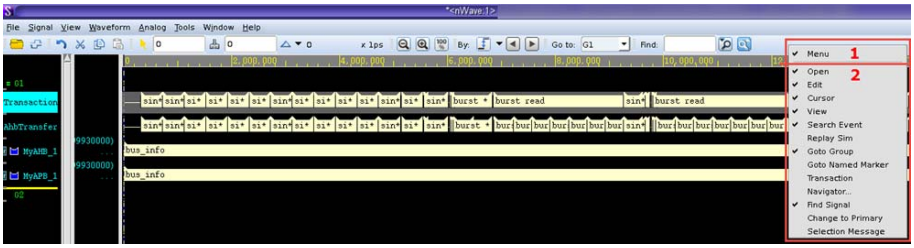


Figure: Configuration Option Menu

The **Menu** option (labeled 1 in the figure above) enables/disables the option menu bar. The options in the bottom section (labeled 2 in the figure above) enables/disables the toolbar icons for different functions.

Signal Pane Right-Click Options

The following list of right-click menu options are available for the signal pane of *nWave*. Some selections are available from the right mouse button menu. Other selections have equivalent selections available from the toolbar menus.

Show Search/Filter

Bind Key: Ctrl+F

Toolbar Icon:

This right-click option allows you to filter and search for a specific signal. This option allows you to display or hide the **Filter/Search** string text fields in the *Signal* pane. By default, the **Filter/Search** string text fields are hidden.

You can also change this default setting using the **Waveform Signal Pane** preference option in the **Search/Filter** page (**General -> Search/Filter -> Enable Search/Filter on -> Waveform Signal Pane**) of the *Preferences* form.

Click **Filter**  icon as shown in the following figure to filter the signals in the *Signal* pane. Multiple selections of the filter types are supported and select or clear the desired object types in the list.

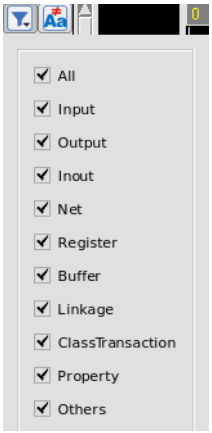


Figure: Filtering Signals

Enter a search string in the **Search String** text field and press the **Enter** key to locate the first match. Click the **Next signal** icon to find the next matching tree node or click the **Previous signal** icon to find the previous signal.

In the **Filter/Search** string text fields, the wildcard characters (* or ?) are supported to specify the pattern-matching string. For example, if *CH** is entered, *CHILD1*, *CHILD2*, and *CHILD3* are all found.

The **Filter/Search** string text fields also supports searching and filtering of multiple signals using the double vertical character (||) in the fields. For example, if *CH** || *En** are entered, then all signals starting with *CH* and *En* are found.

You can also use the **Case sensitive** icon to enable or disable case-sensitivity/insensitivity for the filter string and the search string entered in the **Filter/ Search** string text fields.

You can also close the **Search** text field by clicking the arrow on the left of the **Search** string text field.

NOTE: If the **Filter/Search** string text fields are enabled in the current *nWave* window, then the **Filter/Search** string text fields are not enabled in the consecutive *nWave* windows.

To enable the **Filter/Search** string text fields in all the *nWave* windows, enable the **Waveform Signal Pane** option in the **Search/Filter** page **(General -> Search/Filter -> Enable Search/Filter on -> Waveform Signal Pane)** of the *Preferences* form.

An example of the **Filter/Search** string text fields with the icons is illustrated in the following figure:

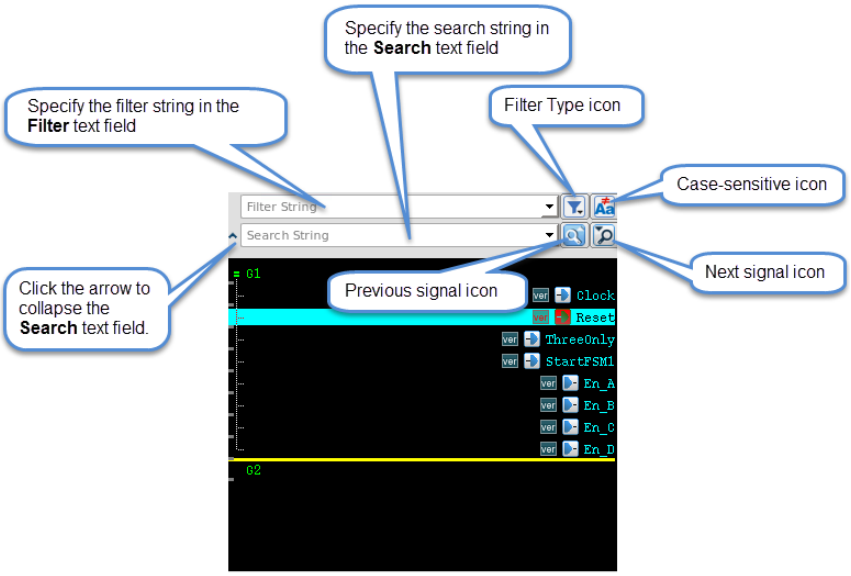


Figure: Example of the Filter/Search String Text Fields With the Icons.

Rename

This option changes the name of the signal or group.

Move

Refer to the **Signal -> Move** option description for details.

Set Signal Cursor

This option moves the signal cursor under the selected signal.

Group Operations

Four subcommands are available in the **Group Operations** menu: **Add Group, Insert Subgroup, Collapse Group,** and **Collapse All Groups**.

NOTE: When dragging a group from one *nWave* window to another *nWave* window, the name of the dragged group and its grouping structure is preserved to the dropped group. The preserved grouping structure includes all signals and subgroups under the group.
If the name of the dragged group already exists in the designated *nWave* window, a numeric suffix (such as #1) is appended.

Add Group

This option creates a first level group below the group where the signal cursor is located. For example, if group *G1* has signal *a* and *b* (*a* is above *b*) and the cursor is located below signal *a*, the new group is placed below signal *b*.

Insert Subgroup

This option creates a subgroup under the selected node. The group name can be edited after the subgroup is created.

Collapse Group

This option collapses the selected group. Double-click the selected group to expand or collapse a group.

Collapse All Groups

This option collapses all groups in the signal pane. Only the 1st level group names can be seen in the signal pane. No signals are displayed in the waveform pane.

Go To

This option jumps to the desired 1st level group.

Drag

Bind Key: Ctrl+C

This option allows the selected signal to be dragged to another location and dropped. This is similar to using the middle mouse button to select and drag.

Drop

Bind Key: Ctrl+V

This option allows the selected signal that is dragged from another location to be dropped in this window. This is similar to releasing the middle mouse button for a drop.

Copy Full Path

Bind Key: Ctrl+H

This option copies the full hierarchical names of the selected signals in the signal pane into the clipboard buffer.

OneTrace

NOTE: The following subcommands require a design to be loaded in *nTrace*. They are not available when a standalone *nWave* window is invoked and are enabled after selecting a signal in the signal pane.

Value Change

Refer to the **OneTrace -> Value Change** command description in the *nTrace* chapter for details.

Driver

This option inserts the final destination driver of the selected signal in *nWave* below the yellow cursor bar. Refer to the **OneTrace -> Driver** command description in the *nTrace* chapter for details.

Load

This option inserts the final destination load of the selected signal in *nWave* below the yellow cursor bar. Refer to the **OneTrace -> Load** command description in the *nTrace* chapter for details.

Connectivity

Refer to the **OneTrace -> Connectivity** command description in the *nTrace* chapter for details.

Chain Driver

This option traces through buffer/inverter chains and stops at the first driver with multiple inputs. The input/output signals of the multi-input instance is added in the signal pane as a new group. The format of the group naming rule is in the `Trigger_signal_simple_name#ChainDriver` format.

Refer to the **OneTrace -> Chain Driver** command description in the *nTrace* chapter for details.

Spice Signal

NOTE: This option does not support digital signals under digital scopes.

NOTE: This option is available when the selected signal is a spice signal with different value types.

Three subcommands are available in the **Spice Signal** menu: **Add Voltage Signal**, **Add Current Signal**, or **Add Logic Signal**.

Add Voltage Signal

NOTE: This option is available when the selected spice signal includes different voltage value types.

This option adds the voltage value of the selected spice signal in the *nWave* signal pane.

Add Current Signal

NOTE: This option is available when the selected spice signal includes different current value types.

This option adds the current value of the selected spice signal in the *nWave* signal pane.

Add Logic Signal

NOTE: This option is available when the selected spice signal includes different logic value types.

This option adds the logic value of the selected spice signal in the *nWave* signal pane.

Show Driver/Load Signals

Three subcommands are available in the **Show Driver/Load Signals** menu: **Driver Signals**, **Load Signals**, and **Chain Driver Signals**.

NOTE: The three subcommands require a design to be loaded in *nTrace*. They are not available when a standalone *nWave* window is invoked.

NOTE: The three subcommands are enabled after selecting a signal in the signal pane.

Driver Signals

This option inserts the final destination driver of the selected signal in *nWave* below the yellow cursor bar.

Load Signals

This option inserts the final destination load of the selected signal in *nWave* below the yellow cursor bar.

Chain Driver Signals

This option traces through buffer/inverter chains and stops at the first driver with multiple inputs. The input/output signals of the multi-input instance is added in the signal pane as a new group. The format of the group naming rule is in the `Trigger_signal_simple_name#ChainDriver` format.

Power

NOTE: This option is available when a CPF/UPF file is loaded.

Refer to the *Signal Pane Right-click Options* section in the *Power Aware Debug* chapter for details.

Cut

Bind Key: Delete

Refer to the **Signal -> Cut** option for details.

Copy

Bind Key: Ctrl+P

Refer to the **Signal -> Copy** option for details.

Paste

Bind Key: Insert

Refer to the **Signal -> Paste** option for details.

Split Inout

Refer to the **Signal -> Split Inout** option for details.

Expand/Collapse

Refer to the **Signal -> Expand/Collapse** option for details.

Collapse to Parent

Refer to the **Signal -> Collapse to Parent** option for details.

Bus Operations

Six subcommands are available in the **Bus Operations** menu: **Expand as Sub-bus, Edit Bus, Create Bus, Adjust Bus, Reverse Bus,** and **Add Full Bus**.

Expand as Sub-bus

Refer to the **Signal -> Bus Operations -> Expand as Sub-bus** option for details.

Edit Bus

Refer to the **Signal -> Bus Operations -> Edit Bus** option for details.

Create Bus

Refer to the **Signal -> Bus Operations -> Create Bus** option for details.

Adjust Bus

Refer to the **Signal -> Bus Operations -> Adjust Bus** option for details.

Reverse Bus

Refer to the **Signal -> Bus Operations -> Reverse Bus** option for details.

Add Full Bus

Refer to the **Signal -> Bus Operations -> Add Full Bus** option for details.

Highlight

Refer to the **Signal -> Highlight** option description for details.

Logical Operation

Refer to the **Signal -> Logical Operation** option description for details.

Add/Remove

The following are the sub-commands of the **Add/Remove** command.

Add Comment

This option is used to insert a comment field. Refer to the **Signal -> Comment -> Insert** option for details.

Add Counter Signal by

NOTE: This option is enabled after a signal is selected.

Three subcommands are available in the **Add Counter Signal by** menu: **Any Changes**, **Rising Edge**, and **Falling Edge**.

Any Changes

This option creates a counter signal with value changes in the same scope of the selected signal. The `AnyCounter_ prefix` is added automatically on the created counter signal.

Rising Edge

NOTE: This option is enabled after a vector or scalar signal is selected. The signal type such as assertion is not supported.

This option creates a counter signal with rising edges in the same scope of the selected signal. The `RisingCounter_ prefix` is added automatically on the created counter signal.

Falling Edge

NOTE: This option is enabled after a vector or scalar signal is selected. The signal type such as assertion is not supported.

This option creates a counter signal with falling edges in the same scope of the selected signal. The `FallingCounter_ prefix` is added automatically on the created counter signal.

Add Blank

This option is used to insert a blank line equivalent to the digital signal height at the current cursor location.

Add Trigger Signal

This option inserts one or more trigger signals to search with the **Active Annotation** option enabled. A signal in the waveform window must be selected (multiple signals are supported) before invoking this option. A trigger box

appears on the toolbar displaying the trigger signals. If a trigger signal is set, the results (cursor snap) of invoking the **Search Forward**/**Search Backward** options are not based on the selected signals in the signal pane. If a signal is already set to "trigger signal" and is to be renamed, the trigger signal in the box is also renamed.

NOTE: Invoking the **File -> Reload** option removes all trigger signals.

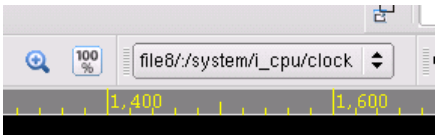


Figure: nWave Toolbar with Trigger Box

Remove Trigger Signal

This option deletes the selected trigger signals in the toolbar trigger box. One or more signals in the signal pane must be selected (for multiple selections, press and hold the **Shift** key to select a range or the **Ctrl** key to select non-contiguous items) before invoking this option.

When a trigger signal is removed from the waveform, the signal is removed from the trigger box automatically.

NOTE: If no trigger signals exist, this option is disabled and the trigger box is not shown.

Sort in Group

This option sorts the signals in the selected group globally in the *nWave Signal* pane either in ascending or descending order as follows:

Ascending

This option sorts all the signals in the selected group and the sub-groups in the *nWave* window in the ascending order of the signal name in the *Signal* pane.

Descending

This option sorts all the signals in the selected group and the sub-groups in the *nWave* window in the descending order of the signal name in the *Signal* pane.

Refer to the **Signal -> Sort Signals** option description for details.

Value Pane Right-Click Options

The following list of right-click menu options are available for the value pane of *nWave*. Some selections are available from the right mouse button menu. Other selections have equivalent selections available from the toolbar menus. The majority of the options are available for array signals; **Edit Alias**, **Remove Alias**, and **Remove Local Alias** are also available for single bit signals.

Radix

Refer to the **Waveform -> Signal Value Radix** option for detailed descriptions about the **Radix** options.

- Binary
- Octal
- Hexadecimal
- Decimal
- ASCII
- Enumerated Literal
- IEEE-754 Floating Point

When the **Radix** option in the *Value* pane is invoked, then the signals are displayed as follows:

- If the click position is in the selection range, then nWave keeps the current selection itself.
- If the click position is out of the selection range, then nWave gets the click position, finds the corresponding signal/group, and changes the selection to this signal/group.

Signal Type	Right-click Menu
Array Digital Signal	Radix Notation Apply Alias Edit Alias Remove Alias Analog Invert Local Format Apply Local Alias Remove Local Alias Copy Value String

Signal Type	Right-click Menu
One Bit Signals	Apply Alias Edit Alias Remove Alias Apply Local Alias Remove Local Alias
SC Fix Point Signals	Binary Decimal
Voltage	mv V
Current	uA mA A
Di/Dt	uA mA A
Analog Expressions	u m _ K

Points to Note:

- The right-click menu option displays appropriate option options for the selected signal.

  For example, if you select Analog Expression Signal `expression_1` and array digital signal `Clk`, and right-click to select **Analog**, only `Clk` is displayed as analog format.

- The toggle **On/Off** option and **Copy Value String** option remain disabled when multiple signals are selected.
- When radix is applied to the composite signals then all members of the signals apply the radix, expect the member signals which are of Enum type. However, you can change the format of the Enum type member signals by selecting the members directly and then applying radix.

NOTE: The toggle **On/Off** option is disabled when multiple signals are selected.

The **Copy Value String** and **Local Format** options remain disabled when multiple signals are selected.

Notation

Refer to the **Waveform -> Signal Value Notation** option for detailed descriptions about the **Notation** options.

- Unsigned
- Signed 2's Complement
- Signed 1's Complement
- Signed Magnitude

Apply Alias

NOTE: This option is not shown unless an alias table is added to the *nWave* window through the **Waveform -> Signal Value Radix -> Add Alias from File**, **Add Alias from Program**, or **Edit Alias** options.

This option applies the specified alias to all *nWave* waveforms of the same signal.

Edit Alias

Refer to the **Waveform -> Signal Value Radix -> Edit Alias** option for details.

Remove Alias

Refer to the **Waveform -> Signal Value Radix -> Remove Alias** option for details.

Analog

This option converts a digital bus signal to an analog waveform. Refer to the **Waveform -> Analog Waveform** option for details.

Invert

This option inverts the selected bus. Refer to the **Waveform -> Invert Waveform** option for details.

Local Format

This option opens the *Local Format Change* form where the radix and notation format can be changed for the selected bus locally (such as current *nWave* waveform only). Active annotation and other copies of the same signal is not affected.

Apply Local Alias

NOTE: This option is not shown unless an alias table is added to the *nWave* window through the **Waveform-> Signal Value Radix -> Add Alias from File**, **Add Alias from Program**, or **Edit Alias** options.

This option applies the specified alias to the current *nWave* waveform.

Remove Local Alias

This option removes the alias for the selected bus locally (such as current *nWave* waveform only). Active annotation and other copies of the same signal is not affected.

Copy Value String

This option copies the value strings of the selected signals in the signal pane into the clipboard buffer.

Waveform Pane Right-Click Options - General

The cursor must be over a signal waveform in the waveform pane of *nWave* before the right-click option is displayed.

Analog Waveform

Refer to the **Waveform -> Analog Waveform** option for details.

Digital Waveform

Refer to the **Waveform -> Digital Waveform** option for details.

Invert Waveform

Refer to the **Waveform -> Invert Waveform** option for details.

Temporal Flow View

Refer to the **Tools ->** ***Temporal Flow View*** option description in the *nTrace* chapter for details.

Following are the sub-commands under the Temporal Flow View command.

Auto Trace

Refer to the **Tools -> Temporal Flow View -> Auto Trace** command description for details.

Trace Glitch

Refer to the **Tools -> Temporal Flow View -> Trace Glitch** command description for details.

New Temporal Flow View

Refer to the **Tools -> Temporal Flow View -> New Temporal Flow View** command description for details.

Add Reference Signals

Refer to the **Tools -> Temporal Flow View -> Add Reference Signals** command description in the *Flow View* chapter for details.

Trace Glitch

Menu Bar:	**Trace -> Trace Glitch**
Mouse Action:	Right-click

NOTE To dump the glitch information during `simv` runtime, add the `+fsdb+glitch=0 +fsdb+sequential` option.

This option enables debugging a glitch in *Temporal Flow View* by identifying annotated glitches and locating the root source of the glitch. This option does not create a report but only helps trace signals with glitches.

When you invoke the **Trace Glitch** option from the *nWave* menu option, or from the right click option on the *Temporal Flow View*, the **Display Glitch** option is automatically turned on.

If a reference signal in the *Temporal Flow View* has a glitch, it is marked with a * symbol and the complete transition value is displayed, even if it is on the input side of the function block. The GUI option is available only when the selected signal contains glitches in its value.

Trace X

Refer to the **Tools -> Trace X** command description in the *nTrace* chapter for details.

Show Fan-in Registers

Bind Keys: Ctrl+W
Ctrl+Q

This option adds the fan-in registers of the selected signals in *nWave* to the waveform window. Two options are available:

- **All Fan-ins (Ctrl+Q):** Show all the fan-in registers on *nWave*.
- **Active Only (Ctrl+W):** Show only the active fan-in registers on *nWave*.

The **Display Signal to New nWave Window** and **When Adding Nodes to nWave** options in the **Temporal Flow View -> View** -> **Automatic Command Page** of the *Preferences* form determine the *nWave* window to which the signals are added (first option) and the location of the new signals (second option).

Show Fan-in Signals

Bind Keys: Ctrl+Y
Shift+A
Ctrl+Shift+T

This option adds the fan-in signals of the selected signals in *nWave* to the waveform window. Three options are available:

- **All Fan-ins (Shift+A):** Show all the fan-in signals in *nWave*.
- **Active Only (Ctrl+Y):** Show only the active fan-in signals in *nWave*.
- **Triggering Only (Ctrl+Shift+T):** Show only the fan-in signals that trigger a transition on the selected signal in *nWave*.

The **Display Signal to New nWave Window** and **When Adding Nodes to nWave** options in the **Temporal Flow View -> View** -> **Automatic Command Page** of the *Preferences* form determine the *nWave* window to which the signals are added (first option) and the location of the new signals (second option).

Show Clock (Domain)

This option adds the clock signal of the selected (left-click) signal in *nWave* to the waveform window. If the clock signal is already displayed on *nWave*, it is highlighted. If not, it is added above the yellow cursor.

The *nWave* window to which the signal is added is based on the state of the **Display Signal to New nWave Window** option in the **Temporal Flow View -> View -> Automatic Command Page** of the *Preferences* form.

Trace This Value

Bind Key: Ctrl+T

This option has the same effect as performing multiple right mouse button menu -> **Show Fan-in Signals -> Active Only** options recursively.

This option compares values on the active data fan-in signals with the value on the fan-out. If a matched value is found, the signal is automatically taken as a new root node and the operation continues. The matched signal is also added to the waveform. If multiple matched values are found or the maximum trace is reached, the operation stops. The first occurrence of a value of interest can be tracked down with a single option.

Several options are associated with the **Trace This Value** option. These options can be turned on/off in the **Temporal Flow View** -> **Trace** -> **Cycle-based Page** of the *Preferences* form.

The **Display Signal to New nWave Window** and **When Adding Nodes to nWave** options in the **Temporal Flow View -> Automatic Command Page** of the *Preferences* form determine the *nWave* window to which the signals are added (first option) and the location of the new signals (second option).

Trace Triggering Path

This option looks for the most recent transitions on the fan-in signals for the selected signals. The input nodes with the most recent transitions is automatically taken as new root nodes, the signal is added to the waveform, and the operation continues until no transitions exist or the specified stop criterion is met.

Several options are associated with the **Trace Triggering Path** option. These options can be turned on/off under the **Temporal Flow View** folder **-> Trace** folder **-> Transition-based Page** of the *Preferences* form (invoked with **Tools -> Preferences**).

Trace Unknown/Error Power State

NOTE: This option is available when a signal that contains an unknown or error value is selected in the waveform window.

This option tracks down the source of the unknown or error power state. When this option is invoked, the trace results are displayed on the *Trace Unknown/ Error Power State Results* form which is similar to the *Trace Trigerring X Results* window opened through the **Trace X** option. The corresponding state option is highlighted in the power source code pane and the reason for the error power

state and all active option expressions and issued supply nets are shown in the *Message* pane.

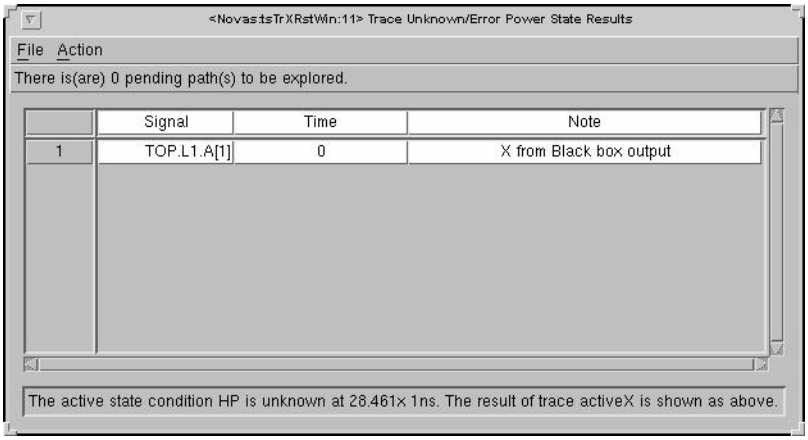
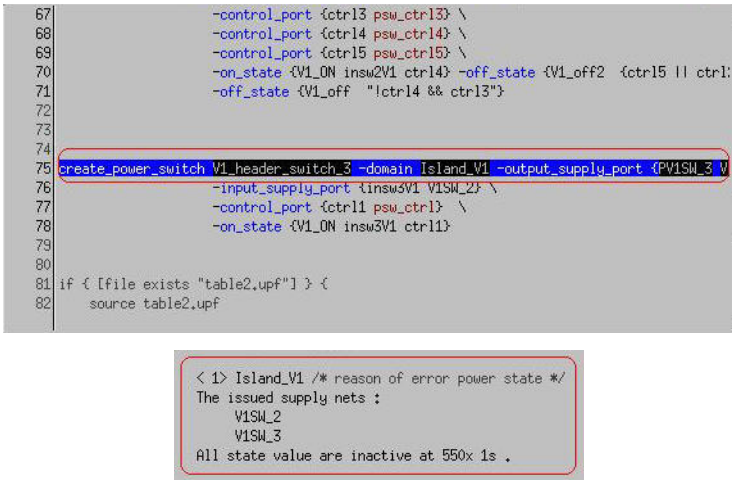


Figure: Trace Unknown/Error Power State Results Form

Refer to the *Trace Triggering X Results Window* section in the *Flow Views* chapter for details.

NOTE: For UPF designs, possible reasons for error power state include `-error_state` is true, `-control_port` is unknown, conflict between `-on_state` and `-off_state`, and none of `-on_state` and `-off_state` is active. Double-click a specific supply name to jump to the `create_power_switch - output_supply_port` whose output supply net is [`supply_name`] and highlight it in the

source code pane.



Add to Spreadsheet

This option opens the *Spreadsheet* window where data can be captured on the same signal from different simulation times. During the debug process, it is often required to compare data sampled at different simulation times or to evaluate an expression that involves data samples from different simulation times. Instead of manually summarizing the data, the Verdi platform provides an integrated *Spreadsheet* window. In the *Spreadsheet* window, data can be manipulated and evaluated so that cross-checking between the specification and implementation at each design checkpoint becomes easier.

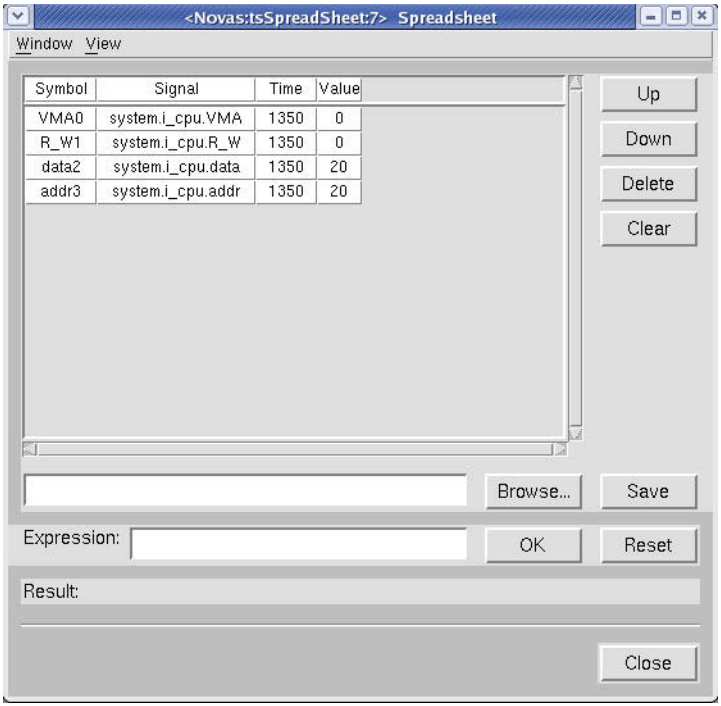


Figure: Spreadsheet Window

The main display area has four columns. The width of each column can be adjusted by placing the cursor over the vertical bar in the heading row and then pressing and holding the left mouse button.

- **Symbol:** A unique identifier for the signal. The default symbol name has the signal name as the base with a numerical value appended to the end. Right-click a row in the symbol column to invoke the context menu. Then, select **Rename** to change the symbol name. This opens another form where a **New Symbol Name** can be entered.
- **Signal:** Full hierarchical path for the selected signal.
- **Time:** Corresponds to the cursor time of the selected signal when the **Add to Spreadsheet** option is invoked.
- **Value:** Current value of the selected signal at the cursor time when the **Add to Spreadsheet** option is invoked.

Left-click anywhere along the row to select a row in the *Spreadsheet* window. You can select additional rows by holding the **Shift** key (selects a range) or the **Ctrl** key (selects individual properties) and by left-clicking. Selected rows can be de-selected with a left-click.

The *Spreadsheet* window has two pull-down menus as described below.

Window Menu Options
The **Close Window** option, located in the window menu, closes the current *Spreadsheet* window.

View Menu Options
The **Signal Value Radix** option in the view menu can change the radix display in the *Spreadsheet* window. Four sub-menus are available: **Binary, Octal, Hexadecimal, and Decimal**.

The *Spreadsheet* window includes the following buttons.

- **Up:** This button moves the selected row up one line at a time. Only one row can be selected for movement.
- **Down:** This button moves the selected row down one line at a time. Only one row can be selected for movement.
- **Delete:** This button deletes the selected rows from the *Spreadsheet* window. Right-click the symbol column for the row to invoke the context menu. Then, select **Delete** to delete a single row.
- **Clear:** This button removes all rows from the *Spreadsheet* window. The rows do not need to be selected.
- **Browse:** To save the data in the *Spreadsheet* window to a text (*.txt*) file, click this button to open a form where the directory structure can be viewed and a file name specified.
- **Save:** Click this button to save the data to the specified file.
- **Expression:** After one or more signals are added to the *Spreadsheet* window, a logical expression can be created using the symbol names. You can type in the symbol name in the expression text field or you enter the symbol name by right-clicking the symbol name and then selecting the **Insert to Expression** option from the context menu.

  The logical expression is a Verilog expression that may contain any of the following operators:

~	B-negation	!	L-negation
&	B-and, R-and	&&	L-and
~&	R-nand	||	L-or
|	B-or, R-or	==	L-equality
~|	R-nor	!=	L-inequality
^	B-xor, R-xor	<	less

~^	B-xnor, R-xnor	>	greater
+	addition	<=	less or equal
-	subtraction	>=	greater or equal
<<	left shift	*	multiplier
>>	right shift		

- **OK:** This button evaluates the expression and reports the result below the text field. If an error occurs in the expression, a warning window opens to describe the problem. The details are also printed in the message pane.
- **Reset:** This button clears the expression text field.
- **Close:** This button closes the *Spreadsheet* window. The information in the *Spreadsheet* window remains until the Verdi platform is exited.

Waveform Pane Right-Click Options - Transaction/ Message

Properties

This option provides more detailed information on the attributes of a selected transaction, as well as its related transactions.

Selecting a transaction highlights the related transactions.

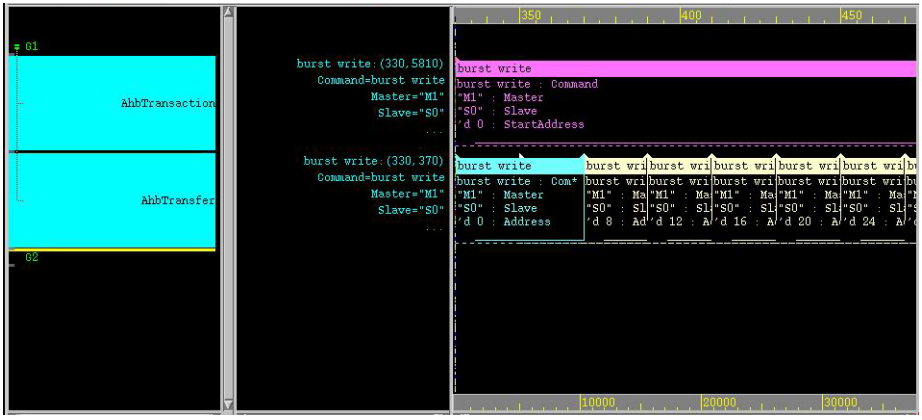


Figure: Transactions with Relationships

Click the label (top bar of the transaction) to select the transaction. The selected transaction is cyan and the related transactions are pink by default.

Click the right mouse button on the selected transaction to open the *Transaction Property* form where the transaction attributes and relationships can be further investigated. It is especially useful if the file contains a lot of attribute/relationship information such that the waveform pane cannot display them all on the screen. Only one attribute/relationship can be selected at a time.

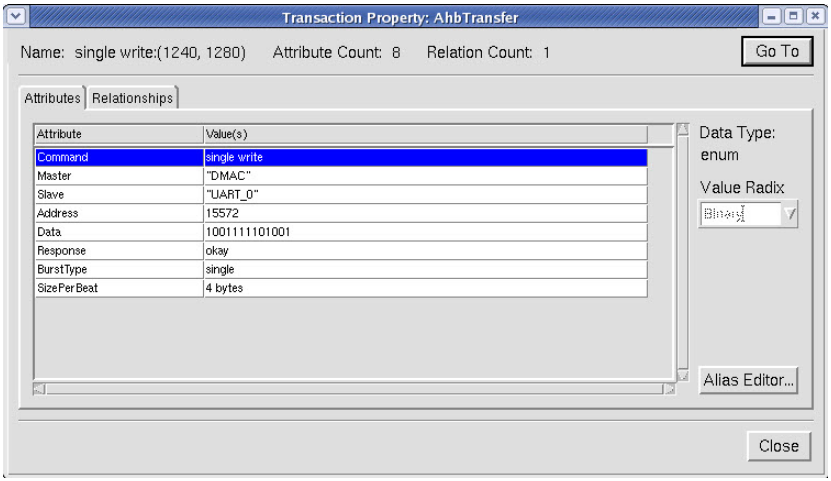


Figure: Transaction Property Form -- Attributes Tab

Both the **Attributes** and **Relationships** tabs include the following fields and button.

- **Name:** The name of the transaction.
- **Attribute Count:** Displays the number of attributes contained in the transaction. The first number is the non-hidden attribute count; the number in parentheses is the total attribute count.

  If the non-hidden attribute count equals the total attribute count, only one number is displayed.
- **Relation Count:** Displays the number of related transactions contained in the transaction.
- **Go To:** Click this button to move the cursor back to the selected transaction.

Attributes Tab

The **Attributes** tab includes the following columns, fields, and buttons:

- **Attribute Column**: Lists the attributes stored in the transaction.
- **Values Column**: Lists the value associated with each attribute in the transaction.

- **Data Type**: Indicates the type of the transaction. The supported attribute types are: **user_define, boolean, char, string, logic bus, int32, unit32, int64, unit64, float, double,** and **enum**.
- **Value Radix**: Specify the value radix of the transaction by selecting one of the following options: **Binary, Octal, Hexadecimal, UDecimal,** or **SDecimal**.
- **Alias Editor**: Click this button to open the *Alias Editor* form where aliases for specific attributes can be defined with *logic* data type. Refer to the **Waveform -> Signal Value Radix ->** **Edit Alias** option for more details on the *Alias Editor* form.

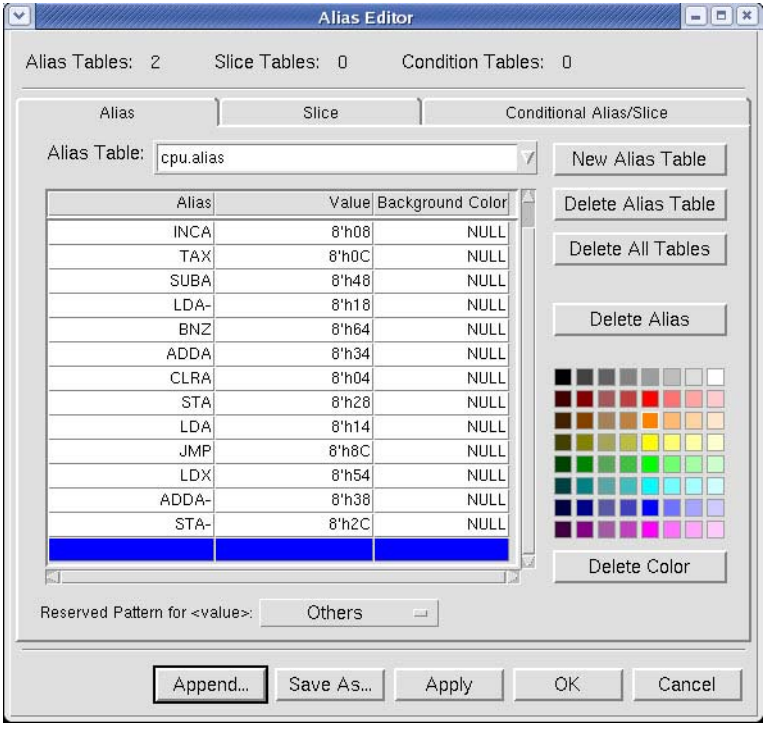


Figure: Alias Editor Form

For example, to set an alias for attribute *data*, create an alias table by giving the alias name (which is *Alias_Data* in this case) and entering the correct value of *data*. After you click **Apply** or **OK**, the alias table *Alias_Data* is created. Then return to the *Transaction Property* form to select the value radix *Alias_Data* as shown in the following figure:

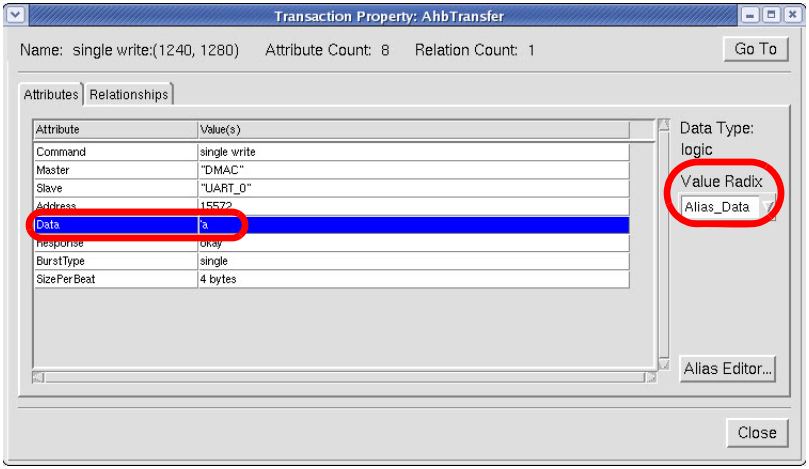


Figure: Transaction Property Form with Different Value Radix

The alias applied on both the value pane and the waveform pane is also shown in the waveform window.

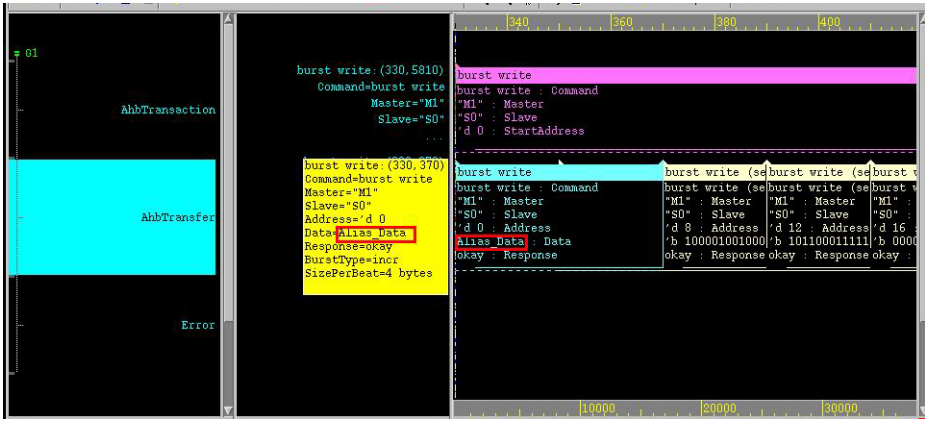


Figure: Alias Shown on the Waveform Window

NOTE: Use this feature to apply the radix to the specified attribute if the data format is correct. If the radix is applied, all attributes with this attribute_name is also changed.

Relationships Tab

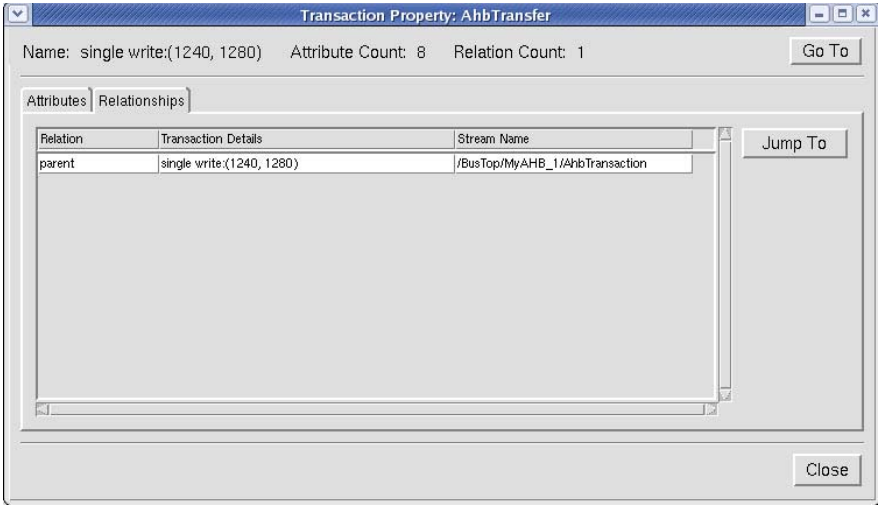


Figure: Transaction Property Form -- Relationships Tab

The **Relationships** tab includes the following columns and button:

- **Relation Column**: Describes the relationship between the transaction selected in the waveform and the related transactions.
- **Transaction Details Column**: Lists the basic information of the related transactions.
- **Stream Name Column**: Lists the stream names that the related transactions come from.
- **Jump To Button:** If a related transaction is selected and this button is clicked, *nWave* moves to select it.

Expand/Shrink Overlapping

Refer to the **Expand/Shrink Overlapping** option for details.

Hide/Display Attributes

Refer to the **Display/Hide Attributes** option for details.

Create Attribute Signals for Selected

Refer to the **Create Attribute Signals** option for details.

Filter/Colorize

Refer to the **Filter/Colorize** option for details.

Clear Filtering Results

Refer to the **Clear Filtering Results** option for details.

Clear Colorization Results

Refer to the **Clear Colorization Results** option for details.

Waveform Pane Right-Click Options - Assertions

Expand/Shrink Overlapping

Refer to the **Expand/Shrink Overlapping** option for details.

Toolbar Icons and Fields

Figure: Toolbar Used in nWave Window

The available toolbar icons may be modified. Refer to the *Toolbars* section of the *User Interface* chapter in the *Verdi User Guide and Tutorial* manual for details.

The different toolbar categories and available icons are described below.

Open Category

Open File

Refer to the **File** -> **Open** option description for details.

Set Active

Click this toolbar icon to select and set the active FSDB file name among the available FSDB files listed in the pull-down menu. Up to 9 files can be displayed. This icon is only activated when multiple FSDB files are opened.

Refer to the **File** -> **Set Active** option description for details.

Get Signals

Refer to the **Signal** -> **Get Signals** option description for details.

Edit Category

Undo

Click this toolbar icon to undo the last action in the signal pane.

Cut

Refer to the **Signal** -> **Cut** option description for details.

Copy

Refer to the **Signal** -> **Copy** option description for details.

Paste 

Refer to the **Signal -> Paste** option description for details.

Cursor Category

Cursor Time 

Click this toolbar icon to set the cursor position in the current waveform window.

Marker Time 

Click this toolbar icon to set the marker position in the current waveform window.

Delta Time 

Use the arrow icon to switch between **Delta Time** and **Frequency.** Click the **Delta Time** icon  to show the delta time between the cursor and marker. That is, **Delta = Marker - Cursor**. Click the **Frequency** icon  to show the frequency between the cursor and marker. That is, **Frequency = 1/|Cursor - Marker**.

Window Time Unit 

Refer to the **Waveform -> Waveform Time -> Set Window Time Unit** option description for details.

View Category

Zoom Out 

Refer to the **View -> Zoom -> Zoom Out** option description for details.

Zoom In 

Refer to the **View -> Zoom -> Zoom In** option description for details.

Zoom All 

Refer to the **View -> Zoom -> Zoom All** option description for details.

Search Event Category

Search By 

Click this toolbar icon to specify the search criteria for the value change of a signal.

Search Backward 

Click this toolbar icon to search backward in time and locate the value in the waveform window. This icon is 'active' only when an FSDB file is loaded and the **Active Annotation** option is enabled.

Search Forward 

Click this toolbar icon to search forward in time and locate the value in the waveform window. This icon is 'active' only when an FSDB file is loaded and the **Active Annotation** option is enabled.

Auto Trace 

NOTE: This toolbar icon is enabled when the **Active Annotation** option is turned on.
This toolbar icon is not available when a standalone *nWave* pane is invoked.

Refer to the **Auto Trace** toolbar icon description in the **Toolbar Icons and Fields** section of the *nTrace* chapter for details.

Goto Group Category

Go to Group  (Linux Only)

This text field displays the signal groups available in the *nWave* signal pane. Select a group name and the display automatically changes to that group.

For Solaris users, use the **Go To** option under the right mouse button context menu in the *nWave* signal pane to go to the desired group, or the **View -> Group Manager** option in *nWave* to perform further operations on the signal group/subgroup.

Goto Named Marker Category

Goto Marker 

Click this toolbar icon to select and go to the desired user-defined marker from the user marker list in the pull-down menu. The cursor moves to the time defined by the selected marker.

Find Signal

NOTE: This toolbar icon is available when the **Find Signal** right-click menu option is enabled.

Enter the desired signal name (such as, `MyAbP`, and `My*`) in the **Find** text field.

NOTE: If the **Match Prefix** option or the **Match Case** option are selected in the **Find Signal** form, then the results matching the specified options are displayed.

Previous 

Click this toolbar icon to search for the previous signal from the current signal selected in the signal pane.

Next 

Click this toolbar icon to search for the next signal in the signal pane.

Selection Message Category

Selection Message

This field displays information about the selected object.

Toolbar Icons for Get Signals Form

Find Signals 

Click this toolbar icon to find the desired signals.

Refer to the **Find Signal** text field in the **Signal -> Get Signals** option for details.

Select/Deselect All Signals 

Click this toolbar icon on to select or deselect all signals.

Select/Deselect All Mirror Signals 

Click this toolbar icon on to select or deselect all of the signals in the mirror signal pane.

Add Selected Signals 

Click this toolbar icon to add/insert the signal selected in the signal list to the position of the signal cursor bar (the yellow bar) in the mirror signal pane.

Sync. 

Click this toolbar icon to synchronize the mirror signal pane with the contents of the signal list.

Cut 

Click this toolbar icon to cut the selected signals from the signal pane into the clipboard buffer.

Copy 

Click this toolbar icon to copy the selected signals into the clipboard buffer.

Paste 

Click this toolbar icon to paste the signals from the clipboard buffer to the mirror signal pane under the signal cursor (the yellow bar).

nState

Overview

The *nState* window is displayed when the Finite State Machine (FSM) block is double-clicked in the *Schematic* window. The *nState* window is docked to the same frame location as the *Source Code* frame as a new tab. Click the **Undock** icon on the toolbar to make the *nState* window as a standalone window.

Refer to *Appendix B* in the *Verdi User Guide and Tutorial* for details on the supported FSM coding styles.

The menu bars for the *nState* window are displayed as follows:

Figure: nState Menu Bar

The menus are explained in the following sections. Refer to the *Icons for Dockable Panes* section for details on the menu bar icons.

NOTE: The nState window does not support global font.

Menu Summary

The *nState* menu options are summarized below. Double-click the commands listed below to jump to the corresponding command description. For the right-click menu options, refer to the *Right-Click Commands* section for details.

File Commands

Print
Capture Window
Close Window

View Commands

Large Font
Junction
State Action
Transition Condition
Transition Action
Port Link
Zoom ->
  Zoom In
  Zoom Out
  Zoom All
  Fit Select Set
  Fit All Always
Pan ->
  Pan Left
  Pan Right
  Pan Up
  Pan Down
Last View

FSM Commands

Find State
Find Signal
State Animation
Active Annotation
Add Signal(s) to Waveform
Previous State
Next State
Edit Search Sequences
State Delay
Machine Properties
Object Properties
Analysis Report

Tools Commands

Duplicate Window
Partial FSM
Preferences
Customize Menu/Toolbar

Bind Keys

For a complete list of bind keys used in *nState*, refer to the *nState* section in the *Bind Key Summary* for details.

File Commands

Print

Menu Bar: File -> Print

This command opens the *nState Print* form where print settings can be specified to print the current view of the *nState* window. The *nState Print* form includes four tabs, that is **Basic** tab, **Page Mode** tab, **Margins** tab, and **Report** tab.

Basic Tab

This tab provides the settings for the paper and printer.

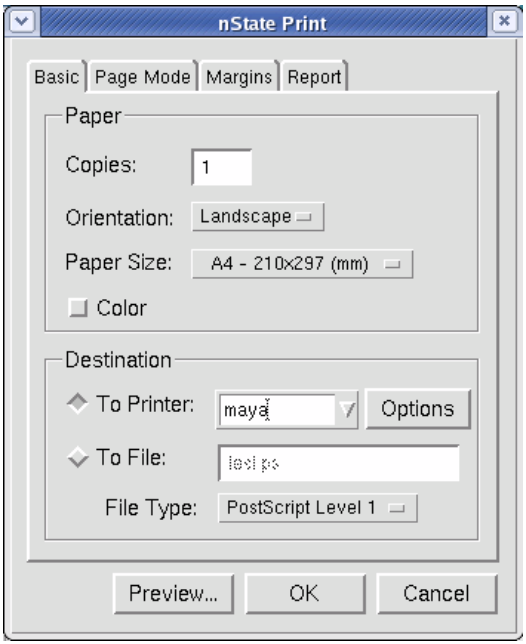


Figure: nState Print Form - Basic Tab

The **Paper** section includes the following options and fields:

- **Copies:** Enter the number of copies to print.
- **Orientation:** Specify the page orientation: **Landscape** or **Portrait**.
- **Paper Size:** Specify the paper size: **Letter, A4, A3, A2, A1, B, C, D,** or **E**.

- **Color:** Select this option to print multiple colors on a color printer. If not, the printout is displayed as black and white.

The **Destination** section includes the following options and fields:

- **To Printer:** Select this option to print the state diagram to a printer. If the printer name is not entered in the text field, *nState* uses the default printer.
- **Options:** Select this option to configure the print options as follows:
  - Enter the name of the printer in the **Printer Name** field.
  - Specify the printer language by selecting one of the following: **Postscript Level 1**, **Postscript Level 2**, or **HP GL/2**.
  - Select the **Paper Size** the printer supports.
  - In the **Print Command and Options** section, the default **Command** is *lpr*, the default **Destination** is *-P*, and the default number of **Copies** is *-#*. These entries can be used for printing the file and it may be configured for a specific print command.
- **To File:** Select this option to save the state diagram file to the working directory. The file name must be specified in the text field. When this option is selected, the **File Type** option must be set as **PostScript Level 2**, **PostScript Level 1**, or **GL2**.

Page Mode Tab

This tab provides the settings for the number of pages to print.

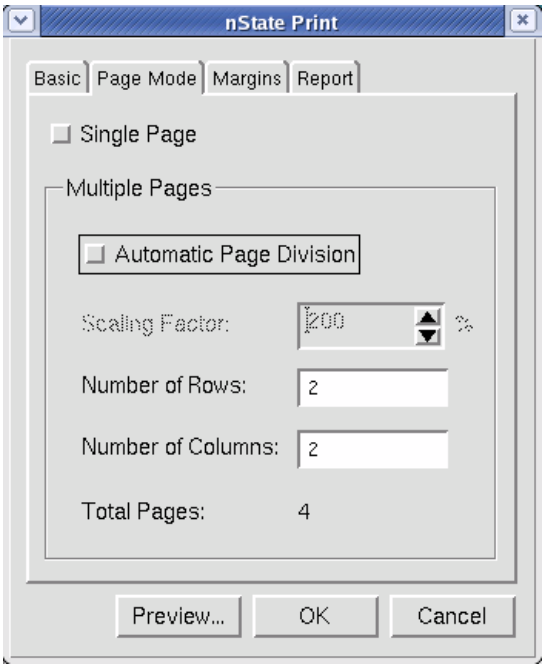


Figure: nState Print Form - Page Mode Tab

This tab includes two modes: **Single Page** and **Multiple Pages**.

- **Single Page:** Prints the FSM diagram on one page.
- **Multiple Pages:** Distributes the FSM diagram across multiple pages. To see how the FSM diagram is divided, click **Preview** in the *nState Print* form. For the **Multiple Pages** mode, specify whether to use **Automatic Page Division** or **Manual Page Division**. The **Automatic Page Division** option calculates the number of pages automatically based on the **Scaling Factor** and displays the number of rows, columns, and total pages. A **Scaling Factor** of 100% represents one page. The **Total Pages** value displays the number of pages the printout occupies in terms of x (columns) and y (rows). These values are recalculated automatically based on the **Scaling Factor**. Use the **Manual Page Division** option to specify the number of pages the printout occupies in terms of x (columns) and y (rows).

Margins Tab

This tab provides the settings for the print description and margins.

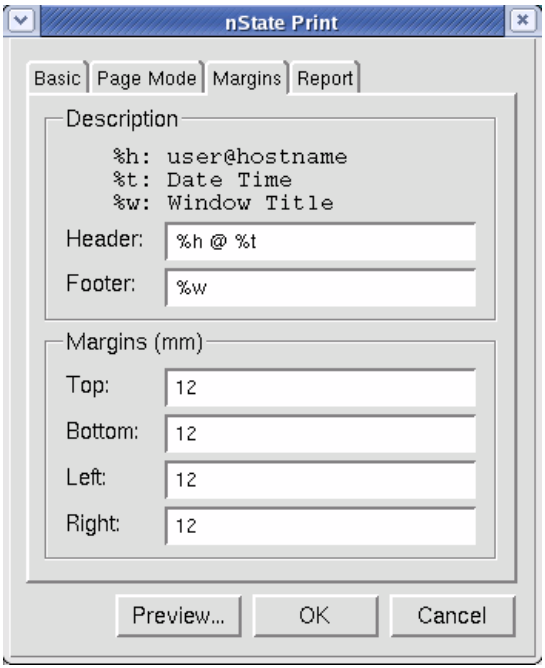


Figure: nState Print Form - Margins Tab

The **Description** section specifies the header and footer.

- **Header:** Displays the header information on the top of a printout. By default, the header includes the file name (%f), which is currently viewed in the *nState* window, user name and host name (%h), and date and time (%t).
- **Footer:** Displays the footer information on the bottom of a printout. Specify the text which needs to appear on the footer.

The **Margins (mm)** section specifies the margins in millimeters for **Top**, **Bottom**, **Left**, and **Right** in the appropriate text fields.

Report Tab

This tab provides the settings for printing the analysis report.

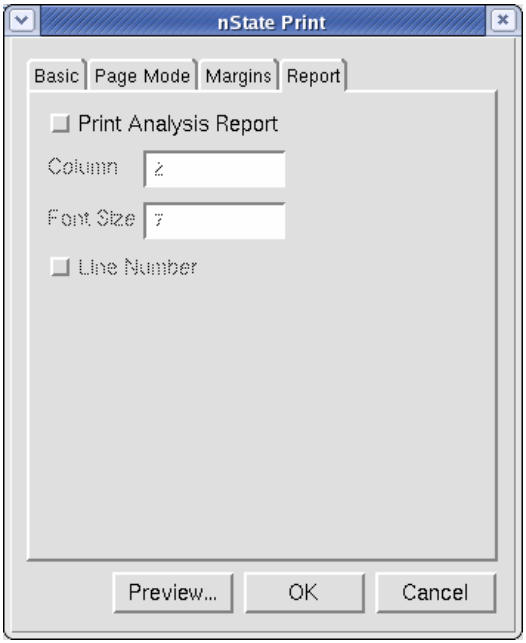


Figure: nState Print Form - Report Tab

The **Report** tab includes the following options and fields:

- **Print Analysis Report:** When an FSM diagram is printed and this option is turned *on*, the Analysis Report (For example, source code, state, and transition summary) for the FSM is also printed.
- **Column:** Enter a number in this text field for the preferred number of columns.
- **Font Size:** Enter a number in this text field for the preferred font size.
- **Line Number:** Select this option to include the line number for the source code of the printed analysis report.

Capture Window

Menu Bar: File -> Capture Window

This command opens the *Capture Window - Preview* form where an image in Portable Network Graphic format (PNG), a GNU standard similar to GIF can be the output.

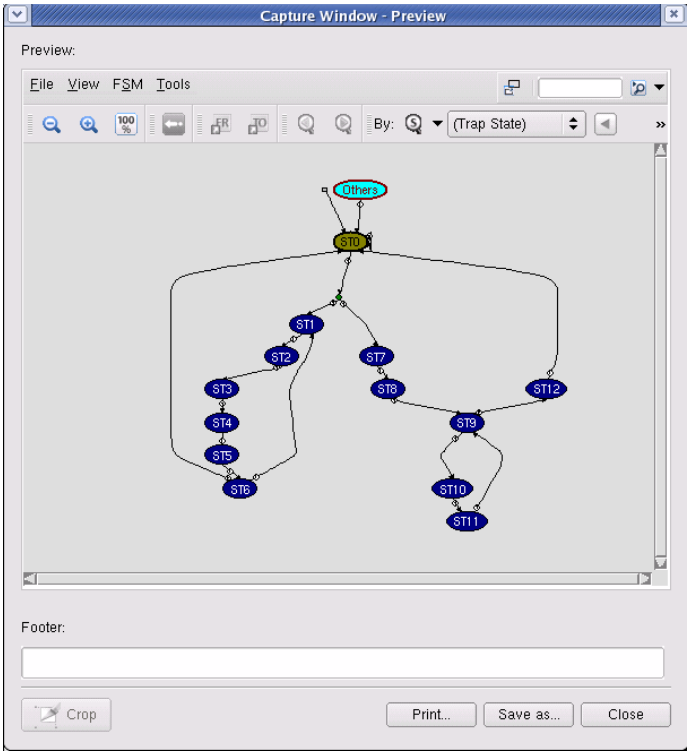


Figure: Capture Window - Preview Form

The following fields and options are available in the *Capture Window - Preview* form:

- **Footer:** Enter the appropriate text to display the content on the footer.
- **Crop:** Drag the left mouse button to select the required area and then click **Crop** to update the display in the *Capture Window - Preview* form.
- **Print:** This option opens a *Print* form where the required print options for creating a hard copy can be specified.
- **Save as:** This option opens a *Save As* form where the directory structure can be viewed and a file name for the PNG file can be specified.

Close Window

Menu Bar: File -> Close Window

This command closes the current *nState* window.

View Commands

Large Font

Menu Bar: View -> Large Font

This toggle command turns the display of large fonts in the *nState* window *on* or *off*.

Junction

Menu Bar: View -> Junction

This toggle command turns the display of junctions in the state diagram *on* or *off*. Junctions are graphic aids for breaking long transitions with complex conditions into shorter transitions with simpler conditions. This option is useful for understanding foreign designs.

State Action

Menu Bar: View -> State Action

This toggle command turns the display of state actions in the state diagram *on* or *off*.

Transition Condition

Menu Bar: View -> Transition Condition

This toggle command turns the display of transition conditions in the state diagram *on* or *off*.

Transition Action

Menu Bar: View -> Transition Action

This toggle command turns the display of transition actions in the state diagram *on* or *off*.

Port Link

Menu Bar: View -> Port Link

NOTE: The **Tools -> Extract Interactive FSM** command in the *nTrace* window must be executed and *nState* opened from the results for this command to be activated.

This toggle command turns the display of port links in the state diagram *on* or *off*.

Zoom

Zoom In

Menu Bar: View -> Zoom -> Zoom In

Toolbar Icon: 

Bind Key: Shift+Z

This command provides a close view of the content in the state diagram. The magnification of the viewing area is changed to half the magnification of the previous view in both the horizontal and vertical directions. Zooming can also be achieved by dragging-left to form a rectangle defining the desired zoom area.

Zoom Out

Menu Bar: View -> Zoom -> Zoom Out

Toolbar Icon: 

Bind Key: Z

This command enables more of the content to be seen in the state diagram at a reduced size. The magnification of the viewing area is changed to two times the

magnification of the previous view from the center point in both the horizontal and vertical directions.

Zoom All

Menu Bar:	View -> Zoom -> Zoom All
Toolbar Icon:	
Bind Key:	F

This command shows the entire content of the state diagram.

Fit Select Set

Menu Bar:	View -> Zoom -> Fit Select Set
Bind Key:	Ctrl+F

This command adjusts the zoom so that the current selected set fits the entire window.

Fit All Always

Menu Bar:	View -> Zoom -> Fit All Always

This toggle command forces the state diagram to fit the entire window at all times. When this command is turned *on*, the **Zoom In, Zoom Out, Zoom All, Fit Select Set, Pan Left, Pan Right, Pan Up, Pan Down**, and **Last View** commands are disabled.

Pan

Pan Left

Menu Bar:	View -> Pan -> Pan Left
Bind Key:	Left Arrow

This command pans the *nState* window to the left.

Pan Right

Menu Bar: View -> Pan -> Pan Right

Bind Key: Right Arrow

This command pans the *nState* window to the right.

NOTE: Panning left and right can also be achieved by moving the horizontal scroll bar in the *nState* window. Press the **Left Arrow** and **Right Arrow** keys, to pan the *nState* window to the left and the right, but to a lesser degree than **Ctrl+Left Arrow** and **Ctrl+Right Arrow**.

Pan Up

Menu Bar: View -> Pan -> Pan Up

Bind Key: Up Arrow

This command pans the *nState* window up.

Pan Down

Menu Bar: View -> Pan -> Pan Down

Bind Key: Down Arrow

This command pans the *nState* window down.

NOTE: Panning up and down can also be achieved by moving the vertical scroll bar in the *nState* window. Press the **Up Arrow** and **Down Arrow** keys, to pan the *nState* window up and down, but to a lesser degree than **Ctrl+Up Arrow** and **Ctrl+Down Arrow**.

Last View

Menu Bar: View -> Last View

Toolbar Icon: 

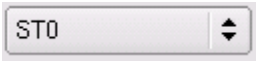
Bind Key: L

This command returns to the view associated with the last invoked viewing command and only saves one last view. When this command is invoked more than once, it toggles between the current and last views.

FSM Commands

Find State

Menu Bar: FSM -> Find State

Toolbar Icon: 

Bind Key: A

This command opens the *Find State* form that contains a list of all the states. When a state is selected from the list, the state diagram is automatically panned so the selected state is visible and highlighted.

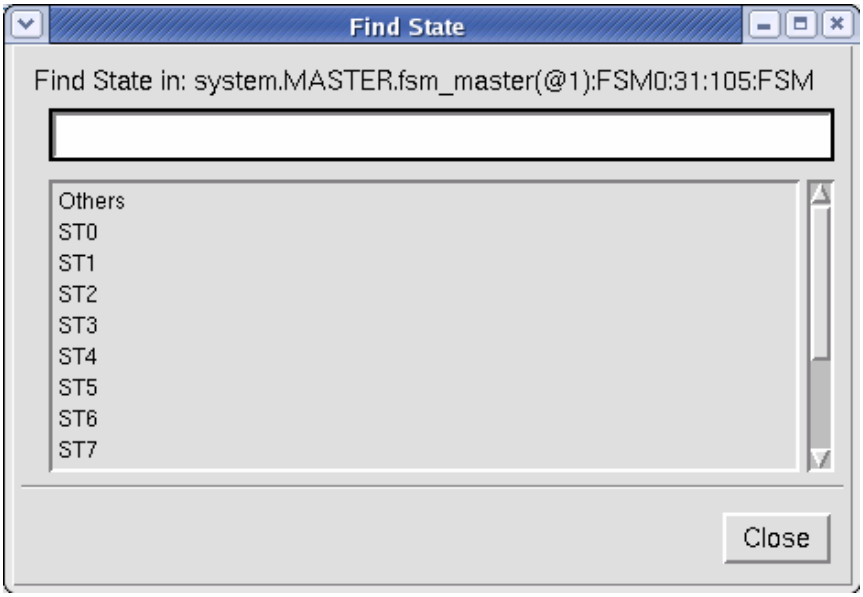


Figure: Find State Form

Find Signal

Menu Bar: FSM -> Find Signal

This command opens the *Find Signal* form where any signal can be located in the FSM. Select the desired hierarchical name from the signal list or enter the name

into the **Find Signal** text field. The FSM window simultaneously highlights the signal transitions and states that contain the local references.

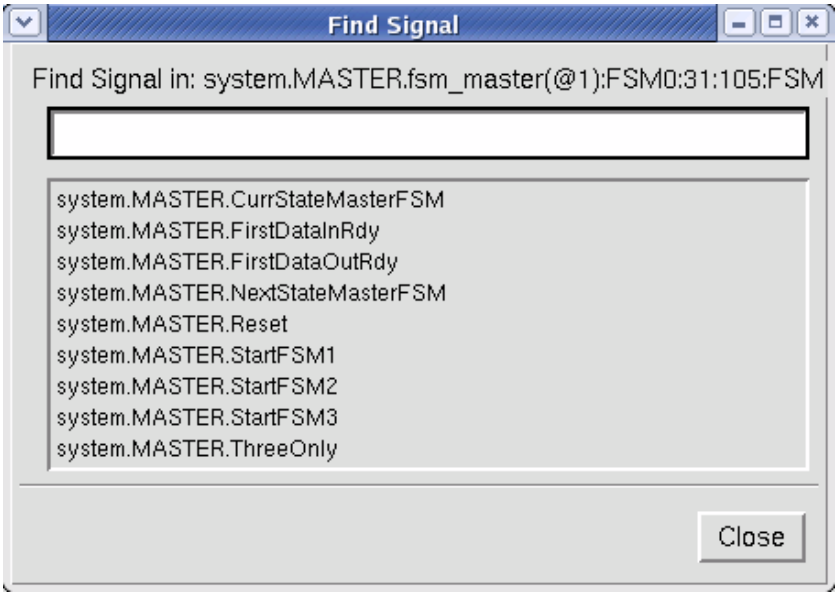


Figure: Find Signal Form

State Animation

Menu Bar: FSM -> State Animation

NOTE: This command is available when a simulation results file is loaded in *nTrace* or *nWave*.

This command enables stepping through the state sequences in the loaded simulation result file using the **Previous State** and **Next State** toolbar icons. *nState* steps through all the states sequentially even if the next/previous state is the same as the present state when the state change occurs. The found states and transitions are highlighted. This action synchronizes the cursor in the *nWave* window if the *nWave* window is open.

Active Annotation

Menu Bar: FSM -> Active Annotation

Bind Key: X

NOTE: This command is available when a simulation result file is loaded in *nTrace* or *nWave*.

This command displays the simulation results (the signal values) in *nState*. The transition arc is highlighted also. With the cursor time information in the toolbar, the signal values can be updated by changing the cursor time.

Add Signal(s) to Waveform

Menu Bar: FSM -> Add Signal(s) to Waveform

This command adds the state signal with its clock signal for the current *nState* window to the cursor bar position of the *Signal* pane in *nWave*. A simulation results file (FSDB file) must be loaded and an *nWave* frame opened for this command to be activated. This command can also be accessed using the right-click option.

Previous State

Menu Bar: FSM -> Previous State

Toolbar Icon: 

This command searches the current dump file backwards (that is, decreasing simulation time) for a state change event. When a state change event is located, *nState* highlights the state before the event, the state after the event, and the transition in between.

Next State

Menu Bar: FSM -> Next State

Toolbar Icon: 

This command searches the current dump file forwards (that is, increasing simulation time) for a state change event. When a state change event is located, *nState* highlights the state before the event, the state after the event, and the transition in between.

Edit Search Sequences

Menu Bar: FSM -> Edit Search Sequences

Bind Key: Ctrl+s

This command opens the *Search* form where a state sequence to be searched for may be defined. Before searching for a state sequence, a simulation results file must be loaded and the **FSM -> State Animation** option must be turned *on.*

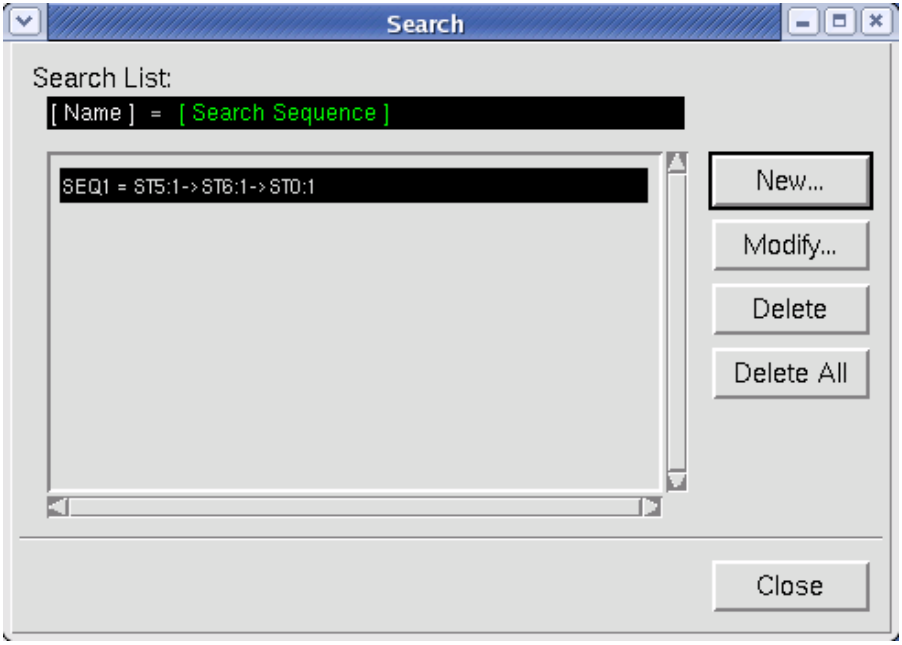


Figure: Search Form

The following options are available:

- **New**: Click this option to open the *New Search Sequence* form and specify the sequence.
- **Modify**: Click this option to open the *Modify Search Sequence* form and modify the selected sequence.
- **Delete**: Click this option to remove the selected search sequence.
- **Delete All**: Click this option to remove all search sequences from the **Search List**.

New/Modify Search Sequence Form

The *New Search Sequence* form specifies a state sequence and the *Modify Search Sequence* form modifies a previously defined sequence. The forms look identical; however, the *New Search Sequence* form is blank and the *Modify Search Sequence* form is filled. Click **Apply** or **OK** to add the state sequence to the **Find State** selection menu on the toolbar.

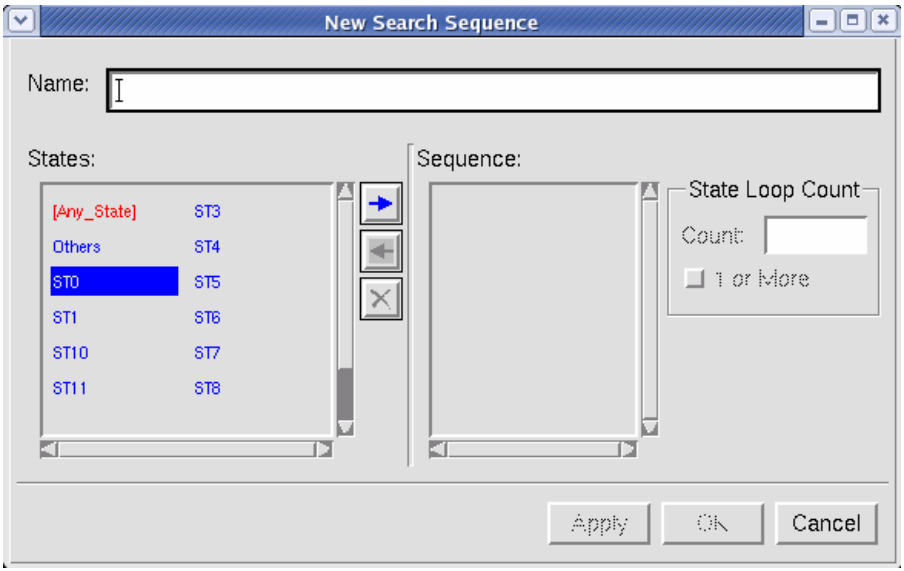


Figure: New Search Sequence Form

The following fields are available:

- **Name**: Enter/modify the name for the state search sequence.
- **States**: Lists the available states in the current view. Click **Add State** to move a state from the *States* pane to the *Sequence* pane. Click **Remove State** to move a state from the **Sequence** pane to the *States* pane.
- **Sequence**: Displays the state sequence. Click **Remove All** to remove all states from the list.

- **State Loop Count**: Specify the number for the state loop in the **Count** text field or enable **1 or More** option if any number of loops can be used.

State Delay

Menu Bar: FSM -> State Delay

NOTE: This command is available when a simulation results file is loaded in *nTrace* or *nWave*.

This command opens the *Specify Delay Time* form where the delay value of state variables can be specified.

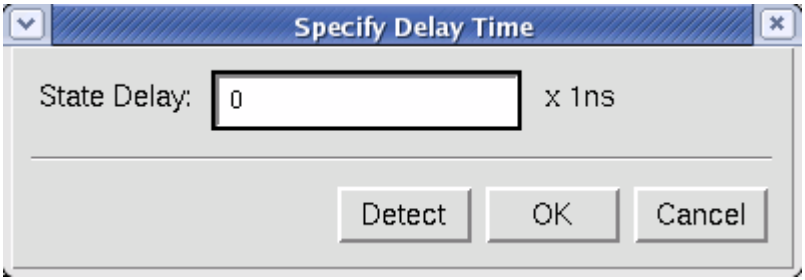


Figure: Specify Delay Time Form

The following fields are available:

- **State Delay:** When the *Specify Delay Time* form is opened for the first time, the **State Delay** text field displays the detected value. When this command is invoked again, the **State Delay** text field displays the current delay time used by the **Sequence Search**. Select **Enter** to check the value in the **State Delay** text field. If the value is valid, this form is closed; otherwise, a warning message is displayed. If the input value is invalid, **OK** is disabled.
- **Detect**: Click this option to display the detected value in the **State Delay** text field.
- **OK:** Click this option to update the delay time used by the **Sequence Search**.
- **Cancel**: Click this option to close the form without updating the delay time to the **Sequence Search**.

Machine Properties

Menu Bar: FSM -> Machine Properties

This command opens the *Machine Properties* form where the properties of the FSM are displayed. The *Machine Properties* form is intended for viewing purposes only. It is not possible to edit any values.

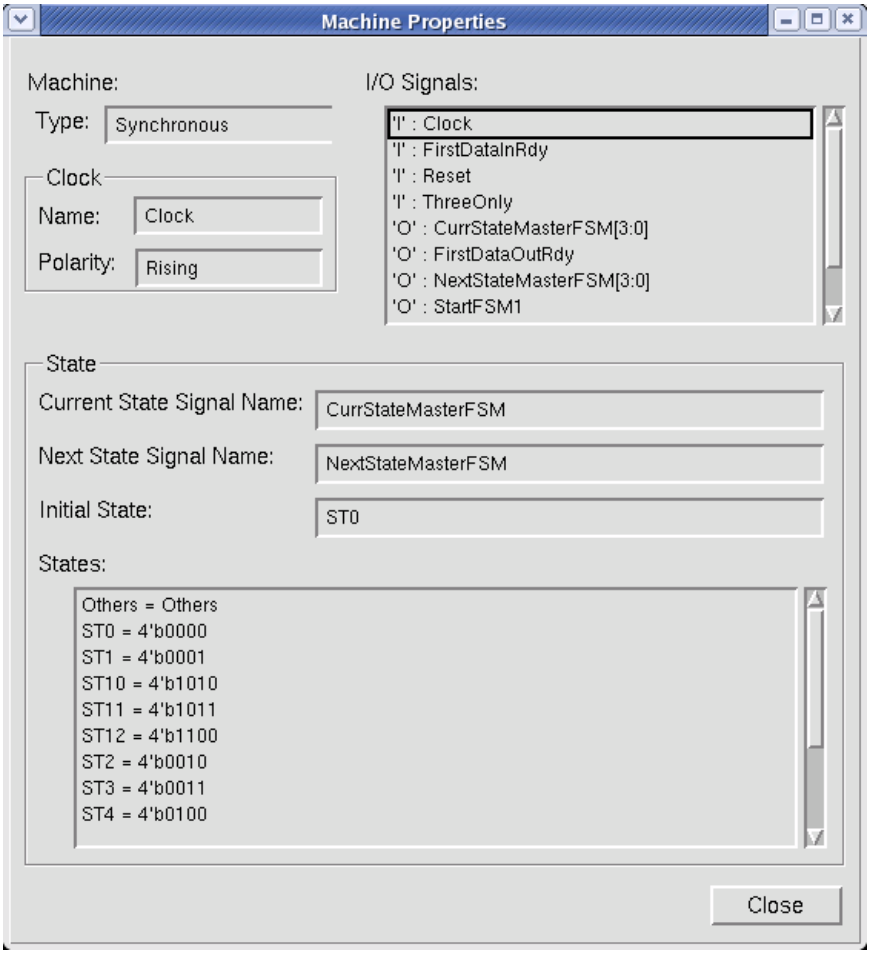


Figure: Machine Properties Form

The following read-only fields are available:

- **Machine Type:** Displays if the FSM is **Synchronous** or **Asynchronous**.
- **Clock**: If the FSM is the **Synchronous** option, then the synchronizing signal appears in the **Name** text field and if it is rising/falling in the **Polarity**

text field. If the FSM is the **Asynchronous** option, then the **Clock** information is not displayed.

- **I/O Signals List**: Lists all input (I) and output (O) signals associated with the FSM in an alphabetical order.

The **State** section displays the state information for the **Current State Signal Name**, **Next State Signal Name**, **Initial State**, and **States** list.

- **Initial State**: This text field displays the initial state of the FSM (the state that is entered when during simulation, action passes to this machine). This state is displayed in dark yellow in the state diagram by default.
- **States**: This table displays the state name with its encoding value (for example, `[State Name] = [State Encoding Value]`).

Object Properties

Menu Bar: FSM -> Object Properties

This command opens the *Object Properties* form displaying the properties of the selected state or transition. This form is used for viewing purposes only. It is not possible to edit any values. The following information is displayed on the form:

State Tab

This tab is displayed when a state object is selected.

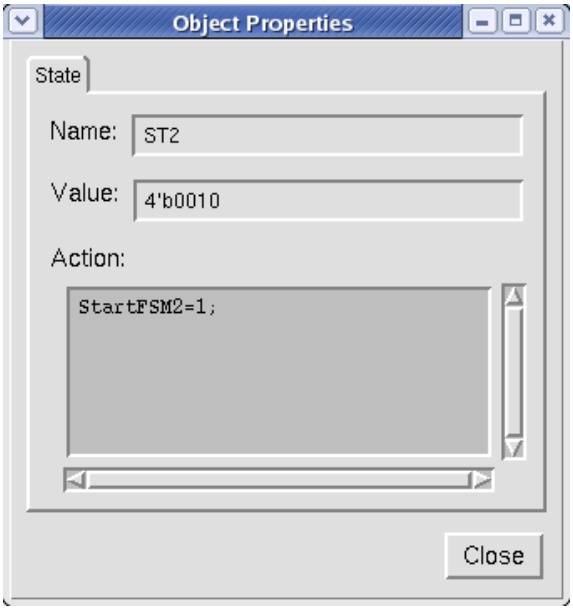


Figure: Object Properties - State Tab

The following fields are available:

- **Name:** Displays the name of the selected state.
- **Value:** Displays the encoding value of the selected state.
- **Action:** Lists the state actions.

Transition Tab

This tab is displayed when a transition object is selected.

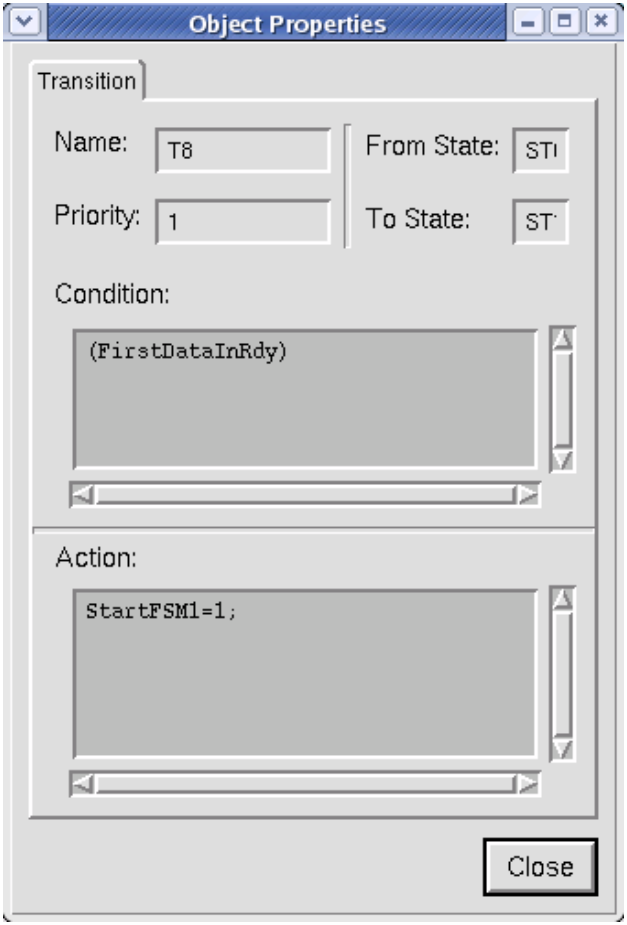


Figure: Object Properties - Transition Tab

The following fields are available:

- **Name:** Displays the selected transition.
- **Priority:** If more than one transition leaves a state, a priority can be assigned to each transition. 1 has the highest priority, followed by 2, 3, and so on. The highest priority is executed first.
- **From State:** Displays the state that the transition is coming from.
- **To State:** Displays the state that the transition is going to.
- **Condition:** Displays the condition statement(s) associated with the selected transition.
- **Action:** Displays the action statement associated with the selected transition.

Analysis Report

Menu Bar: FSM -> Analysis Report

This command opens the *Analysis Report* form where details about the current FSM are summarized.

Load Report File: Click this option after using the Tcl command *fsmAnalyzeFSM* to extract the **Analysis Report** in batch mode. This Tcl command can read the previous **Analysis Report** and use the report to check coverage incrementally. If the **Report Accumulated Statistics** option is turned *on*, then the accumulated report file contains the statistical information for incremental coverage analysis.

Refer to the *Verdi and Siloti Tcl Reference* for details about the Tcl command *fsmAnalyzeFSM*.

Save to File: Click this option to write the **Analysis Report** to the specified file.

Source Code Tab

This tab displays the FSM summary report and the **Detail State Report**.

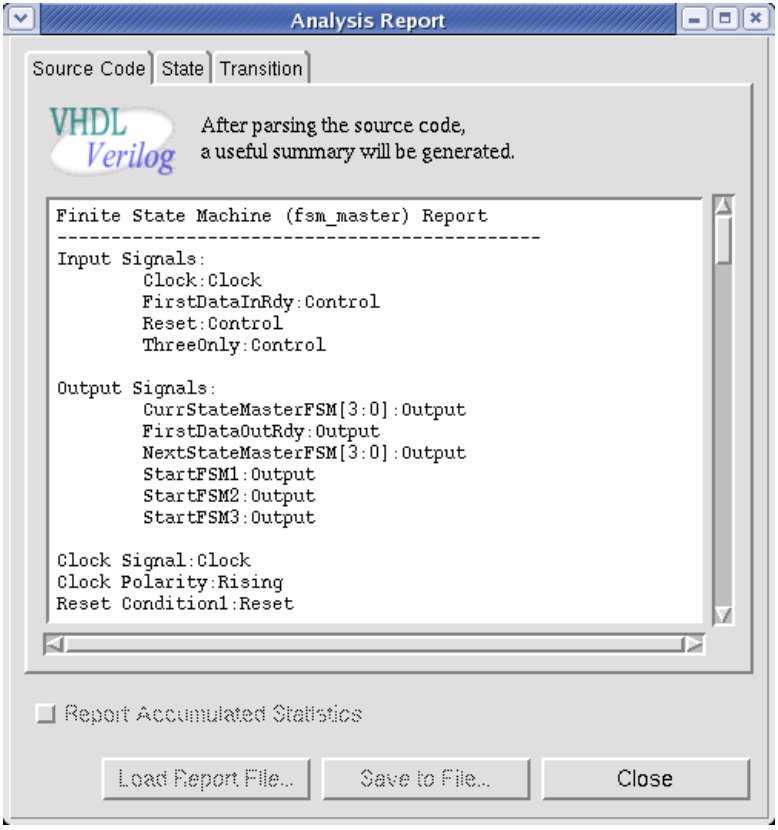


Figure: Analysis Report - Source Code Tab

The FSM summary report displays details about input signals, output signals, the clock signal, the state signal, and the state table. For each state, the **Detail State Report** displays the state name, state type, transition, transition condition, and state outputs. Each state can have multiple state types from the following list:

- **Not Completed:** When a control signal does not have a corresponding transition.
- **Looped:** When a transition appears back into the state.
- **Terminated:** When a state does not have an out transition.
- **None:** When a state does not qualify as any of the above types. This does not appear in the report.

State or Transition Outputs are represented by prefixing them with the pound (#) sign.

State Tab

This tab displays the **State Report** which lists all state names and cycle counts, and displays any untested state in the report.

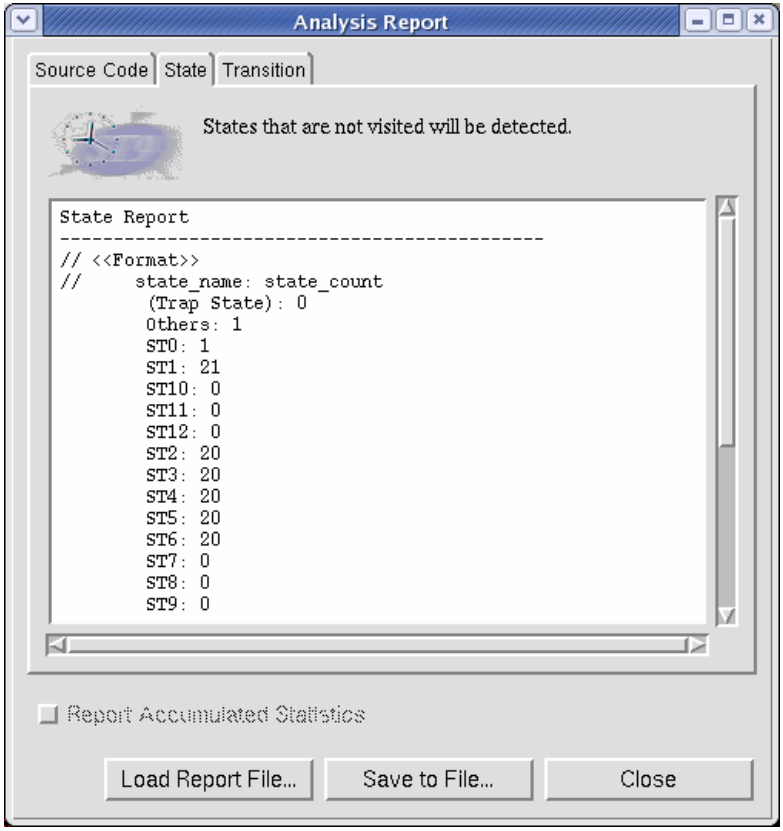


Figure: Analysis Report - State Tab

Transition Tab

This tab displays the **Transition Report** which lists all transitions and cycle counts, and displays any untested transitions.

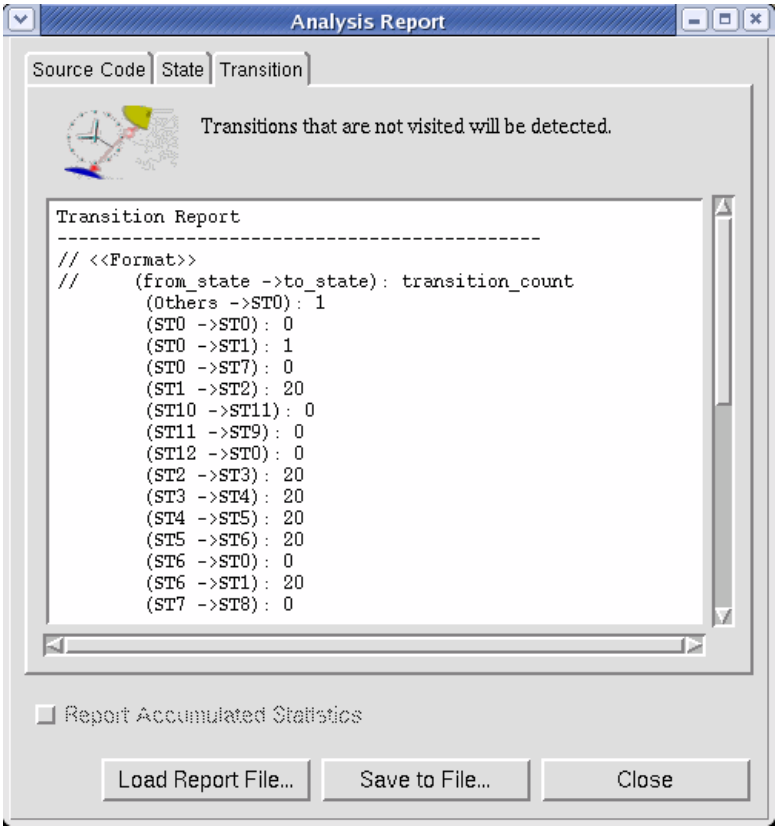


Figure: Analysis Report - Transition Tab

Tools Commands

Duplicate Window

Menu Bar: Tools -> Duplicate Window

This command invokes a new *nState* window which displays the same state diagram displayed in the current *nState* window.

Partial FSM

Menu Bar: Tools -> Partial FSM

This command opens a new *Partial FSM* window (opened as another tab in the current *nState* docked frame) after one or more states is selected.

In the *Partial FSM* window, further modifications can be made to the FSM design: add state(s) by dragging the selected state(s) from a full FSM window, click the **Cut** toolbar icon to remove unnecessary state(s), or double-click a state to expand its connected states (undo this function using the **Undo** toolbar icon).

Preferences

Menu Bar: Tools -> Preferences

The **FSM** page defines the line color, line width, fill, and text options to be used for FSMs. For more information, refer to the *FSM Folder* section in the *Preferences* chapter for details.

Customize Menu/Toolbar

Refer to the **Tools -> Customize Menu/Toolbar** command in the *nTrace* chapter for complete details.

Right-Click Commands

Many of the previously described commands can also be selected from the right-click menu options in *nState*.

nState Frame Right-Click Options

When the right-click option is clicked in the menu, toolbar icon, or frame banner area of the *nState* window or standalone window, a configuration option menu is displayed.This menu can be used to configure the available icons and frames.

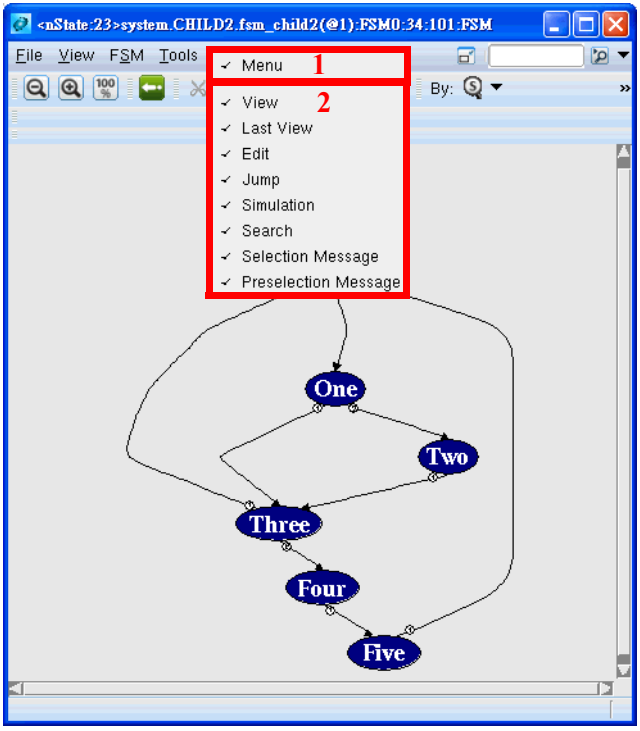


Figure: Configuration Option Menu

The **Menu** option (labeled 1 in the figure above) enables/disables the command menu bar. The options in the below section (labeled 2 in the figure above) enables/disables the toolbar icons for different functions.

General Right-Click Commands

The following commands are available when you right-click in the window except on a state or transition.

Drag

Bind Key: Ctrl+C

Drag the selected state to another frame where it can be dropped. This is similar to using the middle mouse button to select and drag.

Drop

Bind Key: Ctrl+V

Selected signals dragged from another location can be dropped in this window. This is similar to releasing the middle mouse button for a drop.

Zoom All

Bind Key: F

Refer to the **View -> Zoom ->** **Zoom All** command description for details.

Last View

Toolbar Icon: 

Bind Key: L

Refer to the **View ->** **Last View** command description for details.

Edit Search Sequences

Bind Key: Ctrl+S

Refer to the **FSM ->** **Edit Search Sequences** command description for details.

Print

Refer to the **File ->** **Print** command description for details.

Add Signal to Wave

Refer to the **FSM -> Add Signal(s) to Waveform** command description for details.

Machine Properties

Refer to the **FSM -> Machine Properties** command description for details.

State Right-Click Commands

The following commands are available when a state bubble is right-clicked.

Drag

Bind Key: Ctrl+C

Drag the selected state to another frame where it can be dropped. This is similar to using the middle mouse button to select and drag.

Drop

Bind Key: Ctrl+V

Selected signals dragged from another location can be dropped in this window. This is similar to releasing the middle mouse button for a drop.

State Action

Refer to the **View -> State Action** command description for details.

Fit Select Set

Bind Key: Ctrl+F

Refer to the **View -> Zoom -> Fit Select Set** command description for details.

Object Properties

Refer to the **FSM -> Object Properties** command description for details.

Expand

NOTE: The three commands under this menu selection are available in a partial FSM view.

Next States

For the selected state, add the next state(s) to the view. If the state is already displayed, it is highlighted.

Previous States

For the selected state, add the previous state(s) to the view. If the state is already displayed, it is highlighted.

All Adjacent States

For the selected state, add all the next and previous states to the view. If the states are already displayed, is it highlighted.

Transition Right-Click Commands

The following commands are available when a state bubble is right-clicked.

Drag

Bind Key: Ctrl+C

Drag the selected state to another frame where it can be dropped. This is similar to using the middle mouse button to select and drag.

Drop

Bind Key: Ctrl+V

Selected signals dragged from another location can be dropped in this window. This is similar to releasing the middle mouse button for a drop.

Jump to From State

Toolbar Icon: 

This command pans to the beginning state (“from” state) of the selected transition.

Jump to To State

Toolbar Icon: 

This command pans to the end state (“to” state) of the selected transition.

Transition Condition

Refer to the **View -> Transition Condition** command description for details.

Transition Action

Refer to the **View -> Transition Action** command description for details.

Fit Select Set

Bind Key: Ctrl+F

Refer to the **View -> Zoom -> Fit Select Set** command description for details.

Object Properties

Refer to the **FSM -> Object Properties** command description for details.

Toolbar Icons and Fields

Figure: Toolbar Icons Used in nState Window

The available toolbar icons may be modified. Refer to the *Toolbars* section of the *User Interface* chapter in the *Verdi User Guide and Tutorial* manual for details.

The different toolbar categories and icons available are described as follows:

View Category

Zoom Out

Refer to the **View -> Zoom -> Zoom Out** command description for details.

Zoom In

Refer to the **View -> Zoom -> Zoom In** command description for details.

Zoom All

Refer to the **View -> Zoom -> Zoom All** command description for details.

Last View Category

Last View

Refer to the **View -> Last View** command description for details.

Edit Category

Cut

This command removes the selected object and places a copy on the clipboard.

Undo

This command reverses the last action performed, if possible.

Jump Category

Jump to 'From State' 

Click this icon to pan to the beginning state ("from" state) of this transition.

Jump to 'To State' 

Click this icon to pan to the ending state ("to" state) of this transition.

Simulation Category

Previous State 

Refer to the **FSM** -> **Previous State** command description for details.

Next State 

Refer to the **FSM** -> **Next State** command description for details.

Search Category

Search by State  /Search by Sequence 

Click this toolbar icon to select the FSM search criteria: **State** or **Sequence**.

Find State 

Refer to the **FSM** -> **Find State** command description for details.

Search Backward 

NOTE: To enable this icon, the simulation results file must be loaded and the **FSM -> State Animation** command enabled.

Click this toolbar icon to search backwards in the simulation results file to find times when the FSM transitions from the state or sequence specified in the state list on the toolbar.

Search Forward 

NOTE: To enable this icon, the simulation results file must be loaded and the **FSM -> State Animation** command enabled.

Click this toolbar icon to search forwards in the simulation results file to find times when the FSM transitions from the state or sequence specified in the state list on the toolbar.

Cursor Time 

NOTE: To enable this field, the simulation results file must be loaded and the **FSM -> State Animation** command enabled.

The **Cursor Time** field serves two functions:

1. The field displays the cursor time for the state or sequence specified in the state list when the **Search Backward** or **Search Forward** icons are clicked.
2. Type in the cursor time in the field, and the state diagram highlights the related states and transitions at that cursor time.

Selection Message Category

Selection Message

This field displays information about the selected object.

Preselection Message Category

Preselection Message

When the cursor is moved over an object, this field displays information about the pre-selected object.

Flow Views

Overview

A *Flow View* window is displayed when the **Tools -> Temporal Flow View -> New Temporal Flow View** command is invoked from the *nTrace* menu bar (or with a right-click on a signal in the *nWave* waveform pane or the source code frame or selecting the **Temporal Flow View -> New Temporal Flow View** option from the toolbar). The *Flow View* window is docked to the same frame location as the *Message/nWave* window as a new tab.

Click **Undock** icon on the toolbar to make the *Flow View* frame as a standalone window.

The *Flow View* has the following three primary modes:

- **Temporal Flow View**
- **Compact Temporal Flow View**
- **Temporal Register View**

The **Temporal Flow View** mode includes two tracing methods: cycle-based tracing and transition-based tracing. The tracing method can be set under the **Default Trace Method** section of the **Tools** -> **Preferences** -> **Temporal Flow View** folder -> **Trace** folder -> **Trace** page. The main menu bar, illustrated below, is identical for all modes. However, some of the items associated with the drop-down menus differ between modes.

Any open flow views are closed automatically if the primary FSDB file is closed (**File -> Close** in *nWave*), the primary *nWave* window file is closed (**File -> Close Window** in *nWave*), the primary FSDB file is changed (**Window -> Change to Primary** in *nWave*), or the FSDB file is reloaded (**File -> Reload** in *nWave*).

All open flow views are closed automatically if any simulation result is changed, keeping the final tracing result correct.

The menu bar for the *Flow View* window is as follows:

Figure: Temporal Flow View Menu Bar

The drop-down menus are explained in the following sections. Refer to the *Icons for Dockable Panes* section for details on the menu bar icons.

The synchronization buttons on the toolbar of the *Flow View* can be used to synchronize analog signals to the *nWave*, *nTrace*, and *nSchema* windows.

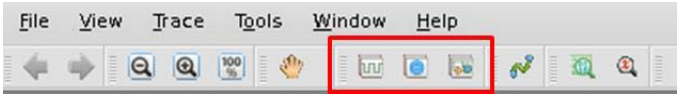


Figure: Toolbar Icons to Synchronize Signals in nWave, nTrace, and nSchema

Refer to the **View -> Enable Show nWave Automatically**, **View -> Enable Show Source Automatically**, and **View -> Enable Flow Schematic Automatically** sections for details.

Signals can be dragged and dropped among to the *nWave*, *nTrace*, and *nSchema* windows. When signals are dragged from *nWave*, *nTrace*, and *nSchema*, the *Flow View* adds the dragged signals as reference signals and displays the respective analog blocks, as illustrated in the following figure:

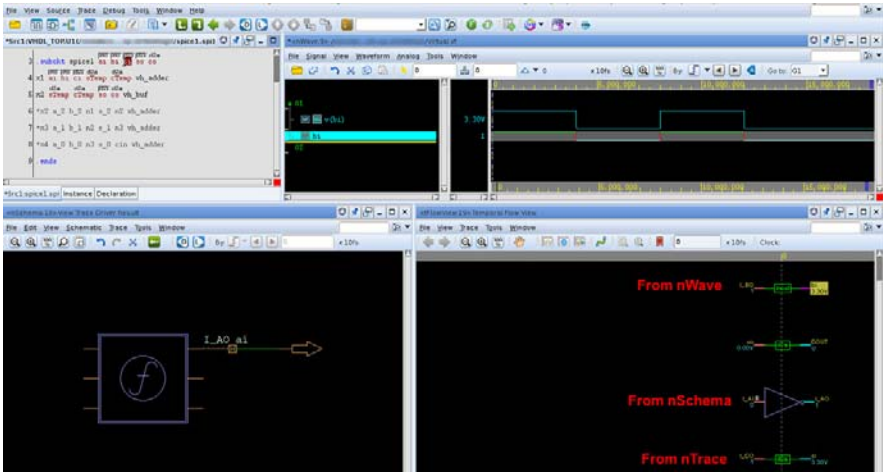


Figure: Signals Dragged from nWave, nTrace, and nSchema Displayed in Flow View

If signals are dragged from the *Flow View* and dropped to the *nWave*, *nTrace*, and *nSchema* windows, drivers and nets of these signals are highlighted in *nTrace* and *nSchema*, respectively. The following figure illustrates the example with the results of the **co** signal that is dragged from the *Flow View* and dropped to the *nWave*, *nTrace*, and *nSchema* windows.

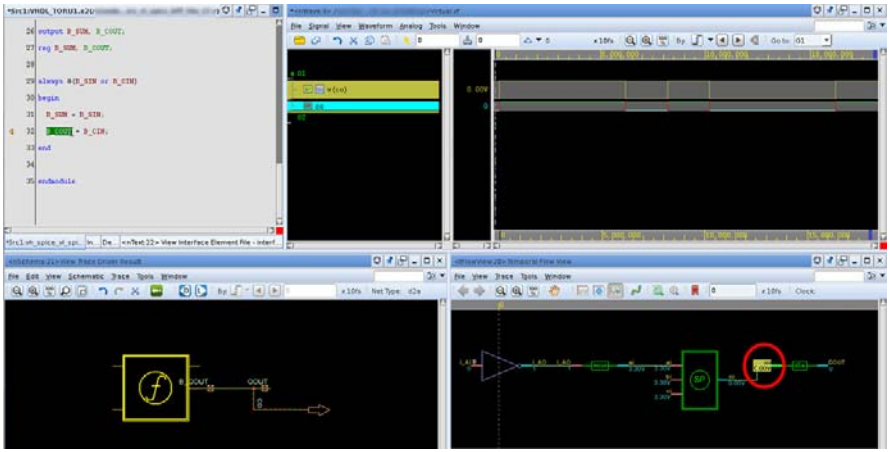


Figure: Signals Dragged from Flow View Displayed in nWave, nTrace, and nSchema

If Interface Elements (IEs) are dragged, the respective entries in IE report are highlighted. The following figure illustrates dragging IEs cell between the **co** and **COUT** signals. Also, the waveform of the signal is inserted into *nWave*.

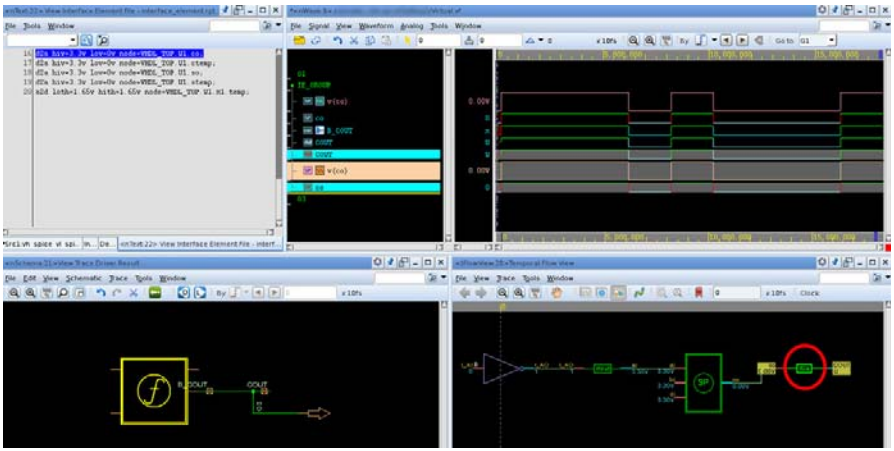


Figure: IE Dragged from Flow View Displayed in nWave, nTrace, and nSchema

Difference between Transition-Based and Cycle-Based

Following are the differences between the transition-based mode and cycle-based mode.

Initial Snap Time

For the transition-based mode, the interval time indicates the latest time of value change. For cycle-based mode, the interval time indicates the latest time of clock cycle change.

Function Logic

In the transition-based mode and cycle-based mode, the active/real drive can be decided.

Trace Selection

The transition-based mode takes the signal that triggers a transition on the output as transition signal, and marks them as green dot in Verdi GUI. The cycle-based mode takes signals that has value change within previous clock cycle of the output fan-outs as transition signals, and marks them as red dot in Verdi GUI.

Qp Port

When tracing a latched signal, transition-based mode does not display the Qp port and it directly snaps to the latest value change point. Cycle-based mode displays the Qp port which connects to its latest value change point. It first snaps to the latest clock cycle, then traces through the Qp port to the latest value change point.

Hold Register

Transition-based mode cannot display the hold register since it does not do clock domain analysis. Cycle-based mode can show the hold register information on the view.

Bus Grouping

Transition-based mode only displays the single bit with transition, for example, L2_2[2].

Cycle-based mode can display entire grouping bus, for example, L2_2[2:0].

Menu Summary

The *Temporal Flow View* menu options are summarized below. All commands can be double-clicked to jump to the corresponding command description. For right-click commands, refer to the *Right-Click Commands* section for details.

File Commands

New Viewer
Load 2nd Waveform for Trace Mismatch
Print
Capture Window
Close

View Commands

Back
Forward
Zoom ->
  Zoom In
  Zoom Out
Zoom In Horizontally
Zoom Out Horizontally
Refresh
Zoom In Vertically
Zoom Out Vertically
Refresh
Clear All Decision Points
Bookmark ->
  Add Bookmark
  Edit Bookmark
Font
Switch to Pointer Mode /
Switch to Pan Mode
Turn On Tip
Signal ->
  Active Nodes Only
  Active Data Nodes Only
  Node With Value 'X' Only
  Triggering Path Only
  Hierarchical Name
  Show Signal Name
  Show Signal Value
  Show Port Name
  Power Cell Name
Enable Show nWave Automatically /
Disable Show nWave Automatically
Enable Show Source Automatically /
Disable Show Source Automatically
Enable Flow Schematic Automatically /
Disable Flow Schematic Automatically
Show Hold Cycle Count
Annotate in Color
Set Window Time Unit
Signal Value Radix ->
  Binary
  Octal
  Hexadecimal
  Decimal
  ASCII
  IEEE-754 Floating Point
  [*Alias name*]
  Add Alias from File
  Add Alias from Program
  Remove Alias
  Edit Alias
Signal Value Notation ->
  Unsigned
  Signed 2's Complement
  Signed 1's Complement
  Signed Magnitude

Trace Commands

Add Reference Signals
Show Active Statement
Show Fan-in Registers
Show Triggering Statement
Trace This Value
Trace This Register
Trace X
Trace Glitch
Trace Triggering Path
Show Active Statement for Partial Bus
Behavior Trace for Waveform Mismatch
Trace Again by Showing Cycle-by-Cycle Details
Show Cycle-by-Cycle Details of Holding Registers
Hide
Show Whole Fan-in Group
Fan-in Display Management
Clock Domain Highlight
Dehighlight All

Tools Commands

Open Flow View->
  Temporal Flow View
  Compact Temporal Flow View
  Temporal Register View
Dump Memory Waveform to FSDB
Show Memory Contents
Show Fan-ins on nSchema ->
  All Fan-ins
  Active Only
Show All Traced Paths on Flow-Schematic
Add to Waveform
Show Selected Signals on nWave
Show Drivers on nWave ->
  All Drivers
  Active Only
Show Clock (Domain) on Waveform
Show Synchronization Signal
Show All Traced Signals on nWave
Preferences
Customize Menu/Toolbar

Bind Keys

For a complete list of bind keys used in the *Temporal Flow View* window, refer to the *Temporal Flow View* section in the *Bind Key Summary* for details.

File Commands

New Viewer

Menu Bar: File -> New Viewer

This command opens a new flow view window with the type as dependent on the context. If the current window is displaying a *Temporal Flow View*, the **File -> New Viewer** command opens a new *Temporal Flow View* window with reference signals only. Similarly, if the current window is displaying a *Compact Temporal Flow View*, the **File -> New Viewer** command opens a new *Compact Temporal Flow View* window with reference signals only. Reference signals are either the original selected signal used to create the flow view or signals that are added with the **Trace ->Add Reference Signals** command.

Load 2nd Waveform for Trace Mismatch

Menu Bar: File -> Load 2nd Waveform for Trace Mismatch

This command opens the second waveform file from the *Load Simulation Results for Trace Mismatch* form. This waveform file is used by the **Behavior Trace for Waveform Mismatch** command.

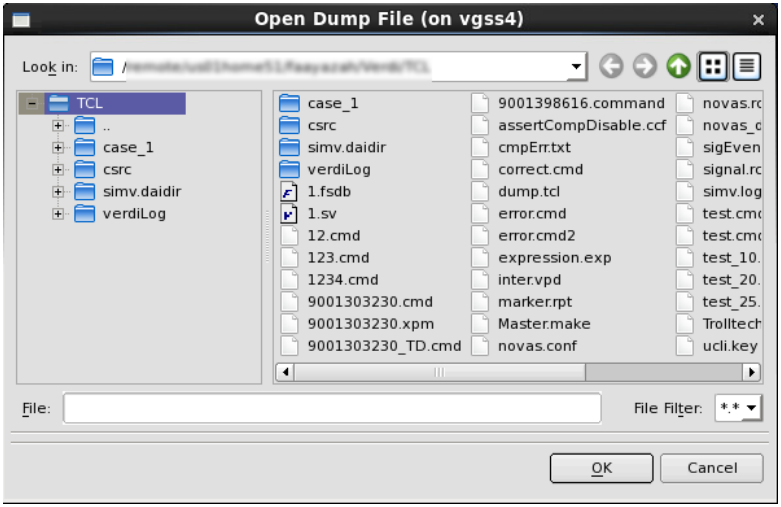


Figure: Load Simulation Results for Trace Mismatch Form

The following file types are converted to Fast Signal Database (FSDB) format, VCD dump file, and ViewSim. When the conversion is performed to FSDB format, write permission and existing file overwrite conditions are checked. Only digital simulation results can be loaded using this form.

Print

Menu Bar: File -> Print

This command opens a *Print* form for printing the current content of the flow view to a printer or file. This includes the time scales and active/inactive indications.

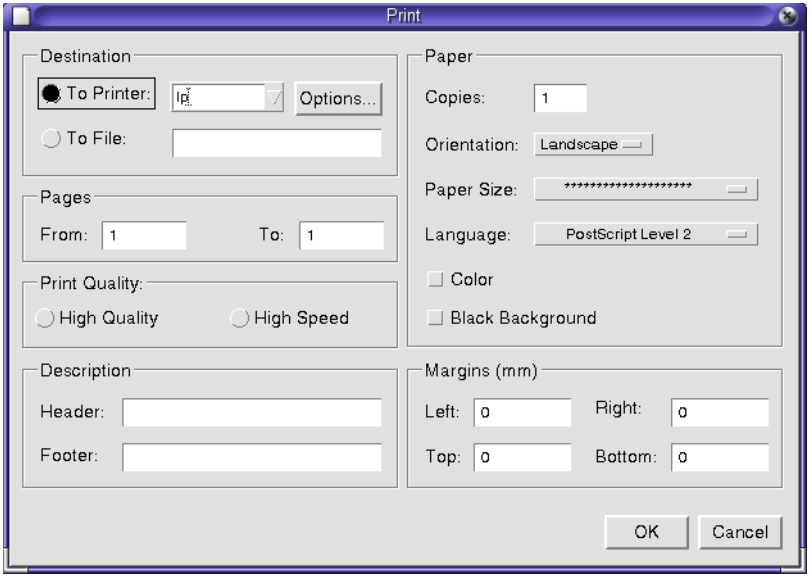


Figure: Print Form

The **Destination** section includes the following options and fields:

- **To Printer:** When this option is turned *on*, the file prints to the printer specified in the associated text field. In the drop-down list next to the text field, select the list of printer names defined using the **Options** button. The *Configuration* form opens, as illustrated in the figure below.
- **To File:** When this option is turned *on*, the file is saved in the working directory with the file name specified in the associated text field.

- **Options:** This button opens a *Configuration* form in which a new **Printer Name** and the printer associated with a set of characteristics can be specified as follows: output format (**Postscript Level 1** or **Postscript Level 2**), **Paper Size** (**Letter, A4, A3, A2, A1, A0, B, C, D, E, User-defined**), and the **Print Command** command line options **Command** (default value is *lpr*), **Destination** (default value is *–P*), and **Copies** (default value is *-#*). If the **User-defined** paper size option is selected, then the **Width** and **Height** must also be specified.

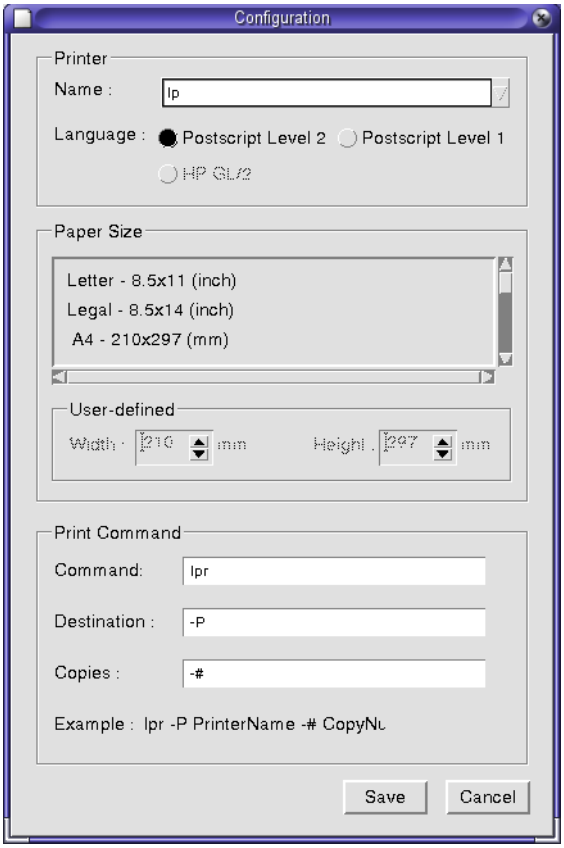


Figure: Configuration Form

The **Pages** section includes the following fields:

- **From:** Specify the first page to print in the associated text field.
- **To:** Specify the last page to print from in the associated text field.

The **Print Quality** section includes the following options:

- **High Quality:** When this option is turned *on*, the print output is of highest quality, but it takes longer to print. The default value for this option is *on*.
- **High Speed:** When this option is turned *on*, the print output is of lower quality, but prints faster.

The **Description** section includes the following fields:

- **Header:** Define the text for the header that appears on the top of the printout. Possible options for the header includes the file name (%f) that is currently being viewed in the flow view window, user name and host name (%h), and the date and time (%t).
- **Footer:** Define the text for the footer that appears at the bottom of the printout. The header options are applicable in the footer also.

The **Paper** section includes the following options and fields:

- **Copies:** Specify the number of copies to be printed in the associated text field. The default value for this option is *1*.
- **Orientation:** Specify the page orientation by selecting either **Landscape** or **Portrait**.
- **Paper Size:** Specify the paper size by selecting one of the following: **Letter, A4, A3, A2, A1, A0, B, C, D, E,** and **User-defined**.
- **Language:** If the **To File** option is selected in the **Destination Section**, then this **Language** option is active and can be selected between **Postscript Level 1** and **Postscript Level 2**.
- **Color:** When this option is turned *on*, the print output is in color (assuming a color printer is in use), otherwise the output is in black-and-white.
- **Black Background:** When this option is turned *on*, the window background is printed in black. When this option is turned *off*, the window background is white.

The **Margins** section includes the following fields:

- **Left:** Enter the required left margin (in mm) in the associated text field. The default value for this option is *0*.
- **Right:** Enter the required right margin (in mm) in the associated text field. The default value for this option is *0*.
- **Top:** Enter the required top margin (in mm) in the associated text field. The default value for this option is *0*.
- **Bottom:** Enter the required bottom margin (in mm) in the associated text field. The default value for this option is *0*.

Capture Window

Menu Bar: File -> Capture Window

This command opens the *Capture Window - Preview* form in which an image in Portable Network Graphic (PNG) format, a GNU standard (similar to GIF) can be the output.

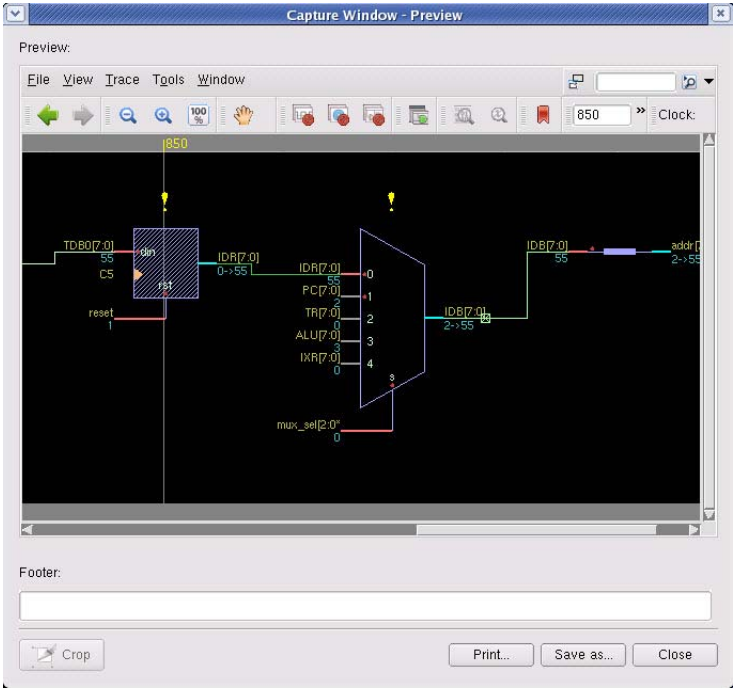


Figure: Capture Window - Preview Form

The following fields and buttons are available in the *Capture Window - Preview* form:

- **Footer:** Enter the appropriate text to be displayed on the footer.
- **Crop:** Drag the left mouse button to select the desired area and then click **Crop** to update the display in the *Capture Window - Preview* form.
- **Print:** This button opens a *Print* form in which the desired print options for creating a hard copy can be specified.
- **Save as:** This button opens a *Save As* form where the directory structure can be viewed and a file name for the PNG file can be specified.

Close

Menu Bar: File -> Close

This command closes the active flow view window (the one in which the selected drop-down menu is associated).

NOTE: The **Compact Temporal Flow View** window and the C**lassic Transaction** windows (Evaluator and Analysis Window) do not support global font.

View Commands

Back

Menu Bar: View -> Back

Toolbar Icon 

Bind Key: L

This command causes the main display area to regress to its previous view/zoom level, so that it only becomes active after changes are made to the view/zoom level. The command can be used multiple times to cycle back through a series of earlier view/zoom levels.

Refer to the corresponding **View -> Forward** command for details.

Forward

Menu Bar: View -> Forward

Toolbar Icon 

Bind Key: N

This command causes the main display area to proceed to the next view/zoom level, so that it only becomes active after the **View -> Back** command is used. The command can be used multiple times to cycle forward through a series of view/zoom levels.

NOTE: One or more **View -> Back** commands must be executed before a similar number of **View -> Forward** commands can be executed.

Refer to the corresponding **View -> Back** command for details.

Zoom

Zoom In

Menu Bar: View -> Zoom -> Zoom In

Toolbar Icon: 

Bind Key: Shift+Z

NOTE: This command is available in the *Temporal Flow View*.

This command provides a close view of the contents in the main display area. The magnification is changed to half the magnification of the previous view.

Additionally, a zoom in can be performed by dragging-left along the zoom-scale ruler (at the top of the display area) or along the full-scale ruler at the bottom of the display.

Refer to the **View -> Zoom -> Zoom Out** and **View -> Fit Time** commands for details.

Zoom Out

Menu Bar: View -> Zoom -> Zoom Out

Toolbar Icon: 

Bind Key: Z

NOTE: This command is available in the *Temporal Flow View*.

This command enables more content to be viewed in the main display area at a reduced size. The magnification is changed to two times the magnification of the previous view.

Additionally, a zoom out can be performed by dragging-left along the gray area at the bottom of the display.

Refer to the **View -> Zoom -> Zoom In** and **View -> Fit Time** commands for details.

Fit Time

Menu Bar: View -> Fit Time

Toolbar Icon: 

Bind Key: F

This command causes the entire flow view to be displayed in the main display area.

Zoom In Horizontally

Menu Bar: View -> Zoom -> Zoom In Horizontally

Toolbar Icon: 

NOTE: This command is available in the *Compact Temporal Flow View* and *Temporal Register View*.

This command provides a close view of the contents in the horizontal direction in the main display area. The magnification is changed to half the magnification of the previous view.

Additionally, a zoom in can be performed by dragging-left along the zoom-scale ruler (at the top of the display area) or along the full-scale ruler at the bottom of the display.

Refer to the **View -> Zoom Out Horizontally** and **View -> Fit Time** commands for details.

Zoom Out Horizontally

Menu Bar: View -> Zoom -> Zoom Out Horizontally

Toolbar Icon: 

NOTE: This command is available in the *Compact Temporal Flow View* and *Temporal Register View*.

This command enables more contents to be viewed in the horizontal direction in the main display area at a reduced size. The magnification is changed to two times the magnification of the previous view.

Additionally, a zoom out can be performed by dragging-left along the gray area at the bottom of the display.

Refer to the **View -> Zoom In Horizontally** and **View -> Fit Time** commands for details.

Zoom In Vertically

Menu Bar: View -> Zoom -> Zoom In Vertically

Toolbar Icon: 

NOTE: This command is available in the *Compact Temporal Flow View* and *Temporal Register View*.

This command increases the vertical spacing between nodes in a flow view, making the display easier to read.

Refer to the **View -> Zoom Out Vertically** command for details.

Zoom Out Vertically

Menu Bar: View -> Zoom -> Zoom Out Vertically

Toolbar Icon: 

NOTE: This command is available in the *Compact Temporal Flow View* and *Temporal Register View*.

This command decreases the vertical spacing between nodes in a flow view, fitting more nodes in the display.

Refer to the **View -> Zoom In Vertically** command for details.

Refresh

Menu Bar: View -> Refresh

This command causes the display area to be refreshed and removes any artifacts that are caused by manipulating the flow view.

Clear All Decision Points

Menu Bar: View -> Clear All Decision Points

This command removes all decision point markers (yellow exclamation points) from the flow view display.

Bookmark

Bookmarks return the flow view display to the bookmarked area by clicking **Go To** in the *Bookmark Editor* form (through the **View -> Bookmark -> Edit Bookmark** command) or using the bookmark cache.

Add Bookmark

Menu Bar:	View -> Bookmark -> Add Bookmark
Toolbar Icon:	
Bind Key:	Ctrl+F2
Mouse Action:	Right-click

This command sets a bookmark for the selected signal or group node in the main display area of the flow view. If the **Always Use System-given Name When Adding a New Bookmark** option is turned *on*, the bookmark is added to the list. When this option is turned *off*, a form opens for specifying the bookmark label. Click **OK** to accept the name and **Cancel** to reject the bookmark addition.

Edit Bookmark

Menu Bar:	View -> Bookmark -> Edit Bookmark
Bind Key:	Ctrl+F3
Mouse Action:	Right-click

This command opens the *Edit Bookmark* form in which the bookmarks can be selected or changed and the bookmark options can be set. Two main tabs are available as follows:

Bookmarks Tab

The **Bookmarks** tab lists all bookmarks for the current flow view window in the main display area. The width of each column can be adjusted by placing the cursor over the vertical bar in the heading row and then pressing and holding the left mouse button.

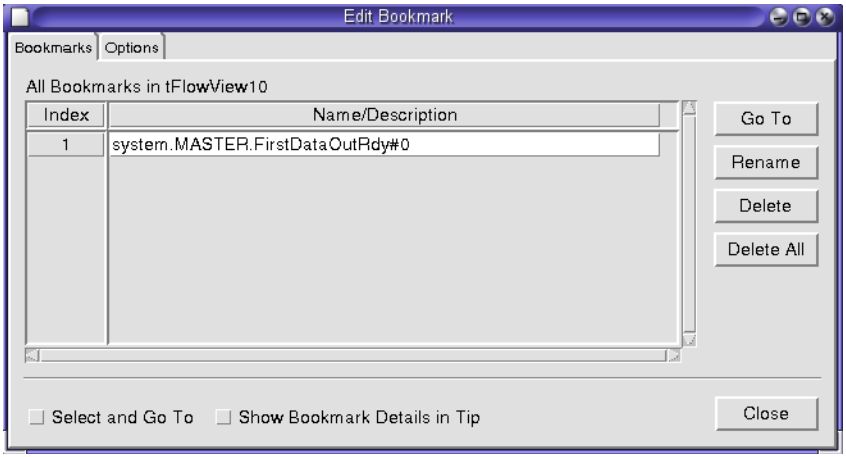


Figure: Edit Bookmark - Bookmarks Tab

A bookmark row may be selected with a left-click anywhere in the row. Additional rows may be selected by holding the **Shift** key (selects a range) or the **Ctrl** key (selects individual rows) and a left-click. Selected rows can be de-selected with a left-click. The following columns are available:

- **Index:** A number identifier for the bookmark. This number can be used with the **Ctrl** key to quickly jump to the associated bookmark.
- **Name/Description:** A label for the bookmark. When the **Always Use System-given Name When Adding a New Bookmark** option is turned *on*, the label defaults to the complete hierarchical path of the signal at the current time. When this option is turned *off*, a user-specified label can be entered for the bookmark.

The *Edit Bookmark* form includes the following options and buttons:

- **Go To:** This button returns the flow view display to the bookmarked area associated with the selected bookmark. If multiple rows are selected, the view changes based on the row with the smallest index.
- **Rename:** This button invokes a form where the description for the selected bookmark can be changed. If multiple rows are selected, the row with the smallest index is opened for editing.
- **Delete:** This button removes the selected bookmark rows from the list.
- **Delete All:** This button removes all bookmarks.
- **Close:** This button closes the *Bookmark Editor* form.
- **Select and Go To:** When this option is turned *on*, the flow view display is automatically returned to the bookmarked area when a bookmark is

selected. When this option is turned *off*, the flow view display is not changed on selecting a bookmark. The default value for this option is *off*.

- **Show Bookmark Details in Tip:** When this option is turned *on*, a tip opens to display the complete hierarchical path for the bookmark when the cursor is moved over the row in the **Name/Description** column. The details of renamed bookmarks including the type (reference/fan-out or fan-in) are displayed. When this option is turned *off*, a tip does not open. The default value for this option is *off*.

Options Tab

The **Options** tab allows the bookmark options to be changed.

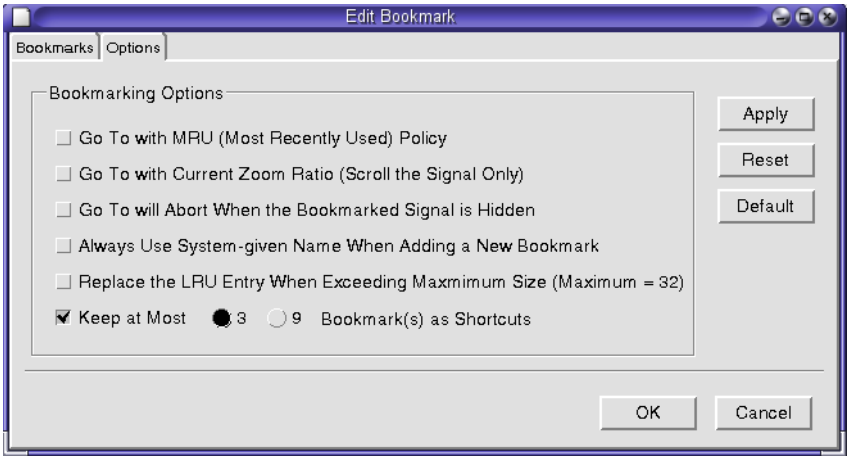


Figure: Edit Bookmark Form - Options Tab

The following Bookmark **Options** are available:

- **Go To with MRU (Most Recently Used) Policy:** This option updates the index order based on the most recently selected bookmark. When this option is turned *on*, the last selected bookmark becomes index 1 and the remaining bookmarks shift down one row. When this option is turned *off*, the bookmarks maintain the original selection order (last selected is the first row, first selected is last row). The default value for this option is *off*.
- **Go To with Current Zoom Ratio (Scroll the Signal Only):** This option affects the zoom ratio in the flow view. When this option is turned *on*, the current zoom ratio is maintained regardless of the zoom ratio saved with the bookmark. When this option is turned *off*, the zoom ratio is changed to match that saved with the bookmark. The default value for this option is *off*.

- **Go To will Abort When the Bookmarked Signal is Hidden:** This option determines whether to go to a bookmark associated with a hidden node. When this option is turned *on*, the flow view display is not updated to display hidden bookmarks. When this option is turned *off*, the flow view display is updated to expand hidden nodes and the associated bookmark. The default value for this option is *off*.
- **Always Use System-given Name When Adding a New Bookmark:** Use this option to name the bookmarks at creation. When this option is turned *on*, the name defaults to the complete hierarchical path and time for the node. When this option is turned *off*, the bookmark name can be changed through a form. The default value for this option is *off*.
- **Replace the LRU Entry When Exceeding Maximum size (Maximum =32):** The maximum number of bookmarks that can be created is 32. This option determines which bookmark to 'drop' when the maximum is exceeded. When this option is turned *on*, the least recently used entry is automatically replaced. When this option is turned *off* and the maximum value is reached, a message window opens and another bookmark cannot be added until an existing bookmark is deleted. The default value for this option is *off*.
- **Keep at Most 3 or 9 Bookmark(s) as Shortcuts:** This option specifies how many bookmarks can be saved as shortcuts. When this option is turned *on*, either 3 or 9 bookmarks are saved in the bookmark cache. When this option is turned *off*, bookmarks are not saved in the cache. The default value for this option is *on* with 3. The order in the bookmark cache is determined by the **Go To with MRU (Most Recently Used) Policy** option.
- **Apply:** This button applies the selected options without closing the *Edit Bookmark* form.
- **Reset**: This button undoes the last changes without closing the *Edit Bookmark* form.
- **Default**: This button changes all options to their default settings.

Font

Menu Bar: View -> Font

This command provides access to a list of alternative fonts (point size) options. Selecting one of these options causes all the text items in the main display area to be updated using the new font. The available fonts are **Auto Scaled Font**, **HELVETICA 8**, **HELVETICA 10**, **HELVETICA 12**, and **HELVETICA14**. The default option is **Auto Scaled Font**.

Switch to Pointer Mode

Menu Bar: View -> Switch to Pointer Mode

Toolbar Icon:  This icon indicates the current mode is pan.

When this command is selected, the mouse cursor provides standard pointer functionality in the flow view's main display area. Dragging-left in an area zooms in the area. Dragging-right selects any object in the area.

NOTE: This is a switch command. After the **View -> Switch to Pointer Mode** command is activated, the next time the **View** drop-down menu is accessed, this command is replaced with its **View -> Switch to Pan Mode** counterpart.

Switch to Pan Mode

Menu Bar: View -> Switch to Pan Mode

Toolbar Icon:  This icon indicates the current mode is pointer.

When this command is selected, the mouse cursor provides pan mode functionality in a flow view's main display area (it switches to a pointer when over a node or a symbol in the main display area). The term “pan mode” means, when you drag-left in the main display area it pans the display up, down, left, and right. Dragging right selects any objects in the area.

NOTE: This is a switch command. After the **View -> Switch to Pan Mode** command is activated, the next time the **View** drop-down menu is accessed and this command is replaced with its **View -> Switch to Pointer Mode** counterpart.

Turn On Tip

Menu Bar: View -> Turn On Tip

When this toggle command is checked, a tip is displayed when the mouse cursor moves over any node in the flow view window. The tips available for display are turned *on* and *off* in the **Tools -> Preferences -> Temporal Flow View** folder **-> View** folder **-> Display** page.

Signal

Active Nodes Only

Menu Bar: View -> Signal ->Active Nodes Only

This command toggles between displaying all nodes that have any relationship to the signal under evaluation and only the nodes (data and control) that are actively contributing to the current value of the signal. This command is automatically enabled when the **Active Data Nodes Only** command is selected.

Active Data Nodes Only

Menu Bar: View -> Signal ->Active Data Nodes Only

This command toggles between displaying all of the nodes that have any relationship to the signal under evaluation and only the data nodes that are actively contributing to the current value of the signal. When this command is selected, the **Active Nodes Only** command is also automatically enabled.

Node With Value 'X' Only

Menu Bar: View -> Signal -> Node With Value `X' Only

This command toggles between displaying nodes with all values (both "good" values and unknown 'X') and only the nodes with 'X' values.

Triggering Path Only

Menu Bar: View -> Signal -> Triggering Path Only

This command toggles between displaying all of the nodes that have any relationship to the signal under evaluation and only the nodes that are actively contributing to the transition of the signal.

Hierarchical Name

Menu Bar: View -> Signal -> Hierarchical Name

This command toggles between displaying only the local names of nodes in the main display area and appending the complete hierarchical names. This command only affects the current window.

The default is to display only the local name; for example, *ACC* (refer the demo circuit used in the *Verdi User Guide and Tutorial* manual). Activating the **View -> Hierarchical Name** command appends the complete hierarchical name, for example, *ACC(tb_CPUsystem.i_CPUsystem.i_CPU)*.

Show Signal Name

Menu Bar: View -> Signal -> Show Signal Name

NOTE: This command is available in the *Temporal Flow View*.

When this toggle command is *on*, the signal names for all nodes are displayed. When this toggle command is *off*, the signal names are not displayed. This command only affects the current window. The default value for this option is *on*.

Show Signal Value

Menu Bar: View -> Signal -> Show Signal Value

NOTE: This command is available in the *Temporal Flow View*.

When this toggle command is *on*, the signal values for all nodes are displayed. When this toggle command is *off*, the signal values are displayed. This command only affects the current window. The default value for this option is *on*.

Show Port Name

Menu Bar: View -> Signal -> Show Port Name

NOTE: This command is available in the *Temporal Flow View*.

When this toggle command is *on*, the symbol input port names are displayed. When this toggle command is *off*, the symbol input port names are not displayed. The default value for this option is *on*.

For RTL level designs, the port names are displayed based on function matching. It is possible that some symbols, such as function blocks, do not have port names. For gate-level designs, the port names are displayed based on either the simulation library or symbol library definitions, depending on the selected source for the symbols. This command only affects the current window.

Power Cell Name

Menu Bar: View -> Signal -> Power Cell Name

NOTE: This command is available when a UPF file is loaded.

When this toggle command is *on*, the power cell names are displayed. When this toggle command is *off*, the power cell names are not displayed. The default value for this option is *off*.

Supported cells include supply set cell, power switch cell, ack port cell, power port cell, Isolation cell, and Retention cell.

Enable Show nWave Automatically

Menu Bar: View -> Enable Show nWave Automatically

Toolbar Icon:  This icon indicates the current mode is disabled.

When this command is selected, the signal for the selected node is automatically displayed and highlighted in the waveform. If a logic symbol in the *Temporal Flow View* is selected, only the output signal is highlighted in the waveform.

NOTE: This is a switch command. After the **View -> Enable Show nWave Automatically** command is activated, the next time the **View** drop-down menu is accessed, this command is replaced with its **View -> Disable Show nWave Automatically** counterpart.

The node selection depends on the current setting for **Node Selection** option under the **Tools -> Preferences -> Temporal Flow View** folder **-> Miscellaneous Page**. The **Mouse Click** (the default) option indicates that a node must be selected in any flow view or a logic symbol in the *Temporal Flow View* to display the waveform associated with that node. By comparison, the **Automatically Select When Mouse over** option displays the waveform when the cursor is moved over a node in any flow view or a logic symbol in the *Temporal Flow View*.

Disable Show nWave Automatically

Menu Bar: View -> Disable Show nWave Automatically

Toolbar Icon:  This icon indicates the current mode is enabled.

When this command is selected, the output signal is not displayed and highlighted in the waveform for the selected node or symbol.

NOTE: This is a switch command. After the **View -> Disable Show Source Automatically** command is activated, the next time the **View** drop-down menu is accessed, this command is replaced with its **View -> Enable Show Source Automatically** counterpart.

Enable Show Source Automatically

Menu Bar: View -> Enable Show Source Automatically

Toolbar Icon:  This icon indicates the current mode is disabled.

When this command is selected, the source code line for the selected node is automatically highlighted in the source code frame. If a logic symbol in the *Temporal Flow View* is selected, the source code for the entire statement is highlighted in the source code frame; however, selecting the output port of a logic symbol highlights the source code of the driving statement.

NOTE: This is a switch command. After the **View -> Enable Show Source Automatically** command is activated, the next time the **View** drop-down menu is accessed, this command is replaced with its **View -> Disable Show Source Automatically** counterpart.

The node selection depends on the current setting for **Node Selection** option under the **Tools -> Preferences -> Temporal Flow View** folder -> **Miscellaneous Page**. The **Mouse Click** (the default) option indicates that a node must be selected in any flow view or a logic symbol in the *Temporal Flow View* to display the source code associated with that node. By comparison, the **Automatically Select When Mouse over** option displays the source code when the cursor is moved over a node in any flow view or a logic symbol in the *Temporal Flow View*.

The window in which the source code is displayed depends on the current setting for the **Show Source Code Automatically on** option under the **Tools -> Preferences -> Temporal Flow View** folder -> **Automatic Command Page**.

Select the **Show Source Code Automatically on New nTrace Window** option to open a new source code tab with the selected nodes highlighted. The **Show Source Code Automatically on Existing nTrace Window** option highlights the source code for the selected node in the current source code frame. In the source code frame, any portions of the source code that are causing an assignment at this time are highlighted.

Disable Show Source Automatically

Menu Bar: View -> Disable Show Source Automatically

Toolbar Icon:  This icon indicates the current mode is enabled.

When this command is selected, the associated code is not displayed in the source code frame for the selected node or symbol.

NOTE: This is a switch command. After the **View -> Disable Show Source Automatically** command is activated, the next time the **View** drop-down menu is accessed, this command is replaced with its **View -> Enable Show Source Automatically** counterpart.

Enable Flow Schematic Automatically

Menu Bar: View -> Enable Flow Schematic Automatically

Toolbar Icon:  This icon indicates the current mode is disabled

When this command is selected, the corresponding instances or wires of the selected node in a flow view are drawn in an *nSchema* window with the flow displayed. The first time a node is left-clicked in the flow view, the corresponding ‘symbol’ is added to *nSchema*; additional selections of the same node highlights the logic in the schematic window.

NOTE: This is a switch command. After the **View -> Enable Flow Schematic Automatically** command is activated, the next time the **View** drop-down menu is accessed, this command is replaced with its **View -> Disable Flow Schematic Automatically** counterpart.

In the *Temporal Register View*, left-click on a node to highlight the corresponding register in *nSchema*. Left-click on a fan-in node to highlight the trace from the fan-in register to the input of the output register.

In the *Temporal Flow View* or *Compact Temporal Flow View*, left-click on a node or fan-in node to highlight the corresponding instance represented by the statement.

Disable Flow Schematic Automatically

Menu Bar: View -> Disable Flow Schematic Automatically

Toolbar Icon:  This icon indicates the current mode is enabled.

When this command is selected, the corresponding instances or wires of the selected node are not displayed in an *nSchema* window.

NOTE: This is a switch command. After the **View -> Disable Flow Schematic Automatically** command is activated, the next time the **View** drop-down menu is accessed, this command is replaced with its **View -> Enable Flow Schematic Automatically** counterpart.

Show Hold Cycle Count

Menu Bar: View -> Show Hold Cycle Count

When this toggle command is *on*, the number of cycles the output of a register is holding steady (that is, no write action; write enable is off) is displayed on the right of the register node. This command only affects the current window. The default value for this option is *off*.

ViewMap

Menu Bar: View -> ViewMap

NOTE: This command is available only in the *Compact Temporal Flow View* and *Temporal Register View*.

When this toggle command is *on*, a small window appears in the bottom right-corner of the main display area. This window displays a thumbnail representation of the entire display area. Drag-left in this thumbnail representation to pan the main display area up-down and left-right (in the latter case the zoom-scale ruler is automatically updated to reflect the new display area). When this toggle command is *off*, the thumbnail window disappears. The default value for this option is *off*.

Annotate in Color

Menu Bar: View -> Annotate in Color

This toggle command turns the annotation coloring *on* or *off* in a flow view window. This command only affects the current window.

The definitions of the value types are as follows:

- 1: logic high
- 0: logic low
- z: high impedance
- x: unknown

To change the color associated with annotations, invoke the **Tools -> Preferences -> Temporal Flow View** folder -> **View** folder -> **View Page**, then select **Item** from the **Color** section. The **Type** drop-down list contains a list of objects in which the color and line style can be specified, including the annotation line coloring for *1*, *0*, *x*, and *z*.

Color Instances by Scope

Menu Bar: View -> Color Instances by Scope

This command opens the *Color Instances by Scope* form where instance colors can be customized by scope in the *Temporal Flow View* window. All scopes are displayed in the **Scopes** table of the *Color Instances by Scope* form. After selecting the preferred scope, change the color for any of the listed scopes.

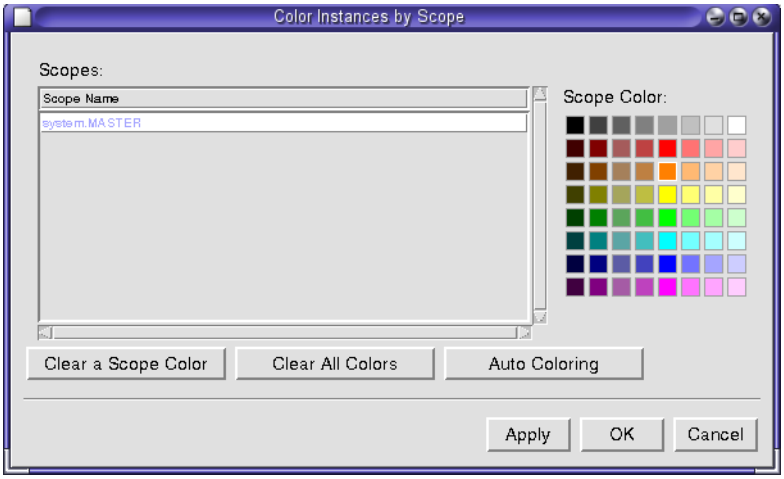


Figure: Color Instances by Scope Form

The *Color Instances by Scope* form includes the following buttons and fields:

- **Scopes:** This table displays all scopes traversed in the current *Temporal Flow View* window.
- **Scope Color:** Specify the desired color for each scope in this color plate.
- **Clear a Scope Color:** Remove the specified color from the selected scope, returning the scope to it default color. The change does not take effect until you click **Apply** or **OK**.
- **Clear All Colors:** Remove the specified color from all scopes, returning the scopes to their default color. The change does not take effect until you click **Apply** or **OK**.
- **Auto Coloring:** Assign a different color to each scope automatically. 16 colors are available (ID_GRAY3, ID_RED3, ID_ORANGE3, ID_YELLOW3, ID_GREEN3, ID_CYAN3, ID_BLUE2, ID_PURPLE2, ID_GRAY6, ID_RED6, ID_ORANGE6, ID_YELLOW6, ID_GREEN6, ID_CYAN6, ID_BLUE6, ID_PURPLE6, ID_UNDEFINED_COLOR) for automatic assignment. If more than 16 scopes exist, the colors are repeated. Any previously assigned colors are overwritten by clicking this button. The change do not take effect until you click **Apply** or **OK**.

Set Window Time Unit

Menu Bar: View -> Set Window Time Unit

This command opens the *Set Window Time Unit* form. In this form, specify the default time unit and the time base (options are **s, ms, us, ns, ps, fs**) to use in the display area and on the time ruler toolbar. This command affects all open flow views.

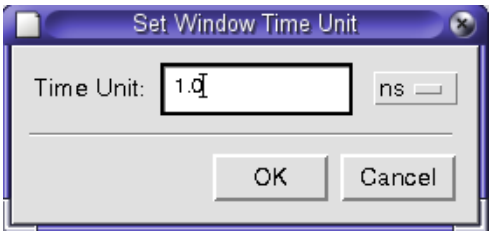


Figure: Set Window Time Unit Form

Signal Value Radix

For all sub-commands under the **Signal Value Radix** command, the target signals in the main display area must be preselected before the command is invoked. The selected signal must be a bus signal. This command affects all open flow views.

NOTE: If the format of one instance of a signal is changed, all other instances of that signal in all windows automatically accept the same format.

Binary

Menu Bar: View -> Signal Value Radix -> Binary

This command displays the selected signal values in **Binary** format.

Octal

Menu Bar: View -> Signal Value Radix -> Octal

This command displays the selected signal values in **Octal** format.

Hexadecimal

Menu Bar: View -> Signal Value Radix -> Hexadecimal

This command displays the selected signal values in **Hexadecimal** format. This option is set as the default value the first time the flow view window is opened.

Decimal

Menu Bar: View -> Signal Value Radix -> Decimal

This command displays the selected signal values in **Decimal** format.

ASCII

Menu Bar: View -> Signal Value Radix -> ASCII

This command displays the selected signal values in **ASCII** format, that is, each byte is treated as an ASCII code and displayed as its corresponding ASCII character.

Enumerated Literal

Menu Bar: View -> Signal Value Radix -> Enumerated Literal

This command displays the selected signal values in **Enumerated Literal** format.

IEEE-754 Floating Point

Menu Bar: View -> Signal Value Radix -> IEEE-754 Floating Point

This command displays the selected signal values in **IEEE-754 Floating Point** format.

Add Alias from File

Menu Bar: View -> Signal Value Radix -> Add Alias from File

Refer to the **Waveform -> Signal Value Radix -> Add Alias From File** command in the *nWave* chapter for details.

Add Alias from Program

Menu Bar: View -> Signal Value Radix -> Add Alias from Program

This command opens a program that accepts a number (or a file containing a number) and prints the alias string (or list of strings) corresponding to that number.

When this command is invoked, the flow view sends the value of signals to the alias program and obtains the corresponding results from the alias program's output. For each value from the flow view, the alias program should write its corresponding output to *std output* (the working display/screen).

For example, alias programs written in C and Perl can be found in the *nWave* chapter of this manual under the **Waveform -> Signal Value Radix** -> **Add Alias From Program** command. Refer to the C example for details.

Remove Alias

Menu Bar: View -> Signal Value Radix -> Remove Alias

This command removes any aliases from the selected signals and reverts to displays their numeric values.

Edit Alias

Menu Bar: View -> Signal Value Radix -> Edit Alias

Refer to the **Waveform -> Signal Value Radix** -> **Edit Alias** command in the *nWave* chapter for details.

Signal Value Notation

Before this command is invoked, one or more target signals in the main display area of the flow view must be preselected first. This command affects all open flow views. When this command is invoked, a number of options (as listed below) are displayed.

Unsigned

Menu Bar: View -> Signal Value Notation -> Unsigned

This command displays the selected signal values in an **Unsigned** format.

Signed 2's Complement

Menu Bar: View -> Signal Value Notation -> Signed 2's Complement

This command displays the selected signal values in **Signed 2's Complement** format. Refer to the **Signed 2's Complement** command description in the *nWave* chapter for the table of signed binary numbers.

Signed 1's Complement

Menu Bar: View -> Signal Value Notation -> Signed 1's Complement

This command displays the selected signal values in **Signed 1's Complement** format. Refer to the **Signed 2's Complement** command description in the *nWave* chapter for the table of signed binary numbers.

Signed Magnitude

Menu Bar: View -> Signal Value Notation -> Signed Magnitude

This command displays the selected signal values in **Signed Magnitude** format. Refer to the **Signed 2's Complement** command description in the *nWave* chapter for the table of signed binary numbers.

Trace Commands

Add Reference Signals

Menu Bar: Trace -> Add Reference Signals

This command opens the *Add Reference Signals* form in which additional reference signals can be added. The recommendation is to select a signal before invoking this command.

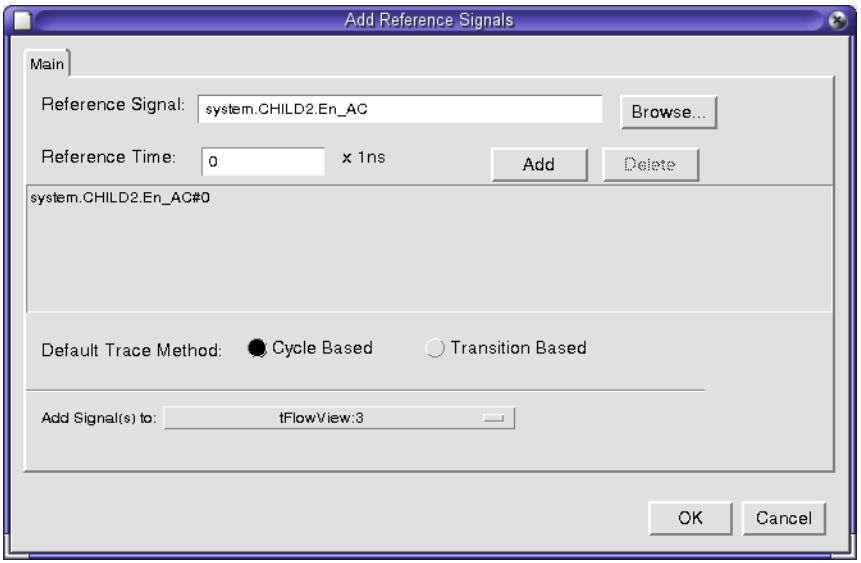


Figure: Add Reference Signals Form

The *Add Reference Signals* form includes the following buttons and fields:

The central list area of the form contains one or more **Reference Signal / Reference Time** pairs as specified. All signals in this section are added to the selected flow view.

- **Reference Signal:** After the reference signal is specified, it becomes the initial output node for tracing in the flow view. When the **Add Reference Signals** command is invoked on a selected node/signal, the complete hierarchical path to the selected signal is entered in the text field.

  To change the reference signal, enter the name of the new reference signal or click **Browse** to open the *Signal Manager* form in which the design

hierarchy can be browsed to select a signal. A white highlighted rectangle indicates that the design scope is selected and the scope's signals are currently displayed in the signal list. The selected scope's inputs, outputs, and I/O's signals are color-coded and identified with a symbol. Internal nets are displayed in black.

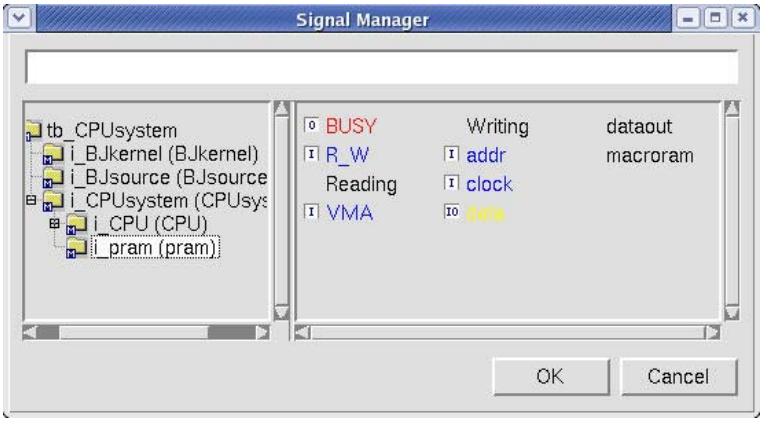


Figure: Signal Manager Form

The left-hand side of *Signal Manager* form contains a hierarchy browser pane that works in a similar manner to the *nTrace* design browser frame. Click the plus (+) symbol associated with a block instance name to reveal its sub-blocks, then left-click a block instance name to display the signals associated with that block in the form's right-side window.

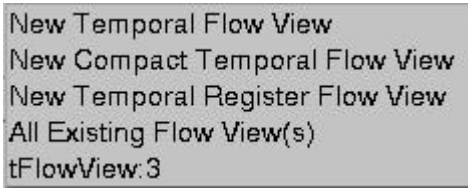
To select the signal, left-click a signal of interest in the right-side window to select it. After selecting the signal, the complete hierarchical path and signal name are entered in the text field. Click **OK** to save the signal and close the form or click **Cancel** to exit the window without saving the signal. Double-click on a signal in the right-side window to select the signal and close the form. The selected signal is entered in the **Reference Signal** text field of the *Add Reference Signals* form.

- **Reference Time:** This option defaults to the current cursor time. Enter the time to be associated with the **Reference Signal** as discussed in the previous section.
- **Default Trace Method:** Specify the default trace method by selecting either **Transition Based** or **Cycle Based**. Only one option can be selected at a time.
  - **Transition Based:** When this option is selected, the *Temporal Flow View* uses the transition-based engine so that the input signals that trigger the output transition are automatically traced. This is used to

visualize the propagation of signal transitions throughout the design and over time which is very useful for gate level debug. This option cannot be used with a *Temporal Register View* window.

- **Cycle Based:** When this option is selected, the *Temporal Flow View* uses the cycle-based engine. That is, the design behavior is unrolled based on clock cycles so that the signals in the *Temporal Flow View* are displayed and traced according to the final stable value within a single clock cycle.

- **Add Signal to:** Determine where the **Reference Signal** is added. Select if you want to add a reference signal to a new *Temporal Flow View*, a new *Compact Temporal Flow View*, a new *Temporal Register View*, all existing flow views, or just the selected flow view.



- **Add:** After entering details for the **Reference Signal** and **Reference Time** fields, click **Add** to add this **Reference Signal/Reference Time** pair to the central list area of the *Add Reference Signals* form.

NOTE: Repeat the actions described in the **Reference Signal**, **Reference Time**, and **Add** sections above to add multiple **Reference Signal/Reference Time** pairs to the central list area of the *Add Reference Signals* form.

- **Delete:** To delete a **Reference Signal/Reference Time** pair from the central list area of the *Add Reference Signals* form, click the pair to select it and then click **Delete**.

Show Active Statement

Menu Bar:	Trace -> Show Active Statement
Mouse Action:	Right-click

NOTE: This command is available in the *Temporal Flow View* and *Compact Temporal Flow View*.

This command displays the symbol (output node) for the driving statement associated with the selected (left-click) input node in the main flow view display area.

Refer to the **View ->Active Nodes Only** command for details.

Show Fan-in Registers

Menu Bar: Trace -> Show Fan-in Registers

Mouse Action: Right-click

NOTE: This command is available in the *Temporal Register View*.

This command displays the first level of fan-in registers associated with the selected (left-click) register in the main flow view display area.

Refer to the **View -> Active Nodes Only** command for details.

Show Triggering Statement

Menu Bar: Trace -> Show Triggering Statement

Mouse Action: Right-click

NOTE: This command is available in the Transition-based *Temporal Flow View* and *Compact Temporal Flow View*.

This command displays the symbol (output node) for the driving statement of the selected (left-click) input node and indicates the most recent transition time on the vertical line.

NOTE: To open a transition-based flow view, enable the **Transition Based** option on the *Setup Temporal Flow View* form (invoke **Tools -> Temporal Flow View -> Setup Temporal Flow View** from the *nTrace* command).

Trace This Value

Menu Bar: Trace -> Trace This Value

Mouse Action: Right-click

This command has the same effect as performing multiple **Trace -> Show Fan-in Registers** (in the *Temporal Register View*) or **Trace -> Show Active Statements** (in the *Temporal Flow View* or *Compact Temporal Flow View*) recursively.

The command compares values on the active data fan-in signals with the commands on the fan-out signal. If the matched values are found, they are automatically taken as the new root nodes and the operation continues until the value cannot be found or the maximum trace is reached. The first occurrence of a value of interest can be tracked down with a single command.

Several options are associated with the **Trace This Value** command. These options can be turned *on/off* through the **Tools -> Preferences -> Temporal Flow View** page -> **Trace** page -> **Cycle-based Page** command.

NOTE: To invoke this command, a node in the main display area must be selected (left-click) first.

Refer to the **View -> Active Nodes Only** command for details.

Points to Note:

- It traces the active pin only if one active fan-in is available, even if there is no value change of the active pin.
- For multiple active pins, it traces the active data pin that has same value as the output value. That is, same-valued active data pins that have higher priority than other active pins.
- If no active data pin has same value as the output value, and there is only one active data pin has value change, then it traces the active fan-in that has value change at the tracing time.
- It terminates the trace if none of the above rules can be applied for the trace.

Trace This Register

Menu Bar: Trace -> Trace This Register

Mouse Action: Right-click

NOTE: This command is available in the *Temporal Register View*.

This command unrolls the selected (left-click) register output node over time and can be used to observe the value changes associated with the register and the fan-in signals driving the register.

Refer to the **View -> Active Nodes Only** command for details.

Trace X

Menu Bar: Trace -> Trace X

Mouse Action: Right-click

The Trace Triggering X Settings form is displayed as illustrated in the following figure:

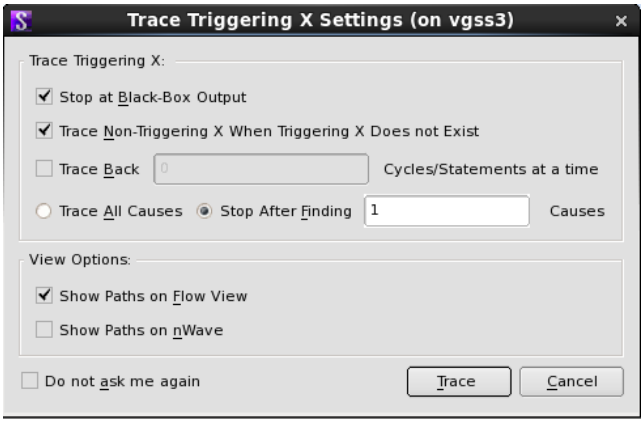


Figure: Trace Triggering X Settings Form

The following options are available in the **Stop Criteria** section.

- **Stop at Black Box Output:** During the trace back process, when this option is turned *on*, stop at signals that are outputs of black boxes (normally testbench, monitor, system tasks, or behavior models). When this option is turned *off*, continue tracing from black box inputs with unknown values. The default is *on*.
- **Trace Non-triggering X When Triggering X Does not Exist**: When this option is *on*, the unknown value on the non-triggering fan-in signals is traced when the triggering inputs have an unknown value. When this option is *off*, only the triggering input with a transition from a known value to an unknown value is traced. By default, the option is *on*.
- **Trace Back 'n' Cycles/Statements at a Time:** The value in this option determines the number of cycles or statements to display for each trace. The default is *0*. If the number of cycles to display is smaller than the number of cycles to the root cause of the unknown, the *Trace Active X Results* window do not open, but the traced cycles are displayed on the flow view.

- **Trace All Causes** or **Stop after Finding 'n' Cause(s)**: The functionality works in a depth first search manner. The algorithm can be set to **Stop after Finding 'n' Cause(s)** or **Trace All Causes**. Only one option can be selected at a time. The default is to stop tracing after the first cause is found.

The following options are available in the **View Options** section.

  - **Show Paths on Flow View:** The results are displayed on the **Trace Triggering X Results** window. When the **Show Paths on Flow View** option is *on*, the results are also displayed on the flow view. Showing the results on the flow view may slow down the search process. By default the option is *on*. You must enable this option only when you want to display the trace X results on the flow view.
  - **Show Paths on nWave:** The results are displayed on the **Trace Triggering X Results** window. When the **Show Paths on nWave** option is enabled, the results are also displayed in the *nWave* window as waveforms. The default is *off*. You must enable this option only when you want to display the trace X results on the *nWave* window.

- Click **Trace** to open the **Trace Triggering X Results** window. For more information see the *Trace* **Triggering** *X Results Window* section in the *Verdi and Siloti Command Reference Manual*.
- If the **Do not ask me again** option is selected in the *Trace X Setting* form before hiding it, then it does not appear the next time the Verdi platform is invoked.

This command tracks down the source of unknown X values, operates in a similar manner to the **Trace -> Trace This Value** command. The only difference between **Trace This Value** and **Trace X** is that the former continues tracing when matching values are found and the latter continues tracing signals with X values. A node containing an unknown 'X' value must be selected (left-click) in the main display area before invoking this command.

Refer to the *Trace X Page* under Temporal Flow View Folder section of the *Preferences* chapter for details.

Trace Glitch

Menu Bar:	Trace -> Trace Glitch
Mouse Action:	Right-click

NOTE To dump the glitch information during simv runtime, add the **+fsdb+glitch=0 +fsdb+sequential** option.

This command enables debugging a glitch in *Temporal Flow View* by identifying annotated glitches and locating the root source of the glitch. This command does not create a report but only helps trace signals with glitches.

When you invoke the **Trace Glitch** command from the *nWave* menu option, or from the right-click menu option on the *Temporal Flow View*, the **Display Glitch** option is automatically turned *on*.

If a reference signal in the *Temporal Flow View* contains a glitch, it is marked with a * symbol and the complete transition value is displayed, even if it is on the input side of the function block. The GUI command is available only when the selected signal contains glitches in its value.

Trace Triggering Path

Menu Bar: Trace -> Trace Triggering Path

Mouse Action: Right-click

NOTE: This command is available in the Transition-based *Temporal Flow View* and *Compact Temporal Flow View*.

This command searches for the most recent transitions on the fan-in signals of the selected transition-based node (green dot or green vertical line with arrows). The input nodes with the most recent transitions are automatically taken as the new root nodes and the operation continues until no transitions exist or the specified stop criterion is met.

Several options are associated with the **Trace Triggering Path** command. These options can be turned *on/off* through the **Tools -> Preferences -> Temporal Flow View** folder **-> Trace** folder -> **Transition-based Page**.

NOTE: To open a transition-based flow view, enable the **Transition Based** option on the *Setup Temporal Flow View* form (invoke **Tools -> Temporal Flow View -> Setup Temporal Flow View** from the *nTrace* menu option).

Show Active Statement for Partial Bus

Menu Bar: Trace -> Show Active Statement for Partial Bus

This command opens the *Partial Bus Trace* form for tracing a specified range of a selected multi-bit bus. For example, the causal logic of a bit range from 2 to 5

can be traced. A multi-bit (bus) cycle-based node must be selected (left-click) in the main display area of the flow view before this command becomes active.

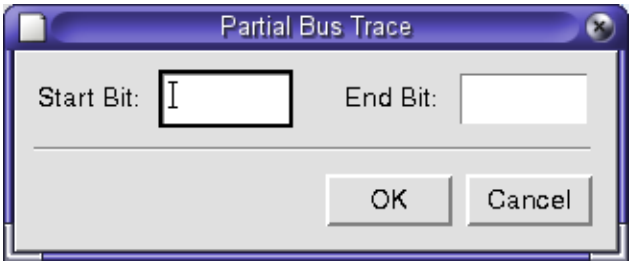


Figure: Partial Bus Trace Form

The *Partial Bus Trace* form includes the following options and buttons:

- **Start Bit:** Specify the start bit of the bus to trace.
- **End Bit:** Specify the end bit of the bus to trace.
- **OK:** After the **Start Bit** and **End Bit** values for the bus to trace are specified, click **OK** to display this bus in the flow view.
- **Cancel:** Click **Cancel** at any time to discard any modifications made to the *Partial Bus Trace* form and close the form.

Refer to the **View -> Active Nodes Only** command for details.

Behavior Trace for Waveform Mismatch

Menu Bar:	Trace -> Behavior Trace for Waveform Mismatch
Mouse Action:	Right-click

This command traces the root cause of a signal mis-match for two different waveform files generated using the same design. The Verdi platform expands the fan-in cone of the selected signal, determines when the fan-in signals are defined, and compares the value with the signals in the second waveform file. If a value mis-match exists, a non-equal sign (!=) is displayed on the node (the *Temporal Flow View* does not use this symbol) and the annotated value is displayed in red with the value from the second waveform file in parentheses.

If multiple fan-in signals with mismatched values exist, the trace priority is as follows:

- Any signal with an active transition
- Active control path

- Active data path

The first time this command is invoked, the *Load Simulation Result for Trace Mismatch* form opens so that the second waveform file can be loaded. The second waveform file is stored until the Verdi platform exits. The second waveform file can be loaded or changed through the **File -> Load 2nd Waveform for Trace Mismatch** command.

If the **Ask Every Time** option in the **Temporal Flow View** -> **Trace** -> **Trace Mismatch** page of the *Preferences* form is turned *on*, then each time the **Behavior Trace for Waveform Mismatch** command is invoked the *Trace Mismatch between Two FSDB Files* form opens.

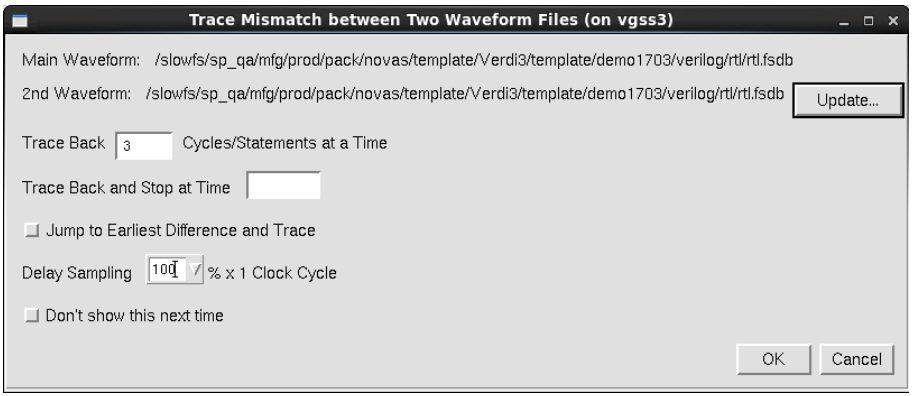


Figure: Trace Mismatch Between Two FSDB Files Form

The *Trace Mismatch between Two FSDB Files* form includes the following fields and options:

- **Main Waveform**: Displays the full path of the first waveform file to be traced.
- **2nd Waveform**: Displays the full path of the second waveform file to be traced. Click **Update** button to load or change the FSDB file.
- **Trace Back 'n' Cycles/Statements at a Time**: Set the maximum number of levels to be traced in a single execution of the command. The default value for this option is *on* with value *3*.
- **Trace Back and Stop at Time**: Specify the time to stop the mismatch trace.
- **Jump to Earliest Difference and Trace:** Jump to the earliest simulation mismatch in the fan-in cone of the selected signal and begin tracing from that time. The default value for this option is *on*.

- **Delay Sampling: % x 1 Clock Cycle:** Specify the value (percentage of clock cycle) for delay sampling by selecting **25**, **50**, or **100**. The value is saved in the *novas.rc* resource file. The default value for this option is *100*.

Enabling the **Don't show this next time** option prevents the form from opening again. To get the *Trace Mismatch between Two Waveform Files* form again, ensure the **Ask Every Time** option is turned *on* in the **Temporal Flow View** -> **Trace** -> **Trace Mismatch** page of the *Preferences* form (invoked with the **Tools -> Preferences** command).

Tooltip Information

In TFV, when you hover the mouse on the green circle, the mismatch information of the signals is displayed in the tooltip.

In cycle-based flow view, the mismatch information is displayed in the tooltip, as illustrated in the following figure:

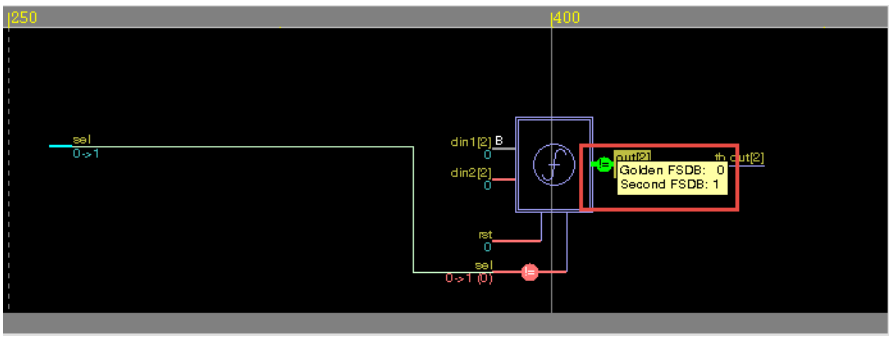


Figure: Mismatch Information Displayed for Cycle-Based Flow View

In transition-based flow view, the mismatch time is also displayed in the tooltip, as illustrated in the following figure:

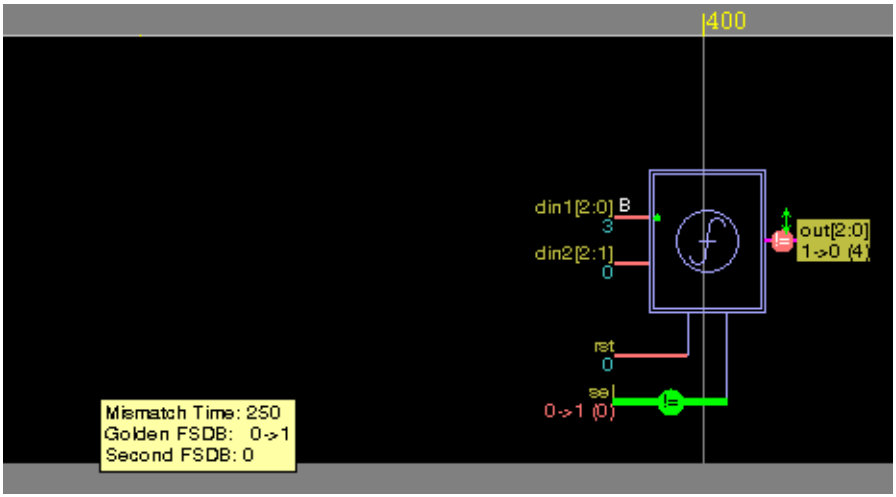


Figure: Mismatch Information Displayed for Transition-Based Flow View

Trace Again by Showing Cycle-by-Cycle Details

Menu Bar: Trace -> Trace Again by Showing Cycle-by-Cycle Details

This command traces back one cycle of a held register instead of jumping to the last transition. An output node with a green 'H' symbol must be selected in the main display area of the flow view before this command becomes active.

Show Cycle-by-Cycle Details of Holding Registers

Menu Bar: Trace -> Show Cycle-by-Cycle Details of Holding Registers

This toggle command turns the display of cycle-by-cycle details of holding registers *on* or *off*. When this toggle command is *on*, every node on the flow view is expanded cycle-by-cycle when a trace command is invoked. When this toggle command is *off*, cycle-by-cycle details of held registers is not displayed when a trace command is invoked. The **Show Cycle-by-Cycle Details of Holding Registers** option can also be turned *on* in the **Temporal Flow View -> Trace -> Trace Page** of the *Preferences* form.

Hide

Menu Bar: Trace -> Hide

Bind Key: Delete

Mouse Action: Right-click

This command removes a selected node (and all of its fan-in nodes) from the main display area. This command does not work on reference signal nodes.

Show Whole Fan-in Group

Menu Bar: Trace -> Show Whole Fan-in Group

Mouse Action: Right-click

This command expands the selected node to display the complete fan-in group of a holding signal (a signal with a green 'h' symbol). After this command is invoked on a selected node, the command is no longer 'active' for that node.

Fan-in Display Management

Menu Bar: Trace -> Fan-in Display Management

Mouse Action: Right-click

This command provides fine-grain control over which fan-in nodes associated with a particular output node are displayed or hidden. Invoke the command on a selected output node to open the *Fan-in Manager* form. Click the box associated with a node in the window to display it or remove it from the display (an “X” in a box indicates that this node is displayed). Click **OK** to accept the changes and dismiss the form or click **Cancel** to discard any modifications and dismiss the form.

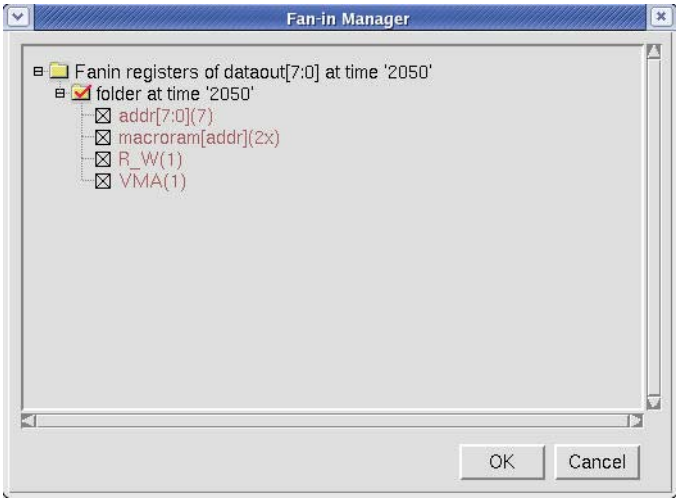


Figure: Fan-in Manager Form

After a node is removed from the display, the logic symbol or input port is turned orange in the *Temporal Flow View* or an orange symbol (3 dots) is added to the associated primary node in the *Compact Temporal Flow View* and *Temporal Register View* to indicate the existence of hidden information.

Clock Domain Highlight

Menu Bar: Trace -> Clock Domain Highlight

This command opens the *Clock Highlight* form in which one of the available clock domains can be selected and a color can be assigned to that domain.

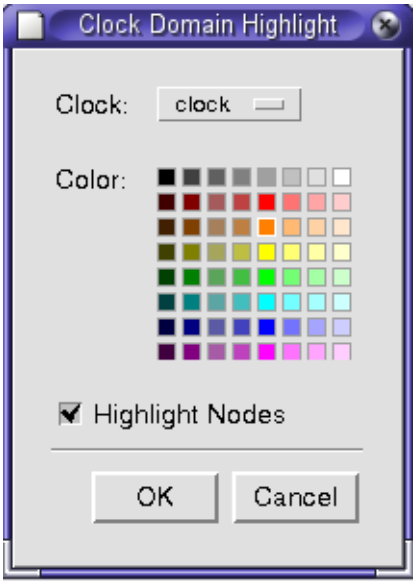


Figure: Clock Highlight Form

- Click **OK** to apply the selected color to all of the nodes that are controlled by the selected clock domain.
- Click **Cancel** to discard any modifications made and close the form.

NOTE: Color settings are not saved when exiting the Verdi platform.

Refer to the **Trace -> Dehighlight All** command for details.

Dehighlight All

Menu Bar: Trace -> Dehighlight All

This command removes the clock color highlights (set with the **Clock Domain Highlight** command) from the flow view.

Refer to the **Trace -> Clock Domain Highlight** command for details.

Tools Commands

Open Flow View

Temporal Flow View

Menu Bar: Tools -> Open Flow View -> Temporal Flow View

Mouse Action: Right-click

This command opens a new *Temporal Flow View* window using the selected (left-click) node in the current flow view as the reference signal.

Compact Temporal Flow View

Menu Bar: Tools -> Open Flow View -> Compact Temporal Flow View

Mouse Action: Right-click

This command opens a new *Compact Temporal Flow View* window using the selected (left-click) node in the current flow view as the reference signal.

Temporal Register View

Menu Bar: Tools -> Open Flow View -> Temporal Register View

Mouse Action: Right-click

This command opens a new *Temporal Register View* window using the selected (left-click) node in the current flow view as the referenced signal. If a node is not selected, a new *Temporal Register View* window opens with the original output node as the reference signal.

Dump Memory Waveform to FSDB

Menu Bar: Tools -> Dump Memory Waveform to FSDB

This command dumps the automatically traced memory value into the specified FSDB file to be displayed in *nWave* or *nMemory*. The resulting FSDB file can also be loaded in the MDA viewer.

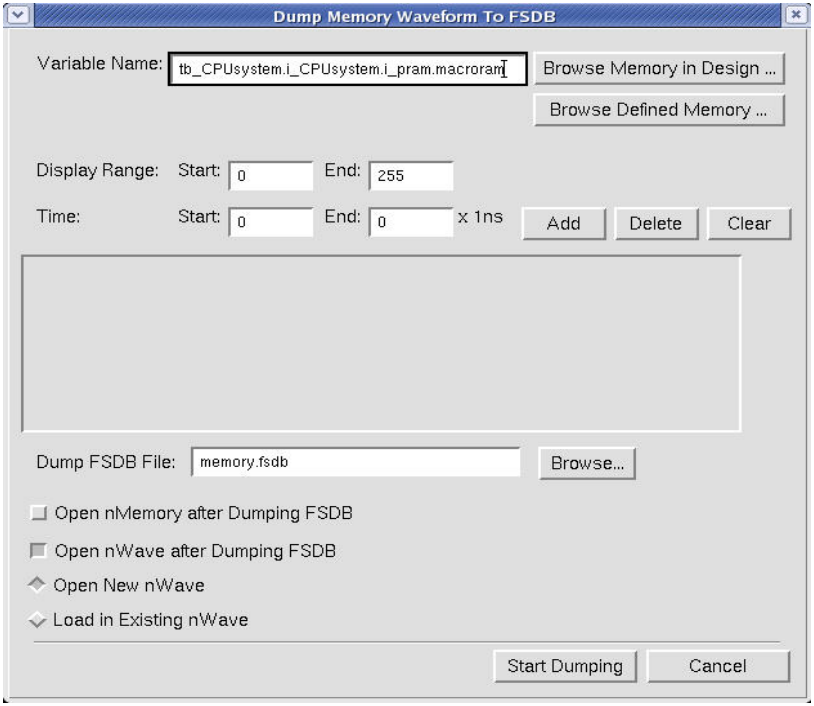


Figure: Dump Memory Waveform to FSDB Form

The *Dump Memory Waveform to FSDB* form includes the following fields, options, and buttons:

- **Variable Name:** Specify the memory to dump in this text field. If the memory is not selected when the **Dump Memory Waveform to FSDB** command is invoked, this field is blank. If a memory is selected, it defaults to the complete hierarchical path of the memory.
- **Browse Memory in Design:** Browse the design hierarchy and select the memory. A white highlighted rectangle indicates that the design scope is selected and the scope's signals are currently displayed in the signal list. The selected scope's inputs, outputs, and I/O's signals are color-coded and identified with a symbol. Internal nets are displayed in black. After selecting the memory, click **OK** to enter the memory in the *Dump Memory Waveform to FSDB* form.

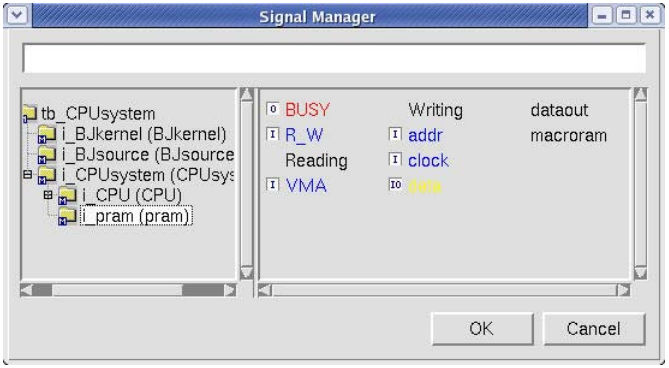


Figure: Browse Memory in Design - Signal Manager Form

- **Browse Defined Memory:** Browse the design hierarchy and select a predefined memory description. A white highlighted rectangle indicates that the design scope is selected and the defined memories are displayed in the signal list. See the **Memory Definition Table** command for details on creating a memory description.

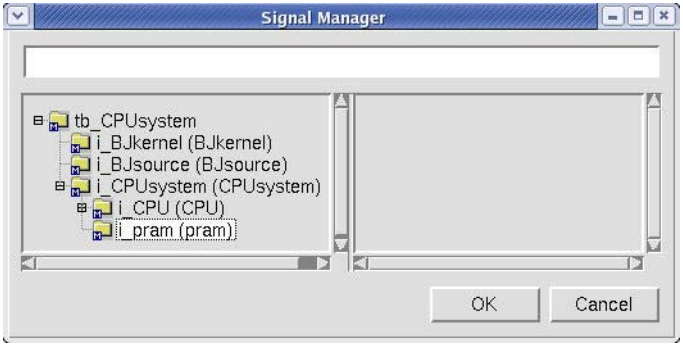


Figure: Browse Defined Memory - Signal Manager Form

- **Display Range:** Specify the amount of memory to dump.
  - **Start**: Specify the start address for the memory.
  - **End**: Specify the end address for the memory
- **Time:** Specify when to start and stop memory dumping.
  - **Start**: Specify the start time for dumping. The default value for this option is *0* when a memory is selected.
  - **End**: Specify the end time for dumping. If a memory is selected in *nTrace*, it defaults to the global cursor time. If a memory is selected in a flow view, it defaults to the last access time for that memory address. The last access time is for a read if the memory is not expanded. The

last access time for a write if the memory is expanded (double-left-click).

- **Add:** Click this button to add the selected memory with specified address and time to the list of memories to dump.
- **Delete:** Click this button to delete the selected memory from the list of memories to dump.
- **Clear:** Click this button to clear all memories from the list.
- **Dump FSDB File:** In this text field, specify the path and the name of the FSDB file to dump the memory contents. The default path is the local directory and the default name is '*memory.fsdb*'. The recommendation is to create a separate FSDB file for the memory and then use the *fsdbmerge* utility to create a single FSDB file which includes the memory and design data. Do not use the original design FSDB file name as the design simulation data is overwritten. Click **Browse** to open a file browser for specifying a different path and file name.

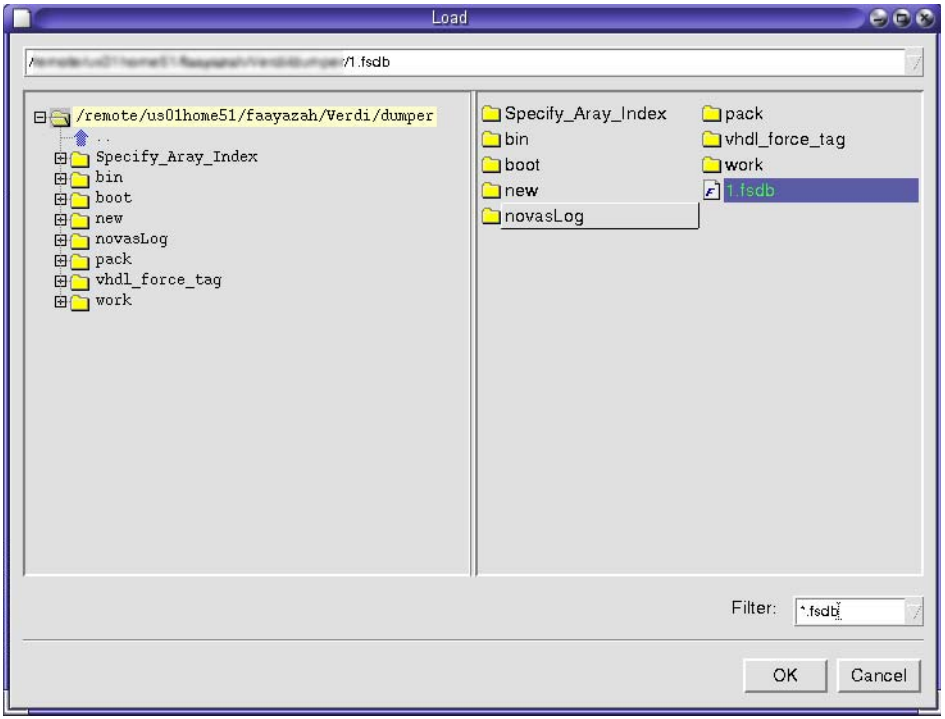


Figure: Dump FSDB File - Load Form

- Click **OK** to apply the selection and click **Cancel** to cancel the selection.

- **Open nMemory after Dumping FSDB:** When this option is turned *on*, open the memory and load the dumped memory contents automatically. The default value for this option is *off*.
- **Open nWave after Dumping FSDB:** When this option is turned *on*, open *nWave* and load the new FSDB file automatically. When this option is turned *on*, the **Open New nWave** option and the **Load in Existing nWave** options are enabled (only one option can be selected at a time). The default value for this option is *on*.
  - **Open New nWave:** When this option is selected, open a new *nWave* window. The default value for this option is *on*.
  - **Load in Existing nWave**: When this option is selected, load the new FSDB file in the current open *nWave* window.
- **Start Dumping:** This button starts the process of dumping the selected memory contents to the FSDB file.
- **Cancel:** This button closes the *Dump Memory Contents to FSDB* window. The selected memories and remaining options are saved.

Show Memory Contents

Menu Bar: Tools -> Show Memory Contents

This command opens the **Calculated by Verdi** tab of the *Get Memory Variable* form in which the memory content which needs to be calculated and displayed in the *nMemory* window can be selected. Refer to the *Calculated by Verdi Tab* section in the *Memory/MDA* chapter for details. Also refer to the **Memory Definition Table** command for details on creating a memory description.

Show Fan-ins on nSchema

Menu Bar: Tools -> Show Fan-ins on nSchema

Mouse Action: Right-click

This command displays the logic between the selected (left-click) node in the main flow view display area and the associated fan-in registers in an *nSchema* window. The command can be invoked multiple times from the same window and new logic is appended to the schematic view. Three options are available as follows:

- **All Fan-ins:**  Displays all the fan-in signals with the active one highlighted.

- **Active Only:** Displays only the active fan-in signals. Available for cycle-based nodes only.
- **Triggering Path Only:** Displays only the fan-in signals that trigger the output transition. Available for transition-based nodes only.

Each *nSchema* window is associated with each flow view. If *nSchema* is not opened previously from the flow view window, a new schematic window is opened. If *nSchema* is already invoked from this flow view, the logic is appended. If this command is invoked by a different flow view window, a different *nSchema* window is opened or used.

Show All Traced Paths on Flow-Schematic

Menu Bar: Tools -> Show All Traced Paths on Flow-Schematic

This command opens an *nSchema* window and automatically displays the logic and connectivity of the paths traced in the associated flow view. Nodes selected (left-click) in the flow view can be highlighted in *nSchema*. The command can be invoked multiple times from the same window and new logic is appended to the schematic view.

Each *nSchema* window is associated with each flow view. If *nSchema* is not previously opened from the flow view window, a new schematic window is opened. If *nSchema* is already opened from this flow view, the logic is appended. If this command is invoked by a different flow view window, a different *nSchema* window is opened or used.

Add to Waveform

The following are sub-commands of the **Add to Waveform** command.

New Waveform

This command enables you to create a new waveform window, open an annotation file, and add signals.

Add to Wave[n]

Bind Key: Ctrl+4

This command enables you to add selected objects to the last adding signal waveform or the new created waveform.

Add to New Group

This command creates a new group in the waveform and adds selected objects to this group.

*Wave[k]

Bind Key: Ctrl+W

This command adds selected objects to the waveform window. For example, *Wave[k], k is the primary waveform window Id.

Wave[m]

This command adds selected objects to waveform window. For example, Wave[m], m is the window id.

Show Selected Signals on nWave

Menu Bar: Tools -> Show Selected Signals on nWave

Toolbar Icon: 

Mouse Action: Right-click

This command adds waveform(s) for the selected signal(s) or the inputs/outputs of the selected instance(s) in the main display area of the flow view to the *nWave* window above the yellow cursor. If the *Temporal Flow View* window is displaying a RTL module, the nets associated with the I/O of the instance is used in *nWave*. If the *Temporal Flow View* window is displaying a gate module and the **Show Port Name for Dropped Instance** option under **Tools-> Preferences-> Waveform** page **-> View Options** page **->** **Miscellaneous Page** is turned *on*, the pin names for the instance I/O is used in *nWave*. If a pin name cannot be found or the **Show Port Name for Dropped Instance** option is turned *off*, the associated net is used.

The *nWave* window in which the signal is added is based on the state of the **Display Signal in New nWave Window** option in the **Tools -> Preferences -> Temporal Flow View** page ->**View** page ->**Automatic Command Page**.

Show Drivers on nWave

Menu Bar:	Tools -> Show Drivers on nWave
Bind Keys:	Ctrl+F Ctrl+A
Mouse Action:	Right-click

This command adds the driver signals of the selected node in the flow view to a waveform window. Three sub-commands are available as follows:

- **All Drivers** (Ctrl+A): Displays all the driver signals in the *nWave* window.
- **Active Only** (Ctrl+F): Displays only the active driver signals in the *nWave* window. This command is available for cycle-based nodes only.
- **Triggering Path Only:** Displays only the driver signals that trigger the output transition in the *nWave* window. This command is available for transition-based nodes only.

To specify the *nWave* window in which signals are added, turn *on* the **Display Signal to New nWave Window** option in the **Preferences -> Temporal Flow View** page -> **View** page -> **Automatic Command Page**. To specify the location of new signals, select one of the **Always Create New Group**, **In the Same New Group**, and **No New Group** options in the **Automatic Command Page**.

Show FSDB Mismatch on nWave

Menu Bar:	Tools -> Show FSDB Mismatch on nWave
Mouse Action:	Right-click

This command synchronizes the signals and displays the mismatched waveforms in the *nWave* window. The following figure illustrates an example of the mismatched waveform:

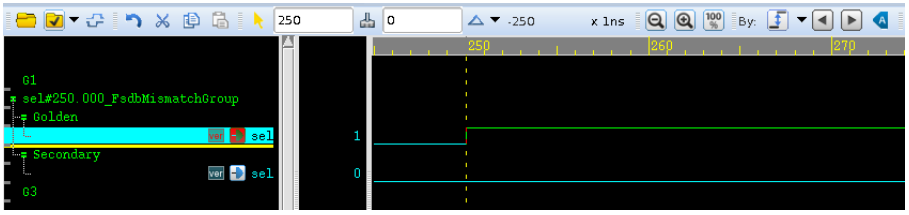


Figure: Mismatched Waveforms

Show Clock (Domain) on Waveform

Menu Bar: Tools -> Show Clock (Domain) on Waveform

Bind Key: Ctrl+K

Mouse Action: Right-click

This command adds the clock signal of the selected register (left-click a 'square' node in the *Compact Temporal Flow View* and Register Flow View or left-click a register symbol in the *Temporal Flow View*) to a waveform window. The clock signal is added above the yellow cursor.

The *nWave* window to which the signal is added is based on the state of the **Display Signal to New nWave Window** option in the **Tools -> Preferences -> Temporal Flow View** folder -> **Automatic Command Page**.

Show Synchronization Signal

Menu Bar: Tools -> Show Synchronization Signal

Mouse Action: Right-click

This command automatically adds the synchronization signals of the selected register in the flow view to a waveform window. Signals in the sensitivity list are recognized as synchronization signals.

For example, in the following Verilog RTL example:

```
always @(posedge clock or negedge reset)
begin
      if (!reset)
          reg = 0;
      else
          reg = reg_d;
end
```

Both clock and reset are considered synchronization signals and are added to the *nWave* window.

Options found on the **Tools -> Preferences -> Temporal Flow View** folder -> **Automatic Command Page** determine the *nWave* window to which the signals are added (**Display Signal to New nWave Window** option) and the location of the new signals (**When Adding Nodes to nWave** options).

Show All Traced Signals on nWave

Menu Bar: Tools -> Show All Traced Signals on nWave

This command adds all the traced signals in the flow view in the main *nWave* window.

Options found on the **Tools -> Preferences -> Temporal Flow View** folder -> **Automatic Command Page** determine the *nWave* window to which the signals are added (**Display Signal to New nWave Window** option) and the location of the new signals (**When Adding Nodes to nWave** options).

Preferences

Menu Bar: Tools -> Preferences

Refer to the *Preferences* chapter for details.

Customize Menu/Toolbar

Refer to the **Tools -> Customize Menu/Toolbar** command in the *nTrace* chapter for details.

Flow View Symbols and Notations

The Verdi platform provides three main views: *Temporal Flow View*, *Compact Temporal Flow View*, and *Register Flow View*. The legends and notations used in the main temporal flow view display areas are designed to be intuitive and easy to understand. Refer to the **Help -> Legend** command, **TFV** tab for details.

Temporal Flow View

The recommended flow view is the *Temporal Flow View*. Normal logic symbols (such as, multiplexor, combinational logic, or registers) are used to represent each statement. The term *output nodes* refers to the output ports of the logic symbols. The term *input nodes* refers to input ports of the logic symbols. Input nodes can be either active (affecting the output node) or inactive (not affecting the output node).

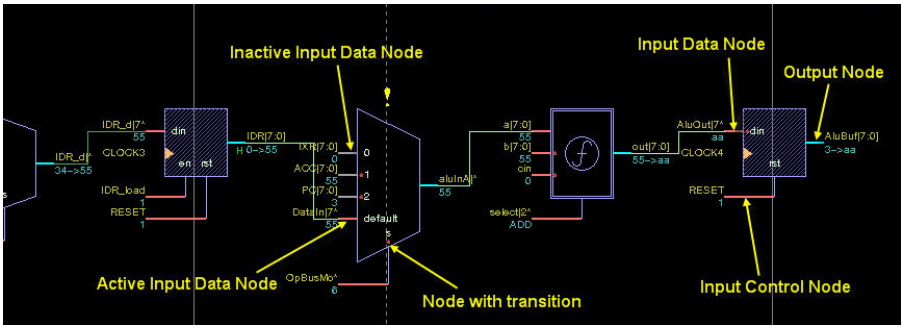


Figure: Temporal Flow View Symbols

The *Temporal Flow View* supports the view of functional block for digital and analog signals. The functional block of digital signal is displayed with a letter 'f' in purple. The functional block of analog signals is displayed with a letter 'SP' to represent SPICE signals in green. Note that TFV now only supports SPICE circuits which is represented by the keyword 'SP'. If an internal analog signal is selected, an output analog signal is mapped. For example, if a signal in the analog block and the signal is neither an input nor an output, then an output pin under the same scope is mapped.

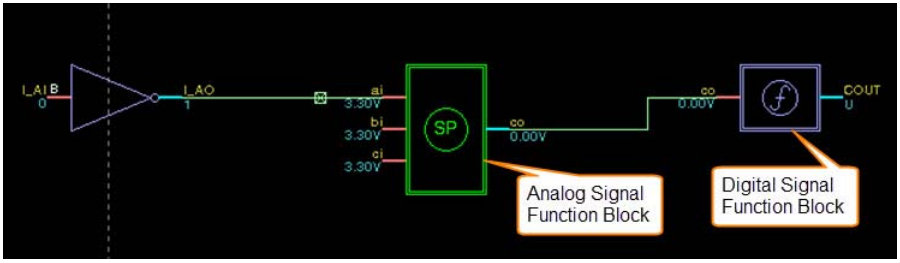


Figure: Digital / Analog Signal Function Blocks

The *Temporal Flow View* can have instances that are added/traced using the cycle-based engine or the transition-based engine. The cycle-based engine is the default instance. That is, the design behavior is unrolled based on clock cycles. Signals are displayed and traced according to the final stable value of the previous clock cycle. The transition-based engine is used to visualize the propagation of signal transitions throughout the design and over time. Transition-based instances add a green vertical line with arrows on either end and input nodes with transitions have a green dot in addition to other symbols.

In addition to the symbols, other annotations exist in the flow view that help in the debug process. The solid gray (red if the node is selected) vertical lines represent the active edges of clock signals on the output node. The dashed vertical lines represent the time when a primary input value changes. The value at the top of the gray lines indicate the time for the clock active edge. Solid white lines link input nodes with output nodes. Signal names and signal values (for the current cursor time) are located above and below the input/output pins. The temporal flow view distinguishes nine types of Interface Elements (IEs) and through nets (TNs), including **a2d**, **d2a**, **a2a**, **d2d**, **e2r**, **r2e**, **n2e**, **e2n**, and **inout**. All these IEs and TNs can be used in Verilog-A, System Verilog, and SPICE.

Some of the other symbols are similar to the symbols displayed in the *Compact Temporal Flow View* and *Temporal Register View* and can be accessed from the **Help -> Legend** command, **TFV** tab. For example, a green 'H' on an output node indicates a register that held its value because the active data path looped back on itself or a yellow 'M' on an input node indicates a memory.

Trace Invoked on Forced Signal

When tracing is invoked on a forced signal, the tracing stops at the driver of the forced signal. The time snaps to the nearest value change or force command/ statement. The denotations are the same in cycle-based and transition-based modes. The forced value is appended with the ^ symbol in front and labeled

(forced). The driver symbol is displayed with its input pins in gray, indicating that there is no contributor among the input pins.

NOTE: The tracing of signals within the fan-in cone of a forced/released signal is not affected.

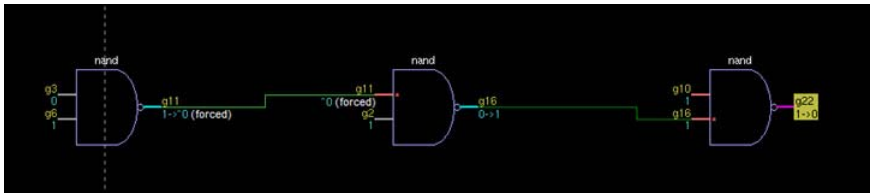


Figure: Forced Signal in Cycle-Based Mode

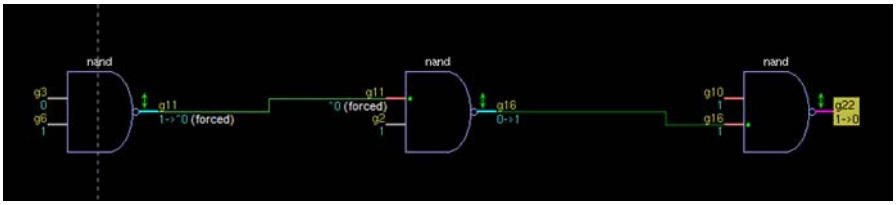


Figure: Forced Signal in transition-Based Mode

The current result of tracing the same signal (g22) is illustrated in the following figure. Tracing stops at the forced signal (g11) and does not expand. The 'B' indicates that the force statement is currently modeled as a black box and cannot be further expanded.

Figure: Current Result of Cycle-Based Tracing g22

If a forced net is composed of multiple segments, each segment that is affected by the forced statement or command is separated from previous segments by a dummy symbol, which has the shape as illustrated in the following figure. It is similar to the cross-boundary shape on net, because left-side and right-side scopes are different. Such an arrangement is to let you know that this net can no

longer be viewed as an equivalent net, since its segments may carry different values.

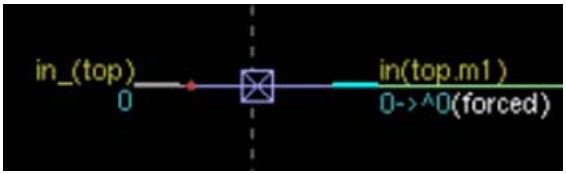


Figure: Dummy Symbol Shape

A force statement 'g6 = 1' occurred at time 40, as illustrated in the following figures. Both n6 and g6 are affected by the force statement. The trace stops at the inserted dummy symbol because the true contributor is the force statement rather than n6. Proceeding to trace n6 leads to the result. The assign symbol is inserted between n6 and g6 to indicate that these two nets can no longer be treated as equivalent nets, even though they have the same value in this case.

Figure: n6 and g6 Both Forced to 1 by Same Statement

Figure: Proceeding to Trace n6

In the following figures, only g6 is affected by the force statement, n6 and g6 have different values, and clearly cannot be considered equivalent. Also, tracing should stop at the assignment symbol driving g6, because the contributor comes from the force statement, but there is no reason to stop at the AND gate driving n6. Note that which segments is affected by a single force statement depending on the net types of each segment.

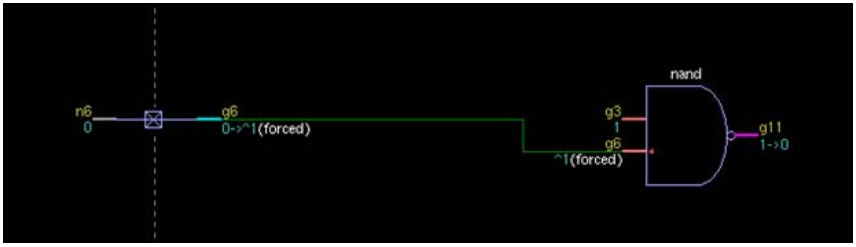


Figure: G6 Forced to 1

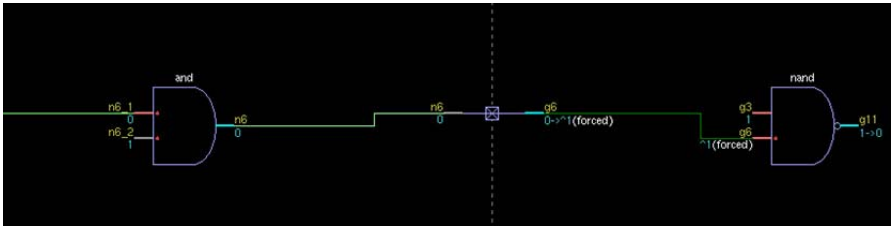


Figure: Proceeding to Trace n6

Trace Invoked on Released Signal

For released signals, the signal is labeled **(released)** only if the time snaps on the exact time the signal is released. If a trace is invoked after a value change occurs on the driver's inputs, as illustrated in the following figure, then the time snaps back to the value change. By then, the released signal is no longer driven by the previous force statement, and tracing behavior is the same as any unforced net, the contributor is the driver.

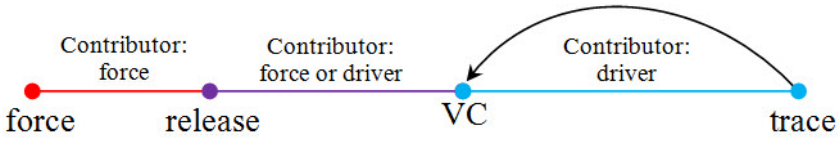


Figure: Snap Back to Nearest Value Change with Contributor as Driver

The following figure illustrates a case in which a value change occurs at the same time g16 is released.

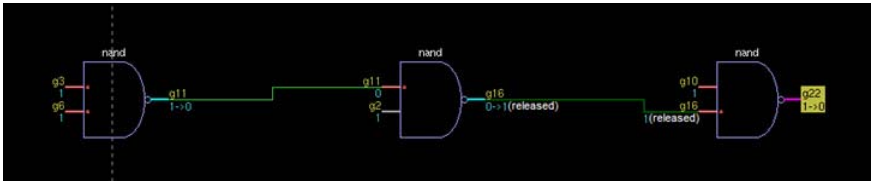


Figure: Tracing Currently Released Signal

If a trace is invoked before a value change occurs on the driver's inputs, then the time snaps back to the force command/statement, as illustrated in the following figure. The tracing behavior is the same as any forced net, the contributor is the force command/statement.

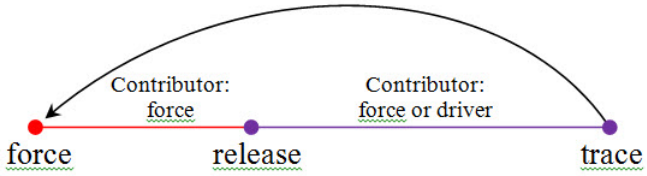


Figure: Snap Back to Previous Force With Contributor as Force

If the released net is composed of multiple segments, value changes caused by release (if any) are only displayed on segments driven by the released signal. In addition, the two segments that are displayed are determined by the same rule as the forced multiple segment nets. The following figures illustrate the expected results in cycle-based mode of tracing the same signal g11 when different segments (n6 and g6) are released at the current time(#40). The expected results and denotations of transition-based mode are the same as cycle-based mode.

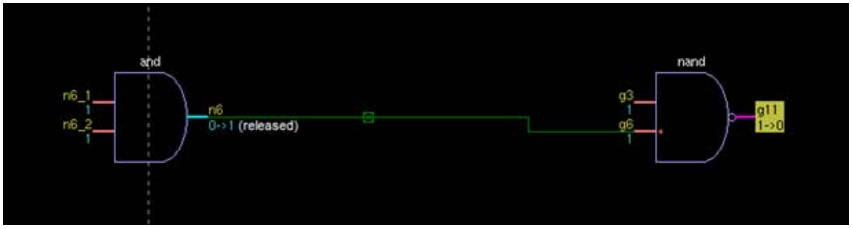


Figure: n6 Released at #40 With Value Change by Release Propagated to g6

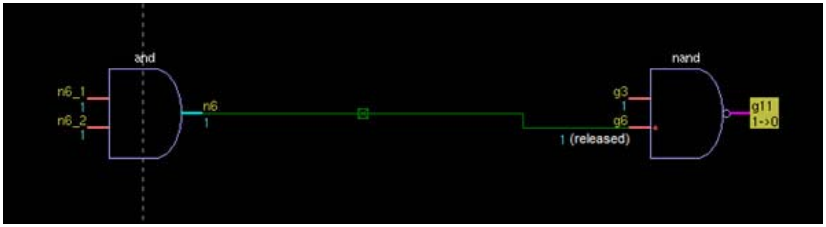


Figure: n6 Released at #40 Without Value Change by Release Propagated to g6

Trace X Caused by Force

When performing the **Trace X** command, tracing continues on input pins that have value X and stops when the root cause is located. When tracing X is caused by force, the same rule should be followed. However, forced X should be considered as a new type of root cause. The following figures illustrates the flow view and table of tracing an X caused by the statement 'force G9 = 1'bx'.

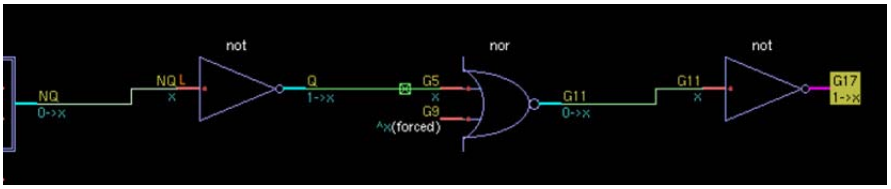


Figure: Trace Stops at G9 as Root Cause Found

	Signal	Time	Note
1	top.G0	10	X from initial block
2	top.s27_1.G9	10	Signal is forced to X

Figure: Root Cause Reported as Signal is Forced to X

Compact Temporal Flow View and Temporal Register View

The *Compact Temporal Flow View* and *Temporal Register View* use a group of symbols to represent logic in the design. If the exact meaning of the symbols is needed, refer to the *Legend* form for explanations. This form can be accessed under the **Help -> Legend** command, **TFV** tab.

Two main classes of signals exist in the flow view: register and inter-register (internal statement-level) signals. Both register and inter-register signals may be referred to as nodes. Register nodes are represented by rectangular boxes in both views. Inter-register nodes are represented by boxes with beveled ends in the *Compact Temporal Flow View*. Nodes are either input or output.

The term *output nodes* refers to the rectangles and/or octagons (default is yellow) displayed at the top of each group of nodes in the *Temporal Register View* (fan-out signal) or *Compact Temporal Flow View* (output signal of a statement). In the *Temporal Register View*, the term *input nodes* refers to groups of related registers (normally a set of fan-in registers) bounded by a red outline. In the *Compact Temporal Flow View*, the term input nodes refers to groups of signals (input signals of a statement) bounded by a red outline. For example, consider the following statement:

```
y = a & b & c;
```

In this case, y is considered an *output node* (from this statement), when a, b, and c are considered *input nodes* (to this statement).

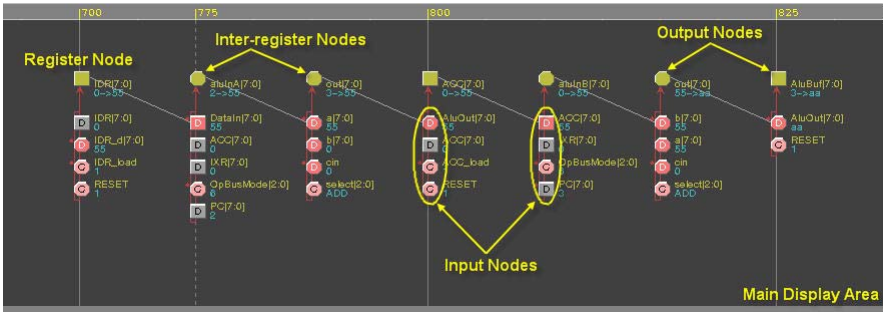


Figure: Compact Temporal Flow View Symbols

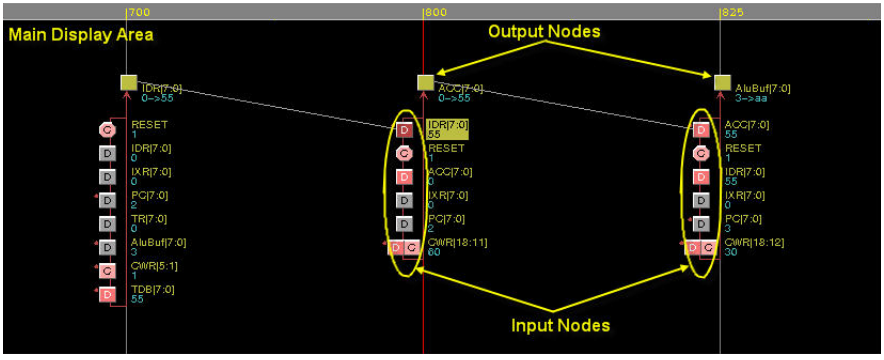


Figure: Temporal Register View Symbols

The pink and red boxes with "D" and "C" annotations indicate active data and control on an input node, respectively. The gray boxes with "D" and "C" annotations indicate inactive data and control on an input node, respectively. Some input nodes may have both "D" and "C" (active and inactive) annotations, which indicates that one or more bits from the node are performing a control function, while others are acting as data.

In addition to the symbols, other annotations exist in the flow view that aid in the debug process. In both views, solid gray (red if the node is selected) vertical lines represent the active edges of clock signals on the output node. In the *Compact Temporal Flow View*, dashed vertical lines represent the time when a primary input value changes. The value at the top of the gray lines indicate the time for the clock active edge. Solid gray horizontal lines link input nodes with output nodes (in the *Compact Temporal Flow View*) and fan-in nodes with fan-out nodes (in the *Temporal Register View*). Signal names and signal values (for the current cursor time) are located to the right of the node.

Additional Symbols

The basic symbols can be augmented by additional notation on the left- side which indicates signal status. It is possible for a single node to have multiple notations. The *Legend* form can be accessed under the **Help -> Legend** command, **TFV** tab.

Some symbols for grouped nodes, tri-state buses, and power aware signals exist in the *Temporal Flow View* but are not included on the legends. These symbols are created by the Verdi platform to better represent certain information. The related signals are considered 'virtual' signals (they do not exist in the design and they do not have simulation information). If possible, the Verdi platform calculates and displays a value for these symbols. If the Verdi platform is unable to calculate the value, nothing is displayed.

Grouped Nodes

The first symbol is 'G' for grouped nodes.

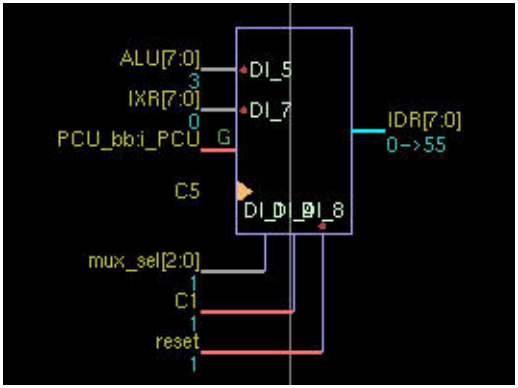


Figure: Symbol with 'G' Node

When a design unit (module/entity-architecture) is set as a library or macro cell, the flow view attempts to skip the cell and does not display the signals inside the design unit as much as possible. However, certain signals, such as flip-flop or latch signals, cannot be skipped and are defined as the internal state of that design unit. The flow view handles internal state signals as follows:

1. When a fan-in cone (or statement) contains multiple internal state signals of the same design unit instance, then the internal state signals are grouped into a single node. The grouped node contains a symbol 'G' on it with the name of 'cell_name:instance_name', where 'instance_name' is the design unit instance encompassing the internal state signals. Invoke the right mouse button menu option and select **Node Information** to examine the detailed information (for example, signal name and FSDB value) of the internal state signals.
2. For an un-grouped internal state signal, the naming follows a convention of 'instance_path.signal', where 'instance_path' is the hierarchical path starting from the upper-most level designated as a library or macro cell. For example, Reg_CMD.inst1.Q, where Reg_CMD is an instance of library cell D flip-flop and Q is a register signal.

Tri-state Buses

The other symbols that may appear is for tri-state bus drivers before bus resolution.

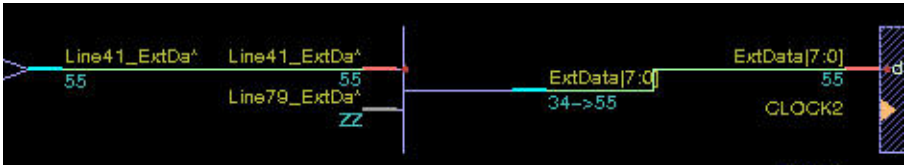


Figure: Tri-state Bus Symbol

The virtual signals associated with the tri-state bus symbol are identified with the source code line number and the associated signal. The value is blank when the symbol library model only flow is used, and the input/enable pin and the output are unknown. The Verdi platform is unable to calculate the value due to the unknown values.

Power Aware Signals

When CPF/UPF files and simulation results are loaded into the Verdi platform, the Temporal Flow View attaches power aware information to the original netlist without actually modifying the netlist. Virtual signals are created to model related power options. In addition, the following color settings are applied to power aware objects in the Temporal Flow View:

- Instances and nets in the same power domain have the same color.
- When an instance is in a power-off power domain, power-off instances is disabled.

The following sections explain these virtual signals and Power Aware-Temporal Flow View symbols.

NOTE: When the **Transition-based Method** option (located in the **Default Trace Method** section of the **Temporal Flow View** -> **Trace** -> **Trace** page of the *Preferences* form) is turned *on*, power signals is not displayed.

PSW

The PSW signal is used to indicate the status of the power domain in which the current instance is located. Its value can be *on*, *off*, or *unknown*. During power network tracing, the PSW signal is are treated like any other signal and can be traced through.

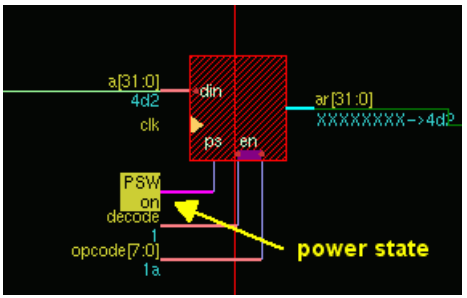


Figure: PSW Signal

PSO

The PSO signal is used to represent the *-shutoff_condition* option in the CPF file.

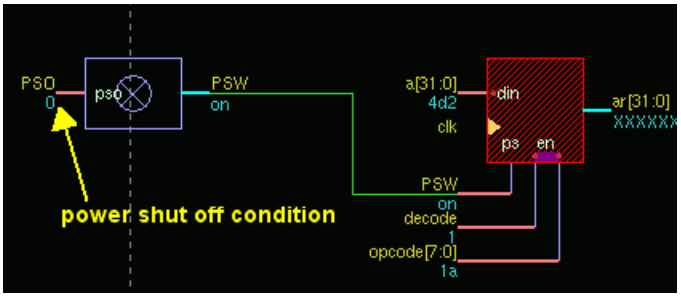


Figure: PSO Signal

CONDITION (For Isolation)

This signal is used to represent the *-isolation_condition* option in the CPF file and the *-isolation_signal* option in the UPF file.

During power network tracing, the CONDITION signal are treated like any other signal and can be traced through to identify the root cause of the value change.

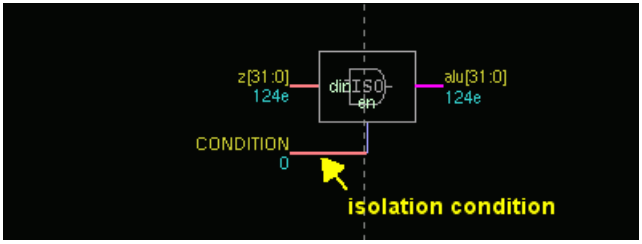


Figure: CONDITION Signal

SAVE, RESTORE, and RET (For Retention)

- **SAVE:** This signal is used to represent the *-save_edge* option in the CPF file and the *-save_signal* option in the UPF file.
- **RESTORE:** This signal is used to represent the *-restore_edge* option in the CPF file and the *-restore_signal* option in the UPF file.
- **RET:** As Retention is modeled as “latch + Flip-flop” (see the *Retention Model in TFV* figure), the RET signal is a temporary signal that connects the latch with the flip-flop and represents the saved circuit status value.

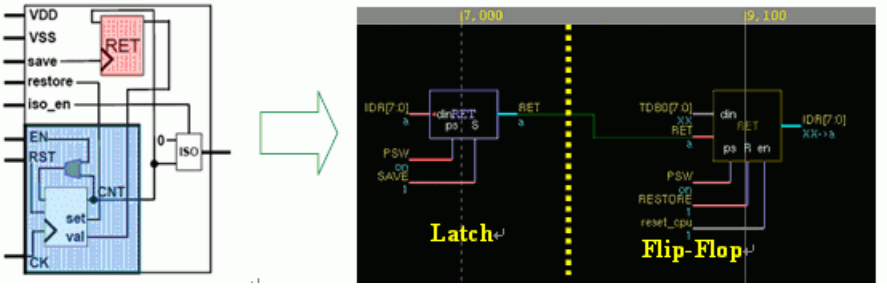


Figure: Retention Model in TFV

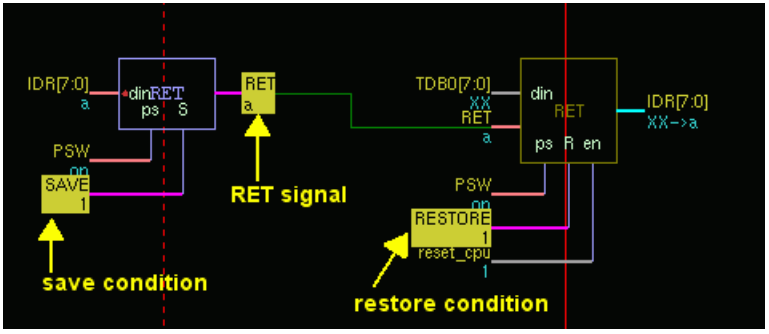


Figure: SAVE, RESTORE, and RET Signals

PRE_SAVE and PRE_RESTORE

- **PRE_SAVE:** This signal is used to represent the *-save_precondition* option in the CPF file.
- **PRE_RESTORE:** This signal is used to represent the *-restore_precondition* option in the CPF file.

During power network tracing, the PRE_SAVE signal and PRE_RESTORE signal behave like any other signal and can be traced through to identify the root cause of the value change.

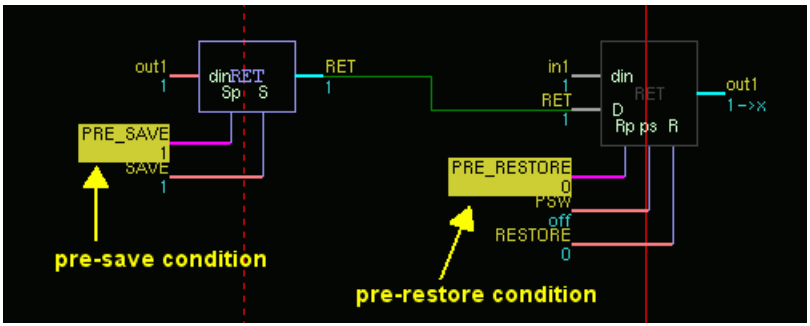


Figure: PRE_SAVE and PRE_RESTORE Signals

ON_STATE, OFF_STATE, ERR_STATE, and ON_PARTIAL_STATE

These signals are used to represent power switch related options in the UPF file. The values can be *true* or *false*.

- **ON_STATE:** This signal is used to represent the *-on_state* option.
- **OFF_STATE:** This signal is used to represent the *-off_state* option.
- **ERR_STATE:** This signal is used to represent the *-error_state* option.
- **ON_PARTIAL_STATE:** This signal is used to represent the *-on_partial_state* option.

During power network tracing, the ON_STATE signal, OFF_STATE signal, ERR_STATE signal, and ON_PARTIAL_STATE signal are treated like any other signal and can be traced through to identify the root cause of the value change.

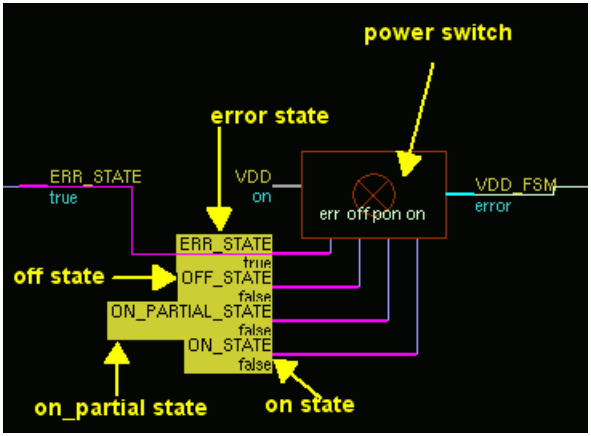


Figure: Power Switch Symbols

Right-Click Commands

Many of the previously described commands can also be selected from the right-click menu option in *Flow View*.

Flow View Frame Right-click Options

After the right mouse button is clicked anywhere in the menu, toolbar icon, or frame banner area of the *Flow View* window or standalone window, a configurationfigure the available icons.

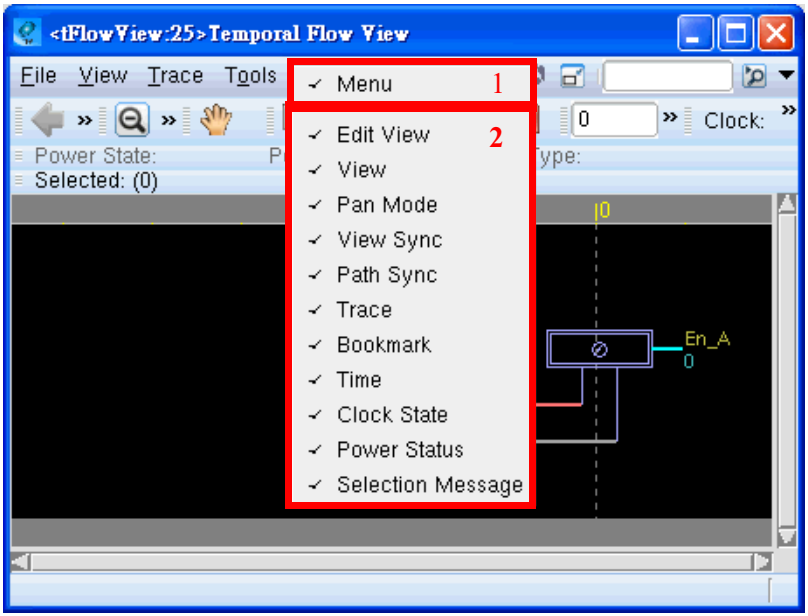


Figure: Configuration Option Menu

The **Menu** option (labeled 1 in the figure above) enables/disables the command menu bar. The options in the bottom section (labeled 2 in the figure above) enable/disable the toolbar icons for different functions.

Right-Click on Input and Output Node Commands

The following is the list of right-click menu options available for both input and output nodes.

Trace This Value

Refer to the **Trace -> Trace This Value** command description for details. This command is available for cycle-based output nodes.

Trace X

Refer to the **Trace -> Trace X** command description for details. This command is available for cycle-based output nodes.

Trace Glitch

Refer to the **Trace -> Trace Glitch** command description for details. This command is available only when a signal has a glitch.

Trace This Register

Refer to the **Trace -> Trace This Register** command description for details. This command is available for *Temporal Register View* output nodes.

Trace Triggering Path

Refer to the **Trace -> Trace Triggering Path** command description for details. This command is available for transition-based output nodes.

Behavior Trace for Waveform Mismatch

Refer to the **Trace -> Behavior Trace for Waveform Mismatch** command description for details. This command is available for cycle-based output nodes.

Bookmark

There are two sub-commands under the **Bookmark** menu.

Add Bookmark

Refer to the **View -> Bookmark -> Add Bookmark** command description for details.

Edit Bookmark

Refer to the **View -> Bookmark -> Edit Bookmark** command description for details.

Hide

Refer to the **Trace -> Hide** command description for details.

Show Detailed RTL

This command opens a flattened schematic frame and displays the symbolic representation of the RTL source code. This command is available for *Compact Temporal Flow View* and *Temporal Register View*.

Open Flow View

There are three sub-commands under the **Open Flow View** menu.

Temporal Flow View

Refer to the **Tools -> Open Flow View ->** **Temporal Flow View** command description for details.

Compact Temporal Flow View

Refer to the **Tools -> Open Flow View ->** **Compact Temporal Flow View** command description for details.

Temporal Register View

Refer to the **Tools -> Open Flow View ->** **Temporal Register View** command description for details.

Show Fan-ins on nSchema

Refer to the **Tools ->** **Show Fan-ins on nSchema** command description for details. There are two sub-commands: **All Fan-ins** and **Active Only**.

Show Selected Signals on nWave

Refer to the **Tools ->** **Show Selected Signals on nWave** command description for details.

Show Drivers on nWave

Refer to the **Tools ->** **Show Drivers on nWave** command description for details. There are two sub-commands: **All Drivers** and **Triggering Path Only**.

Show Clock (Domain) on Waveform

Refer to the **Tools ->** **Show Clock (Domain) on Waveform** command description for details.

Show Synchronization Signal

Refer to the **Tools -> Show Synchronization Signal** command description for details.

Node Information

This command displays additional information for the selected node.

NOTE: This command is available for 'U' (unintended latch nodes) and 'G' (Grouped Nodes).

For 'U' nodes, this command indicates the reason this signal is inferred as an unintended latch. Typically unintended latches are due to the fact that signals used in an always block are missing from the sensitivity list; therefore, the line number of the always block and the missing signals are identified.

For 'G' nodes, this command displays the signal names and FSDB values of the internal state signals for the grouped node.

Right-Click on Output Node Commands

The following is the list of right-click menu options available for output nodes only.

Collapse

Bind Key: Ctrl+O

This command collapses the selected output node. All input nodes associated with this output node are removed from the display.

Expand

Bind Key: Ctrl+E

This command expands the selected output node. All input nodes associated with this output node are added back to the display.

Right-Click on Input Node Commands

The following is the list of right-click menu options available for input nodes only.

Show Active Statement

Refer to the **Trace -> Show Active Statement** command description for details. This command is available for *Temporal Flow View* and *Compact Temporal Flow View*.

Show Fan-in Registers

Refer to the **Trace -> Show Fan-in Registers** command description for details. This command is available for *Temporal Register View*.

Show Triggering Statement

Refer to the **Trace -> Show Triggering Statement** command description for details. This command is available for transition-based input nodes.

Show Whole Fan-in Group

Refer to the **Trace -> Show Whole Fan-in Group** command description for details.

Fan-in Display Management

Refer to the **Trace -> Fan-in Display Management** command description for details.

Set Color

NOTE: This command is available in the *Temporal Register View* or the *Compact Temporal Flow View*. It does not exist in the *Temporal Flow View*.

This command opens the *Set Color* form where a new display color can be associated with the selected node. The new color is applied to every instance of this node in the main display area.

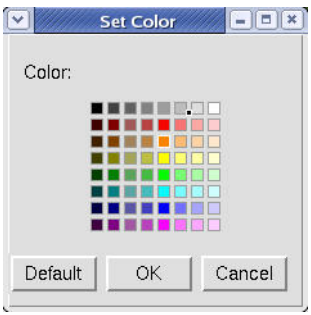


Figure: Set Color Form

The *Set Color* form includes the following options and buttons:

- **Color:** Select the new color by clicking on the color matrix.
- **OK:** Click this button to accept the selected color, apply it to all instances of the selected node in the main display area, and close the *Set Color* form.
- **Cancel:** Click this button at any time to discard any selected color and close the *Set Color* form.
- **Default:** Click this button to return all instances of the selected node in the main display area to their default color and close the *Set Color* form.

Reset Color

NOTE: This command is available in the *Temporal Register View* or the *Compact Temporal Flow View*. It does not exist in the *Temporal Flow View*.

This command resets the color to the original color on the selected node.

Refer to the associated **Set Color** command for details.

Show Driver on nTrace

This command locates and highlights the selected node's driver in the *nTrace* window.

Toolbar Icons and Fields

Figure: Temporal Flow View - Options Enabled

The available toolbar icons may be modified. Refer to the *Toolbars* section of the *User Interface* chapter in the *Verdi User Guide and Tutorial* manual for details.

The different toolbar categories and available icons are described below.

Edit View Category

Back 

Click this toolbar icon to revert the main display area to its previous zoom level. This command can be used multiple times to cycle back through a series of earlier zoom levels.

Refer to the corresponding **Forward** toolbar icon and the **View -> Back** and **View -> Forward** drop-down menu options for details.

Forward 

Click this toolbar icon causes the main display area to proceed to the next zoom level.

NOTE: One or more **Back** commands must be executed before a similar number of **Forward** commands can be executed.

Refer to the corresponding **Back** toolbar icon and the **View -> Forward** and **View -> Back** drop-down menu options for details.

View Category

The following icons are available in the default *Temporal Flow View* window.

Zoom In 

Refer to the **View -> Zoom -> Zoom In** command description for details.

Zoom Out 

Refer to the **View -> Zoom -> Zoom Out** command description for details.

The following icon is available with all windows.

Fit Time 

Click this toolbar icon to display all the folders and nodes that are currently under evaluation with the main display area to zoom in or zoom out. Refer to the **View -> Zoom ->** **Fit Time** command description for details.

The following icons are available in the default *Compact Temporal Flow View* and *Temporal Register View* windows.

Zoom Out Horizontally 

Refer to the **View -> Zoom ->** **Zoom Out Horizontally** command description for details.

Zoom In Horizontally 

Refer to the **View -> Zoom ->** **Zoom In Horizontally** command description for details.

Zoom Out Vertically 

Refer to the **View -> Zoom ->** **Zoom Out Vertically** command description for details.

Zoom In Vertically 

Refer to the **View -> Zoom ->** **Zoom In Vertically** command description for details.

Pan Mode Category

Switch to Pointer Mode  /Switch to Pan Mode 

Click this toolbar icon which causes the mouse cursor to provide standard pointer functionality when in a flow view's main display area. This acts in an identical way to the **View ->** **Switch to Pointer Mode** drop-down menu command.

NOTE: This is a switch command. After the **Switch to Pointer Mode** icon is clicked, the icon changes into an arrow. The function associated with this new icon is **Switch to Pan Mode**, behaves similarly to selecting the **View ->** **Switch to Pan Mode** command.

View Sync Category

Enable/Disable Show nWave Automatically 

Refer to the **View -> Enable Show nWave Automatically** and **Disable Show nWave Automatically** commands for details.

Enable/Disable Show Source Automatically 

Refer to the **View -> Enable Show Source Automatically** and **Disable Show Source Automatically** commands for details.

Enable/Disable Flow Schematic Automatically 

Refer to the **View -> Enable Flow Schematic Automatically** and **Disable Flow Schematic Automatically** commands for details.

Path Sync Category

Show All Traced Paths on Flow Schematic 

Refer to the **Tools -> Show All Traced Paths on Flow-Schematic** command for details.

Trace Category

Trace This Value 

Refer to the **Trace -> Trace This Value** pull down menu for details.

Trace X 

Refer to the **Trace -> Trace X** pull down menu for details.

Bookmark Category

Bookmark 

Refer to the **View -> Bookmark -> Add Bookmark** pull down menu for details.

Time Category

Current Time

When any node in the main display area is clicked, the time associated with that node is displayed in this field.

Refer to the **View -> Set Window Time Unit** command in which the default time units and time base (s, ms, us, ns, ps, fs) used in the main display area and reflected on the toolbar may be set.

Clock State Category

Clock

When a register node in the main flow view display area is clicked, its active clock edge (rising or falling) and its clock domain (the name of its associated clock) is displayed in this field.

Power Status Category

The following toolbar menu appears in the *Temporal Flow View* window when a CPF/UPF file is loaded through the Import CPF/UPF Files command in *nTrace*.

Power Domain Status

This field is available when a CPF/UPF file is loaded and a node is clicked. This field displays the power domain status as *on* or *off*.

Power Domain

This field is available when a CPF/UPF file is loaded and a node is clicked. This field displays the power domain name with the specified highlight color. When an instance is selected, the power domain of this instance is displayed. When the fan-out signal of an instance is selected, the power domain of this signal is displayed. The color can be redefined by the **Tools** -> **Highlight Power Domain** command in *nTrace* or the **Highlight Power Domain** option (default value for this option is *on*) on the **General** -> **Power** page of the *Preferences* form (invoked with the **Tools** -> **Preferences** command).

Type

This field is available when a CPF/UPF file is loaded and a Retention/Isolation signal is selected. This field displays the power type as Retention signals *(ret)* and/or Isolation signals (iso).

NOTE: When tracing a signal specified to be a boundary port in the power code (using the **-boundary_ports** option), the *Temporal Flow View* creates a new cell to model the -boundary_ports and displays the corresponding power domain state.

Trace Method Category

Trace Method

The **Trace Method** field displays the selected trace method. The trace method can be specified in the **Temporal Flow View folder -> Trace folder -> Trace** page of the *Preferences* form (invoked with the **Tools -> Preferences** command). The default value for this option is *Transition-based*.)

Selection Message Category

Selection Message

This field displays information about the selected object.

Trace Triggering X Results Window

Overview

The *Trace Triggering X Results* window is opened when **OK** is clicked on the *Trace Triggering X Settings* form after the **Trace X** command is invoked from *nTrace*, *nWave*, or a flow view. Alternatively, it can be opened when **OK** is clicked on the *Trace Triggering X Setting* form after the **Trace X** command is invoked from a flow view. The *Trace Triggering X Results* window is docked to the same frame location as the source code frame as a new tab (see red circle below). The *Trace Triggering X Results* window can become a standalone window by clicking the **Be Window** icon on the toolbar. Refer to the *Icons for Dockable Panes* section for details on the menu bar icons.

A signal can be dragged from the *Trace Triggering X Results* window and dropped to the *Temporal Flow View* window. The signal added to the current *Temporal Flow View* window are highlighted.

This form displays the resulting signal(s), time, and reason for unknown.

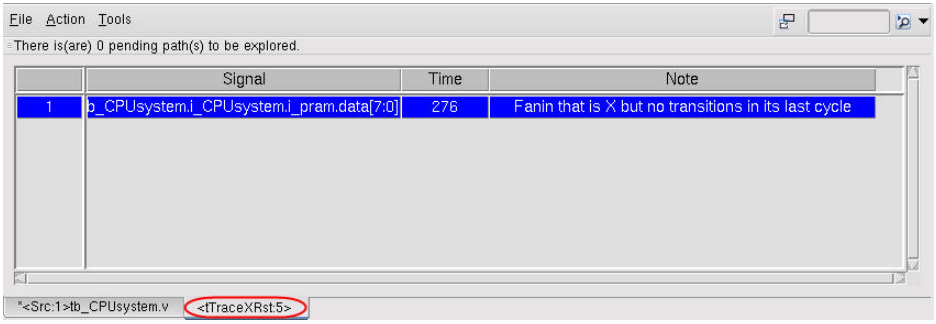


Figure: Trace Triggering X Results Window

The results form has three columns. The width of each column can be adjusted by placing the cursor over the vertical bar in the heading row and then pressing and holding the left mouse button. The three columns include:

- **Signal:** Complete hierarchical path for the signal.
- **Time:** The time the signal became unknown.
- **Note:** The reason code for the unknown. See the following for the note list:
  - *Fanin that is X but no transitions in its last cycle:* The tracing stops because the transition time of the fan-in exceeded the last cycle time of

the fan-out. This message appears only when the **Stop at Fan-in that is X but No Transition in Its Last Cycle** option is turned *on*.

- *X from primary input of the working scope*
- *X from constant X:* A constant 'X' is assigned to a signal. This can also be in the simulation model for a gate.
- *X from un-initialized integer:* An 'integer' data-type signal is not initialized.
- *X from un-initialized signal:* A 'reg' data-type signal is not initialized.
- *X from un-initialized storage element:* A storage element (latch or register) is not initialized.
- *X from un-initialized array (memory) element:* The memory net is not initialized.
- *X from black box output:* This message appears only when the **Stop at Black Box Output** option is turned *on*.
- *X from UDP:* Input does not match any entry in the UDP table.
- *X from BUS contention*
- *X from BUS (BUS keeper?)*: Reported when a tri-state bus has an 'X' but the 'X' is not caused by bus contention. The unknown may be due to BUS keeper behavior.
- *X caused by timing violation:* Transition found on interfere signal.
- *X from setup/hold timing violation; or bad (X or Z) clock, or set/reset signal*
- *X caused by floating net*
- *Triggering input isn't 'X':* This message appears only when the **Stop when Triggering Input is not 'X'** option on the *Trace Triggering X Setting* form is turned *on*.
- *No fanin that is X:* The default statement when the X cause cannot be determined.
- *X is caused by power off; the state of root supply net/port is OFF*.
- *X is caused by power off; the state of root supply net/port is PARTIAL_ON*.
- *X is caused by power off; the state of root supply net/port is ERROR*.
- *X is caused by power off; the state of root supply net/port is UNDETERMINED*.
- *X is caused by power off; the isolation supply net/port is OFF*.
- *X is caused by power off; the isolation supply net/port is PARTIAL_ON*.
- *X is caused by power off; the isolation supply net/port is ERROR*.

- *X is caused by power off; the isolation supply net/port is UNDETERMINED.*
- *X is caused by power off; the 'off' state of power switch is active.*
- *X is caused by power off; the 'error' state of power switch is active.*
- *X is caused by power off; the 'on_partial' state of power switch is active.*
- *X is caused by power off; a conflict exists in the power state of the power switch.*
- *X is caused by power off; active power state is not available on power switch.*
- *X is caused by power off; the isolation supply net/port is OFF.*
- *X is caused by power off; the isolation supply net/port is PARTIAL_ON.*
- *X is caused by power off; the isolation supply net/port is UNDETERMINED.*
- *X is caused by power off; the isolation supply net/port is ERROR.*
- *The X is caused by the X value from the enable pin of an isolation cell.*
- *The state of root supply net/port is OFF.*
- *The state of root supply net/port is PARTIAL_ON.*
- *The state of root supply net/port is ERROR.*
- *The state of root supply net/port is UNDETERMINED.*
- *The 'off' state of power switch is active.*
- *The 'error' state of power switch is active.*
- *The 'on_partial' state of power switch is active.*
- *A conflict exists in the power state of the power switch.*
- *Active power state is not available on the power switch.*
- *The control signal of the power state is unknown.*
- *The retention cell is restoring the X value.*
- *The retention cell is saving the X value.*

The result can be selected with a left-click anywhere in the row. Only one result may be selected at a time.

The drop-down menus for the *Trace Triggering X Results* window are explained in the following sections.

File Commands

Save to File

Menu Bar: *Trace Triggering X Results*, File -> Save to File

This command opens the *Save to File* form for saving the Trace X results in text format for later review outside the Verdi platform. Click **Browse** to invoke the *Browse* form where the directory structure can be viewed to select a file name for saving the results. Click **OK** to close the window and save the results. Click **Cancel** to close the window without saving anything.

Close

Menu Bar: *Trace Triggering X Results*, File -> Close

This command closes the current *Trace Triggering X Results* window.

Action Commands

Trace Next ‘n’ Causes

Menu Bar: *Trace Triggering X Results*, Action -> Trace Next ‘n’ Causes

This command continues tracing and adds the next '*n*' sources of an unknown to the results form. This command is enabled when the **Stop After Finding n Cause(s)** option is turned *on* in the *Trace Triggering X Settings* form. The command is disabled when no further sources of unknown can be found.

Show Source Code on nTrace

Menu Bar: *Trace Triggering X Results*, Action -> Show Source Code on nTrace

The associated source code for the selected result is highlighted in *nTrace*. The hierarchical scope in *nTrace* is updated to reflect the current scope. This command is also available by right-clicking on the result to invoke the context menu. Attentively, the source code can be highlighted by double-clicking-left on the result.

Add All Fan-in Signals to nWave

Menu Bar: *Trace Triggering X Results*, Action -> Add All Fan-in Signals to nWave

This command adds all fan-in signals for the signal in the selected result to the *nWave* window. This command is also available by right-clicking on the result to invoke the context menu.

Add Active Fan-in Signals to nWave

Menu Bar: *Trace Triggering X Results*, Action -> Add Active Fan-in Signals to nWave

This command adds only the active fan-in signals for the signal in the selected result to the *nWave* window. This command is also available by right-clicking on the result to invoke the context menu.

Add Reference Signal

Menu Bar: *Trace Triggering X Results*, Action -> Add Reference Signal

This command opens the *Add Reference Signals* form. Refer **Trace -> Add Reference Signals** for details. This command is also available by right-clicking on the result to invoke the context menu.

Continue to Trace Selected Signal

Menu Bar: *Trace Triggering X Results*, Action -> Continue to Trace Selected Signal

This command continues tracing to the next source of the unknown for the signal in the selected result. This command is enabled for certain types of causes (that is, Fanin that is X but no transitions in its last cycle). This command is also available by right-clicking on the result to invoke the context menu.

Show Low Possibility Causes

Menu Bar: *Trace Triggering X Results*, Action -> Show Low Possibility Causes

This command is available if the Verdi platform locates a low possibility cause, which is typically a result of an 'X without transition'. This is only reported if the **Stop at Fan-in that is X but No Transition in Its Last Cycle** option on the *Trace Active X* form is turned *on*.

Expand Selected Cause

Menu Bar: *Trace Triggering X Results*, Action -> Expand Selected Cause

When the same signal has the same 'X' cause at different times, the results are automatically merged into a single line. This command expands individual causes on the selected (left-click) result to individual lines. This command is also available by right-clicking on the result to invoke the context menu.

Merge Causes on Same Signal

Menu Bar: *Trace Triggering X Results*, Action -> Merge Causes on Same Signal

This command merges the results of individual signals with the same 'X' cause at different times into a single line. This command is also available by right-clicking on the result to invoke the context menu.

Highlight Selected Trace Path

Menu Bar: *Trace Triggering X Results*, Action ->Highlight Selected Trace Path

This command highlights all pins, connections and function blocks on the trace path when a root cause is selected in the *Trace Triggering X Results* window. The default value for this option is *off*.

Tools Commands

Customize Menu/Toolbar

Refer to the **Tools -> Customize Menu/Toolbar** command in the *nTrace* chapter for details.

nCompare

Overview

The *nCompare* window can be invoked in nWave from the **Tools -> nCompare** command. It is docked to the same frame location as the *Source Code* frame as a new tab. Click the **Undock** icon on the toolbar to make the *nCompare* window a standalone window

The *nCompare* module is a waveform comparison tool which supports the comparison of different types of simulation results at different design phases, such as digital or analog; behavior, RTL, or gate-level simulation runs; before/after Placement and Routing; and before/after delay annotation.

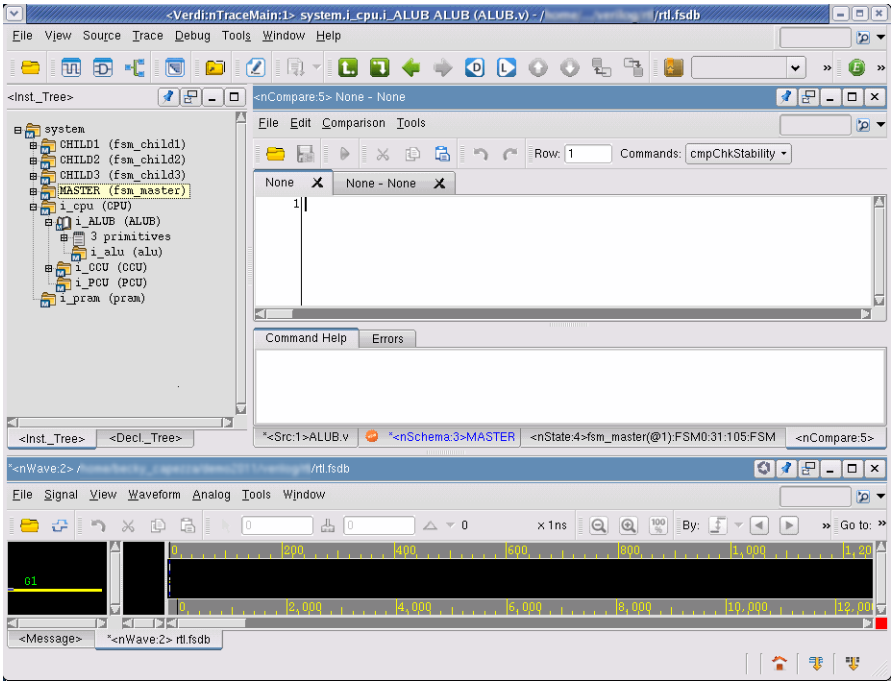


Figure: nCompare Window

The *nCompare* module processes a rule file compatible with Tcl language syntax, which enables more flexible waveform comparison. The *nCompare* module can

run in console mode so that large amounts of daily regression tests can be performed. It can also run in GUI mode so that discrepancies between two simulation runs can be viewed with *nWave*. With the *nCompare* module, different simulation runs can be compared using flexible and powerful comparison rule commands.

The *nCompare* module provides the following features:

- Compares all the selected signals by time sequence, therefore finding the design error with minimum comparison time.
- Displays the mismatch results in the *nCompare* window.
- Sorts the mismatch results in the *nCompare* window by simulation time or design hierarchy.
- Tightly integrates with *nWave* for easy viewing of the mismatch results in the *nWave* window.
- Provides batch mode utility for regression tests.
- Defines type variables for name mapping rules.
- Compares signals with different styles.
- Supports analog waveform comparison.
- Supports VHDL.
- Compares memory signals and non-digital values, such as integer, enum, and real.
- Supports arbitrary state mapping.
- Includes a flexible comparison trigger setting.
- Supports definable name-mapping rules for comparing RTL signals and after-synthesis gate-level signals.

The menu bars for the *nCompare* window are as follows:

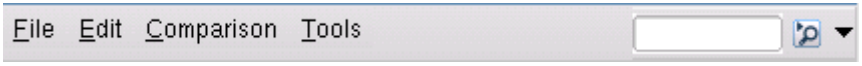


Figure: nCompare Menu Bar

The menus are explained in the following sections. Refer to the *Icons for Dockable Panes* section for details on the menu bar icons.

Menu Summary

The *nCompare* menu options are summarized below. Double-click on the commands to jump to the corresponding command description. For right-click menu options, refer to the *Right-Click Commands* section for details.

File Commands

New Rule File
Open Rule File
Save Rule File
Save Rule File As
Open Error File
Save Error
Save Error As
View Comparison Log
Exit

Edit Commands

Undo
Redo
Cut
Copy
Paste

Comparison Commands

Run
Pause
Stop

View Commands

First Error
Previous Error
Next Error
Last Error
Expand All
Collapse All
View by Hierarchy
View by Time
View on Waveform->
  Selected Error
  All Errors
Properties

Tools Commands

New Waveform
Sync Waveform View
Filter
Reset Filtering Results
Preferences
Customize Menu/Toolbar

Bind Keys

For a complete list of bind keys used in *nCompare*, refer to the *nCompare* section in the *Bind Key Summary* for details.

File Commands

New Rule File

Menu Bar: File -> New Rule File

This command adds a blank rule file tab to the *nCompare* window.

Open Rule File

Menu Bar: File -> Open Rule File

Toolbar Button: 

Bind Key: Ctrl+O

This command opens the *Open Rule File* form where a previously created rule file can be loaded.

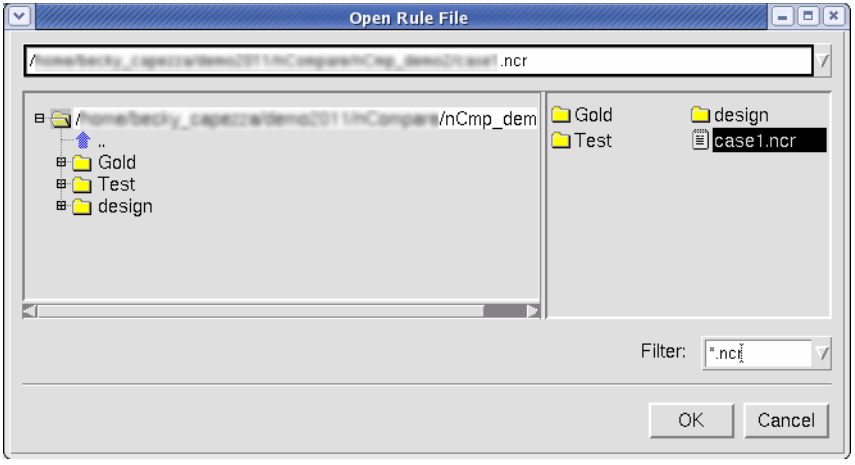


Figure: Open Rule File Form

The imported rule file appears in the selected **Rule File** tab and the imported file name appears on the title of the *nCompare* window.

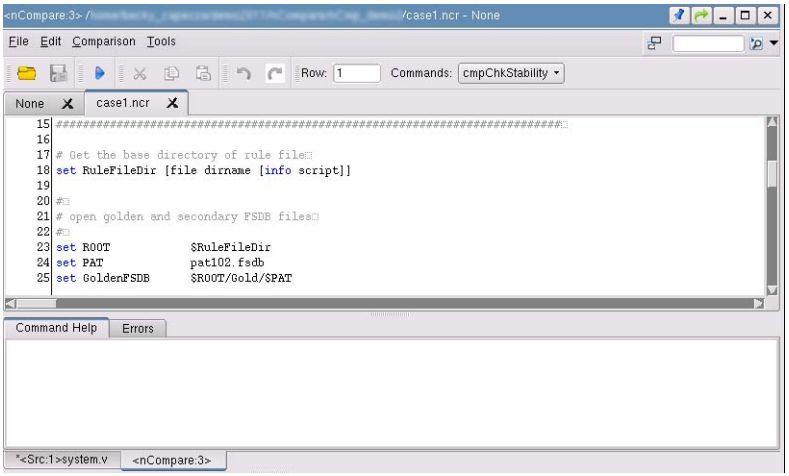


Figure: Rule File Tab

Save Rule File

Menu Bar: File -> Save Rule File

Toolbar Button:

Bind Key: Ctrl+S

This command saves the current rule file after editing.

Save Rule File As

Menu Bar: File -> Save Rule File As

This command saves the current rule file to the name specified in the *Save Rule As* form.

Open Error File

Menu Bar: File -> Open Error File

This command imports a previously saved error report file (*.nce) through the *Open Error File* form.

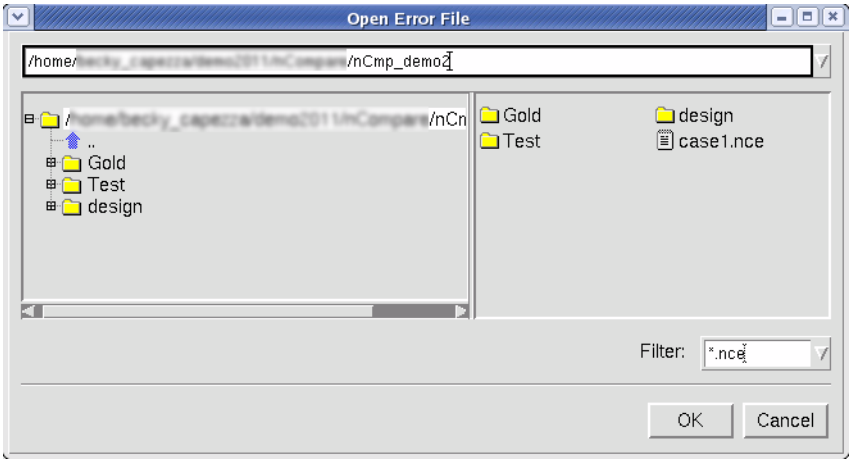


Figure: Open Error File Form

For the imported error report file, the *nCompare* window displays the mismatch errors on a **Comparison Error** tab either by design hierarchy or by time sequence. The imported file name is also displayed on the title of the *nCompare* window.

Save Error

Menu Bar: File -> Save Error

This command saves the current mismatch error as an error report file. The default extension name is *.nce*.

The mismatch error results are saved in an ASCII file format so a general text editor or viewer can be used to browse it. The error report file can also be loaded into the *nCompare* window. If the error report file name is not specified, then this command is disabled. Use the **Save Error As** command to specify the output file name and the location first.

Save Error As

Menu Bar: File -> Save Error As

This command saves the current mismatch errors with the name specified in the *Save Error As* form.

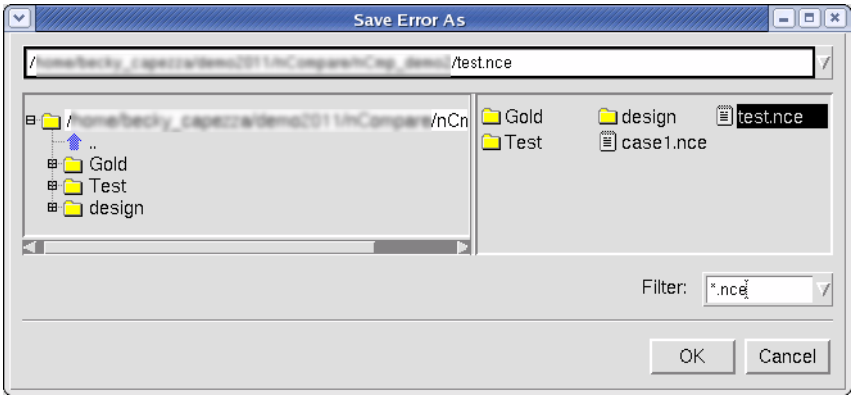


Figure: Save Error As Form

View Comparison Log

Menu Bar: File -> View Comparison Log

This command opens the *View Comparison Log* form where a quick summary of the comparison results can be viewed.

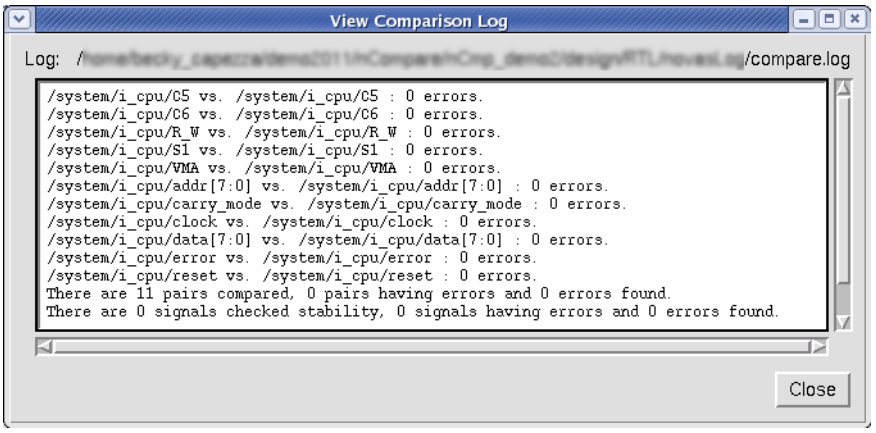


Figure: View Comparison Log Form

Exit

Menu Bar: File -> Exit

This command displays a *Question* dialog box, which verifies the request to close the *nCompare* window.

Edit Commands

NOTE: This menu is displayed when a **Rule File** tab is selected.

Undo

Menu Bar: Edit -> Undo

Toolbar Button: 

This command reverses the last action performed, if possible. To undo an action, invoke the **Edit -> Redo** command.

Redo

Menu Bar: Edit -> Redo

Toolbar Button: 

This command repeats the last action performed, if possible.

Cut

Menu Bar: Edit -> Cut

Toolbar Button: 

This command removes the selected section and places a copy on the clipboard.

Copy

Menu Bar: Edit -> Copy

Toolbar Button: 

This command places a copy of the current selection on the clipboard.

Paste

Menu Bar: Edit -> Paste

Toolbar Button: 

This command places the information cut or copied to the clipboard into the **Rule File** tab at the current cursor position.

Comparison Commands

Run

Menu Bar: Comparison -> Run

Toolbar Button: 

Bind Key: Ctrl+R

This command starts the waveform comparison.

In run mode, the mismatch error count is displayed on the FSDB node  so the accurate time to pause or to stop the waveform comparison can be determined.

Information (for example, rule file syntax, commands that did not produce results, or signals that cannot be found) about the comparison run is listed in the **Errors** tab and in the **Rules File** row of the **Comparison Error** tab.

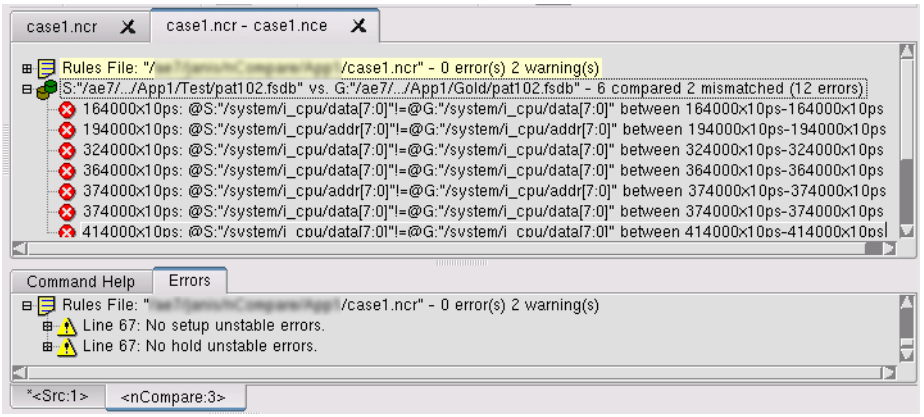


Figure: Run Error Information

Pause

Menu Bar: Comparison -> Pause

Toolbar Button: 

Bind Key: Ctrl+P

This command temporarily stops waveform comparison. Executing this command again resumes the comparison process.

Stop

Menu Bar: Comparison -> Stop

Toolbar Button: 

Bind Key: Ctrl+C

This command stops the waveform comparison.

View Commands

NOTE: This menu appears when a **Comparison Error** tab is selected.

First Error

Menu Bar: View -> First Error

Toolbar Button: 

Bind Key: Ctrl+Home

This command jumps to the first mismatch error. If the **Tools -> Sync Waveform View** option is turned *on*, this command changes the cursor time of the waveform tool to the corresponding mismatch time.

Previous Error

Menu Bar: View -> Previous Error

Toolbar Button: 

Bind Key: Ctrl+Up Arrow

This command jumps to the previous mismatch error. If the **Tools -> Sync Waveform View** option is turned *on*, this command changes the cursor time of the waveform tool to the corresponding mismatch time.

Next Error

Menu Bar: View -> Next Error

Toolbar Button: 

Bind Key: Ctrl+Down Arrow

This command jumps to the next mismatch error. If the **Tools -> Sync Waveform View** option is turned *on*, this command changes the cursor time of the waveform tool to the corresponding mismatch time.

Last Error

Menu Bar: View -> Last Error

Toolbar Button: 

Bind Key: Ctrl+End

This command jumps to the last mismatch error. If the **Tools -> Sync Waveform View** option is turned *on*, this command changes the cursor time of the waveform tool to the corresponding mismatch time.

Expand All

Menu Bar: View -> Expand All

This command expands all nodes listed on the active result tab.

Collapse All

Menu Bar: View -> Collapse All

This command collapses all nodes on the active result tab.

View by Hierarchy

Menu Bar: View -> View by Hierarchy

Toolbar Button: 

This command sorts the mismatch results by design hierarchy. If the design hierarchy of the mismatch errors is equal, the *nCompare* module sorts them by the time sequence.

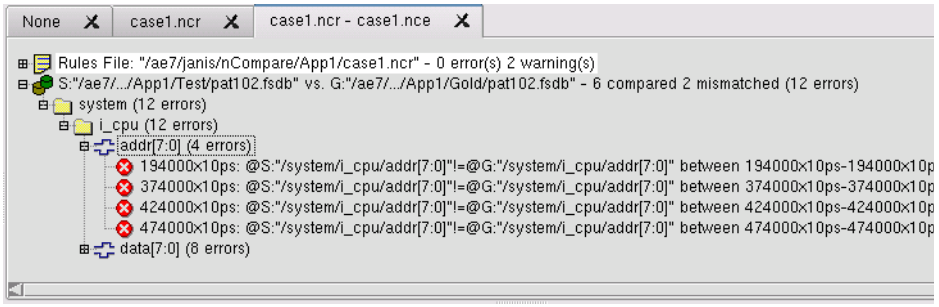


Figure: View by Hierarchy

View by Time

Menu Bar: View -> View by Time

Toolbar Button: 

This command sorts out the mismatch results by time sequence. If the error times of mismatch errors are equal, the *nCompare* module sorts them by design hierarchy.

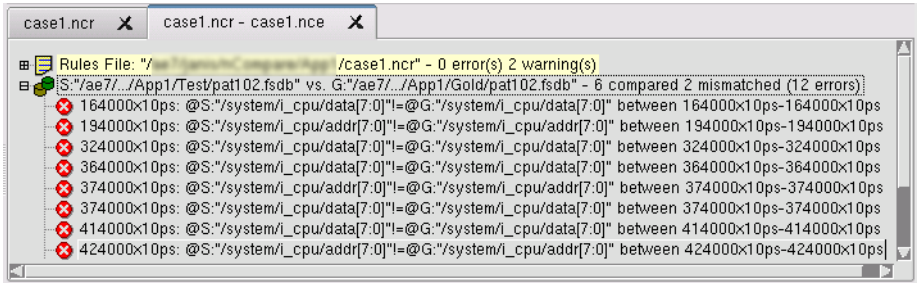


Figure: View by Time

View on Waveform

These commands display errors in the *nWave* window.

Selected Error

Menu Bar: View -> View on Waveform -> Selected Error

This command opens the *nWave* window (if it is not already open), adds the current selected mismatch signals into the *Waveform* pane, and changes the cursor time of the *nWave* window to the mismatch time. Alternatively, the mismatch error node can be double-clicked to add the selected error to the *Waveform* pane.

All Errors

Menu Bar: View -> View on Waveform -> All Errors

This command opens the *nWave* window (if it is not open previously) and adds all mismatch signals into the *Waveform* pane.

Properties

Menu Bar: View -> Properties

This command opens the *Properties* form and displays the properties of the selected mismatches. The properties include the name of the compared golden/secondary FSDB files, the compared golden/secondary signals, and the mismatch time range and comparison options.

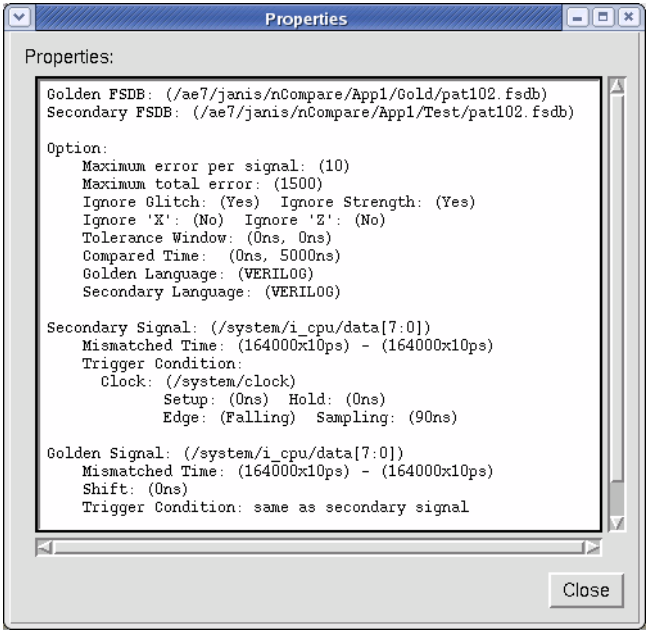


Figure: Properties Form

Tools Commands

New Waveform

Menu Bar: Tools -> New Waveform

Toolbar Button: 

This command opens an *nWave* window and enables communication with the *nCompare* window so the mismatch error can be displayed on the *Waveform* pane through the *nCompare* window. You can double-click a mismatch error node to launch the waveform tool, which adds the mismatch signals into the waveform tool and changes the cursor time of the *Waveform* pane to the mismatch time.

Sync Waveform View

Menu Bar: Tools -> Sync Waveform View

Toolbar Button: 

This toggle command turns the synchronization with the *nWave* window *on* or *off*. When this option is turned *on* (checked), the selected error result in the *nCompare* module is the current selected error on the *Waveform* pane. If this option is turned *off* (unchecked), double-click of to add the error to the *Waveform* pane.

Filter

Menu Bar: Tools -> Filter

Bind Key: Ctrl+F

This command opens the *Filter* form where the filtering options can be specified. After specifying the filter conditions and selecting **OK**, the results matching the criteria is highlighted in green in the **Comparison Error** tab.

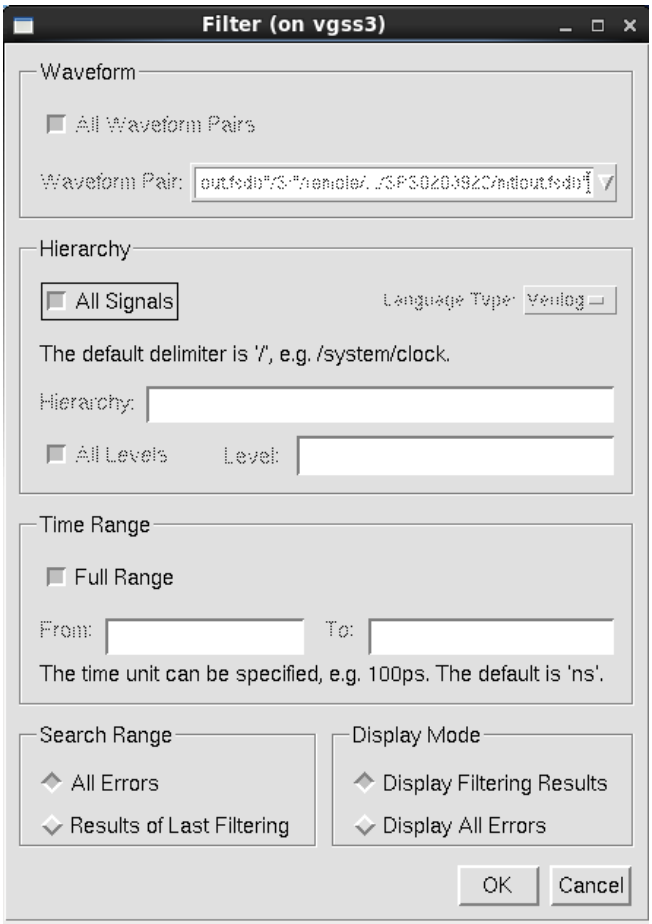


Figure: Filter Form

The following options are available in the **Waveform** section:

- **All Waveform Pairs**: When this option is turned *on*, all waveform pairs in the **Comparison Error** tab appear. When this option is turned *off*, the desired waveform pair can be selected. By default, this option is *on*.
- **Waveform Pair**: Select the required waveform pair from the list. Only the selected results appear in the **Comparison Error** tab. The **All Waveform Pairs** option must be turned *off* to activate this selection field.

The following options are available in the **Hierarchy** section:

- **All Signals**: When this option is turned *on*, all signal results are displayed. When this option is turned *off*, the required signals for display can be specified. By default, this option is *on*.
- **Language Type**: When the **All Signals** option is turned *off*, you can filter **Verilog** or **VHDL** signal types.
- **Hierarchy**: When the **All Signals** option is turned *off*, specify the scope to display results.
- **All Levels**: When this option is turned *on*, all signals for all levels of the specified scope are displayed. When this option is turned *off*, specify the desired level in the **Level** field. By default, this option is *on*. The **All Signals** option must be turned *off* to activate this option.
- **Level**: When the **All Levels** option is turned *off*, specify the level. Signals at the specified level for the specified scope are displayed.

The following options are available in the **Time Range** section:

- **Full Range**: When this option is turned *on*, results for the entire time range are displayed. When this option is turned *off*, the required time range can be specified. By default, this option is *on*.
- **From/To**: When the **Full Range** option is turned *off*, specify the start time range in the **From** field and the end time range in the **To** field.

The following options are available in the **Search Range** section (only one may be selected at a time):

- **All Errors**: When this option is selected, filter the results based on all errors. By default, this option is selected.
- **Results of Last Filtering**: When this option is selected, filter the results based on the last filter.

The following options are available in the **Display Mode** section (only one can be selected at a time):

- **Display Filtering Results**: When this option is selected, only the filtered results are displayed. By default, this option is selected.
- **Display All Errors**: When this option is selected, all results are displayed in the **Comparison Error** tab and the results matching the filter are highlighted in green.

Reset Filtering Results

Menu Bar: Tools -> Reset Filtering Results

This command resets the filtering results and displays all the mismatch errors.

Preferences

Menu Bar: Tools -> Preferences

This command displays a *Preferences* form for specifying the settings on the **Editor Window**, **Error Window**, and **Waveform Tool** tabs.

Editor Window Tab

This tab associates different colors and fonts with objects selected in the main display area.

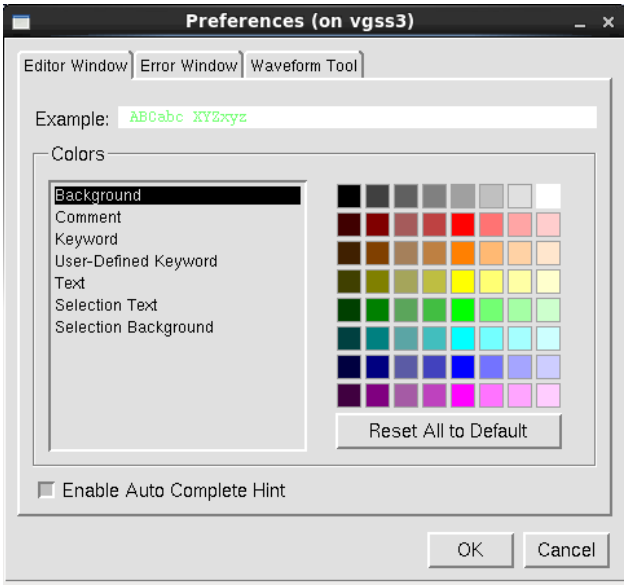


Figure: Editor Window Tab

The following fields and options are available:

- **Example:** Displays an example of **Font** and **Color** for the selected type which includes **Background**, **Comment**, **Keyword**, **User-defined Keyword**, **Text**, **Selection Text**, and **Selection Background**.
- **Reset All to Default:** Returns the selected type to its default color and font settings.
- **Colors:** Select the color of choice for **Background, Comment, Keyword, User-defined Keyword, Text, Selection Text,** or **Selection Background** in the color matrix.
- **Enable Auto Complete Hint:** When this option is turned *on*, a hint for auto completion is displayed.

Error Window Tab

This tab provides the attributes for the error results.

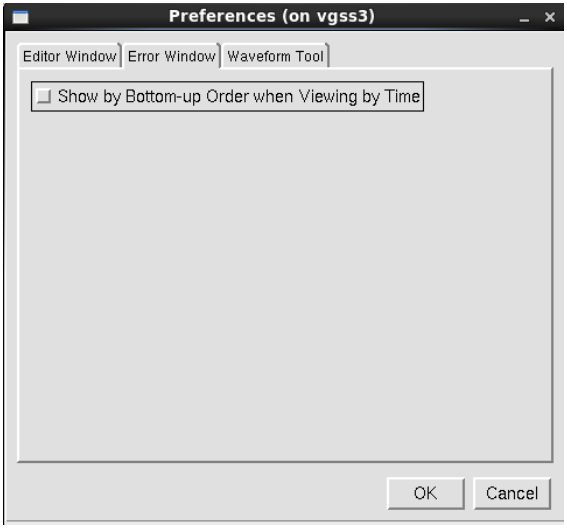


Figure: Error Window Tab

When the **Show by Bottom-up Order when Viewing by Time** option is turned *on*, *nCompare* displays the mismatches in bottom-up order. When the **Show by Bottom-up Order when Viewing by Time** option is turned *off*, *nCompare* displays the mismatches from the top level. If the level is the same, mismatches are displayed in alphabetical order. By default, the option is *off*.

Waveform Tool Tab

This tab specifies the default waveform viewer for displaying errors from the report.

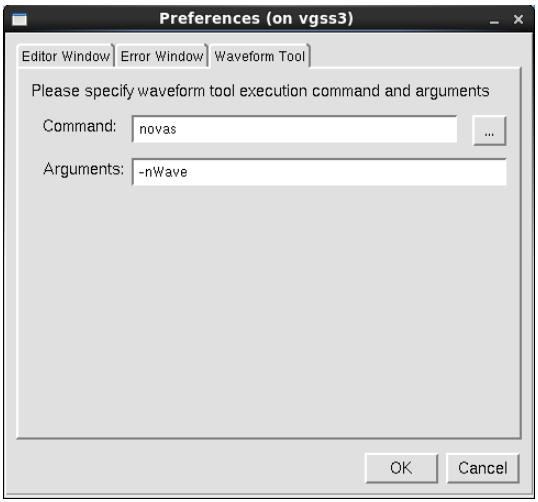


Figure: Waveform Tool Tab

The following fields are available:

- **Command:** Specify the executable file for viewing waveforms. Click the **...** button to open the *Choose Waveform* form to view the directory structure and load a previously saved error rule file.
- **Arguments: -nWave** is displayed in this field as a default argument and cannot be changed. Double-click on an error rule in the **Comparison Error** tab of the *nCompare* window to view the rule in the *nWave* window.

Customize Menu/Toolbar

Refer to the **Tools -> Customize Menu/Toolbar** command in the *nTrace* chapter for details.

Right-Click Commands

Many of the previously described commands can also be selected from the right-click menu options on the input or output nodes in the flow views.

Register Frame Right-Click Options

When the right-mouse button is clicked anywhere in the menu, toolbar icon, or frame banner area of the *nCompare* window or standalone window, a configuration option menu is displayed.This menu can be used to configure the available icons.

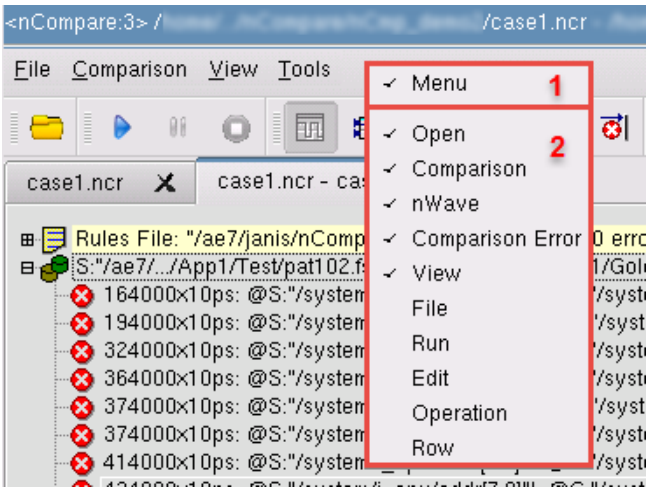


Figure: Configuration Option Menu

The **Menu** option (labeled 1 in the figure above) enables/disables the command menu bar. The options in the bottom section (labeled 2 in the figure above) enable/disable the toolbar icons for different functions.

Comparison Error Tab Right-Click Options

Selected Error

Refer to the **View -> View on Waveform ->** **Selected Error** command description for details.

Expand All

Refer to the **View -> Expand All** command description for details.

Expand

This command expands the selected result node.

Collapse

This command collapses the selected result node.

Properties

Refer to the **View -> Properties** command description for details.

Command Help Tab Right-Click Options

Copy

Bind Key: Ctrl+C

This command places a copy of the current selection on the clipboard. The copy of the selection can be pasted to the current cursor location in the active **Rule File** tab with the **Ctrl+V** bind key.

Select All

Bind Key: Ctrl+A

This command selects the command syntax currently displayed on the **Command Help** tab. The syntax is added to the tab by selecting the desired command from the list in the **Commands** field on the toolbar.

Toolbar Icons and Fields

In the *nCompare* window, there are separate toolbars for the **Rule File** and **Comparison Error** tabs.

Figure: Toolbar Used in the nCompare Rule File Tab

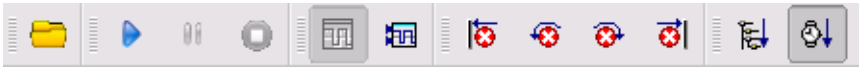


Figure: Toolbar Used in the nCompare Comparison Error Tab

The available toolbar icons may be modified. Refer to the *Toolbars* section of the *User Interface* chapter in the *Verdi User Guide and Tutorial* manual for details.

The different toolbar categories and available icons are described below.

Open Category

Open Rule File

Refer to the **File -> Open Rule File** command description for details. This icon is also in the **File Category**.

File Category

Save Rule File

Refer to the **File -> Save Rule File** command description for details.

Comparison Category

Run

Refer to the **Comparison -> Run** command description for details. This icon is also in the **Run Category**.

Pause

Refer to the **Comparison -> Pause** command description for details.

Stop 

Refer to the **Comparison** -> **Stop** command description for details.

nWave Category

New Waveform 

Refer to the **Tools** -> **New Waveform** command description for details.

Sync Waveform View 

Refer to the **Tools** -> **Sync Waveform View** command description for details.

Comparison Error Category

First Error 

Refer to the **View** -> **First Error** command description for details.

Previous Error 

Refer to the **View** -> **Previous Error** command description for details.

Next Error 

Refer to the **View** -> **Next Error** command description for details.

Last Error 

Refer to the **View** -> **Last Error** command description for details.

View Category

View by Hierarchy 

Refer to the **View** -> **View by Hierarchy** command description for details.

View by Time 

Refer to the **View** -> **View by Time** command description for details.

Edit Category

Cut 

Refer to the **Edit -> Cut** command description for details.

Copy 

Refer to the **Edit -> Copy** command description for details.

Paste 

Refer to the **Edit -> Paste** command description for details.

Operation Category

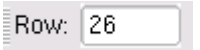
Undo 

Refer to the **Edit -> Undo** command description for details.

Redo 

Refer to the **Edit -> Redo** command description for details.

Row Category

Row 

This field has two functions. The first function is to identify the current row location of the cursor. The second function is to specify a row number to move to that row by entering a new value.

Commands 

This field contains a list of available *nCompare* rule file commands. After a command is selected, the command name is added to the active rule file at the current cursor location. In addition, the syntax for the rule is displayed on the **Command Help** tab in the bottom pane, as illustrated in the following figure.

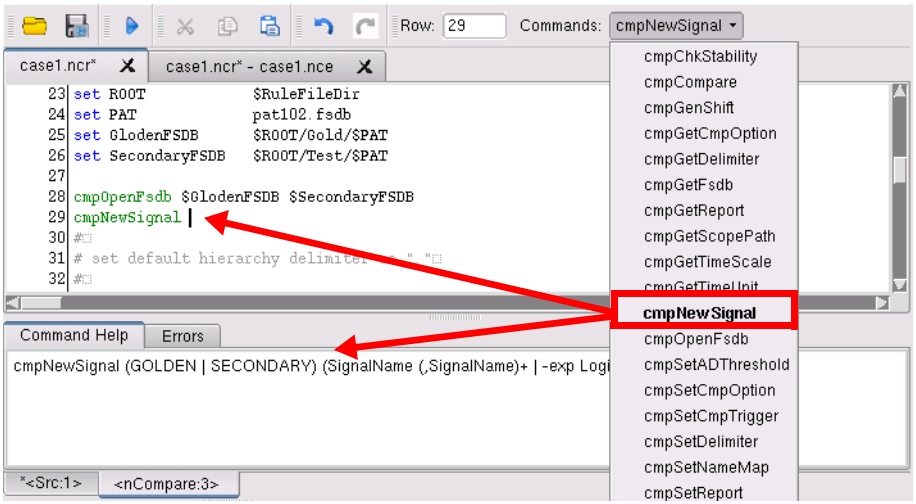


Figure: Command Help

If the syntax for a command is entered incorrectly in the rule file, an error is listed on the **Errors** tab in the bottom pane when the comparison is run.

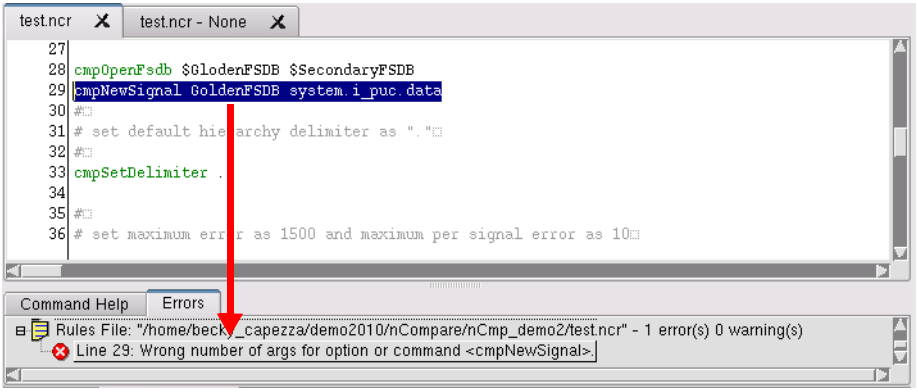


Figure: Errors Tab

Visibility

NOTE: The **Visibility** commands can be invoked from the **Tools** menu of the *nTrace* window.

NOTE: All **Visibility** commands listed in this chapter require a Siloti license and some commands may require other additional licenses as mentioned in the description.

Essential Signal Analysis

Generate ESDB

Menu Bar: **Tools -> Visibility -> Essential Signal Analysis -> Generate ESDB**

This command opens the *Generate ESDB* form where the options for the Essential Signal Database (ESDB) are specified. The **Generate ESDB** command is useful when primary inputs and registers are associated with the scope of interest.

After selecting the preferred options, click the **OK** button to close the *Generate ESDB* form and generate the Essential Signal list. Alternately, click the **Cancel** button to close the *Generate ESDB* form without generating the Essential Signal list and leave the entries intact. Essential signals are signals that are not expandable with the Data Expansion (DE) engine. Essential signals include, but are not limited to, input ports of the given scope, signals inside 'black boxes', registers, and signals being forced.

NOTE: An equivalent *esa* utility command for batch mode execution is written to the *esa.command* file when the **OK** button is clicked.

NOTE: When a module is set as a black box on the **Behavior Analysis** folder **-> Black Box** page of the *Preferences* form (invoked with the **Tools -> Preferences** command), the Essential Signal Analysis engine dumps the boundary signals and all the internal signals of the module to the Essential Signal list.

NOTE: If a signal is forced, it is included in the Essential Signal list as are all of its equivalent nets.

The *Generate ESDB* form has four tabs: **Main**, **Exclude Signals**, **Additional Signals**, and **Optimization** and the following option:

- **Save ESA Settings**: When this option is turned *on*, all Essential Signal Analysis settings are saved in the *novas.rc* resource file. When this option is turned *off,* the details are not saved. The default value of this option is *on*.

Main Tab

The **Main** tab is used to specify the options for Essential Signal Analysis.

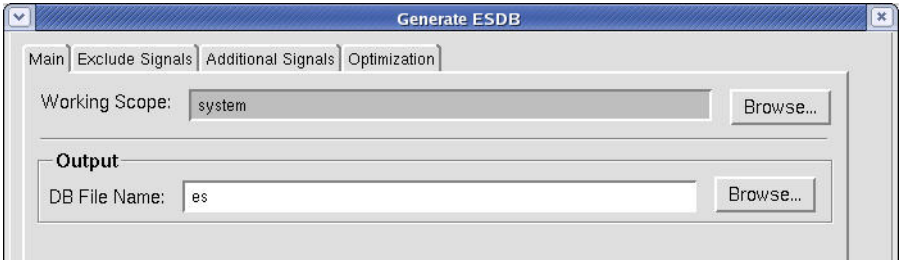


Figure: Generate ESDB Form - Main Tab

Working Scope: When a working scope is specified, the Essential Signal Analysis engine analyzes the design starting with this scope and everything below this scope.

To change the working scope, click the **Browse** button to open the *Add Scope* form. Click the icon to add the selected scope, click the icon to remove the selected scope, or click the icon to remove all scopes. After changing the scope, press the **OK** button in the *Add Scope* form to update the working scope in the *Generate ESDB* form. The working scope can also be entered by directly typing in the text field.

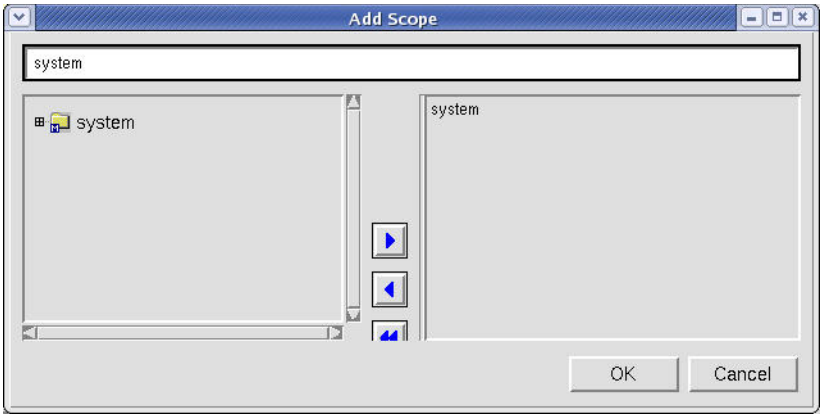


Figure: Add Scope Form

The working scope is saved to the *novas.rc* resource file; so, on subsequent invocations of the Verdi and Siloti system with the same resource file, the default value of the working scope is set to the last value the field contained when the system was exited.

NOTE: The ESDB working scope should contain all scopes needed by the *$fsdbDumpvars* dump command. For example, if *$fsdbDumpvars(0, dut1)* and *$fsdbDumpvars(0, dut2)* are used, the ESDB scope must contain both "dut1" and "dut2".

The **Output** section has the following field:

- **DB File Name**: Enter the file name to output the database to in the option text field directly or click the **Browse** button to open the *File Manager for DB Format Output* form to view the directory structure and save the Essential Signal list to the specified file name.

Exclude Signals Tab

The **Exclude Signals** tab is used to specify modules or signals to be excluded from the Essential Signal list.

NOTE: The exclude options on the **Exclude Signals** tab take precedence over any add options on the **Additional Signals** tab.

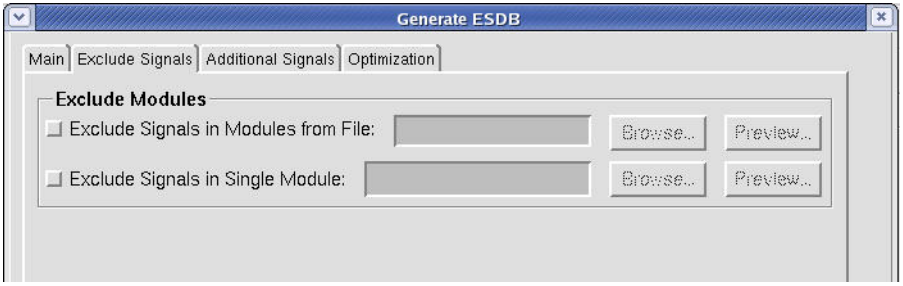


Figure: Generate ESDB Form - Exclude Signals Tab

The **Exclude Modules** section has the following options (multiple options may be selected at a time; these are optional):

- **Exclude Signals in Modules from File**: When this option is turned *on*, all signals inside the modules listed in the specified file are excluded from the Essential Signal list. Only signals driven by output or inout ports of an instantiation of these modules are included in the Essential Signal list.

  Enter the file name directly or click the **Browse** button to open the *File Manager for Exclude Modules* form where the directory structure can be viewed and an instance list can be loaded. Click the **OK** button to enter the file name and close the *File Manager for Exclude Modules* form when the selection is complete or click the **Cancel** button to exit the form without saving the file name.

  In the module file, each module is listed on a separate line. Regular expressions are allowed. For example:

  ```
  CPU
  CCU
  pram[1-2]
  alu*
  ```

  If a listed module is outside the current working scope, a warning is displayed in the *Message* frame and the module is ignored.

  Click the **Preview** button to open the *Information* form summarizing the modules that are excluded.

- **Exclude Signals in Single Module**: When this option is turned *on*, all internal signals for the specified module are excluded from the Essential Signal list. Only signals driven by output or inout ports of an instantiation of this module are included in the Essential Signal list. When this option is turned *on*, a module name must be specified.

  Enter the instance name (include the full hierarchical path) directly or click the **Browse** button to open the *Module Manager* form. In the *Module*

Manager form, browse the design hierarchy and select a scope. A white highlighted rectangle indicates the selected scope and the scope's full path is entered on the top text box. Click the **OK** button to save the scope and close the *Module Manager* form when the selection is complete or click the **Cancel** button to exit the form without saving the scope.

Regular expressions are supported for the module name.

Syntax	Expression	Description
? (question mark)	Any character	Matches any one character.
* (asterisk)	Zero or more	Finds zero or more occurrences of the preceding expression.
| (pipe)	OR	Matches the expression before or after the pipe sign "|". Mostly used within a group. For example, "[sponge]|[mud] bath" matches "sponge bath" and "mud bath".
+ (plus)	One or more	Matches at least one occurrence of the preceding expression. For example, "ba+" matches "ba", "baa", "baaa".
- (dash)	Prevent match	An example of "-Y" means to prevent a match when Y appears at this point in the expression.

Click the **Preview** button to open the *Information* form summarizing the modules that are excluded.

Additional Signals Tab

The **Additional Signals** tab is used to specify additional signals to be included in the Essential Signal list.

NOTE: The exclude options on the **Exclude Signals** tab take precedence over any add options on the **Additional Signals** tab.

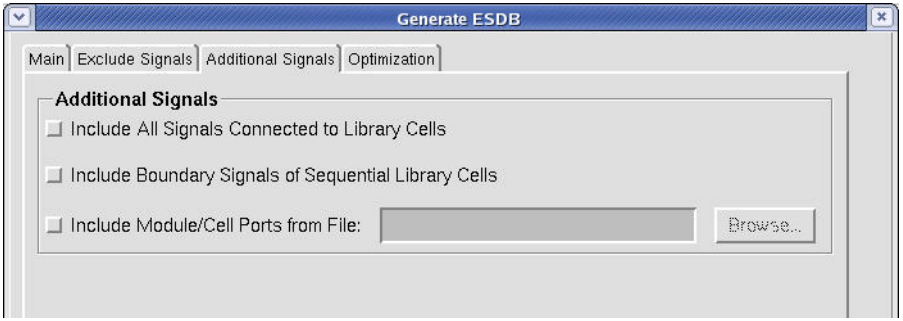


Figure: Generate ESDB Form - Additional Signals Tab

The **Additional Signals** section has the following options (multiple options may be selected at a time, these are optional):

- **Include All Signals Connected to Library Cells**: When this option is turned *on*, all signals connected to any library cell are included in the Essential Signal list. The default value of this option is *off*.
- **Include Boundary Signals of Sequential Library Cells**: When this option is turned *on*, all signals connected to sequential library cells are included in the Essential Signal list. The default value of this option is *off*.
- **Include Module/Cell Ports from File:** When this option is turned *on*, the module/cell ports in the specified file are included in the Essential Signal list. When this option is turned *on*, a file name must be specified. The default value of this option is *off*.

  Enter the file name directly or click the **Browse** button to open the *File Manager for Cell Port* form where the directory structure can be viewed and an instance list can be loaded. Click the **OK** button to enter the file name and close the *File Manager for Cell Port* form when the selection is complete or click the **Cancel** button to exit the form without saving the file name.

  In the module/cell port file, each module/cell is listed on a separate line, but multiple pins of a module/cell can be included on the same line. The module/cell name and pin name(s) are separated by spaces. Wildcard characters are not supported and # is recognized as a comment line. For example:

```
# Generic example.
# [module/cell name] [pin name] ([pin name] ...)
# Specify pin in a cell to be included.
MXSCANDFFA SIN
# Specify multiple pins in a cell to be included.
MXSCANDFFA SIN Q
```

Optimization Tab

The **Optimization** tab is used to specify an option for optimization of the Essential Signal list.

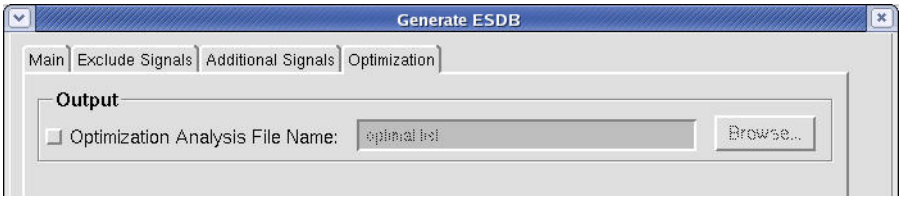


Figure: Generate ESDB Form - Optimization Tab

The **Output** section has the following option:

- **Optimization Analysis File Name**: When this option is turned *on*, the suggested modules to exclude for optimized Essential Signal Analysis results are reported in the specified text file name. The file name can be specified by entering the file name directly or clicking the **Browse** button to open the *File Manager for FSDB Clock Analysis* form and view the directory structure to select a file name. Click the **OK** button to enter the file name and close the form when the selection is complete or click the **Cancel** button to exit the form without saving the file name. The default value of this option is *off*.

  The resulting text file lists the reason for the proposed exclusion. The recommended modules may also be edited in this text file. The file can be used with the *$fsdbSuppress* dump command during simulation. The format is as follows:

  ```
  #reason(~estimated reduce-ES%)
  module
  ```

  The following are all possible reasons for an ESA-O excluded module:

  "sequential library cell,";

  "protected module,";

  "non-synthesizable module,";

  "many signals with high-frequency value change,";

  "many clock signals,";

  "many high-overhead signals,";

  "many task signals,";

  If there are modules that should not be excluded, insert a '#' at the beginning of the line and then pass the resulting file to the **“-suppress_mfile”** simulation runtime option.

Alternatively, the listed modules or the file itself could be used with one of the exclude options (module or from file) on the **Exclude Signals** tab if Essential Signal Analysis is executed again.

Data Expansion

Setup Data Expansion

Menu Bar: **Tools -> Visibility -> Data Expansion -> Setup Data Expansion**

After loading the design and an FSDB file (only the FSDB file format is supported), this command opens the *Setup Data Expansion* form where the options for data expansion can be specified, allowing the Data Expansion engine to calculate the combinational values automatically. If the FSDB file is reloaded after Data Expansion is enabled and signals are added to the waveform, the signals remain in *nWave* and the values are automatically calculated again.

After a signal is expanded, it is also displayed in the *Signal List* pane (middle column) of *nWave's Get Signals* form.

The expanded signal color in the *nWave* value pane can be changed with the **computedAnnotationColor** key in the **Type** field under the **Waveform -> Color/Font/Pattern** -> **Color** page of the *Preferences* form.

The expanded signal color for Active Annotation in *nTrace* can be changed with the **ComputedSignal** key in the **Type** field under the **Source Code** -> **Color** page of the *Preferences* form.

The Data Expansion settings are saved to (and loaded from) the *novas.rc* resource file.

NOTE: The dump-off time range is X in the Siloti-expanded FSDB file.

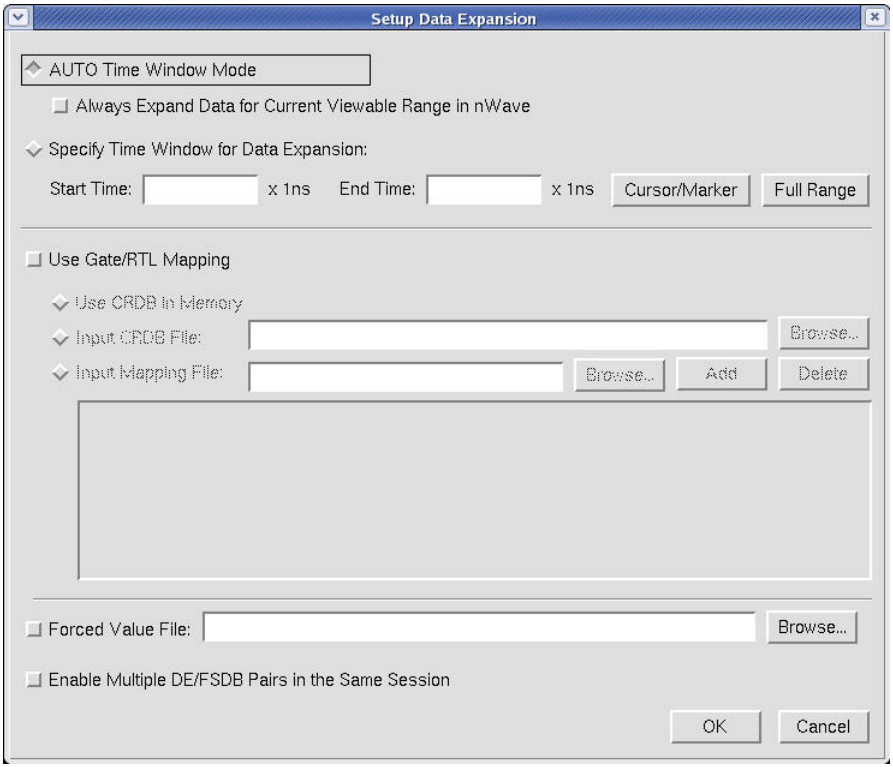


Figure: Setup Data Expansion Form

The *Setup Data Expansion* form has the following options (only one option may be selected at a time):

- **AUTO Time Window Mode**: When this option is selected, the data expansion time window is automatically determined as needed by the application (such as, a snapshot for active annotation or a signal waveform within the zoom range in *nWave*). By default, this option is selected.

NOTE: Specify the `-de` option on the Verdi command line to enable Data Expansion in auto time window mode.

- **Always Expand Data for Current Viewable Range in nWave**: When this option is turned *on*, Data Expansion automatically calculates values for the current viewable time range for all signals in the *nWave* frame. The default value of this option is *off*.

NOTE: Specify the `-DE "-de_enforce"` option on the Verdi command line to enable expansion of all signals.

NOTE: Expansion of unexpanded regions in *nWave* can be forced in the Data Expansion auto-time window mode using the following methods:
- On left-clicking any signal's unexpanded region (blue bar), the signal's waveform is displayed from the last transition to the next transition encompassing the point of selection.
- On dragging-left any signal, *nWave* zooms into that area and expand the waveforms for all signals in the waveform pane if the region is small enough.
- Select one or more signals in the signal pane, press and hold the **Ctrl** key and drag-left on the top ruler to expand the selected signals for that time range.
- Press and hold the **Ctrl** key and drag-left on any signal's unexpanded region (blue bar) to expand the waveform for the time range on the selected signal.

- **Specify Time Window for Data Expansion:** When this option is selected, the following fields and buttons are available for specifying a data expansion time window. By default, this option is not selected.

NOTE: Specify **-DE "[options]"** on the Verdi command line to enable Data Expansion with a time window. Possible options are:

```
-start_time value and -end_time value
-full_time
-force file
-g2r_mapping file
-on_demand
-de_enforce
```

  - **Start Time:** Specify the start time for data expansion calculations. This field is empty by default. This field is mandatory when the **Specify Time Window for Data Expansion** option is selected.
  - **End Time:** Specify the end time for data expansion calculations. This field is empty by default. This field is mandatory when the **Specify Time Window for Data Expansion** option is selected. The end time must be greater than the start time.
  - **Cursor/Marker:** Click this button to automatically fill in the start/end times based on the cursor and marker positions in the primary *nWave* frame.

NOTE: If an *nWave* frame has not been opened, the start and end times are 0 by default. If the *nWave* frame was open but is closed when the **Cursor/Marker** button is clicked, the last cursor/marker positions are used.

  - **Full Range:** Click this button to automatically fill in the start/end times based on the full time range of the loaded FSDB file.

- **Use Gate/RTL Mapping:** When this option is turned *on*, select the gate/RTL mapping source to automatically convert an FSDB file from the gate level and expand it on the RTL. When this option is turned *off*, the FSDB file should match the design source for Data Expansion (such as, the FSDB file was generated from the RTL and expanded on the RTL). This is optional and the default value of this option is *off*.

NOTE: A Correlation license is required to use this option.

Select the gate/RTL mapping source from the following options;only one option may be selected at a time:

- **Use CRDB in Memory:** Use the correlation database in memory to convert an FSDB file generated at the gate level and expand it on the RTL.
- **Input CRDB File:** Use the specified correlation database to convert an FSDB file generated at the gate level and expand it on the RTL.

  Enter the file name directly or click the **Browse** button to open the *CRDB File* form and view the directory structure to load a correlation database. Click the **OK** button to enter the file name and close the *CRDB File* form when the selection is complete or click the **Cancel** button to exit the form without saving the file name.
- **Input Mapping File:** Use the specified mapping files that were created with the **Generate Result** command to convert an FSDB file generated at the gate level and expand it on the RTL. One or more mapping files can be specified.

  Enter the file name directly or click the **Browse** button to open the *Mapping File* form and view the directory structure to load a mapping file. Click the **OK** button to enter the file name and close the *Mapping File* form when the selection is complete or click the **Cancel** button to exit the form without saving the file name.

  After a file is entered in the **Input Mapping File** field, click the **Add** button to add it to the mapping file list in the center pane. Remove a file name by left-clicking to select and then clicking the **Delete** button. Only one file can be removed at a time.

- **Forced Value File:** Specify a file name containing a list of signals and their force values. This field is optional.

  The forced value file is intended for signals whose values are not dumped to the FSDB file. This is typically required in emulation or FPGA prototyping applications where some Essential Signals may not be available in the

FSDB file. It is not intended for exploration of simulation results with different stimuli.

Enter the file name directly or click the **Browse** button to open the *Forced Value File* form and view the directory structure to load a forced value file. Click the **OK** button to enter the file name and close the *Forced Value File* form when the selection is complete or click the **Cancel** button to exit the form without saving the file name.

In the forced value file, each signal with the full hierarchical path (buses must be bit-blasted) and its force value (such as, 1'b0, 1'b1, 1'bx, 1'bz, or another signal) separated by => is listed on a separate line. Regular expressions are not allowed. For example:

```
system.i_cpu.b[0] => 1'b0
system.i_cpu.carry => 1'bx
system.i_cpu.b[1] => system.i_cpu.b[2]
```

Data Expansion uses the value on the right-hand side to set the value for the left-hand side signal. The signal is then treated as a dumped signal and used to expand other computable signals within its logic cone. The specified value is expected to be the same as the full simulation run. Data Expansion ignores the force value if the signal is already dumped in the FSDB file.

- **Enable Multiple DE/FSDB Pairs in the Same Session:** When this option is turned *on*, Data Expansion can be performed on all Essential Signal Dump (ESD) FSDB files that are loaded into the same session. The default value of this option is *off*.

Enable Data Expansion

Menu Bar:	**Tools** -> **Visibility** -> **Data Expansion** -> **Enable Data Expansion**
Toolbar Icon:	 This icon indicates the current mode is disabled.

When this command is selected, Behavior Analysis is performed, Data Expansion is enabled, and new signal values are calculated. The design and an FSDB file must be loaded for this command to be activated. The command uses the Data Expansion options from the last invocation of the **Visibility -> Data Expansion -> Setup Data Expansion** command. If **Setup Data Expansion** has not been executed, the command uses auto time window mode by default.

NOTE: This is a toggle command. The next time the Visibility menu is accessed, this command is replaced with the **Visibility -> Data Expansion -> Disable Data Expansion** command.

Disable Data Expansion

Menu Bar: **Tools** -> **Visibility** -> **Data Expansion** -> **Disable Data Expansion**

Toolbar Icon:  This icon indicates the current mode is enabled.

When this command is selected, Data Expansion is disabled and new signal values are not calculated. The design and an FSDB file must be loaded for this command to be activated.

NOTE: This is a toggle command. The next time the Visibility menu is accessed, this command is replaced with the **Visibility -> Data Expansion -> Enable Data Expansion** command.

Refresh Signal Values

Menu Bar: **Tools** -> **Visibility** -> **Data Expansion** -> **Refresh Signal Values**

Toolbar Icon: 

When this command is selected, all signal values are refreshed including those calculated by Data Expansion. The design and an FSDB file must be loaded for this command to be activated.

Gate/RTL Correlation

Open Slave Process

Menu Bar: **Tools -> Visibility -> Gate/RTL Correlation -> Open Slave Process**

This command opens the *Open Slave Process* form where the environment for the secondary (gate) process can be set up. After specifying the pertinent information, click the **OK** button to close the form and open another process or click the **Cancel** button to close the form without starting another process and leave the entries intact.

NOTE: This command is available in the primary process.

NOTE: A Correlation license is required to use this command.

NOTE: The second process has reduced functionality. Only correlation commands are available; however, Verdi debug mode can be enabled.

NOTE: The RTL design must be loaded in the primary (original) process and the gate-level design in the second process.

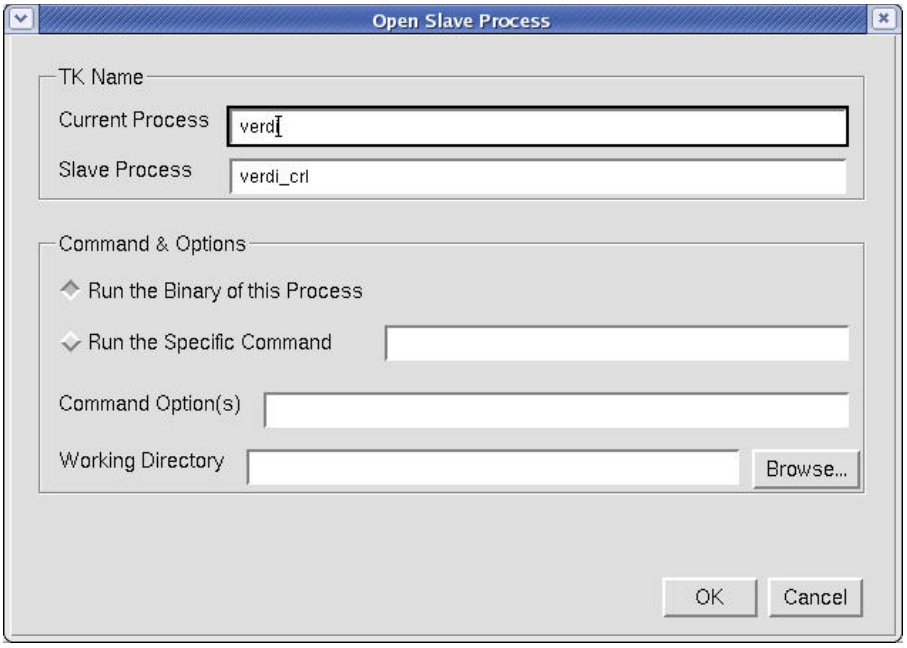


Figure: Open Slave Process Form

The **TK Name** section has the following fields:

- **Current Process:** Specify the name for the current process in this field. This field is required. By default, value of this field is *novas*. If multiple sessions are running, a #<number> may be added (such as, *novas#2*). The default name can be changed.
- **Slave Process:** Specify the name for the slave process in this field. This field is required. By default, value of this field is *novas_crl*. If multiple sessions are running, a #<number> may be added (*novas#2_crl*). The default name can be changed.

The **Command & Options** section has the following options; only one option may be selected at a time.

- **Run the Binary of this Process:** When this option is selected, the same binary (path and executable) is used for the new process. When this option is selected, specify the preferred Verdi command line options (such as, `-f run.f`, `-logdir slave`) in the **Command Option(s)** field. If no command line options are specified, the new process window is empty.
- **Run the Specific Command**: When this option is selected, the Verdi command (path and executable) to use for the new process must be specified in this text field. When this option is selected, the preferred Verdi

command line options (such as, `-f run.f`, `-logdir slave`) can also be specified in the corresponding text field or in the **Command Option(s)** field.

The **Command & Options** section also has the following fields:

- **Command Option(s):** Specify the preferred Verdi command line options in this field. Refer to the *Utilities* chapter for a complete list. At a minimum, `-f <filename>` should be specified, where *filename* signifies the design source files.
- **Working Directory:** Specify the working directory for the new process. If this field is blank, the current working directory is assumed as the working directory.

  Enter the path to the working directory directly or click the **Browse** button to open the *Directory File Manager* form and view the directory structure to locate the new working directory. Click the **OK** button to enter the path and close the *Directory File Manager* form when the selection is complete or click the **Cancel** button to exit the form without saving the path.

Prepare CRDB

Menu Bar: **Tools -> Visibility -> Gate/RTL Correlation -> Prepare CRDB**

This command opens the *Prepare CRDB* form where different naming rules for correlation can be specified and the correlation database created. Design extraction analyzes the design starting with the working scope (specified on the *Behavior Analysis* form) and everything below the working scope.

NOTE: This command is available in the primary process.

NOTE: A Correlation license is required to use this command.

NOTE: After clicking **Finish**, any changes to the *Prepare CRDB* form are automatically saved to the correlation database when the **Correlation -> Gate/RTL Correlation -> Save CRDB File** command is executed. These changes are loaded when the correlation database is restored with the **Correlation -> Gate/RTL Correlation -> Load CRDB File** command.

After modifying fields on any of the tabs, click the **Finish** button to perform correlation, or click the **Exit** button to close the form (keeping the entries intact until the session is exited) without performing correlation.

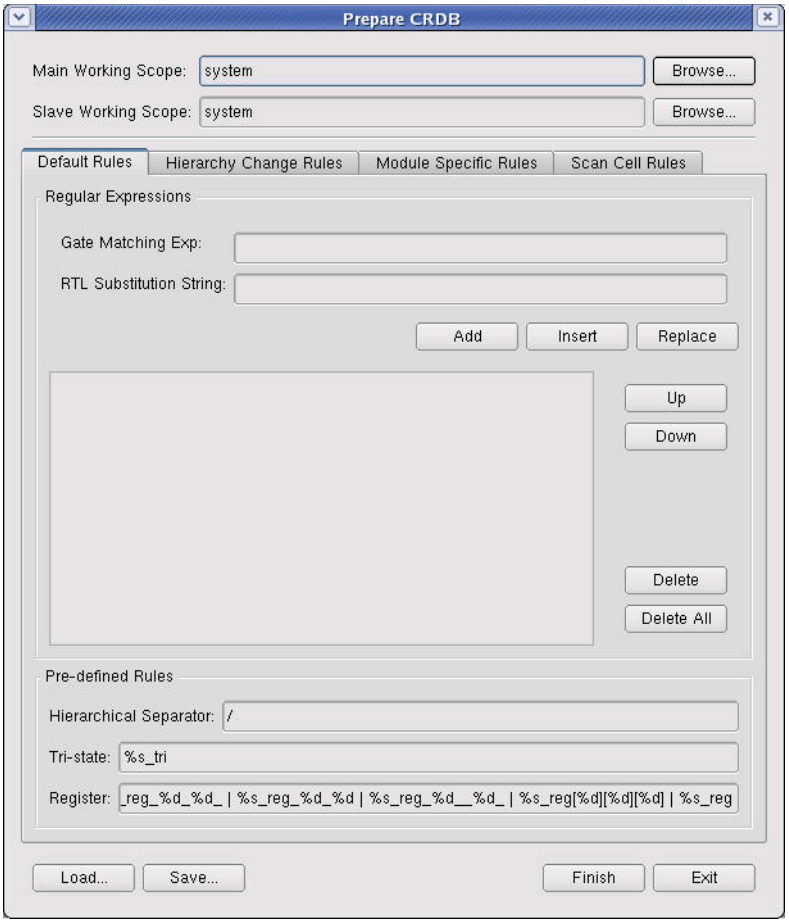


Figure: Prepare CRDB Form - Default Rules Tab

Main Working Scope: Specify the working scope for the design loaded in the primary process in this field. Behavior Analysis analyzes the design starting with this scope and everything below this scope. The default working scope is the last scope specified for Behavior Analysis either by executing the **Tools -> Temporal Flow View -> Behavior Analysis** command or as saved in the *novas.rc* resource file.

To change the working scope, click the **Browse** button to open the *Instance Manager* form. After changing the scope, press **OK** to update the working scope in the *Generate ESDB form*. The working scope can also be entered by directly typing in the text field.

Slave Working Scope: Specify the working scope for the design loaded in the secondary (slave) process in this field. Behavior Analysis analyzes the design

starting with this scope and everything below this scope. The default working scope is the same as the **Main Working Scope** text field which is the last scope specified for Behavior Analysis either by executing the **Tools -> Temporal Flow View -> Behavior Analysis** command or as saved in the *novas.rc* resource file.

To change the working scope, click the **Browse** button to open the *Instance Manager* form. After changing the scope, press **OK** to update the working scope in the *Generate ESDB* form. The working scope can also be entered by directly typing in the text field.

Load: To specify a naming rule file to load, click this button to open the *Load Naming Rules* form.

Enter the file name (including the full path) to load in the file name text field directly or view the directory structure (left pane) to locate the naming rule file (right pane) to load. The default file name is *naming_rules.lst* in the current directory. Click the **Open** button to enter the file name and close the form when the selection is complete or click the **Cancel** button to exit the form without saving the file name.

Save: To specify the name of the text file in which to save the mapping rules, click this button to open the *Save Naming Rules* form.

Enter the file name (including the full path) in the text field directly or view the directory structure to find the location to save the specified file name. The default file name is *naming_rules.lst* in the current directory. Click the **Save** button to save the file name and close the form or click the **Cancel** button to exit the form without saving the file name.

- **Save as Preference:** When this option is turned *on*, the naming rules are saved in the *novas.rc* resource file. When this option is turned *off*, the details are not saved. The default value is *off*.

There are five tabs to select from for specifying name mapping rules for the design.

Default Rules Tab

The *Regular Expressions* section is used to specify name-based mapping rules which are composed of a matching string and a substitution string. The matching string specifies the regular expression for matching the input string. The substitution string uses the matched intermediates to generate an output string.

Regular Expressions work on a single design name only; they do not work for names crossing multiple hierarchies with delimiters.

The following fields are available in the **Regular Expressions** section:

- **Gate Matching Exp:** Specify the regular expression for the input matching string from the gate design in this field. The available syntax for the matching string is as follows.

a	Matches character a
abc	Matches string abc
[abc]	Matches a, b, or c
[a-e]	Matches one of the range of characters from a to e
(abc)	Matches abc and referable
ab|c	Matches ab or c
^abc	Matches abc only at the beginning of the string
abc$	Matches abc only at the end of the string
.	Matches any single character
*	Matches 0 or more characters
+	Matches 1 or more characters
%d	[0-9]+
%a	[a-zA-Z]+
%s	[0-9a-zA-Z]+
\	Escape character for % . * ^ $ | ( ) [ ] /

- **RTL Substitution String:** Specify the regular expression for the substitution string from the RTL design. The available syntax for the substitution string is as follows:

abc	Replaces each mapped string with abc
@n	Replaces with the string that matches n-th %d, %a, %s or ( ).
#(expr)	The expr keyword is any expression containing only constants, operators, and @n, which results in a constant integer.

The following buttons are available in the **Regular Expressions** section:

- **Add:** Click this button to add the specified rule to the main display area. The rule is added to the end of the list.
- **Insert:** Click this button to add the specified rule to the main display area. The rule is added above the current selected rule.

Replace: Click this button to change the content of the rule currently selected in the main display area. The selected rule is updated to match the new expressions.The mapping rules are applied sequentially in the order listed in the mapping rule table. The order of the rules can be changed.

- **Up/Down:** Select a rule in the main display area and click one of these buttons to move the rule up or down in the list.
- **Delete:** Click this button to remove the currently selected rule from the main display area.
- **Delete All:** Click this button to remove all rules form the main display area.

The *Pre-defined Rules* section is used to specify the predefined naming conventions. The following fields are available in the *Pre-defined Rules* section:

- **Hierarchical Separator:** Specify the hierarchical separator for a flattened signal/instance in this field. Only one character is allowed. The default is the forward slash (/) character.

NOTE: Common hierarchy separators such as underscores '_' and dots '.' do not need to be specified as they are handled internally. Specifying the underscore may create problems as it is commonly used in signal names.

- **Tri-state:** Specify the naming convention for tri-state instances and tri-state output nets in this field. Multiple substitutions may be specified and or'ed together using the vertical bar (|) character.
- **Register:** Specify the naming convention for register instances and register output nets only. The register substitution consists of the register string and the register index format. Multiple substitutions may be specified and or'ed together using the vertical bar (|) character.

NOTE: A rule in the **Register** or the **Tri-state** field can only contain one '%s' and it must be at the beginning of the rule. For example, the rule "%s_reg_%d" is valid but the rule "\%s_reg_%d" is not.

Hierarchy Change Rules Tab

Use this tab to specify rules for the instances that have a different hierarchical path in the gate and the RTL designs. As many rules as needed may be specified.

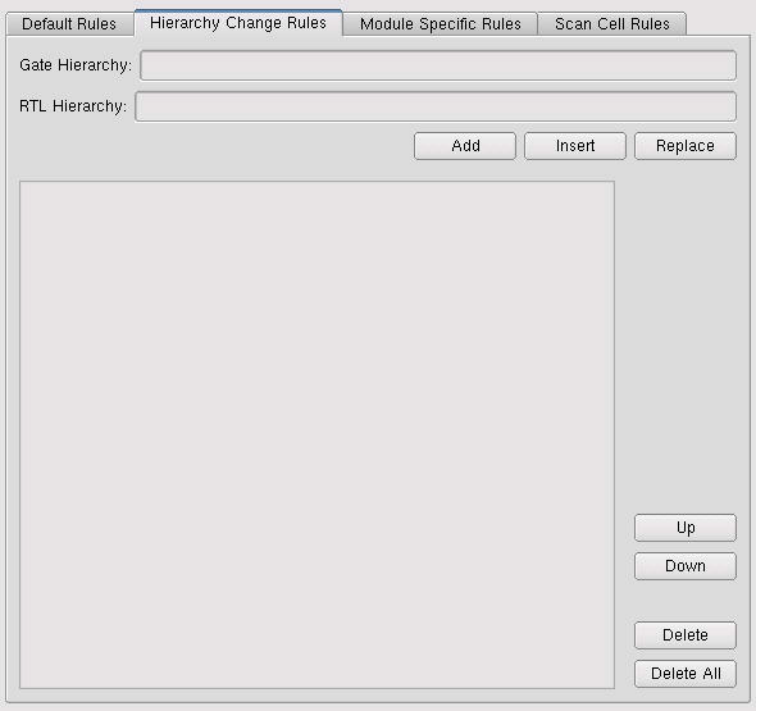


Figure: Prepare CRDB Form - Hierarchy Change Rules Tab

The following fields are available in the *Hierarchy Change Rules* tab:

- **Gate Hierarchy:** Specify the full hierarchical path for the instance that has a different hierarchical path in the RTL design and the gate netlist in this field. Enter the hierarchical path manually or drag and drop the module from the design browser frame.
- **RTL Hierarchy:** Specify the full hierarchical path for the instance in the RTL design corresponding to the instance in the gate netlist in this field. Enter the hierarchical path manually or drag and drop the module from the design browser frame.

The following buttons are available in the *Hierarchy Change Rules* tab:

- **Add:** Click this button to add the specified rule to the main display area. The rule is added to the end of the list.
- **Insert:** Click this button to add the specified rule to the main display area. The rule is added above the current selected rule.
- **Replace:** Click this button to change the content of the rule currently selected in the main display area. The selected rule is updated to match the new expressions.The hierarchy name change rules are applied sequentially

in the order listed in the main display area. The order of the rules can be changed.

- **Up/Down:** Select a rule in the main display area and click one of these buttons to move the rule up or down in the list.
- **Delete:** Click this button to remove the currently selected rule from the main display area.
- **Delete All:** Click this button to remove all rules from the main display area.

Module Specific Rules Tab

Use this tab to specify rules that are to be applied only to specific modules.

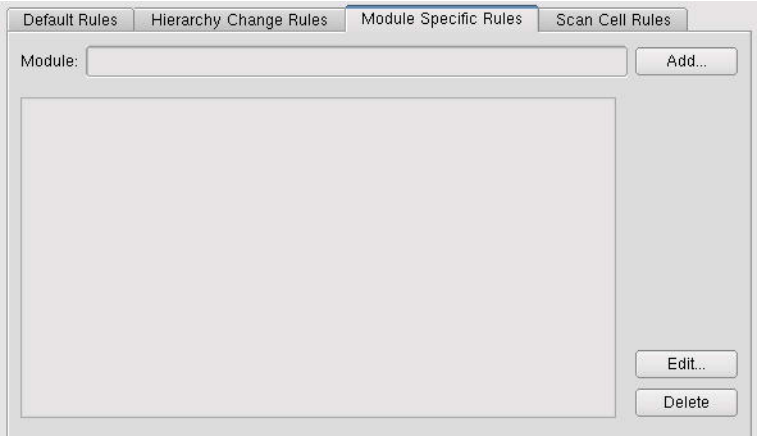


Figure: Prepare CRDB Form - Module Specific Rules Tab

The following fields are available in the *Module Specific Rules* tab:

- **Module:** Specify the module name in this text field. Enter the module name manually or drag and drop the module from the design browser frame.
- **Add:** Click this button to open the *Setup Module Specific Rules* form where the name-based mapping rules composed of a matching string and a substitution string can be specified for the module. The rules in the *Setup Module Specific Rules* form are the same as those on the **Default Rules Tab** of the *Perform CRDB Correlation* form.
- **Edit:** Click this button to open the *Setup Module Specific Rules* form where the rules for the currently selected rule can be modified in the main display area.
- **Delete:** Click this button to remove the currently selected rule from the main display area.

Scan Cell Rules Tab

Use this tab to specify rules for scan chains.

Figure: Prepare CRDB Form - Scan Cell Rules Tab

The following field/section available in the *Scan Cell Rules* tab:

- The **PortName for Scan Cell in Gate-level Netlist** section contains the following additional fields, which help the Siloti system recognize scan related signals so it stops tracing them:
  - **Clock:** Specify the naming convention for scan cell clock input nodes in this field.
  - **Scan-in:** Specify the naming convention for scan cell scan input nodes in this field.**Scan-enable:** Specify the naming convention for scan cell scan enable input nodes in this field.

Correlation Console

Menu Bar:	**Tools -> Visibility -> Gate/RTL Correlation -> Correlation Console**

This command opens the *Gate/RTL Correlation Console* where the gate/RTL correlation results are displayed and can be reviewed.

NOTE: This command is available in the primary process.

NOTE: A Correlation license is required to use this command.

The *Gate/RTL Correlation Console* form has several menus with associated toolbar icons, correlate fields, and history.

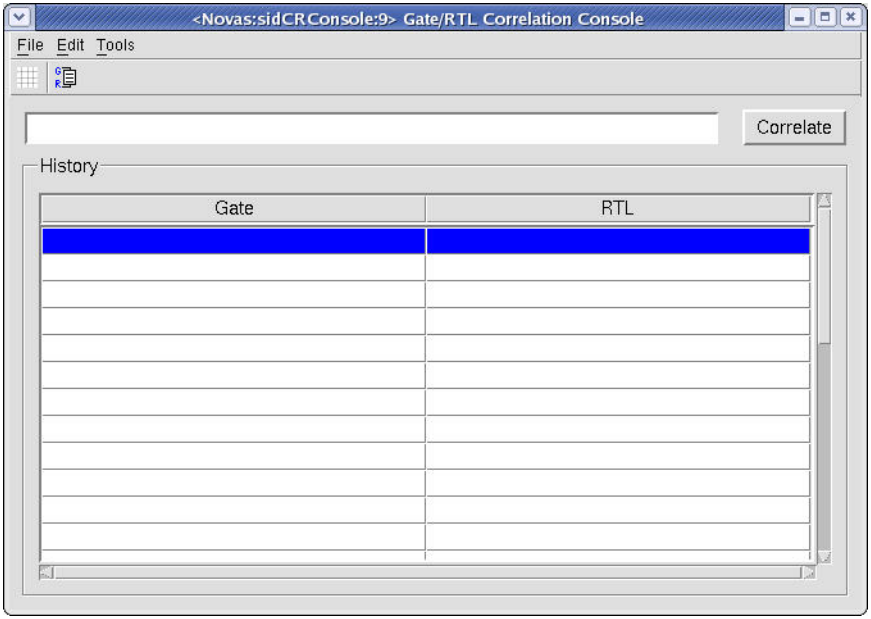


Figure: Setup Naming Rules Form

Correlate

Click the **Correlate** button to map the gate-level signal (including full hierarchy) specified in the text field to the equivalent RTL signal and vice versa. After **Correlate** is clicked, the signal name remains in the text field, the mapping results are added to the **History**, and the equivalent source code is identified in both the processes.

Enter the signal name in the text field directly or select a signal in a different Siloti/Verdi window (such as, *nWave*, *nTrace*, *nSchema*, and so on) and use the middle mouse button to drag and drop it in the text field. Automatic mapping can also be performed by dropping the signal in the **History** section.

History

The **History** section lists the results for each mapped signal. The results remain in the list until the **Clear History** toolbar icon is clicked or the session is exited. The **History** section lists the full signal path for the original signal (bold and black) and its corresponding correlated signal (bold and black for a match or blue for a non-match). If an exact signal match does not exist, the lines of RTL source code that are the likely match are listed in red color.

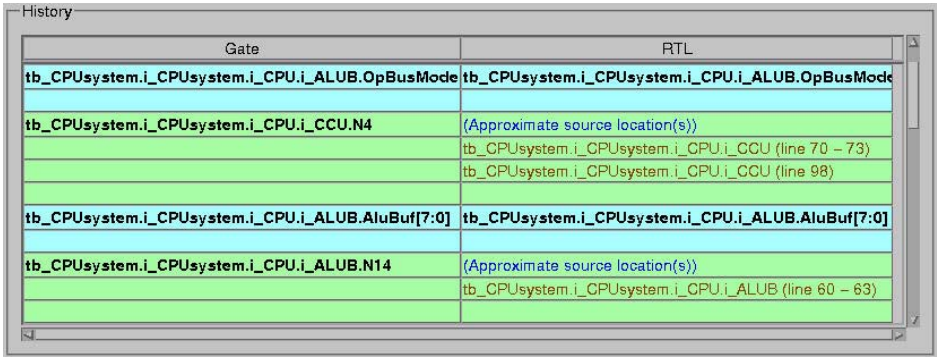


Figure: Mapping History

If a bus is selected and it correlates to multiple non-bussed signals, the results are collapsed in a single row.

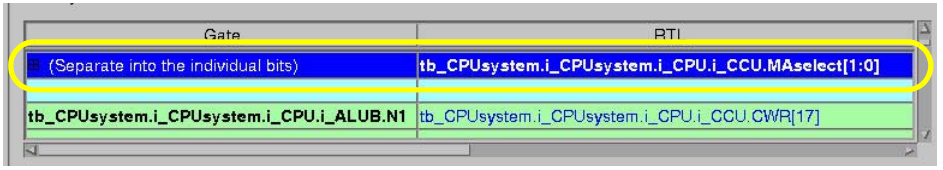


Figure: Mapping History - Bus Collapsed

Double-click the row to display the individual bit results. Double-clicking the row again collapses the row.

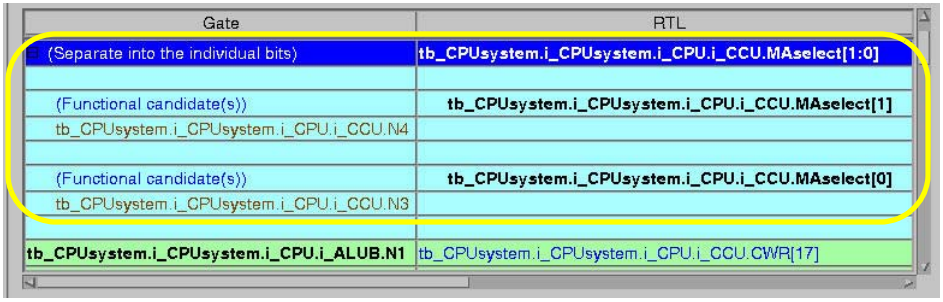


Figure: Mapping History - Bus Expanded

File

The **File** menu in the *Gate/RTL Correlation Console* form includes the following command:

Close Window

Menu Bar	*Correlation Console*, **File** -> **Close Window**

This command closes the *Gate/RTL Correlation Console* form. The correlation history is saved until the session is exited or the **Clear History** command or toolbar icon is selected.

Edit

The **Edit** menu in the *Gate/RTL Correlation Console* form includes the following command:

Clear History

Menu Bar	*Correlation Console*, **Edit** -> **Clear History**
Toolbar Icon	

This command clears the correlation results from the **History** section of the *Gate/RTL Correlation Console* form.

Tools

The **Tools** menu includes the following command:

Perform CRDB Correlation

Menu Bar	*Correlation Console*, **Tools** -> **Perform CRDB Correlation**
Toolbar Icon	

This command opens the *Perform CRDB Correlation* form so the mapping rules can be added or changed.

Generate Result

Menu Bar:	**Tools** -> **Visibility** -> **Gate/RTL Correlation** -> **Generate Result**

This command opens the *Generate Correlation Result* form where a correlation file can be created by using the specified input type as the source to correlate the RTL to the gate-level design signals. After selecting the preferred option, click the **OK** button to close the *Generate Correlation Result* form and generate the

correlation results or click the **Cancel** button to close the form without performing correlation and leave the entries intact.

The file created with the **Generate Result** command can be used as the mapping file input for the **Setup Data Expansion** command.

NOTE: This command is available in the primary process.

NOTE: A Correlation license is required to use this command.

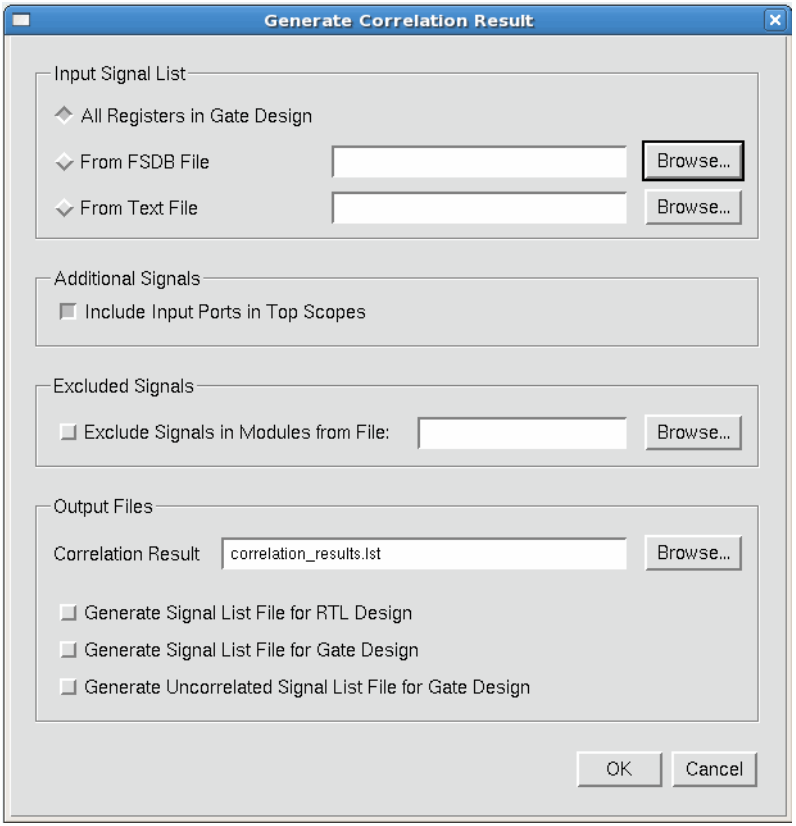


Figure: Generate Correlation Result Form

The **Input Signal List** section has the following options (only one option can be selected at a time):

- **All Registers in Gate Design:** When this option is selected, all register signals in the gate design are mapped to the RTL.

- **From FSDB File:** When this option is selected, all signals in the FSDB file are mapped from the gate to the RTL. When this option is selected, an FSDB file name must be specified.

  Enter the file name directly or click the **Browse** button to open the *File Manager for FSDB* form and view the directory structure to load an FSDB file. Click the **OK** button to enter the file name and close the *File Manager for FSDB* form when the selection is complete. Click the **Cancel** button to exit the form without saving the file name.

- **From Text File:** When this option is selected, all signals in the text file are mapped from the gate to the RTL. When this option is selected, a text file name must be specified.

  Enter the file name directly or click the **Browse** button to open the *File Manager for Text* form and view the directory structure to load a text file. Click the **OK** button to enter the file name and close the *File Manager for Text* form when the selection is complete or click the **Cancel** button to exit the form without saving the file name.

The **Additional Signals** section has the following option:

- **Include Input Ports in Top Scopes:** When this option is turned *on*, input ports that were successfully mapped from the gate-level design to the RTL design are included in the top scope. The default value of this option is *off*.

The **Excluded Signals** section has the following option:

- **Exclude Signals in Modules from File:** When this option is turned *on*, all signals except port signals in the excluded modules are excluded from the mapping file. The port signals in the excluded modules are listed in the mapping file. Click the **Browse** button to specify the file that contains modules to be excluded. The default value of this option is *off*.

The **Output Files** section has the following field and options:

- **Correlation Result:** Specify the name of the file in which to save the correlation information in this field (this field is mandatory).

  Enter the file name in the field directly or click the **Browse** button to open the *File Manager for Output* form to view the directory structure and save the Essential Signal list to the specified file name. The default file name is *correlation_results.lst*. Click the **OK** button to save the file name and close the *File Manager for Output* form or click the **Cancel** button to exit the form without saving the file name.

- **Generate Signal List File for RTL Design:** When this option is turned *on*, a list of mapped RTL signals are saved to the current directory. The file has

the same base name as that specified in the **Correlation Result** field with a **.rtl* extension.

- **Generate Signal List File for Gate Design:** When this option is turned *on*, a list of mapped gate-level signals are saved to the current directory. The file has the same base name as that specified in the **Correlation Result** field with a **.gate* extension.
- **Generate Uncorrelated Signal List File for Gate Design:** When this option is turned *on*, a list of unmapped gate-level signals are saved to the current directory. The file has the same base name as that specified in the **Correlation Result** field with a **.gate.uncor* extension.

Advanced Commands

Extract CRDB, **Perform CRDB Correlation**, **Save CRDB File**, **Load CRDB File**, and **Correlation Analyzer** commands are located under the **Advanced Commands** menu item.

Extract CRDB

Menu Bar:	**Tools -> Visibility -> Gate/RTL Correlation -> Advanced Commands -> Extract CRDB**

This command opens the *Extract Design* form where the RTL or gate correlation database can be extracted. After specifying the pertinent information, click the **OK** button to close the *Extract Design* form and perform the extraction. Click the **Cancel** button to close the form without performing the extraction.

NOTE: This command requires a Correlation license.

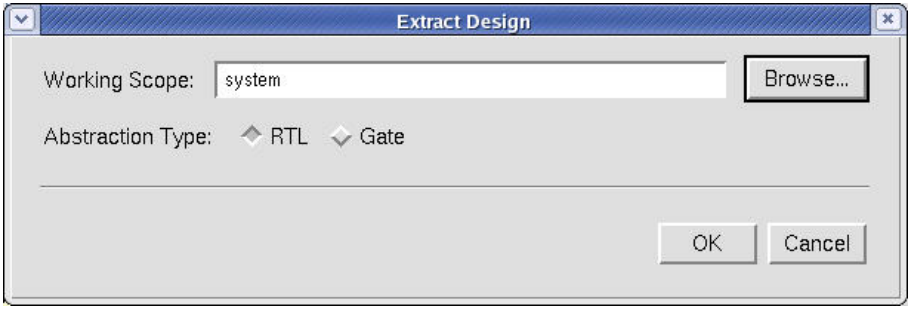


Figure: Extract Design Form

Working Scope: Specify the working scope. Design extraction analyzes the design starting with this scope and everything below this scope.

To change the working scope, click the **Change** button to open the *Behavior Analysis* form. After changing the scope, click the **OK** button to perform Behavior Analysis and update the working scope in the *Extract Design* form.

NOTE: The **Symbol Library Model First** option on the **Behavior Analysis -> Symbol Library** page of the *Preferences* form (invoked with the **Tools -> Preferences** command) must be turned *on* for correlation.

The working scope is saved to the *novas.rc* resource file, so, subsequent invocations of the Verdi and Siloti system with the same resource file set the default value of the working scope to the last value the field contained when the system is exited.

Abstraction Type: Specify the abstraction type as either **RTL** (for the RTL design) or **GATE** (for the gate level design) for the current correlation database extraction.

Perform CRDB Correlation

Menu Bar: **Tools -> Visibility -> Gate/RTL Correlation -> Advanced Commands -> Perform CRDB Correlation**

This command opens the *Perform CRDB Correlation* form where different naming rules for correlation can be specified. This form is equivalent to the *Prepare CRDB* form except the main and slave working scopes cannot be specified. Refer to the **Prepare CRDB** command for a detailed description of the tabs.

NOTE: This command is available in the primary process.

NOTE: A Correlation license is required to use this command.

NOTE: After clicking **OK**, any changes to the *Perform CRDB Correlation* form are automatically saved to the correlation database when the **Correlation -> Gate/RTL Correlation -> Save CRDB File** command is executed. These changes are loaded when the correlation database is restored with the **Correlation -> Gate/RTL Correlation -> Load CRDB File** command.

After modifying fields on any of the tabs, click the **OK** button to close the form and perform correlation. Click the **Cancel** button to close the form (keeping the entries intact until the session is exited) without performing correlation.

Save CRDB File

Menu Bar: **Tools -> Visibility -> Gate/RTL Correlation -> Advanced Commands -> Save CRDB File**

This command opens the *Save CRDB File* form where a correlation database (CRDB) to be saved can be specified. After specifying the file name, click the **OK** button to close the form and save the CRDB or click the **Cancel** button to close the form without saving any changes.

NOTE: A Correlation license is required to use this command.

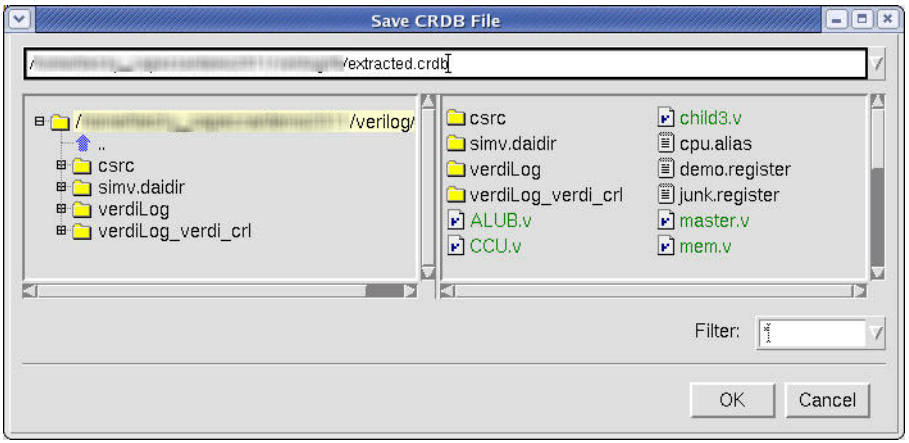


Figure: Save CRDB File Form

Enter the path and file name in the file name text field directly or view the directory structure (left pane) to specify the location to save the CRDB file. The default file name is *extracted.crdb* in the current directory. Click the **OK** button to enter the file name and close the form or click the **Cancel** button to exit the form without entering the file name.

Load CRDB File

Menu Bar: **Tools -> Visibility -> Gate/RTL Correlation -> Advanced Commands -> Load CRDB File**

This command opens the *Load CRDB File* form where a correlation database (CRDB) to be loaded can be specified. After specifying the file name, click the **OK** button to close the form and load the CRDB or click the **Cancel** button to close the form without loading the database.

NOTE: A Correlation license is required to use this command.

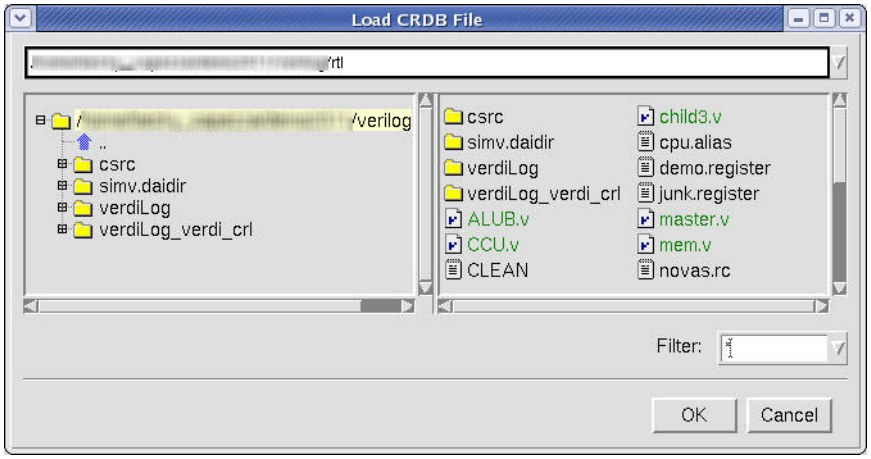


Figure: Load CRDB File Form

Enter the path and file name in the file name text field directly or view the directory structure (left pane) to locate the CRDB file (right pane) to be loaded. The default file name is *extracted.crdb* in the current directory. Click the **OK** button to enter the file name and close the form or click the **Cancel** button to exit the form without entering the file name.

Correlation Analyzer

Menu Bar:	**Tools -> Visibility -> Gate/RTL Correlation -> Advanced Commands -> Correlation Analyzer**

NOTE: Before invoking the **Correlation Analyzer** command, the correlation database (CRDB) must be extracted from gate-level and RTL designs.

This command opens the *Correlation Analyzer* window where the correlated and uncorrelated design objects for both the gate-level and the RTL design objects can be compared. In the window, browse the uncorrelated objects to decide how to add name-matching rules to get them correlated.

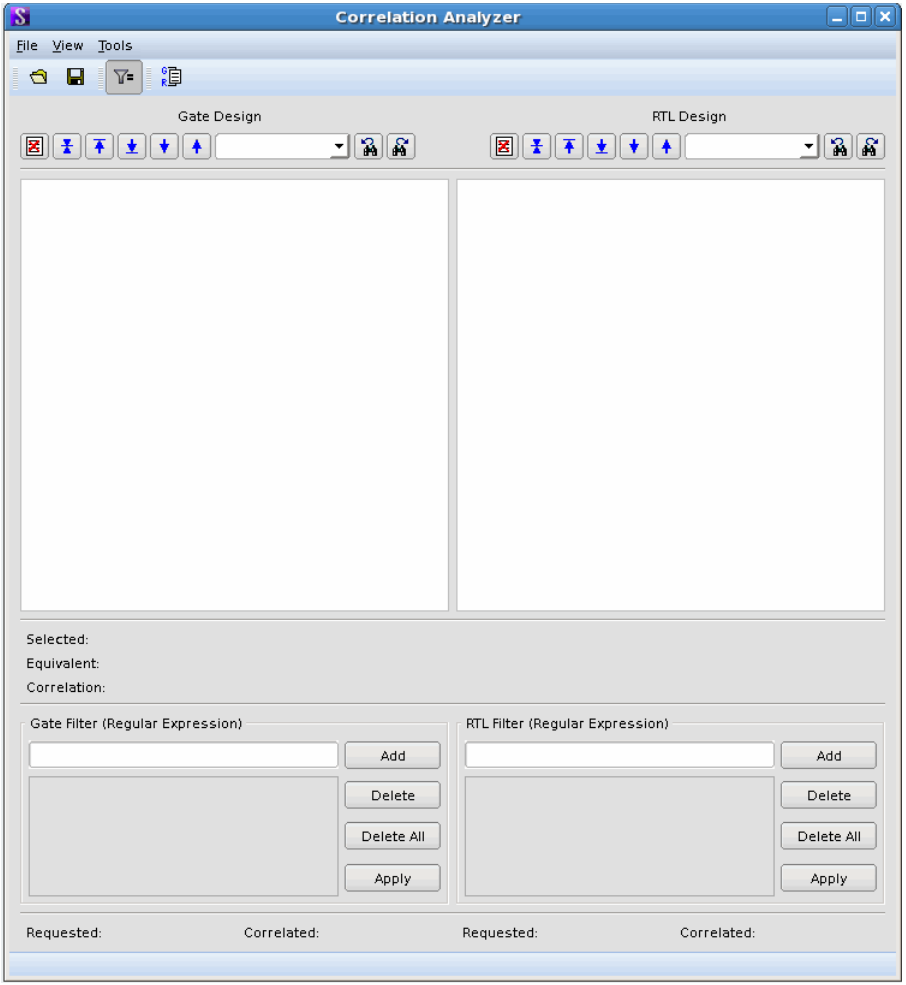


Figure: Correlation Analyzer Window

Editable Schematic

Overview

To cater to documentation purposes, the *Editable Schematic* frame is provided with useful editing and exporting commands to print the schematic.

The *Editable Schematic* frame is displayed when the **Tools -> New Schematic from Source -> Editable Window for Selected** command is invoked from *nTrace* or when the **Tools -> New Schematic -> Editable Window for Selected** command is invoked from *nSchema* or the **Tools -> New Schematic -> Editable Window for All** command is invoked from *nSchema* . The *Editable Schematic* frame is docked to the same frame location as the *nSchema* frame as a new tab. An *Editable Schematic* frame can become a standalone window by clicking the **Undock** icon on the toolbar.

The *Editable Schematic* frame is used to edit the default schematic view. The location of instances can be changed by selecting and dragging with the right mouse button. An instance can be resized by clicking on the selection and using the left mouse button to drag the two-sided arrow near the control points to the desired size. Rotating the selection and adding text comments for notes is also supported.

Instance(s) can be dragged from *nTrace* or *nSchema* and dropped to the current *Editable Schematic* window. If the dropped objects are in the current schematic, the objects are highlighted. If the dropped objects are not in the current schematic, the objects are added to the current schematic. If the dropped objects contain instance(s) which do not exist in the current schematic, a connection model must be specified between the **Only Connect to Current Net(s)** and **Fully Connect to Current Instance(s)** options.

The menu bar for the *Editable Schematic* window/frame is shown below.

Figure: Editable Schematic Menu Bar

The pull-down menus for the *Editable Schematic* window are summarized and explained on the following pages. Refer to the *Icons for Dockable Panes* section for details on the menu bar icons.

Menu Summary

The *Editable Schematic* menu commands are summarized below. All commands can be double-clicked to jump to the corresponding command description. For right-click commands, refer to the *Right-Click Commands* section for details.

File Commands

Print
Capture Window
Close Window

Edit Commands

Undo
Redo
Remove from Viewing Objects

View Commands

Instport Name
IO Port Name
Module Name
Entity Name
Arch Name
Instance Name
Net Name
Short Name
Constants
Preselect
Parameter List
High Contrast
Zoom->
  Zoom In
  Zoom Out
  Zoom All
  Fit Select Set
Pan ->
  Pan Left
  Pan Right
  Pan Up
  Pan Down

Schematic Commands

Find in Current Scope
Auto Fit Found Object(s)
Selection ->
  Select All
  Select Instances/Signals
  Select Signals
  Select Traced Results
  Deselect All
Focus Connection
Add Selected to Waveform
Change Color
Active Annotation
Annotate in Color
Leading Zeros

Trace Commands

Driver
Load
Connectivity

Object Commands

Add Comment
Edit Comment Content
Remove Comment Content
Rotate Left
Rotate Right
Flip Vertically
Flip Horizontally
Align ->
  Align Left
  Align Center
  Align Right
  Align Top
  Align Middle
  Align Bottom
Move ->
  Move Left
  Move Right
  Move Up
  Move Down

Tools Commands

Rearrange Schematic
Preferences
Options ->
  Hide Extraneous Bus
Customize Menu/Toolbar

File Commands

Print

Menu Bar: **File -> Print**

This command opens an *nSchema Print* form where the print settings of the current view in the *Editable Schematic* window can be specified. After completing the print setup, click the **Preview** button to open the *Print Preview* form. For details, refer to the *Preview: Click* **Preview** *to display the Print Preview form, as illustrated in the following figure:* section in the *nSchema* chapter.

The *nSchema Print* form includes five tabs as described below.

Basic Tab

This tab is for general settings, including the **Paper** section and the **Destination** section.

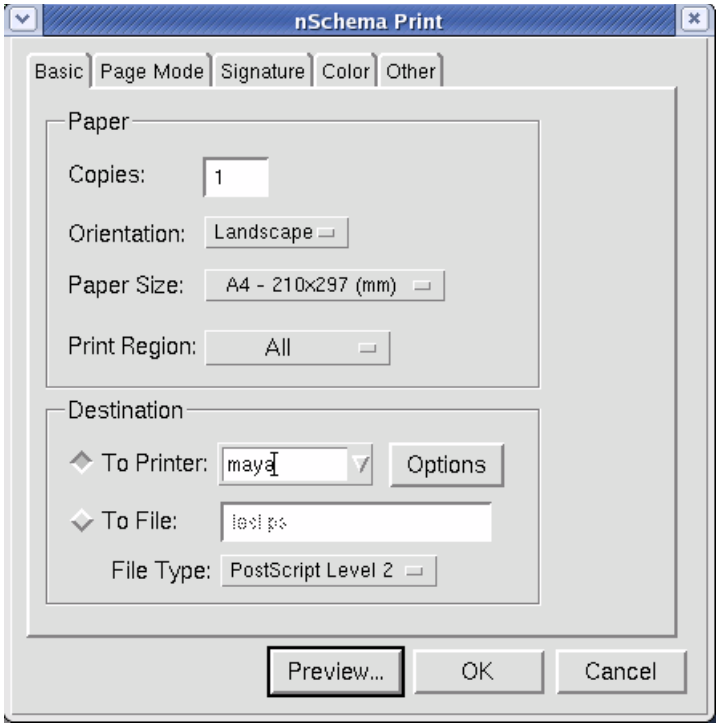


Figure: nSchema Print Form - Basic Tab

The **Paper** section includes the following fields:

- **Copies:** Specify the number of copies to be printed.
- **Orientation:** Specify the orientation for the printed page by selecting either **Landscape** or **Portrait**. The default value of this field is **Landscape**.
- **Paper Size:** Specify the paper size to be printed. The default value of this field is **A4**.
- **Print Region:** Specify a printing region by selecting from **All**, **Current View**, **Fit To Page**, and **Actual Size** options. The default value of this field is **Fit To Page**.

The **Destination** section includes the following options, field, and button:

- **To Printer:** This option sends the file specified in the text field to the printer.
- **Options:** This button opens the *Configuration* form where advanced printing options can be selected. For details, refer to the *Configuration Form* section.

- **To File:** This option saves the print file to the current working directory using the specified postscript file type. When this option is selected, the file type can be selected from **PostScript Level 2**, **PostScript Level 1**, **MIF**, and **Enhanced MetaFile** sub-options.

Page Mode Tab

Refer to the *Page Mode Tab* section in the *nSchema* chapter for details.

Signature Tab

Refer to the *Signature Tab* section in the *nSchema* chapter for details.

Color Tab

This tab specifies the color for the selected item.

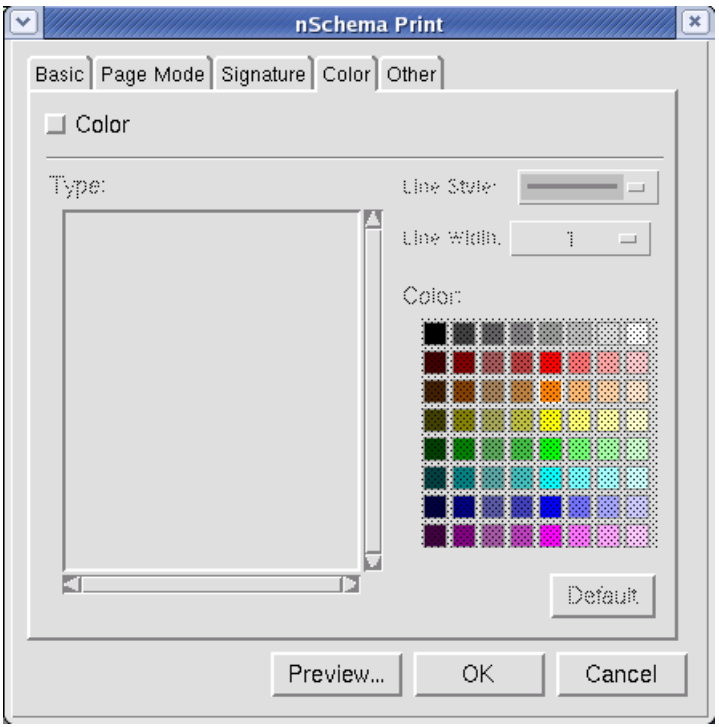


Figure: nSchema Print Form - Color Tab

- **Color:** Turn this option *on* to print multiple colors on a color printer; disable this option to get a black-and-white printout on a color printer. When the **Color** option is turned *on*, specify the color preferences in the area below the **Color** option. The default value of this option is *off*.

- **Type:** When the **Color** option is turned *on*, specify the **Line Style**, **Line Width**, and **Color** for the selected item in **Type** section. The new color preferences are saved and applied after the **OK** button is clicked.
- **Default:** Click this button to restore to the default values.

NOTE: The color preferences on the *nSchema Print* form are for printouts only. They do not affect the colors being displayed in the *Editable Schematic* window.

Other Tab

This tab provides miscellaneous print settings.

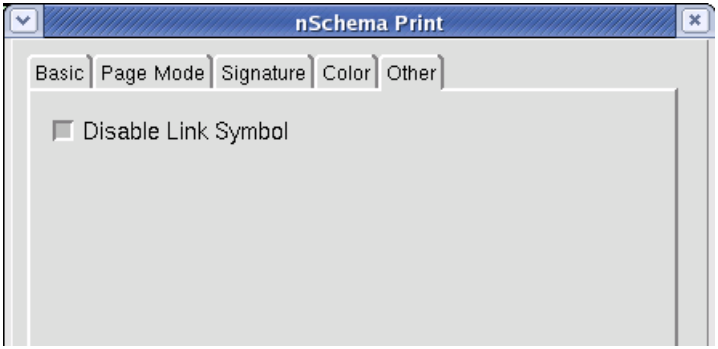


Figure: nSchema Print Form - Other Tab

Disable Link Symbol: When this option is turned *on*, the display of design sheet or external file link icons is disabled. The default value is *off*.

To link to other documents, invoke the **Object -> Edit Comment Content** command in the *Editable Schematic* window to open the *Comment Box Content* form where the **Insert File** button can be clicked to add the desired documents.

Configuration Form

The **Printer** section includes the following fields:

- **Printer Name:** Enter the printer name in the text field.
- **Language:** Select an output format from one of the following: **PostScript Level 2**, **PostScript Level 1**, and **HP GL/2**.

In the **Paper Size** section, select the paper size from the list. If **User-defined** is selected, the **User-defined** section is enabled automatically to specify the width and the height of the paper.

The **Print Command** section contains the following options for configuring the print statement:

- **Command:** Specify the print command. The default is *lpr*.
- **Destination:** Specify the print destination option. The default is *-P*.
- **Copies:** Specify the copy number option. The default is *-#*.

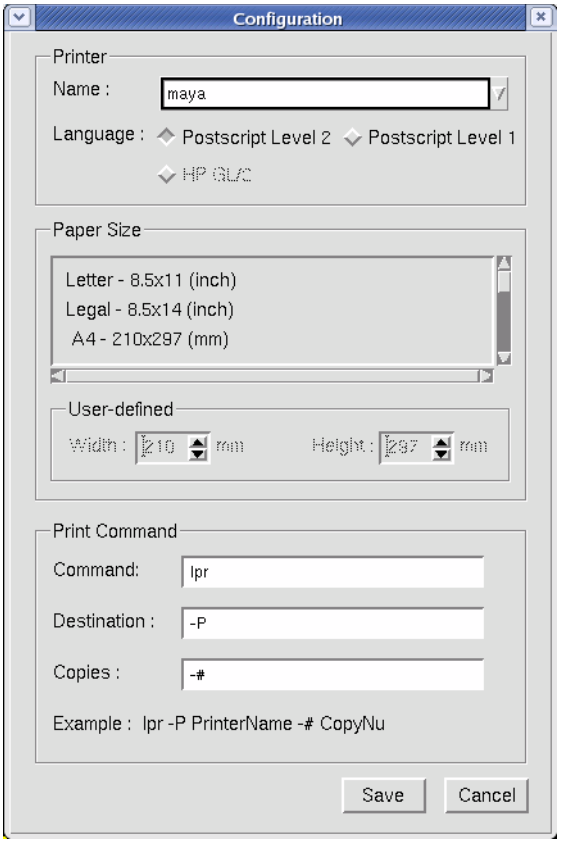


Figure: Configuration Form

Capture Window

Menu Bar: **File -> Capture Window**

Refer to the **File -> Capture Window** command description in the *nSchema* chapter for details.

Close Window

Menu Bar: **File -> Close Window**

This command closes the current *Editable Schematic* window.

Edit Commands

Undo

Menu Bar: **Edit -> Undo**

Toolbar Icon: 

This command reverses the previously performed action.

Redo

Menu Bar: **Edit -> Redo**

Toolbar Icon: 

This command repeats the previously performed action.

Remove from Viewing Objects

Menu Bar: **Edit -> Remove from Viewing Objects**

Bind Key: <Key> Delete

Toolbar Icon: 

This command removes the selected objects from the current *Editable Schematic* window.

View Commands

Instport Name

Menu Bar: **View -> Instport Name**

This toggle command turns the display of instance port names in the current *Editable Schematic* window *on* or *off*. The default value is *off*.

IO Port Name

Menu Bar: **View -> IO Port Name**

This toggle command turns the display of input/output (IO) port names in the current *Editable Schematic* window *on* or *off*. The default value is *off*.

Module Name

Menu Bar: **View -> Module Name**

This toggle command turns the display of module names in the current *Editable Schematic* window *on* or *off*. The default value is *on*.

Entity Name

Menu Bar: **View -> Entity Name**

This toggle command turns the display of entity names inside the modules in the current *Editable Schematic* window *on* or *off*. The default value is *on*.

Arch Name

Menu Bar: **View -> Arch Name**

This toggle command turns the display of architecture names in the current *Editable Schematic* window *on* or *off*. The default value is *on*.

Instance Name

Menu Bar: **View -> Instance Name**

This toggle command turns the display of instance names in the current *Editable Schematic* window *on* or *off*. The default value is *off*.

Net Name

Menu Bar: **View -> Net Name**

This toggle command turns the display of local net names in the current *Editable Schematic* window *on* or *off*. The default value is *off*.

Short Name

Menu Bar: **View -> Short Name**

This toggle command turns the display of short names in the current *Editable Schematic* window *on* or *off*. The default value is *off*.

Constants

Menu Bar: **View -> Constants**

This toggle command turns the display of constants connected to gates or block symbols in the current *Editable Schematic* window *on* or *off*. The default value is *off*.

Preselect

Menu Bar: **View -> Preselect**

When this command is *on*, signals or blocks are highlighted and a tip is displayed when the cursor moves over nets/instances in the current *Editable Schematic* window. The default value is *off*. The message bar at the bottom of the main framework also shows the name of the selected signal or block.

Parameter List

Menu Bar: **View -> Parameter List**

This toggle command turns the display of the parameter list on the module block in the current *Editable Schematic* window *on* or *off*. The parameter list must be defined in the design file. The default value is *off*.

High Contrast

Menu Bar: **View -> High Contrast**

This toggle command turns the high contrast display of selected and traced items in the current *Editable Schematic* window *on* or *off*. The default value is *off*.

Zoom

This command includes four subcommands for viewing the schematic: **Zoom In**, **Zoom Out**, **Zoom All**, and **Fit Select Set**.

Zoom In

Menu Bar: **View -> Zoom -> Zoom In**

Toolbar Icon: 

Bind Key: Shift+Z

This command provides a close-up view of the contents in the *Editable Schematic* window. The magnification of the viewing area is changed to half the magnification of the previous view.

NOTE: A specific area can be zoomed by dragging-left to form a rectangle around the area to be zoomed.

Zoom Out

Menu Bar: **View -> Zoom -> Zoom Out**

Toolbar Icon: 

Bind Key: Z

This command enables more of the contents to be seen in the *nSchema* window at a reduced size. The magnification of the viewing area is changed to two times the magnification of the previous view from the center point in both horizontal and vertical directions.

Zoom All

Menu Bar:	**View -> Zoom -> Zoom All**
Toolbar Icon:	
Bind Key:	F

This command shows all of the contents of the schematic.

Fit Select Set

Menu Bar:	**View -> Zoom -> Fit Select Set**
Bind Key:	Ctrl+F

This command fits the selected objects in the schematic window.

Pan

This command includes four subcommands for moving the schematic: **Pan Left**, **Pan Right**, **Pan Up**, and **Pan Down**.

Pan Left

Menu Bar:	**View -> Pan -> Pan Left**
Bind Key:	Left Arrow

This command pans the schematic window to the left.

Pan Right

Menu Bar:	**View -> Pan -> Pan Right**
Bind Key:	Right Arrow

This command pans the schematic window to the right.

NOTE: Panning left and right can also be achieved by moving the horizontal scroll bar in the *nSchema* window.

Pan Up

Menu Bar: **View -> Pan -> Pan Up**

Bind Key: Up Arrow

This command pans the schematic window up.

Pan Down

Menu Bar: **View -> Pan -> Pan Down**

Bind Key: Down Arrow

This command pans the schematic window down.

NOTE: Panning up and down can also be achieved by moving the vertical scroll bar in the *nSchema* window.

Schematic Commands

Find in Current Scope

Menu Bar: **Schematic -> Find in Current Scope**

Bind Key: A

This command opens the *Find* form where all signals or instances in the *Editable Schematic* window are listed. Enter a string in the **Find** text field to search for the desired signal(s) or instance(s) and the results are highlighted in the **Find** list. In the **Type** section, enable the **Signal** option to only find signals or enable the **Instance** option to only find instances matching the specified string.

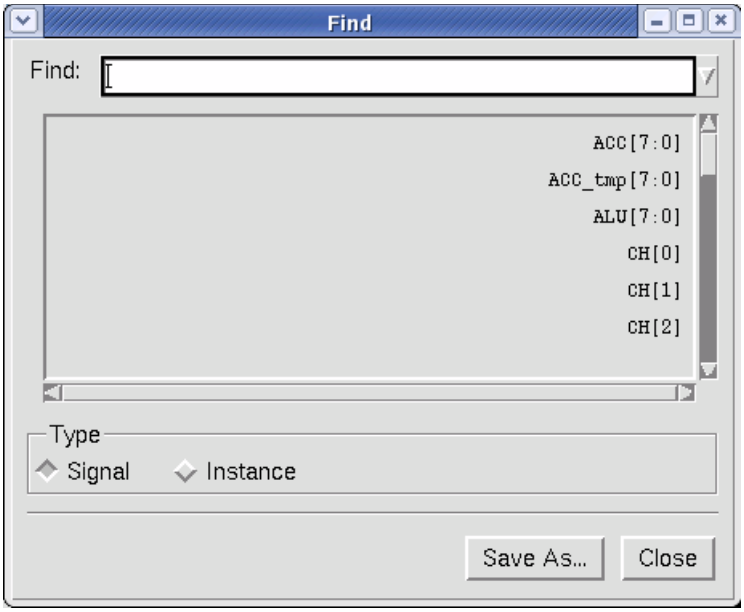


Figure: Find Form

Auto Fit Found Object(s)

Menu Bar: **Schematic -> Auto Fit Found Object(s)**

When this command is turned *on*, the current *Editable Schematic* window automatically zooms to fit the display for the selected item. The default vale is *off*.

Selection

Select All

Menu Bar: **Schematic -> Selection -> Select All**

Bind Key: Ctrl+A

This command selects all the objects in the *Editable Schematic* window.

Select Instances/Signals

Menu Bar: **Schematic -> Selection -> Select Instances/Signals**

This command opens the *Select Instances/Signals* form; the instance(s) and/or the signal(s) to select in the *Editable Schematic* window can be chosen.

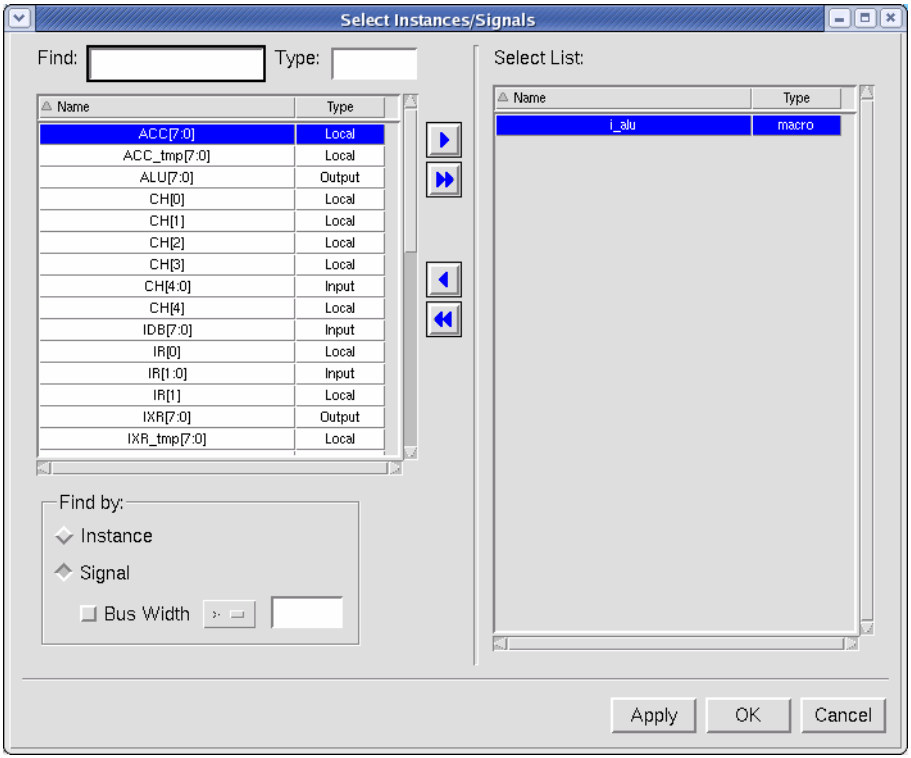


Figure: Select Instances/Signals Form

Several methods are available for selecting signals/instances. The left pane lists all instances or signals available in the current *Editable Schematic* window in alphabetical order. Instances/signals of interest can be selected in the left pane (drag the mouse or hold the Shift key for multiple object selection), and then the selected instances/signals can be added to the **Select List** with **Add Object** or **Add All** icons or unwanted instances/signals can be removed from the Select List with the **Remove Object** or **Remove All** icons. Double-click item(s) in the *Instance*/*Signal* pane to move the item(s) to the **Select List** automatically, or double-click item(s) in the **Select List** to move the item(s) back to the *Instance*/*Signal* pane.

When instance(s)/signal(s) are added to the **Select List**, clicking the **Apply** or **OK** buttons highlight the selected instance/signal in the current *Editable Schematic* window simultaneously.

The *Select Instances/Signals* form includes the following fields:

- **Find:** Enter a search string in the text field; signals/instances matching the string in the **Name** column are displayed in the list below. Wildcard characters are supported.
- **Type:** Enter a signal/instance type in the text field; signals/instances matching the string in the **Type** column are displayed in the list below. Wildcard characters are supported.
- **Find By:** Select the **Instance** option to only find instances matching the specified string or enable the **Signal** option to only find signals. When the **Signal** option is selected, set the criteria associated with the bus width. A conditional operator of greater than (>), greater than or equal to (>=), equal to (=), less than or equal to (<=), or less than (<) can be selected. This conditional operator is combined with the number of bits specified. When the Enter key is pressed, the buses meeting the criteria are added into the candidate list.

Select Signals

Menu Bar: **Schematic -> Selection -> Select Signals**

This command shows a  cursor when the command is invoked. Move the cursor to cross the desired signal(s); the selected signal(s) are highlighted in the *Editable Schematic* window.

Select Traced Results

Menu Bar: **Schematic -> Selection -> Select Traced Results**

If driver/load/connectivity has been traced in a partial hierarchy window, this command automatically selects the trace results.

Deselect All

Menu Bar: **Schematic -> Selection -> Deselect All**

This command de-selects all objects in the current *Editable Schematic* window.

Focus Connection

Menu Bar: **Schematic -> Focus Connection**

Mouse Action: Double-click the signal in the schematic window

This command highlights the connection (driving instance to loading instance(s) with the connecting net) for the selected signal in the current *Editable Schematic* window. Use *n* (previous) or *N* (next) to browse the connecting instances and net.

Add Selected to Waveform

Menu Bar: **Schematic -> Add Selected to Waveform**

Bind Key: Ctrl+W

This command adds the selected signal(s) in the *Editable Schematic* window to the signal pane in *nWave*. The added signals are at the current cursor position. Alternatively, the selected signals can be dragged from the *Editable Schematic* window to the *nWave* frame.

Change Color

Menu Bar: **Schematic -> Change Color**

Bind Key: C

This command opens the *Change Selection Color* form where the color of the selected signal can be changed.

Figure: Change Selection Color Form

This form shows the selected signal/instance with its current color in the color palette. The color can be changed instantly by clicking another color on the color palette. Set the selected signal color as the default by clicking the **Default** button. The new color is applied when these signals are dropped to the *nWave* frame.

NOTE: A quick way to change a signal's color within a predefined color set is to select the signal, then continue pressing the (t) key to sequence through the color set until the preferred color is reached.

Active Annotation

Menu Bar: **Schematic -> Active Annotation**

Bind Key: X

After an FSDB file with simulation results is loaded from *nTrace* or *nWave*, this command is selectable. When this toggle command is turned *on*, the values associated with signals are displayed in the *Editable Schematic* window. The value is updated whenever the current simulation time changes.

Annotate in Color

Menu Bar: **Schematic -> Annotate in Color**

This toggle command turns the annotation style and line coloring *on/off* in the current *Editable Schematic* window. The default is *off*.

The definitions of the value types are as follows:

1: logic high
0: logic low
z: high impedance
x: unknown

The annotation color and line style can be changed by selecting the **Value 1**, **Value 0**, **Value x**, or **Value z** items in the **Type** list of the **Schematics** -> **Color/Font** page of the *Preferences* form (invoked with the **Tools -> Preferences** command) and change their annotation line style and color.

Leading Zeros

Menu Bar: **Schematic** -> **Leading Zeros**

Refer to the **Schematic** -> **Leading Zeros** command description in the *nSchema* chapter for details.

Trace Commands

Driver

Menu Bar: Trace -> Driver

Toolbar Icon: 

This command traces all possible drivers of a selected signal and shows the results in the current *Editable Schematic* window.

Load

Menu Bar: Trace -> Load

Toolbar Icon: 

This command traces all possible loads of a selected signal and shows the results in the current `Editable Schematic` window.

Connectivity

Menu Bar: Trace -> Connectivity

This command traces all possible loads and drivers of a selected signal and shows the results in the current *Editable Schematic* window. This command combines the functions of the **Trace Driver** and **Trace Load** commands.

Object Commands

Add Comment

Menu Bar: **Object -> Add Comment**

Toolbar Icon: 

This command adds a text comment where notes can be added.

The comment can be moved by holding the right-mouse button on the comment and then releasing it when the comment is in the desired position. The comment can be resized by dragging the two-sided arrow near the control points to the desired size.

The **Viewing Properties** command in the right-click command menu can be invoked for the preferred color, font, size, style, and alignment of the text in the comment.

Edit Comment Content

Menu Bar: **Object -> Edit Comment Content**

After a comment is selected, this command is enabled to open the *Comment Box Content* form where the comment description can be edited.

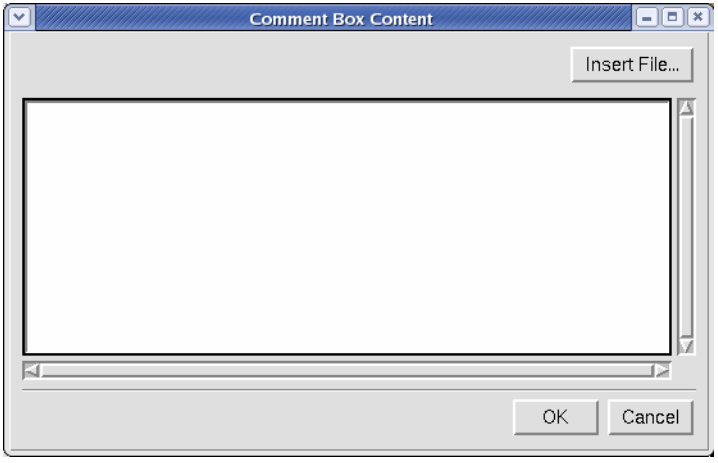


Figure: Comment Box Content Form

After clicking the **Insert File** button to add an external file, the  icon appears on the upper left corner of the comment. Double-click the icon to view the comment in a source code window.

Remove Comment Content

Menu Bar: **Object -> Remove Comment Content**

After a comment is selected with a link to an external file, this command is enabled to delete the content in the comment.

Rotate Left

Menu Bar: **Object -> Rotate Left**

Toolbar Icon: 

This command rotates the selected object or instance(s) to the left by 90 degrees. Multiple instances can be selected for rotation.

Rotate Right

Menu Bar: **Object -> Rotate Right**

Toolbar Icon: 

This command rotates the selected object or instance(s) to the right by 90 degrees. Multiple instances can be selected for rotation.

Flip Vertically

Menu Bar: **Object -> Flip Vertically**

Toolbar Icon: 

This command flips the selected object or instance(s) vertically. Multiple instances can be selected to flip.

Flip Horizontally

Menu Bar: Object -> Flip Horizontally

Toolbar Icon: 

This command flips the selected object or instance(s) horizontally. Multiple instances can be selected to flip.

Align

The following alignment commands are available.

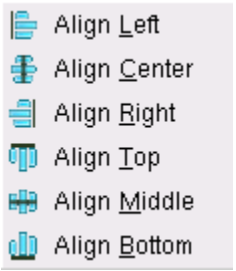


Figure: Align Menu and Icons

Align Left

Menu Bar: **Object -> Align -> Align Left**

After multiple objects are selected, this command is enabled to align the selection to the left-hand side.

Align Center

Menu Bar: **Object -> Align -> Align Center**

After multiple objects are selected, this command is enabled to align the selection to the center.

Align Right

Menu Bar: **Object -> Align -> Align Right**

After multiple objects are selected, this command is enabled to align the selection to the right-hand side.

Align Top

Menu Bar: **Object -> Align -> Align Top**

After multiple objects are selected, this command is enabled to align the selection to the top.

Align Middle

Menu Bar: **Object -> Align -> Align Middle**

After multiple objects are selected, this command is enabled to align the selection to the middle.

Align Bottom

Menu Bar: **Object -> Align -> Align Bottom**

After multiple objects are selected, this command is enabled to align the selection to the bottom.

Move

Move Left

Menu Bar: **Object -> Move -> Move Left**

Bind Key: Ctrl+Left Arrow

After an instance, an instance port, a port, a net segment, or an object is selected, this command is enabled to move the selection to the left. Multiple objects of the same type can be selected.

Move Right

Menu Bar: **Object -> Move -> Move Right**

Bind Key: Ctrl+Right Arrow

After an instance, an instance port, a port, a net segment, or an object is selected, this command is enabled to move the selection to the right. Multiple objects of the same type can be selected.

Move Up

Menu Bar:	**Object** -> **Move** -> **Move Up**
Bind Key:	Ctrl+Up Arrow

After an instance, an instance port, a port, a net segment, or an object is selected, this command is enabled to move the selection up. Multiple objects of the same type can be selected.

Move Down

Menu Bar:	**Object** -> **Move** -> **Move Down**
Bind Key:	Ctrl+Down Arrow

After an instance, an instance port, a port, a net segment, or an object is selected, this command is enabled to move the selection down. Multiple objects of the same type can be selected.

Tools Commands

Rearrange Schematic

Menu Bar: **Tools -> Rearrange Schematic**

This command automatically optimizes the size and the placement of all instances in the current view and adjusts the placement of pins, ports, and net routing.

This command is used when the schematic looks messy after a module is moved or reshaped and all connections are rearranged.

Preferences

Menu Bar: **Tools -> Preferences**

This command specifies the settings for all schematic windows. Refer to the *Schematics Folder* section in the *Preferences* chapter for details.

Options

Hide Extraneous Bus

Menu Bar: **Tools -> Options**

This command controls the bus place and route style.

Refer to the **Trace -> Options -> Hide Extraneous Bus (Flattened View Only)** command description in the *nSchema* chapter for details.

Customize Menu/Toolbar

Refer to the **Tools -> Customize Menu/Toolbar** command in the *nTrace* chapter for details.

Right-Click Commands

Many of the previously described commands can also be selected from the right-click command menu in the *Editable Schematic* frame.

Editable Schematic Frame Right-Click Options

After the right mouse button is clicked anywhere in the menu, toolbar icon, or frame banner area of the *Editable Schematic* frame or standalone window, a configuration option menu is displayed.This menu can be used to configure the available icons.

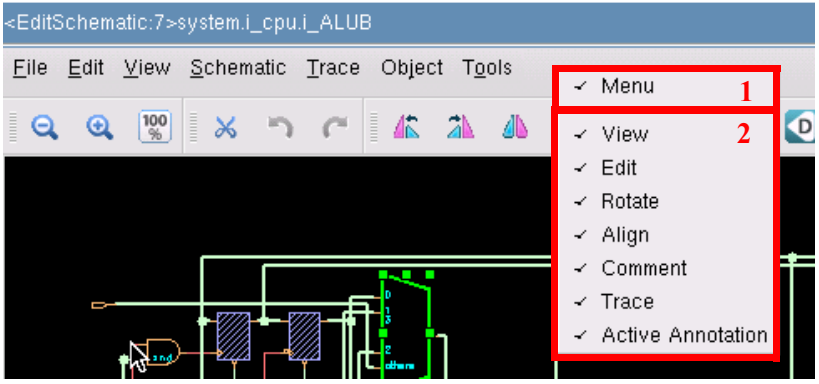


Figure: Configuration Option Menu

The **Menu** option (labeled 1 in the figure above) enables/disables the command menu bar. The options in the bottom section (labeled 2 in the figure above) enable/disable the toolbar icons for different functions.

Main Window Right-Click Commands

The following is the list of right-click menu options available for the full hierarchical view of *Editable Schematic*. Some selections are available from the right mouse button menu. Other selections have equivalent selections available from the toolbar menus.

Find in Current Scope

Refer to the **Schematic -> Find in Current Scope** command description for details.

Zoom All

Refer to the **View -> Zoom ->** **Zoom All** command description for details.

Selection

This menu includes the following subcommands:

Select All

Refer to the **Schematic -> Selection ->** **Select All** command description for details.

Select Instances/Signals

Refer to the **Schematic -> Selection ->** **Select Instances/Signals** command description for details.

Select Signals

Refer to the **Schematic -> Selection ->** **Select Signals** command description for details.

Add Comment

Refer to the **Object ->** **Add Comment** command description for details.

Drop

Bind Key: Ctrl+V

This command drops a signal that has been dragged from another location. This is similar to releasing the middle mouse button for a drop.

Instance Object Type Right-Click Commands

The following is the list of right-click menu options available when instance object types (primitive or module) are selected:

Remove from Viewing Objects

NOTE: This command is available for primitive instances.

Refer to the **Edit ->** **Remove from Viewing Objects** command description for details.

Move

There are four subcommands available. Refer to the **Object -> Move** command description for details.

Rotate

Refer to **Object** -> **Rotate Left** and **Object** -> **Rotate Right** command descriptions for details.

Flip

Refer to **Object** -> **Flip Vertically** and **Object** -> **Flip Horizontally** command descriptions for details.

Trace

This menu includes the following subcommands; these commands are available for primitive instances:

Driver

Refer to the **Trace -> Driver** command description for details.

Load

Refer to the **Trace -> Load** command description for details.

Connectivity

Refer to the **Trace -> Connectivity** command description for details.

Focus Connection

Refer to the **Schematic -> Focus Connection** command description for details.

Show No. of Driver/Show No. of Load

These commands display the number of drivers/loads on a node. They are only valid only if a net, an instance port, or an I/O port are right-clicked.

Drag

Bind Key: Ctrl+C

This command drags the selected signal to another location where it can be dropped. This is similar to using the middle mouse button to select and drag.

Drop

Bind Key: Ctrl+V

This command drops a signal that has been dragged from another location. This is similar to releasing the middle mouse button for a drop.

Signal/Net Object Type Right-Click Commands

Following is the list of right-click menu options available when signal/net object types are selected:

Remove from Viewing Objects

Refer to the **Edit -> Remove from Viewing Objects** command description for details.

Trace

This menu includes the following subcommands:

Driver

Refer to the **Trace -> Driver** command description for details.

Load

Refer to the **Trace -> Load** command description for details.

Connectivity

Refer to the **Trace -> Connectivity** command description for details.

Focus Connection

Refer to the **Schematic -> Focus Connection** command description for details.

Show No. of Driver/Show No. of Load

These commands display the number of drivers/loads on a node. They are only valid only if a net, an instance port, or an I/O port are right-clicked.

Drag

Refer to the **Drag** command description.

Drop

Refer to the **Drop** command description.

Note Object Type Right-Click Commands

The following is the list of right-click menu options available when signal/net object types are selected:

Viewing Properties

This command opens the *Object Properties* form where the default font, size, and color for the selected note can be changed on the **Font** tab. The text alignment can be changed on the **Comment Box** tab.

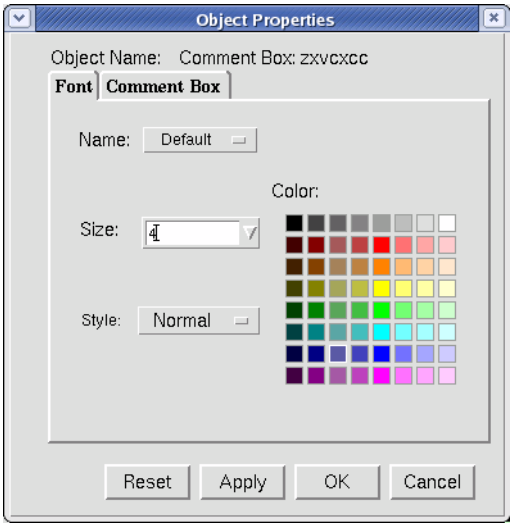


Figure: Object Properties - Font Tab

Move

There are four subcommands available. Refer to the **Object -> Move** command description for details.

Remove from Viewing Objects

Refer to the **Edit -> Remove from Viewing Objects** command description for details.

Toolbar Icons and Fields

Figure: Toolbar Used in the Editable Schematic Window

The available toolbar icons may be modified. Refer to the *Toolbars* section of the *User Interface* chapter in the *Verdi User Guide and Tutorial* manual for details.

The different toolbar categories and available icons are described below.

View Category

Zoom Out

Refer to the **View -> Zoom -> Zoom Out** command description for details.

Zoom In

Refer to the **View -> Zoom -> Zoom In** command description for details.

Zoom All

Refer to the **View -> Zoom -> Zoom All** command description for details.

Edit Category

Remove from Viewing Objects

Refer to the **Edit -> Remove from Viewing Objects** command description for details.

Undo

Refer to the **Edit -> Undo** command description for details.

Redo

Refer to the **Edit -> Redo** command description for details.

Rotate Category

Rotate Left 

Refer to the **Object -> Rotate Left** command description for details.

Rotate Right 

Refer to the **Object -> Rotate Right** command description for details.

Flip Horizontally 

Refer to the **Object -> Flip Horizontally** command description for details.

Flip Vertically 

Refer to the **Object -> Flip Vertically** command description for details.

Align Category

Align Left 

Refer to the **Object -> Align -> Align Left** command description for details.

Align Center 

Refer to the **Object -> Align -> Align Center** command description for details.

Align Right 

Refer to the **Object -> Align -> Align Right** command description for details.

Align Top 

Refer to the **Object -> Align -> Align Top** command description for details.

Align Middle 

Refer to the **Object -> Align -> Align Middle** command description for details.

Align Bottom 

Refer to the **Object -> Align -> Align Bottom** command description for details.

Comment Category

Add Comment 

Refer to the **Object -> Add Comment** command description for details.

Trace Category

Trace Driver 

Refer to the **Trace -> Driver** command description for details.

Trace Load 

Refer to the **Trace -> Load** command description for details.

Active Annotation Category

Search By 

Click this toolbar icon to specify the search criteria for the value change of a signal.This icon is 'active' only when an FSDB file is loaded and **Active Annotation** is enabled.

Search Backward 

Click this toolbar icon to search backward in time and locate the value in the waveform window. This icon is active only when an FSDB file is loaded and the **Schematic -> Active Annotation** command is enabled.

Search Forward 

Click this toolbar icon to search forward in time and locate the value in the waveform window. This icon is 'active' only when an FSDB file is loaded and the **Schematic -> Active Annotation** command is enabled.

Cursor Time Entry/Display 

This text field displays the cursor time and also sets the cursor time. The field updates all back-annotated values at a specified cursor time. This field is 'active' only when an FSDB file is loaded and the **Schematic -> Active Annotation** command is enabled.

nECO

Overview

As indicated in the *Introduction* chapter, the *nECO* module is an additional Synopsys product that can be used with the Verdi platform.

The *nECO* pane is displayed when the **Tools -> New Schematic from Source -> ECO Window for Selected** option is invoked from the *nTrace* menu bar or the **Tools -> New Schematic -> ECO Window for Selected Instance(s)** or **Tools -> New Schematic -> ECO Window for All Instances** option is invoked from the *nSchema* menu bar. The *nECO* pane is docked to the same pane location as the *nSchema* window as a new tab. An *nECO* pane can become a standalone window by clicking the **Undock** icon on the toolbar.

The menu bar for the *nECO* window is shown below.

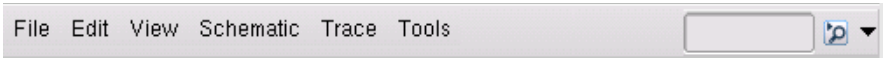


Figure: nECO Menu Bar

The pull-down menus for *nECO* are explained on the following pages. Refer to the *Icons for Dockable Panes* section for details on the menu bar icons.

Menu Summary

The *nECO* menu options are summarized below. All options can be double-clicked to jump to the corresponding option description. For right-click options, refer to the *Right-click Options* section for details.

File Options

Save ECO Netlist - nECO
ECO Report - nECO
Save ECO Script - nECO
Save ECO Netlist - nECO
Save in HDL
Commit Change
ECO Change Status
Print
Capture Window
Close Window

Edit Options

Undo
Redo
Enable Hierarchy Connector Editing
Rename Net
Rename Instance
Connect to Net
Edit Concate Net
Copy Instance
Paste Instance
Remove from Viewing Objects
Delete Connection
Make Connection
Tie Up
Tie Down
Create Port
Add Buffers (Non-Freeze Silicon Mode)
Add Buffers (Freeze Silicon Mode)
Delete Buffer
Add Instance (Non-Freeze Silicon Mode)
Add Instance (Freeze Silicon Mode)
Delete Instance
Replace Instance (Non-Freeze Silicon Mode)
Replace Instance (Freeze Silicon Mode)
Change Instance Scope
Clone Module
Spare Cell
Count Spare Cell

View Options

Preselect
Port Name
Instance Name
Local Net Name
Parameter List
High Contrast
Short Name
Fan In/Out Number

Zoom->
  Zoom In
  Zoom Out
  Zoom All
  Fit Select Set
Pan ->
  Pan Left
  Pan Right
  Pan Up
  Pan Down
Last View

Schematic Options

Find in Current Scope
Auto Fit Found Object(s)
Selection ->
  Select
  Select Common Pin
  Select Pin By Net
  Select All Objects
  Select All Instances
  Select All Signals
  Deselect All
Focus Connection
Add Selected to Waveform

Change Color
All Objects to Default Color
Active Annotation
Annotate in Color
Leading Zeros
SDF Annotation
Delay Type
Delay Precision
Options ->
  Hide Extraneous Bus
  Keep Placement

Trace Options

Driver
Load

Connectivity

Tools Options

New Schematic ->
  Browser Window
  Flattened Window
  Fan-in Cone
  Fan-out Cone
  Driver
  Load
  Connectivity

Preferences
Customize Menu/Toolbar

File Options

Save ECO Netlist - nECO

Menu Bar: File -> Save ECO Netlist

This option opens the *Save ECO Netlist* form.

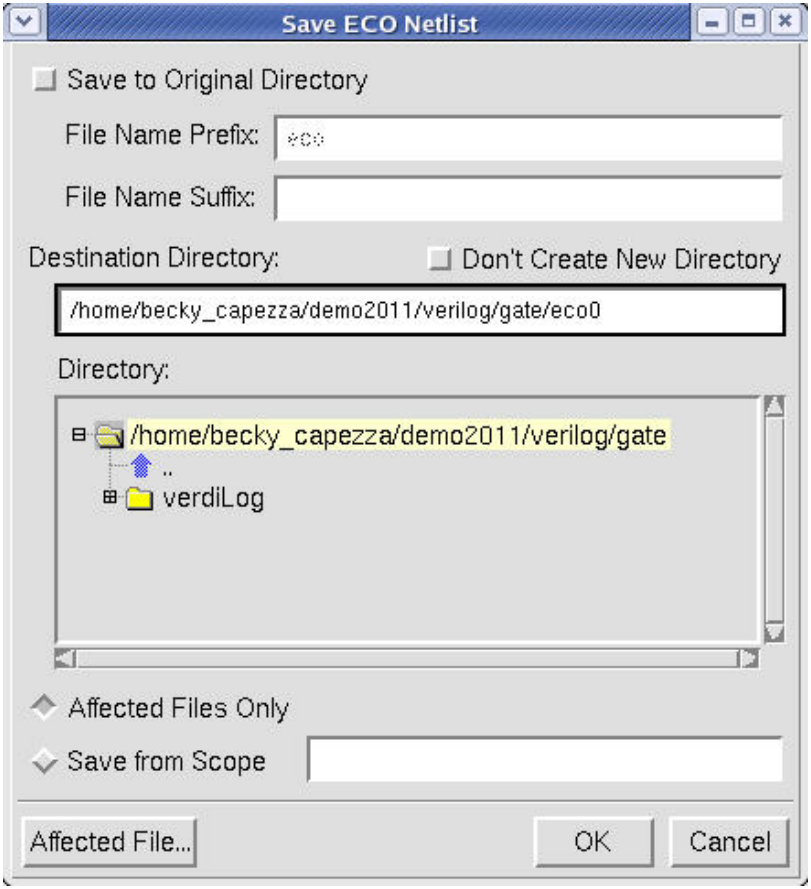


Figure: Save ECO Netlist

This form includes the following options, fields, and button:

- **Save to Original Directory:** Turn this option *on* to save the netlist to the same directory where the design was imported from.

- **File Name Prefix:** Enter the prefix for the files to be managed in this text field.
- **File Name Suffix:** Enter the prefix for the files to be managed in this text field.
- **Destination Directory:** Type in the desired directory to save the netlist to a directory different than the original one.
- **Affected Files Only:** When this option is turned *on*, only the affected files are seen; the original floating files are not included.
- **Save from Scope:** When this option is turned *on*, the design files that are related to the scope specified in text field are saved.
- **Affected File:** While clicking this button, the **Affected File List** opens to indicate the original and the affected file.

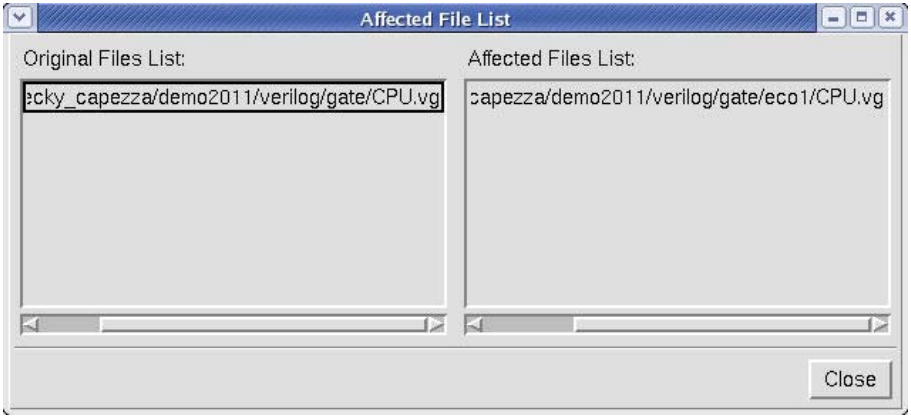


Figure: Affected File List

ECO Report - nECO

Menu Bar: File -> ECO Report

This option generates a report for all the changes made during the ECO session.

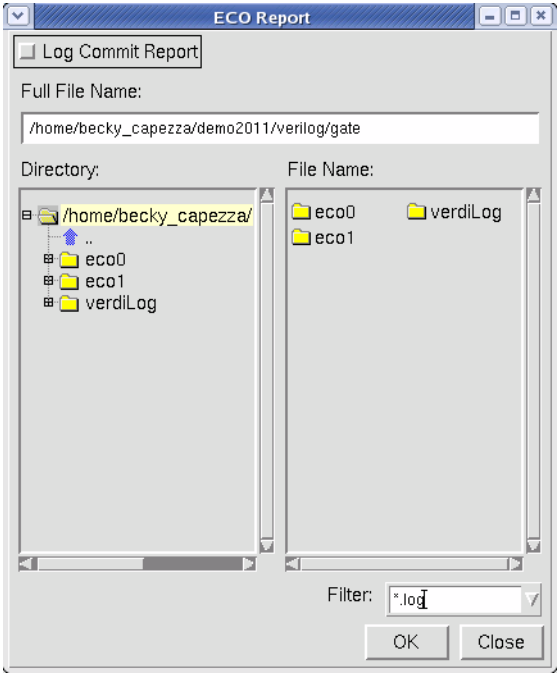


Figure: ECO Report Form

This form includes the following option and field:

- **Log Commit Report:** When this option is turned *on*, the commit report is appended to the original ECO report and the commit report is logged after the ECO report.
- **Full File Name:** Specify the full file name of the ECO Report.

Save ECO Script - nECO

Menu Bar: File -> Save ECO Script

After ECO changes are made, invoke this option to open the *Save ECO Script* form where the ECO script can be saved in the Novas ECO format, ICC Tcl format, or First Encounter format.

For the script file that *nECO* supports, refer to *Appendix A: Novas ECO Script* in the *nECO User Guide and Tutorial* for details about Novas ECO Format.

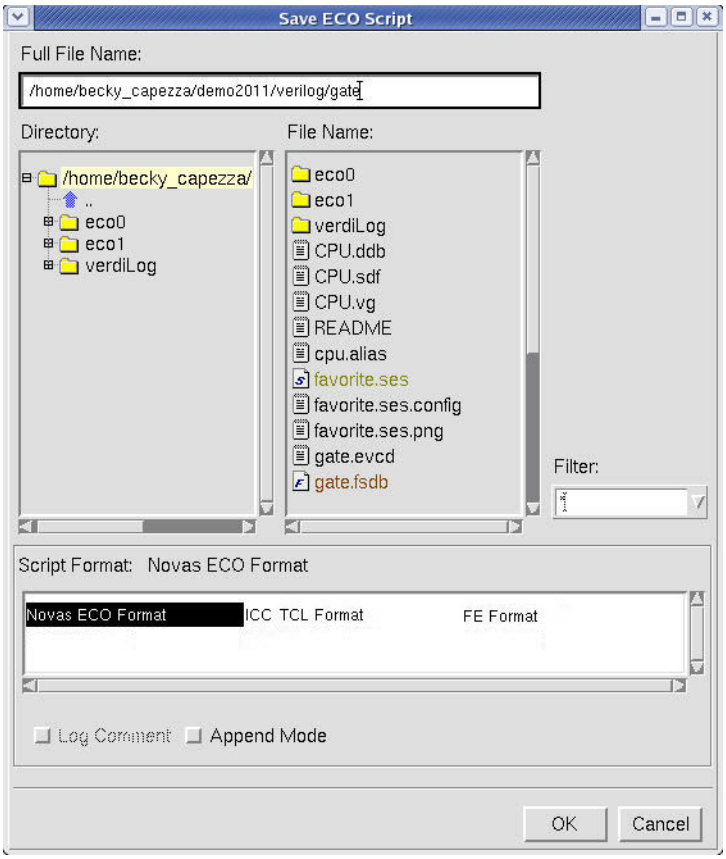


Figure: Save ECO Script Form

The *Save ECO Script* form includes the following options:

- **Log Comment:** When the **Log Comment** option is turned *on*, each log contains +C [comment], +T [tool], +D [day month date time year], +A [author], and [+/-] [Avanti_Option] [Avanti_Format]. When this option is turned *off*, only [+/-] [Avanti_Option] [Avanti_Format] is logged without +C information. Special characters ($, <, >, !) are not allowed on the +C line.
- **Append Mode:** When saving the ECO script to an existing file, if this option is turned *on*, the new script content is appended to the end of the existing content. If this option is turned *off*, current results overwrites the existing content. The default is *off*.

Save Spare Cell Script - nECO

Menu Bar: File -> Save Spare Cell Script

This option opens the *Spare Cell Report* form for viewing the directory structure and specifying a file to save the Astro spare-cell report to.

Save in HDL

Menu Bar: File -> Save in HDL

The option writes the source code corresponding to the instances to a specified output file, sorted by scope, source file name, begin line, and end line. Specify an output file name and keynote text string (the keynote text string appears as the first line of the output file).

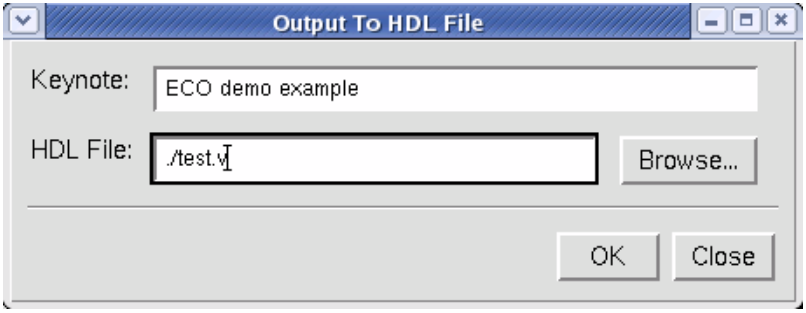


Figure: Output to HDL File Form

Commit Change

Menu Bar: File -> Commit Change

Toolbar Icon: 

This option opens the *Commit Changes Report* form which includes the **Affected Modules & Instances** tab, **Instance pin** tab, **Affected net** tab, **Module** tab, **Instance** tab, and **Net/Port** tab.

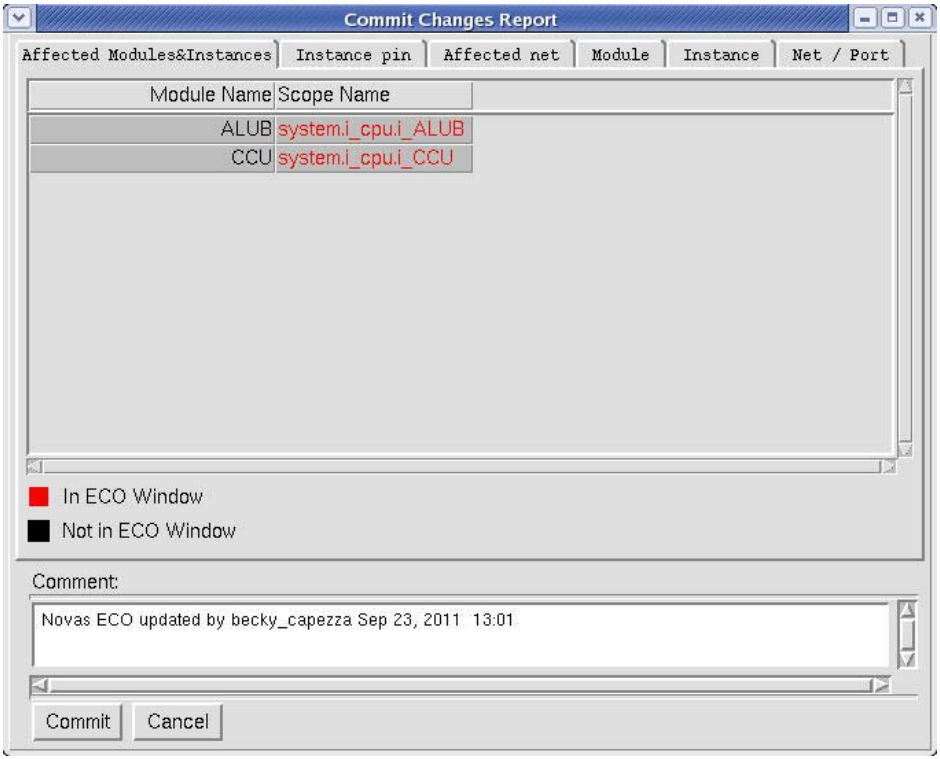


Figure: Commit Changes Report Form

Affected Modules & Instances Tab

This tab summarizes the modified modules and instances and provides a comment field.

For a module that has multiple instances, the modification to any instance also changes other instances even they are not in the *nECO* pane. These instances are also listed. To indicate the different situations, different colors are used to highlight these.

- An instance modified in the *nECO* pane is shown in red.
- An instance that is not in the *nECO* pane, but is influenced by another instance modification is shown in black.

An "*" following the tab name indicates one of the following: (1) input pins are floating, (2) instance pin has multiple drivers, or (3) net has multiple drivers or no driver.

Comment: Add a comment in this text field. The comment is visible in the *Commit Changes Report* and in front of the modified HDL source code. By

default, the comment contains the user name, date, and time. These can be set on the **Tools -> Preferences -> Schematics** page **-> ECO** page.

Instance pin Tab

This tab shows the number of changed instance pins such as **Floating Input Pins**, **Floating Output Pins**, and **Multiple Drive**.

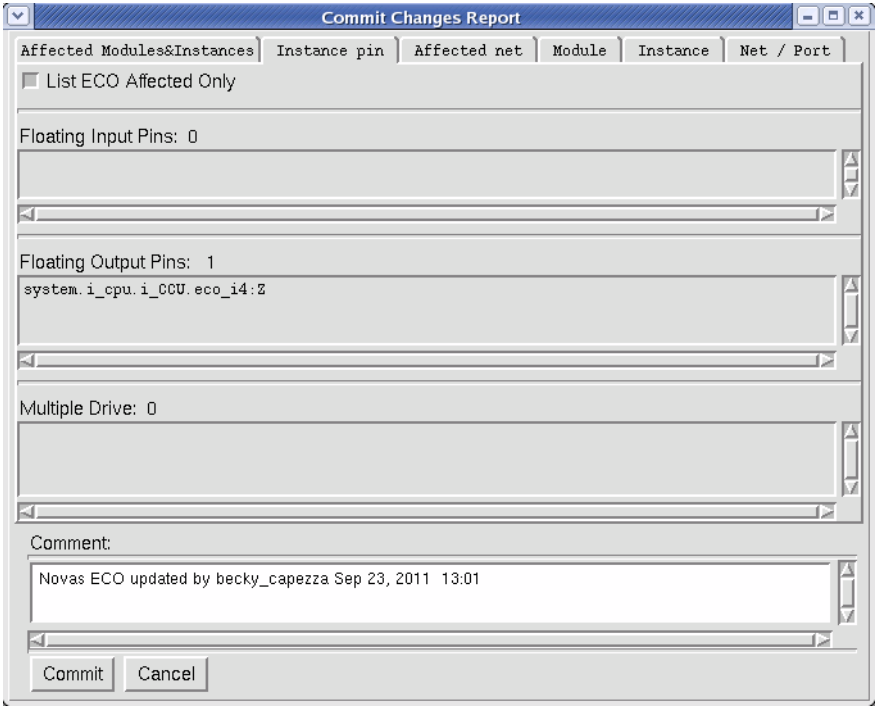


Figure: Commit Changes Report Form - Instance pin Tab

List ECO Affected Only: When this option is turned *on*, original floating pins are not shown and only ECO affected is shown in the report. The default is *on*.

Affected net Tab

This tab shows the modified nets. The net status of **No Driver**, **Multiple Driver**, and **No Load** is displayed.

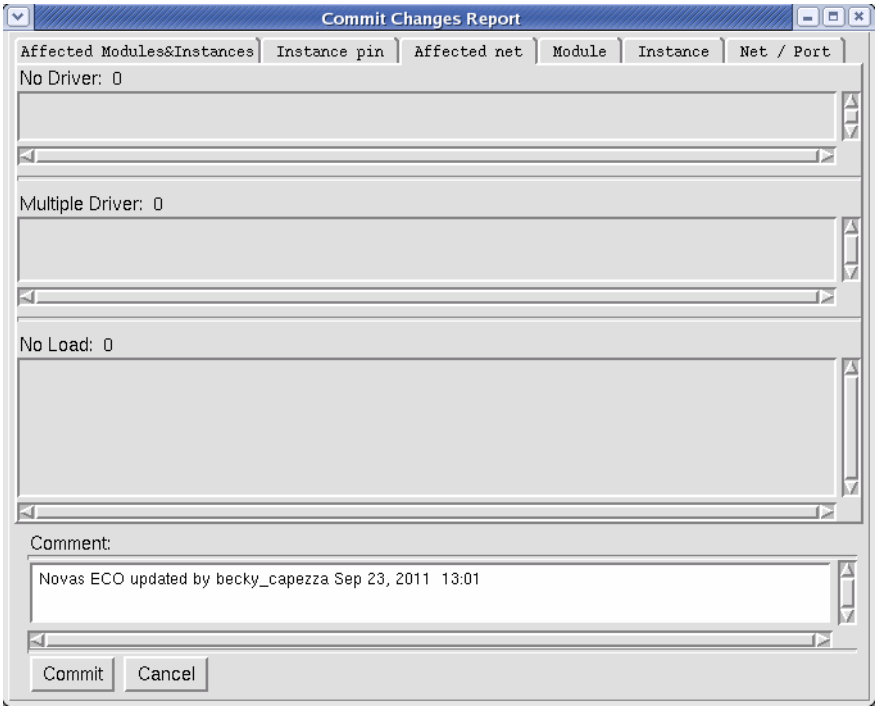


Figure: Commit Changes Report Form - Affected net Tab

Module Tab

This tab shows all the **Cloned Modules** activities that have been made in the *nECO* pane.

Instance Tab

This tab shows the number of changed instances, such as **Deleted Instance**, **Added Instance**, and **Renamed Instance**.

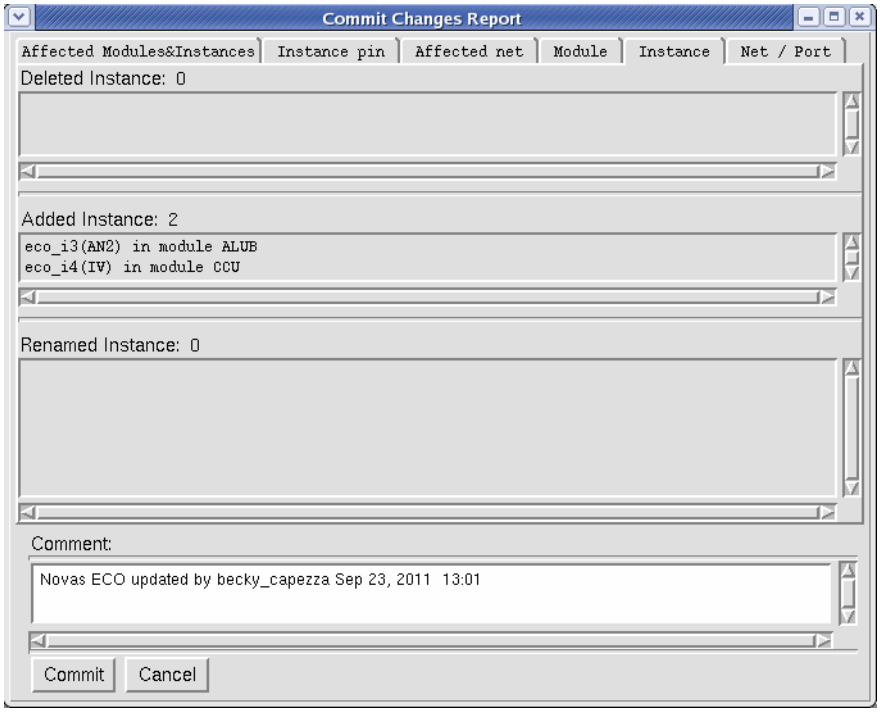


Figure: Commit Changes Report Form - Instance Tab

Net/Port Tab

This tab shows the number of added nets and ports, such as Added Net and Added Ports.

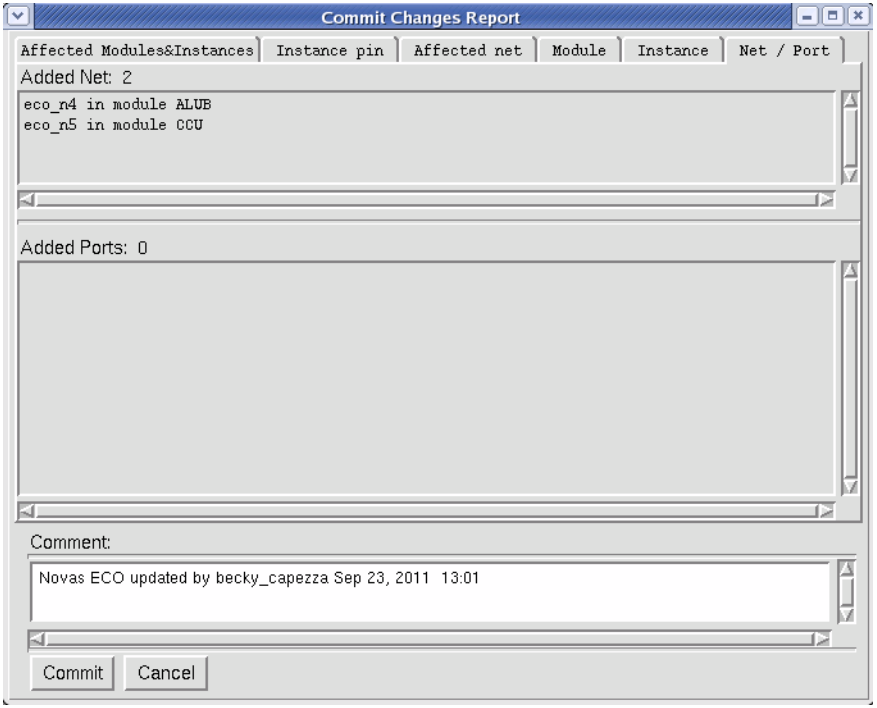


Figure: Commit Changes Report Form - Net/Port Tab

ECO Change Status

Menu Bar: File -> ECO Change Status

This option generates and opens a report summarizing the current changes. It is identical to the form associated with the **Commit Change** option except changes cannot be committed. This option is supported in the right mouse button option menu as well.

Print

Menu Bar: File -> Print

nECO uses the same **Print** functions as *nSchema*. Refer to the **File -> Print** option description in the *nSchema* chapter for details.

Capture Window

Menu Bar: File -> Capture Window

Refer to the **File -> Capture Window** option description in the *nSchema* chapter for details.

Close Window

Menu Bar: File -> Close Window

This option closes the current *nECO* window.

Edit Options

Undo

Menu Bar: Edit -> Undo

Toolbar Icon: 

Bind Key: Shift+U

This option undoes the previous action and removes whatever was added from the view.

Redo

Menu Bar: Edit -> Redo

Toolbar Icon: 

Bind Key: Shift+R

This option repeats the last action performed if possible.

Enable Hierarchy Connector Editing

Menu Bar: Edit -> Enable Hierarchy Connector Editing

This option reroutes and expand all collapsed connectors in the *nECO* window and make the connector selectable. By default, this option is *on.*

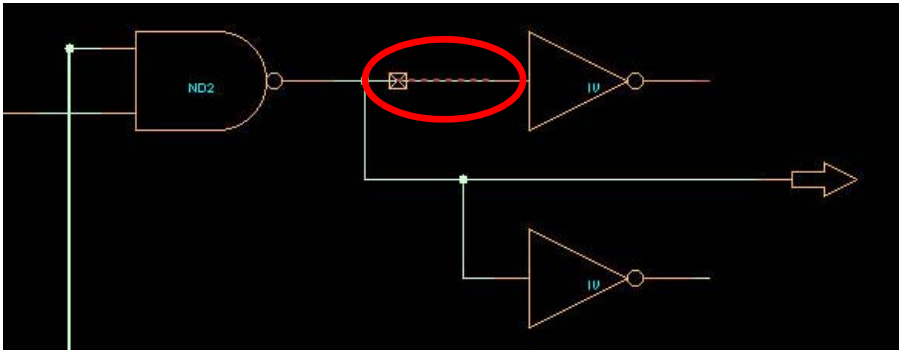


Figure: ECO Window with Enable Hier. Connector Editing off

In the figure above, only one connector *system.i_cpu.i_ALUB.S1* exists between nets *system.i_cpu.i_ALUB.S1* and *system.i_cpu.i_PCU.S1*. After turning **Enable Hier. Connector Editing** *on*, an additional connector *system.i_cpu.i_PCU.S1* is seen. All connectors are selectable and ready for editing.

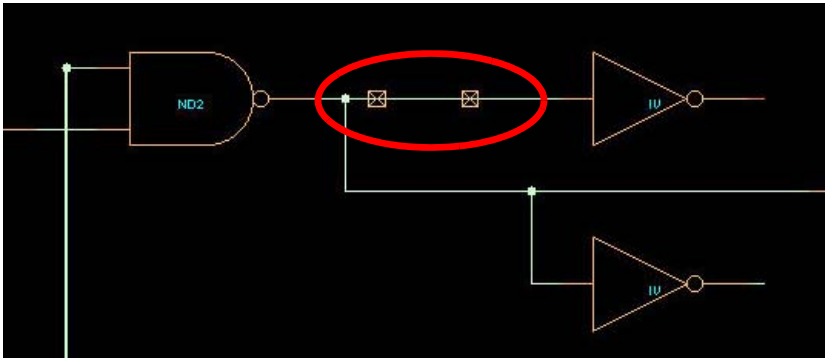


Figure: ECO Window after Turning Enable Hier. Connector Editing on

To delete the connection between nets *system.i_cpu.S1* and *system.i_cpu.i_PCU.S1*, select the **Disconnect Pin from Net** option from the right-click option menu to display the *Delete Connection* form.

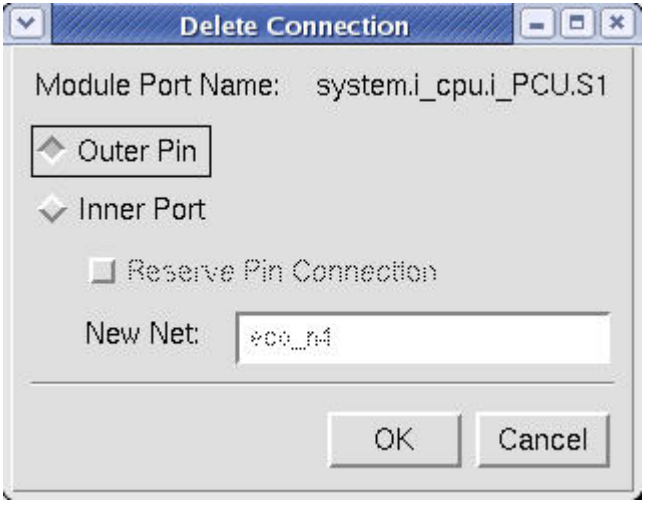


Figure: Delete Connection Form

In the *Delete Connection* form, select the **Outer Pin** option and click **OK**. The connector outer pin is floating:

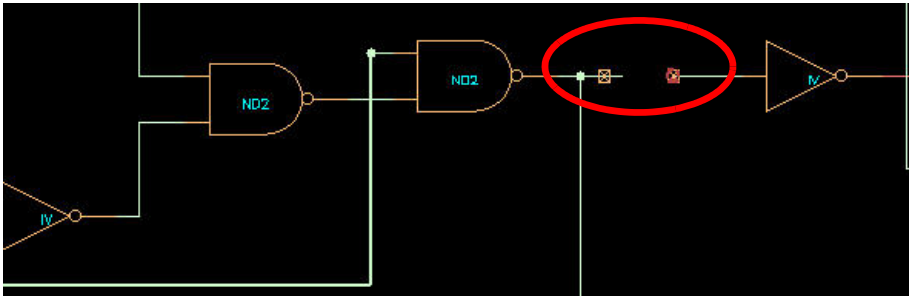


Figure: ECO Window after Delete Connection

After the changes are committed with the **File -> Commit Change** option, the *Commit Changes Report* form indicates the change as follows:

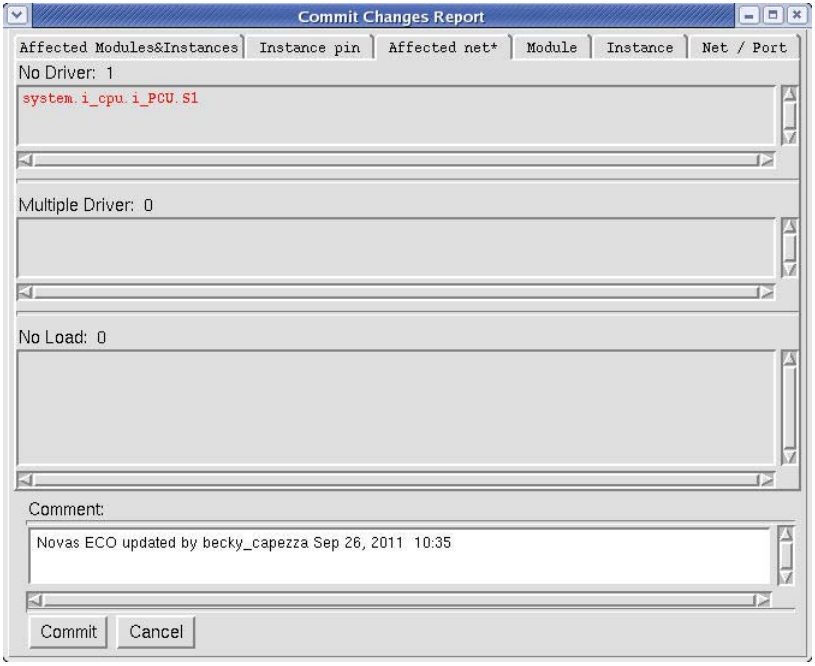


Figure: Commit Changes Report

Then the *nECO* window is shown below:

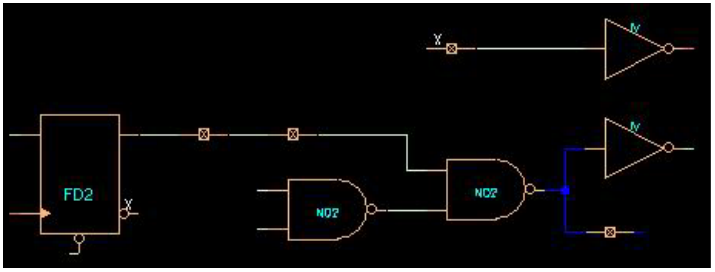


Figure: ECO Window after Commit Change

Rename Net

Menu Bar: Edit -> Rename Net

Bind Key: Shift+T

This option opens the *Rename Net Name* form where the net name can be changed.

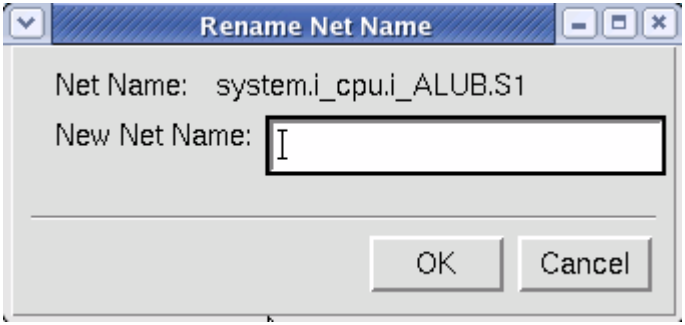


Figure: Rename Net Name Form

Rename Instance

Menu Bar: Edit -> Rename Instance

Bind Key: Shift+I

This option opens the *Rename Instance* form where the instance name can be changed.

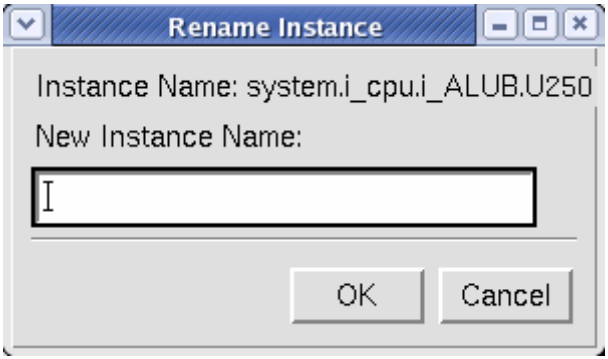


Figure: Rename Instance Form

Connect to Net

Menu Bar: Edit -> Connect to Net

This option connects the selected pin to the specified net. After an instance pin (one bit-width or multiple bit-width) is selected, invoke this option from the pull-down menu or from the right-click option menu to connect the selected instance pin to the specified net (net or concatenated net) in the *Net Connection* form.

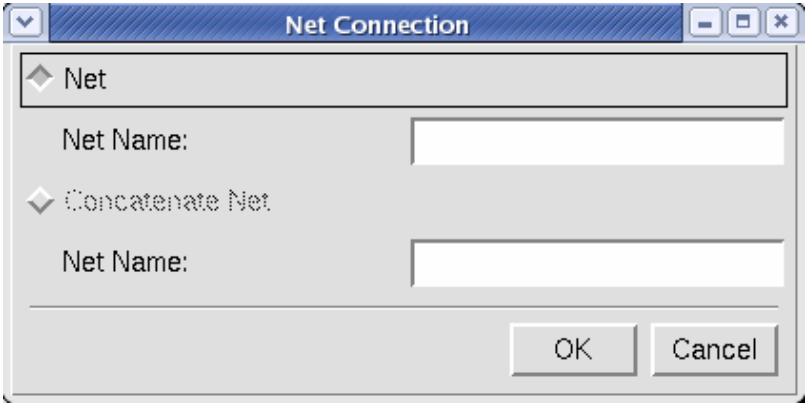


Figure: Net Connection Form

The pin can be connected to a net or a concatenated net by enabling either the **Net** or **Concate Net** options and entering net names in the **Net Name** text fields.

If the **Net** option is selected, the net is a normal net (scalar net or bus). In the **Net Name** text field, specify the local or full scope net name.

If the **Concate Net** option is selected, the net is a concatenated net. In the **Net Name** text field, specify only the local net name; specifying a scope path for the concatenated net is not allowed. The concatenated net name must be specified in Verilog style, such as {data[3:0], sd[2:1]}. The bit-width of the concatenated net and instance pin must be the same.

Clicking the **OK** button opens the *Net Management* form where the instance pin can be connected again or the net can be merged.

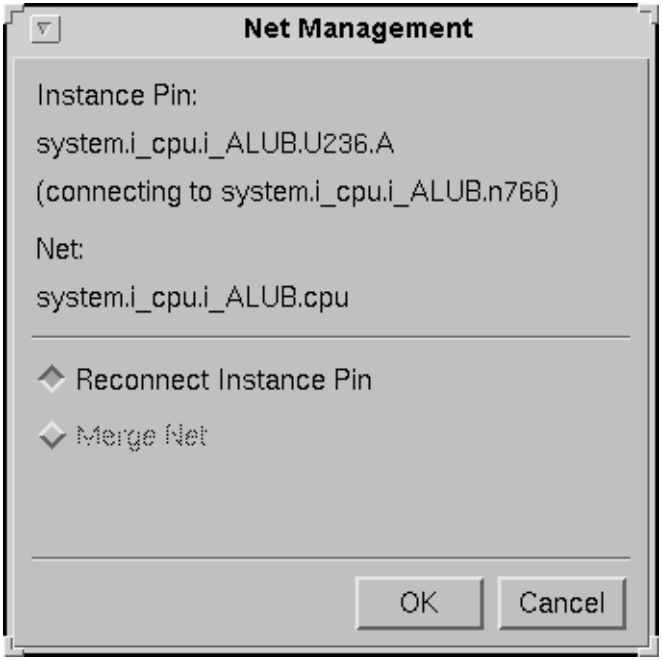


Figure: Net Management Form

If the **Reconnect Instance Pin** option is selected and the **OK** button is clicked, the current pin connections is deleted and the pin is connected to the desired net. If the **Merge Net** option is selected and the **OK** button is clicked, the current connected net and the specified net is merged together. Click the **Cancel** button to cancel the net connection.

NOTE: Continuous assignment can also be handled by the **Connect to Net** option.

Edit Concate Net

Menu Bar: Edit -> Edit Concate Net

This option edits the elements of the concatenated net. After a concatenated net is selected, invoke this option to open the *Edit Concate Net* form where elements of the concatenated net are listed and can be edited.

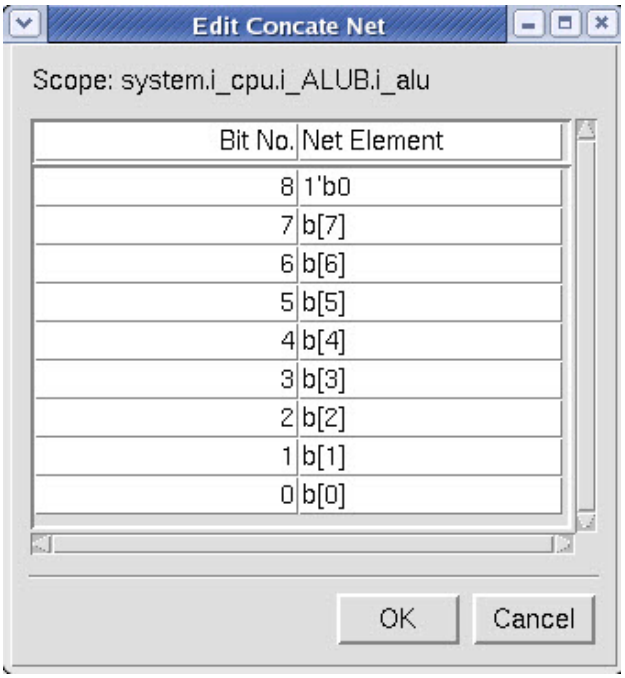


Figure: Edit Concate Net Form

Copy Instance

Menu Bar: Edit -> Copy Instance

This option copies the selected instance.

Paste Instance

Menu Bar: Edit -> Paste Instance

This option pastes the previously copied instance.

Remove from Viewing Objects

Menu Bar: Edit -> Remove from Viewing Objects

Toolbar Icon: 

Bind Key: <Key> Delete

This option removes the selected object from the *nECO* window.

Delete Connection

Menu Bar: Edit -> Delete Connection

Toolbar Icon: 

Bind Key: Shift+D

This option deletes the connection of the selected instance pin or deletes the selected net and its connected instance pins in the *nECO* window.

Make Connection

Menu Bar: Edit -> Make Connection

Toolbar Icon: 

Bind Key: M

This option connects an instance pin to another instance pin or to a net in the *nECO* window. Two methods are available:

- Select the instance pins to be connected first and then invoke this option to complete the connection.
- Select an instance pin and then invoke this option. A dangling line is shown. Stop the dangling line at a desired instance pin or net to complete the connection.

When this option is invoked and the selected instance port is floating or the net connection crosses hierarchical boundaries, the *New Ports/Nets* form opens to create ports and upper/lower nets.

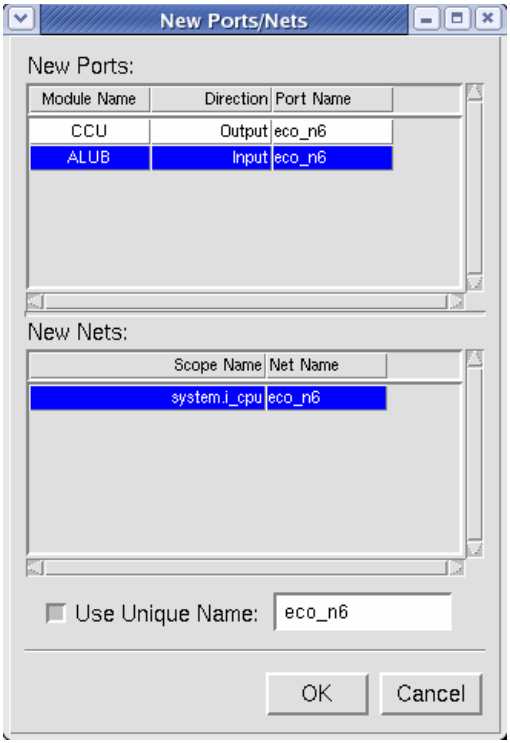


Figure: New Ports/Nets Form

Turn the **Use Unique Name** option *on* to specify a port name for modules. If an instance port connects to an existing net, the net name is created as the port name of its module, so that netlist changes are minimized.

This option also connects net to net. If the bit-width of the selected nets match after two signals are selected in the *nECO* pane, the *Net Merge* form is opened after this option is invoked. The net to be reserved from the two nets must be specified.

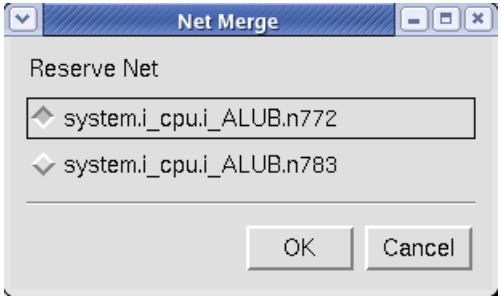


Figure: Net Merge Form

If the bit-width of the two nets do not match, the *Bit Range Select* form is opened. The bus range to be merged to must be specified. The **Start Bit** and **End Bit** fields are for specifying the begin and end bits respectively for the reserved net.

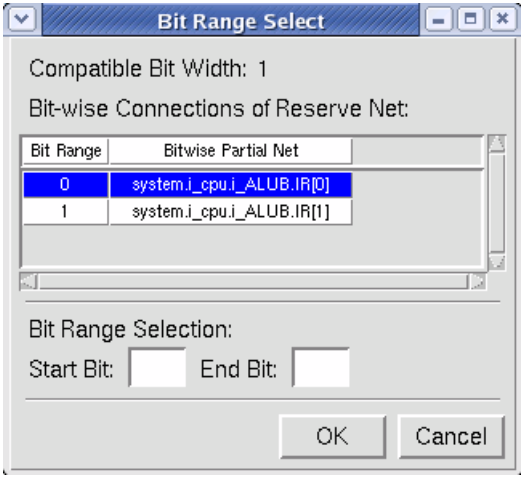


Figure: Bit Range Select Form

If the net-merging results in multiple-drive connections, the *Multiple Driving Pins* form is opened.

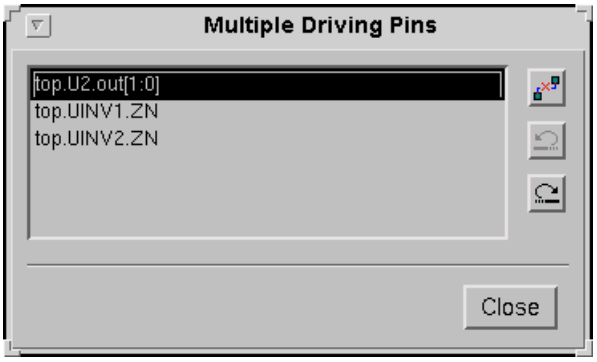


Figure: Multiple Driving Pins Form

Highlight a pin in the table and click the  button to delete the pin connection. Click the  button to undo the previous operation. Click the  button to redo the previous operation.

Connect Bus Pin to Net

Menu Bar: Edit -> Connect Bus Pin to Net

This option opens the *Connect Bus Pin to Net* form to specify each pin bit with an action and connects the bus pin to a net in the *nECO* pane.

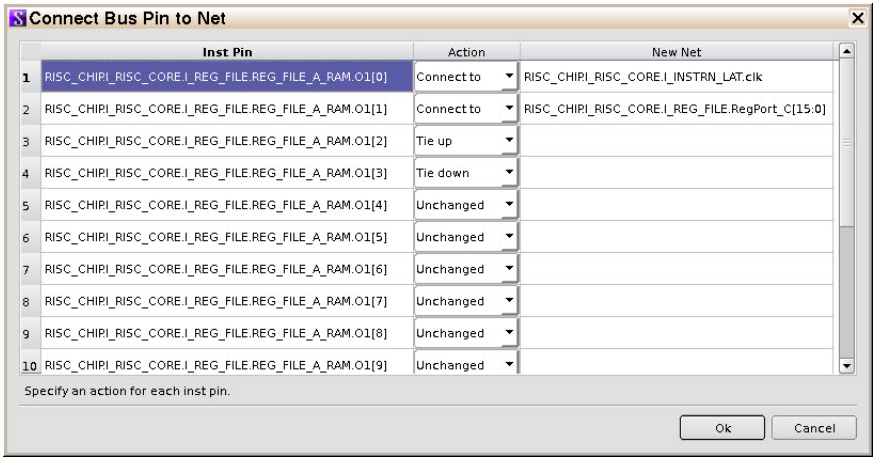


Figure: Connect Bus Pin To Net Form

The *Connect Bus Pin to Net* form includes the following columns, field, and buttons:

- **Inst Pin:** List all bits of the selected instance pin.
- **Action:** Specify an action for the bit from the pull-down menu. The following options are available:
  - **Unchanged:** Connect the bit to the corresponding net.
  - **Connect to:** Connect the bit to a new net specified in the **New Net** column.
  - **Tie up:** Tie the selected bit to high (power).
  - **Tie down:** Tie the selected bit to low (ground).
- **New Net:** Specify a new net for the bit that is specified with the **Connect to** action. Enter the name of the net manually or drop a net in this text field by dragging it from the *nECO* pane.
- **OK:** Connect the bus pin to the net according to settings in the *Connect Bus Pin to Net* form and close the form.
- **Cancel:** Cancel changes to the *Connect Bus Pin to Net* form and close the form.

Disconnect Bus Pin from Net

Menu Bar: Edit -> Disconnect Bus Pin from Net

This option opens the *Disconnect Bus Pin from Net* form to specify pin bits and disconnect the bus pin from the net in the *nECO* pane.

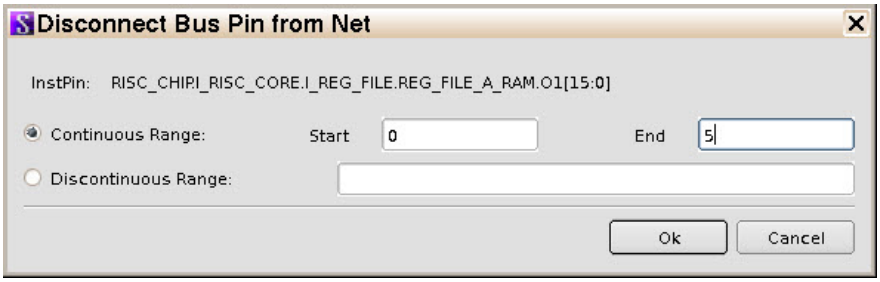


Figure: Disconnect Bus Pin from Net Form

The *Disconnect Bus Pin from Net* form includes the following options, fields, and buttons:

- **InstPin:** Display the name of the selected instance pin.
- **Continuous Range:** When this option is turned *on,* specify the continuous bit range that is to be disconnected from the net. Enter the range number in the **Start** and **End** text field to specify the bit range.
- **Discontinuous Range:** When this option is turned *on,* specify the discontinuous bit range that is to be disconnected from the net. Enter range numbers in the text field following this option to specify the bit range. The number delimiter is the comma (,) character.

Tie Up

Menu Bar: Edit -> Tie Up

Bind Key: P

This option ties the selected instance pin to high (power).

Tie Down

Menu Bar: Edit -> Tie Down

Bind Key: W

This option ties the selected instance pin to low (ground).

Create Port

Menu Bar: Edit -> Create Port

This option creates a port the **Module Name**, the **Port Name**, and the **Port Direction** are entered in the *Create Port* form.

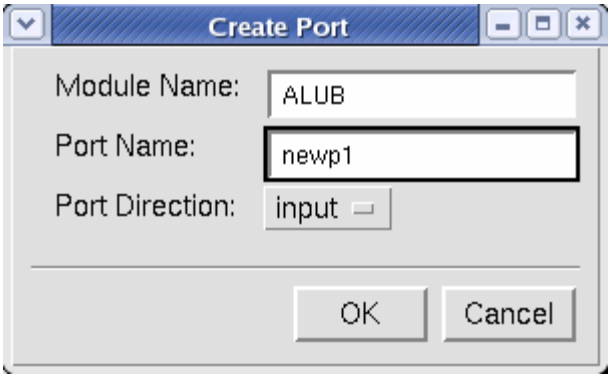


Figure: Create Port Form

If the specified module has multiple instantiations, the new port is created for all instantiations and an *Add Port* form is opened where the new ports of the instances to be displayed in the *nECO* pane must be specified.

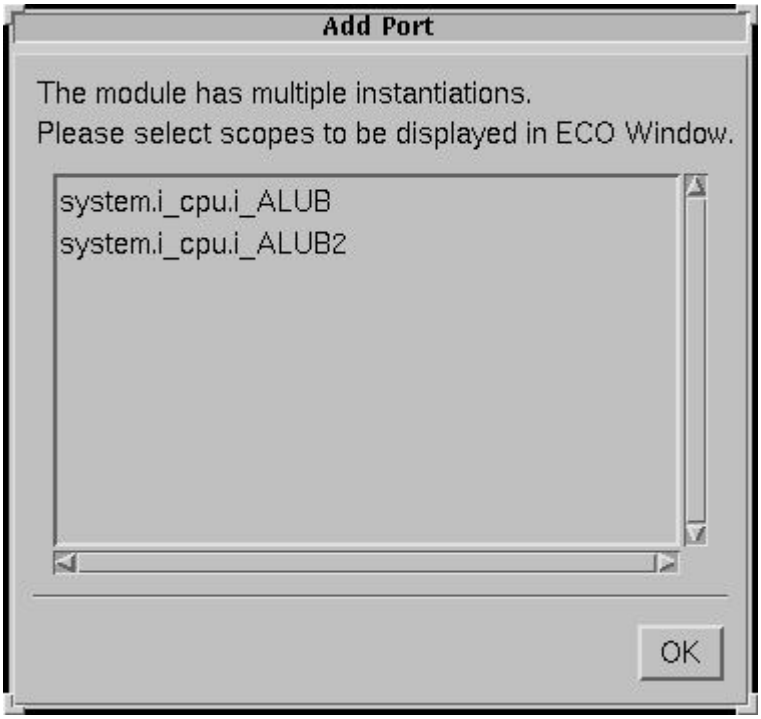


Figure: Add Port -- Multiple Instances

Add Buffers (Non-Freeze Silicon Mode)

Menu Bar: Edit -> Add Buffers

Bind Key: Shift+B

This option adds a signal bit net, connector, and instance pin from the cell library to the design. It is disabled if a net or instance port connecting to any bus assignment is selected. After a net, connector, or pin is selected, invoke this option to open the *Add Buffers* form.

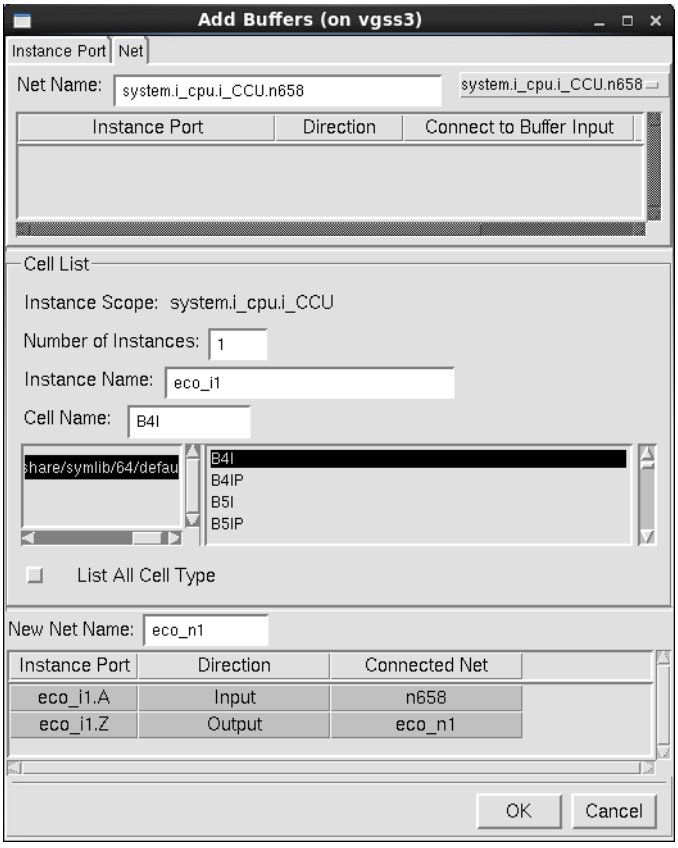


Figure: Add Buffers Form

The *Add Buffers* form includes the **Instance Port** tab and the **Net** tab. If a buffer is added to an instance port, the **Instance Port** tab is selected and the selected instance port name is displayed. If a buffer is added to a net, this form automatically switches to the **Net** tab where all nets are shown and the net to insert the buffer into can be selected.

Net Name: This field shows the net that has been selected. The selection field next to the **Net Name** text field shows all the nets connected to the selection.

The table below the **Net Name** text field shows net connection information. The **Net Name** table includes the following columns:

- **Instance Port:** Lists all ports that the selected net is connected to.
- **Direction:** Indicates the direction (either input or output) of the instance port.

- **Connect to Buffer Input:** A check (v) to show if the selected instance port is connected to a buffer input. Click the column to change the connection and *nECO* connects the new net to the buffer input port automatically.
- **Connect to Buffer Output:** A check (v) to show if the selected instance port is connected to a buffer output. Click the column to change the connection and *nECO* connects the new net to the buffer output port automatically.

The **Cell List** section includes the **Cell List** table showing the cell library name and cells in the cell library. Other fields and options are described below:

- **Instance Scope:** Displays the name of the current instance scope.
- **Number of Instances**: Enter the total number of additional buffers that can be added to the specified scope.

NOTE: In the freeze silicon mode, this option is disabled because it does not allow more than 1 buffer to be added in the scope.

- **Instance Name:** Enter a new instance name in this text field or select a name shown in the table below the **Cell List**.
- **Cell Name:** Enter or select the cell name for the new cell.
- **List All Cell Type:** Turn this option *on* to view all the cell types.

The **New Net Name** section includes the **New Net Name** table showing port, direction and connected net(s) of the new net. The connected net can be changed by clicking the desired **Connected Net** cell. For example, click the *CH0* cell and a list widget is shown to change the net to *eco_n_1*.

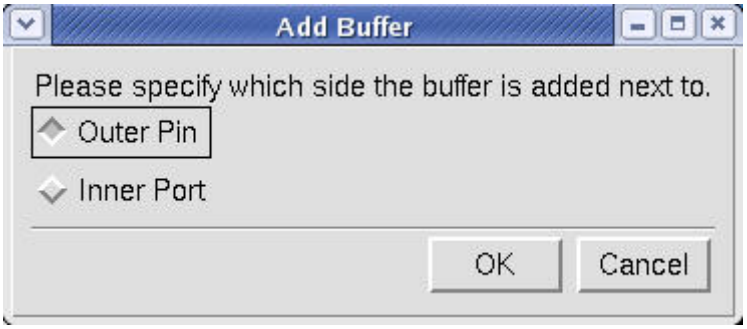
The **New Net Name** section also includes the following field:

- **New Net Name:** Specify the net name in this field.

NOTE: Depending on the type of net and where it is selected, a different *Add Buffer* form is shown.

When a hierarchical connector (a square with an “x” inside) is selected and the **Add Buffers** option is invoked, a reduced *Add Buffer* form

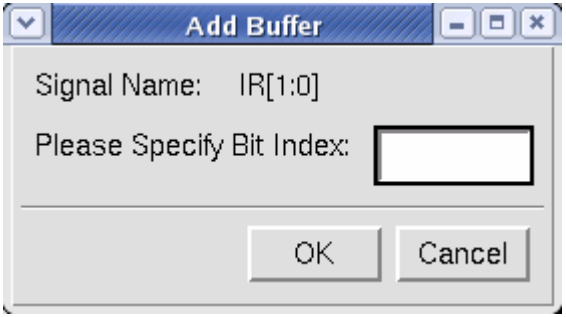
opens as shown below.



To select a connector, the **Edit -> Enable Hier. Connector Editing** option must be enabled to make connector editing possible.

If an **Assign** statement is selected, the buffer is added to the **Inner Port** automatically.

If a connector bus is selected and the **Add Buffers** option is invoked, a bit index value must be entered in the *Add Buffer* form as shown below.



After the bit index is specified, click the **OK** button to open the complete *Add Buffer* form as described previously. This is the same form that opens when a signal is selected and the **Add Buffers (Non-Freeze Silicon Mode)** option is invoked.

Add Buffers (Freeze Silicon Mode)

Add Buffers (with Manage Spare Cells by Cell Number Selected)

Menu Bar: Edit -> Add Buffers

Bind Key: Shift+B

This option is enabled after a spare cell is imported by invoking the **Edit -> Spare Cell** option which opens the *Spare Cell Manager* form. Click the **Load Spare Cell Number from File** option to import the spare cell.

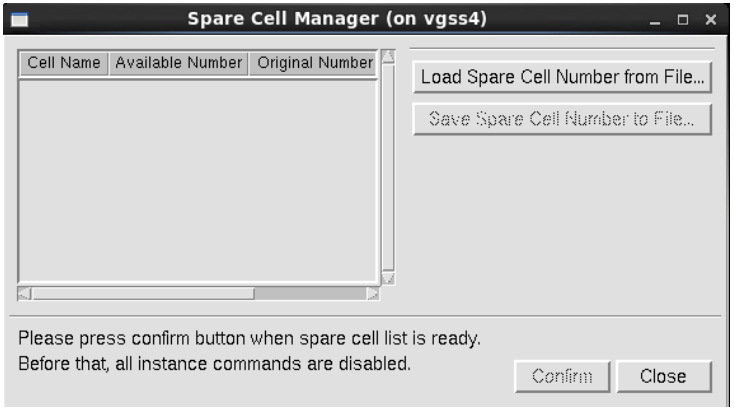


Figure: Spare Cell Manager Form

After the spare cells are imported, if the **Manage Spare Cells by Cell Number** option is turned *on* in the **Tools -> Preferences -> Schematics** page **-> ECO** page **-> Freeze Silicon** page, invoking the **Add Buffers** option opens the *Add Buffers* form.

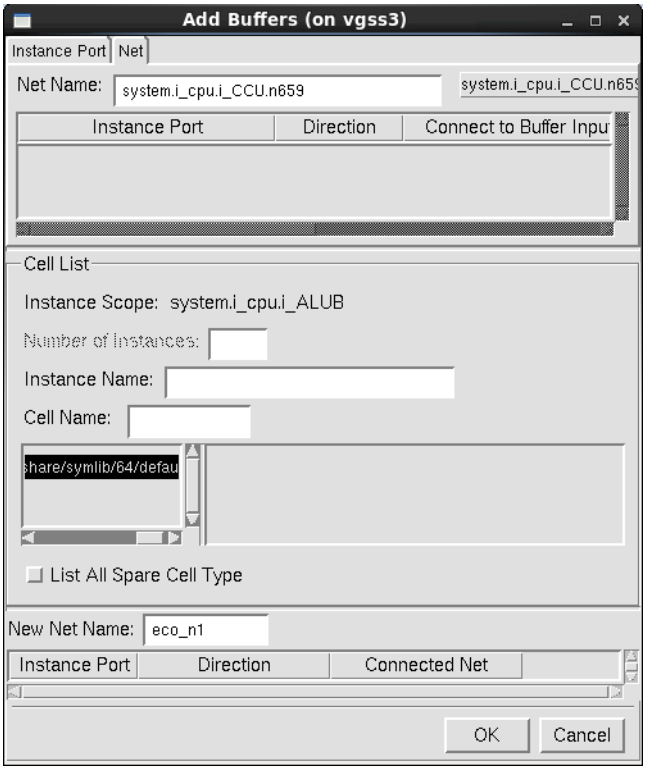


Figure: Add Buffers Form

The behavior of options in the *Add Buffers* form is similar to the form opened by invoking the **Add Buffers (Non-Freeze Silicon Mode)** option. Refer to the **Add Buffers (Non-Freeze Silicon Mode)** option for details.

Add Buffers (with Manage Spare Cells by Instance Name Selected)

Menu Bar:	Edit -> Add Buffers
Bind Key:	Shift+B

This option is enabled after a spare cell is imported by invoking the **Spare Cell** option (which opens the *Spare Cell* form).

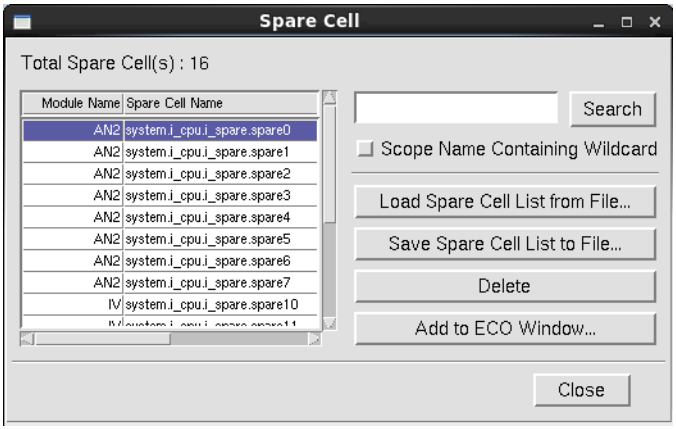


Figure: Spare Cell Form

When the **Manage Spare Cells by Instance Name** option is turned *on* and the **Spare Cell Quantity Control** option is turned *off* in the **Tools -> Preferences -> Schematics page -> ECO** page **-> Freeze Silicon** page, invoking the **Add Buffers** option opens the *Add Buffers* form.

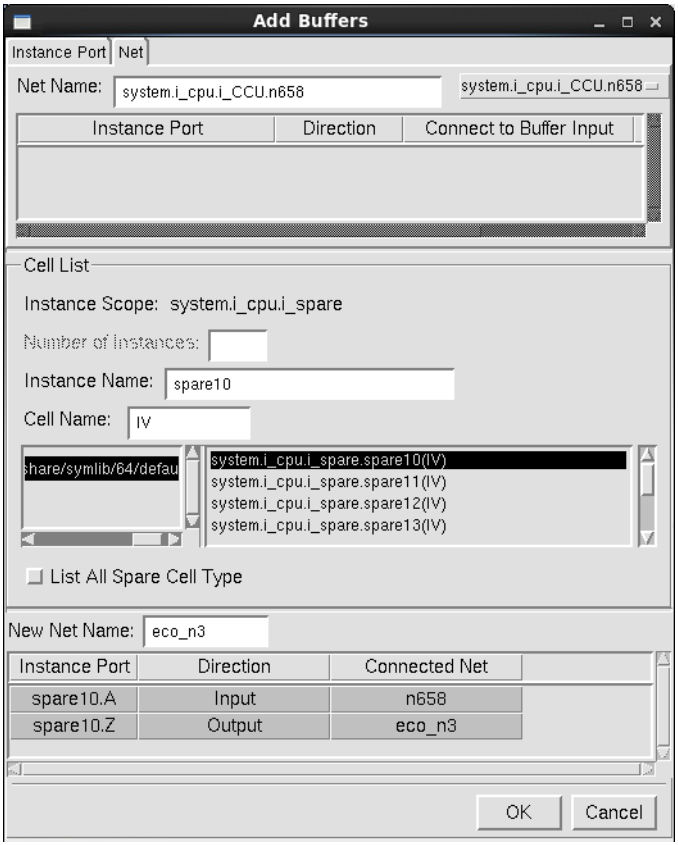


Figure: Add Buffers Form

The behavior of options in the *Add Buffers* form is similar to the form opened by invoking the **Add Buffers (Non-Freeze Silicon Mode)** option. Refer to the **Add Buffers (Non-Freeze Silicon Mode)** option description for details.

Add Buffer by Name (with Manage Spare Cells by Instance Name and Spare Cell Quantity Control Enabled)

Menu Bar: Edit -> Add Buffer ->Add Buffer by Name

This option is enabled after a spare cell is imported by invoking the **Spare Cell** option (which opens the *Spare Cell* form).

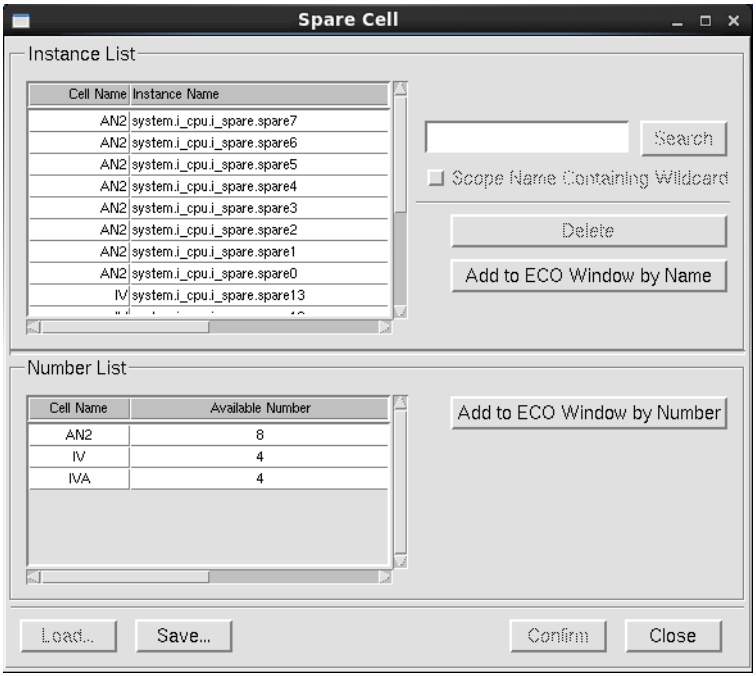


Figure: Spare Cell Form

In the **Instance List** section, after an instance is selected and the **Add to ECO Window by Name** button is clicked, the selected spare cell is removed from the **Instance List** table and is added to the *nECO* window automatically. When any instances on the **Instance List** are added, the number in the **Available Number** column of the **Number List** table is decreased appropriately.

In the **Number List** section, after an instance is selected and the **Add to ECO Window by Number** button is clicked and then the **Confirm** button is clicked, the *Add Instance* form opens.

After the spare cell is imported, if the **Manage Spare Cells by Instance Name** and the **Spare Cell Quantity Control** options are turned *on* in the **Tools -> Preferences -> Schematics** page **-> ECO** page **-> Freeze Silicon** page, the **Add Buffer by Name** option is enabled. Invoking the **Add Buffer by Name** option opens the *Add Buffers* form.

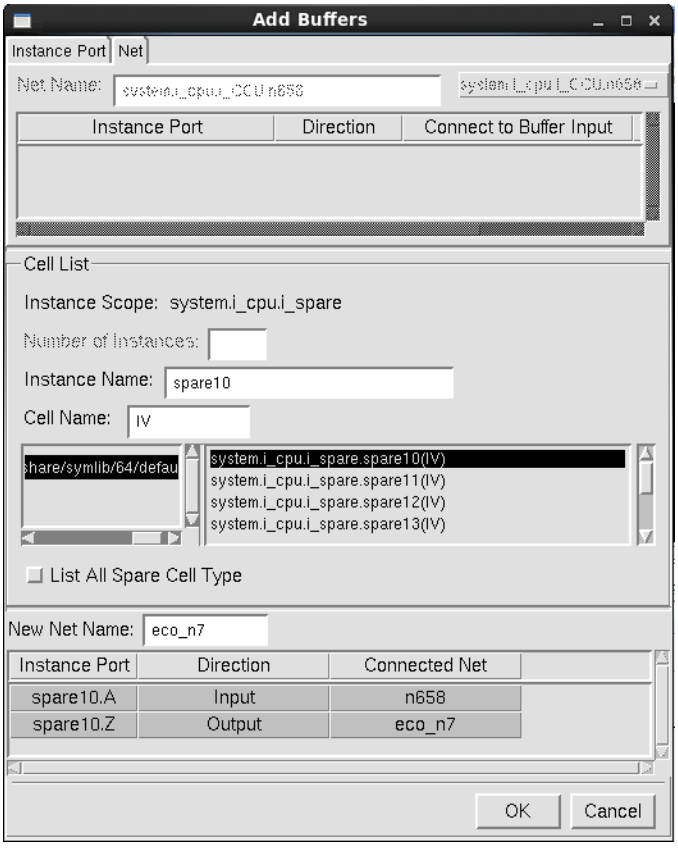


Figure: Add Buffers Form

The behavior of options in the *Add Buffers* form is similar to the form opened by invoking the **Add Buffers (Non-Freeze Silicon Mode)** option. Refer to the **Add Buffers (Non-Freeze Silicon Mode)** option for details.

Add Buffer by Number (with Manage Spare Cells by Instance Name and Spare Cell Quantity Control Enabled)

Menu Bar: Edit -> Add Buffer -> Add Buffer by Number

After the spare cell is imported from the **Spare Cell** option, if the **Manage Spare Cells by Instance Name** and the **Spare Cell Quantity Control** options are turned *on* in the **Tools -> Preferences -> Schematics** page **-> ECO** page **->**

Freeze Silicon page, the **Add Buffer by Number** option is enabled. Invoking the **Add Buffer by Number** option opens the *Add Buffers* form.

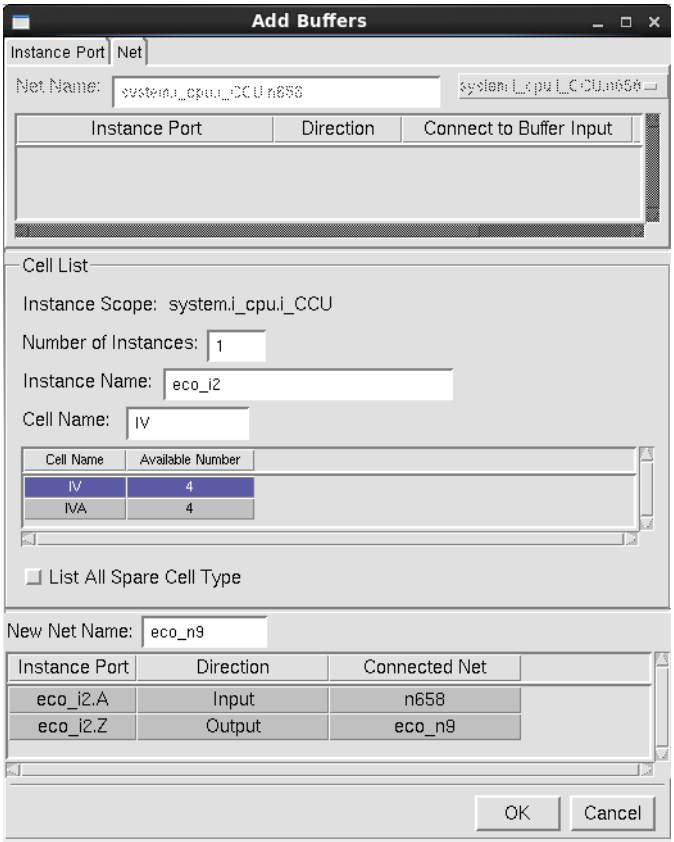


Figure: Add Buffers Form

The behavior of options in the *Add Buffers* form is similar to the form opened by invoking the **Add Buffer by Name** option. The only difference is the **Cell List** table. The **Add Buffer by Name** option shows the cell list by the cell names and the **Add Buffer by Number** option shows the cell list by the cell number. For more detailed usage, refer to the **Add Buffers (Non-Freeze Silicon Mode)** option description.

Delete Buffer

Menu Bar: Edit -> Delete Buffer

Bind Key: Shift+K

This option deletes the selected buffer (for example, BUF, NOT) from a net.

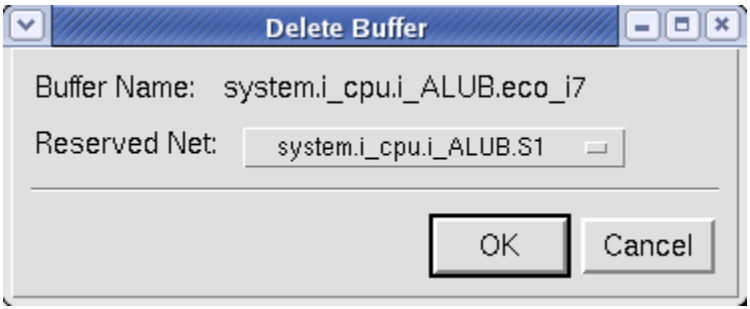


Figure: Delete Buffer Form

Buffer Name: Shows the instance name of the buffer.

Reserved Net: Lists all nets the buffer is connected to. The selected net is reserved after the buffer is deleted.

NOTE: The **Delete Buffer** option can also handle continuous assignment.

Add Instance (Non-Freeze Silicon Mode)

Menu Bar: Edit -> Add Instance

Bind Key: I

This option adds an instance from the cell library to the design. When adding an instance, the scope to add the instance into (destination scope) must be specified. All the scopes referenced in the *nECO* pane are listed in the *Add Instance* form. Select one of these scopes or enter the name of another scope.

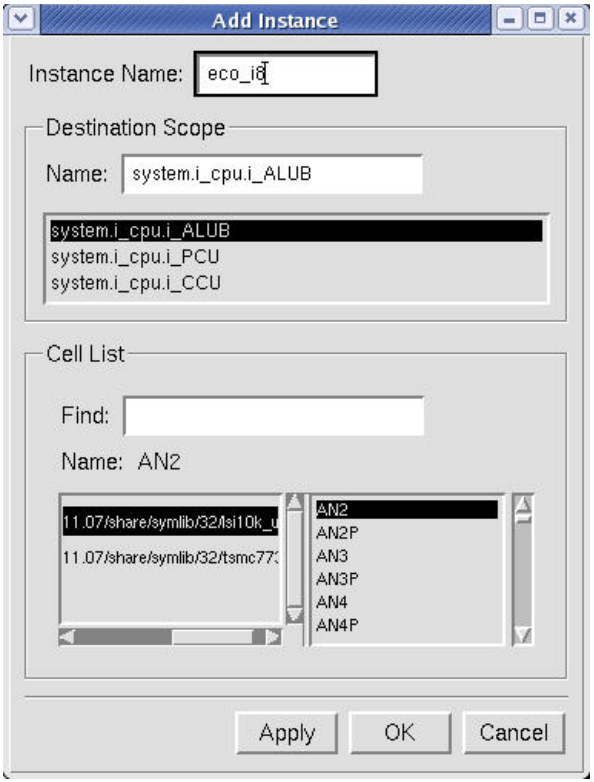


Figure: Add Instance Form

Instance Name: Specify the name of the newly added instance in this text field.

Destination Scope: In this section, enter the destination scope for the new instance. Type the scope manually in the **Name** text field, or select the scope name from the list.

Cell List: In this section, find the cell type for the new instance. Type the name of the desired cell in the **Find** text field. After pressing the enter key, the **Name** field displays the letters type and cells matching the search pattern is shown in the cell list. The wildcard character (*) is supported for the **Find** text field. If the **Name** field is left empty, the cell list lists all of the cell names.

Add Instance (Freeze Silicon Mode)

Add Instance (with Manage Spare Cells by Cell Number Selected)

Menu Bar: Edit -> Add Instance

Bind Key: I

This option is enabled after a spare cell is imported by invoking the **Spare Cell** option (which opens the *Spare Cell Manager* form). In the *Spare Cell Manager* form, click the **Load Spare Cell Number from File** option to import a spare cell.

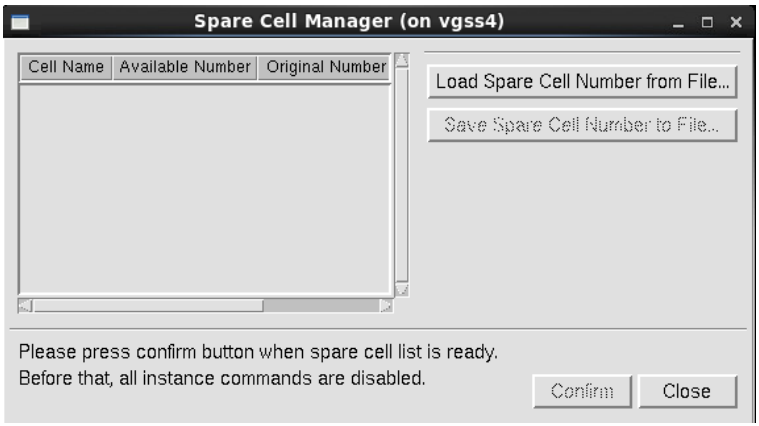


Figure: Spare Cell Manager Form

If the **Manage Spare Cells by Cell Number** option is turned *on* in the **Tools -> Preferences -> Schematics** page **-> ECO** page **-> Freeze Silicon** page, invoking the **Add Instance** option opens the *Add Instance* form.

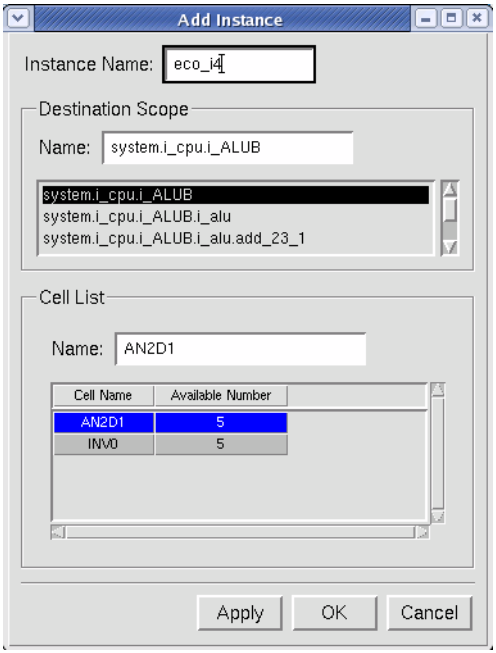


Figure: Add Instance Form

The behavior of the options in the *Add Instance* form is similar to the *Add Instance* form invoked by the **Add Instance by Number** option. Refer to the **Add Instance by Number (with Manage Spare Cells by Instance Name and Spare Cell Quantity Control Enabled)** option description for details.

Add Instance (with Manage Spare Cells by Instance Name Selected)

Menu Bar: Edit -> Add Instance

Bind Key: I

When the **Manage Spare Cells by Instance Name** option is turned *on* and the **Spare Cell Quantity Control** option is turned *off* in the **Tools -> Preferences -> Schematics** page **-> ECO** page **-> Freeze Silicon** page, invoking the **Add Instance** option opens the *Spare Cell* form where the desired instance(s) can be added.

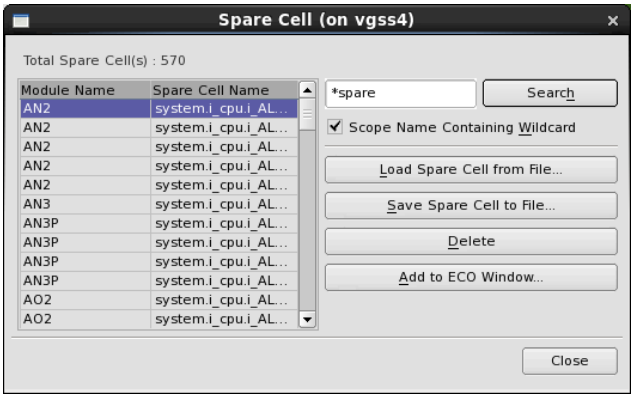


Figure: Spare Cell Form

All the available instances with their module name and spare cell name are listed on the left table of the *Spare Cell* form. Select an instance in the **Total Spare Cell(s)** table and add it to the *nECO* window by clicking the **Add to ECO Window** button. When the selected instance is successfully added to the *nECO* window, the instance is removed from the **Total Spare Cell(s)** table.

Add Instance by Name (with Manage Spare Cells by Instance Name and Spare Cell Quantity Control Enabled)

Menu Bar: Edit -> Add Instance by Name

When the **Manage Spare Cells by Instance Name** and the **Spare Cell Quantity Control** options in the **Tools -> Preferences -> Schematics** page **-> ECO** page **-> Freeze Silicon** page are turned *on*, invoking the **Add Instance by Name** option opens the *Spare Cell* form.

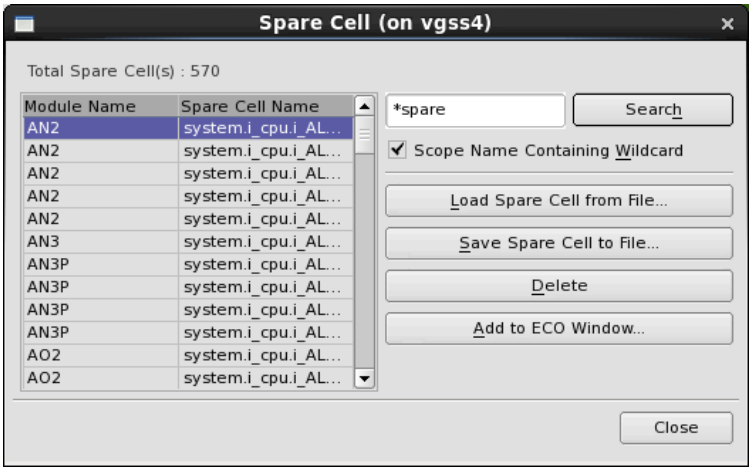


Figure: Spare Cell Form

In the **Instance List** section, after an instance is selected and the **Add to ECO Window by Name** button is clicked, the selected spare cell is removed from the **Instance List** table and is added to the *nECO* window automatically. When any instances are added from the **Instance List**, the number in the **Available Number** column of the **Number List** table is decreased appropriately.

In the **Number List** section, after an instance is selected and the **Add to ECO Window by Number** button is clicked and then the **Confirm** button is clicked, the *Add Instance* form (which is also opened by invoking the **Add Instance by Number** option) opens. Refer to the **Add Instance by Number (with Manage Spare Cells by Instance Name and Spare Cell Quantity Control Enabled)** option description for details.

Add Instance by Number (with Manage Spare Cells by Instance Name and Spare Cell Quantity Control Enabled)

Menu Bar: Edit -> Add Instance by Number

When the **Manage Spare Cells by Instance Name** and the **Spare Cell Quantity Control** options in the **Tools -> Preferences -> Schematics** page **-> ECO** page **-> Freeze Silicon** page are turned *on*, invoking the **Add Instance by Number** option opens the *Add Instance* form.

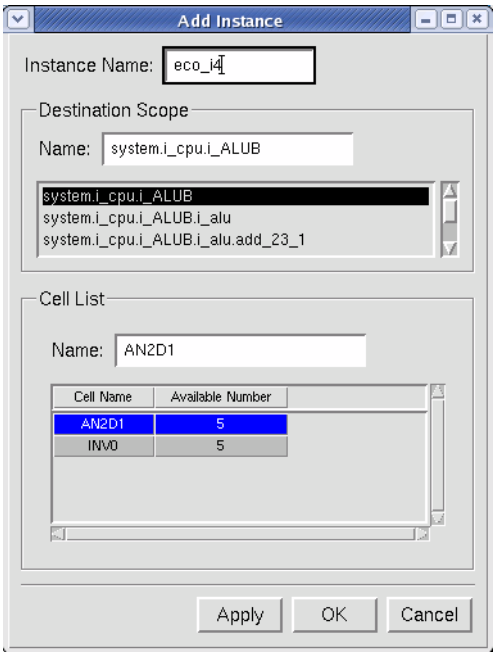


Figure: Add Instance Form

After a desired instance is highlighted on the **Cell List** table and the **Apply** or **OK** buttons are clicked, the selected instance is added to the scope specified as the **Destination Scope** in the *nECO* window. The number in the **Available Number** column of the **Cell List** table is decreased by one each time the **Apply** button is clicked.

Instance Name: Shows a temporary instance name which is generated automatically. Enter a desired name in the text field to replace it.

Destination Scope: This section lists all the scopes referenced in the *nECO* window. Specify the scope to add the instance to.

Cell List: The table displays the cell name and the total number of cells available.

Delete Instance

Menu Bar: Edit -> Delete Instance

Bind Key: E

This option deletes the selected instance and its port connections. If **Freeze Silicon Mode** is turned on, the deleted instance is added to the spare cell list.

NOTE: **Delete Instance** can also handle continuous assignment.

Replace Instance (Non-Freeze Silicon Mode)

Menu Bar: Edit -> Replace Instance

Bind Key: R

This option replaces a cell from the cell library to this design. In Non-Freeze Silicon mode, invoking the **Replace Instance** option with a cell selected in the *nECO* window opens the *Replace Instance* form.

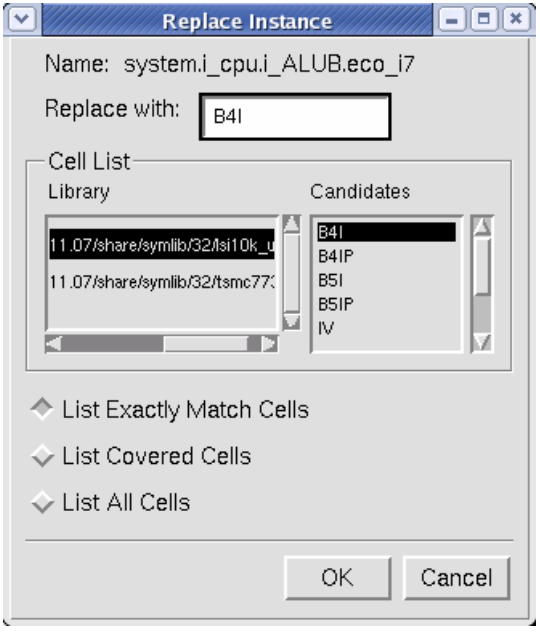


Figure: Replace Instance Form

Name: Displays the current instance name.

Replace with: This field shows the cell name type. Enter the name manually or select a name from the **Candidates** list.

Library: Lists the library path of the cells.

Candidates: Lists the available cell names.

Click the **List Exactly Match Cells**, **List Covered Cells**, or **List All Cells** options to display different replacement cell results in the **Candidates** table. Only one option may be selected at a time.

- **List Exactly Match Cells:** Displays the cells that are an exact pin-to-pin match with the original instance.
- **List Covered Cells:** Displays all cells that can cover the pins of the original instance.
- **List All Cells:** Displays all cell names regardless of whether the pin count matches the original instance.

If either of the **List Covered Cells** or **List All Cells** options are selected, the correct port mapping must be specified in the *Port Mapping* form after the **OK** button is clicked.

Replace Instance (Freeze Silicon Mode)

Replace Instance (with Manage Spare Cells by Cell Number Selected)

Menu Bar:	Edit -> Replace Instance -> Replace Instance
Bind Key:	R

This option is enabled after a spare cell is imported by invoking the **Spare Cell** option which opens the *Spare Cell Manager* form. Click the **Load Spare Cell Number from File** button to import a spare cell.

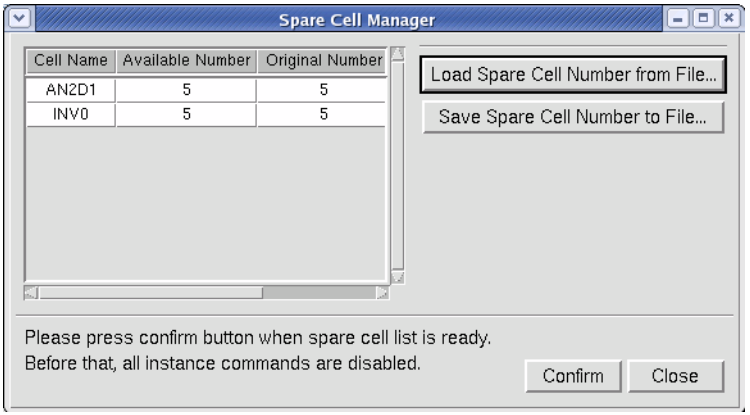


Figure: Spare Cell Manager Form

If the **Manage Spare Cells by Cell Number** option is turned *on* in the **Tools -> Preferences -> Schematics** page **-> ECO** page **-> Freeze Silicon** page, invoking the **Replace Instance** option opens the *Replace Instance* form.

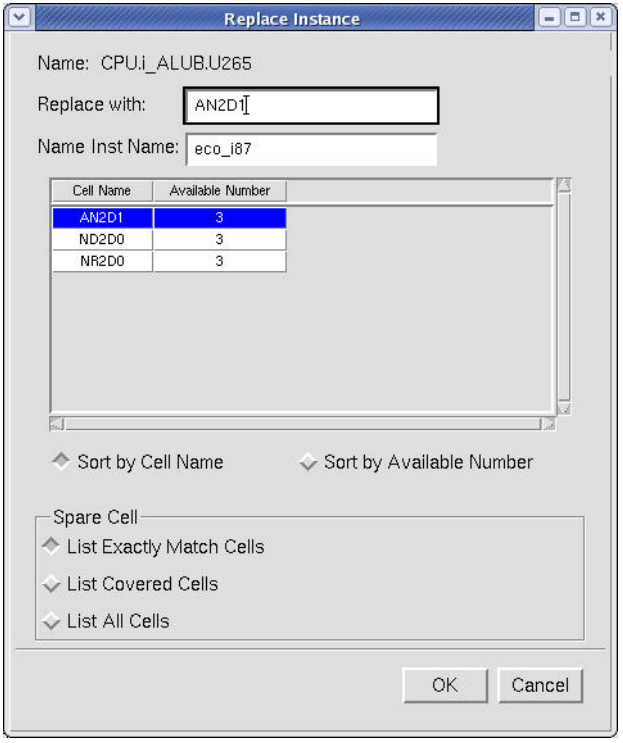


Figure: Replace Instance Form

This *Replace Instance* form is the same as the form opened by invoking the **Replace Instance by Number** option. Refer to the **Replace Instance by Number (with Manage Spare Cells by Instance Name and Spare Cell Quantity Control Enabled)** option description for details.

Replace Instance (with Manage Spare Cells by Instance Name Selected)

Menu Bar: Edit -> Replace Instance -> Replace Instance

Bind Key: R

This option is enabled after a spare cell is imported by invoking the **Spare Cell** option which opens the *Spare Cell* form.

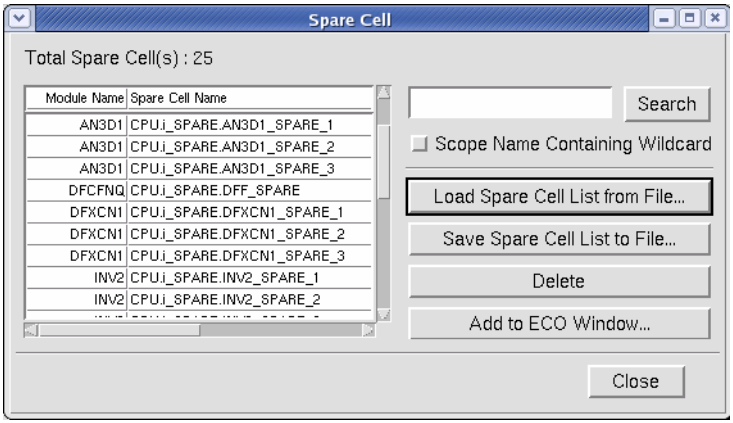


Figure: Spare Cell Form

When the **Manage Spare Cells by Instance Name** option is turned *on* and the **Spare Cell Quantity Control** option is turned *off* in the **Tools -> Preferences -> Schematics** page **-> ECO** page **-> Freeze Silicon** page, invoking the **Replace Instance** option opens the *Replace Instance* form.

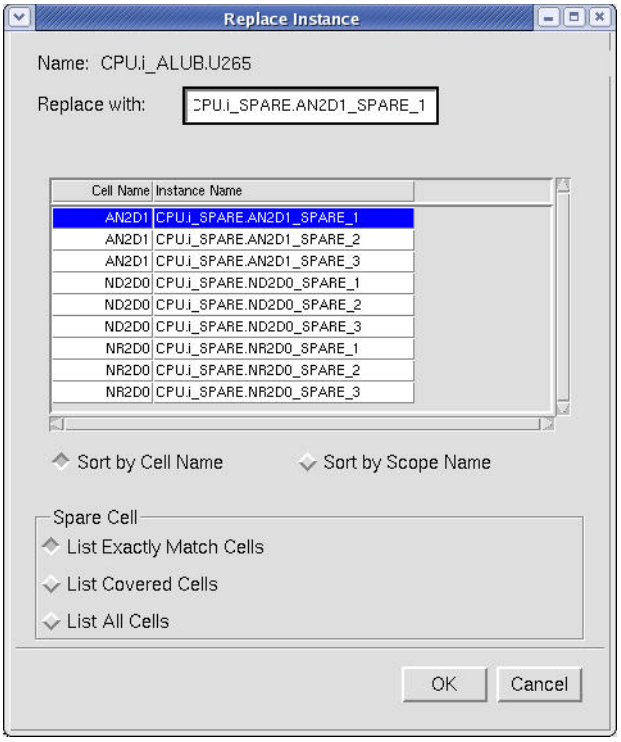


Figure: Replace Instance Form

This *Replace Instance* form is the same as the form opened by invoking the **Replace Instance by Name** option. Refer to the **Replace Instance by Name (with Manage Spare Cells by Instance Name and Spare Cell Quantity Control Enabled)** option description for details.

Replace Instance by Name (with Manage Spare Cells by Instance Name and Spare Cell Quantity Control Enabled)

Menu Bar: Edit -> Replace Instance -> Replace Instance by Name

This option is enabled after a spare cell is imported by invoking the **Spare Cell** option which opens the *Spare Cell* form.

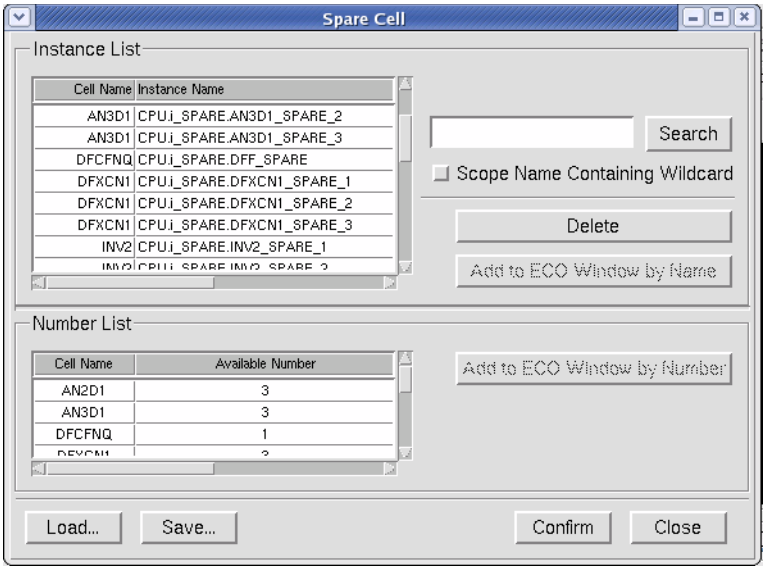


Figure: Spare Cell Form

After spare cells are imported, if the **Manage Spare Cells by Instance Name** and the **Spare Cell Quantity Control** options in the **Tools -> Preferences -> Schematics** page **-> ECO** page **-> Freeze Silicon** page are turned *on*, invoking the **Replace Instance by Name** option with an instance selected in the *nECO* window opens the *Replace Instance* form.

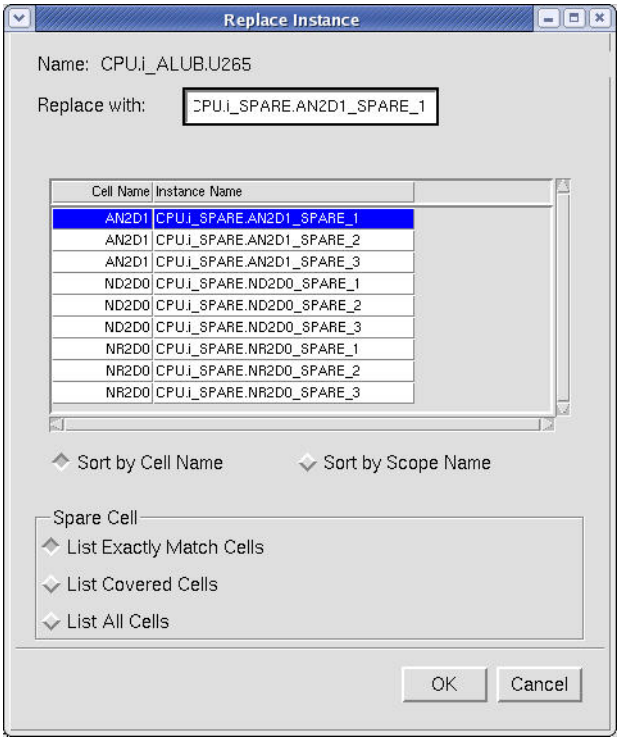


Figure: Replace Instance Form

Name: This field shows the full name of the instance that is to be replaced. The name is automatically specified based on the instance selected in the *nECO* window.

Replace with: The text field shows the cell name of the instance that is to replace the original instance. The **Cell Name** column in the table indicates the available cell types, while the **Instance Name** column indicates the available instances.

Sort by Cell Name: Select this option to sort instances by cell name.

Sort by Scope Name: Select this option to sort instances by the instance scope.

Spare Cell: Click the **List Exactly Match Cells**, **List Covered Cells**, or **List All Cells** options in this section to display different replacement cell results in the table. Only one option can be selected at a time.

- **List Exactly Match Cells:** Displays the cells that are an exact pin-to-pin match with the original instance.
- **List Covered Cells:** Displays all cells that can cover the pins of the original instance.

- **List All Cells:** Displays all cell names regardless of whether the pin count matches the original instance.

If either of the **List Covered Cells** or **List All Cells** options are selected, the correct port mapping must be specified in the *Port Mapping* form after the **OK** button is clicked.

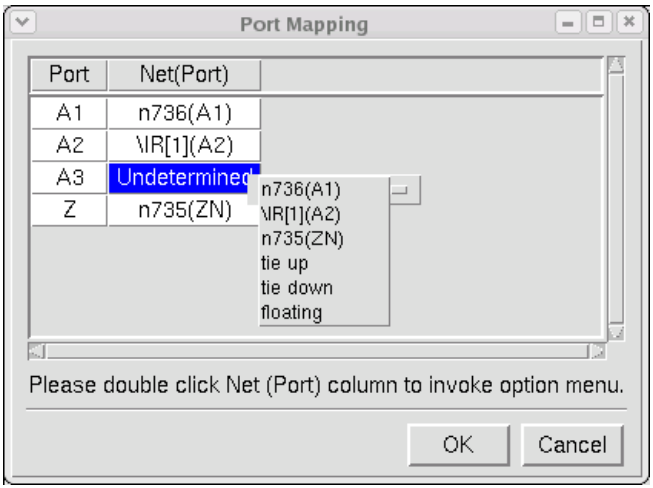


Figure: Port Mapping Form

In the **Net (Port)** column, *nECO* matches the ports using the following rules:

1. Match ports with the same name.
2. Match a port for a single input or signal output regardless of whether the name is the same or not.
3. Mark **Undetermined** for unmatched ports.

The matched result can be changed. When any cell in the **Net (Port)** column is clicked, a selection field is displayed with the available connection options where the preferred connection for each port can be selected as the figure above shows.

Replace Instance by Number (with Manage Spare Cells by Instance Name and Spare Cell Quantity Control Enabled)

Menu Bar: Edit -> Replace Instance -> Replace Instance by Number

After a spare cell is imported, if the **Manage Spare Cells by Instance Name** and the **Spare Cell Quantity Control** options in the **Tools -> Preferences -> Schematics** page **-> ECO** page **-> Freeze Silicon** page are turned *on*, invoking

the **Replace Instance by Number** option with an instance selected in the *nECO* window opens the *Replace Instance* form.

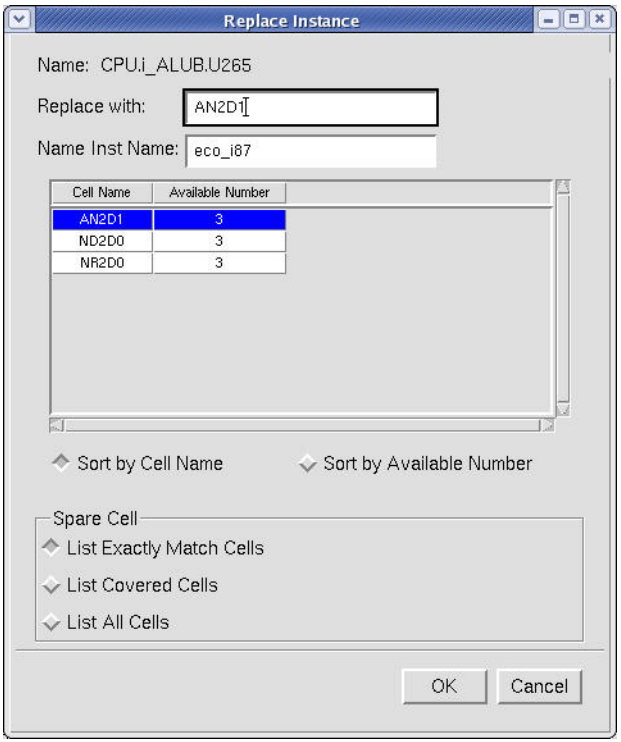


Figure: Replace Instance Form

This *Replace Instance* form is similar to the form opened by invoking the **Replace Instance by Name** option. The only difference is the instance display method. **Replace Instance by Name** shows the instance list by the instance names. **Replace Instance by Number** shows the instance list by the instance number. Refer to the **Replace Instance by Name (with Manage Spare Cells by Instance Name and Spare Cell Quantity Control Enabled)** option description for details.

Change Instance Scope

Menu Bar: Edit -> Change Instance Scope

Bind Key: S

This option opens the *Change Instance Scope* form where an instance scope can be changed in an *nECO* window.

NOTE: This option can only move the primitive instance to another scope. Prior connections are not kept.

NOTE: This option is disabled in Freeze Silicon mode.

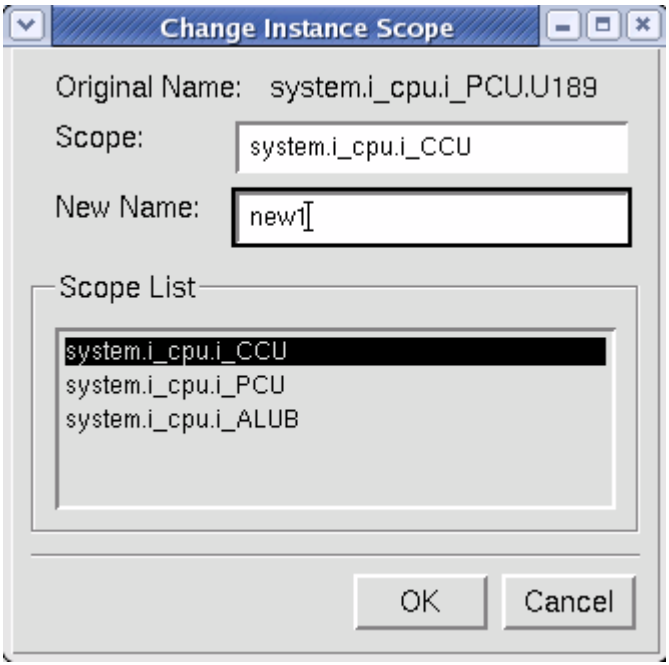


Figure: Change Instance Scope Form

The following fields and section are available:

- **Scope:** Select the scope to be changed in this text field. This field also serves as a filter for the instance list.
- **New Name:** Specify the new instance name of the scope to be changed in this text field. If a name is entered that conflicts with any name used in the *nECO* window, a warning message is shown indicating the invalid instance name.

- **Scope List:** This section displays the available scopes to be changed. The scope can be specified by double-clicking on the scope name.

Clone Module

Menu Bar: Edit -> Clone Module

Bind Key: Shift+M

This option clones an existing module to a new module. After selecting the original module and invoking this option, the *Clone Module* form opens where all instantiations of the module of the selected instance are listed.

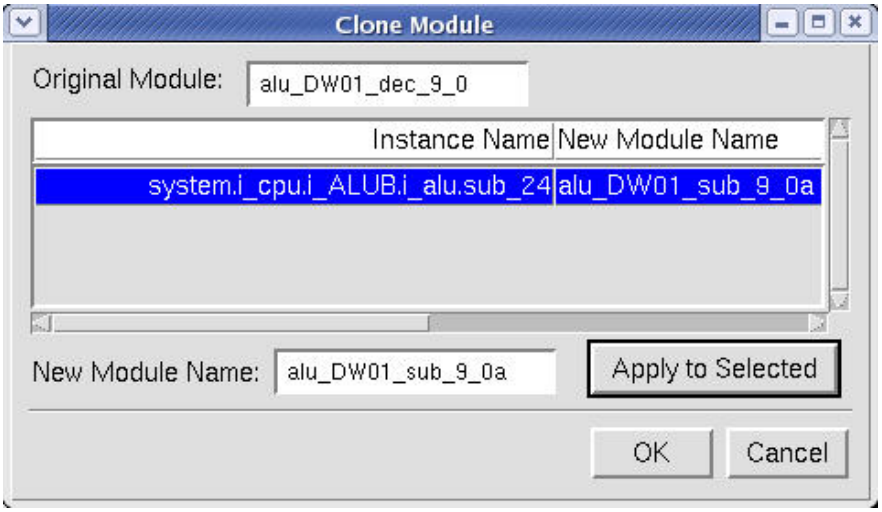


Figure: Clone Module Form

Input the new module name in the **New Module Name** text field and click **Apply to Selected** to change the name in the **New Module Name** column for the module to be cloned. Then click the **OK** button to close the form.

Spare Cell

Menu Bar: Edit -> Spare Cell

Bind Key: Shift+S

This option opens the *Spare Cell* form. Different settings in the *Preferences* form (invoked with **Tools -> Preferences**) opens different *Spare Cell* forms. The

detailed description of the three different settings are described in the following section.

Setting 1 - Opens Spare Cell Manager Form
If the **Manage Spare Cells by Cell Number** option is turned *on* in the **Tools -> Preferences -> Schematics** page **-> ECO** page **-> Freeze Silicon** page, the *Spare Cell Manager* form opens for instance manipulation through options such as **Add Buffers**, **Delete Buffer**, **Replace Instance**, **Delete Instance**, **Add Instance**, and **Change Instance Scope**. If the new instances match the cell type of the spare cells, the *Spare Cell Manager* form is opened.

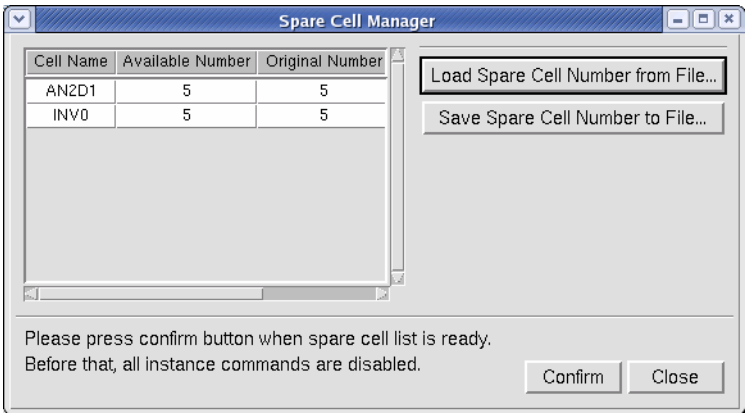


Figure: Spare Cell Manager Form

In the *Spare Cell Manager* form, the table on the left side displays the spare cell list. Three columns are available in the table. The width of each column can be adjusted by placing the cursor over the vertical bar in the heading row and then pressing and holding the left mouse button. The column headings are summarized as follows:

- **Cell Name:** Indicates the cell name.
- **Available Number:** Indicates the current available numbers. This includes the remaining numbers (from **Original Number**) and recycled numbers.
- **Original Number**: Indicates the original total number that was imported.

The **Load Spare Cell Number from File** option imports the spare cell list with the instance name from a specified file. The file format of the spare cell list is the same as the original usage that lists instance names with their full scope name. Only the instances that exist in the netlist is appended into the instance list. Note that every time the list is imported from a file, the information in the instance list is refreshed. The list is not appended to the previous list.

The **Save Spare Cell Number to File** option saves the spare cell number list into a file.

If the current spare cell number list is correct, the **Confirm** button must be clicked to confirm the spare cell number list. After the confirmation, all of the instance options are enabled for further editing.

Click the **Close** button to close the *Spare Cell Manager* form. If the **Close** button is clicked without confirming the changes, all information in the spare cell number list remains unchanged.

Setting 2 - Opens Spare Cell Form
If the **Manage Spare Cells by Instance Name** option is turned *on* and the **Spare Cell Quantity Control** option is turned *off* in the **Tools -> Preferences -> Schematics** page **-> ECO** page **-> Freeze Silicon** page, the *Spare Cell* form opens to search the desired instance name. If any instances match the specification, the *Spare Cell* form is opened.

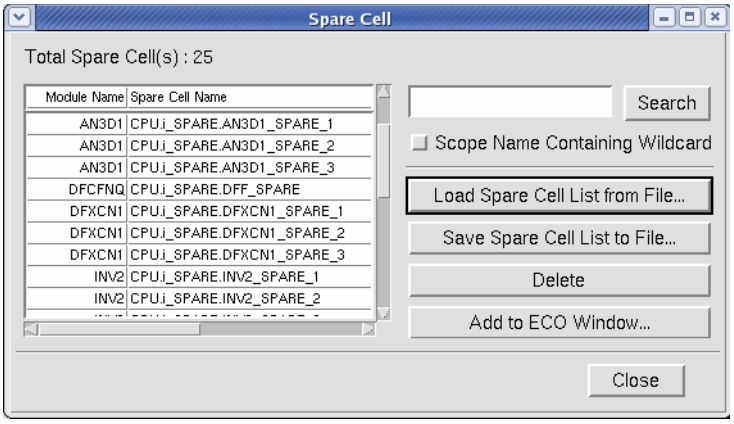


Figure: Spare Cell Form

The **Total Spare Cell** table, on the left side of the *Spare Cell* form, is generated by either the **Search** option or the **Load Spare Cell List from File** option.

NOTE: The **Search** option adds the newly searched results to the **Total Spare Cells** table without clearing the previous content in the table. This looks inconsistent to other **Search** functions in the Verdi platform. However, this is reasonable, because the table not only keeps the added search results but also the spare cells released from **Delete Instance** or **Replace Instance ECO** options.

The spare cell prefix can be entered or use the prefix set for the **Spare Cell Name Rule** field in the **Schematics** page **-> ECO** page **-> Freeze Silicon** of the

Preferences form and click the **Search** button to get the desired spare cell(s) in the netlist. When the **Scope Name Containing Wildcard** option is turned *on*, searching uses the specified wildcard characters. Otherwise, only names exactly matching the search string specification is shown. Also, clicking the **Load Spare Cell List from File** button imports the spare cell list from a specified file.

Click the **Save Spare Cell List to File** button to save a spare cell list.

Click the **Delete** button to delete spare cell(s).

The **Add to ECO Window** option adds the selected spare cell in the **Total Spare Cell** table to an *nECO* window.

After the **Add to ECO Window** button is clicked another *Spare Cell* form is opened where the spare cell can be changed.

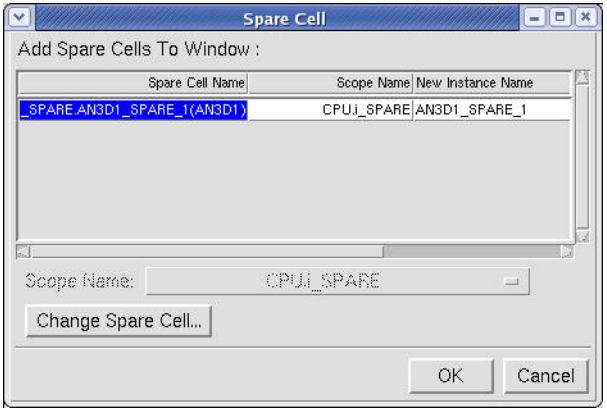


Figure: Spare Cell Form - Add to ECO Window

Select the spare cell in the **Spare Cell Name** column first and then click the **Change Spare Cell** button to open the *Change Spare Cell* form where the spare cell to change can be selected. Click the **OK** button to confirm the change.

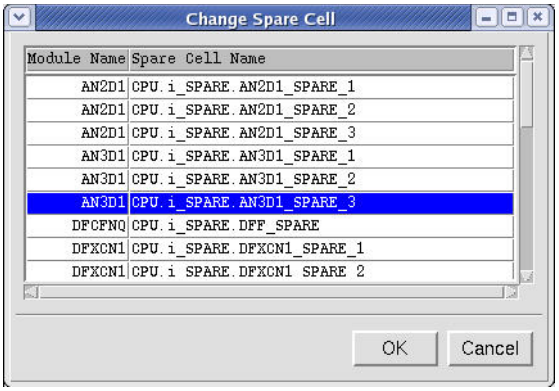


Figure: Change Spare Cell Form

In the *Spare Cell* form, highlight the **Scope Name** column, and the **Scope Name** selection field is enabled for specifying the desired scope. The **Change Spare Cell** button is disabled. The new instance name can be modified in the **New Instance Name** column.

Consumed Spare Cells Form

After the **OK** button is clicked in the *Spare Cell* form, if the spare cell to be replaced has multiple instances, then when one instance is modified, other instances must be modified appropriately to map all the changes belonging to a spare cell. The *Consumed Spare Cells* form opens and lists all used spare cells to map the changes for every instance which belongs to its modified cell. The spare cells that is consumed by this action can be specified. ECO provides a default consumption list by sequentially assigning available cells to each of the module instantiations, where the order of available cells is sorted by cell name. After clicking **OK** on the *Consumed Spare Cells* form, if any spare cell names are duplicated, a warning message is shown. ECO does not add instances into each module until the consuming spare cell list is correct.

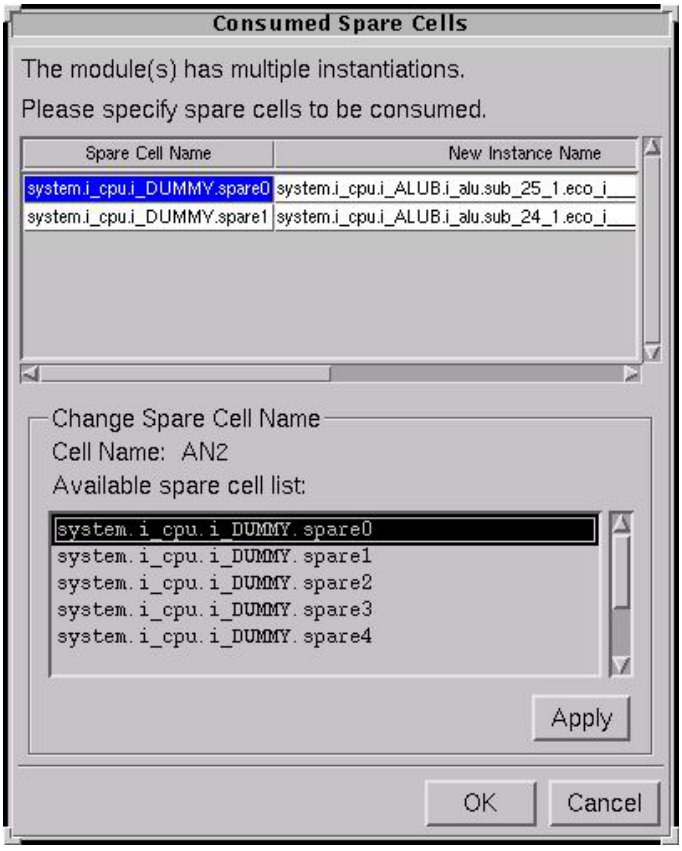


Figure: Consumed Spare Cells Form

In the **Spare Cell Name** column, when any of the spare cell names are selected, the **Cell Name** and **Available Spare Cell List** in the **Change Spare Cell Name** section is updated with the spare cell name that was selected. The **Available Spare Cell List** displays all of the available instance names of the specified cell type. The spare cell name can be modified by highlighting the spare cell name and changing it with one of the other available spare cells. The new spare cell that is selected in the **Available Spare Cell List** is applied to the field by clicking the **Apply** button or double-clicking the spare cell.

The **New Instance Name** column cannot be edited. The format of this field is *New_Instance_ Name(Module_Name)* where the new instance name is followed by the module name in brackets.

In the **Change Spare Cell Name** section, the **Cell Name** field shows the cell type of the spare cell. The **Available Spare Cell List** field displays all of the spare cells available to choose from.

Setting 3 - Opens Spare Cell Form with Instance/Number List
If the **Manage Spare Cells by Instance Name** and the **Spare Cell Quantity Control** options are turned *on* in the **Schematics -> ECO -> Freeze Silicon** page of the *Preferences* form, a *Spare Cell* form is opened when the **Edit -> Spare Cell** option is invoked.

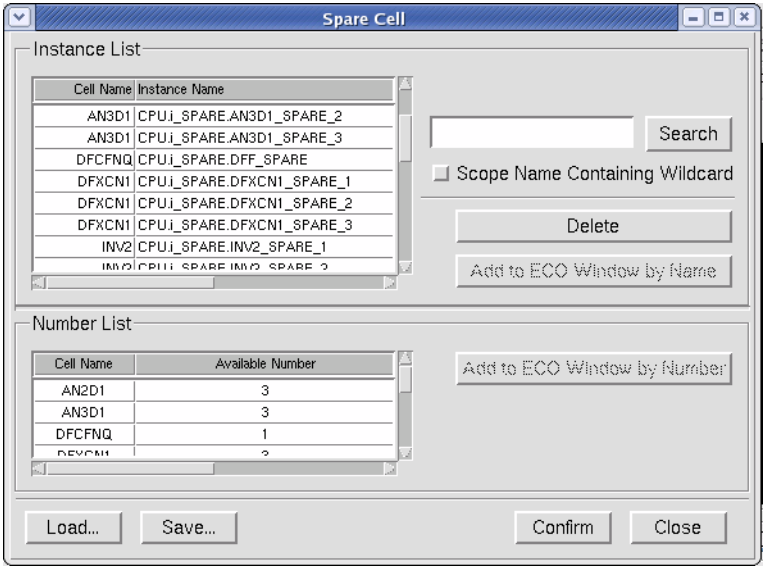


Figure: Spare Cell Form with Number List

After loading a spare cell list, the *Spare Cell* form displays information in the **Instance List** section and the **Number List** section.

In the **Instance List** section, the **Instance List** table displays the cell name and instance name information.

The **Search** button can be used to search for spare cells in the netlist. When the **Scope Name Containing Wildcard** option is turned *on*, searching uses any specified wildcard characters. Otherwise, only names exactly matching the search string specification is shown.

Click the **Delete** button to edit the instance list manually. When any instances are deleted from the **Instance List**, the number in the **Available Number** column of the **Number List** is adjusted appropriately.

To enable the **Add to ECO Window by Name** option, the **Confirm** button must be clicked first. The spare cell selected in the **Instance List** is added to the *nECO* window.

In the **Number List** section, the **Number List** table displays the cell name and the total number of cells.

To enable the **Add to ECO Window by Number** option, the **Confirm** button must be clicked first. After clicking the **Add to ECO Window by Number** button, the *Add Instance* form is opened. Select a desired spare cell from the **Cell List** section. After clicking the **Apply** button, the specified spare cell is added to the scope specified on the **Destination Scope** in the *nECO* window. The cell number for the selected spare cell decreases by one each time the **Apply** button is clicked. Refer to the **Add Instance by Number (with Manage Spare Cells by Instance Name and Spare Cell Quantity Control Enabled)** option description for details.

Click the **Load** button to import the spare cell list from a file. In the file, the spare cell list includes the instance names with scope names. See the following example.

When the **Spare Cell Quantity Control** option is turned *on* in Freeze Silicon mode, the output spare-list file starts with the leading token "@@nECO". If the cell quantity is known, a token "CELL" is provided to specify the cell quantity: CELL cell-name cell-quantity.

A typical example of the spare-cell list is shown below:

```
@@nECO
@@nECOCELL AN2 4
@@nECOCELL IV 2
@@nECOCELL OR2 2
CPU.i_spare.spare1
CPU.i_spare.spare2
CPU.i_spare.spare3
CPU.i_spare.spare4
CPU.i_spare.spare5
CPU.i_spare.spare6
CPU.i_spare.spare7
CPU.i_spare.spare8
```

Click the **Save** button to save the results as a file that includes the cell name and cell quantity format.

Until the **Confirm** button is clicked, instance modification options, such as **Add Buffers, Delete Buffer, Replace Instance, Delete Instance,** and **Add Instance**, do not work. These options are enabled only after the **Confirm** button is clicked.

Click the **Close** button to close the *Spare Cell* form.

Count Spare Cell

Menu Bar: Edit -> Count Spare Cell

If new instances are created in the netlist, these instances must not exceed the number of available spare cells in the netlist. This option prepares the spare cell number list for counting spare cells. Search by string or import the spare cell list from file. When the number list is ready, click the **Save Spare Cell Number to File** button to save the file.

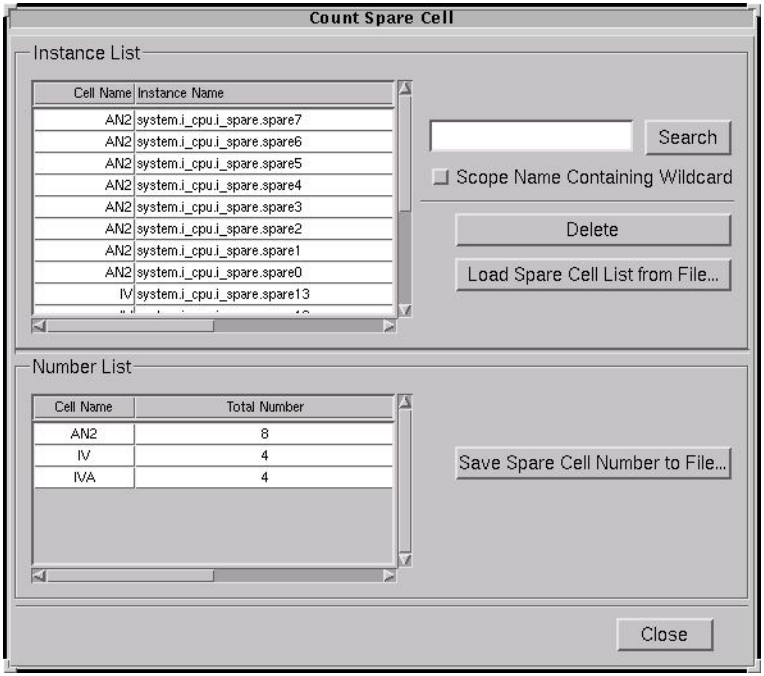


Figure: Count Spare Cell Form

Instance List: Similar to the *Spare Cell* form, type the spare cell prefix or use the prefix that was set in the **Spare Cell Name Rule** field on the *Preferences* and click **Search**. The instance list with the spare cell prefix is displayed.

Scope Name Containing Wildcard: When this option is turned *on*, all of the spare cells that contain the specified prefix is displayed.

Delete: Click this button to remove the unwanted instances from the list.

Load Spare Cell List from File: Click this button to import the spare cell list with the instance name from a file. The file format of the spare cell list is the same as the original usage that lists the instance name with the full scope name. Only

the instances that exist in the netlist is appended into the instance list. With each import, all information from the instance list is refreshed. The information is not appended into the last list.

Number List: Lists the summary of the instance list.

Save Spare Cell Number to File: Click this button to save the information of the spare cell number list into a file.

View Options

Preselect

Menu Bar: View -> Preselect

This toggle option highlights signals or blocks when the cursor moves over them in the *nECO* window. The message bar at the bottom of the main framework window also shows the name of the selected signal or block regardless of whether the **Preselect** toggle option is *on* or *off*. The default is *on*.

Port Name

Menu Bar: View -> Port Name

This toggle option turns the display of the port name *on* or *off*.

Instance Name

Menu Bar: View -> Instance Name

This toggle option turns the display of the instance name *on* or *off*.

Local Net Name

Menu Bar: View -> Local Net Name

This toggle option turns the display of the local net name *on* or *off*.

Parameter List

Menu Bar: View -> Parameter List

When this toggle option is *on*, the parameter list is displayed on the module block.

NOTE: The parameter list must be defined in the design file.

High Contrast

Menu Bar: View -> High Contrast

When this toggle option is *on*, the rest of the objects in the schematic window is darker to make the selected instance or module easier to view.

Short Name

Menu Bar: View -> Short Name

When this toggle option is *on*, the full hierarchical names for instance names or local net names are shown if both the **Instance Name** or **Local Net Name** options are turned *on*.

Fan In/Out Number

Menu Bar: View -> Fan In/Out Number

When this toggle option is *on*, the Fan In/Out number is shown on the input/output pins of the instances.

Zoom

Zoom In

Menu Bar: View -> Zoom -> Zoom In

Toolbar Icon: 

Bind Key: Shift+Z

This option provides a close-up view of the content in the *nECO* window. The magnification of the viewing area is changed to half the magnification of the previous view.

NOTE: A specific area can be zoomed by dragging-left to form a rectangle as the zoomed area.

Zoom Out

Menu Bar: View -> Zoom -> Zoom Out

Toolbar Icon: 

Bind Key: Z

This option enables more of the content to be seen in the *nECO* window at a reduced size. The magnification of the viewing area is changed to two times the magnification of the previous view from the center point in both the horizontal and vertical directions.

Zoom All

Menu Bar: View -> Zoom -> Zoom All

Toolbar Icon: 

Bind Key: F

This option shows all contents of the schematic.

Fit Select Set

Menu Bar: View -> Zoom -> Fit Select Set

Bind Key: Ctrl+F

This option fits the selected objects in the schematic window.

Pan

Toolbar Icon: 

Click this toolbar icon to pan the *nEco* pane to the left, right, up, or down.

Pan Left

Menu Bar: View -> Pan -> Pan Left

Bind Key: Left Arrow

This option pans the schematic window to the left.

Pan Right

Menu Bar: View -> Pan -> Pan Right

Bind Key: Right Arrow

This option pans the schematic window to the right.

NOTE: The horizontal scroll bar in the schematic window can be moved to pan left and right.

Pan Up

Menu Bar: View ->Pan -> Pan Up

Bind Key: Up Arrow

This option pans the schematic window up.

Pan Down

Menu Bar: View -> Pan -> Pan Down

Bind Key: Down Arrow

This option pans the schematic window down.

NOTE: The vertical scroll bar in the schematic window can be moved to pan up and down.

Last View

Menu Bar: View -> Last View

Toolbar Icon: 

Bind Key: l (lowercase L)

This option returns to the schematic view of the last invoked viewing option. The **Last View** option only shows the last view. When this option is invoked more than once, it switches between the current and last views.

Schematic Options

Find in Current Scope

Menu Bar: Schematic -> Find in Current Scope

Bind Key: A

This option opens the *Find* form that lists all signals, instances, instance ports (instport), ports, or modules in the *nECO* window. Select from one of the **Signal**, **Instance**, **Instport**, **Port** or **Module** options to filter uninterested objects. Use the **Find** text field to find the desired signal, instance, instance port, port, or module.

The selected signal, instance, instance port, port, or module in the *Find* form are highlighted in the *nECO* window.

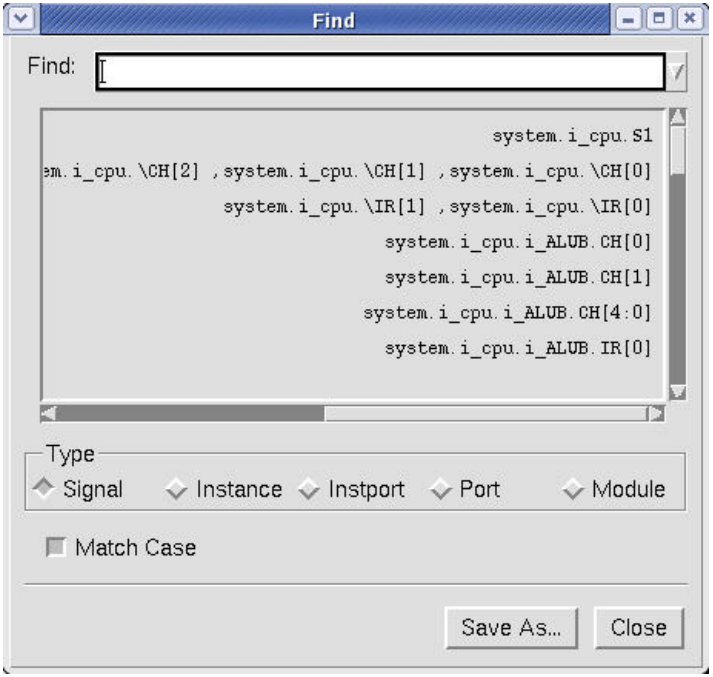


Figure: Find Form

Find: Type the instance or signal to be found in this text field. This text field also serves as a filter for the signal/instance list described below.

Auto Fit Found Object(s)

Menu Bar: Schematic -> Auto Fit Found Object(s)

This option fits the selected object automatically in the *nECO* window. This option applies to the drag-and-drop, **Schematic -> Find**, and **Tools -> Options -> Sync. Signal Selection** options.

Selection

Select

Menu Bar: Schematic -> Selection -> Select

This option opens the *Select* form where the instance(s) and/or signal(s) in the schematic window can be selected by clicking the **Instance**, **Signal**, or **Floating Pin** buttons.

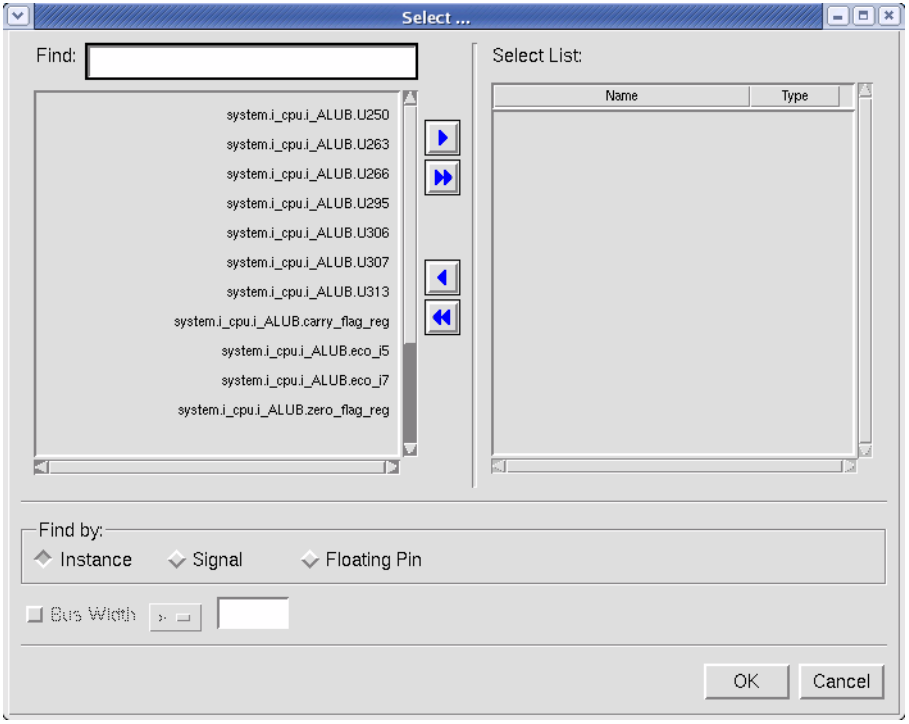


Figure: Select Form for Instances/Signals

This form lists all of the instances or signals under the current module in alphabetical order. Type the instance or signal string to be located in the **Find** text field. After selecting the instances/signals of interest, use the **Add Object** or **Add All** icons to add the instances/signals to the select list. Alternatively, unwanted instances/signals can be removed by clicking either the **Remove Object** or **Remove All** icons.

Below the **Find by** section, the option to set the criteria associated with the **Bus Width** is available. A conditional operator of greater than (>), greater than or equal to (>=), equal to (=), less than or equal to (<=), or less than (<) can be selected. This conditional operator is combined with the number of bits specified. When the **Enter** key is pressed, the buses meeting the criteria is added into the select list.

Select Common Pin

Menu Bar: Schematic -> Selection -> Select Common Pin

This option selects the same pin for multiple selected instances.

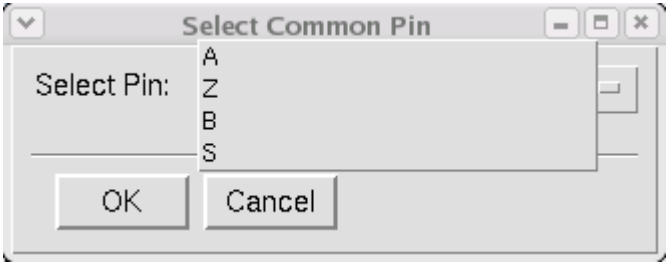


Figure: Select Common Pin

The **Select Pin** field displays the common pin for the selected set of instances. Select the desired pin and click **OK** to complete the selection.

Select Pin By Net

Menu Bar: Schematic -> Selection -> Select Pin By Net

This option selects the common instance pin after selecting a net and connected instances.

Select All Objects

Menu Bar: Schematic -> Selection -> Select All Objects

This option selects all objects in the *nECO* window.

Select All Instances

Menu Bar: Schematic -> Selection -> Select All Instances

This option selects all instances in the *nECO* window.

Select All Signals

Menu Bar: Schematic -> Selection -> Select All Signals

This option selects all signals in the *nECO* window.

Deselect All

Menu Bar: Schematic -> Selection -> Deselect All

This option deselects all objects in the *nECO* window.

Focus Connection

Menu Bar: Schematic -> Focus Connection

Mouse Action: Double-click the signal in the schematic window

This option highlights the connection on the same hierarchy of the selected signal. Use *n* or *N* to browse the connecting instance.

Add Selected to Waveform

Menu Bar: Schematic -> Add Selected to Waveform

Bind Key: Ctrl+W

This option adds the selected signal(s) in the *nECO* window to the cursor bar position of the signal pane in *nWave*. Alternatively, the selected signals can be dragged from the *nECO* window to the *nWave* window.

Change Color

Menu Bar: Schematic -> Change Color

Bind Key: c

This option opens the *Change Selection Color* form where the color of the selected signal can be changed.

Figure: Change Selection Color Form

This form shows the selected signal/instance with its current color in the color plate. Change the signal's color instantly by clicking a color on the color plate.

Set the selected signal color as the default by clicking **Default**. When the signals are dropped to the *nWave* window, they inherit the new color.

NOTE: A quick way to change a signal's color within a predefined color set is to select a signal first, then continue pressing 't' to cycle through the color set until the desired color is reached.

All Objects to Default Color

Menu Bar: Schematic -> All Objects to Default Color

This option changes all the signals to their default colors.

Active Annotation

Menu Bar: Schematic -> Active Annotation

NOTE: This option is available after a simulation results file is loaded from either *nTrace* or *nWave*.

This toggle option enabled/disables active annotation. When this option is *on*, each signal's value appears on the schematic and updates as the current simulation time changes.

Annotate in Color

Menu Bar: Schematic -> Annotate in Color

This toggle option enables/disables the annotation style and line coloring.

The definitions of the value types are as follows:

1: logic high
0: logic low
z: high impedance
x: unknown

To change the color or the line style for annotations, invoke the **Tools -> Preferences** option in the *nTrace* window to display the *Preferences* form and select the **Schematics** page, then the **Color** page. In the **Type** list, a list of objects

can be found whose color and line style can be set, including the annotation line coloring for value 1, value 0, value x, and value z.

Leading Zeros

Menu Bar: Schematic -> Leading Zeros

Refer to the **Schematic** -> **Leading Zeros** option in the *nSchema* window for details.

SDF Annotation

Menu Bar: Schematic -> SDF Annotation

After loading the SDF file from the **File -> Load SDF Files** option in *nTrace*, this toggle option turns the SDF annotation *on* and *off* in the *nECO* window. It is enabled when any INTERCONNECT delays exist in the loaded SDF file.

To see the cell delay (IOPath delay), select a cell and invoke the **Show Cell Delay** option from the right-click option menu. The *nECO* window opens a message form that lists all IOPath_delays in this cell.

Delay Type

Menu Bar: Schematic -> Delay Type

Specify the delay type of the SDF file by selecting from **Minimum**, **Typical**, or **Maximum**. The selected type is shown in the menu option. The default setting is **Typical**.

Delay Precision

Menu Bar: Schematic -> Delay Precision

Specify the delay precision of the SDF file by selecting from **0.1**, **0.01**, and **0.001**. The selected precision is displayed in the menu option. The default setting is 0.01.

Options

Hide Extraneous Bus

Menu Bar: Tools -> Options -> Hide Extraneous Bus

This toggle option controls the bus place-and-route style. When this option is *on*, all bus members are treated as individual signals and they are not displayed into a bus. If a bus is added into the schematic window while the **Hide Extraneous Bus** option is *on*, the *nECO* window highlights all the nets that belong to the bus. *nECO* always keeps the selected set whenever **Hide Extraneous Bus** is turned *on* or *off*.

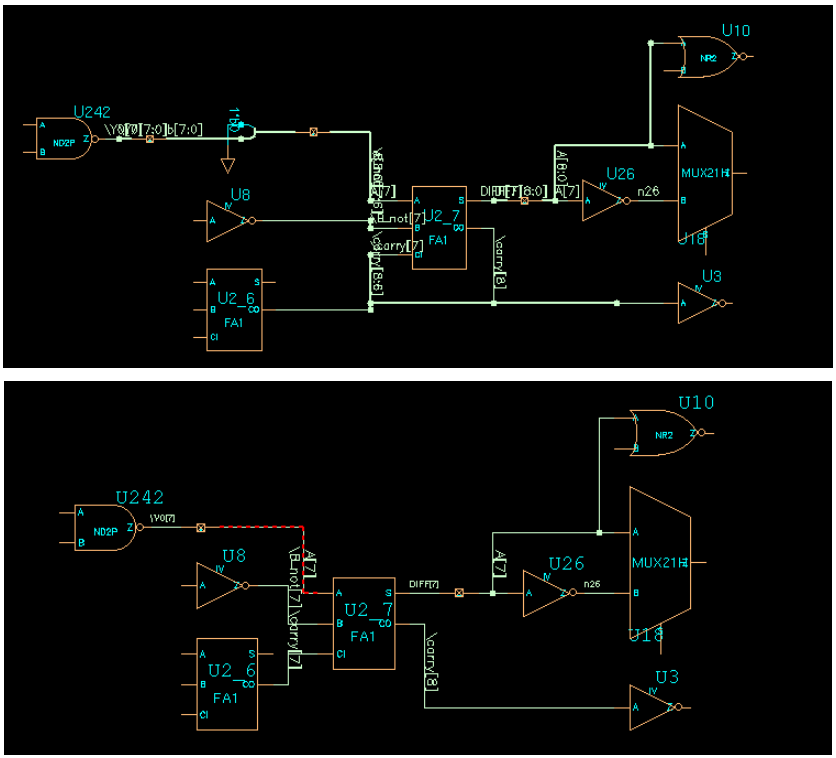


Figure: Before and After Turning Hide Extraneous Bus On

In the figures above, the split bus is shown in red after the **Hide Extraneous Bus** option is turned *on*.

Keep Placement

Menu Bar: Tools -> Options -> Keep Placement

Toolbar Icon: 

This toggle option controls cell placement during editing. When this option is turned *on*, all the existing cells are fixed in their current location for any editing action made to this window. When this option is turned *off*, *nECO* rearranges the cell location according to the current connectivity.

Trace Options

Driver

Menu Bar: Trace -> Driver

This option traces all the possible drivers of a selected signal and shows the results on the current *nECO* window.

Load

Menu Bar: Trace -> Load

This option traces all the possible loads of a selected signal and shows the results on the current *nECO w*indow.

Connectivity

Menu Bar: Trace -> Connectivity

This option traces all the possible loads and drivers of a selected signal and shows the results on the current *nECO* window. This option is the combination of the **Trace Driver** and **Trace Load** options.

Tools Options

New Schematic

Browser Window

Menu Bar: Tools -> New Schematic -> Browser Window

This option creates a partial hierarchical view in *nSchema* for the selected signals or instances in the *nECO* window. The port is displayed by default to switch to the upper hierarchy.

Flattened Window

Menu Bar: Tools -> New Schematic -> Flattened Window

This option creates a zoomed in view in *nSchema* for the selected primitive instances in the *nECO* window.

Fan-in Cone

Menu Bar: Tools -> New Schematic -> Fan-in Cone

This option creates a new partial flattened schematic window based on the result of tracing the fan-in cone of the selected signal in the *nECO* window.

Fan-out Cone

Menu Bar: Tools -> New Schematic -> Fan-out Cone

This option creates a new partial flattened schematic window based on the result of tracing the fan-out cone of the selected signal in the *nECO* window.

Driver

Menu Bar: Tools -> New Schematic -> Driver

This option creates a new partial flattened schematic window based on the result of tracing the drivers of the selected signal or instance in the *nECO* window.

Load

Menu Bar: Tools -> New Schematic -> Load

This option creates a new partial flattened schematic window based on the result of tracing the loads of the selected signal or instance in the *nECO* window.

Connectivity

Menu Bar: Tools -> New Schematic -> Connectivity

This option creates a new partial flattened schematic window based on the result of tracing the connectivity of the selected signal or instance in the *nECO* window.

Preferences

Menu Bar: Tools -> Preferences

Refer to the *ECO Folder - nECO* section in the *Preferences* chapter for details on setting preferences for the *nECO* window.

Customize Menu/Toolbar

Refer to the **Tools -> Customize Menu/Toolbar** option in the *nTrace* chapter for details.

Right-click Options

Many of the previously described options can also be selected from the right-click option menu in *nECO*.

nECO Frame Right-click Options

After the right mouse button is clicked anywhere in the menu, toolbar icon, or banner area of the *nECO* window or standalone window, a configuration option menu is displayed. This menu can be used to configure the available icons.

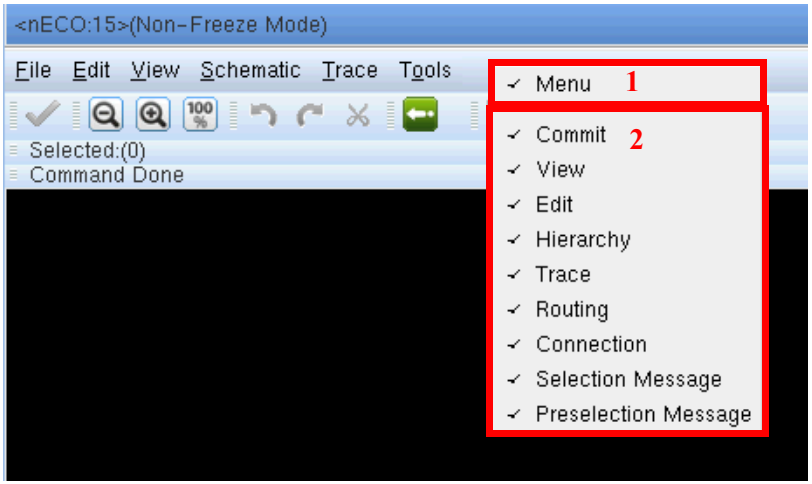


Figure: Configuration Option Menu

The **Menu** option (labeled 1 in the figure above) enables/disables the option menu bar. The options in the bottom section (labeled 2 in the figure above) enable/disable the toolbar icons for different functions.

Main Window Right-click Options

The following is a list of right-click menu options available for *nECO*. Some selections are available from the right mouse button menu. Other selections have equivalent selections available from the toolbar menus.

Drag

Bind Key: Ctrl+C

The selected signal can be dragged to another location and dropped. This is similar to using the middle mouse button to select and drag.

Drop

Bind Key: Ctrl+V

The selected signal dragged from another location can be dropped in this window. This is similar to releasing the middle mouse button for a drop.

ECO Change Status

Refer to the **File -> ECO Change Status** option description for details.

Rename Net

Refer to the **Edit -> Rename Net** option description for details.

Rename Instance

Refer to the **Edit -> Rename Instance** option description for details.

Connect to Net

Refer to the **Edit -> Connect to Net** option description for details.

Edit Concate Net

Refer to the **Edit -> Edit Concate Net** option description for details.

Copy Instance

Refer to the **Edit -> Copy Instance** option description for details.

Paste Instance

Refer to the **Edit -> Paste Instance** option description for details.

Remove from Viewing Objects

Refer to the **Edit -> Remove from Viewing Objects** option description for details.

Delete Connection

Refer to the **Edit -> Delete Connection** option description for details.

Make Connection

Refer to the **Edit -> Make Connection** option description for details.

Tie Up

Refer to the **Edit -> Tie Up** option description for details.

Tie Down

Refer to the **Edit -> Tie Down** option description for details.

Create Port

Refer to the **Edit -> Create Port** option description for details.

Add Buffers

Refer to the **Edit** -> **Add Buffers (Non-Freeze Silicon Mode)** and **Add Buffers (Freeze Silicon Mode)** option descriptions for details.

Delete Buffer

Refer to the **Edit -> Delete Buffer** option description for details.

Add Instance

Refer to the **Edit** -> **Add Instance (Non-Freeze Silicon Mode)** and **Add Instance (Freeze Silicon Mode)** option descriptions for details.

Delete Instance

Refer to the **Edit -> Delete Instance** option description for details.

Replace Instance

Refer to the **Edit** -> **Replace Instance (Non-Freeze Silicon Mode)** and **Replace Instance (Freeze Silicon Mode)** option descriptions for details.

Change Instance Scope

Refer to the **Edit -> Change Instance Scope** option description for details.

Clone Module

Refer to the **Edit -> Clone Module** option description for details.

Spare Cell

Refer to the **Edit -> Spare Cell** option description for details.

Toolbar Icons and Fields

Figure: Toolbar Used in the nECO Window

The available toolbar icons may be modified. Refer to the *Toolbars* section of the *User Interface* chapter in the *Verdi User Guide and Tutorial* manual for details.

The different toolbar categories and available icons are described below.

Commit Category

Commit Change

Refer to the **File -> Commit Change** option description for details.

View Category

Zoom Out

Refer to the **View -> Zoom -> Zoom Out** option description for details.

Zoom In

Refer to the **View -> Zoom -> Zoom In** option description for details.

Zoom All

Refer to the **View -> Zoom -> Zoom All** option description for details.

Pan

Click this toolbar icon to pan the schematic window to the left, right, up, or down. Refer the **View -> Pan** command description for more details.

Edit Category

Undo

Refer to the **Edit -> Undo** option description for details.

Redo 

Refer to the **Edit -> Redo** option description for details.

Remove from Viewing Objects 

Refer to the **Edit -> Remove from Viewing Objects** option description for details.

Hierarchy Category

Last View 

Refer to the **Edit -> Last View** option description for details.

Trace Category

Trace Driver 

Refer to the **Trace -> Driver** option description for details.

Trace Load 

Refer to the **Trace -> Load** option description for details.

Routing Category

Keep Placement 

Refer to the **Tools -> Options -> Keep Placement** option description for details.

Connection Category

Make Connection 

Refer to the **Edit -> Make Connection** option description for details.

Delete Connection 

Refer to the **Edit -> Delete Connection** option description for details.

Selection Message Category

Selection Message

This field displays information about the selected object.

Preselection Message Category

Preselection Message

When the cursor is moved over an object, this field displays information about the preselected object.

Register Window

Overview

The *Register Window* frame is displayed when the **Tools -> Register** command is invoked from *nTrace* or *nWave*. The *Register Window* frame is docked to the same frame location as the source code frame as a new tab. The *Register Window* frame can become a standalone window by clicking the **Undock** icon on the toolbar.

Signals that are dragged from the *nTrace*, *nSchema*, or *nWave* windows can be placed anywhere on the screen and then edits can be made. The results can be saved to a register file.

The menu bar for the *Register Window* frame is as follows

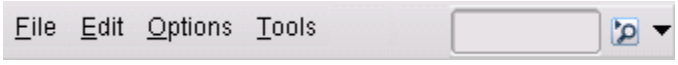


Figure: Register Window Menu Bar

NOTE: In the *Register Window*, the text boxes do not support global font.

For details on the menu bar icons, see the *Icons for Dockable Panes* section.

This chapter consists of the following sections:

Menu Summary

All *Register Window* menu commands can be double-clicked to jump to the corresponding command description. For right-click commands, see the *Right-Click Commands* section.

The *Register Window* menu commands are summarized below:

File Commands

Open
Save
Reload
Print
Close

Edit Commands

Lock/Unlock
Undo
Redo
Cut
Copy
Paste
Orientation
Order
Align
Select All
Unselect All
Fit All

Options Commands

Hierarchical Name
Leading Zeros
Show Transition
Auto Expand
Value Space

Tools Commands

Preferences
Customize Menu/Toolbar

Bind Keys

For a complete list of bind keys used in the *Register Window*, see the *nRegister* section in the *Bind Key Summary* for details.

File Commands

Open

Menu Bar: File -> Open

This command opens the *Open Register File* form where you can select and load the desired register file.

Save

Menu Bar: File -> Save

This command opens the *Save Register* form where the edited contents can be saved to a register file (**.register*). The alias information is also saved.

Reload

Menu Bar: File -> Reload

This command reloads the previously opened register file. The alias information is restored.

Print

Menu Bar: File -> Print

This command opens the *Print* form where you can print the file with the current signal values.

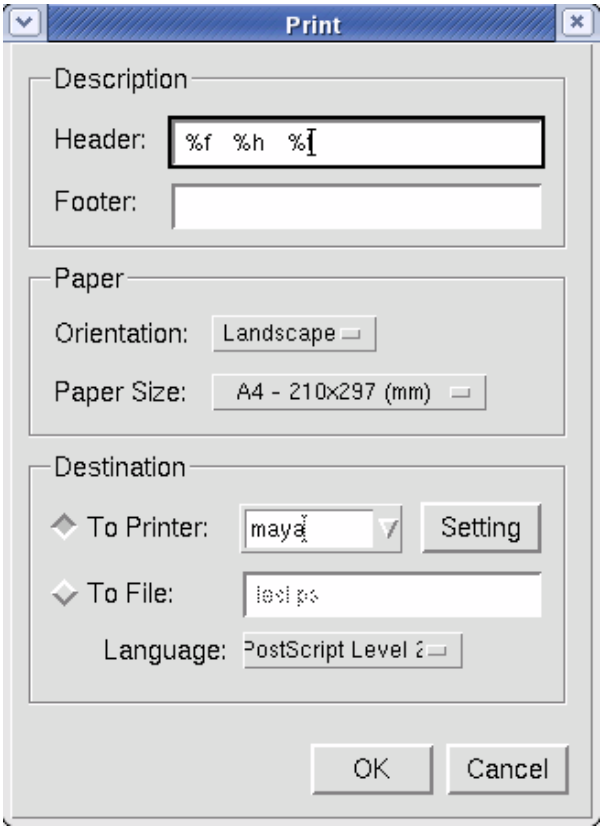


Figure: The Print Form

The **Description** section includes the following options:

- **Header:** Specifies the information to be presented on the top of the printout. By default, the header includes the file name (%f) that is currently being viewed in the *nWave* window, the user name and host name (%h), and the date and time (%t).
- **Footer:** Specifies the information to be presented on the bottom of the printout.

The **Paper** section includes the following options:

- **Orientation:** Specifies the orientation for the printed page by selecting either **Landscape** or **Portrait**.
- **Paper Size:** Selects the paper size of the printed page from the supplied list.

The **Destination** section includes the following options:

- **To Printer:** When this option is selected, the file gets printed on the printer specified in the text field.
- **To File:** This command saves the print details to a postscript file in the working directory. The file name must be specified in the text field.

Close

Menu Bar: File -> Close

This command closes the *Register Window*.

Edit Commands

Lock

Menu Bar: Edit -> Lock

Toolbar Icons: 

When this command is activated, all editing functions in the *Register Window* are frozen to prevent accidental changes.

NOTE: This is a switch command. After the **Edit -> Lock** command is activated, the next time the **Edit** pull-down menu is accessed, this command is replaced with its **Edit -> Unlock** counterpart. For details on the **Edit -> Unlock** command, see *Unlock*.

Unlock

Menu Bar: Edit -> Unlock

Toolbar Icons: 

When this command is activated, all editing functions in the *Register Window* are unlocked for additional edits.

NOTE: This is a switch command. After the **Edit -> Unlock** command is activated, the next time the **Edit** pull-down menu is accessed, this command is replaced with its **Edit -> Lock** counterpart. For details on the **Edit -> Lock** command, see *Lock*.

Undo

Menu Bar: Edit -> Undo

Toolbar Icon: 

This command reverses the last action performed, if possible. To undo an action, invoke the **Edit -> Redo** command.

Redo

Menu Bar: Edit -> Redo

This command repeats the last action performed, if possible.

Cut

Menu Bar: Edit -> Cut

Toolbar Icon: 

Bind Key: Delete

This command removes the selected object and places a copy on the clipboard.

Copy

Menu Bar: Edit -> Copy

Bind Key: Ctrl+C

This command places a copy of the current selection on the clipboard.

Paste

Menu Bar: Edit -> Paste

Bind Key: Ctrl+V

This command places the information cut or copied to the clipboard into the *Register Window* at the current cursor position.

Orientation

Menu Bar: Edit -> Orientation

This command rotates the display of the selected text or signals to one of the following directions: **Normal**, **Bottom-Up**, **Upside Down**, or **Top-Down**.

Order

Menu Bar: Edit -> Order

This command changes the order of the selected objects to one of the following: **Bring to Front**, **Send to Back**, **Bring Forward**, or **Send Backward**. Press and hold the Ctrl key while left-clicking to select multiple objects.

Align

Menu Bar: Edit -> Align

This command aligns the display of the selected objects to one of the following: **Align Left**, **Align Right**, **Align Top**, or **Align Bottom**. The active object (highlighted in purple) is the source for the alignment. Press and hold the Ctrl key while left-clicking to select multiple objects.

Select All

Menu Bar: Edit -> Select All

This command selects all objects in the *Register Window*.

Unselect All

Menu Bar: Edit -> Unselect All

This command deselects all objects in the *Register Window*.

Fit All

Menu Bar: Edit -> Fit All

This command fits all objects in the *Register Window*.

Options Commands

Hierarchical Name

Menu Bar: Options -> Hierarchical Name

This toggle command enables/disables the display of a signal with/without a full hierarchical name.

Leading Zeros

Menu Bar: Options -> Leading Zeros

This toggle command enables/disables the display of leading zeros for signal values.

Show Transition

Menu Bar: Options -> Show Transition

This toggle command enables/disables the display of the value transition for a signal.

Auto Expand

Menu Bar: Options -> Auto Expand

This toggle command enables/disables the display of auto expand for a signal.

Value Space

Menu Bar: Options -> Value Space

This command provides an option to insert a character in the values posted to the *Register Window* via the *Configure Value Space* form.

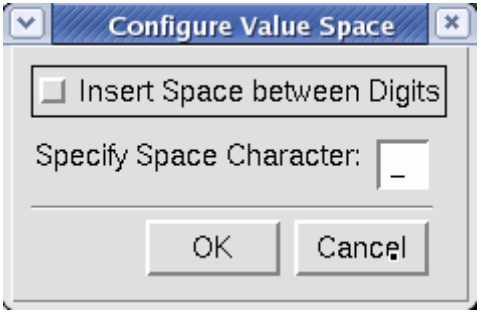


Figure: Configure Value Space Form

When the **Insert Space between Digits** option is turned *on*, the character is configured as “_”, ".", or a space to increase the readability of the value (for example, *12ab_f001* instead of *12abf001*). Its default value is “_”.

Tools Commands

Preferences

Menu Bar: Options -> Preferences

This command configures the display in the *Register Window*. It opens the **Preferences** dialog box that consists of *Font* and *Pattern* tabs.

Pattern Tab

The *Pattern* tab enables you to modify the width for **Rectangle**, **Line**, **Ellipse**, and **Arrow** objects.

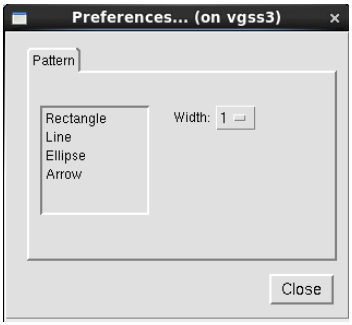


Figure: Preferences - Pattern Tab

Width: Specify the width from 1 to 5 pixels.

Customize Menu/Toolbar

For details on this command, see the **Tools -> Customize Menu/Toolbar** command in the *nTrace* chapter.

Right-Click Commands

Many of the previously described commands can also be selected from the right-click command menu on the input or output nodes in the flow views.

Register Frame Right-Click Options

When the right-mouse button is clicked anywhere in the menu, toolbar icon, or frame banner area of the *Register Window* or standalone window, a configuration option menu is displayed. This menu is used to configure the available icons and frames.

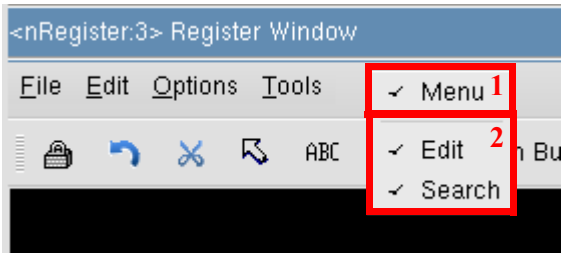


Figure: Configuration Option Menu

The **Menu** option (labeled 1 in the figure above) enables/disables the command menu bar. The options in the bottom section (labeled 2 in the figure above) enable/disable the toolbar icons for different functions.

Right-Click on Signal Object Commands

Signal Name

This toggle command enables/disables the display of the signal name.

Border

This toggle command enables/disables the display of the border.

Value Shadow

This toggle command enables/disables the display of the value shadow (background).

Edit Name

This command edits the signal name. The signal name field in the *Register Window* becomes editable. Press Enter to accept the edit.

Shift Time

This command specifies the time offset from the current time for a list of signal objects.

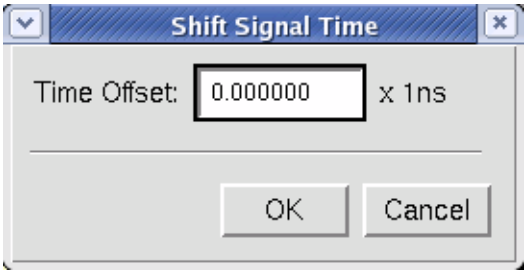


Figure: The Shift Signal Time Form

Color/Font

This command configures color and font for signals. The *Color* tab specifies the color for **Signal Name**, **Value**, or **Border** objects. Click **Default** to revert to the default color.

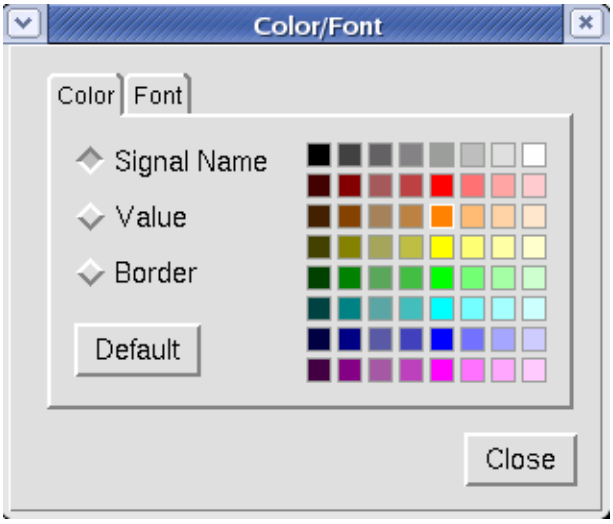


Figure: The Color/Font Form - The Color Tab

The *Font* tab specifies the font style for signals.

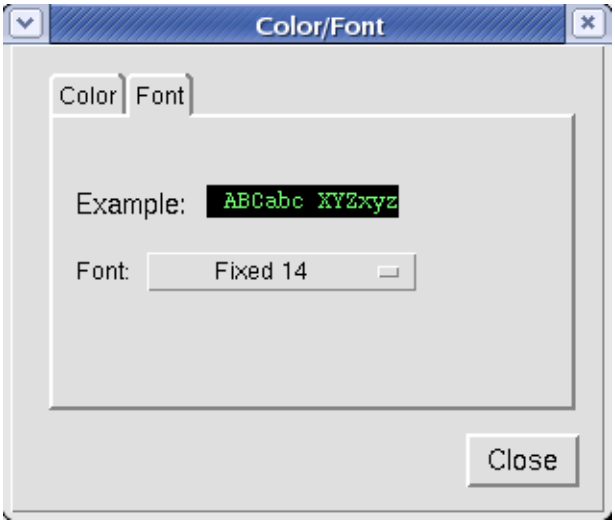


Figure: The Color/Font Form - The Font Tab

Orientation

For details on orientation, see the **Edit** -> **Orientation** command description.

Order

For details on order, see the **Edit** -> **Order** command description.

Radix

If an alias is applied to a signal in *nWave*, the same alias is reflected in the *Register Window*. Changing the radix in the *Register Window* affects the display of the same signal in other windows.

Binary

This command displays the signal value in the *Binary* format in the *Register Window*.

Octal

This command displays the signal value in the *Octal* format in the *Register Window*.

Hexadecimal

This command displays the signal value in the *Hexadecimal* format in the *Register Window*. This option is set as the default value when the *Register Window* is initially invoked.

Decimal

This command displays the signal value in the *Decimal* format in the *Register Window*.

ASCII

This command displays the signal value in the *ASCII* format in the *Register Window*. The *ASCII* format treats bytes as ASCII code and displays the characters.

Enumerated Literal

This command displays the signal value in the *Enumerated Literal* format in the *Register Window*.

Add Alias From File

For details on this, see the **Waveform -> Signal Value Radix -> Add Alias From File** command description in the *nWave* chapter.

Add Alias From Program

For details on this, see the **Waveform -> Signal Value Radix -> Add Alias From Program** command description in the *nWave* chapter.

Remove Alias

For details on this, see the **Waveform -> Signal Value Radix -> Remove Alias** command description in the *nWave* chapter.

Edit Alias

For details on this, see the **Waveform -> Signal Value Radix -> Edit Alias** command description in the *nWave* chapter.

Notation

Notation consists of **Unsigned**, **Signed**, **Signed 2's Complement, Signed 1's Complement,** and **Signed Magnitude**.

Unsigned

This command displays the signal value in the *Unsigned* format in the *Register Window*.

Signed

This command displays the signal value in the *Signed* format in the *Register Window*.

Signed 2's Complement

This command displays the signal value in the *Signed 2's Complement* format in the *Register Window*. For a table of signed binary numbers, see the **Signed 2's Complement** command description in the *nWave* chapter.

Signed 1's Complement

This command displays the signal value in the *Signed 1's Complement* format in the *Register Window*. For a table of signed binary numbers, see the **Signed 1's Complement** command description in the *nWave* chapter.

Signed Magnitude

This command displays the signal value in the *Signed Magnitude* format in the *Register Window*. For a table of signed binary numbers, see the **Signed Magnitude** command description in the *nWave* chapter.

Right-Click on Drawing Object Commands

Fill

This toggle command enables/disables the fill of an ellipse or rectangle.

Color/Pattern

This command displays the **Color/Pattern** form (see the figure below). The form enables you to specify the color (*Color* tab) and line width (*Pattern* tab) for the selected object.

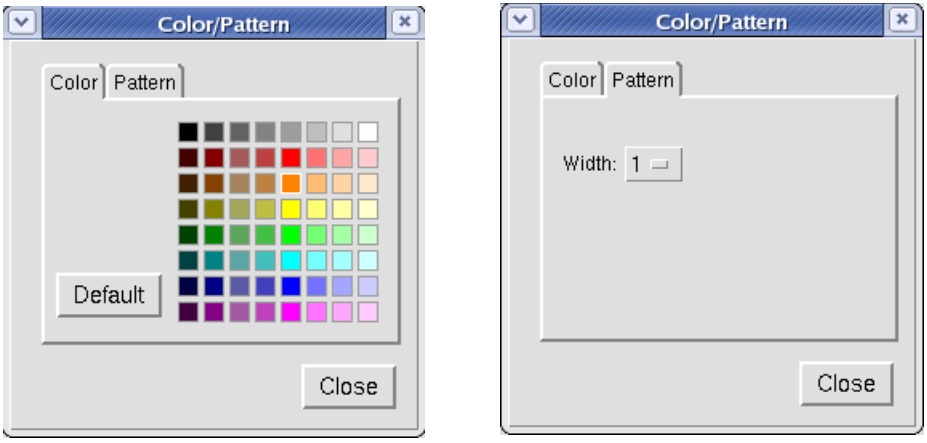


Figure: Color/Pattern - Color and Pattern Tabs

Order

For details, see the **Edit** -> **Order** command description.

Right-Click on Widget Object Commands

Edit

The **Edit** command is available for widget objects. It displays the *Edit Option Menu Widget* form (see the figure below).

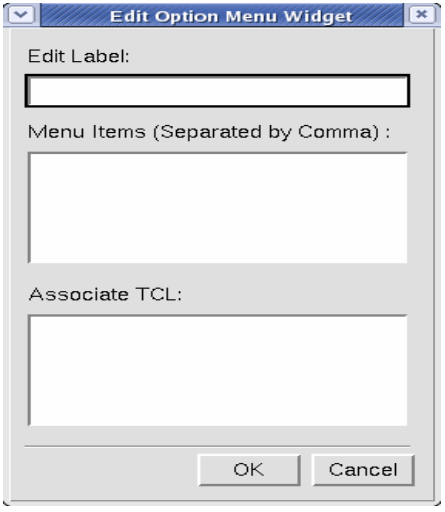


Figure: Edit Option Menu Widget Form

The *Edit Option Menu Widget* form includes the following fields:

Edit Label: Set the label of the widget object. Note that the label string must not contain more than one word (empty spaces and special characters are not supported). This setting is available for **Push Button** and **Option Menu** objects.

Menu Item: Set menu items for **Option Menu** objects. Commas are used to separate items.

Associate TcL: Associate a Tcl command with the widget object. If this field is specified, the Tcl command is evaluated after the widget object is activated by either clicking the object or pressing the Enter key. If not, then an event is triggered so that the shared library can detect whether to respond to this event or not.

Editor Option Menu Widget Example

To specify and use a menu widget, perform the following steps:

1. To recognize the object value, you must source the Tcl file, *Op123.tcl* first. To source the Tcl file, the following two options are available:
   a. Enter `source Op123.tcl` in the Tcl command entry window. The command entry window is opened by turning *on* the **Enable TCL Command Entry Form** option in the *General* page of the *Preferences* form invoked with the **Tools -> Preferences** command. Then, click the **OK** button.
   b. Execute `verdi -play Op123.tcl` on the command line.
2. Enter `Op123 $doObjVal` in the **Associate TCL** text field.

   This places the value of the menu item selected (`Op1`, `Op2`, or `Op3` in this example) in a Tcl variable called `$doObjVal`.

```
file: Op123.tcl
1  proc Op123 arg1 {
2    if [string match $arg1 "Op1"] {
3        sysInfo "User selected item1"
4    } elseif [string match $arg1 "Op2"] {
5        sysInfo "User selected item2"
6    } else {
7        sysInfo "User selected item3"
8        wvCreateWindow
9    }
10 }
```

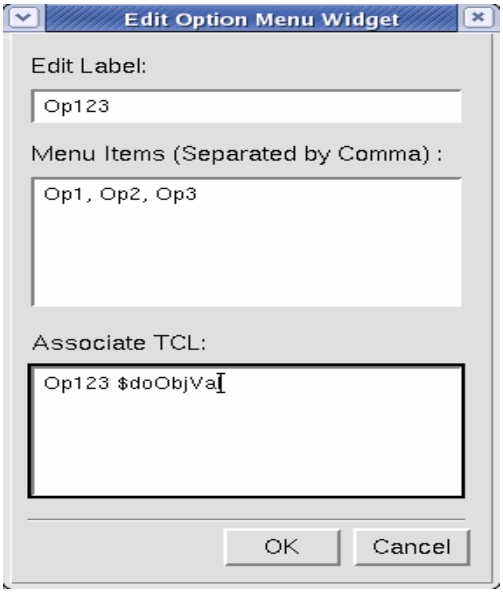


Figure: Edit Option Menu Widget Form

Toolbar Icons and Fields

The available toolbar icons may be modified. For details, see the *Toolbars* section of the *User Interface* chapter in the *Verdi User Guide and Tutorial* manual. The following figure shows the toolbar used in the *Register window*.

Figure: Toolbar Used in the Register Window

The different toolbar categories and available icons are described below.

Edit Category

Lock /Unlock

For details, see the **Edit -> Lock** and **Unlock** command descriptions.

Undo

For details, see the **Edit -> Undo** command description.

Cut

For details, see the **Edit -> Cut** command description.

Select

Click this icon to switch to the **Select** mode while editing.

Add Paragraph

Click this icon to insert a paragraph or notes in the *Register Window*. Click the right-mouse button to switch to the **Edit** mode, or change color/font settings.

Draw List

Click this icon to add lines, rectangles, ellipses, unidirectional arrows, and bidirectional arrows in the *Register Window*. The shape to the left of the triangle gets changed based on new selections. The following figure shows the options for Draw list:

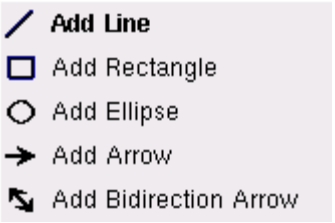


Figure: Options for Draw List

After an object is placed, click the right-mouse button to change color/pattern settings.

Add Widget List

Click this icon to add widget objects. Three types of widgets are supported: **Push Button**, **Text**, and **Option Menu**. The following figure shows the options for Widget list:

Figure: Options for Widget List

In the figure, the option in bold is the current selected object.

To edit the contents of a widget, right-click and select **Edit** from the menu list.

Color

Click this icon to display the available color palette and choose the preferred color for the selected object.

Search Category

Search By

Selects one or more signals in the *Register Window* and then sets the search criteria using the **Search By** command from one of the following options: **Any Change**, **Rising Edge**, or **Falling Edge**.

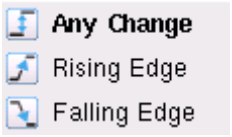


Figure: Search By List

Search Backward

Click **Search Backward** to search for the previous value change specified in the **Search By** list. The cursor time is displayed in the **Cursor Time** text field.

Search Forward

Click **Search Forward** to search for the next value change specified in the **Search By** list. The cursor time is displayed in the **Cursor Time** text field.

Cursor Time

The current cursor time is shown on the right end of the toolbar. The time unit for the *Register Window* is synchronized with *nWave*.

nEditor

Overview

nEditor is a complete text editor included with the Verdi platform. The *nEditor* pane is displayed when the **Source -> Edit Source File** menu option is selected or the equivalent toolbar icon is clicked from *nTrace*. The *nEditor* pane is docked to the same pane location as the source code pane as a new tab. The *nEditor* pane can become a standalone window by clicking the **Undock** icon on the toolbar.

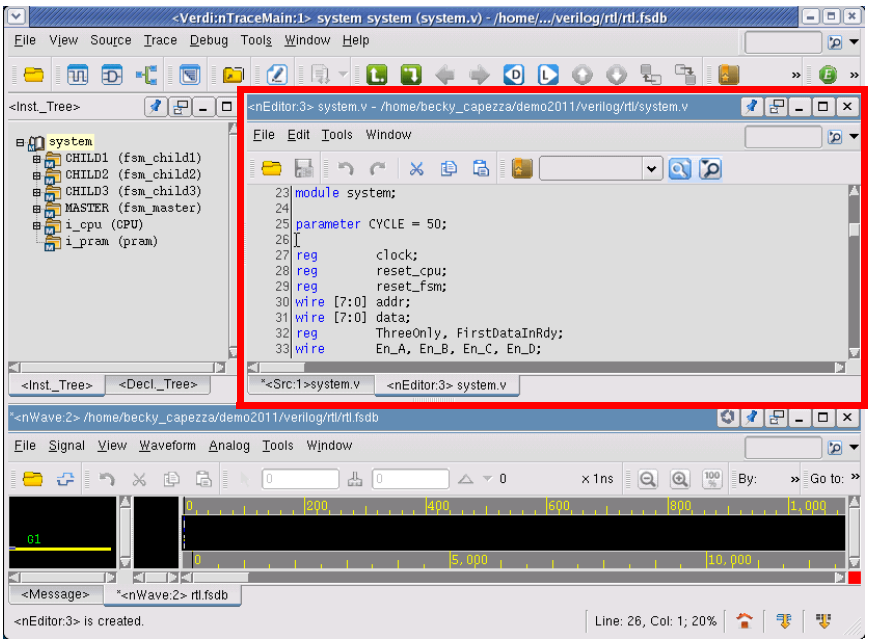


Figure: nEditor Window

When *nEditor* is invoked, the source code for the current selected module in the design browser pane is displayed.

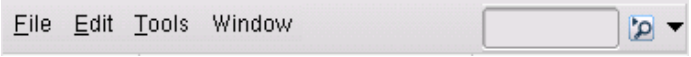


Figure: nEditor Window Menu Bar

The menus are explained on the following pages. Refer to the *Icons for Dockable Panes* section for details on the menu bar icons.

Select nEditor as the Default

The **Tools -> Preferences** option opens the *Preferences* form. To select the *nEditor*, open the **Editor** folder and click the **nEditor** page, then turn *on* the **Set as Default Editor** option and click **OK**.

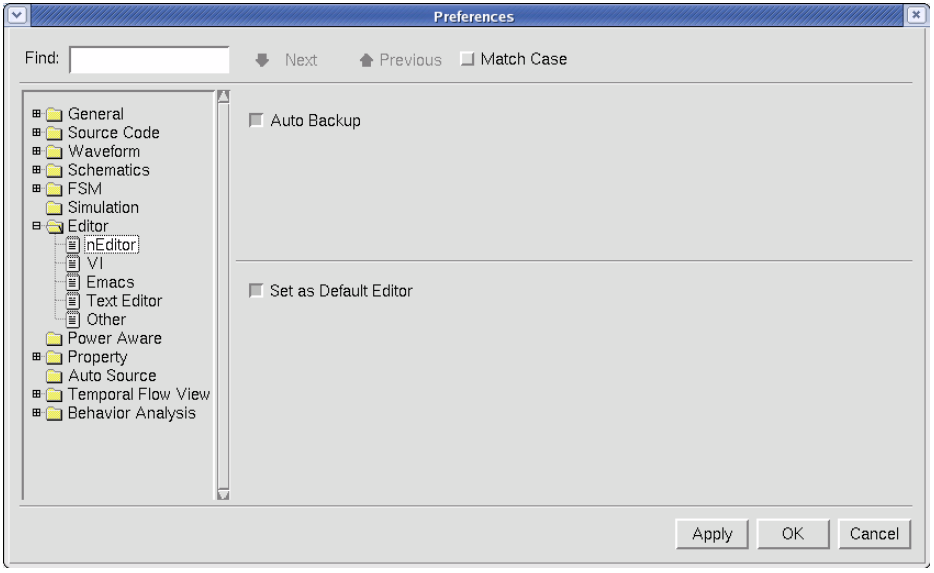


Figure: Preferences Form - nEditor Page

Menu Summary

The *nEditor* menu options are summarized below. All options can be double-clicked to jump to the corresponding option description.

File Options

New
Open
Save
Save As
Print
Close Window

Edit Options

Undo
Redo
Cut
Copy
Paste
Delete
Select All
Find String
Replace String
Go to Line
Set/Unset Bookmark
Previous Bookmark
Next Bookmark
Manage Bookmarks

Tools Options

Customize Menu/Toolbar

Window Options

New Window

Bind Keys

For a complete list of bind keys used in *nEditor*, refer to the *nEditor* section in the *Bind Key Summary* for details.

File Options

New

Menu Bar: File -> New

Bind Key: Ctrl+N

This option opens a new file in the current *nEditor* window. If a file in the *nEditor* window is not saved, a *Question* dialog window is displayed asking whether to save the current file.

Open

Menu Bar: File -> Open

Toolbar Icon: 

Bind Key: Ctrl+O

This option opens the *Open* form where the desired file can be selected.

Save

Menu Bar: File -> Save

Toolbar Icon: 

Bind Key: Ctrl+S

This option saves the current file in the *nEditor* window. If the file is new, a *Save File* form opens and the file name and location can be specified.

Save As

Menu Bar: File -> Save As

This option saves the file in the *nEditor* window with the name specified in the *Save File* form.

Print

Menu Bar: File -> Print

Bind Key: Ctrl+P

This option prints the contents of the *nEditor* window. Specify the printing options in the *Print* form.

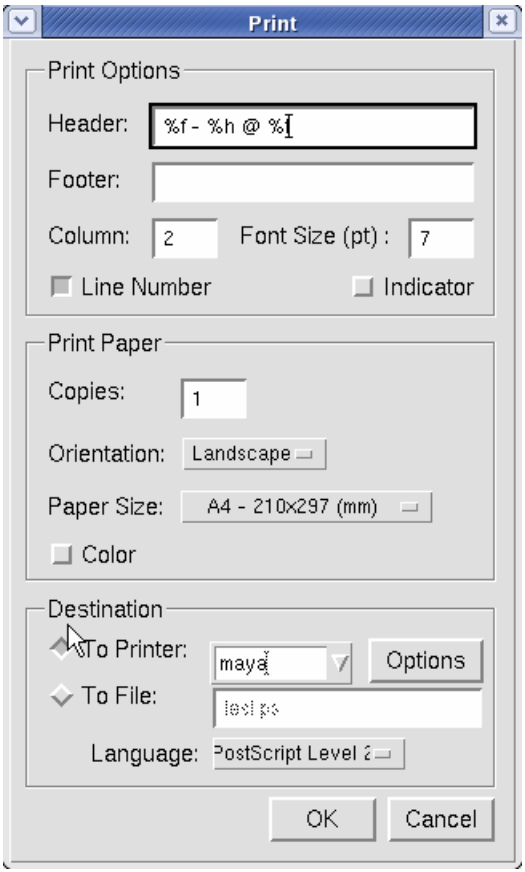


Figure: Print Form

The **Print Options** section includes the following options and fields:

- **Header:** Shows the header information on the top of a printout. By default, the header includes the file name (%f), which is currently viewed in the *nEditor* window, user name and host name (%h), and date and time (%t).
- **Footer:** Shows the footer information on the bottom of a printout. Specify the text to appear on the footer.

- **Column:** Specify how many columns per page.
- **Font Size (pt):** Specify the font size of the printout.
- **Line Number**: Shows the line number.
- **Indictor:** Turn *on* to show the indictor.

The **Print Paper** section includes the following options and fields:

- **Copies:** Enter the number of copies to print.
- **Orientation:** Specify the page orientation by selecting either **Landscape** or **Portrait**.
- **Paper Size:** Specify the supported paper size by selecting one of the following: **Letter, A4, A3, A2, A1, A0, B, C, D, E,** or **User-defined**.
- **Color:** Turn on this option to print multiple colors on a color printer. Otherwise, the printout is black and white.

The **Destination** section includes the following options and fields:

- **To Printer:** When this option is selected *on*, the printed file goes to the printer identified in the text field.
- **Options:** Click this button to configure print options in the *Configuration* form.
  - Enter the name of the printer in the **Printer** section **Name** text field.
  - Specify the printer language by selecting one of the following: **Postscript Level 1**, **Postscript Level 2**, or **HP GL/2**.
  - Choose the **Paper Size**. If **User-defined** is chosen, the width and the height must be configured under **User-defined**.
  - In the **Print Command and Options** section, the default **Command** is *lpr*, the default **Destination** is *-P*, and the default number of **Copies** is *-#*. These entries can be used for printing the file and they may be configured for a specific print command.
- **To File:** Specify the file name in the text field, as the printed file can be saved to the working directory.
- **Language:** Specify the language one of the following: **PostScript Level 2**, **PostScript Level 1**, or **GL2**.

Close Window

Menu Bar: File -> Close Window

This option closes the current *nEditor* window.

Edit Options

Undo

Menu Bar: Edit -> Undo

Toolbar Icon: 

Bind Key: Ctrl+Z

This option reverses the last action performed, if possible. To undo an action, invoke the **Edit -> Redo** option.

Redo

Menu Bar: Edit -> Redo

Toolbar Icon: 

Bind Key: Ctrl+Y

This option repeats the last action performed, if possible.

Cut

Menu Bar: Edit -> Cut

Toolbar Icon: 

Bind Key: Ctrl+X

This option removes the selected section and places a copy on the clipboard.

Copy

Menu Bar: Edit -> Copy

Toolbar Icon: 

Bind Key: Ctrl+C

This option places a copy of the current selection on the clipboard.

Paste

Menu Bar: Edit -> Paste

Toolbar Icon: 

Bind Key: Ctrl+V

This option places the information cut or copied to the clipboard into the *nEditor* window at the current cursor position.

Delete

Menu Bar: Edit -> Delete

This option removes the selected text.

Select All

Menu Bar: Edit -> Select All

Bind Key: Ctrl+A

This option selects the entire contents of the *nEditor* window.

Find String

Menu Bar: Edit -> Find String

Toolbar Icon: 

Bind Key: Ctrl+F

This option opens the *Find String* form where the specified text can be searched.

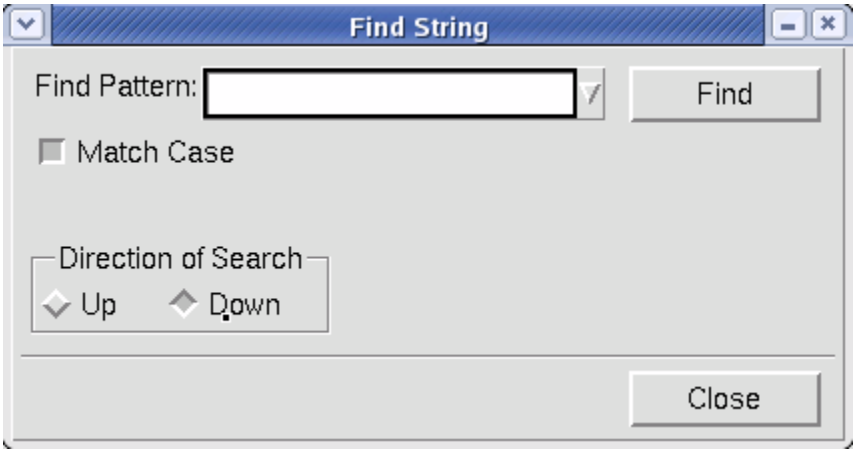


Figure: Find String Form

The *Find String* form includes the following text field, options and button:

- **Find Pattern:** Type the text in this text field or drag the string from the *nTrace* window.
- **Find:** Click this button to highlight the instances matching the specifications in the *nEditor* window one by one.
- **Match Case:** When this option is turned *on*, find only those instances where the capitalization matches the text typed in the **Find Pattern** text field. The default is *on*.
- **Direction of Search:** Specify the search direction from the current position. Turn *on* the **Up** option to search backward or turn *on* the **Down** option to search forward.

Replace String

Menu Bar: Edit -> Replace String

Bind Key: Ctrl+H

This option opens the *Replace String* form where the specified text can be searched and replaced.

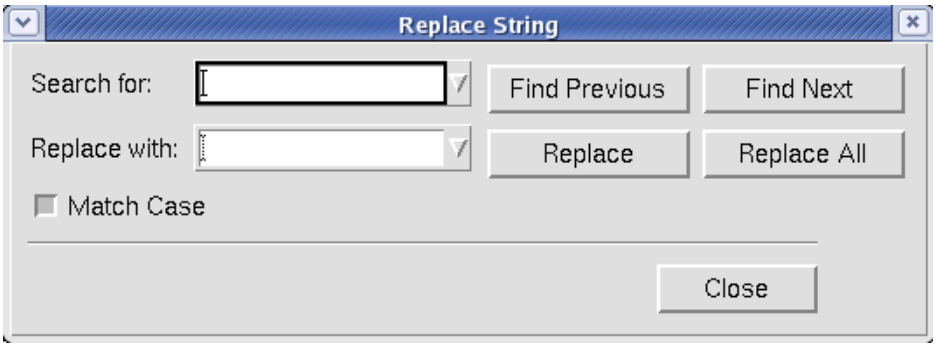


Figure: Replace String Form

The *Replace String* form includes the following text fields, buttons and option:

- **Search for:** Enter the text in this text field for the string to be searched for.
- **Find Previous:** Click this button to search for the previous string specified in the **Search for** text field.
- **Find Next:** Click this button to search for the next string specified in the **Search for** text field.
- **Replace with:** Enter the string to be replaced in this text field.
- **Replace:** Click this button to replace the string in the **Search for** text field.
- **Replace All:** Click this button to replace the string in the **Search for** text field automatically.
- **Match Case:** When this option is turned *on*, find only those instances where the capitalization matches the text typed in the **Search for** text field. The default is *on*.

Go to Line

Menu Bar: Edit -> Go to Line

Bind Key: Ctrl+G

This option opens the *Go to Line* form where the line number can be entered in the **Line Number** text field. Click the **OK** button to place the cursor at the beginning of the specified line in the *nEditor* window.

Set/Unset Bookmark

Menu Bar: Edit -> Set/Unset Bookmark

Toolbar Icon: 

Bind Key: Ctrl+F2: Toggle Bookmark

This option sets or clears a bookmark at the current line of the source code.

Previous Bookmark

Menu Bar: Edit -> Previous Bookmark

Bind Key: Shift+F2

This option shows the previous bookmark in the current file.

Next Bookmark

Menu Bar: Edit -> Next Bookmark

Bind Key: F2

This option shows the next bookmark in the current file.

Manage Bookmarks

Menu Bar: Edit -> Manage Bookmarks

This option opens the *Manage Bookmarks* form where the set bookmarks can be deleted, renamed, or located. Refer to the **Source -> Manage Bookmarks** option description in the *nTrace* chapter for details.

Tools Options

Customize Menu/Toolbar

Refer to the **Tools -> Customize Menu/Toolbar** option description in the *nTrace* chapter for details.

Window Options

New Window

Menu Bar: Window -> New Window

This option opens a new blank *nEditor* window as a new tab.

nEditor Window Right-click Options

When the right mouse button is clicked anywhere in the menu, toolbar icon, or banner area of the *nEditor* pane or standalone window, a configuration option menu is displayed. This menu can be used to configure the available icons and panes.

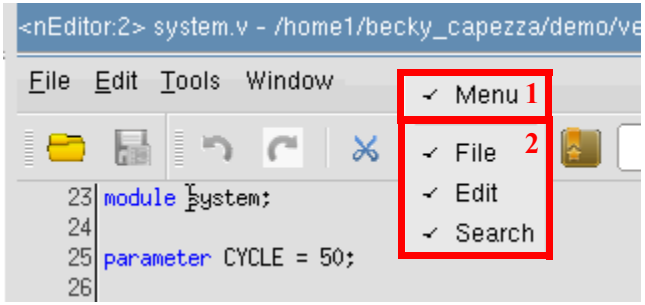


Figure: Configuration Option Menu

The **Menu** option (labeled 1 in the figure above) enables/disables the option menu bar. The options in the bottom section (labeled 2 in the figure above) enable/disable the toolbar icons for different functions.

Toolbar Icons and Fields

Figure: Toolbar Used in the nEditor Window

The available toolbar icons may be modified. Refer to the *Toolbars* section of the *User Interface* chapter in the *Verdi User Guide and Tutorial* manual for details.

The different toolbar categories and available icons are described below.

File Category

Open 

Click this toolbar icon to open a source file for editing. Refer to the **File** -> **Open** option description for details.

Save 

Click this toolbar icon to save changes to the source file. Refer to the **File** -> **Save** option description for details.

Edit Category

Undo 

Refer to the **File** -> **Undo** option description for details.

Redo 

Refer to the **File** -> **Redo** option description for details.

Cut 

Refer to the **File** -> **Cut** option description for details.

Copy 

Refer to the **File** -> **Copy** option description for details.

Paste 

Refer to the **File** -> **Paste** option description for details.

Search Category

Set/Unset Bookmark 

Refer to the **Edit** -> **Set/Unset Bookmark** option description for details.

Find String 

Refer to the **Edit** -> **Find String** option description for details.

Find Previous 

Click this toolbar icon to find the previous string specified in the **Search for** text field.

Find Next 

Click this toolbar icon to find the next string specified in the **Search for** text field.

Memory/MDA Pane

Overview

The *Memory/MDA* pane is displayed when you invoke the **Tools -> Memory/MDA** option from *nTrace* or *nWave* window or you select the **Debug Memory -> Show Memory Contents** option from the right-click option in *nTrace window*. The *Memory/MDA* pane is docked to the right of the *Message/nWave* window. The *Memory/MDA* pane can become a standalone pane when you click the **Undock** icon on the toolbar.

In the *Memory/MDA* pane, the memory data stored in an FSDB file can be viewed. Each *Memory/MDA* pane only displays the contents of a memory variable. Refer to the *FSDB Dumping Commands* chapter of the *Linking Novas Files with Simulators and Enabling FSDB Dumping* document for a list of FSDB dumping commands for memories.

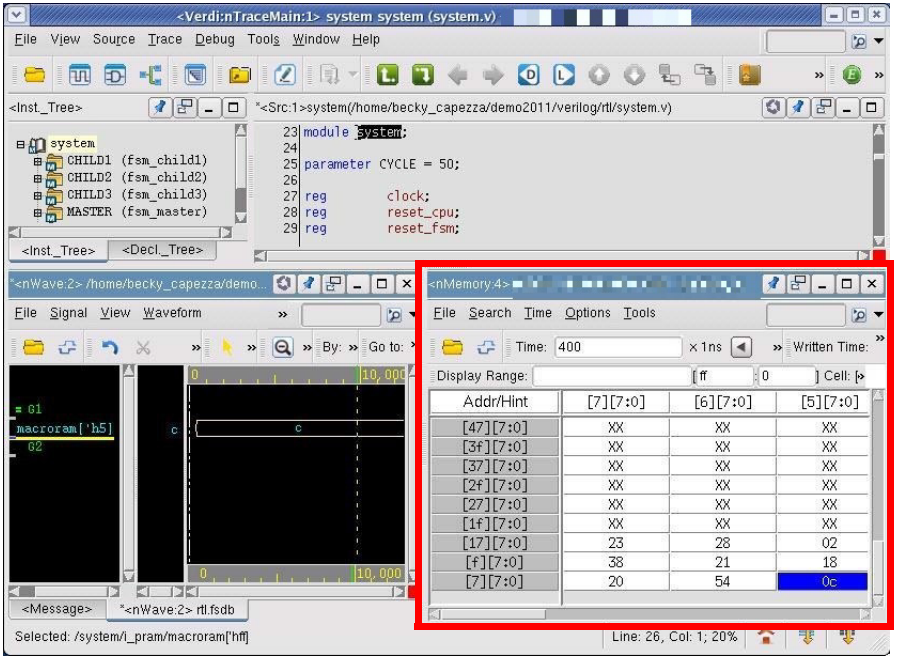


Figure: Memory/MDA Pane

After loading an FSDB file containing a memory variable (including two-dimensional and greater arrays), one or more array elements can be selected and added to the *nWave* window. When the array is greater than 2048 vector signals (such as `mem[0:4095][0:7]`), it cannot be displayed as a signal waveform. If the array element is less than 2048 vector signals (such as `mem2[0:1023][0:7]` or `mem3[0:255][0:4095]`), it is automatically displayed as a signal waveform (such as `mem2[0:1023]` or `mem3[0:255]`). After the array element is displayed in *nWave* window, double-click the signal for further expansion.

NOTE: The selected two-dimensional array can be dragged from the *nTrace* window source code pane and dropped to the *nWave* waveform window.

The menu bar for the *Memory/MDA* pane is as follows:

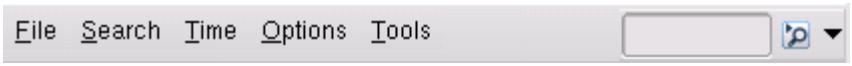


Figure: Memory/MDA Pane Menu Bar

The pull-down menus are explained on the following pages. Refer to the *Icons for Dockable Panes* section for details on the menu bar icons.

Menu Summary

The *Memory/MDA* pane menu options are summarized below. You can double-click all options to jump to the corresponding option description. For right-click options, refer to the *Right-Click Options* section for details.

File Menu Options

Get Memory Variable
Create Slice Variable
Create Concatenate Variable
Restore Session
Save Session
Save to File
Close

Search Menu Options

Find

Time Menu Options

Search By
Next Dump
Previous Dump
Update
Sync Cursor Time

Options Menu Options

Value Radix ->
  Binary
  Octal
  Hexadecimal
  Decimal
  ASCII
  Add Alias from File
  Add Alias from Program
  Remove Alias
  Edit Alias
Value Notation ->
  Unsigned
  Signed 2's Complement
  Signed 1's Complement
  Signed Magnitude
Address Radix ->
  Octal
  Hexadecimal
  Decimal
Display Mode ->
  Address Hint
  Address Value
Words Shown in One Row
Preferences

Tools Menu Options

Customize Menu/Toolbar

File Menu Options

Get Memory Variable

Menu Bar: **File -> Get Memory Variable**

Toolbar Icon:

This option opens the *Get Memory Variable* form. When you click the **OK** button, the memory contents is shown in the *Memory/MDA* pane. The correct memory value can be attained from mapping/searching the rows and columns.

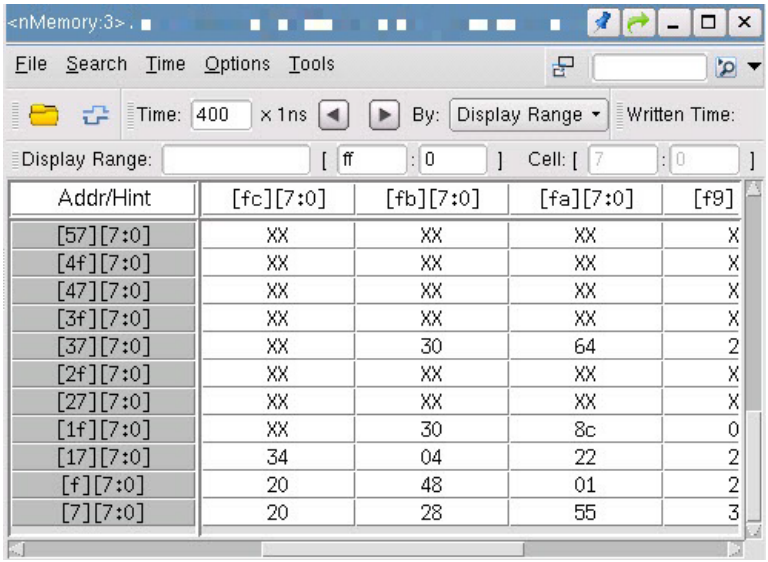


Figure: Memory/MDA Pane with Loaded Memory Variable

The *Get Memory Variable* form includes the **Dumped by Simulator** and **Calculated by Verdi** tabs.

Dumped by Simulator Tab

In the **Dumped by Simulator** tab, select the memory variable (including two-dimensional and greater arrays) to display it in the *Memory/MDA* pane from the loaded FSDB file.

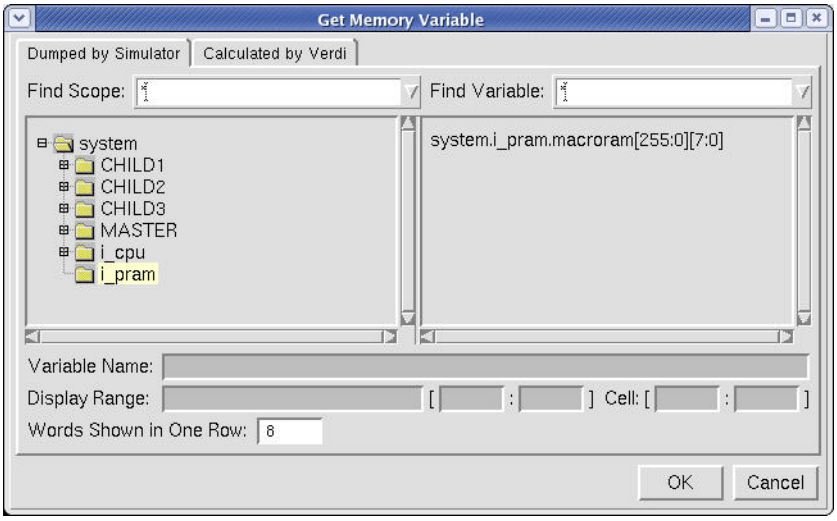


Figure: Get Memory Variable Form - Dumped by Simulator Tab

The **Dumped by Simulator** tab includes the following fields and buttons:

- **Find Scope:** This text field displays the scope selected in the design hierarchy section. Enter a partial scope name in the **Find Scope** text field, and the matched scopes are highlighted in the left pane. The memory variables associated with the matched scopes are listed in the right pane. The scope name is case-sensitive.
- **Find Variable:** This text field filters the variables to be displayed in the *Memory/MDA* pane. The variable name is case-sensitive.
- **Variable Name:** When you select the variable from the variable list, the **Variable Name** text field shows the full name of the variable.
- **Display Range:** This field displays the address range of the selected variable. Specify the displayed address range to be displayed in the *Memory/MDA* pane.
- **Cell:** This field displays the word bit range of the selected variable. Specify the word bit range to be displayed in the *Memory/MDA* pane.
- **Words Shown in One Row:** This text field specifies the number of words to be displayed in a row in the *Memory/MDA* pane. The default is *8*.
- **OK:** Click **OK** to import and display the specified memory variable in the *Memory/MDA* pane.
- **Cancel:** Click **Cancel** to close the *Get Memory Variable* form without making any changes.

Calculated by Verdi Tab

In the **Calculated by Verdi** tab, you can specify the memory variable, display range, and time conditions for memory calculation. The memory contents are calculated for the specified time and displayed in the *Memory/MDA* pane.

This tab can also be invoked from the **Debug Memory** -> **Show Memory Contents** option in the *nTrace* right-click option or from the **Tools** -> **Show Memory Contents** option in the *Temporal Flow View* menu bar.

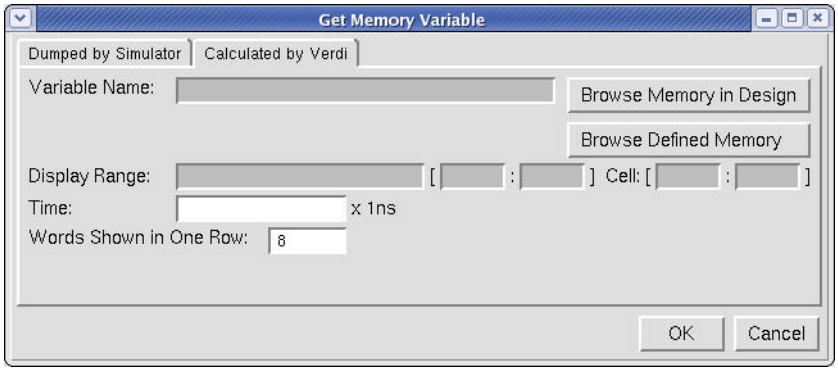


Figure: Get Memory Variable Form - Calculated by Verdi Tab

The **Calculated by Verdi** tab includes the following fields and buttons:

- **Variable Name:** When you select a memory type, the **Variable Name** text field shows the full hierarchical path of the memory by default. If no memory is selected, this field is blank. To add or change the memory variable, enable the **Browse Memory in Design** or **Browse Defined Memory** options to locate and select memory variables. You are not allowed to type in this field.
  - **Browse Memory in Design:** Click this button to open the *Signal Manager* form where you can browse the design hierarchy and select the memory variable. Signals under the selected design scope are displayed in the signal list section. After selecting the memory variable, click the **OK** button to enter the variable in the **Variable Name** text field.

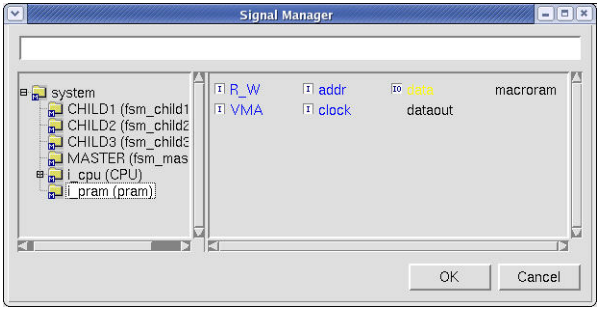


Figure: Signal Manager Form for Memory Browsing

- **Browse Defined Memory:** This button opens the *Signal Manager* form where the design hierarchy can be browsed and a predefined memory description can be selected. The defined memories under the selected design scope are displayed in the signal list section. After selecting the predefined memory, click the **OK** button to enter the defined variable in the **Variable Name** text field. See the **Memory Definition Table** option description in the *nTrace* chapter for details on creating a memory description.

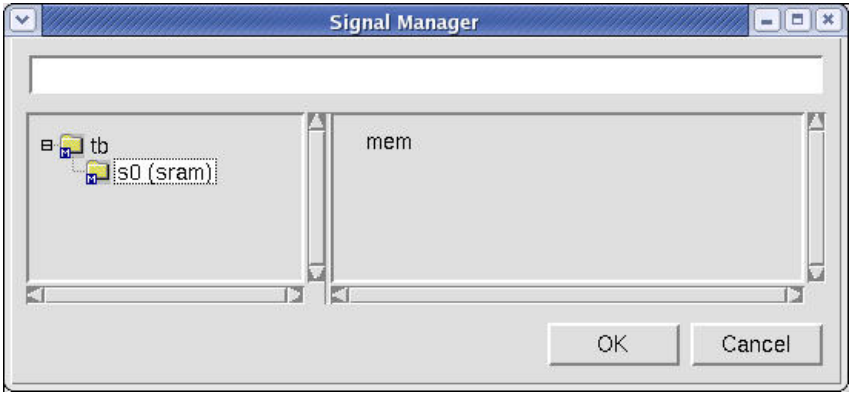


Figure: Signal Manager Form for Defined Memory Browsing

- **Display Range:** Specify the start and end address for the memory contents. When you select the variable name in the **Variable Name** field, the start and end address for the memory range is shown automatically. By default, full range for the memory is shown automatically. Partial range is shown when you type in the range text field. You cannot modify modifications can be made in the cell range of the **Cell** text fields. When an MDA is selected, the address range of the memory shown on the left text field of **Display Range** can also be modified.

- **Time:** Specify the time to calculate and display the memory contents. If no memory is selected when the **Show Memory Contents** option is invoked, the field is left blank. If a memory is selected in *nTrace*, it uses the global cursor time by default. If a memory is selected in a flow view, it goes to the last access time for that memory address by default. The last access time is for a read if the memory has not been expanded. The last access time is for a write to that address location if the memory has been expanded (double-left-click).
- **Words Shown in One Row:** This text field specifies the number of words to be displayed in a row in the *Memory/MDA* pane. The default is *8*.
- **OK:** Click this button to calculate the memory variable contents for the specified time and display the result in the *Memory/MDA* pane.
- **Cancel:** Click this button to close the *Get Memory Variable* form without making any changes.

Create Slice Variable

Menu Bar: **File -> Create Slice Variable**

This option opens the *Create Slice Variable* form where a memory slice, instead of the whole memory variable, can be created.

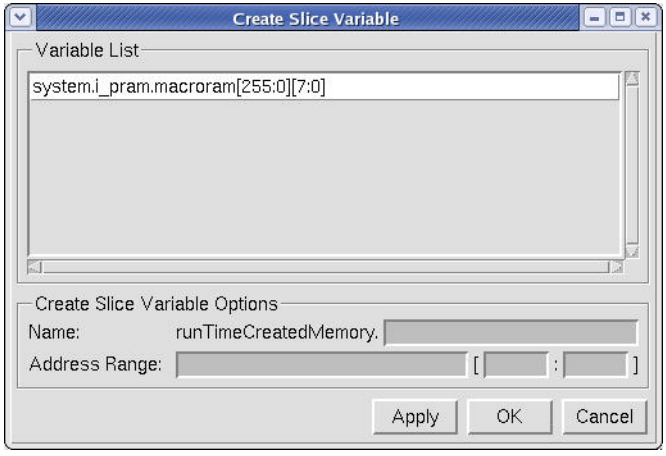


Figure: Create Slice Variable Form

The *Create Slice Variable* form includes the following two sections:

- **Variable List:** This field displays the name and address range of the originally dumped and newly created memory variable.

- **Create Slice Variable Options**
  - **Name:** Specify the name of the newly created memory slice.
  - **Address Range:** Specify the address range of the newly created memory slice.

Create Concatenate Variable

Menu Bar: **File -> Create Concatenate Variable**

This option opens the *Create Concatenate Variable* form where two or more memory arrays can be concatenated into one memory variable. For example, if there are two memory variables, memA and memB and both have word type [0:7] and range from 0 to 255, the new memory variable can be either (1) with word type [0:15] and range from 0 to 255, or (2) with word type [0:7] and range from 0 to 512.

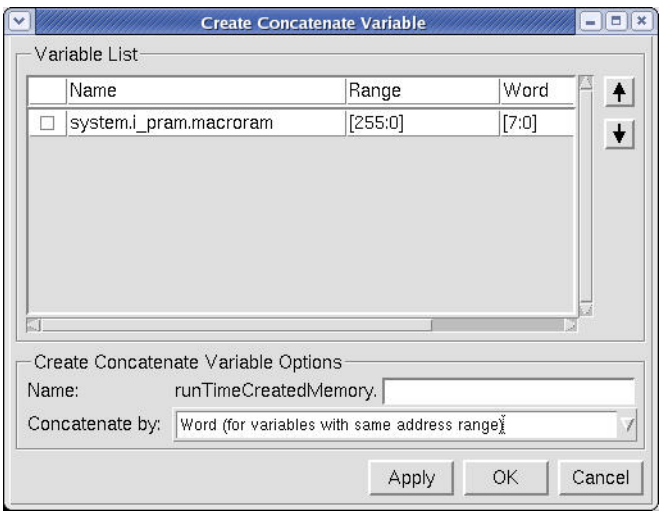


Figure: Create Concatenate Variable Form

The *Create Concatenate Variable* form includes the following two sections:

- **Variable List:** This column displays the name, address range, and words of the originally dumped and newly created memory variable(s). Check the desired memory variable(s) to be concatenated. Click ↑ or ↓ to move the selected memory variable up or down. The memory variables can also be sorted by clicking on the **Name**, **Range**, and **Word** column heading. By sorting the memory variables, a different concatenated variable can be attained, as the order of memory variables has been changed.

- **Create Concatenate Variable Options**
  - **Name:** Specify the name of the newly created memory variable.
  - **Concatenate By:** Specify a concatenating scheme. If the specified method is **Word**, the memory variables must be of the same address range so that they can be concatenated word by word. If the method chosen is **Address Range**, the memory variables must be of the same bit range so that they can be linked into a variable with a larger address range.

Restore Session

Menu Bar: **File -> Restore Session**

Toolbar Icon: 

This option opens the *Restore Session* form where the previously saved session file can be loaded. The restored session file includes such information as the concatenation and slice memory settings in the *Memory/MDA* pane. Refer to the **File -> Print** option description in the *nTrace* chapter for details.

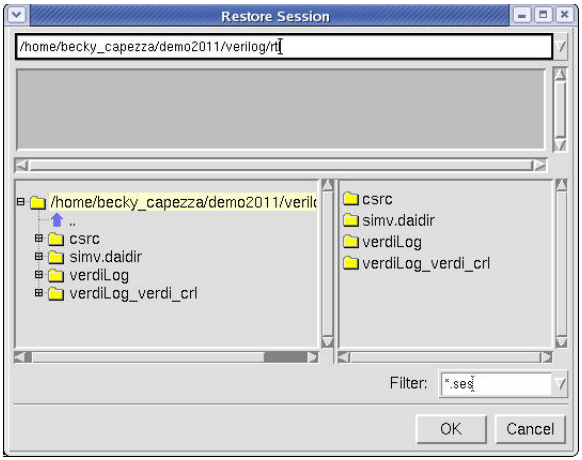


Figure: Restore Session Form

Save Session

Menu Bar: **File -> Save Session**

This option opens the *Save Session* form where the settings for concatenation and slice memories, currently loaded memories, and the memory display settings in the *Memory/MDA* pane can be saved to a given session name. By default, the file is saved with the ".ses" file extension. The session files are shown automatically when the **Restore Session** option is invoked.

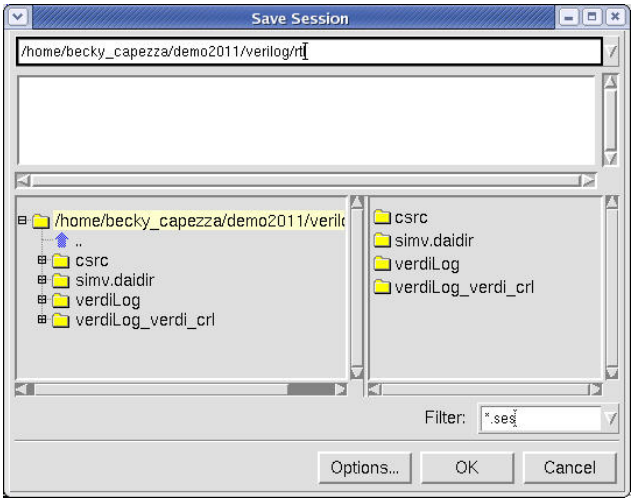


Figure: Save Session Form

Click the **Options** button to open the *Session File Options* form where the memory settings to be saved into the session file can be selected.

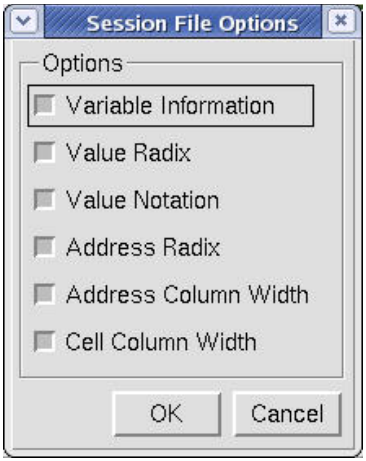


Figure: Session File Options Form

The *Session File Options* form includes the following options (multiple options can be turned on simultaneously):

- **Variable Information:** When this option is turned on, the memory information with the FSDB file path is included in the saved session file. The default is *on*.
- **Value Radix:** When this option is turned on, the variable value radix is included in the saved session file. The default is *on*.
- **Value Notation:** When this option is turned on, the variable value notation is included in the saved session file. The default is *on*.
- **Address Radix:** When this option is turned on, the address format is included in the saved session file. The default is *on*.
- **Address Column Width:** When this option is turned on, the address column width is included in the saved session file. The default is *off*.
- **Cell Column Width:** When this option is turned on, the cell column width is included in the saved session file. The default is *off*.

Save to File

Menu Bar: **File -> Save to File**

This option opens the *Save to File* form where the currently displayed memory can be saved into a text (`.txt`) file with the following information: memory block name with full hierarchy, date, simulation time, and display range.

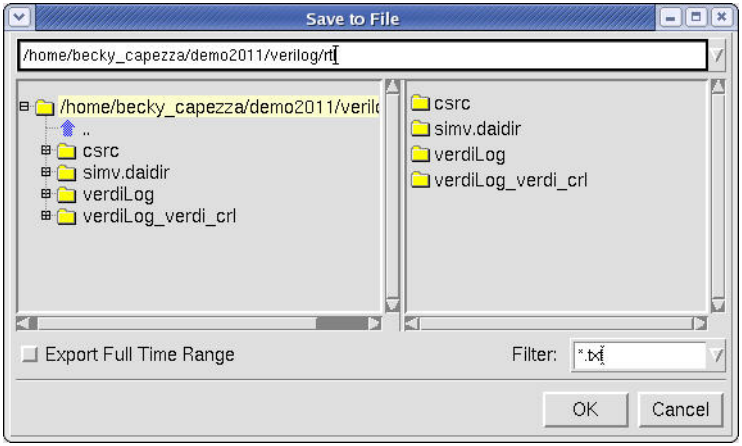


Figure: Save To File Form

Export Full Time Range: When this option is turned on, all dumped memories are saved. When this option is turned off, only currently dumped memory is saved.

Filter: This is the file format filter for the file list. The default file extension in the **Filter** field is `*.txt`. Additional filters can be added in this selection field using a semicolon `;` as the separator.

Close

Menu Bar: **File -> Close**

This option closes the *Memory/MDA* pane.

Search Menu Options

Find

Menu Bar: **Search -> Find**

This option opens the *Find* form where a pattern to be searched for can be specified in the **Pattern** text field. Click  or  to find the next or previous pattern match. Click **Close** to close the *Find* form.

Figure: Find Form

Time Menu Options

Search By

Menu Bar: **Time -> Search By**

Toolbar Icon: Display Range / Selected

The search criterion for the value change of the memory can be chosen as **Display Range** or **Selected**.

- **Display Range:** Find the dump with value changes in the current display range whether the cells are selected or not.
- **Selected:** Find the dump with value changes in the selected display cells.

Next Dump

Menu Bar: **Time -> Next Dump**

Toolbar Icon:

This option jumps the time field to the next available time point. The *Memory/MDA* pane shows the memory contents at that time point.

Previous Dump

Menu Bar: **Time -> Previous Dump**

Toolbar Icon: 

This option jumps the time field to the previous available time point. The *Memory/MDA* pane shows the memory contents at that time point.

Update

Menu Bar: **Time -> Update**

Bind Key: Ctrl+U

For **Interactive Mode** only, this option is used to dump the memory contents until the current time from the simulator.

Sync Cursor Time

Menu Bar: **Time -> Sync Cursor Time**

This toggle option turns the synchronization of the current time in the *Memory* and *nWave* windows on and off. When it is turned off, the *Memory/MDA* pane works independently. When it is turned on, the time is set according to the current cursor time in *nWave*.

Options Menu Options

Value Radix

Another way to invoke the **Value Radix** option is to place the cursor on any of the value cells and right-click to open the right-click menu option.

Binary

Menu Bar: **Options -> Value Radix -> Binary**

This option displays the variable value in **Binary** format in the *Memory/MDA* pane.

Octal

Menu Bar: **Options -> Value Radix -> Octal**

This option displays the variable value in **Octal** format in the *Memory/MDA* pane.

Hexadecimal

Menu Bar: **Options -> Value Radix -> Hexadecimal**

This option displays the variable value in **Hexadecimal** format in the *Memory/MDA* pane. This option is set as the default value when the *Memory/MDA* pane is initially invoked.

Decimal

Menu Bar: **Options -> Value Radix -> Decimal**

This option displays the variable value in **Decimal** format in the *Memory/MDA* pane.

ASCII

Menu Bar: **Options -> Value Radix -> ASCII**

This option displays the variable value in **ASCII** format in the *Memory/MDA* pane. The ASCII format treats bytes as ASCII code and displays the characters.

Add Alias from File

Menu Bar: **Options -> Value Radix -> Add Alias from File**

This option associates an alias definition file with the loaded variables. The alias is displayed instead of the variable's numeric value when the variable value matches the predefined alias string. Refer to **Waveform**-> **Value Radix** -> **Add Alias From File** option description in the *nWave* chapter for syntax information.

Add Alias from Program

Menu Bar: **Options -> Value Radix -> Add Alias from Program**

Refer to **Waveform -> Value Radix** -> **Add Alias From Program** option description in the *nWave* chapter for details.

Remove Alias

Menu Bar: **Options -> Value Radix -> Remove Alias**

This option removes any aliases from the selected signals and reverts them back to display their numeric values.

Edit Alias

Menu Bar: **Options -> Value Radix -> Edit Alias**

Refer to the Waveform -> Value Radix-> **Edit Alias** option description in the *nWave* chapter for details.

Value Notation

This option can also be invoked by placing the cursor on any of the **Value** cells and right-clicking to choose the desired notation from the right-click menu option.

Unsigned

Menu Bar: **Options -> Value Notation -> Unsigned**

This option displays the variable value in **Unsigned** format in the *Memory/MDA* pane.

Signed 2's Complement

Menu Bar: **Options -> Value Notation -> Signed 2's Complement**

This option displays the variable value in **Signed 2's Complement** format in the *Memory/MDA* pane. Refer to the **Signed 2's Complement** option description in the *nWave* chapter for the table of signed binary numbers.

Signed 1's Complement

Menu Bar: **Options -> Value Notation -> Signed 1's Complement**

This option displays the variable value in **Signed 1's Complement** format in the *Memory/MDA* pane. Refer to the **Signed 2's Complement** option description in the *nWave* chapter for the table of signed binary numbers.

Signed Magnitude

Menu Bar: **Options -> Value Notation -> Signed Magnitude**

This option displays the variable value in **Signed Magnitude** format in the *Memory/MDA* pane. Refer to the **Signed 2's Complement** option description in the *nWave* chapter for the table of signed binary numbers.

Address Radix

This option changes the address display format. This option can also be invoked by placing the cursor on any of the **Address Hint** cells and right-clicking to choose a radix from the menu. The following address formats are available:

Octal

Menu Bar: **Options -> Address Radix -> Octal**

This option displays the address radix in **Octal** format in the *Memory/MDA* pane.

Hexadecimal

Menu Bar: **Options -> Address Radix -> Hexadecimal**

This option displays the address radix in **Hexadecimal** format in the *Memory/MDA* pane.

Decimal

Menu Bar: **Options -> Address Radix -> Decimal**

This option displays the address radix in **Decimal** format in the *Memory/MDA* pane.

Display Mode

Two memory display modes are available: **Address Hint** and **Address Value**.

Address Hint

Menu Bar: **Options -> Display Mode ->Address Hint**

An **Address Hint** row is added above the cell value pane. Click to select a row in the **Address Hint** column and the corresponding address for each selected cell is shown on the **Address Hint** row.

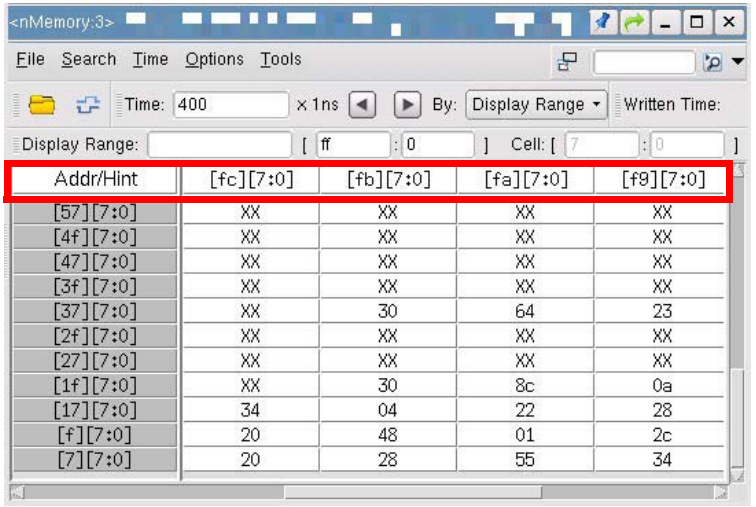


Figure: Address Hint Mode

Address Value

Menu Bar: **Options -> Display Mode ->Address Value**

The contents of each cell are shown in **Address-Value** format. Putting the address and value in the same cell makes it easier to locate the address and corresponding value quickly; however, this causes the address to occupy more space.

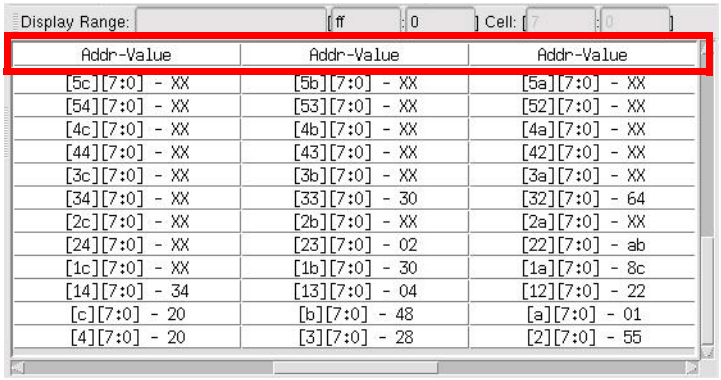


Figure: Address Value Mode

Words Shown in One Row

Menu Bar: **Options -> Words Shown in One Row**

This option opens the *Words Shown in One Row* form where the number of words to be displayed in a row can be specified. The default is *8*.

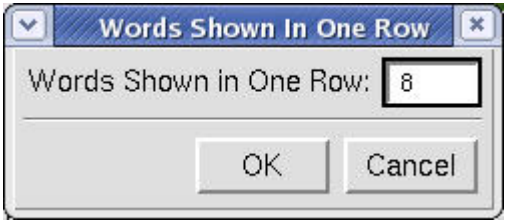


Figure: Words Shown in One Row Form

Preferences

Menu Bar: **Options -> Preferences**

This option opens the *Preferences* form where the display settings for the *Memory/MDA* pane can be specified. The settings are applied to the current pane and other newly opened *Memory/MDA* panes.

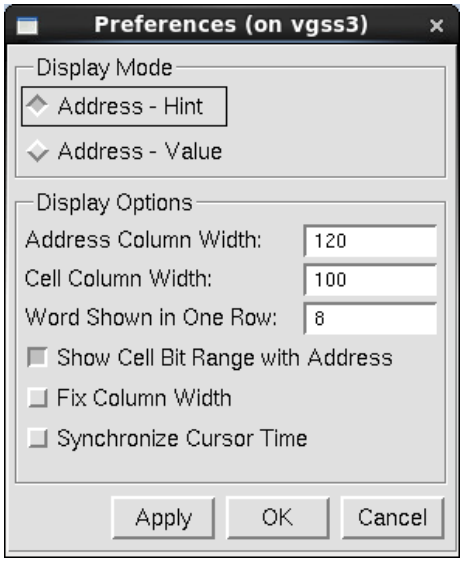


Figure: Preferences Form

Two memory display modes are included in the **Display Mode** section: **Address Hint** and **Address Value**. Refer to the **Display Mode** option for details.

The **Display Options** section includes the following options and fields:

- **Address Column Width:** This text field specifies the address column width.
- **Cell Column Width:** This text field specifies the cell column width.
- **Words Shown in One Row**: This text field specifies how many words to display in a row. The default is *8*.
- **Show Cell Bit Range with Address:** This option turns the display of the cell bit range with address *on* and *off*. The default is *on*.
- **Fix Column Width**: When this option is turned on, the column width is fixed. When this option is turned off, the column width can be changed. The default is *off*.
- **Synchronize Cursor Time**: Refer to the **Sync Cursor Time** option for details.

Tools Menu Options

Customize Menu/Toolbar

Refer to the **Tools -> Customize Menu/Toolbar** option in the *nTrace* chapter for details.

Right-Click Options

Many of the previously described options can also be selected from the right-click menu options on the input or output nodes in the flow views.

Register Pane Right-Click Options

When the right mouse button is clicked anywhere in the menu, toolbar icon, or pane banner area of the *Memory/MDA* pane or standalone pane, a configuration option menu is displayed.This menu can be used to configure the available icons and panes.

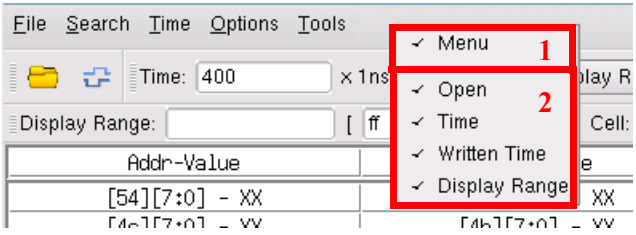


Figure: Configuration Option Menu

The **Menu** option (labeled 1 in the figure above) enables/disables the option menu bar. The options in the bottom section (labeled 2 in the figure above) enable/disable the toolbar icons for different functions.

Heading Row Right-Click Options

Fix Column Width

Refer to the **Fix Column Width** option description on the *Preferences* form (opened with the **Options -> Preferences** option).

Show Cell Bit Range

Refer to the **Show Cell Bit Range with Address** option description on the *Preferences* form (opened with the **Options -> Preferences** option).

Address Hint Column Right-Click Options

Octal

Refer to the **Options -> Address Radix -> Octal** option description for details.

Hexadecimal

Refer to the **Options -> Address Radix -> Hexadecimal** option description for details.

Decimal

Refer to the **Options -> Address Radix -> Decimal** option description for details.

Show Cell Bit Range

Refer to the **Show Cell Bit Range with Address** option description on the *Preferences* form (opened with the **Options -> Preferences** option).

Cells Right-Click Options

Binary

Refer to the **Options -> Value Radix -> Binary** option description for details.

Octal

Refer to the **Options -> Value Radix -> Octal** option description for details.

Hexadecimal

Refer to the **Options -> Value Radix -> Hexadecimal** option description for details.

Decimal

Refer to the **Options -> Value Radix -> Decimal** option description for details.

ASCII

Refer to the **Options -> Value Radix ->** **ASCII** option description for details.

Notation

Refer to the **Options ->** **Value Notation** option descriptions for details.

Remove Alias

Refer to the **Options -> Value Radix ->** **Remove Alias** option description for details.

Toolbar Icons and Fields

Figure: Toolbar Used in the Memory/MDA Pane

The available toolbar icons may be modified. Refer to the *Toolbars* section of the *User Interface* chapter in the *Verdi User Guide and Tutorial* manual for details.

The different toolbar categories and available icons are described below.

Open Category

Restore Session 

Refer to the **File ->** **Restore Session** option description for details.

Get Memory Variable 

Refer to the **File ->** **Get Memory Variable** option description for details.

Time Category

Time 

This field displays the current cursor time in the *Memory/MDA* pane. The time unit is synchronized with *nWave*.

Search Backward 

Click **Search Backward** to search for the previous value change specified in the **Search By** list. The cursor time is displayed in the **Time** text field.

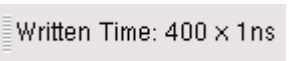
Search Forward 

Click **Search Forward** to search for the next value change specified in the **Search By** list. The cursor time is displayed in the **Time** text field.

Search By 

Refer to the **Time -> Search By** option description for details.

Written Time Category

Written Time 

This field displays the time when the selected memory cell was written.

Display Range Category

Display Range 

These fields specify and display the range for the loaded memory.

Transaction/Message Analyzer

Overview

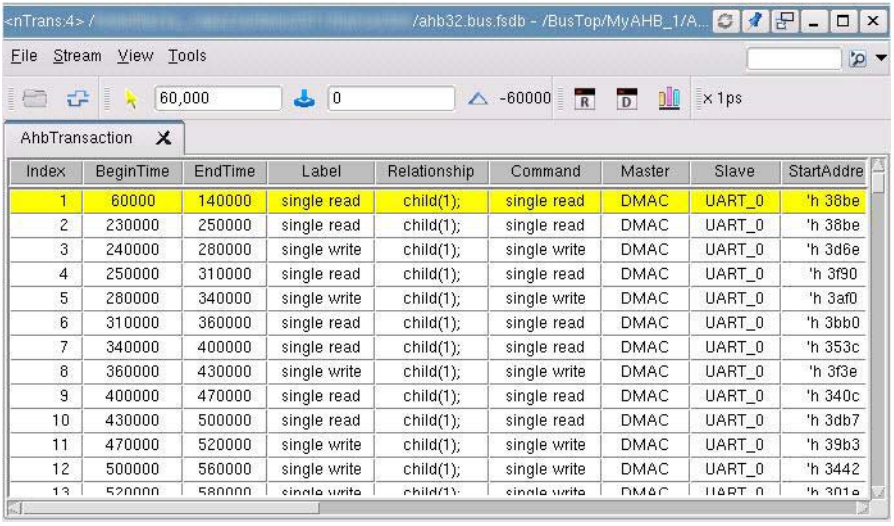
The *Transaction/Message Analyzer* frame is displayed when **Tools -> Transaction Debug -> Classic Transaction -> Analysis Window** or **Tools -> Transaction Debug -> Classic Message -> Analysis Window** commands are invoked from *nTrace* or *nWave* menu bar. An FSDB file with transaction or message data must be loaded into the Verdi platform for these commands to be utilized. The *Transaction/Message Analyzer* frame is docked to the right of the *nWave* frame as a new frame. The default view in the *Transaction/Message Analyzer* frame is the *Analysis Window*.

The *Analysis Window* view acts as the main console for the *Transaction/Message Analyzer* frame. Other views, such as, *Comparison Window*, *Data Window*, *Statistics Window*, and *Relationship Window* can be opened from the *Analysis Window*. Refer to the appropriate section in this chapter for details.

Analysis Window

In the *Analysis Window*, transactions or messages are displayed as a spreadsheet and the results can be sorted in a variety of ways. Multiple streams can be loaded into the same *Analysis Window* view or different *Transaction/Message Analyzer* frames can be used for different streams. The maximum number of streams that can be loaded in one *Analysis Window* view is 10 (with two rows of tabs).

Transactions or messages can be added to the *Analysis Window* by dragging them from *nWave* or using the **Stream -> Get Stream** command. A transaction/message in the *Analysis Window* view can also be dragged and dropped to the *nWave* frame.



Index	BeginTime	EndTime	Label	Relationship	Command	Master	Slave	StartAddre
1	60000	140000	single read	child(1);	single read	DMAC	UART_0	'h 38be
2	230000	250000	single read	child(1);	single read	DMAC	UART_0	'h 38be
3	240000	280000	single write	child(1);	single write	DMAC	UART_0	'h 3d6e
4	250000	310000	single read	child(1);	single read	DMAC	UART_0	'h 3f90
5	280000	340000	single write	child(1);	single write	DMAC	UART_0	'h 3af0
6	310000	360000	single read	child(1);	single read	DMAC	UART_0	'h 3bb0
7	340000	400000	single read	child(1);	single read	DMAC	UART_0	'h 353c
8	360000	430000	single write	child(1);	single write	DMAC	UART_0	'h 3f3e
9	400000	470000	single read	child(1);	single read	DMAC	UART_0	'h 340c
10	430000	500000	single read	child(1);	single read	DMAC	UART_0	'h 3db7
11	470000	520000	single write	child(1);	single write	DMAC	UART_0	'h 39b3
12	500000	560000	single write	child(1);	single write	DMAC	UART_0	'h 3442

Figure: Transaction/Message Analyzer - Analysis Window

Each transaction/message stream or merged stream has its own tab with a complete summary displayed in the *Analysis Window*. The default order for the stream is *BeginTime*. A transaction/message may be selected with a left-click anywhere along the row. The width of each column can be adjusted by selecting the vertical line between column headers and dragging-left. The columns can be sorted by clicking the column headers. The *Analysis Window* includes four standard columns: *Index*, *BeginTime*, *EndTime*, and *Label*. The standard column headings are summarized as follows:

- **Index:** Indicates the number assigned to each transaction or message according to its begin time.
- **BeginTime:** Corresponds to the begin time of the selected transaction or message.

- **EndTime:** Corresponds to the end time of the selected transaction or message.
- **Label:** Displays the label information of the selected transaction or message.

Additional column headings may be included as defined by attributes in transactions or messages.

The *Analysis Window* view can become a standalone window by clicking the **Undock** icon on the toolbar.

The menu bar for the *Analysis Window* is as follows:

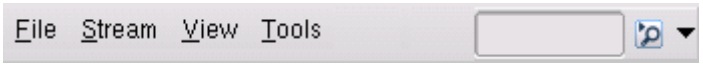


Figure: Analysis Window Menu Bar

The pull-down menus for the *Analysis Window* are summarized and explained on the following pages. Refer to the *Icons for Dockable Panes* section for details on the menu bar icons.

File Commands

Open FSDB

Menu Bar: *Analysis Window*, **File** -> **Open FSDB**

Toolbar Icon: 

This command opens the *Select FSDB File* form where a waveform dump file can be located. To open a file, type in the exact file name, directory name, or a wildcard string in the *File Name* text field. When a wildcard string is used, all files under the current directory that match the given string are filtered. These are listed in the *File List* section. After specifying the file name, click **OK** to load the file and close the form or click **Cancel** to close the form without loading a file.

If an FSDB file has already been loaded in *nWave*, this command is not available since only one FSDB file can be loaded in the *Analysis Window*. The *Analysis Window* uses the FSDB file in *nWave* as the loaded FSDB file by default. To use a different FSDB file in the *Analysis Window*, the current file must be closed before opening a new window. If the specified FSDB file does not contain transaction streams, a warning message "*There is no transaction in <your_directory>/your_FSDB_file.fsdb*" is displayed.

Close FSDB

Menu Bar: *Analysis Window*, **File** -> **Close FSDB**

This command closes the current FSDB file.

Save to File

Menu Bar: *Analysis Window*, **File** -> **Save to File**

This command opens a *Save As* form where the directory structure can be viewed and a text file name can be specified for saving the *Analysis Window* contents to it.

Save Session

Menu Bar: *Analysis Window*, **File** -> **Save Session**

This command opens the *Save Session* form where the directory structure can be viewed and information such as stream merging and column configuration can be saved to the specified session name. By default, the file is saved with a ".ses" file extension. The session files are shown automatically when the **Restore Session** command is used.

Restore Session

Menu Bar: *Analysis Window*, **File** -> **Restore Session**

This command opens the *Restore Session* form where the previously saved session file containing information such as stream merging and column configuration can be loaded in the *Analysis Window*. This feature is used to continue a previous debugging session without repeating the setup steps.

Close Window

Menu Bar: *Analysis Window*, **File** -> **Close Window**

This command closes the *Transaction/Message Analysis Window* frame.

Stream Commands

Get Stream

Menu Bar: *Analysis Window*, **Stream** -> **Get Stream**

Toolbar Icon: 

This command opens the *Select Stream* form where a stream can be selected and added to the *Analysis Window*. The stream relationship in the entire FSDB file is listed as a tree-like structure.

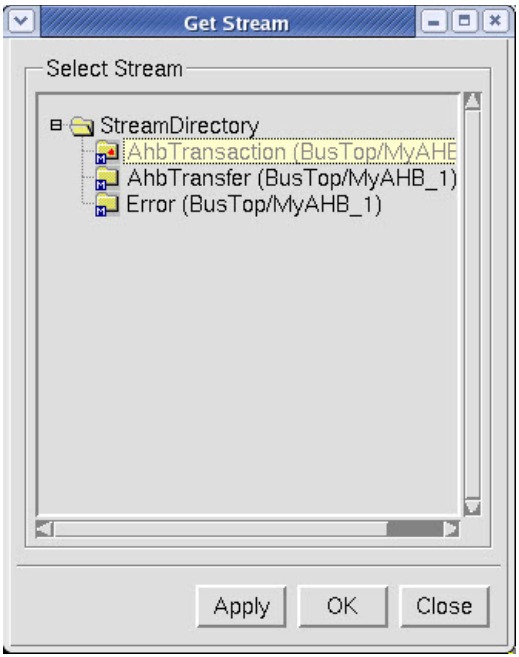


Figure: Select Stream Form

This form displays all transaction/message streams with full hierarchical path in the opened FSDB file. Stream(s) which are displayed in the *Analysis Window* are in gray color and cannot be added to the *Analysis Window* again. Only one stream can be selected at a time.

Apply: Click this button to add the selected stream to the *Analysis Window* as a new tab and keep the *Select Stream* form open. Alternatively, double-click a stream to add it to the *Analysis Window.*

OK: Click this button to add the selected stream to the *Analysis Window* as a new tab and close the *Select Stream* form.

Close: Click this button to close the form without adding a stream.

Close Stream

Menu Bar: *Analysis Window*, **Stream** -> **Close Stream**

This command closes the active stream on the current selected tab.

Merge Stream

Menu Bar: *Analysis Window*, **Stream** -> **Merge Stream**

This command opens the *Merge Stream* form where two or more streams can be selected to merge into a new stream. All the transactions/messages, along with their attributes are merged and displayed in the *Analysis Window*.

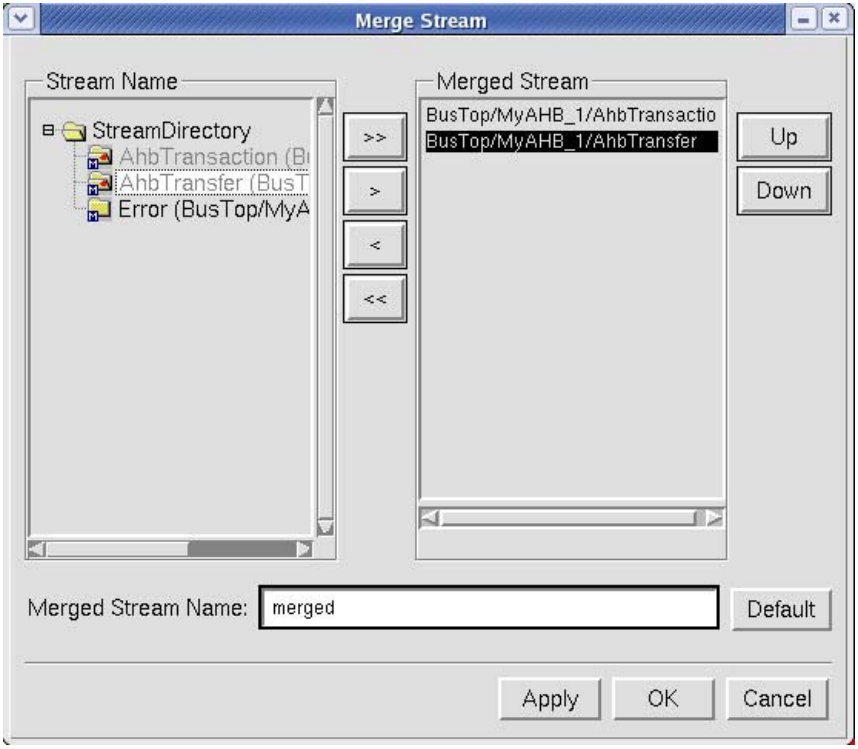


Figure: Merge Stream Form

The *Merge Stream* form includes the following buttons and fields:

- **Stream Name:** This section lists all the streams with full hierarchical path present in the selected FSDB file.
- **Merged Stream:** This section lists all the streams added from the *Stream Name* column in the order they were added.
- **Move All on Left to Right** >> **:** This button adds all the streams in the **Stream Name** column to the *Merged Stream* column. All the streams in the *Stream Name* column are set to gray color.
- **Moved Selected on Left to Right** > **:** This button adds the selected stream in the *Stream Name* column to the *Merged Stream* column. Only one stream can be selected at a time. To add a stream to the *Merged Stream* column, double-click the stream. The selected streams in the *Stream Name* column are set to gray color.
- **Move Selected on Right to Left** < **:** This button removes the selected stream from the *Merged Stream* column. The stream name is restored (color is set to black) to the *Stream Name* column. Only one stream can be selected at a time.
- **Move All on Right to Left** << **:** This button removes all streams from the *Merged Stream* column. The stream names are restored (color is set to black) to the *Stream Name* column.
- **Up:** This button moves the selected streams in the *Merged Stream* column one row up. The default order for the merged stream is based on the *BeginTime*. If the streams have the same begin time, the results for the top stream are listed first.
- **Down:** This button moves the selected streams in the *Merged Stream* column one row down. The default order for the merged stream is based on the *BeginTime*. If the streams have the same begin time, the results for the top stream are listed first.
- **Merged Stream Name:** Enter the desired name for the merged stream in this text field. Click the **Default** button to unite all of the stream names in the *Merged Stream* column as the merged stream name, connected with "_" in between. For example, the default merged stream name for streams *MyAHB_1/AhbTransaction* and *MyAPB_1/ApbTransaction* is *AhbTransaction_ApbTransaction*.

View Commands

Search

Menu Bar: *Analysis Window*, **View** -> **Search**

This command opens the *Search* form where the attributes to search for in the transaction/message streams in the *Analysis Window* can be set.

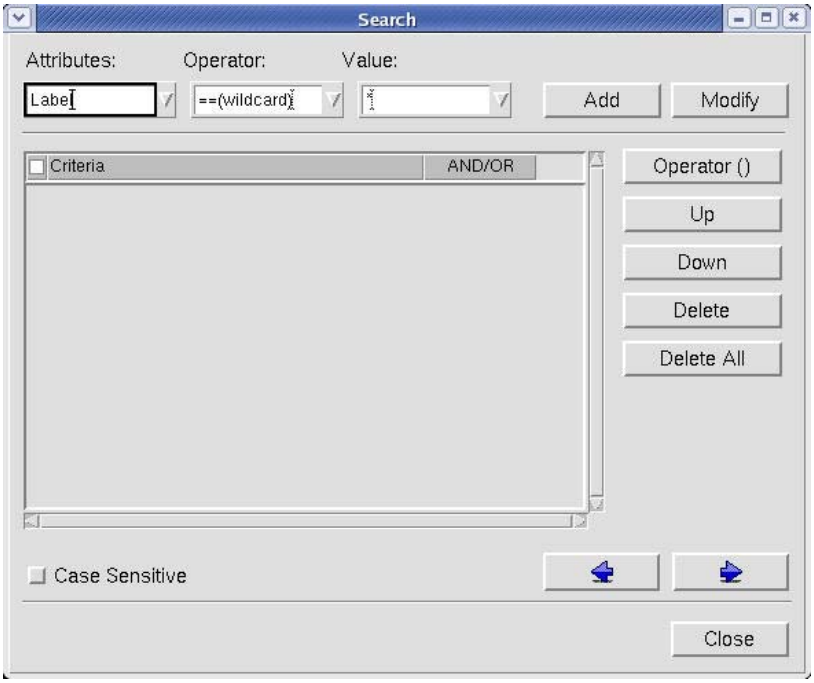


Figure: Search Form

The following fields, buttons, and options are available:

- **Attributes:** Select the attribute name from the **Attributes** selection list or type the attribute name directly in this text field.
- **Operator:** Select the operator from the **Operator** selection list. The following operators are available: **== (wildcard)**, **=~ (regular expression)**, **!=**, **>**, **>=**, **<**, **<=**.
- **Value:** Enter the value of the attribute(s).
- **Add:** Click this button to add the specified criteria to the criteria list with the check box enabled by default.

- **Modify:** When the attribute, operator, or value is changed, click the **Modify** button to replace the current criteria with the updated criteria.

The criteria display area lists the added criteria. In the **Criteria** column, if the check box is enabled, the search is conducted based on the specified criteria. All the check boxes can be checked or unchecked by toggling *on* or *off* the column header. In the **AND/OR** column, select either **AND** or **OR** to define the relationship between the current criteria row and the criteria in the row above.

- **Operator ():** Click this button to add an open parenthesis "(" to the first row and a close parenthesis ")" with the AND operator to the last row of the criteria list.
- **Up:** Click this button to move the selected item in the criteria list one row up.
- **Down:** Click this button to move the selected item in the criteria list one row down.
- **Delete:** Click this button to remove the selected criteria from the criteria list.
- **Delete All:** Click this button to remove all criteria from the criteria list.
- **Case Sensitive:** When this option is turned *on*, attributes are searched for with case-sensitivity. When this option is turned *off*, the attributes are searched for without case-sensitivity. The default value of this option is *off*.
- **Search Backward**  **:** Click this button to find the previous change based on the specified search criteria.
- **Search Forward**  **:** Click this button to find the next change based on the specified search criteria.

Filter/Colorize

Menu Bar: *Analysis Window*, **View** -> **Filter/Colorize**

This command opens the *Filter/Colorize* form where the filter/color criteria can be specified for the selected transaction in the *Analysis Window*.

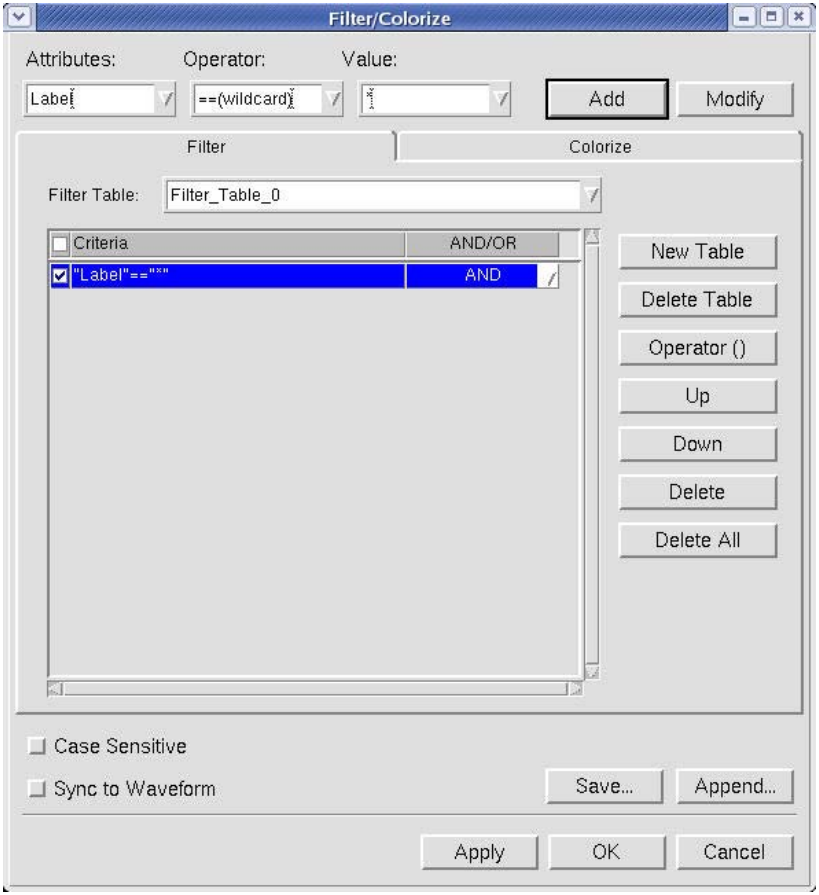


Figure: Filter/Colorize Form - Filter Tab

The following fields, options, and buttons are available in the *Filter/Colorize* form:

- **Attributes:** Select the attribute name from the *Attributes* selection list or type the attribute name directly in this text field.
- **Operator:** Select the operator from the *Operator* selection list. The following operators are available: == (wildcard), =~ (regular expression), **!**=, >, >=, <, <=.
- **Value:** Enter the value of the attribute(s).
- **Add:** Click this button to add the specified criteria to the criteria list with the check box enabled by default.
- **Modify:** When the attribute, operator, or value is changed, click the **Modify** button to replace the current criteria with the updated criteria.

- **Case Sensitive:** When this option is turned *on*, attributes are searched for with case-sensitivity. When this option is turned *off*, the attributes are searched for without case-sensitivity. The default value of this option is *off*.
- **Sync to Waveform:** When this option is turned *on*, the filtered results are reflected in the *Transaction/Message Analysis Window* as well as the *nWave* frame. When this option is turned *off*, the filtered results are only reflected in the current *Transaction/Message Analysis Window*. The default value of this option is *off*.
- **Save:** Click this button to open the *Save Filter/Colorize Tables to File* form where the directory structure can be viewed and a file specified to save the filter and color tables to.
- **Append:** Click this button to open the *Load Filter/Colorize File* form where the directory structure can be viewed and previously saved filter/ colorize files can be loaded.

Filter Tab

The following fields and buttons are available in the *Filter* tab:

- **Filter Table:** This text field lists all the available filter tables with the default name. The default name can be changed by typing in this text field.

  The criteria display area lists the added criteria. In the *Criteria* column, if the check box is enabled, the search is conducted based on the specified criteria. All the check boxes can be checked or unchecked by toggling the column header *on* or *off*. In the AND/OR column, select either AND or OR to define the relationship between the current criteria row and the criteria in the row above.
- **New Table:** Click this button to add a new filter table to the *Filter Table* text field with a default name (Filter_Table_n, where ‘n’ is the next available number).
- **Delete Table:** Click this button to remove the current filter table from the **Filter Table** text field.
- **Operator ():** Click this button to add an open parenthesis "(" to the first row and a close parenthesis ")" with the AND operator to the last row of the criteria list. This button is available on the *Filter* tab.
- **Up:** Click this button to move the selected item in the criteria list up one row.
- **Down:** Click this button to move the selected item in the criteria list down one row.
- **Delete:** Click this button to remove the selected criteria from the criteria list.

- **Delete All:** Click this button to remove all criteria from the criteria list.

Colorize Tab

In addition to the buttons described previously for the *Filter* tab, the following fields and buttons are available on the *Colorize* tab:

- **Color Table:** This text field lists all the available color tables with the default name.

  The criteria display area lists the default criteria. If the check box is enabled in the **Criteria** column, the colors are applied based on the specified criteria. All check boxes can be checked or unchecked by toggling the column header *on* or *off*.

- **Add Predefined:** Click this button to add the predefined severity color to the user-defined color table. This button is available on the *Colorize* tab.
- **Default**: Click this button to reset all colors back to their original default settings. This button is available on the *Colorize* tab.

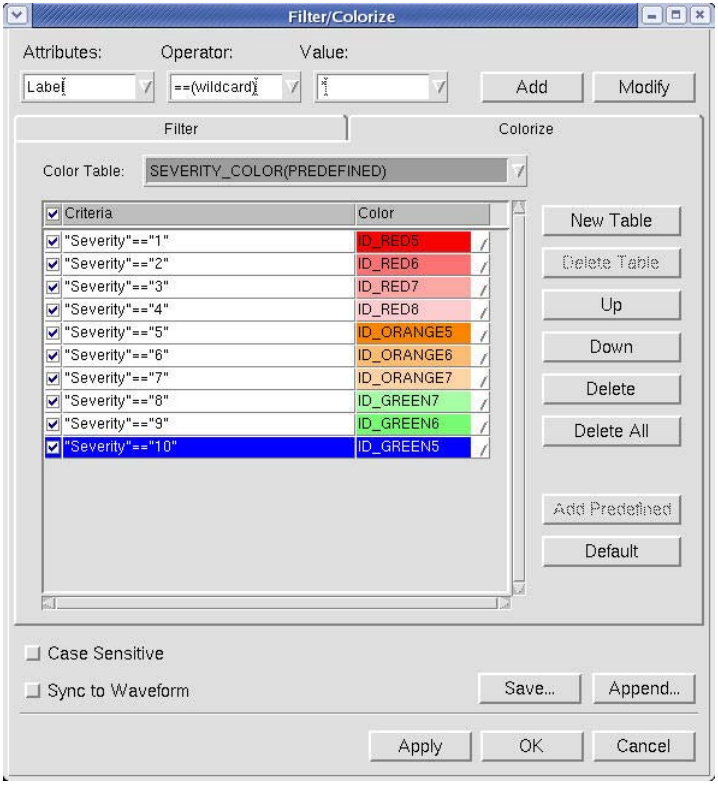


Figure: Filter/Colorize Form - Colorize Tab

Clear Filtering Results

Menu Bar: *Analysis Window*, **View -> Clear Filtering Results**

This command clears the results from a previously applied filter in the *Analysis Window*.

Clear Colorization Results

Menu Bar: *Analysis Window*, **View -> Clear Colorization Results**

This command clears the color results in the *Analysis Window*.

Configure Columns

Menu Bar: *Analysis Window*, **View -> Configure Columns**

This command opens the *Config Bus Table* form where the columns for the active stream in the *Analysis Window* can be configured, and the columns to be displayed and the desired order can be selected.

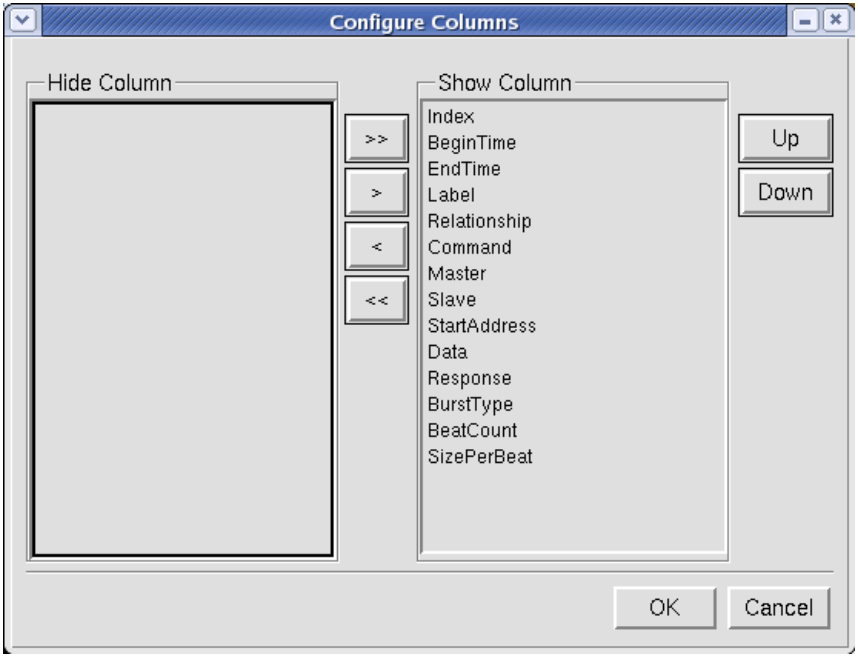


Figure: Config Bus Table Form

The following sections and buttons are available in the *Config Bus Table* form:

- **Hide Column:** This section lists the column(s) to be removed from the display in the *Analysis Window*.
- **Show Column:** This section lists the column(s) to be displayed in the *Analysis Window*.
- **Move All on Left to Right** >> **:** This button adds all of the columns in the **Hide Column** list to the **Show Column** list.
- **Move Selected on Left to Right** > **:** This button moves the selected column name from the *Hide Column* list to the *Show Column* list. Only one column can be selected at a time.
- **Move Selected on Right To Left** < **:** This button moves the selected column name from the *Show Column* list to the *Hide Column* list. Only one column can be selected at a time.
- **Move All on Right to Left** << **:** This button moves all columns in the *Show Column* list back to the *Hide Column* list.
- **Up:** This button moves the selected column to an upper display order. The column at the top of the list corresponds to the left-most column.
- **Down:** This button moves the selected column to a lower display order. The column at the bottom of the list corresponds to the right-most column.

Jump to Cursor

Menu Bar: *Analysis Window*, **View** -> **Jump to Cursor**

Toolbar Icon: 

This command jumps to the cursor row in the *Analysis Window*. A cursor row is created with a left-click anywhere on a row. The transaction/message is then highlighted in yellow color.

Jump to Marker

Menu Bar: *Analysis Window*, **View** -> **Jump to Marker**

Toolbar Icon: 

This command jumps to the marker row in the *Analysis Window*. A marker row is created with a middle-click anywhere on a row. The transaction/message is then highlighted in red color.

Sync. Cursor Time

Menu Bar: *Analysis Window*, **View -> Sync. Cursor Time**

This command synchronizes the *Transaction/Message Analysis Window* cursor time with Verdi or *nWave* cursor time. When this command is turned *on*, the *Transaction/Message Analysis Window* changes the cursor time for all streams when Verdi or *nWave* cursor time is changed. Likewise, when the Verdi cursor time in the active stream is changed by an event, Verdi or *nWave* cursor time is changed.

Sync Active Selection to Waveform

Menu Bar: *Analysis Window*, **View -> Sync Active Selection to Waveform**

This command synchronizes the active transaction/message between the *Transaction/Message Analysis* window and the *nWave* frame as well as between different *Transaction/Message Analysis* windows. When this command is turned *on*, the selected transaction/message in a *Transaction/Message Analysis Window* is reflected in the *nWave* frame as well as other *Transaction/Message Analysis* windows. Likewise, when the active transaction/message in the *nWave* frame is changed, the corresponding transaction/message in the *Transaction/Message Analysis* window is changed. When this command is turned *off*, transactions/messages in the *Transaction/Message Analysis* window and the *nWave* frame work independently. The default value of this command is *off*.

Tools Commands

New Analysis Window

Menu Bar: *Analysis Window*, **Tools -> New Analysis Window**

This command creates a new *Analysis Window* as a new tab in the existing *Transaction/Message Analyzer* frame location. If the FSDB file that is currently open in the source window contains transaction/message streams, then the new window loads the same FSDB file.

Comparison Window

Menu Bar: *Analysis Window*, **Tools** -> **Comparison Window**

This command opens a *Comparison* window where transactions/messages can be compared. The *Comparison* window is opened as a new tab in the existing *Transaction/Message Analyzer* frame location. Alternatively, the **Tools -> Transaction -> Comparison Window** command in *nWave* can be used to invoke the *Comparison* window.

Refer to the *Comparison Window* section for details.

Data Window

Menu Bar: *Analysis Window*, **Tools** -> **Data Window**

Toolbar Icon: 

This command creates a *Data Window* with the address and data of the current cursor row in the *Analysis Window*. The *Data Window* is opened as a new tab in the existing *Transaction/Message Analyzer* frame location.

Refer to the *Data Window* section for details.

Statistics Window

Menu Bar: *Analysis Window*, **Tools** -> **Statistics Window**

Toolbar Icon: 

This command opens the *Perform Statistical Calculation* form where statistics from the transaction/message window can be calculated.

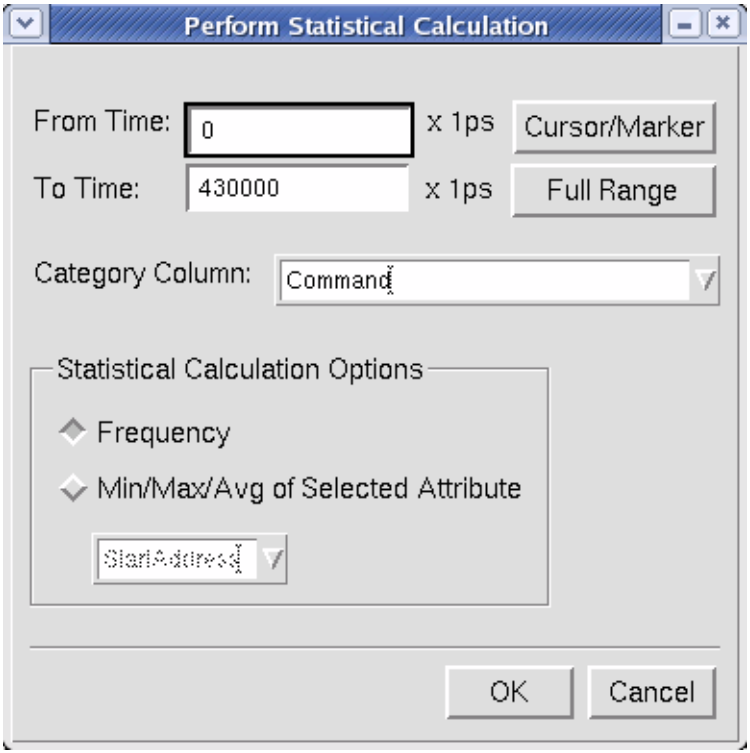


Figure: Perform Statistical Calculation Form

The *Perform Statistical Calculation* form includes the following buttons, options, and fields:

- **From Time/To Time:** Specify the time range in these text fields. The default value is the cursor/marker time of the parent *Analysis Window*.
- **Cursor/Marker:** Click this button to refresh the current cursor/marker time in the *From Time*/*To Time* fields.
- **Full Range:** Click this button to enter the current full FSDB file time range in the *From Time*/*To Time* fields.
- **Category Column:** Specify a column (attribute) from the *Transaction/Message Analysis Window* active stream tab. The function classifies transactions/messages with the same attribute value into the same category. For instance, if *Command* is selected as a category column, five different values (*single_read*, *burst_read*, *single_write*, *burst_write*, and *busy*) are available in the *Command* columns. There are five categories formed and five bars appear in the resulting histogram *Statistics Window*.

- **Statistical Calculation Options**: Specify a calculation method by selecting from either **Frequency** or **Min/Max/Avg of Selected Attribute**.
  - **Frequency:** The **Frequency** method calculates the number of transactions/messages for each category. For instance, the frequency for the category *single_read* is the total number of transactions/messages in the category. If five transactions/messages exist in the entire stream tab with *Command* attribute being *single_read*, then the frequency for the category is *5*.
  - **Min/Max/Avg of Selected Attribute:** The **Min/Max/Avg of Selected Attribute** method calculates the minimum, maximum, and average value of the specified value attribute for each category.
- **OK:** Click this button to open the *Statistics Window* as a new tab in the existing *Transaction/Message Analyzer* frame location.

Refer to the *Statistics Window* section for details.

Relationship Window

Menu Bar: *Analysis Window*, **Tools** -> **Relationship Window**

Toolbar Icon: 

This command opens a *Relationship Window* for viewing the related transactions/messages, browsing the detailed attributes of transactions/messages, configuring columns of interest, sorting by column attribute, filtering relationships, and expanding related transactions/messages.

Refer to the *Relationship Window* section for details.

Customize Menu/Toolbar

Menu Bar: *Analysis Window*, **Tools** -> **Customize Menu/Toolbar**

Refer to the **Tools -> Customize Menu/Toolbar** command in the *nTrace* chapter for details.

Analysis Window Frame Right-Click Options

When the right mouse button is clicked anywhere in the menu, toolbar icon, or frame banner area of the *Transaction/Message Analyzer* frame or standalone window, a configuration option menu is displayed.This menu can be used to configure the available icons and frames.

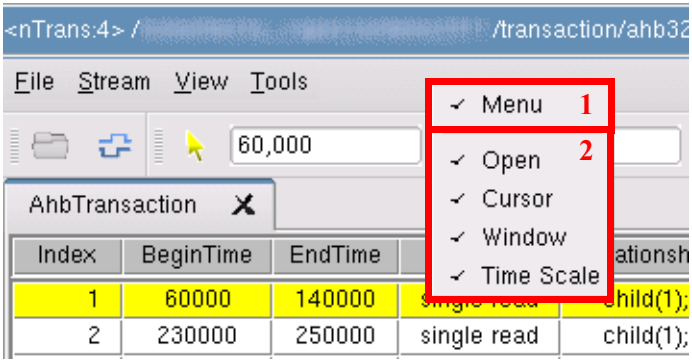


Figure: Configuration Option Menu

The **Menu** option (labeled 1 in the figure above) enables/disables the command menu bar. The options in the bottom section (labeled 2 in the figure above) enable/disable the toolbar icons for different functions.

Analysis Window Column Heading Row Right-Click Commands

Freeze This Column

This command freezes the column in the *Analysis Window*. This is similar to the Excel freeze column function where any columns to the left cannot be scrolled through (only columns to the right can be scrolled).

Un-Freeze Column

This command unfreezes the selected column in the *Analysis Window*, allowing all columns to be scrolled through.

Value Radix

This command sets the radix for a column. The default radix for each column comes from the FSDB file.

Radix Format

The **Radix Format** subcommand provides five address display formats: **Binary**, **Octal**, **Hexadecimal**, **Decimal**, and **ASCII**.

Leading Zeros

This subcommand displays leading zeros for attributes in the *Analysis Window*. The number of leading zeros depends on the format set for the attribute. To set the attribute format, select any attribute and right-click to display the **Radix Format** right-click menu, and choose the desired format for the attribute display in the *Analysis Window*.

Display Prefix

This subcommand displays the prefix for attributes in the *Analysis Window*.

Set Alignment

This command specifies alignment of the selected column. The default value of this command is **Center**.

Alignment Format

The **Alignment Format** subcommand provides three display alignments: **Left**, **Center**, and **Right**.

Analysis Window Toolbar Icons and Fields

Figure: Toolbar Used in Transaction/Message - Analysis Window

The available toolbar icons may be modified. Refer to the *Toolbars* section of the *User Interface* chapter in the *Verdi User Guide and Tutorial* manual for details.

The different toolbar categories and available icons are described below.

Open Category

Open FSDB 

Refer to **File -> Open FSDB** pull-down menu command for details.

Get Stream 

Refer to the **Stream** -> **Get Stream** pull-down menu command for details.

Cursor Category

Cursor Time 

Click this toolbar icon to set the cursor position in the current waveform window.

Marker Time 

Click this toolbar icon to set the marker position in the current waveform window.

Delta Time 

Click this toolbar icon to show the delta time between the cursor and the marker,that is, **Delta = Marker - Cursor**.

Window Category

Relationship Window 

Refer to the **Tools** -> **Relationship Window** pull-down menu command for details.

Data Window 

Refer to the **Tools** -> **Data Window** pull-down menu command for details.

Statistics Window 

Refer to the **Tools** -> **Statistics Window** pull-down menu command for details.

Time Scale Category

Window Time Unit 

This toolbar icon displays the time unit in the *Analysis Window*.

Comparison Window

The *Comparison Window* is displayed in a new tab when the **Tools -> Comparison Window** command is invoked from the *Transaction/Message Analyzer* window or the **Tools -> Transaction -> Comparison Window** command is invoked from the *nWave* frame. The *Comparison Window* frame can become a standalone window by clicking the **Undock** icon on the toolbar.

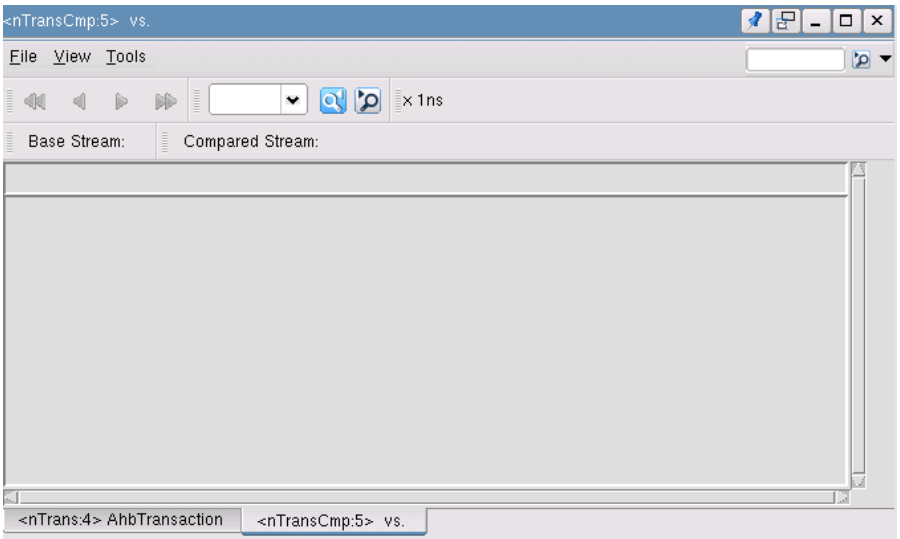


Figure: Blank Comparison Window

The menu bar for the *Comparison Window* is shown below:

Figure: Comparison Window Menu Bar

The pull-down menus for the *Comparison Window* are summarized and explained on the following pages. Refer to the *Icons for Dockable Panes* section for details on the menu bar icons.

File Commands

Compare

Menu Bar: *Comparison Window*, **File** -> **Compare**

This command opens a *Compare Options* form where the transaction/message compare criteria can be set.

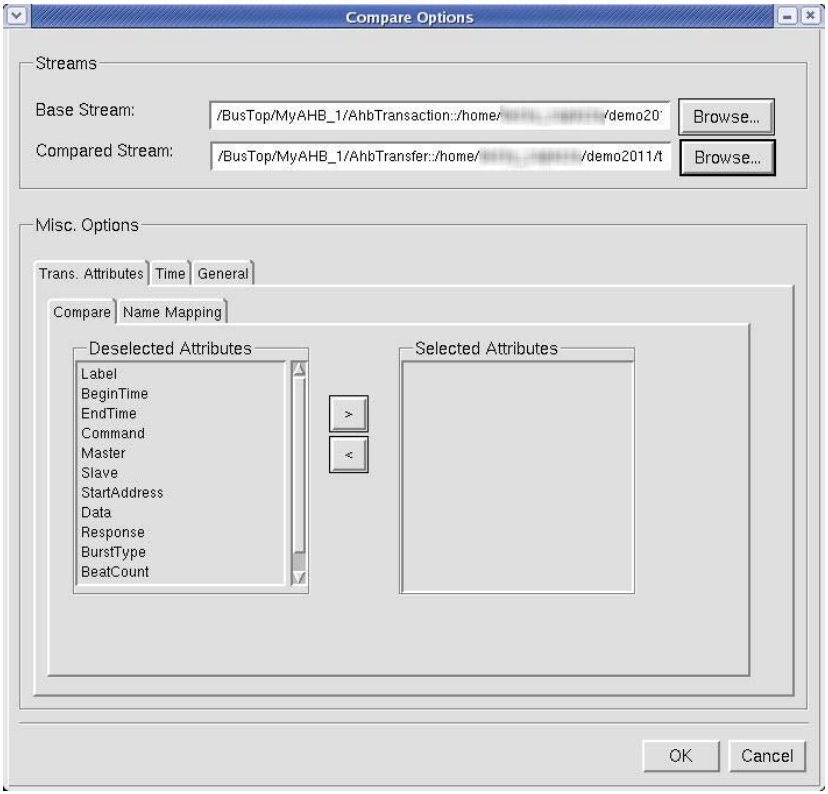


Figure: Compare Options Form - Trans. Attributes Tab -> Compare Tab

The **Streams** section includes the following fields:

- **Base Stream:** This text field specifies the base stream name in the FSDB file. The base stream name can be specified by either typing directly in the text field or by clicking the **Browse** button and selecting the stream name in the *Select Stream* form.
- **Compared Stream:** This text field specifies the stream name to be compared to in the FSDB file. The stream name can be specified by either typing directly in the text field or by clicking the **Browse** button and selecting the stream name in the *Select Stream* form.

The **Misc. Options** section includes the following tabs:

Trans. Attributes Tab

The following fields and buttons are available:

- **Compare:** Use this tab to specify which attributes in the base stream are to be compared.
  - **Deselected Attributes:** This column lists the attributes that are not to be compared.
  - **Selected Attributes:** This column lists the attributes that are to be compared.

  Use the > and < buttons to move the selected attributes to compare from *Deselected Attributes* to *Selected Attributes*, and vice versa. Only one attribute can be selected at a time.
- **Name Mapping:** Use this tab to rearrange name mapping of each attribute in the compared stream to the attribute in the base stream.

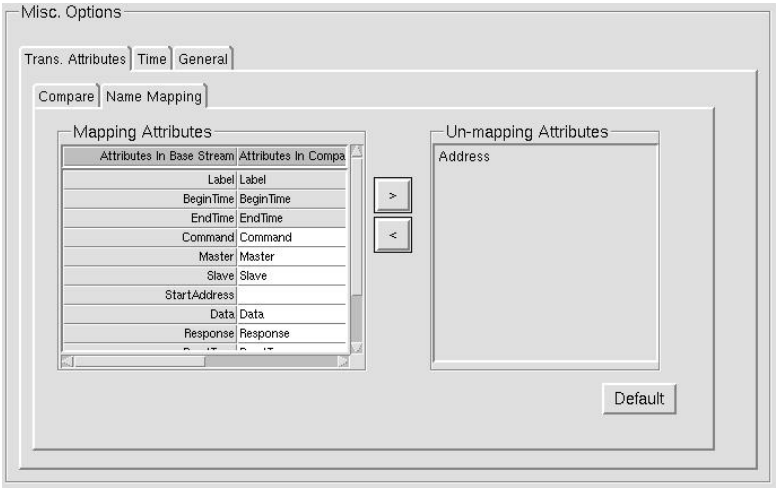


Figure: Compare Options Form - Trans. Attributes Tab - Name Mapping Tab

  - **Mapping Attributes:** This column lists the attributes being mapped. Select the attribute in the white cells and use > and < buttons to move the selected attributes from *Mapping Attributes* to *Unmapping Attributes*, and vice versa. The gray cells cannot be selected.
  - **Unmapping Attributes:** This column lists the attributes without mapping. Use > and < buttons to move the selected attributes from *Unmapping Attributes* to *Mapping Attributes*, and vice versa.

Click the **Default** button to reset all attributes back to the attribute's default name mapping.

Time Tab

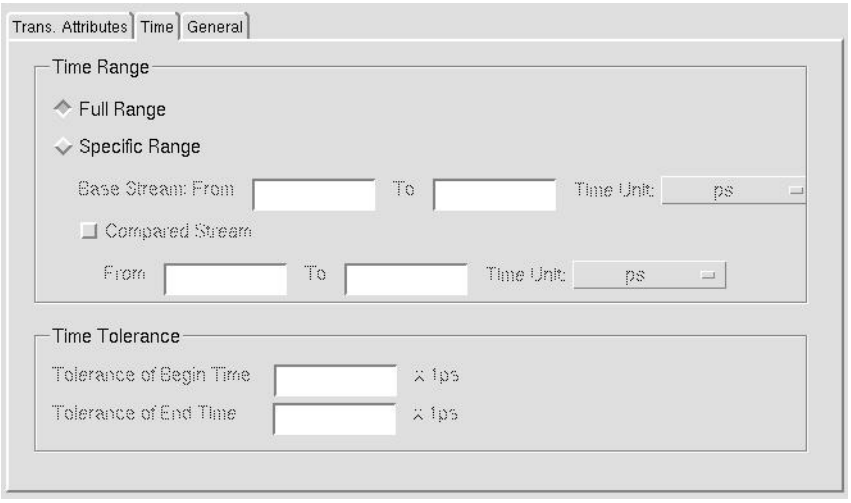


Figure: Compare Options Form - Time Tab

The **Time Range** section includes the following fields, options, and buttons:

Only one of the following options can be selected at a time.

- **Full Range:** When this option is turned *on*, the full transaction time range is used for comparison. The default value of this option is *on*.
- **Specific Range:** When this option is turned *on*, the partial transaction time range in the stream is used for comparison. The default value of this option is *off*.
  - **Base Stream:** Specify the base stream time range to be used for comparison in **From** and **To** text fields. The time unit can be specified using the following options: *s, ms, us, ns, ps, fs, Ks, Ms,* and *Gs*.
  - **Compared Stream:** Turn *on* this option to specify a different time range to use for the stream being compared in **From** and **To** text fields. When this option is turned *off*, the compared stream uses the same time range as the base stream. The time unit can be specified using the following options: *s, ms, us, ns, ps, fs, Ks, Ms,* and *Gs*.

The **Time Tolerance** section includes the following fields:

- **Tolerance of Begin Time:** Specify the time tolerance for the comparison begin time. This text field is available only when the **BeginTime** attribute is selected to be compared. The tolerance time unit is the same as the file time unit of the base stream.
- **Tolerance of End Time:** Specify the time tolerance for the comparison end time. This text field is available only when the **EndTime** attribute is selected to be compared. The tolerance time unit is the same as the file time unit of the base stream.

General Tab

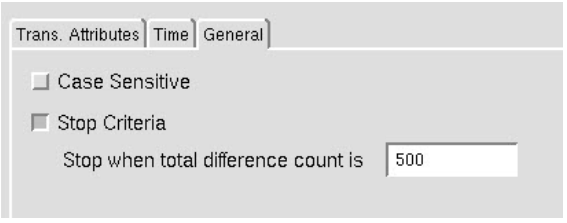


Figure: Compare Options Form - General Tab

The **General** tab includes the following options:

- **Case Sensitive:** When this option is turned *on*, attributes are compared with case-sensitivity. When the this option is turned *off*, the attributes are compared without case-sensitivity. The default value of this option is *off*.
- **Stop Criteria:** When this option is turned *on*, the maximum number of errors allowed before stopping the comparison is specified. The default value of this option is *on*, and the default difference count is *500*.

After specifying the compare options, click the **OK** button to save the settings and close the *Compare Options* form, or click the **Cancel** button to close the *Compare Options* form without making any changes.

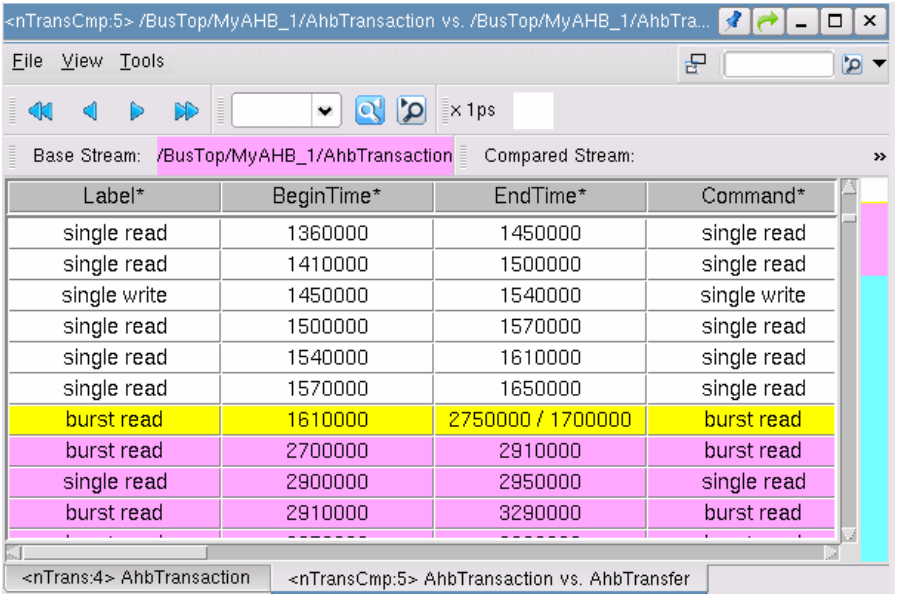


Figure: Transaction/Message Comparison Window

In the *Comparison* window, the column headings marked with * indicate the attributes selected for comparison in the *Compare* tab of the *Compare Options* form. The blue row represents the selected transaction/message (a transaction/message may be selected with a left-click anywhere along the row; only one transaction/message can be selected at a time). The yellow rows indicate the base stream and the compared stream have different attribute values.

The *Difference Map* on the right side of the *Comparison* window displays the summary of different mismatch areas. Each of the difference areas is marked in a different color. For example, differences where only the base stream has the transaction/message are marked with pink color; differences where only the compared stream has the transaction/message are marked with cyan color.

Save to File

Menu Bar: *Comparison Window*, **File -> Save to File**

This command opens a *Save As* form where the directory structure can be viewed and a text file name specified to save the *Comparison* window contents to.

Close Window

Menu Bar: *Comparison Window*, **File** -> **Close Window**

This command closes the current *Comparison* window.

View Commands

Find

Menu Bar: *Comparison Window*, **View** -> **Find**

This command opens the *Find String* form where text can be specified to be searched for in the current *Comparison* window.

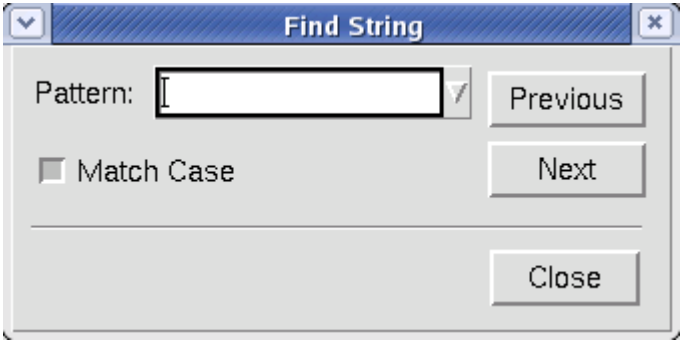


Figure: Find String Form

This form includes the following field, option, and buttons:

- **Pattern:** Type the text in the *Pattern* text field to find the string in the *Comparison* window. A partial string can be entered. Wildcard characters are not supported.
- **Previous:** Click this button or the  toolbar icon to find the previous matched string.
- **Next:** Click this button or the  toolbar icon to find the next matched string.
- **Match Case:** When this option is turned *on*, case-sensitivity is considered (such as, ‘b’ finds ‘b’ only) during the search operation. When this option is turned *off*, case-sensitivity is ignored (such as, ‘a’ finds ‘a’ and ‘A’) during the filter operation. The default value of this option is *on*.

Configure Columns

Menu Bar: *Comparison Window*, **View -> Configure Columns**

Refer to the **View -> Configure Columns** pull-down menu in the *Analysis Window* section for details.

Jump to Previous Difference

Menu Bar: *Comparison Window*, **View -> Jump to Previous Difference**

Toolbar Icon: 

Bind Key P

This command moves the cursor to the previous mismatch area in the *Comparison Window*.

Jump to Next Difference

Menu Bar: *Comparison Window*, **View -> Jump to Next Difference**

Toolbar Icon: 

Bind Key N

This command moves the cursor to the next mismatch area in the *Comparison Window*.

Jump to First Difference

Menu Bar: *Comparison Window*, **View -> Jump to First Difference**

Toolbar Icon: 

Bind Key F

This command moves the cursor to the first mismatch area in the *Comparison Window*.

Jump to Last Difference

Menu Bar: *Comparison Window*, **View -> Jump to Last Difference**

Toolbar Icon: 

Bind Key L

This command moves the cursor to the last mismatch area in the *Comparison Window*.

Tools Commands

Customize Menu/Toolbar

Refer to the **Tools ->** **Customize Menu/Toolbar** command in the *nTrace* chapter for details.

Comparison Window Frame Right-Click Options

When the right mouse button is clicked anywhere in the menu, toolbar icon, or frame banner area of the *Comparison Window* frame or standalone window, a configuration option menu is displayed.This menu can be used to configure the available icons and frames.

The **Menu** option enables/disables the command menu bar. The options in the bottom section enable/disable the toolbar icons for different functions.

Comparison Window Toolbar Icons and Fields

The available toolbar icons may be modified. Refer to the *Toolbars* section of the *User Interface* chapter in the *Verdi User Guide and Tutorial* manual for details.

The different toolbar categories and available icons are described below.

Difference Category

Jump to First Difference 

Refer to **View ->** **Jump to First Difference** pull-down menu command for details.

Jump to Previous Difference 

Refer to **View ->** **Jump to Previous Difference** pull-down menu command for details.

Jump to Next Difference 

Refer to **View -> Jump to Next Difference** pull-down menu command for details.

Jump to Last Difference 

Refer to **View -> Jump to Last Difference** pull-down menu command for details.

Find Utility Category

Find 

Refer to **View -> Find** pull-down menu command for details.

Find Previous 

Click this icon to find the previous matched string.

Find Next 

Click this icon to find the next matched string.

Window Time Unit Category

Window Time Unit 

This toolbar icon displays the time unit in the *Comparison Window*.

Base Stream Info and Compared Stream Info Categories

These categories display the name of the base stream and compared stream.

Data Window

The *Data Window* is displayed in a new tab when the **Tools -> Data Window** command is invoked from the *Analysis Window*. The *Data Window* frame becomes a standalone window if the **Undock** icon on the toolbar is clicked.

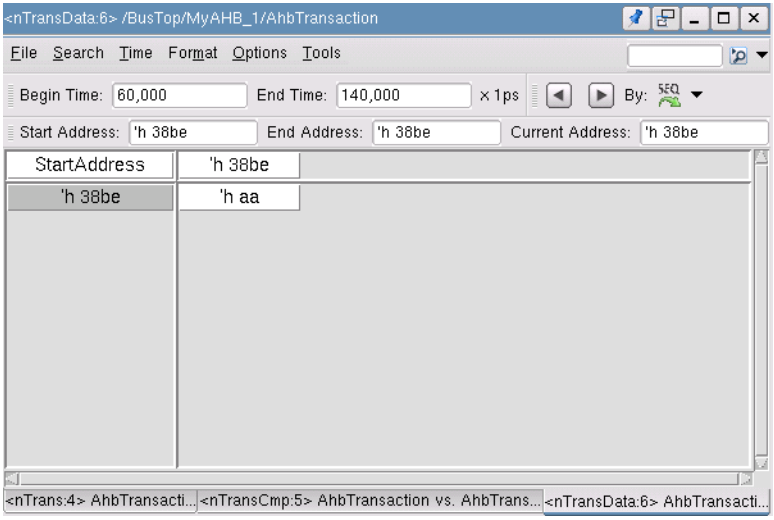


Figure: Data Window

The menu bar for the *Data Window* is shown below:

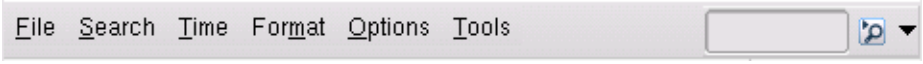


Figure: Data Window Menu Bar

The pull-down menus for the *Data Window* are summarized and explained on the following pages. Refer to the *Icons for Dockable Panes* section for details on the menu bar icons.

File Commands

Save to File

Menu Bar: *Data Window*, **File** -> **Save to File**

This command opens a *Save As* form where the directory structure can be viewed and a text file name specified for saving the *Data Window* contents to.

Exit

Menu Bar: *Data Window*, **File** -> **Exit**

This command closes the *Data Window*.

Search Commands

Pattern

Menu Bar: *Data Window*, **Search** -> **Pattern**

This command finds an address or a data pattern in the current transaction/message for the specified string.

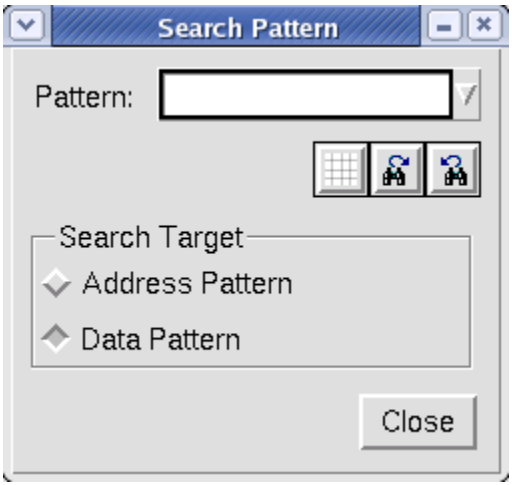


Figure: Search Pattern Form

This form includes the following field, buttons and options:

- **Pattern:** Specify the characters to be searched for in this text field and then click **Clear Pattern**, **Find Next**, or **Find Previous** buttons to perform the search. Wildcard characters are supported.
- **Search Target:** Specify the search target for the pattern by selecting either **Address Pattern** or **Data Pattern**.

Time Commands

Set Dump Type

Menu Bar: *Data Window*, **Time** -> **Set Dump Type**

Specify the method to locate the previous/next transaction/message with the **Prev Dump** and **Next Dump** toolbar icons. Data in the new transaction/message is refreshed in the *Data Window*. The following dump types are available:

- **By Sequence**  : Trace previous/next transaction/message in sequence in the *Data* category in the *Transaction/Message Analyzer* window.
- **By Read**  : Trace the previous/next transaction/message based on the *Command* category in the *Transaction/Message Analyzer* window with the value *READ*.
- **By Write**  : Trace the previous/next transaction/message based on the *Command* category in the *Transaction/Message Analyzer* window with the value *WRITE*.
- **By Read/Write**  : Trace the previous/next transaction/message based on the *Command* category in the *Transaction/Message Analyzer* window with the value *READ* or *WRITE*.

Prev Dump

Menu Bar: *Data Window*, **Time** -> **Prev Dump**

Toolbar Icon: 

This command jumps to the previous transaction/message and updates the time.

Next Dump

Menu Bar: *Data Window*, **Time** -> **Next Dump**

Toolbar Icon: 

This command jumps to the next transaction/message and updates the time.

Sync Cursor Time

Menu Bar: *Data Window*, **Time** -> **Sync Cursor Time**

This toggle command turns the synchronization of the current time in the *Analysis Window* and the *Data Window on* and *off*. When this command is turned *off*, the *Data Window* works independently. When this command is turned *on*, the time is set according to the current cursor time of the active window.

If the *Analysis Window* enables its **Sync Cursor Time** command to link to the *nWave* frame, then these three windows are synchronized.

Format Commands

Address Radix

This command includes the following address formats. The default value of this command is *Decimal*.

Octal

Menu Bar: *Data Window*, **Format -> Address Radix -> Octal**

This command displays the address radix in *Octal* format in the *Data Window*.

Hexadecimal

Menu Bar: *Data Window*, **Format -> Address Radix -> Hexadecimal**

This command displays the address radix in *Hexadecimal* format in the *Data Window*.

Decimal

Menu Bar: *Data Window*, **Format -> Address Radix -> Decimal**

This command displays the address radix in *Decimal* format in the *Data Window*.

Leading Zeros

Menu Bar: *Data Window*, **Format -> Address Radix -> Leading Zeros**

When this toggle command is *on*, leading zeros are displayed in the address category. The number of leading zeros depends on the format set for the address category. The default value of this command is *off*.

Display Prefix

Menu Bar: *Data Window*, **Format** -> **Address Radix** -> **Display Prefix**

When this toggle command is *on*, the prefix of the radix format in the address category is displayed. The default value of this command is *on*.

Data Radix

This command includes the following radix formats. The default value of this command is *Decimal*.

Binary

Menu Bar: *Data Window*, **Format** -> **Data Radix** -> **Binary**

This command displays the data radix in *Binary* format in the *Data Window*.

Octal

Menu Bar: *Data Window*, **Format** -> **Data Radix** -> **Octal**

This command displays the data radix in *Octal* format in the *Data Window*.

Hexadecimal

Menu Bar: *Data Window*, **Format** -> **Data Radix** -> **Hexadecimal**

This command displays the data radix in *Hexadecimal* format in the *Data Window*.

Decimal

Menu Bar: *Data Window*, **Format** -> **Data Radix** -> **Decimal**

This command displays the data radix in *Decimal* format in the *Data Window*.

ASCII

Menu Bar: *Data Window*, **Format** -> **Data Radix** -> **ASCII**

This command displays the data radix in *ASCII* format in the *Data Window*.

Leading Zeros

Menu Bar: *Data Window*, **Format** -> **Data Radix** -> **Leading Zeros**

When this toggle command is *on*, leading zeros are displayed in the data category. The number of leading zeros depends on the format set for the data category. The default value of this command is *off*.

Display Prefix

Menu Bar: *Data Window*, **Format** -> **Data Radix** -> **Display Prefix**

When this toggle command is *on*, the prefix of the radix format in the data category is displayed. The default value of this command is *on*.

Options Command

Display

Menu Bar: *Data Window*, **Options** -> **Display**

Specify how many words can be displayed in one row. If the *Words Shown in One Row* field is null or 0, all data is displayed in a single row. The default value of this command is *3*.

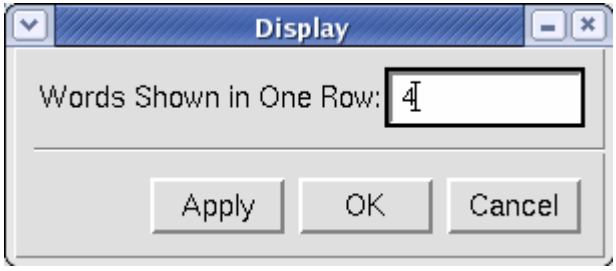


Figure: Display Rule Form

Tools Commands

Customize Menu/Toolbar

Refer to the **Tools -> Customize Menu/Toolbar** command in the *nTrace* chapter for details.

Data Window Frame Right-Click Options

When the right mouse button is clicked anywhere in menu, toolbar icon, or frame banner area of the *Data Window* frame or standalone window, a configuration option menu is displayed.This menu can be used to configure the available icons and frames.

The **Menu** option enables/disables the command menu bar. The options in the bottom section enable/disable the toolbar icons for different functions.

Data Window Toolbar Icons and Fields

The available toolbar icons may be modified. Refer to the *Toolbars* section of the *User Interface* chapter in the *Verdi User Guide and Tutorial* manual for details.

The different toolbar categories and available icons are described below.

Transaction Time Category

This category displays the begin time, end time, and time unit for the selected transaction. The fields cannot be edited.

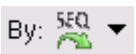
Search Dump Category

Prev Dump 

Refer to the **Time -> Prev Dump** pull-down menu for details.

Next Dump 

Refer to the **Time -> Next Dump** pull-down menu for details.

Search By 

Refer to the **Time -> Set Dump Type** pull-down menu for details.

Data Address Category

This category displays the start, end, and current address of the selected transaction. The fields cannot be edited.

Statistics Window

The *Statistics Window* is displayed in a new tab when the **Tools -> Statistics Window** command is invoked from the *Analysis Window*. The *Statistics Window* frame can become a standalone window by clicking the **Undock** icon on the toolbar.

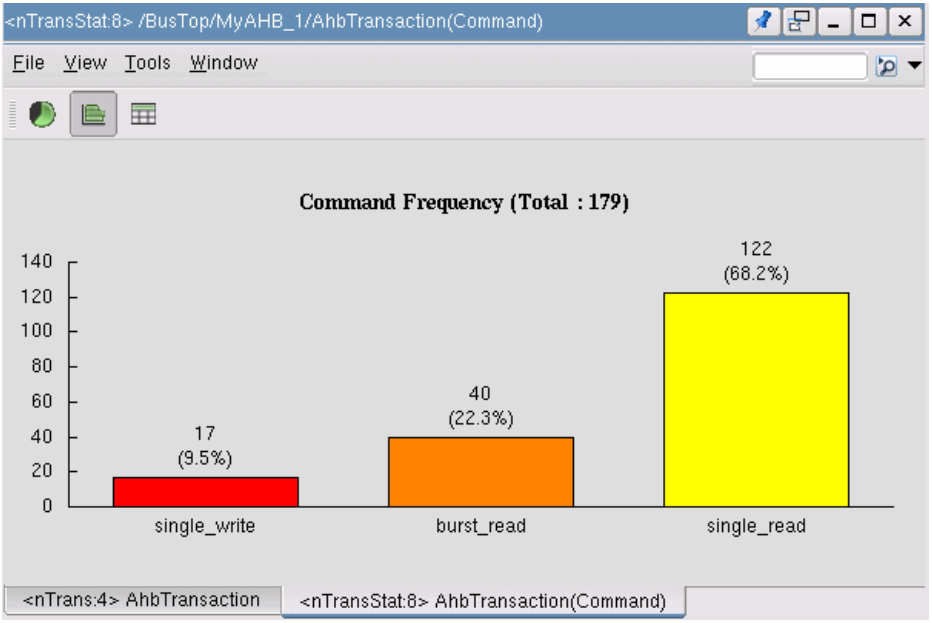


Figure: Statistics Window - Bar Graph

Three display methods are available to view the statistical information: **Pie Chart**, **Bar Graph**, and **Table**. The default display method is **Bar Graph**.

The title of the statistical information consists of three elements:

- The attribute name specified in the **Category Column** of the *Perform Statistical Calculation* form.
- The calculation method selected in the **Statistical Calculation Options** of the *Perform Statistical Calculation* form.
- The total number of transactions/messages in the stream.

Depending on the method selected to display the statistical information, value attributes, quantities of value attributes, and the percentages of value attributes are seen. Statistical information may be selected with a left-click anywhere along the pie slice/bar/row.

The menu bar for the *Statistics Window* is shown below:

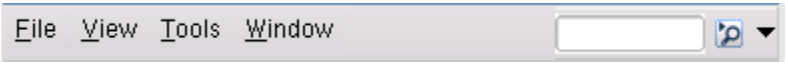


Figure: Statistics Window Menu Bar

The pull-down menus for the *Statistics Window* are summarized and explained on the following pages. Refer to the *Icons for Dockable Panes* section for details on the menu bar icons.

File Commands

Export

Menu Bar: *Statistics Window*, **File** -> **Export**

The statistical information in the *Statistics Window* can be dumped into a text file.

Capture Window

Menu Bar: *Statistics Window*, **File** -> **Window**

Refer to the **File -> Capture Window** command description in the *nTrace* chapter for details.

Close

Menu Bar: *Statistics Window*, **File** -> **Close**

This command closes the *Statistics Window*.

View Commands

Pie Chart

Menu Bar: *Statistics Window*, **View** -> **Pie Chart**

Toolbar Icon: 

This command displays the transaction/message result data in the *Statistics Window* as a pie chart.

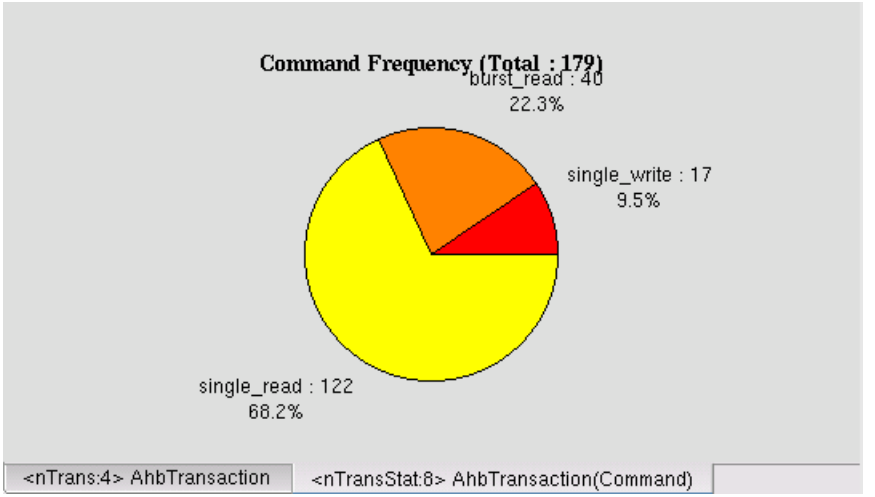


Figure: Statistics Window in Pie Chart

Bar Graph

Menu Bar: *Statistics Window*, **View** -> **Bar Graph**

Toolbar Icon:

This command displays the transaction/message result data in the *Statistics Window* as a bar graph.

Table

Menu Bar: *Statistics Window*, **View** -> **Table**

Toolbar Icon:

This command displays the transaction/message result data in the *Statistics Window* as a text table.

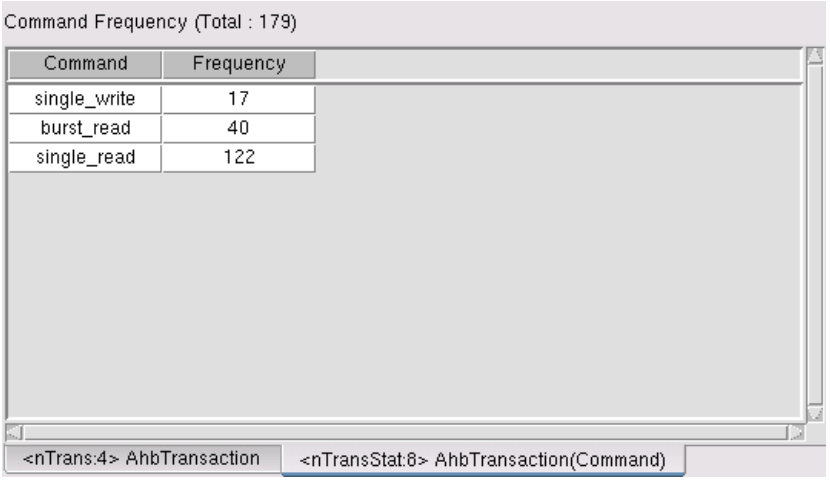


Figure: Statistics Window in Table

In the text table format, the width of each column may be changed by selecting the vertical line between column headers and dragging-left. The columns in the table consist of either attribute name and **Frequency** or attribute name, **Min**, **Max**, and **Avg**. The column headings are summarized as follows:

- **Attribute Name:** Displays the category information of the selected attribute.
- **Frequency:** Displays the total occurrences of the corresponding category.
- **Min:** Displays the minimum value of the corresponding value attribute for each category.
- **Max:** Displays the maximum value of the corresponding value attribute for each category.
- **Avg:** Displays the average value of the corresponding value attribute for each category.

Tools Commands

Customize Menu/Toolbar

Refer to the **Tools -> Customize Menu/Toolbar** command in the *nTrace* chapter for details.

Window Command

Duplicate Window as

Menu Bar: *Statistics Window*, **Window** -> **Duplicate Window as**

A duplicate *Statistics Window* can be opened with the **Pie Chart**, **Bar Graph**, or **Table** format.

Statistics Window Frame Right-Click Options

When the right mouse button is clicked anywhere in the menu, toolbar icon, or frame banner area of the *Statistics Window* frame or standalone window, a configuration option menu is displayed.This menu can be used to configure the available icons and frames.

The **Menu** option enables/disables the command menu bar. The options in the bottom section enable/disable the toolbar icons for different functions.

Statistics Window Toolbar Icons and Fields

The available toolbar icons may be modified. Refer to the *Toolbars* section of the *User Interface* chapter in the *Verdi User Guide and Tutorial* manual for details.

The available icons for **View Category** are described below.

Pie Chart 
Refer to the **View** -> **Pie Chart** pull-down menu command for details.

Bar Graph 
Refer to the **View** -> **Bar Graph** pull-down menu command for details.

Table 
Refer to the **View** -> **Table** pull-down menu command for details.

Relationship Window

The *Relationship Window* is displayed as a new tab when the **Tools -> Relationship Window** command is invoked from the *Analysis Window*. The *Relationship Window* is opened as a new tab in the existing *Transaction/Message Analyzer* frame location.The *Relationship Window* frame can become a standalone window by clicking the **Undock** icon on the toolbar.

The *Relationship Window* can interact with the *Analysis Window* through a drag and drop action or by invoking the **Go To Transaction** command in the *Relationship Window*.

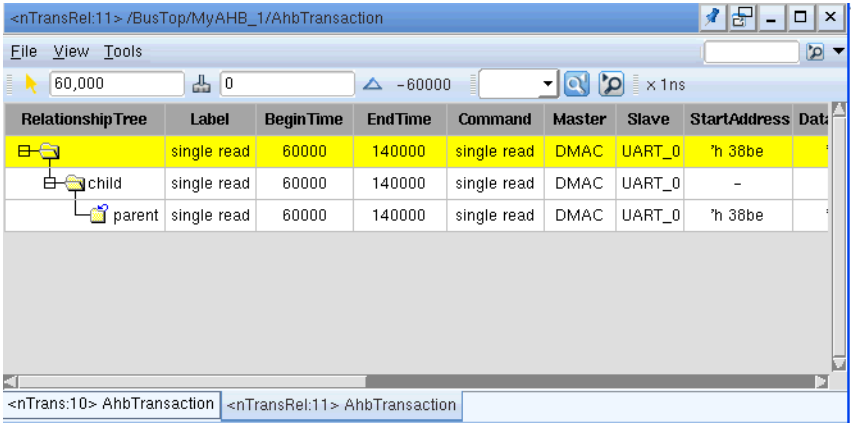


Figure: Relationship Window

A transaction/message may be selected with a left-click anywhere along the row. The width of each column can be adjusted by selecting the vertical line between column headers and dragging-left. The columns can be sorted by clicking the column headers. The *Relationship Window* includes five standard columns: *Relationship Tree*, *Label*, *StreamName#Index*, *BeginTime*, and *EndTime*.

The standard column headings are summarized as follows:

- **Relationship Tree:** This column displays the related transactions/messages in a tree format. Each node is named with the relationship to the previous transaction/message, such as *child*. The relationship tree is initially expanded two levels when invoking the *Relationship Window*.
- **Label:** This column displays the label information of the selected transaction/message.
- **StreamName#Index**: This column displays the transaction/message index. If the stream in the parent *Analysis Window* is open, then the transaction/

message index is shown. When the stream is closed in the parent *Analysis Window or* the stream is not opened yet, the transaction/message index is shown as '-'.

- **BeginTime:** This column corresponds to the begin time of the selected transaction/message.
- **EndTime:** This column corresponds to the end time of the selected transaction/message.

Additional column headings may be included as defined by attributes in the transactions/messages.

The menu bar for the *Relationship Window* is shown below:

Figure: Relationship Window Menu Bar

The pull-down menus for the *Relationship Window* are summarized and explained on the following pages. Refer to the *Icons for Dockable Panes* section for details on the menu bar icons.

File Commands

Close Window

Menu Bar: *Relationship Window*, **File** -> **Close Window**

This command closes the *Relationship Window*.

View Commands

Find

Menu Bar: *Relationship Window*, **View** -> **Find**

Refer to **View -> Find** pull-down menu in the *Analysis Window* section for details.

Relationship Filter

Menu Bar: *Relationship Window*, **View** -> **Relationship Filter**

This command opens the *Relationship Filter* form where the relationships can be filtered in the *Relationship Window*.

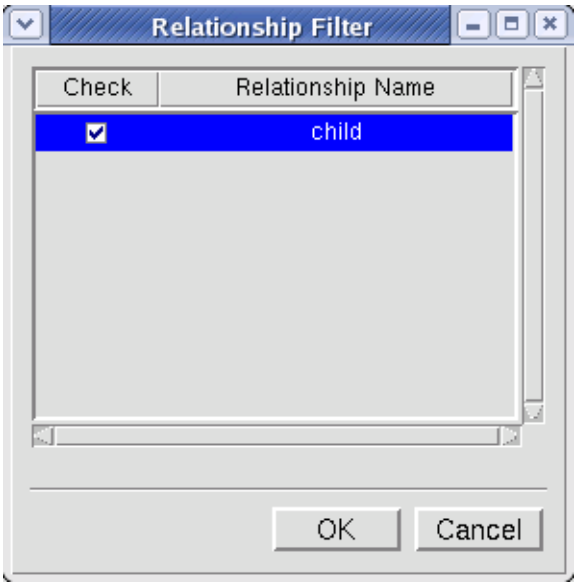


Figure: Relationship Filter Form

The *Relationship Filter* form includes the following table headings:

- **Check:** Check/select the relationships to filter out. All of the relationship names displayed in the current *Relationship Window* are checked as default.
- **Relationship Name:** Lists all the relationship names of the top transaction/message.

Click the **OK** button to remove the relationships that meet the filter criteria in the *Relationship Window*.

Configure Columns

Menu Bar: *Relationship Window*, **View** -> **Configure Columns**

Refer to the **View -> Configure Columns** pull-down menu in the *Analysis Window* section for details.

Set as Top

Menu Bar: *Relationship Window*, **View** -> **Set as Top**

This command sets the current selected transaction/message as the top of the relationship tree. The *Relationship Window* then re-generates the relationship tree.

Go to Transaction

Menu Bar: *Relationship Window*, **View** -> **Go to Transaction**

This command jumps to and highlights the specified transaction/message in the parent *Analysis Window*. If the stream is not displayed in the *Analysis Window* yet, a message form appears asking whether to open the stream. If the stream is open and the transaction/message is filtered out of the *Analysis Window*, a message form appears asking whether to restore the stream. The *Analysis Window* becomes the active window and the active stream switches to the stream for the specified transaction/message.

Expand Node

Menu Bar: *Relationship Window*, **View** -> **Expand Node**

This command expands the related transactions/messages of the active transaction/message by the level specified. The default level is *3*.

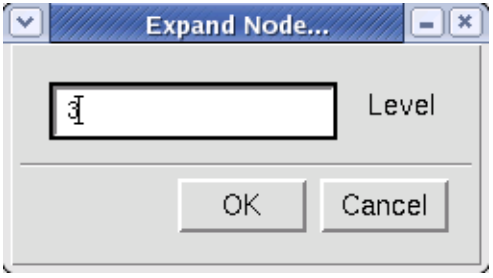


Figure: Expand Node Form

Hide Recursive Nodes

Menu Bar: *Relationship Window*, **View** -> **Hide Recursive Nodes**

When this command is toggled *on*, all recursive node(s) in the *Relationship Window* are hidden. When this command is toggled *off*, all recursive node(s) are displayed.

Tools Commands

Customize Menu/Toolbar

Refer to the **Tools -> Customize Menu/Toolbar** command in the *nTrace* chapter for details.

Relationship Window Frame Right-Click Options

When the right mouse button is clicked anywhere in menu, toolbar icon, or frame banner area of the *Relationship Window* frame or standalone window, a configuration option menu is displayed.This menu can be used to configure the available icons and frames.

The **Menu** option enables/disables the command menu bar. The options in the bottom section enable/disable the toolbar icons for different functions.

Relationship Tree Column Right-Click Commands

When the right mouse button is clicked anywhere in the column below the heading row, the following commands are available.

Set as Top

Refer to the **View -> Set as Top** pull-down menu for details.

Go to Transaction

Refer to the **View -> Go to Transaction** pull-down menu for details.

Expand Node

Refer to the **View -> Expand Node** pull-down menu for details.

Relationship Window Toolbar Icons and Fields

The available toolbar icons may be modified. Refer to the *Toolbars* section of the *User Interface* chapter in the *Verdi User Guide and Tutorial* manual for details.

The different toolbar categories and available icons are described below.

Cursor/Marker Category

Cursor Time 

Click this toolbar icon to display the cursor time for the selected transaction in the *Relationship Window*.

Marker Time 

Click this toolbar icon to display the marker time for the selected transaction in the *Relationship Window*.

Delta Time 

Click this toolbar icon to show the delta time between the cursor and the marker.

Search Category

Find 

Refer to **View -> Find** pull-down menu in the *Analysis Window* section for details.

Find Previous 

Click this icon to find the previous matched string.

Find Next 

Click this icon to find the next matched string.

Window Time Unit Category

Window Time Unit 

This toolbar icon displays the time unit in the *Comparison Window*.

Clock Analyzer

Overview

The frames associated with *Clock Analyzer* are displayed when the **Clock Analyzer** commands or the **Clock Tree** commands are invoked from *nTrace* or *nSchema* menu bars. The *Clock Analyzer* windows are docked to the same frame location as *Message* and *nWave* frames as a new tab.

The following windows are associated with *Clock Analyzer* and can be invoked with the listed commands:

- *Clock Domains Window*
  - The *nTrace* menu commands to invoke the *Clock Domains* window :
    - **Tools -> Clock Analyzer -> Extract Clock Information**
    - **Tools -> Clock Analyzer -> List Clock Domains**
  - The *nSchema* menu commands to invoke the *Clock Domains* window:
    - **Trace -> Clock Analyzer -> Extract Clock Information**
    - **Trace -> Clock Analyzer -> List Clock Domains**
- *Clock Tree Browser Window*
  - The *nTrace* menu commands to invoke the *Clock Tree Browser* window:
    - **Tools -> New Schematic from Source -> Clock Tree**
    - **Tools -> Clock Analyzer -> List Specified Clock Domains**
  - The *nSchema* menu commands to invoke the *Clock Tree Browser* window:
    - **Trace -> Clock Analyzer -> List Specified Clock Domains**
    - **Tools -> New Schematic -> Clock Tree**
  - The *Clock Domains* window menu command to invoke the *Clock Tree Browser* window:
    - **Tools -> New Clock Tree Browser**
- *Crossing Paths Window*
  - The *nTrace* menu commands to invoke the *Crossing Paths* window:
    - **Tools -> Clock Analyzer -> Check Crossing Paths**
    - **Tools -> Clock Analyzer -> List Crossing Paths**

- The *nSchema* menu commands to invoke the *Crossing Paths* window:
  - **Trace -> Clock Analyzer -> Check Crossing Paths**
  - **Trace -> Clock Analyzer -> List Crossing Paths**
- The *Clock Domains* menu commands to invoke the *Crossing Paths* window:
  - **Tools -> Check Crossing Paths**
  - **Tools -> List Crossing Paths**

NOTE: The Clock Domain window does not support global font.

In addition, there are several forms associated with these windows:

- *Clock Extraction Settings Form*
  - The *nTrace* menu commands to invoke the *Clock Extraction Settings* form:
    - **Tools -> New Schematic from Source -> Clock Tree**
    - **Tools -> Clock Analyzer -> Extract Clock Information**
  - The *nSchema* menu commands to invoke the *Clock Extraction Settings* form:
    - **Trace -> Clock Analyzer -> Extract Clock Information**
    - **Tools -> New Schematic -> Clock Tree**
  - The *Clock Domains* window menu command to invoke the *Clock Extraction Settings* form:
    - **Settings toolbar icon**
- *Highlight Clock Domains Form*
  - The *nTrace* menu command to invoke the *Highlight Clock Domains* form:
    - **Tools -> Clock Analyzer -> Highlight Clock Domains**
  - The *nSchema* menu command to invoke the *Highlight Clock Domains* form:
    - **Trace -> Clock Analyzer -> Highlight Clock Domains**
  - The *Clock Domains* window menu command to invoke the *Highlight Clock Domains* form:
    - **Tools -> Highlight Clock Domains**
- *Check Crossing Paths Form*
  - The *nTrace* menu command to invoke the *Check Crossing Paths* form:
    - **Tools -> Clock Analyzer -> Check Crossing Paths**

- The *nSchema* menu command to invoke the *Check Crossing Paths* form:
  - **Trace -> Clock Analyzer -> Check Crossing Paths**
- The *Clock Domains* window menu command to invoke the *Check Crossing Paths* form:
  - **Tools -> Check Crossing Paths**

Refer to the appropriate section in this chapter for details.

Clock Extraction Settings Form

The *Clock Extraction Settings* form is used to specify settings for extraction of all the clock domains in a design. This form can open *Clock Tree Browser Window* and/or *Clock Domains Window*, depending on the specifications in the **Show Clock Domains** option and/or the **Show Details of Specified Domains (Clock Tree)** option in the *Clock Extraction Settings* form. The two methods with different default settings/specifications are as follows:

- Invoke the **Tools -> New Schematic from Source -> Clock Tree** command from *nTrace* or the **Tools -> New Schematic -> Clock Tree** command from *nSchema*. Then, in the **Options** tab of the *Clock Extraction Settings* form, turn *off* the **Show Clock Domains** and the **Entire Design** options and turn *on* the **Show Details of Specified Clock Domains (Clock Tree)** option.

  Click the **OK** button in the *Clock Extraction Settings* form; the *Clock Extraction Settings* form is closed and the extracted clock tree specified in the **Clock Sources** table of the **Clock Sources** tab of the *Clock Extraction Settings* form is displayed in the *Clock Tree Browser* window.

- Invoke the **Tools -> Clock Analyzer -> Extract Clock Information** command from *nTrace* or the **Trace -> Clock Analyzer -> Extract Clock Information** command from *nSchema*. Then, in the **Options** tab of the *Clock Extraction Settings* form, turn *on* the **Show Clock Domains** option and turn *off* the **Show Details of Specified Clock Domains (Clock Tree)** option.

  If the **Entire Design** option is turned *on*, the possible clock sources for the entire design are obtained and displayed in the *Clock Tree Browser* window.

  Click the **OK** button in the *Clock Extraction Settings* form; the *Clock Extraction Settings* form is closed and the *Clock Domains* window is opened that lists all currently loaded or extracted clock domains.

The *Clock Extraction Settings* form can also be opened by clicking the **Settings** toolbar icon in the *Clock Domains* window.

The *Clock Extraction Settings* form includes Clock Sources Tab, CTS Attributes Tab, Gated Clock Cell Tab, Options Tab, Import Button, and Export Button. The tabs and buttons are explained in the following pages.

Clock Sources Tab

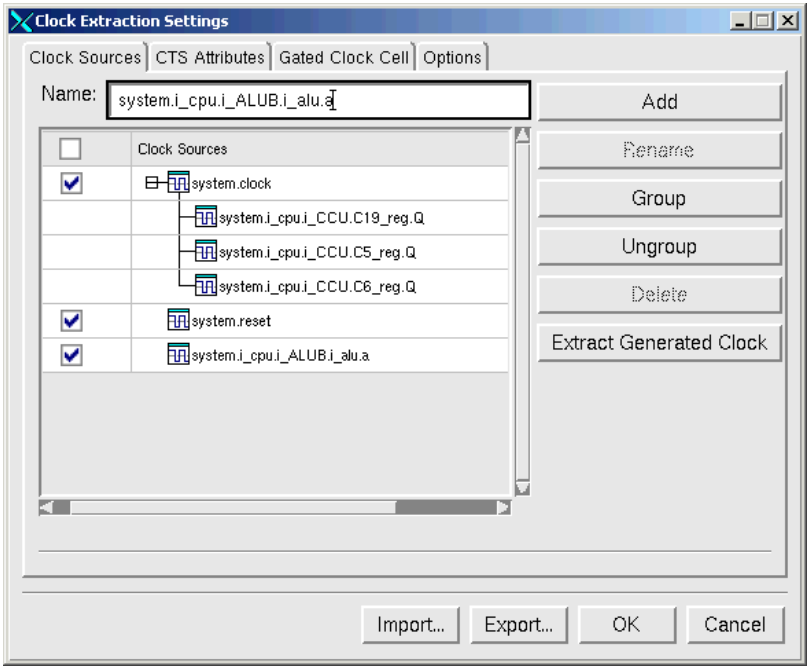


Figure: Clock Extraction Settings Form - Clock Sources Tab

The *Clock Sources* tab includes the following field and options:

- **Name:** The clock source can also be specified by typing in this text field.
- **Clock Sources:** In this table, instance pin(s), instance(s), or signal(s) can be specified as the clock source by selecting single or multiple instance pin(s), instance(s), or signal(s) and dragging them from *nTrace* window or *nSchema* window and dropping them in the *Clock Sources* table.

  A box is displayed at the beginning of the row for each clock source or group.

  The default state for the box is checked (*on*). When the box is checked, this specified clock source is treated as the active clock source and is extracted when the **OK** button is clicked.

When the box column header is checked, consequently all rows are also checked.

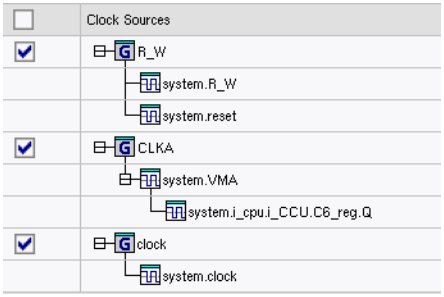
NOTE: If the SDC command is imported as `create_generated_clock -combinational`, a generated clock is displayed in the *xxxx*(ComboGenerated) format (such as, `system.i_cpu.i_CCU.C19_reg.Q(ComboGenerated)`) in the **Clock Sources** table.

- **Add:** After a clock source is typed in the *Clock Source* text field or dragged from *nTrace* or *nSchema* window and dropped to the *Clock Source* text field, click this button to add the clock source. If a clock source is not selected in the *Clock Sources* table, the clock source which is input in the *Clock Source* text field is added as a new clock root. If a clock root or a generated clock is selected, clicking the **Add** button adds the selection as the child (generated) clock of the selected clock. Only one clock source can be added at a time.

  The signal name or the instance port name is checked before being added. If the signal/instance does not exist in the current design, a warning message is reported for the invalid name.

- **Rename:** When a clock is specified in the *Clock Sources* table, the clock name is mapped to the *Clock Sources* table header automatically. After a new name is entered and the **Rename** button is clicked, the content is updated to the new name. This option is enabled only when a clock is specified in the *Clock Sources* table.

- **Group:** When an SDC file containing grouped clock sources is loaded, clock sources are grouped as shown in the figure below. The grouped clock is represented by the  icon.



A new group can be created by selecting one or more root clock sources and then clicking the **Group** button. The grouped clock source is shown with the  icon with the default name *NewGroup1, NewGroup2...*

NewGroupN. If the selection is not a root clock source, a warning message is reported.

A root clock source can be added to an existing group by dragging the selection to the desired clock group.

- **Ungroup:** Select the desired clock group and click this button to ungroup the existing clock group. All root clock sources in the group become roots of all the trees.
- **Delete:** After at least one clock is selected in the *Clock Sources* table, click this button to remove the selected clock (including its generated clocks).
- **Extract Generated Clock:** Click this button to open the *Extract Generated Clock List* form where all generated clocks of the known clock sources listed in the *Clock Sources* table are displayed.

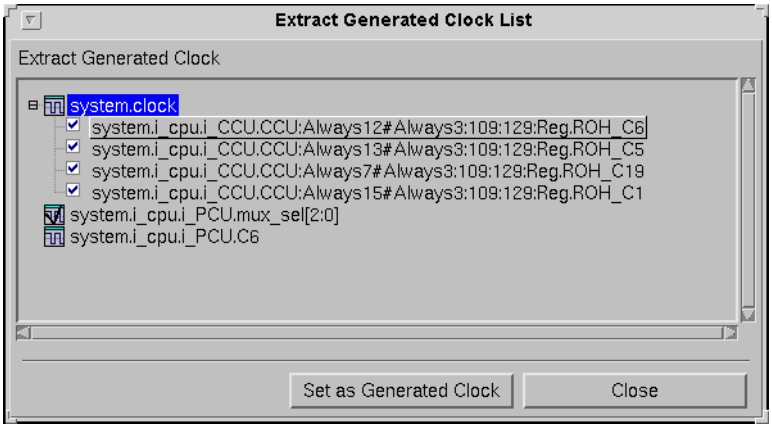


Figure: Extract Generated Clock List

Click the **Set as Generated Clock** button to select the clock pin and set it as a generated clock. The selected clock pin is added as a generated clock under the corresponding clock source in the *Clock Sources* table. Multiple selections are allowed. All clock pins are checked *on* by default.

CTS Attributes Tab

The CTS Attributes tab includes two CTS modes: **ICC Mode** and **First Encounter Mode**. The default mode is **First Encounter Mode**. Different modes result in different CTS settings. When the **ICC Mode** option is enabled, only the *Global CTS* tab is displayed and when the **First Encounter Mode** option is enabled, *Local CTS* and *Global CTS* tabs are displayed. If the imported CTS file is in ICC format, **ICC Mode** is used automatically.

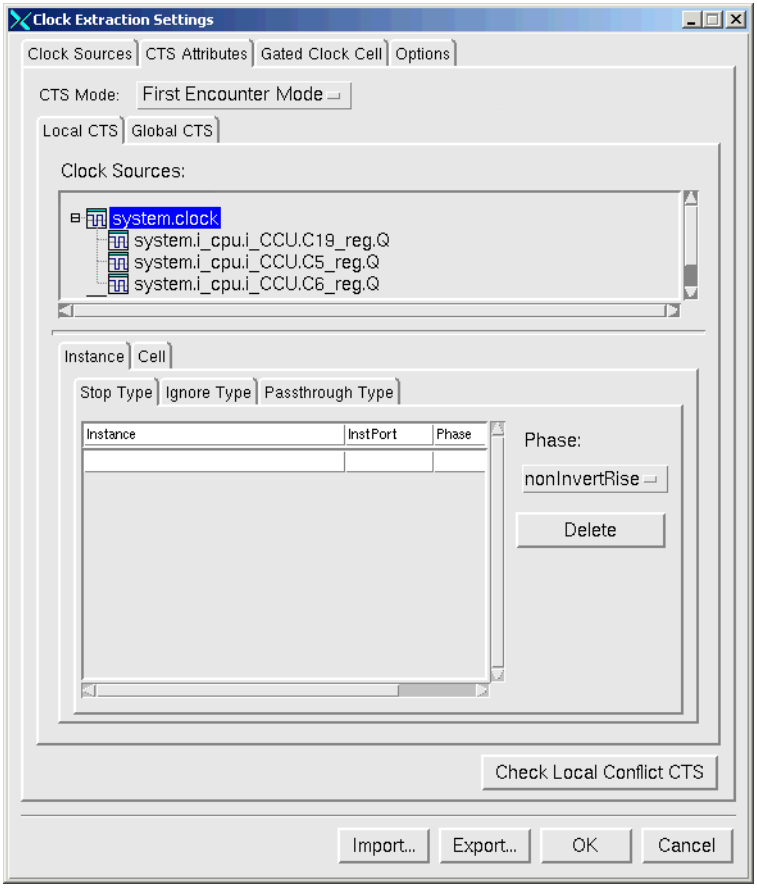


Figure: Clock Extraction Settings Form - CTS Attributes Tab

ICC Mode

The *Global CTS* tab in the ICC mode includes **Instance** and **Cell** tabs. The **Instance** and the **Cell** tabs both have **Stop Type**, **Ignore Type**, **Passthrough Type**, **Trigger Type**, and **Exception Type** tabs and function the same. The **Instance** and the **Cell** tabs are for instances and cells respectively.

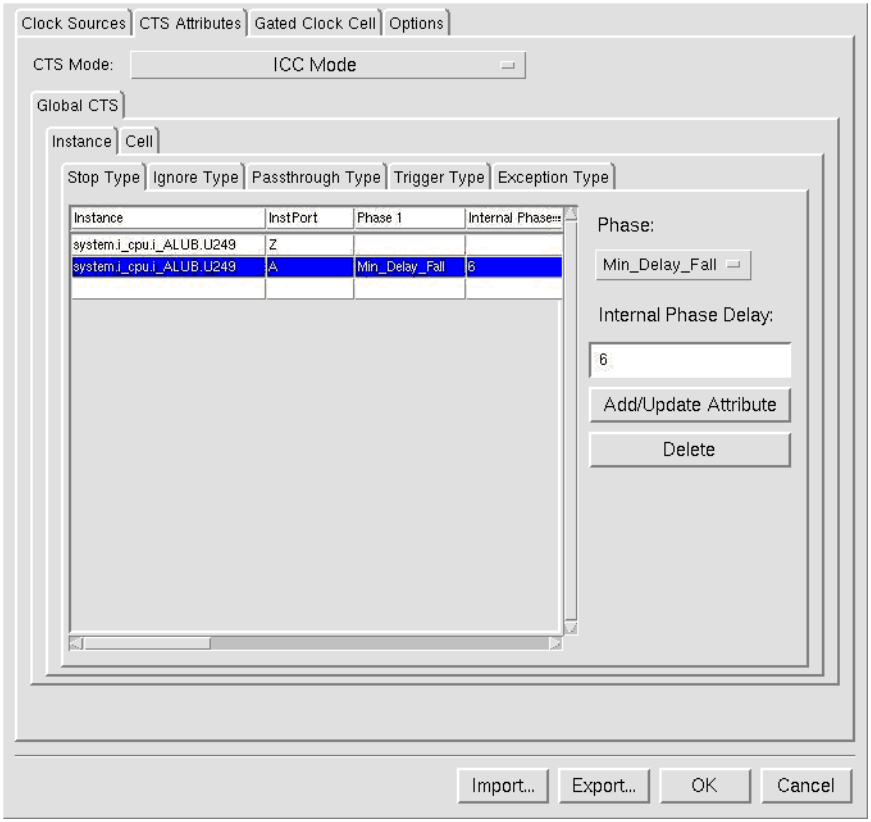


Figure: Global CTS Subtab for ICC Mode

- **Stop Type:** Input CTS pin attributes or primary output ports to stop the path on the pin/port. In the table below the *Stop Type* tab, each instance or cell has four pairs of phase fields: Phase 1, Internal Phase Delay 1; Phase 2, Internal Phase Delay 2; Phase 3, Internal Phase Delay 3, and Phase 4, Internal Phase 4. The type in the *Phase* column must be selected in the **Phase** selection field; the value in the *Internal Phase Delay* column must be entered in the **Internal Phase Delay** text field.

  Phase: Set pin attributes as **Max_Delay_Rise**, **Min_Delay_Rise**, **Max_Delay_Fall**, or **Min_Delay_Fall** from the selection field. Each selection option can only be used once and the options must be evenly distributed to **Phase 1**, **Phase 2**, **Phase 3**, or **Phase 4** columns.

  Internal Phase Delay: Enter a value in this text field to set the internal phase delay. The value always goes with the currently selected **Phase**.

  Add/Update Attributes: After the target instance or the cell is selected, click this button to add or renew the value change of internal phase delay.

Delete: Click this button to remove any instances/cells from the stop list.

- **Ignore Type:** CTS pin attributes or primary output ports can be specified in this section to ignore/not show the path of the specified pin/port in the results.

  Paths specified in the SDC file as `set_disable_timing` are also treated as **Ignore Type**. This attribute is written when the CTS constraints are exported.

  To specify clock trees to be ignored, drag the selected clock(s) in the *nSchema* window and drop it (them) in the **Ignore Type** section. Turn the **Show Ignore Tree** option *on* and click the **Set Ignored Tree Color** button to open the *Set Ignore Tree Highlight Color* form. A color for the ignored tree can be specified by choosing a preferred color.

  Delete: Click this button to remove any instances/cells from the ignore list.

  Show Ignored Tree and **Set Ignored Tree Color:** The specified ignore trees can be highlighted in the *nSchema* window and the *Clock Tree Browser Window* with the **Show Ignore Tree** option. When the **Apply to nTrace Window** option is turned *on* in the *Set Ignore Tree Highlight Color* form (opened by clicking the **Set Ignored Tree Color** button), the instance/port shown in the *nTrace* window is also highlighted. When the **Show Ignore Tree** option is turned *off*, the specified clock tree in the **Ignore Type** section is not displayed in the *nSchema* window.

- **Passthrough Type:** CTS pin attributes can be specified in this section to guide the clock tree to force passing sequential logic or macros which should have stopped originally.

  The rules for passthrough are as follows:

  - Passthrough must be specified on only one input pin if only single output exists for that macro/sequential logic.
  - Passthrough must be specified on only one input pin and one output pin if multiple outputs exist for that macro/sequential logic.

  When an instance is specified as the passthrough type at the input port of the instance, the instance or its output port is taken as a passthrough type automatically. If only the output port is specified as the passthrough type, all the input ports are taken as passthrough types automatically.

  Generated Clock: This selection field sets the polarity for the generated clock with the following available passthrough points: **Don't Care, Non-invert**, and **Invert**. When the passthrough point is the output of a storage element, such as a flip-flop, a latch or a macro, specify the polarity

of the generated clock. If **Non-invert** is specified at the passthrough point, the generated clock has the same polarity as the input clock. If **Invert** is specified at the passthrough point, the generated clock has the opposite polarity to the input clock. Specify **Don't Care** when no passthrough point is set.

Delete: Click this button to remove any instances/cells from the passthrough type table.

- **Trigger Type:** Set the instance port or the cell port with **Positive** or **Negative** in the **Trigger Type** selection field. The setting works on unknown trigger type instances and affects the polarity of signals driven by unknown instances, such as XOR, as the output from XOR is usually unknown. This setting corresponds to the specification for the **set_clock_sense** SDC command.

  Delete: Click this button to remove the selected instance or cell from the trigger type list.

- **Exception Type:** Set the type from one of the **dont_size_cells**, **dont_touch_subtrees**, or **dont_buffer_nets** types for an instance port.

  Type: Set the instance or cell type as **dont_size_cells**, **dont_touch_subtrees**, or **dont_buffer_nets**. This setting is shown in the **Type** column of the table in the *Clock Tree Browser* window.

  This setting corresponds to the **dont_size_cells**, **dont_touch_subtrees**, or **dont_buffer_nets** options for the **set_clock_tree_exceptions** SDC command.

  Delete: Click this button to remove the selected instance or cell from the exception type list.

First Encounter Mode

This mode shows *Local CTS* and *Global CTS* tabs where local and global clock tree synthesis constraints can be viewed and set.

Local CTS Subtab

CTS constraints for a specified clock source can be set in this tab.

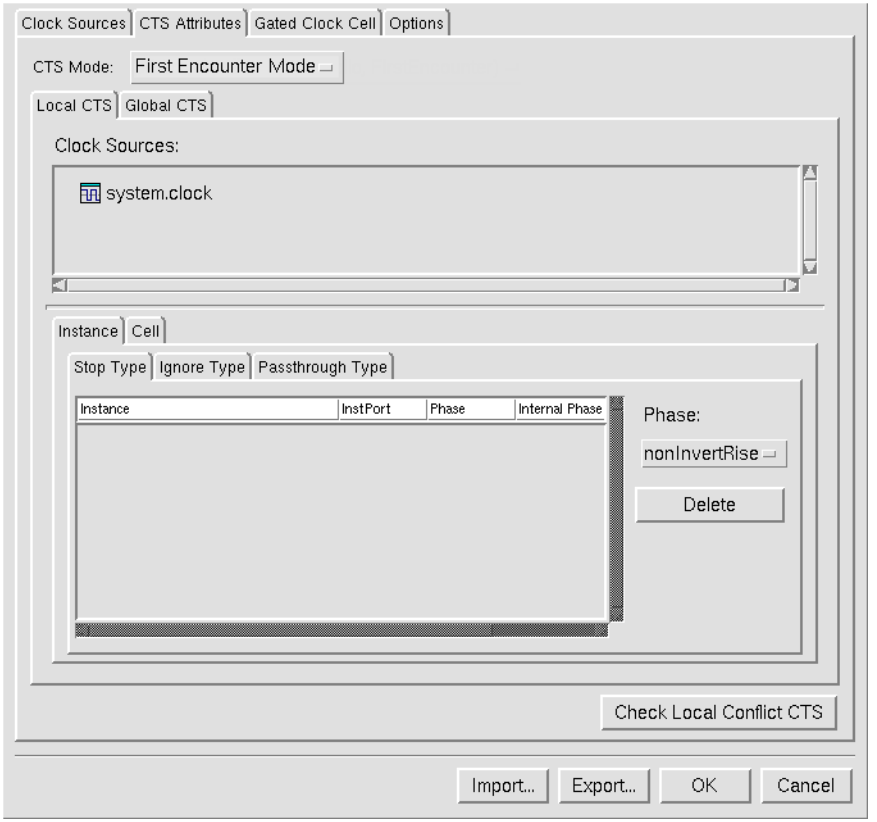


Figure: Local CTS Subtab for First Encounter Mode

- **Clock Sources:** This section shows the clock sources set in the *Clock Sources* table of the *Clock Sources* tab. The  icon before a clock source indicates that a local CTS constraint has been set on the clock source.
- **Instance/Cell:** After selecting a clock source in the *Clock Sources* table, set the CTS constraints under the selected clock source. An asterisk (*) shown in the upper-right side of *Instance*, *Cell*, *Stop Type*, *Ignore Type*, or *Passthrough Type* tabs indicates a local CTS constraint has been set.

  To specify clock trees to stop, ignore, or passthrough, drag the selected clock(s) in the *nSchema* window and drop it (them) in the table in *Stop Type*, *Ignore Type*, or *Passthrough Type* tab.

  The options on the *Instance* and the *Cell* tabs are similar and are described below.

  - **Stop Type:** CTS pin attributes or primary output ports can be specified in this table to stop the path on the pin/port in the results.

Phase: Pin attributes can be set with **nonInvertRise**, **nonInvertFall**, **invertRise**, or **invertFall** in this selection field.

Delete: Click this button to remove any instances/cells from the stop list.

- **Ignore Type:** CTS pin attributes or primary output ports can be specified in this table to ignore/not show the path of the specified pin/port in the results.

  Delete: Click this button to remove any instances/cells from the ignore list.

- **Passthrough Type:** CTS pin attributes can be specified in this table to guide the clock tree to force passing sequential logic or macros that should have stopped originally. The rules for passthrough are as follows:

  - Passthrough must be specified on only one input pin if only a single output exists for that macro/sequential logic.
  - Passthrough must be specified on only one input pin and one output pin if multiple outputs exist for that macro/sequential logic.

  When an instance is specified as the passthrough type at the input port of the instance, the instance or its output port are considered as a passthrough type automatically. If only the output port is specified as the passthrough type, all the input ports are considered as passthrough types automatically.

  Generated Clock: This selection field sets the polarity for the generated clock with the following available passthrough points: **Don't Care, Non-invert**, and **Invert**. When the passthrough point is the output of a storage element, such as a flip-flop, a latch, or a macro, the polarity of the generated clock can be specified. If **Non-invert** is specified at the passthrough point, the generated clock has the same polarity as the input clock. If **Invert** is specified at the passthrough point, the generated clock has opposite polarity to the input clock. Specify **Don't Care** when no passthrough point is set.

  Delete: Click this button to remove any instances/cells from the *Passthrough Type* table.

Global CTS Subtab

Global CTS settings can be specified in this tab. The *Global CTS* and the *Local CTS* subtabs are similar. The only difference is **Show Ignore Tree** and **Set Ignore Color** options in the *Ignore Type* tab.

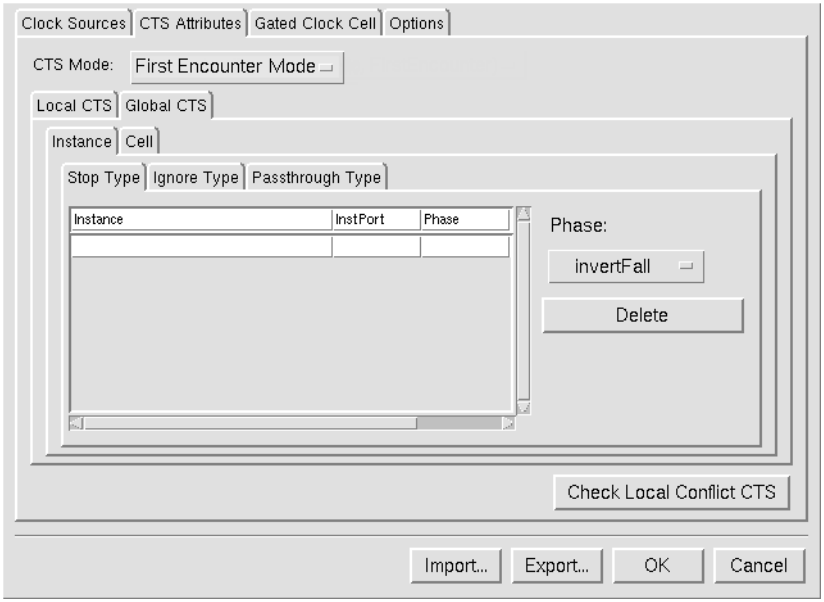


Figure: Global CTS Subtab for First Encounter Mode

The sections and options on **Instance** and **Cell** tabs are similar and are described below.

- **Stop Type:** CTS pin attributes or primary output ports can be specified in this section to stop the path on the pin/port in the results.

  Phase: Pin attributes can be set with **nonInvertRise**, **nonInvertFall**, **invertRise**, or **invertFall** in this selection field.

  Delete: Click this button to remove any instances/cells from the stop list.

- **Ignore Type:** CTS pin attributes or primary output ports can be specified in this section to ignore/not show the path of the pin/port in the results.

  Paths specified in the SDC file as `set_disable_timing` are also treated as “Ignore Type”. This attribute is written out when the CTS constraints are exported.

  To specify clock trees to ignore, drag the selected clock(s) in the *nSchema* window and drop it (them) in the **Ignore Type** section. Turn the **Show Ignore Tree** option *on* and click the **Set Ignored Tree Color** button to open the *Set Ignore Tree Highlight Color* form. A color for the ignored tree can be specified by choosing a preferred color.

  Delete: Click this button to remove any instances/cells from the ignore list.

Show Ignored Tree and **Set Ignored Tree Color:** The specified ignore trees can be highlighted in the *nSchema* window and the *Clock Tree Browser Window* with the **Show Ignore Tree** option. If the **Apply to nTrace Window** option is turned *on* in the *Set Ignore Tree Highlight Color* form which is opened by clicking the **Set Ignored Tree Color** button, the instance/port shown in the *nTrace* window is also highlighted. If the **Show Ignore Tree** option is turned *off*, the specified clock tree in the **Ignore Type** section is not displayed in the *nSchema* window.

- **Passthrough Type:** CTS pin attributes can be specified in this section to guide the clock tree to force passing sequential logic or macros which should have stopped originally.

  The rules for passthrough are as follows:

  - Passthrough must be specified on only one input pin if a single output exists for that macro/sequential logic.
  - Passthrough must be specified on only one input pin and one output pin if multiple outputs exist for that macro/sequential logic.

  When an instance is specified as the passthrough type at the input port of the instance, the instance or its output port is considered as a passthrough type automatically. If only the output port is specified as the passthrough type, all the input ports are considered as passthrough types automatically.

  Generated Clock: This selection field sets the polarity for the generated clock with the following available passthrough points: *Don't Care*, *Non-invert*, and *Invert*. When the passthrough point is the output of a storage element, such as a flip-flop, a latch, or a macro, the polarity of the generated clock can be specified. If *Non-invert* is specified at the passthrough point , the generated clock has the same polarity as the input clock. If *Invert* is specified at the passthrough point , the generated clock has opposite polarity to the input clock. Specify *Don't Care* when no passthrough point is set.

  Delete: Click this button to remove any instances/cells from the *Passthrough Type* table.

- **Check Local Conflict CTS:** This option checks for setting conflicts between the local CTS settings. If an inconsistency exists among the types of CTS constraints (such as, one instance/cell is set as Ignore point in A clock root, but set as Passthrough point in B clock root; one instance/cell is set as Stop point in A clock root, but not set as Stop type in B clock root), all conflicting instances/cells and their CTS constraint types are shown in the *Local Conflict CTS Point* form.

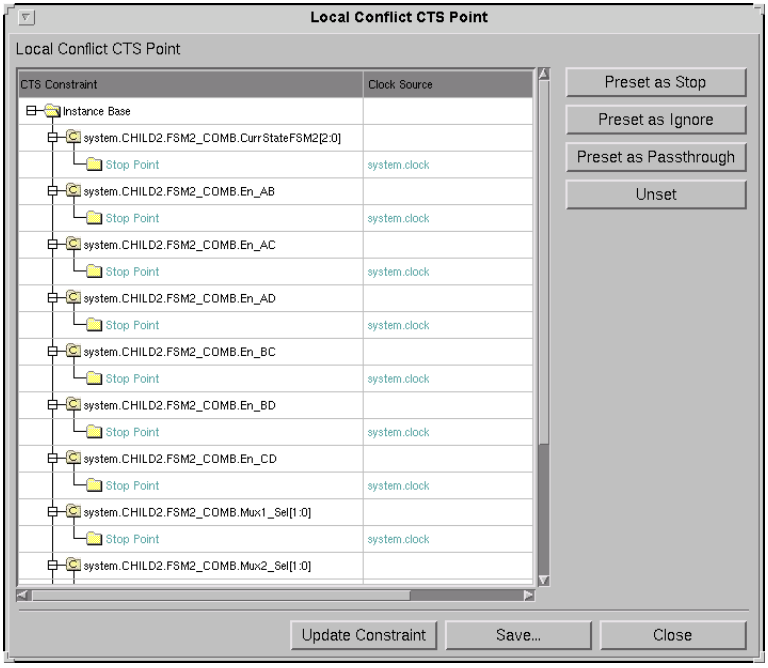


Figure: Local Conflict CTS Point Form

Select a point in the table and double-click the row to go to local CTS settings in the *CTS Attributes* tab of the *Clock Extraction Settings* form. Click the **Preset as Stop**, **Preset as Ignore**, **Preset as Passthrough**, and **Unset** buttons in the *Local Conflict CTS Point* form to modify the CTS constraint settings. After the **Update Constraint** button is clicked, changes are immediately applied on the **CTS Attributes** tab -> **Local CTS** tab of the *Clock Extraction Settings* form.

Gated Clock Cell Tab

Cells dragged from the *nSchema* window and dropped to this tab are defined as a gated clock cell.

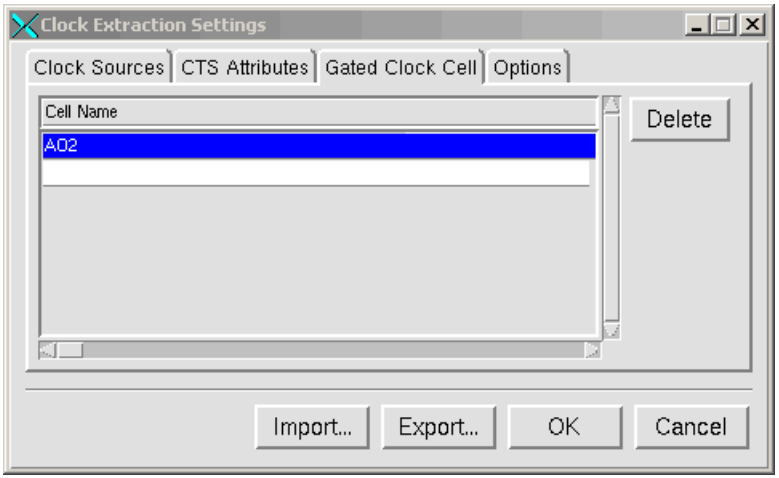


Figure: Clock Extraction Settings Form - Gated Clock Cell Tab

Options Tab

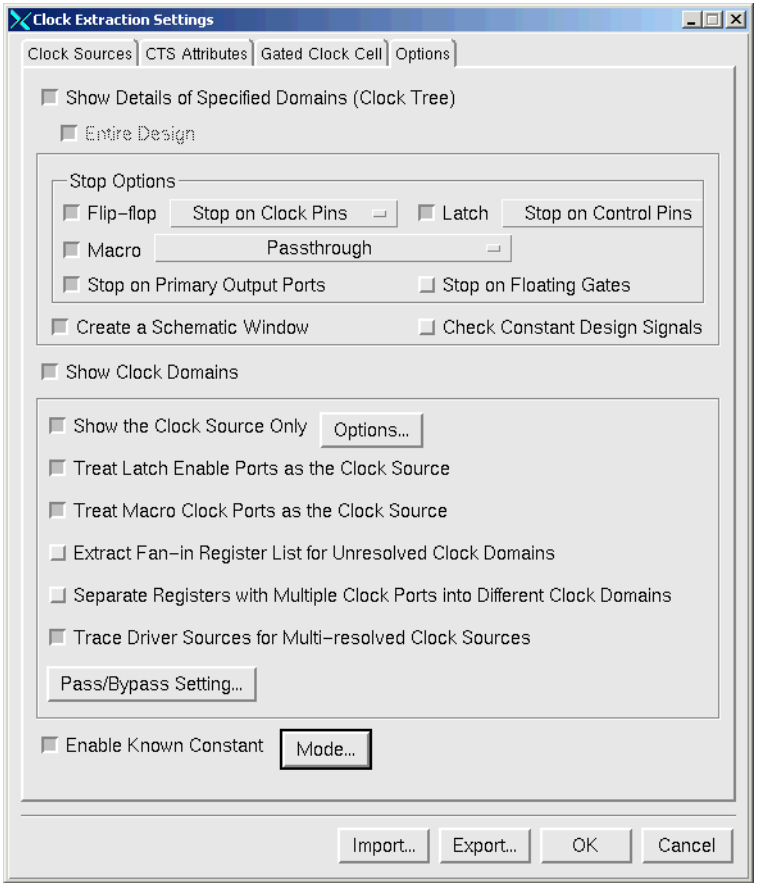


Figure: Clock Extraction Settings Form - Options Tab

The **Options** tab includes the following options:

- **Show Details of Specified Domains (Clock Tree):** When this option is turned *on* and the **OK** button in the *Clock Extraction Settings* form is clicked, the extracted clock tree specified in the *Clock Sources* table of the *Clock Sources* tab of the *Clock Extraction Settings* form is shown in the *Clock Tree Browser* window.
  - **Entire Design:** When this option is turned *on*, the possible clock sources for the entire design are identified and displayed in the *Clock Tree Browser* window.

In the *Stop Options* section, several options can be selected to determine whether or not to trace through flip-flops, latches, macro cells, primary output ports, or floating gates:

- **Flip-flop:** When this option is turned *off*, *nAnalyzer* traces through the flip-flops. When this option is turned *on*, *nAnalyzer* stops tracing the flip-flops based on the condition selected from the following options: **Stop on Clock Pins**, **Stop on Non-clock Pins**, and **Stop on All Pins**. The default is *on* with **Stop on Clock Pins**.
- **Latch:** When this option is turned *off*, *nAnalyzer* traces through the latches. When this option is turned *on*, *nAnalyzer* stops tracing through latches based on the condition selected from the following options: **Stop on Control Pins**, **Stop on Non-control Pins**, and **Stop on All Pins**. The default is *on* with **Stop on Control Pins**.
- **Macro:** When this option is turned *off*, *nAnalyzer* traces through the macro cells. When this option is turned *on*, *nAnalyzer* stops tracing through the macro cells based on the condition selected from the following options: **Stop on Clock Pins**, **Stop on Non-clock Pins**, **Stop on All Pins**, and **Passthrough**. The default is *on* with **Passthrough**.

  If **Stop on Non-clock Pins** is selected, it indicates the clock tree result has reached a non-clock pin of a macro cell which also has a clock pin. Macro cells without clock pins are never included in the clock tree results when either **Stop on Clock Pins** or **Stop on Non-clock Pins** is chosen for macro cells. They only appear when the **Stop on All Pins** option is selected.
- **Stop on Primary Output Ports:** When this option is turned *off*, *nAnalyzer* traces the primary output ports. When this option is turned *on*, *nAnalyzer* stops tracing on the primary output ports.
- **Stop on Floating Gates:** When this option is turned *off*, *nAnalyzer* traces the floating gates. When this option is turned *on*, *nAnalyzer* stops tracing on the floating gates. The default value is *off*.
- **Create a Schematic Window:** When this option is turned *on*, the clock tree can be viewed in a schematic window with the clock tree elements as specified in the *Clock Extraction Settings* form. When this option is turned *on* and the **OK** button at the lower-right corner of the *Clock Extraction Settings* form is clicked, both the *Clock Tree Browser* window and the *nSchema* window (showing the created clock tree) are opened. When the **Create a Schematic Window** option is turned *off*, the *Clock Tree Browser* window is opened without opening the *nSchema* window after the **OK** button is clicked. The default value is *on*.
- **Check Constant Design Signals:** When this option is turned *on*, clock tree extraction checks constant design signals. This option is disabled when the **Enable Known Constant** option is disabled. The default value is *off*.

- **Show Clock Domains:** When this option is turned *on* and the **OK** button in the *Clock Extraction Settings* form is clicked, the *Clock Domains* window opens and lists all the currently loaded or extracted clock domains.

In the *Extract Clock Domain Options* section, any of the following options can be selected to determine the extraction settings:

- **Show the Clock Source Only:** This option controls the extraction results. When this option is turned *off*, the clock source is not merged, because the internal clock node is treated as a clock source and all leaf cells are split into their internal clock nodes.

  When the option is turned *on*, only the clock sources are treated as a clock source, so the clock sources are merged. Clock sources are "User specified", "Register output", "Primary input", or "Multi domains"; other types belong to internal clock nodes. By default, the **Show the Clock Source Only** option is turned *on*.

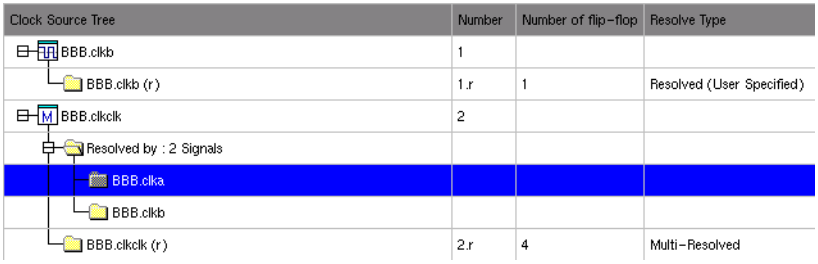
  Use the **Options** button after **Show the Clock Source Only** to merge gated clock (GATECLK) cells and/or combinational logic for clock domain extraction. When the **Show the Clock Source Only** option is turned *on*, clicking the **Options** button opens the *Merge Cell Options* form.

  Turn the **Merge GATECLK Cell** and/or **Merge Combinational Logic** options *on* or *off* depending on whether to merge cells based on gated clock cells or combinational logic. By default, if the **Show the Clock Source Only** option is turned *on*, the **Merge GATECLK Cell** and **Merge Combinational Logic** options are also turned *on*.

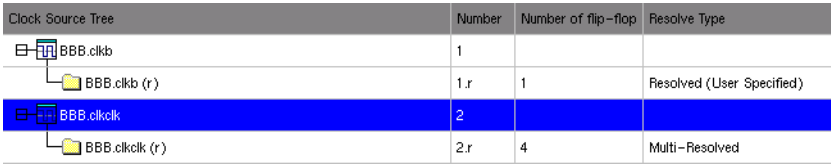
- **Treat Latch Enable Ports as the Clock Source:** This option ignores latch-enable ports during clock domain extraction. When this option is turned *on*, latch enable ports are taken as a clock source. When tracing fan-ins of clock sources to build a clock source tree, the data ports and enable ports of tri-state instances are traced through. When this option is turned *off*, latch enable ports are not considered as a clock source. When tracing fan-ins of clock sources to build a clock source tree, only data ports (not enable ports of tri-state instances) are traced through. The default value of this option is *on*.
- **Treat Macro Clock Ports as the Clock Source:** When this option is turned *on*, *nSchema* stops on the clock pin of macro cells during clock tree tracing actions. The macro cells include:
  - MACRO user-defined cell types in *.map* or *.lib* files.
  - Cell types that *nSchema* cannot recognize.

- **Extract Fan-in Register List for Unresolved Clock Domains:** This option enables viewing of the unresolved clock domain register list in *Long List* text format in a *Text Viewer* frame. When this option is turned *on* and the desired unresolved clock domain is highlighted on the *Clock Source Tree* display area of the *Clock Domains* window and the **Tools-> New File Viewer-> Long List** command is invoked, the trace results (including instance names and their cell types) of the selected clock domain are displayed in a *Text Viewer* frame.
- **Separate Registers with Multiple Clock Ports into Different Clock Domains:** When this option is turned *on*, special flip-flops with multiple clock ports are extracted into different clock domains. The default value is *off*.
- **Trace Driver Sources for Multi-resolved Clock Sources:** This option traces source information of multi-resolved clock sources and displays the results in the *Clock Domains* window or *nSchema*. When extracting clock domains and this option is turned *on*, if the clock source is a multi-resolved domain, the  icon is displayed before the clock source. Under the clock source is the 'Summary Node' (shown in the manner of “Resolved by: XX Signals” with XX indicating the number of multiple clock sources).

  The clock signals driving the clock source are listed under the 'Summary Node'. The last row is the leaf node that is named based on the multi-resolved clock. See the following figure for an example:



When this option is turned *off* and the clock source is a multi-resolved domain, the extraction results in the *Clock Domains* window are as shown in the following figure:



In the *nSchema* window, when the **Trace Driver Sources for Multi-resolved Clock Sources** option is turned *on*, the results from the clock source to the domain are included as shown below.

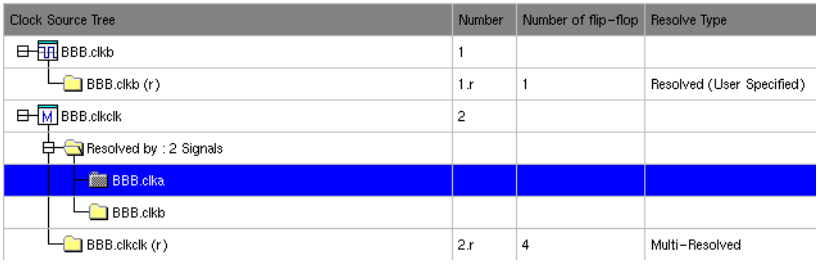


Figure: Trace Driver Sources for Multi-resolved Clock Sources - ON

Otherwise, when the **Trace Driver Sources for Multi-resolved Clock Sources** option is turned *off*, only the clock domain is included; the clock signal driving the clock source is not included. See the following figure for an example. The default value of this option is *on*.

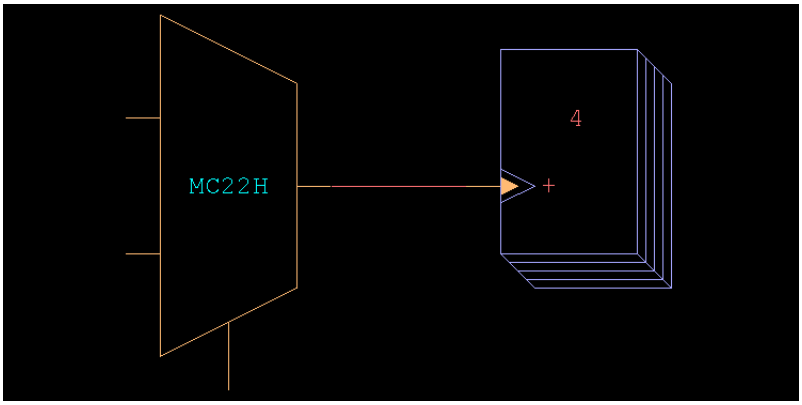


Figure: Trace Driver Sources for Multi-resolved Clock Sources - OFF

- **Pass/Bypass Setting:** Click this button to open the *Pass/Bypass Module Setting* form where module(s) can be specified to be passed or not during clock domain extraction. After entering the name of the desired module in the **Module Name** text field or dragging a module from the *nTrace* window and dropping it to the text field, enabling either the **Pass** or the **Bypass** options, and clicking the **Add** button, the specified module is shown in the main display area of the *Pass/Bypass Module Setting* form.

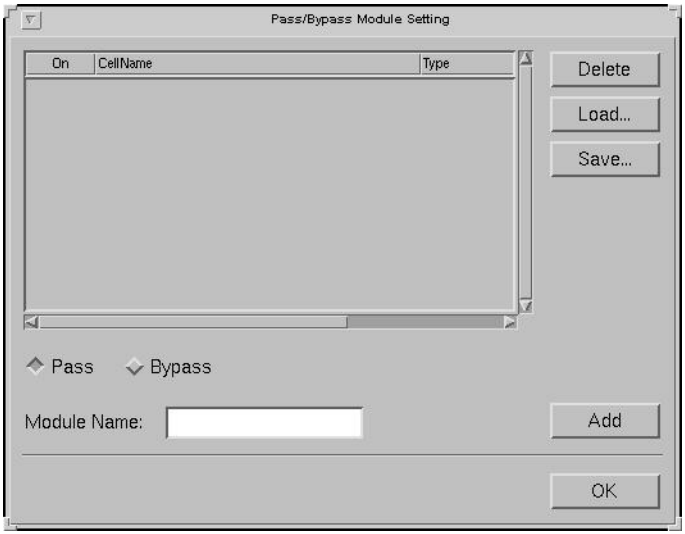


Figure: Pass/Bypass Module Setting Form

The *Pass/Bypass Module Setting* form includes the following options:

- **Pass:** When this option is selected, specify the module(s) to be passed.
- **Bypass:** When this option is selected, specify the module(s) not to be passed.
- **Module Name:** Enter the name of the desired module in this text field or drag a module from the *nTrace* window and drop it to this text field.
- **Add:** After this button is clicked, the specified module is shown in the main display area of the *Pass/Bypass Module Setting* form. The **On** check box is also checked automatically.
- **Delete:** Select the module(s) in the main display area of the *Pass/Bypass Module Setting* form and click this button to delete the setting(s).
- **Load:** To load a previously saved module setting file, click this button to open the *Open File* form so the directory structure can be viewed and a file name can be specified.
- **Save:** To save the module settings to a file, click this button to open the *Save File* form so the directory structure can be viewed and a file name can be specified.

- **Enable Known Constant:** When this option is turned *on,* a constant can be set on certain signals to act as guides to find clock sources during clock tree tracing. When the **Enable Known Constant** option is enabled, a signal can

be specified as a known constant if it is used to control the passage of a clock signal through combinational logic (such as, a MUX or a NAND).

Click the **Mode** button to open the *Mode* form where a constant can be set on certain signals in the **Editing Mode** section to act as guides to find clock sources during clock tree tracing. Refer to the *Mode Form* section for details. In the **Mode List** section, settings for the specified signal or cell mode(s) are shown.

Mode Form

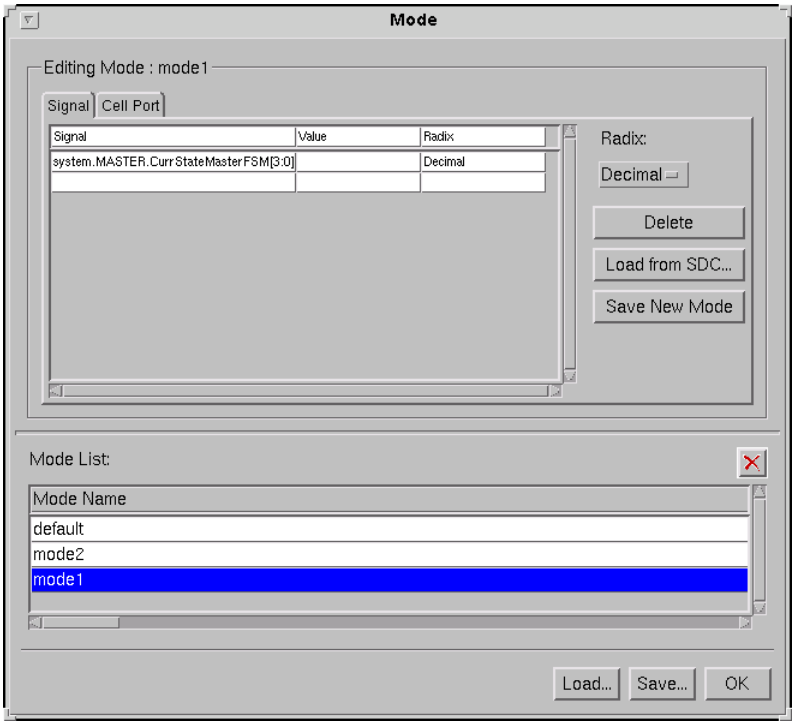


Figure: Mode Form

In the *Editing Mode* section, specify constants on input signals or input pins of the combinational gates under the *Signal* and the *Cell Port* tabs to identify the clock sources of the design. On the table under *Signal* and *Cell Port* tabs, a signal name with full hierarchical path is entered in the *Signal* tab and a specific cell type and port with constant value is entered in the *Cell Port* tab. After a signal/cell is dragged from the *nSchema* window to the table on *Signal* or *Cell Port* tabs, a value can be entered in the *Value* column.

The *Mode* form includes the following options and section:

- **Radix:** Set the radix value to **Decimal**, **Hex**, **Octal**, or **Binary** in this selection field.
- **Delete:** Click this button to remove the selected signal or port from the table.
- **Load from SDC:** Click this button to load the SDC file contents to the *Mode* form. The supported SDC command includes `set_logic_one`, `set_logic_zero`, `set_case_analysis`, and `set_disable_timing`.
- **Save New Mode:** Click this button to add the mode settings to the *Mode List* section. The default mode names are Mode1, Mode2, ... ModeN. Modes previously set are listed in the **Mode** selection field (

) of the *Clock Tree Browser* window. Specify the desired mode to view the different results in the *Clock Tree Browser* window. Also, the *nSchema* window can be opened for different schematic views of different mode settings.
- **Mode List:** This section displays the mode setting names that are defaulted to Mode1, Mode2, ... ModeN. A preferred name can be typed in the **Mode List** table to replace the default mode name.

  Click the  button to remove the selected mode from the list. Drag-left to select multiple modes and then delete them.
- **Load:** Click this button to load the mode settings from a saved file (**.mode*).
- **Save:** Click this button to save the mode settings to a file with the following format:

  ```
  [Clock_Tree_Mode_Section]
  $ModeName = $filePath
  ```

  Example:

  ```
  [Clock_Tree_Mode_Section]
  Mode1 = /prod1/home/yingsung/Mode1.sdc
  ```

  The SDC file is saved where the mode file is saved and the file name is the same as the mode name ($ModeName.sdc).
- **OK:** Click this button to close the *Mode* form and return to the *Clock Extraction Settings* form.

Import Button

Click the **Import** button to open the *Import* form where the directory structure can be viewed and a previously saved SDC (Synopsys Design Constraint) file, CTS (Clock Tree Synthesis) file, or Novas file can be imported. The file format is determined by the **Format** selection field.

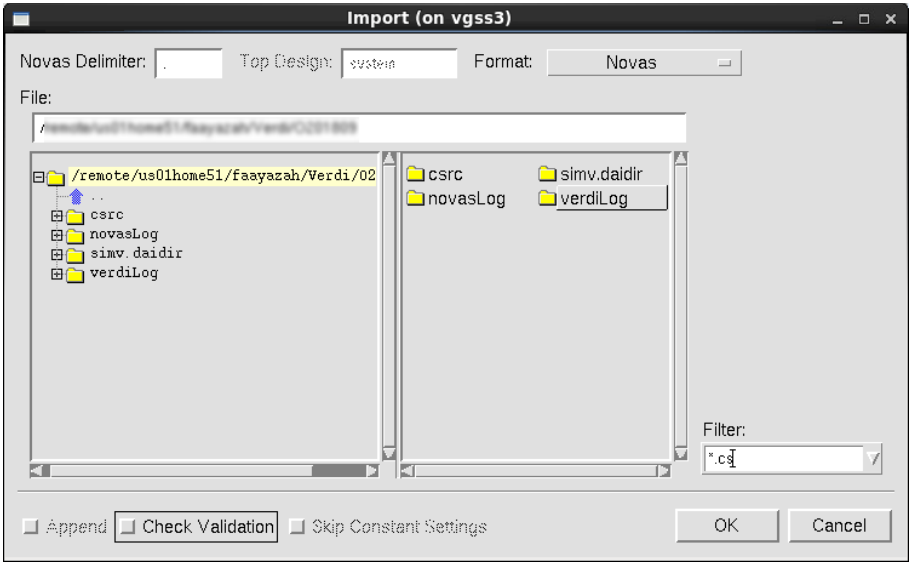


Figure: Import Form

The *Import* form includes the following options and fields:

- **SDC/CTS/Novas Delimiter:** Specify the delimiter in the **Delimiter** text field. The default varies based on the specification in the **Format** selection field.
  - **SDC:** When **SDC** is selected as the format, the default is the forward slash (/) character.
  - **CTS:** When **CTS (FirstEncounter)** or **CTS (ICC)** are selected as the format, the default is the forward slash (/) character.
  - **Novas:** When **Novas** is selected as the format, the default is the period (.).

  Other supported delimiters include at (@) sign, caret (^), pound (#) sign, and pipe (|) character.
- **Top Design:** Specify the design top by typing the name before an SDC file, a CTS file, or a Novas file is loaded. The default is the current top scope.

- **Format:** Specify the file format by selecting one of the following: **SDC**, **CTS (FirstEncounter)**, **CTS (ICC)**, and **Novas**. The default is **Novas**.
- **File:** Specify the SDC file, the CTS file, or the Novas file to be loaded.
- **Filter:** Displays the file format filter for the file. The filter value changes based on the specification in the **Format** selection field. Additional filters can be added in this selection field using a semicolon (;) as the separator.

  Changes to the file format filter are not saved in the *novas.rc* resource file.
- **Append:** This option is disabled when the file format is **Novas**. When this option is turned *on*, the Verdi platform checks whether any setting conflicts exist before SDC file, CTS file, or Novas file is loaded. If the signals (or cell port) are the same (value conflict), the original signal or cell port is overwritten. If a conflict exists between the **Signal** tab and the **Cell Port** tab of the *Mode Form*, the **Signal** tab has priority over the **Cell Port** tab. For example, the signal *n517* is connected to port *A* of an *AN2* lib cell. Assume the value of *n517* in **Signal** tab is 1 and *AN2.A* in the **Cell Port** tab is 0, the values of all signals connected to port *A* of *AN2* lib cells is 0 except *n517* (which is 1).

  A warning message appears indicating that some values were set previously. The content of the tabs must be modified when the conflict occurs.
- **Check Validation:** When this option is turned *on*, the Verdi platform confirms whether the constant cell port, the constant signal, or the known clock source in the SDC file, CTS file, or Novas file to be loaded exists in the current design. If not, a message appears that signal, net, or instance does not exist. By default this option is *off*.

NOTE: The following SDC commands are supported:

```
[Clock Definition]
create_clock
create_generated_clock

[Constant Setting]
set_logic_one
set_logic_zero
set_case_analysis 0 | 1
set_disable_timing
```

- **Skip Constant Settings:** This option is disabled when the file format is **Novas**. When this option is turned *on*, then the constant settings (`set_logic_one`, `set_logic_zero`, `set_case_analysis`,

`set_disable_timing`) are not added in the SDC file or CTS file specified.

Export Button

Click the **Export** button to open the *Export* form where the directory structure can be viewed and a file to save the SDC (Synopsys Design Constraint) file, CTS (Clock Tree Synthesis) file, or Novas file can be specified. The file format is determined by the **Format** selection field.

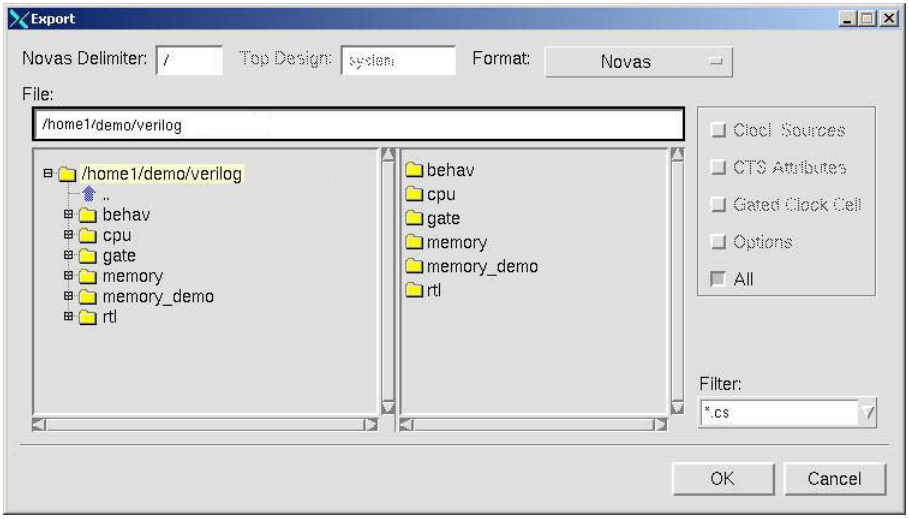


Figure: Export Form

The *Export* form includes the following options and fields:

- **SDC/CTS/Novas Delimiter:** Specify the delimiter in the **Delimiter** text field. The default varies based on the specification in the **Format** selection field.
  - **SDC:** When **SDC** is selected as the format, the default is the forward slash (/) character.
  - **CTS:** When **CTS (FirstEncounter)** or **CTS (ICC)** is selected as the format, the default is the forward slash (/) character.
  - **Novas:** When **Novas** is selected as the format, the default is the period (.) character.

  Other supported delimiters include at (@) sign, caret (^), pound (#) sign, and pipe (|) character.
- **Top Design:** Specify the design top by typing the name before an SDC file, a CTS file, or a Novas file is loaded. The default is the current top scope.

- **File:** Specify the SDC file, CTS file, or Novas file to be loaded.
- **Format:** Specify the file format by selecting one of the following: **SDC**, **CTS (FirstEncounter)**, **CTS (ICC)**, and **Novas**. The default is **Novas**.

  When **Novas** is specified as the format, **Clock Sources**, **CTS Attributes**, **Gated Clock Cell**, **Options**, and **All** options can be enabled and the settings specified in *Clock Sources*, *CTS Attributes*, *Gated Clock Cell*, and *Options* tabs of the *Clock Extraction Settings* form can be included in the output file.

  When **Novas** is selected, **All** is enabled by default. To activate **Clock Sources**, **CTS Attributes**, **Gated Clock Cell** and **Options** (multiple selections are allowed), disable **All**.
- **File:** Specify the SDC file, the CTS file, or the Novas file to be loaded.
- **Filter:** Displays the file format filter for the file. The filter value changes based on the specification in the **Format** selection field. Additional filters can be added in this selection field using a semicolon (;) as the separator.

  Changes to the file format filter are not saved in the *novas.rc* resource file.

Highlight Clock Domains Form

The *Highlight Clock Domains* form is displayed when the **Tools -> Clock Analyzer -> Highlight Clock Domains** command is invoked from the *nTrace* window, the **Trace -> Clock Analyzer -> Highlight Clock Domains** command is invoked from the *nSchema* frame, or the **Tools -> Highlight Clock Domains** command is invoked from the *Clock Domains* window. This form lists all the current loaded clock domains. Specify the preferred color in the general color palate for each clock domain. After clicking **Apply** or **OK** buttons, all open schematic windows are immediately highlighted with the preferred colors and future schematic windows use the specified colors.

If the **Apply on Source Code Window** option is turned *on*, clock domain/ crossing path/synchronizer/crossing register is highlighted in *nTrace*. The default value of this option is *off*.

NOTE: If the highlighted clock includes subclocks with multi-resolved type, the highlighted path is from the clock to its leaf nodes and the path from this clock to the multi-resolved nodes. If the highlighted clock type is multi-resolved, the highlighted path is from the clock (multi-resolved node) to its leaf nodes, but does not include the path from the clock source (parent clock) to the multi-resolved nodes.

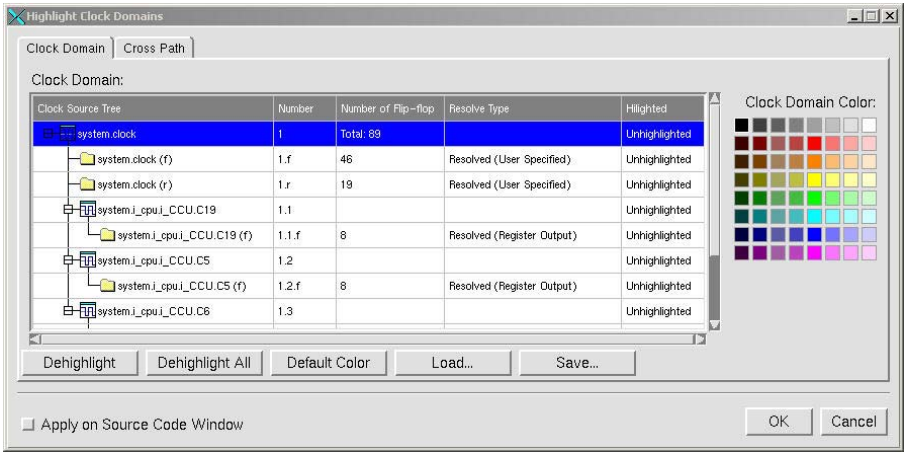


Figure: Highlight Clock Domains Form - Clock Domain Tab

The *Highlight Clock Domains* form includes *Clock Domain* and *Cross Path* tabs.

Clock Domain Tab

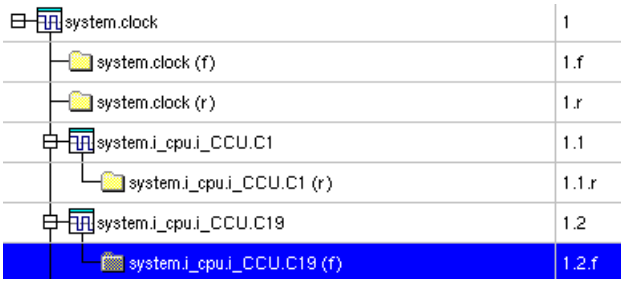
Five columns are displayed in the *Clock Domain* table under the *Clock Domain* tab. The width of each column can be adjusted by placing the cursor over the

vertical bar in the heading row and then pressing and holding the left mouse button. The column headings are summarized below.

- **Clock Source Tree:** Indicates the clock source of each clock domain.
- **Number:** Indicates the number assigned to each clock domain. In this column, the trigger type (**r** or **f**) of each clock domain is also indicated; **r** stands for rising edge trigger of flip-flops and level high for latches and **f** stands for falling edge trigger of flip-flops and level low for latches.

NOTE: The entire clock tree (all nodes including sources and domains under the clock root) is defined as “clock domain.” All domains under the clock root are defined as “subclock domains”.

A number for all tree nodes is displayed. All domains and sub-domains are displayed in hierarchical layers where the source clock signal and generated clock signal can easily be found.



Using the blue highlights as an example, **1** indicates that it is clock source number **1** (based on the hierarchy order in the tree). **2**, attached to **1**, indicates that it is a subclock domain of the first clock source. **f** indicates trigger type falling and **r** indicates rising.

- **Number of Flip-flop:** Indicates the number of flip-flops and latches covered in each clock domain.
- **Resolve Type:** Identifies whether the clock source of each clock domain is **Resolved**, **Unresolved**, **Multi-resolved**, or **Floating**. The type (**Primary Input**, **Register Output**, and **User Specified** for **Resolved** and **Combinational Logic**, **Feedback Loop**, **Fixed Value**, and **Floating** for **Unresolved**) of the **Resolved**/**Unresolved** clock source is also indicated. If a clock source belongs to a primary input of the design, an output of a flip-flop, an output of a latch, or a known clock source, it is defined as resolved. Otherwise, it is unresolved.
- **Hilighted:** Indicates if the clock source of each clock domain is **Hilighted** or **Unhilighted**.

The following options are also available in the *Clock Domain* tab:

- **Clock Domain Color:** Specify the preferred color for each clock domain.
- **Dehilight:** Click this button and then click **Apply**/**OK** to dehighlight the selected clock domain(s).
- **Dehilight All:** Click this button and then click **Apply**/**OK** to dehighlight all of the clock domains in the schematic windows.
- **Default Color:** Click this button to automatically assign a different color to each clock domain. All clock sources and their subclock domains (since they belong to the same clock domain) are assigned the same highlight color. Eight colors are available (gray, red, orange, yellow, green, blue, cyan, and purple) for automatic assignment. If more than eight clock domains exist, the colors are repeated. Any colors that were previously assigned are overwritten by this button.
- **Save:** Click this button to save the highlighted clock domain information in text format. The default file extension is *.rc*.
- **Load:** Click this button to load a text file containing the previously saved highlighted clock domain information.

Cross Path Tab

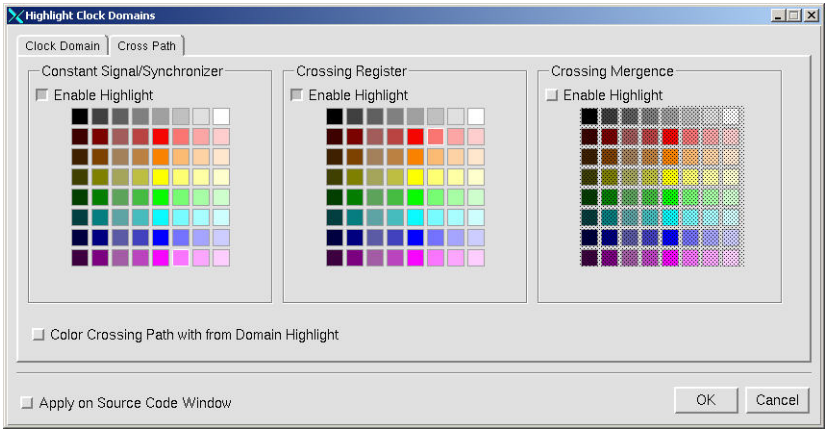


Figure: Highlight Clock Domains Form - Cross Path Tab

In the **Constant Signal/Synchronizer** section, after toggling *on* the **Enable Highlight** option, specify the preferred color for each constant signal or synchronizer element.

In the *Crossing Register* section, toggle *on* the **Enable Highlight** option to specify the preferred color for each crossing register.

In the *Crossing Mergence* section, after toggling *on* the **Enable Highlight** option, specify the preferred color for clock domains that merge through combinational logic.

Check Crossing Paths Form

The *Check Crossing Paths Form* is displayed when the **Tools -> Clock Analyzer -> Check Crossing Paths** command is invoked from the *nTrace* window, the **Trace -> Clock Analyzer -> Check Crossing Paths** command is invoked from the *nSchema* frame, or the **Tools -> Check Crossing Paths** command is invoked from the *Clock Domains* window.

This form includes the *CDC Settings* tab, the *Synchronizer Settings* tab, and the *Pass Scope Settings* tab. In the *CDC Settings* tab, select start-clock-domains and end-clock-domains to check for crossing paths. Multiple start-clock-domains and end-clock-domains can be selected by dragging-left or ctrl-left-click. In the *Synchronizer Settings* tab, specify the synchronizer by cell, but not by instance. In the *Pass Scope Settings* tab, specify the desired scopes for crossing path checking instead of checking the entire design.

Click **Found** or **OK** buttons in the *Check Crossing Paths* form to open the *Crossing Paths Window* section where the crossing path results are listed. Click the **Cancel** button to close the *Check Crossing Paths* form without saving changes. If no crossing paths are found, a warning message is displayed.

CDC Settings Tab

In the *CDC Settings* tab, specify CDC settings for crossing path checking.

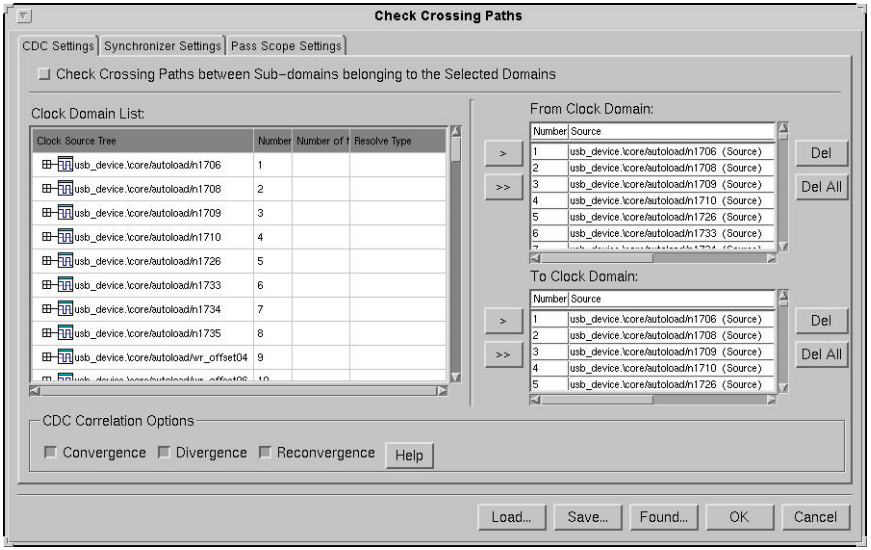


Figure: Check Crossing Paths Form - CDC Settings Tab

The following option and buttons are available:

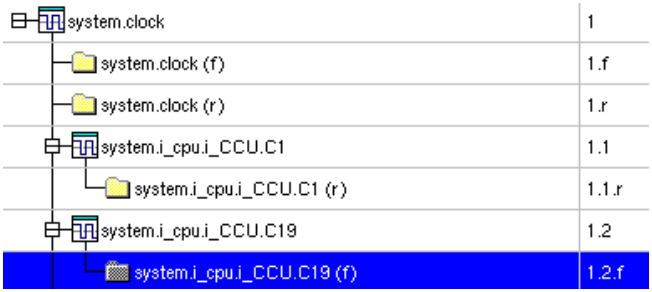
- **Check Crossing Paths between Sub-domains belonging to the Selected Domains:** Select whether to include sub-domains when checking crossing paths. When this option is turned *off*, only crossing paths between the selected domains are checked. Crossing paths in sub-domains are not checked. When this option is turned *on*, crossing paths between all the selected clock domains and sub-domains are checked. The default value is *off*.
- **Clock Domain List:** This table lists all clock domains. Clock source trees in the *Clock Domain List* table are collapsed. Only clock roots are displayed. Double-click the clock root to expand the clock source trees.

  Four columns are displayed in the *Clock Domain List* table. The width of each column can be adjusted by placing the cursor over the vertical bar in the heading row and then pressing and holding the left mouse button. The column headings are summarized below.

  - **Clock Source Tree:** Indicates the clock source of each clock domain. In the form, the clock source relationship in the whole design is listed as a tree-like concept. Each row in the list window represents a clock domain. In this column, the trigger type (**r** or **f**) of each clock domain is also indicated. **r** indicates rising edge trigger for flip-flops and level high for latches. **f** indicates falling trigger for flip-flops and level low for latches.
  - **Number:** Indicates the number assigned to each clock domain.

NOTE: The entire clock tree (all nodes including sources and domains under the clock root) is defined as “clock domain”. All domains under the clock root are defined as “subclock domains”.

A number for all tree nodes is displayed. All domains and sub-domains are displayed in hierarchical layers where the source clock signal and generated clock signal can easily be found.



Using the blue highlights as an example, **1** indicates that it is clock

source number **1** (based on the hierarchy order in the tree). **2**, attached to **1**, indicates that it is a subclock domain of the first clock source. **f** indicates trigger type falling and **r** indicates rising.

- **Number of flip-flop:** Indicates the number of flip-flops and latches covered in each clock domain.
- **Resolve Type:** Identifies whether the clock source of each clock domain is *Resolved*, *Unresolved*, *Multi-Resolved*, or *Floating*. The type (*Primary Input*, *Register Output*, and *User Specified* for *Resolved* and *Combinational Logic*, *Feedback Loop*, *Fixed Value*, and *Floating* for *Unresolved*) of the *Resolved*/*Unresolved* clock source is also indicated. A clock source belonging to a primary input of the design, an output of a flip-flop, an output of a latch, or a known clock source is defined as resolved. Otherwise, it is unresolved.

Select domain(s) in the *Clock Domain List* table and add to the *From Clock Domain/To Clock Domain* tables by clicking the `>` `>>` buttons.

- `>>` **:** Click this button to move all selections from the *Clock Domain List* table to the *From Clock Domain* and/or *To Clock Domain* tables.
- `>` **:** Click this button to move a single selection from the *Clock Domain List* table to the *From Clock Domain* and/or *To Clock Domain* tables.
- **Del:** Select domain(s) in the *Clock Domain List* table and click this button to delete the selection.
- **Del All:** Click this button to delete all domains in the *Clock Domain List* table.

In the *CDC Correlation Options* section, enable one or more of the **Convergence** (different start registers are converged to a point and then stop at the same end register), **Divergence** (one start register diverges to different end registers), and **Reconvergence** (after the end register of multiple register pairs, the path is converged to one point again) options to check cross domain checking (CDC) correlation of the crossing paths between the **From Clock Domain** and the **To Clock Domain**.

Click the **Help** option to open a help file where more information about convergence, divergence, and reconvergence can be obtained. Click the **OK** button to open the *Crossing Paths* window which shows crossing-paths details, CDC correlation results, and register pair details.

If a synchronizer is not specified in the **Synchronizer Settings** tab before enabling the CDC correlation options, a warning message is displayed.

Synchronizer Settings Tab

In the *Synchronizer Settings* tab, the *Synchronizer* table displays the synchronizer that was specified.

Click the **Edit** button on the *Check Crossing Paths* form to open the *Edit* form. The *Edit* form can be used to specify a new synchronizer or to edit an existing synchronizer for the *Synchronizer* display area on the *Check Crossing Paths* form. To change the order of a synchronizer with combined library cells or to add some library cells to an existing synchronizer, choose the synchronizer and then click the **Edit** button.

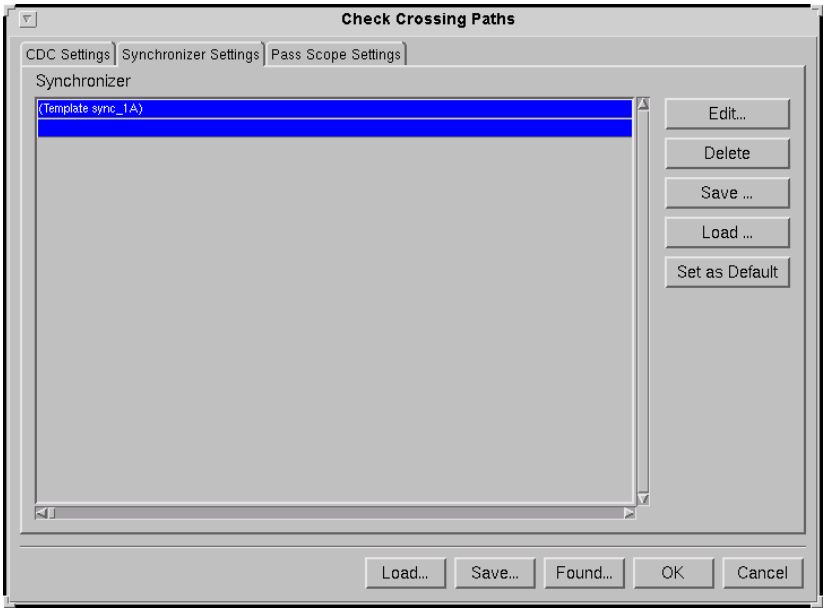


Figure: Check Crossing Paths Form - Synchronizer Settings Tab

Edit: To add a new synchronizer, drag and drop a library cell (cells are supported in symbol libraries) to any area of the **Synchronizer** section. Alternatively, click the **Edit** button to open the *Edit* form to add a new synchronizer (a blank line must be selected) or modify an existing synchronizer (the synchronizer to be edited must be selected first). Refer to the *Edit Form* section for details.

Delete: Click the **Delete** button to delete the selected signal from the **Synchronizer** list.

Save: Save synchronizer to a file (a file browser form opens).

Load: Load synchronizer from a file (a file browser form opens).

Set as Default: Save the desired synchronizer to the *novas.rc* resource file as the default setting.

Edit Form

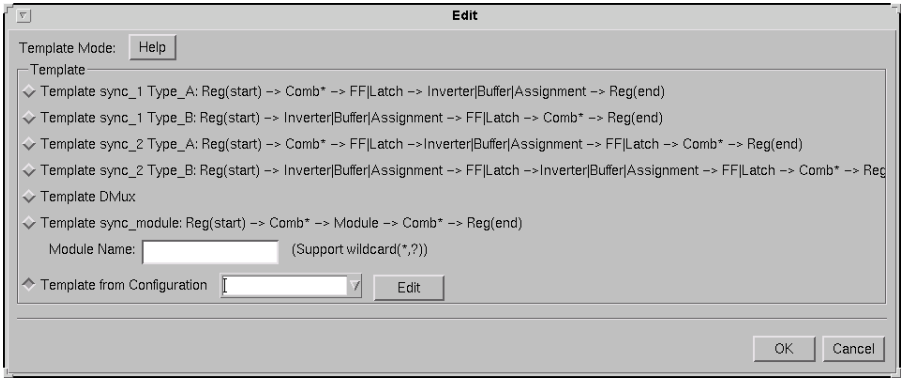


Figure: Edit Form

Template Mode

The **Help** button opens the *Synchronizer Template* file and provides related synchronizer template information. This file contains the same information as *Appendix A: Synchronizer Template* in the *nAnalyzer User Guide and Tutorial* document.

The *Template* section provides seven templates that can be selected. Template sync1 (1A/1B) contains a single flip-flop or latch synchronizer. Template sync2 (2A/2B) contains multiple flip-flop or latch synchronizers.

1. **Template sync_1 Type A: Reg(start) -> Comb* -> FF|Latch -> Inverter|Buffer|Assignment -> Reg(end)**

   Between the start and end registers, the synchronizer consists of 0~n combinational logic followed by a flip-flop or a latch followed by inverter, buffer, or assignment.

2. **Template sync_1 Type B: Reg(start) -> Inverter|Buffer|Assignment -> FF|Latch -> Comb* -> Reg(end)**

   The template sync_1 Type B is the reverse order of template sync_1 Type A. Between the start and end registers, the synchronizer consists of inverter, buffer, or assignment followed by a flip-flop or a latch followed by 0~n combinational logic.

3. **Template sync_2 Type A: Reg(start) -> Comb* -> FF|Latch -> Inverter|Buffer|Assignment -> FF|Latch -> Comb* -> Reg(end)**

Between the start and end registers, the synchronizer consists of 0~n combinational logic followed by a flip-flop or a latch followed by inverter, buffer, or assignment, followed by a flip-flop or a latch followed by 0~n combinational logic.

4. **Template sync_2 Type B: Reg(start) -> Inverter|Buffer|Assignment -> FF|Latch ->Inverter|Buffer|Assignment -> FF|Latch -> Comb* -> Reg(end)**

   Between the start and end registers, the synchronizer consists of inverter, buffer, or assignment, followed by a flip-flop or a latch followed by inverter, buffer, or assignment, followed by a flip-flop or latch followed by 0~n combinational logic.

5. **Template DMux**

   The following four conditions meet the rules of the MUX synchronizer:

   a. The selected input of MUX comes from the *To* domain and is synchronized by nl_sync2 (control path synchronizer).
   b. One of the MUX inputs comes from the *To* domain, the holding loop.
   c. The MUX inputs can be *From* domain, *To* domain, or user-specified static signals.
   d. The logic between the MUX synchronizer and the *To* flip-flop is driven by the *To* domain or the static signals.

   When this synchronizer template is turned *on*, if the specified MUX synchronizer is found during crossing path extraction, the synchronizer is shown in the **Synchronizer** column of the **Register Pair Details** table in the *Crossing Paths* window. After left-clicking to select and highlight a row in the **Register Pair Details** table, invoke the **Tools -> New Schematic** command or click the  icon. The specified synchronizer is highlighted in the *nSchema* window. The highlight color can be specified in the **Cross Path** tab of the **Tools -> Highlight Crossing Path** command.

6. **Template sync_module: Reg(start) -> Comb* -> Module -> Comb* -> Reg(end)**

   If **Template sync_module** is selected, enter a module name in the text field to specify a module synchronizer template. Wildcard characters (* or ?) can be used in the module name.

NOTE: The wildcard character (*) indicates the combinational logic may exist from 0~n times.

---

7. **Template from Configuration**

Enable this option for a flexible synchronizer configuration. Clicking the **Edit** button on the right side of the **Template from Configuration** option opens the *Synchronizer Customization Form*.

Synchronizer Customization Form

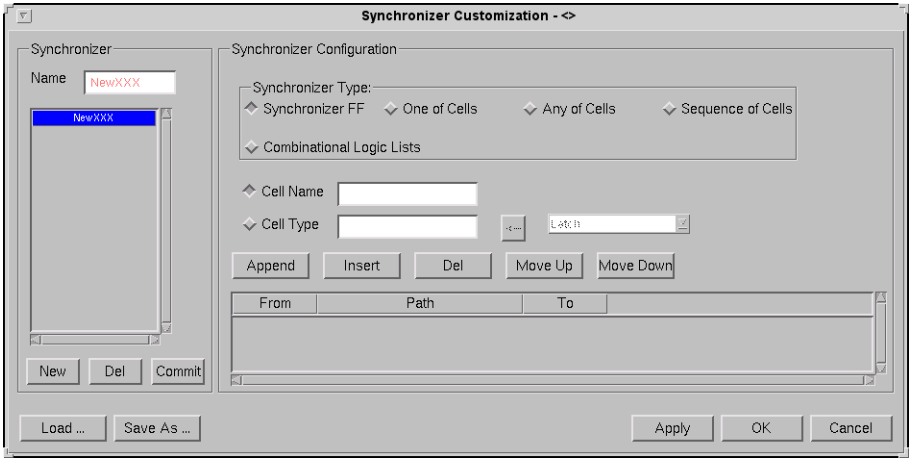


Figure: Synchronizer Customization Form

To create the desired synchronizer template, perform the following steps:

1. Click the **New** button in the *Synchronizer* section to create a new template. The default template name, *NewXXX*, is displayed in red in the **Name** text field. Enter the preferred name in the text field and press the **Enter** key.
2. In the *Synchronizer Configuration* section, enable one of the following four options: **Synchronizer FF**, **One of Cells**, **Any of Cells**, **Sequence of Cells**, and **Combination Logic Lists**. Only one option can be enabled at a time. The default is **Synchronizer FF**.
3. Specify the cell type or the cell name for the synchronizer.
   - When the **Synchronizer FF** option is enabled, either the **Cell Name** option can be enabled and a name entered in the text field or the **Cell Type** option can be enabled and a cell type specified from the selection field. Only one option can be enabled between the **Cell Name** and the **Cell Type** options.
   - When **One of Cells**, **Any of Cells**, or **Sequence of Cells** option is enabled, the **Cell Name** option is disabled. Enter the name of a cell type in the **Cell Type** text field or select a type from the **Cell Type** selection field and input the selection to the text field by clicking the **<-** button.

- When the **Combination Logic Lists** option is enabled, enter a name or drop a cell in the **Cell Name** text field. The **Cell Type** option is disabled.

4. Click **Append** or **Insert** to add the selection in the cell type form. As many cell types as desired can be entered. To adjust the order of different cell types, click **Move Up** and **Move Down** buttons.
5. Click the **Commit** button in the *Synchronizer* section to finalize the desired settings. The synchronizer name changes from red to green after the **Commit** button is clicked.
6. Click **Apply** in the *Synchronizer Customization* form. Then, in the *Edit* form, the customized template is loaded into the selection field of the **Template from Configuration** option.

Synchronizer Section

To define a new template configuration, click the **New** button and type a name in the **Name** text field. Then press the **Enter** key to complete the action. To delete a template configuration, left-click to select and then click the **Del** button. To apply the current settings on the template, click the **Commit** button.

Synchronizer Configuration Section

Use **Synchronizer FF**, **One of Cells**, **Any of Cells**, **Sequence of Cells**, and **Combination Logic Lists** options to specify a desired type of synchronizer. The synchronizer can be constructed from a list of cells. The list of cells is a path starting from a storage element of one clock domain to a storage element of another clock domain. The path can contain **Synchronizer FF**, which are storage elements (such as flip-flops or latches). Start point, end point, and **Synchronizer FF** divide the path into segments. Different combinations of cells in each segment can be configured by **One of Cells**, **Any of Cells**, or **Sequence of Cells** options. The cell pattern matches if a cell is found in the **One of Cells** list, any cells (zero or more cells) are found in the **Any of Cells** list, or a sequence of cells matches the **Sequence of Cells** list. The **Any of Cells** cell type cannot be mixed with **One of Cells** or **Sequence of Cells** in each segment. For more information about the synchronizer template, refer to *Appendix A: Synchronizer Template* in the *nAnalyzer User Guide and Tutorial* document.

- **Synchronizer FF:** Click this option to find the path including flip-flops or latches.
- **One of Cells:** Click this option to find the path with one of the following cell types: Mux, And, Or, Buffer, Inverter, Assignment, or Combinational (Comb).

- **Any of Cells:** Click this option to find the path with any of the following cell types: Mux, And, Or, Buffer, Inverter, Assignment, or Combinational (Comb).
- **Sequence of Cells:** Click this option to find the path meeting the sequence of specified cell types. The desired order of cell types can be specified from Mux, And, Or, Buffer, Inverter, Assignment, Combinational (Comb), and Comb*.

NOTE: Comb refers to one of any kind of combinational logic cell. It can be selected in the selection field when **One of Cells** or **Any of Cells** is enabled. Comb* refers to zero or any combinational logic cells. Comb* can be set at the beginning or end of the path, but it can only be selected when **Sequence of Cells** is enabled.

- **Combination Logic Lists:** Specify the cell name of the combinational cell logic. After the **Combination Logic Lists** option is enabled, the **Cell Type** option is disabled. Type a cell name or drop a cell from the *nSchema* window in the *Cell Name* text field to create a synchronizer list.

NOTE: **Combination Logic Lists** sets synchronizer lists that do not include storage types, while **Synchronizer FF**, **One of Cells**, **Any of Cells**, and **Sequence of Cells** set synchronizer lists that include storage types. After a combinational logic cell is added into the synchronizer list, **Synchronizer FF**, **One of Cells**, **Any of Cells**, and **Sequence of Cells** options are disabled and no storage type synchronizer can be appended to the synchronizer list, unless all combinational logic cells in the synchronizer list are deleted. The opposite holds true for synchronizer lists including storage types.

Cell Name: Use this text field to specify the cell name.

Cell Type: Use this option to select cell types from Mux, And, Or, Buffer, Inverter, Assignment, Comb, and Comb*.

Append: Click this button to add the selected cell type or cell name to the synchronizer list form as an end flip-flop.

Insert: After a cell type or a cell name is highlighted in the synchronizer list form, click this button to add the specified cell type or cell name above the highlighted cell type or cell name.

Del: Click this button to delete the selected item(s) from the synchronizer list form.

Move Up: Click this button to move the order of the selected item(s) up.

Move Down: Click this button to move the order of the selected item(s) down.

Load: Click this button to open the *Load Template Setting File* form where a directory can be specified and a template file loaded.

Save As: Click this button to open the *Save Template Setting File* form where a directory can be specified and the current template settings can be saved to a file.

Apply: Click this button to apply the current synchronizer settings to the **Template from Configuration** option in the *Edit* form. The *Synchronizer Customization* form remains open.

OK: Click this button to apply the current synchronizer settings to the **Template from Configuration** option in the *Edit* form and close the *Synchronizer Customization* form.

Cancel: Click this button to close the *Synchronizer Customization* form without saving any changes.

Pass Scope Settings Tab

In the *Pass Scope Settings* tab, crossing paths are checked only within the specified scopes. Specify the scopes by entering the scope name directly in the text field or by dragging hierarchical instances/signals from *nSchema* or scope trees from *nTrace* and dropping the selection to **Scope 1** and **Scope 2** text fields.

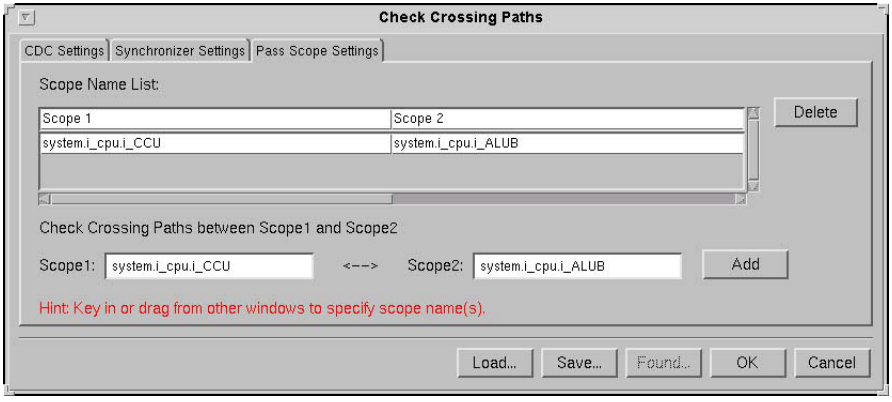


Figure: Check Crossing Paths Form - Pass Scope Settings Tab

Scope Name List: Scope 1 and **Scope 2** columns show the names of the specified scopes.

Delete: Select a scope to remove it from the list. Multiple selections are allowed.

Check Crossing Paths between Scope 1 and Scope 2: Enter a scope name in the **Scope 1/ Scope 2** text field or drag a hierarchical instance/signal in *nSchema* or a scope tree in *nTrace* and drop the selection to this field. "<->" is used to separate **Scope1** and **Scope2**. Click the **Add** button and the selected scopes are

shown in the *Scope Name List* table. If **Scope 1** or **Scope 2** text fields are left blank, all scopes are added as default.

Add: Adds scopes in **Scope 1 <--> Scope 2** to the **Scope Name List**.

Clock Domains Window

The *Clock Domains Window* frame is displayed when **Tools -> Clock Analyzer -> Extract Clock Information** or **Tools -> Clock Analyzer -> List Clock Domains** commands are invoked from the *nTrace* window or **Trace -> Clock Analyzer -> Extract Clock Information** or **Trace -> Clock Analyzer -> List Clock Domains** commands are invoked from the *nSchema* frame.

The *Clock Domains* window lists all currently loaded or extracted clock domains in a tabular format.

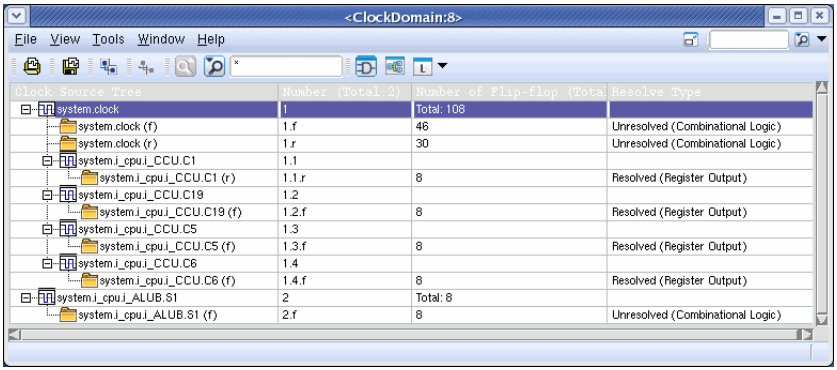


Figure: Clock Domains Window

The main display area shows currently loaded or extracted clock domains. Four columns are shown in the table. The width of each column can be adjusted by selecting the vertical line between column headers and dragging-left. The *Clock Domains* window includes four standard columns: **Clock Source Tree**, **Number**, **Total of Flip-flop**, and **Resolve Type**. The standard column headings are summarized as follows:

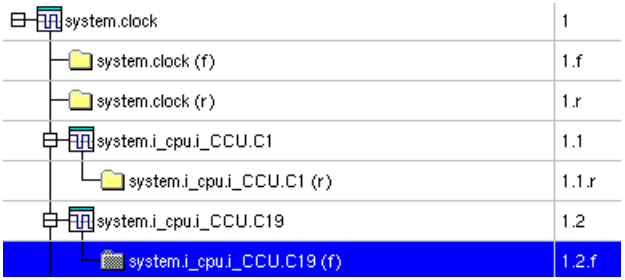
- **Clock Source Tree:** Indicates the clock source of each clock domain. In this table, the clock source relationship in the whole design is listed as a tree-like concept. Each row in the list window represents a clock domain. In this column, the trigger type (**r** or **f**) of each clock domain is also indicated. **r** indicates the rising edge trigger of flip-flops and level high of latches. **f** indicates the falling edge trigger of flip-flops and level low of latches.

  Two conditions result in the parent-child clock source signal relationship: clock frequency divider and/or gated clock.

- **Number:** Indicates the number assigned to each clock domain. The number in the **Total** parentheses refers to the total number of clock roots.

NOTE: The entire clock tree (all nodes including sources and domains under the clock root) is defined as “clock domain.” All domains under the clock root are defined as “subclock domains”.

A number for all tree nodes is displayed. All domains and sub-domains are displayed in hierarchical layers where the source clock signal and generated clock signal can easily be found.



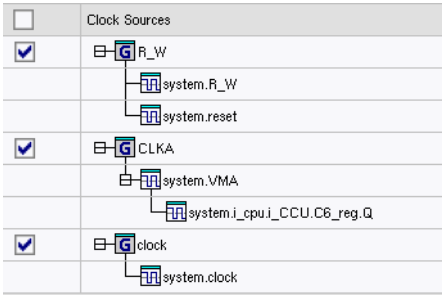
Using the blue highlights as an example, **1** indicates that it is clock source number **1** (based on the hierarchy order in the tree). **2**, attached to **1**, indicates that it is a subclock domain of the first clock source. **f** indicates trigger type is falling and **r** indicates rising.

- **Number of Flip-flop:** Indicates the number of flip-flops and latches covered in each clock domain. The number in the **Total** parenthesis refers to the total number of flip-flops added from all clocks. The flip-flop total is not fixed, because of the filter clock domain functionality. The number is based on the visible clock domains in the *Clock Domains* window.
- **Resolve Type:** Identifies whether the clock source of each clock domain is **Resolved**, **Unresolved**, **Multi-Resolved**, or **Floating**. The type (**Primary Input**, **Register Output**, and **User Specified** for **Resolved** and **Combinational Logic**, **Feedback Loop**, **Fixed Value**, and **Floating** for **Unresolved**) of the **Resolved**/**Unresolved** clock source is also indicated. If a clock source belongs to a primary input of the design, an output of a flip-flop, an output of a latch, or a known clock source, it is defined as resolved. Otherwise, it is unresolved.

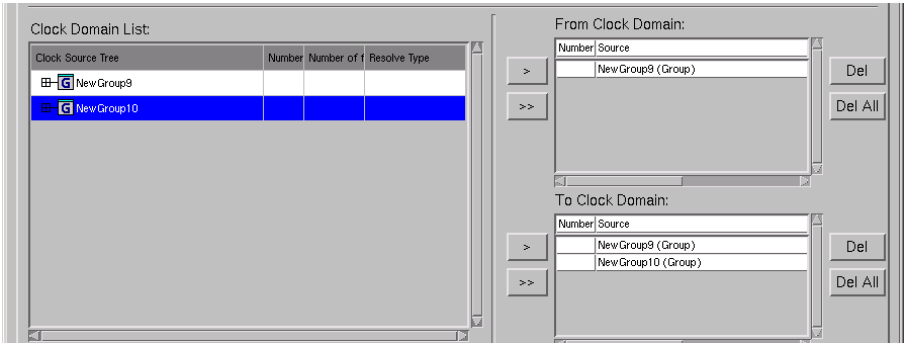
In the *Clock Domains* window, select the desired clock group(s) and invoke the **Group** command from the right-click menu. The grouped clock source is shown in the icon with the default name *NewGroup1, NewGroup2... NewGroupN*. Select the desired clock group and click **Ungroup** or **Rename** command in the right-click menu for group-editing functions. Behaviors and functions for **Group**, **Ungroup**, or **Rename** commands in the right-click menu are the same

as **Group**, **Ungroup**, and **Rename** options in the **Clock Sources** tab of the *Clock Extraction Settings* form.

The grouped clock sources and the right-click menu is shown in the example below:



If clock sources are grouped in the *Clock Domains* window, the clock sources are also grouped in the *Check Crossing Paths* form. To check crossing paths using group settings, select the group and add it to the **From Clock Domain** or **To Clock Domain** sections.



The *Clock Domains Window* frame can become a standalone window by clicking the **Undock** icon on the toolbar. Refer to the *Icons for Dockable Frames* section for details on the menu bar icons.

The menu bar for the *Clock Domains* window is shown below:

Figure: Clock Domains Window Menu Bar

The pull-down menus for the *Clock Domains* window are summarized and explained on the following pages.

File Commands

Save Clock DB

Menu Bar: **File -> Save Clock DB**

Toolbar Icon: 

This command opens the *Save Clock DB* form where the directory structure can be viewed and a file specified to save the clock extraction results to. The default extension (and format) is *.cdb* (Clock Database) file.

Load Clock DB

Menu Bar: **File -> Load Clock DB**

Toolbar Icon: 

This command opens the *Import Clock DB* form where the directory structure can be viewed and a *.cdb* (Clock Database) located to load.

Exit

Menu Bar: **File -> Exit**

This command closes the *Clock Domains* window without making any changes.

View Commands

This menu provides two display results in the *Clock Source Tree* column of the *Clock Domain* table.

Clock Source

Menu Bar: **View -> Clock Source**

This toggle command turns the display of signals in the clock source tree column *on* or *off*.

Instance Pin

Menu Bar: **View -> Instance Pin**

This toggle command turns the display of signals with the related instance pin for each node in the clock source tree column *on* or *off*.

Tools Commands

Set as Clock Root

Menu Bar: **Tools -> Set as Clock Root**

After a clock source root has been specified in the *Clock Domains* window, invoke this command from the **Tools** pull-down menu or from the right-click command menu to open the *Clock Extraction Settings* form where the specified clock source root is treated as the known clock source and displayed in the **Clock Source** table. Click the **OK** button in the *Clock Extraction Settings* form and the extraction results of the specified clock source root are displayed on the *Clock Domains* window.

NOTE: If the specified clock source is a multi-resolved domain, signals driving the clock source tree are also added to the *Clock Source* table. For example, if clock source “BBB.clkclk” is driven by two signals “BBB.clka” and “BBB.clkb,” when “BBB.clkclk” is specified as the clock root, both “BBB.clka” and “BBB.clkb” are added to the *Clock Source* table.

Set as Derived Clock

Menu Bar: **Tools -> Set as Derived Clock**

After a clock source (including register output clock source or clock domain) has been specified in the *Clock Domains* window, invoke this command from the **Tools** pull-down menu or from the right-click command menu to open the *Clock Extraction Settings* form where the specified clock source can be set as the generated clock and displayed under its clock root in the *Clock Source* table. Click the **OK** button in the *Clock Extraction Settings* form and the extraction results of the specified generated clock are displayed on the *Clock Domains* window.

More than one clock source can be selected and then set as derived clocks. If the clock root of selected nodes is a multi-resolved clock source, the selected clock source is added under all of the roots.

Check Crossing Paths

Menu Bar: **Tools -> Check Crossing Paths**

Toolbar Icon: 

This command opens the *Check Crossing Paths* form. Refer to the *Check Crossing Paths Form* section for details.

List Crossing Paths

Menu Bar: **Tools -> List Crossing Paths**

Toolbar Icon: 

This command opens the *Crossing Paths* window. Refer to the *Crossing Paths Window* section for details.

Filter

Menu Bar: **Tools -> Filter**

This command filters undesired clock sources or clock domains from the extraction results and focuses on the clock source or the clock domain of interest.

Invoking this command opens the *Filter* form which consists of the *By Name*, *By Resolve Type*, and *By Clock Source Type* tabs. These three tabs can be used for filtering. The relationship between *By Name*, *By Resolve Type*, and *By Clock Source Type* is AND, but the operation among every specification in *By Name* or *By Resolve Type* is still OR.

By Name Tab
Enter the name of the desired clock domain in the **Filter** text field or drag and drop a clock domain from the *nTrace* window to the **Filter** text field. Then, click the **Add** button and the specified clock domain is shown in the section under the ***By Name*** tab. The **On** check box is also enabled automatically.

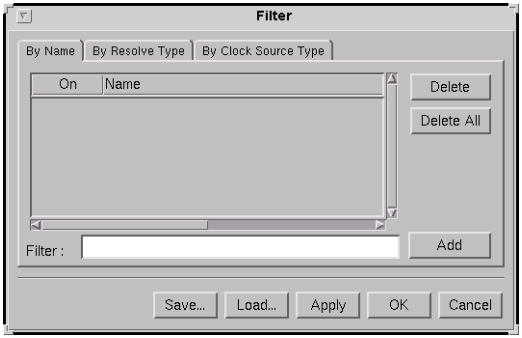


Figure: Filter Form - By Name Tab

Setting(s) can be deleted by left-clicking to select a clock domain (hold the **Ctrl** key and left-click to make multiple selections) and then clicking the **Delete** button. Click the **Delete All** button to delete all clocks whether they are selected or not.

By Resolve Type Tab

Check (enable) the box in the **On** column of the desired resolve type(s) to display the resolve type in the *Resolve Type* column of the *Clock Domains* window. Uncheck (disable) the box of resolve type(s) for clock domain(s) that should not be displayed.

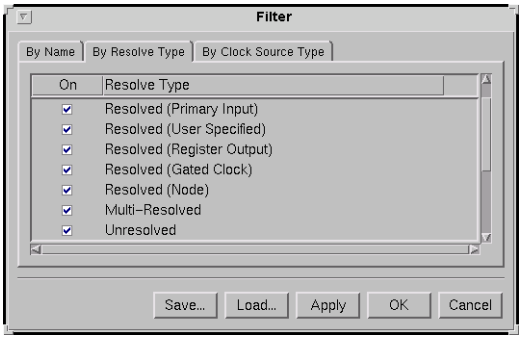


Figure: Filter Form - By Resolve Type Tab

By Clock Source Type

Check (enable) the box in the **On** column of **Known Clock Source** and **Not Known Clock Source** for the types of clock domain(s) to be displayed in the *Clock Source Tree* table on the *Clock Domains* window. Un-check (disable) the **On** box of **Known Clock Source** and **Not Known Clock Source** for the types of clock domain(s) that should not be displayed.

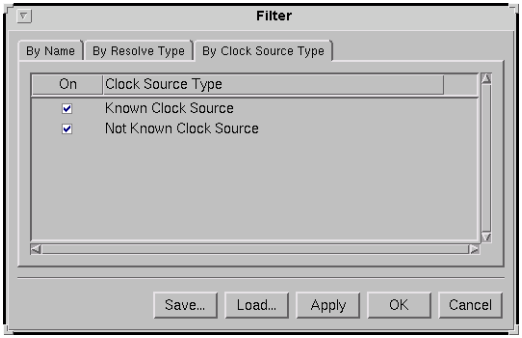


Figure: Filter Form - By Clock Source Type Tab

Find

Menu Bar: **Tools -> Find**

This command opens the *Find* form to locate a clock domain or a clock source based on its name.

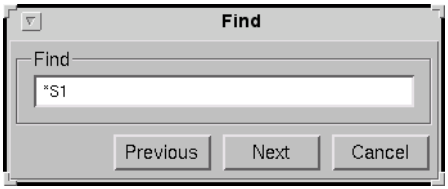


Figure: Find Form

Enter the name of the clock domain or the clock source in the text field of the *Find* form or the * field in the toolbar to search for clock domain or clock source. Wildcard characters are supported. On clicking the **Previous** button (or the toolbar icon) or the **Next** button (or the toolbar icon), the clock domain/clock source matching the setting is highlighted in the **Clock Source Tree** table of the *Clock Domains* window. Click the **Cancel** button to stop the search and close the *Find* form.

Highlight Clock Domains

Menu Bar: **Tools -> Highlight Clock Domains**

This command opens the *Highlight Clock Domains* form. Refer to the *Highlight Clock Domains Form* section for details.

New Schematic

Menu Bar: **Tools -> New Schematic**

Toolbar Icon: 

In the **Clock Source Tree** column, select either a single clock domain (left-click) or multiple clock domains (Ctrl-left-click) and then invoke this command to view the selected clock domain(s) in a flattened schematic window.

NOTE: If the specified clock domain is a multi-resolved domain, this command can only be invoked when the clock source is selected. For example, if clock source `BBB.clkclk` is driven by two signals `BBB.clka` and `BBB.clkb`, only when `BBB.clkclk` is selected, the results can be viewed in the *nSchema* window; otherwise, if `BBB.clka` or `BBB.clkb` is selected, a warning message indicating the *nSchema* results were not viewed is displayed.

New Clock Tree Browser

Menu Bar: **Tools -> New Clock Tree Browser**

Toolbar Icon: 

This command opens the *Clock Tree Browser* window. Refer to the *Clock Tree Browser Window* section for details.

New File Viewer

After selecting either a single clock domain (left-click) or multiple clock domains (Ctrl-left-click), choose from the three commands below to view the clock domain(s) results in different formats.

Long List

Menu Bar: **Tools -> New File Viewer -> Long List**

Toolbar Icon: 

This command displays a detailed instance list for the extracted clock domain in the *Text Viewer* frame. The clock domain is displayed in ASCII format.

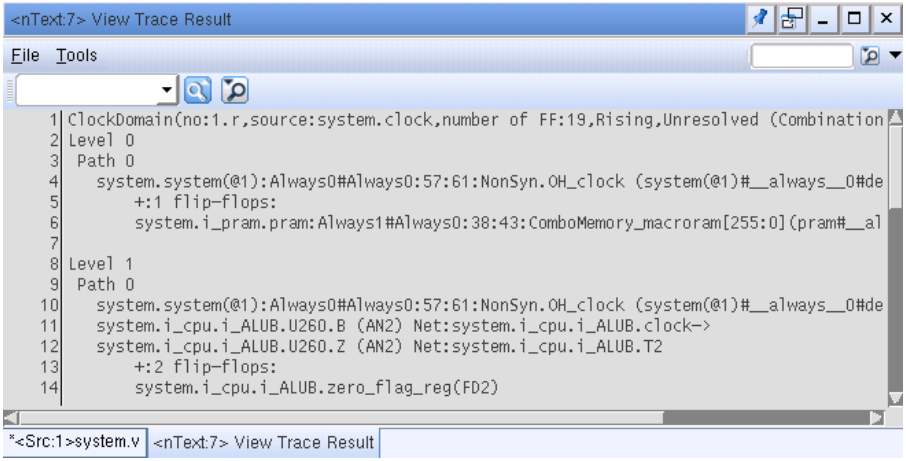


Figure: Text Viewer Frame - Long List View

Refer to the *Text Viewer Frame* section for details about the pull-down menu commands in the *Text Viewer* frame.

Short List

Menu Bar: **Tools -> New File Viewer -> Short List**

Toolbar Icon:

This command displays the clock domain tree in the *Text Viewer* frame in the same format as the extracted results shown in the *Clock Domains* window. The clock domain is displayed in ASCII format.

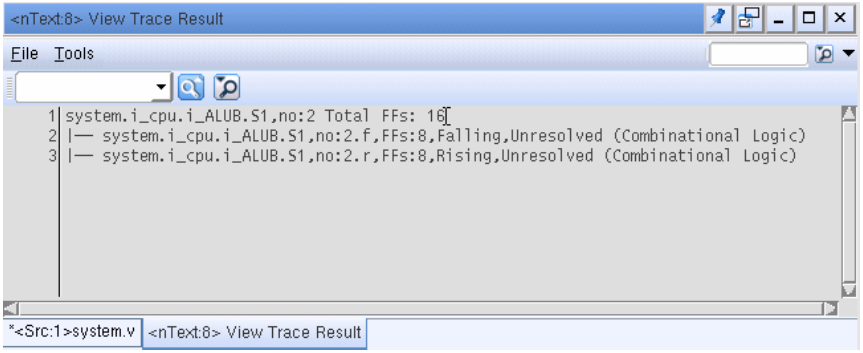


Figure: Text Viewer Frame - Short List View

Refer to the *Text Viewer Frame* section for details about the pull-down commands in the *Text Viewer* frame.

Show Disabled Gate

Menu Bar: **Tools -> New File Viewer -> Show Disabled Gate**

Toolbar Icon: 

This command opens a *Text Viewer* frame showing the disabled logic gates that were propagated by the set constant value of signals. The flip-flop instance name and pin name are included in the disabled report. Clock domains from disabled gates are not extracted.

Options

Menu Bar: **Tools -> Options**

When the **New File Viewer -> Long List** command is *on*, invoke the **Options** command and use the **Delimiter for Long List Format** text field to specify a delimiter. The default value is the period (.) character.

Customize Menu/Toolbar

Menu Bar: **Tools -> Customize Menu/Toolbar**

Refer to the **Tools -> Customize Menu/Toolbar** command description in the *nTrace* chapter for complete details.

Clock Domains Window Frame Right-Click Options

When the right mouse button is clicked anywhere in the menu, toolbar icon, or frame banner area of the *Clock Domains Window* frame or standalone window, a configuration option menu is displayed.This menu can be used to configure the available icons and frames.

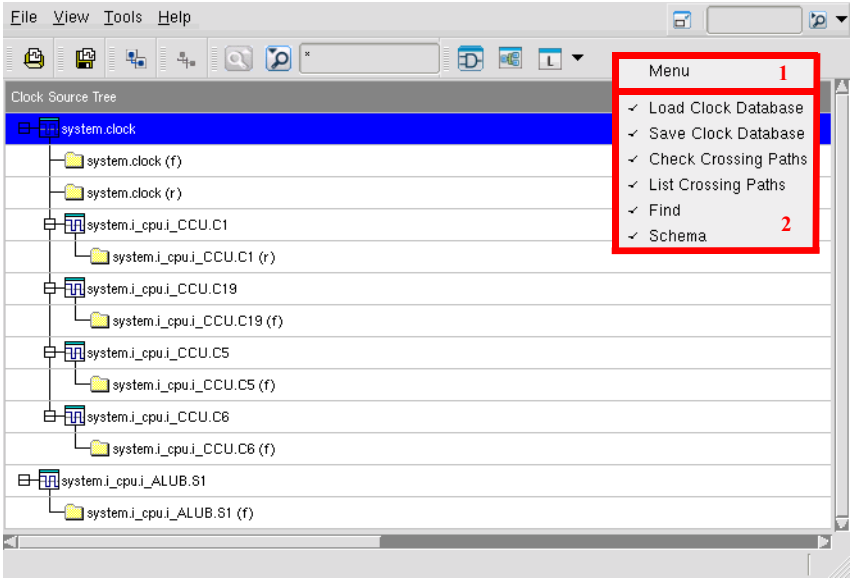


Figure: Configuration Option Menu

The **Menu** option (labeled 1 in the above figure) enables/disables the command menu bar. The options in the bottom section (labeled 2 in the above figure) enable/disable the toolbar icons for different functions.

Clock Domains Body Right-Click Options

The right-click commands in the *Clock Domains* window include the following:

Set as Clock Root

Refer to the **Tools -> Set as Clock Root** command description for details.

Set as Derived Clock

Refer to the **Tools -> Set as Derived Clock** command description for details.

Group

Select single or multiple root clock sources to create a new group. The grouped clock source is shown in the icon with the default name *NewGroup1, NewGroup2... NewGroupN*.

If the selection is not a root clock source, a warning message is reported.

Ungroup

Select the desired clock group to ungroup the existing clock group. All root clock sources in the specified group become roots of all trees.

Move to Group

Select single or multiple root clock sources to add into the specified clock group.

Rename

Select the desired clock group to rename the group name.

Clock Domains Window Toolbar Icons and Fields

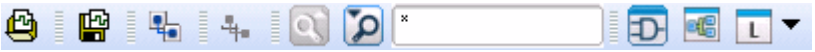


Figure: Toolbar in Clock Domains Window

The available toolbar icons may be modified. Refer to the *Toolbars* section of the *User Interface* chapter in the *Verdi User Guide and Tutorial* manual for details.

Load Clock Database Category

Load Clock DB 

Refer to the **File ->** **Load Clock DB** command description for details.

Save Clock Database Category

Save Clock DB 

Refer to the **File ->** **Save Clock DB** command description for details.

Check Crossing Paths Category

Check Crossing Paths 

Refer to the **Tools ->** **Check Crossing Paths** command description for details.

List Crossing Paths Category

List Crossing Paths 

Refer to the **Tools ->** **List Crossing Paths** command description for details.

Find Category

Find Next 

Refer to the **Tools -> Find** command description for details.

Find Previous 

Refer to the **Tools -> Find** command description for details.

Find 

Refer to the **Tools -> Find** command description for details.

Schema Category

New Schematic 

Refer to the **Tools -> New Schematic** command description for details.

New Clock Tree Browser 

Refer to the **Tools -> New Clock Tree Browser** command description for details.

New File Viewer

The **New File Viewer** command includes **Long List**  , **Short List**  , and **Show Disabled Gate**  commands. Refer to the **Tools -> New File Viewer** command description for details.

Clock Tree Browser Window

The *Clock Tree Browser Window* frame is displayed when **Tools -> Clock Analyzer -> List Specified Clock Domains** or **Tools -> New Schematic from Source -> Clock Tree** commands are invoked from the *nTrace* window or **Trace -> Clock Analyzer -> List Specified Clock Domains** or **Tools -> New Schematic -> Clock Tree** commands are invoked from the *nSchema* frame. The **Tools -> New Clock Tree Browser** command in the *Clock Domains* window can also invoke the *Clock Tree Browser* window.

NOTE: For the **List Specified Clock Domains** commands to be enabled, the **Tools -> New Schematic from Source -> Clock Tree** command in *nTrace* or the **Tools -> New Schematic -> Clock Tree** command in *nSchema* must be executed first.

The *Clock Tree Browser* window list displays the clock tree results in a tabular format.

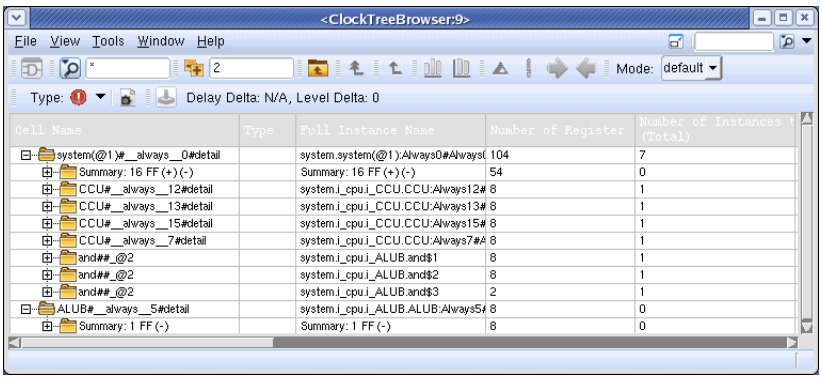


Figure: Clock Tree Browser Window

The *Clock Tree Browser* window displays the clock tree results. The clock tree table summarizes the clock tree for the selected signal. A total of fourteen columns are available in the table. The width of each column can be adjusted by selecting the vertical line between column headers and dragging-left.

The column headings are summarized as follows:

- **Cell Name:** Indicates the name of the cell.
- **Type:** Indicates the tree nodes type (such as, FF or Latch).
- **Full Instance Name:** Indicates the name of the instance.

- **Number of Register:** Indicates the number of registers covered in each clock tree.
- **Number of Instances to Leaves (Total):** Indicates the total number of instances from the specified instance to its leaf registers. The number includes the instance.
- **Number of Instances to Leaves (BUF, MUX, Gated Clock Cell, Others):** The four numbers in parentheses correspond to the number of buffers and inverters, multiplexers, gated clock cells, and other kinds of cells (such as AND/OR) respectively from the specified instance to its leaf registers.
- **Current Level (Total):** Indicates the total number of levels from the clock source to the specified instance.
- **Current Level (BUF, MUX, Gated Clock Cell, Others):** The four numbers in parentheses correspond to the number of levels for buffers and inverters, multiplexers, gated clock cells, and other kinds of cells (such as AND/OR) respectively from the clock source to the specified instance.
- **Min Level to Leaves (Total):** Indicates the total number of instances from an instance to its leaf registers on the shortest path.
- **Min Level to Leaves (BUF, MUX, Gated Clock Cell, Others):** The four numbers in parentheses correspond to the number of buffers and inverters, multiplexers, gated clock cells, and other kinds of cells (such as AND/OR) respectively from an instance to its leaf registers on the shortest path.
- **Max Level to Leaves (Total):** Indicates the total number of instances from an instance to its leaf registers on the longest path.
- **Max Level to Leaves (BUF, MUX, Gated Clock Cell, Others):** The four numbers in parentheses correspond to the number of buffers and inverters, multiplexers, gated clock cells, and other kinds of cells (such as AND/OR) respectively from an instance to its leaf registers on the longest path.

If an SDF file is already loaded before extracting clock tree, the following three column headings are displayed in the *Clock Tree* table.

- **Delay from Source:** Indicates the insertion delay from the clock source (clock source delay is included) to the current input node. In the *Cell Name* column where the summary row (an instance with 'Summary' in the folder description, such as 

) is displayed, the *Delay from Source* column is shown as:Max, Min. The two values come from the maximum and minimum values of leaf nodes that are listed below the summary row.
- **Delay to Leaves:** Indicates the delay from the current input node to the node's leaf input. The two values in parentheses refer to the minimum delay and maximum delay respectively. In the summary row (an instance with 'Summary' in the folder description, such as

), if the

Delay to Leaves column is shown as (0, 0), it indicates no differences exist between the minimum and the maximum delay.

- **Skew to Leaves:** Indicates the difference between the maximum delay to leaf registers and the minimum delay to leaf registers.

The *Clock Tree Browser Window* frame can become a standalone window by clicking the **Undock** icon on the toolbar. Refer to the *Icons for Dockable Frames* section for details on the menu bar icons.

The menu bar for the *Clock Tree Browser* window is shown below:

Figure: Clock Tree Browser Window Menu Bar

The pull-down menus for the *Clock Tree Browser* window are summarized and explained on the following pages.

File Commands

Save Clock Setting

Menu Bar: **File -> Save Clock Setting**

This command opens the *Export* form where the clock source, constant, gated clock cell, and/or CTS attribute settings can be saved in a *novas.rc* resource file format.

Save Clock Tree

Save Path Info

Menu Bar: **File -> Save Clock Tree -> Save Path Info**

After a clock tree is created or highlighted, use this command to save the path information of the trace results, including path type attributes (such as, Data Pin of Register, Duplication, Loop, GATECLK, EXOR, Pin Name, and CTS Stop Pin - including Primary Outputs). This command opens a *Save As* form where the directory and the file name can be specified to save the trace results.

Save All Info

Menu Bar: **File -> Save Clock Tree -> Save All Info**

After a clock tree is created or highlighted, use this command to save all the information shown on the main display area of the *Clock Tree Browser* window. The information is not saved for the columns specified as **Hidden Columns** on the *Configure Columns* form that is opened by invoking the **View -> Configure Level Column** command. This command opens a *Save As* form where the directory and the file name for the trace results can be specified.

Save as Astro Format

Menu Bar: **File -> Save Clock Tree -> Save as Astro Format**

This command dumps the clock tree in the Astro format. A *Save Clock Tree Browser Content* form opens where the directory and the file name can be specified to use for trace results.

Save Clock Tree Database

Menu Bar: **File -> Save Clock Tree -> Save Clock Tree Database**

This command opens the *Save Clock Tree Browser Content* form where the clock tree extraction results and CTS settings can be saved into a text (*.log*) file.

Load Clock Tree Database

Menu Bar: **File -> Load Clock Tree Database**

This command opens the *Load As* form where the clock tree extraction results file can be loaded. If loading is successful, the *Clock Tree Browser* window opens with the saved clock tree extraction results displayed.

Import Path Data File

Menu Bar: **File -> Import Path Data File**

This command opens the *Import Report File* form where the timing report can be imported for the current browser tree window. Refer to the **File -> Import Path Data Files - nAnalyzer** command in the *nTrace* chapter for details.

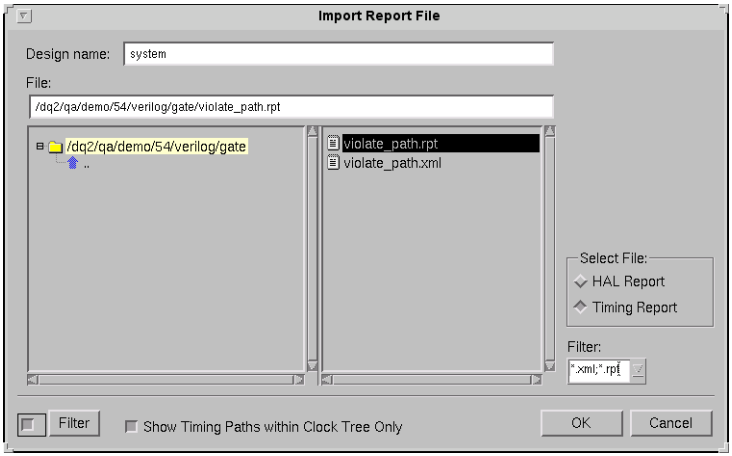


Figure: Import Report File Form

The main difference between the *Import Report File* form being opened from the *nTrace* window and the *Clock Tree Browser* window is the **Show Timing Paths within Clock Tree Only** option. After the desired file is specified, enable the **Show Timing Paths within Clock Tree Only** option to open the *Timing Report* form.

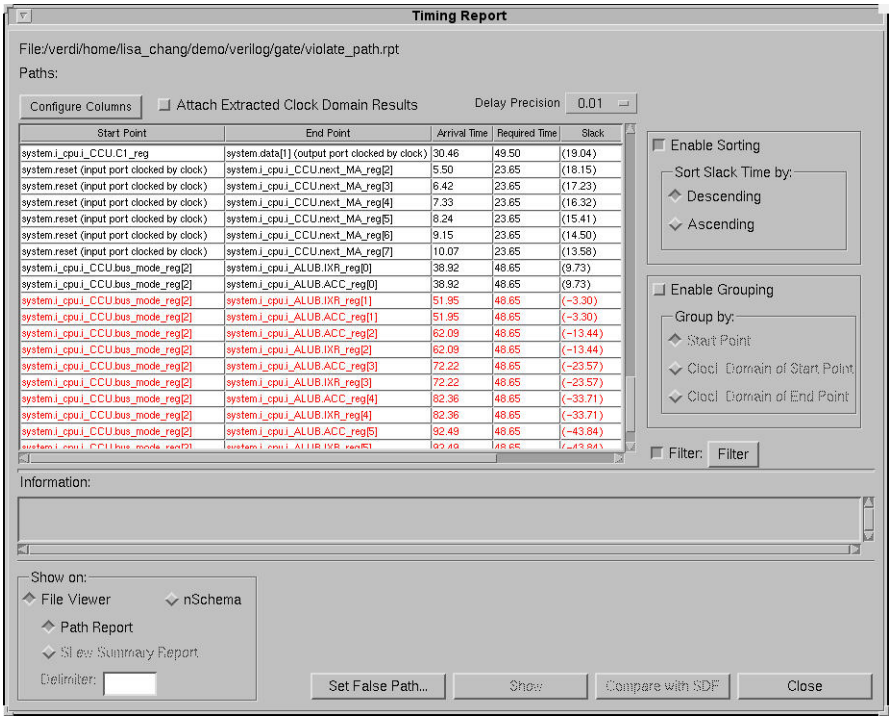


Figure: Timing Report Form

The main display area on the *Timing Report* form shows the columns of *Start Point*, *End Point*, *Arrival Time*, *Required Time,* and *Slack* of paths in the same clock tree based on the clock tree tracing result. To view the columns of *Clock Source of Start Point*, *Clock Source of End Point*, and *Clock Skew*, use the **Configure Columns** button for the column settings.

SDF

The SDF command includes the two subcommands: **Load SDF Files** and **Close SDF Files**.

Load SDF Files

Menu Bar: File -> SDF -> Load SDF Files

This command opens the *Import SDF* form where an SDF file to be imported can be specified. Refer to the **File -> SDF ->** **Load SDF Files - nAnalyzer** command in the *nTrace* chapter for details.

Close SDF Files

Menu Bar: **File -> SDF -> Close SDF Files**

This command closes the SDF file loaded by the **Load SDF Files** command.

Exit

Menu Bar: **File -> Exit**

This command closes the *Clock Tree Browser* window without making any changes.

View Commands

Clock Source Name

Menu Bar: **View -> Clock Source Name**

This toggle command turns the display of full clock source name on the root node *on* or *off*. The default is *off*.

Configure Level Columns

Menu Bar: **View -> Configure Level Columns**

This command opens the *Configure Columns* form where the display *Clock Tree* table of the *Clock Tree Browser* window can be customized.

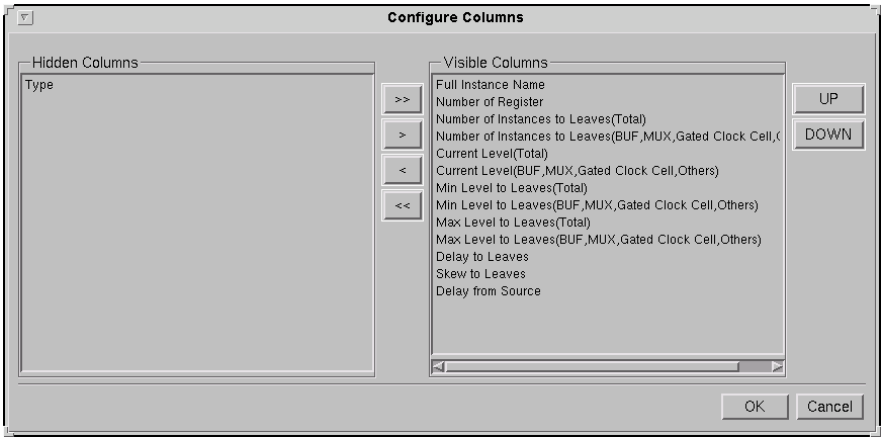


Figure: Configure Columns Form

The *Configure Columns* form has two main columns: *Hidden Columns* which lists the column(s) not to be displayed and *Visible Columns* which lists the column(s) to be displayed. Use the `>>` `>` `<` `<<` buttons to specify what to view in the *Clock Tree* table. The selections can be moved between the columns.

`>>` **:** This button moves all selections from *Hidden Columns* to *Visible Columns*.

`>` **:** This button moves the single selection from *Hidden Columns* to *Visible Columns*.

`<` **:** This button moves the single selection from *Visible Columns* to *Hidden Columns*.

`<<` **:** This button moves all selections from *Visible Columns* to *Hidden Columns*.

UP: This button moves the selected column to an upper display order. The column at the top of the list is the left-most column.

DOWN: This button moves the selected column to a lower display order. The column at the bottom of the list is the right-most column.

OK: This button applies the column settings to the *Clock Tree* table immediately.

Cancel: This button closes the *Configure Columns* form without changing the previous settings.

Collapse All Top Trees

Menu Bar:	**View -> Collapse All Top Trees**
Toolbar Icon:	
Bind Key:	A

This command collapses all clock trees shown in the *Clock Tree* table to the clock source root.

Collapse All Trees and Sub-trees

Menu Bar:	**View -> Collapse All Trees and Sub-trees**
Bind Key:	C

This command collapses all clock trees and their subtrees shown in the *Clock Tree* table to the clock source root.

Jump to Root

Menu Bar:	**View -> Jump to Root**
Toolbar Icon:	
Bind Key:	R

This command brings the selected clock tree(s) to the clock source root after a node(s) is (are) selected.

Jump to Upper Level

Menu Bar:	**View -> Jump to Upper Level**
Toolbar Icon:	
Bind Key:	U

This command brings the selected clock tree(s) one level up after a node(s) is (are) selected.

Delay Scale

Menu Bar: **View -> Delay Scale**

Use this command to select the SDF delay scale from one of the **Default**, **10fs**, **100fs**, **1ps**, **10ps**, **100ps**, **1ns**, or **10ns** options after an SDF file is loaded.

Delay Type

Menu Bar: **View -> Delay Type**

Use this command to select the SDF delay type from one of the **Minimum**, **Typical**, or **Maximum** options after an SDF file is loaded. The default is **Typical**.

Delay Precision

Menu Bar: **View -> Delay Precision**

Use this command to select the SDF delay precision from one of the **0.1**, **0.01**, or **0.001** options after an SDF file is loaded.

Trigger Type

Menu Bar: **View -> Trigger Type**

This command switches the SDF delay display type on the *Clock Tree Browser* window between falling or rising after an SDF file is loaded.

Re-calculate Delay

Menu Bar: **View -> Re-calculate Delay**

This command calculates the delay information and displays the results on the *Clock Tree Browser* window after an SDF file is loaded.

Sync. Selected Instance with nTrace

Menu Bar: **View -> Sync. Selected Instance with nTrace**

This toggle command turns the instance synchronization between *Clock Tree Browser* window and *nTrace on* or *off*. When this command is turned *on*, *nTrace* automatically selects the same instance that was previously selected in the *Clock Tree Browser* window. The default is *off*.

Tools Commands

Find

Menu Bar: **Tools -> Find**

This command opens the *Find Instance* form where the desired instance can be found in the *Clock Tree* display table of the *Clock Tree Browser* window.

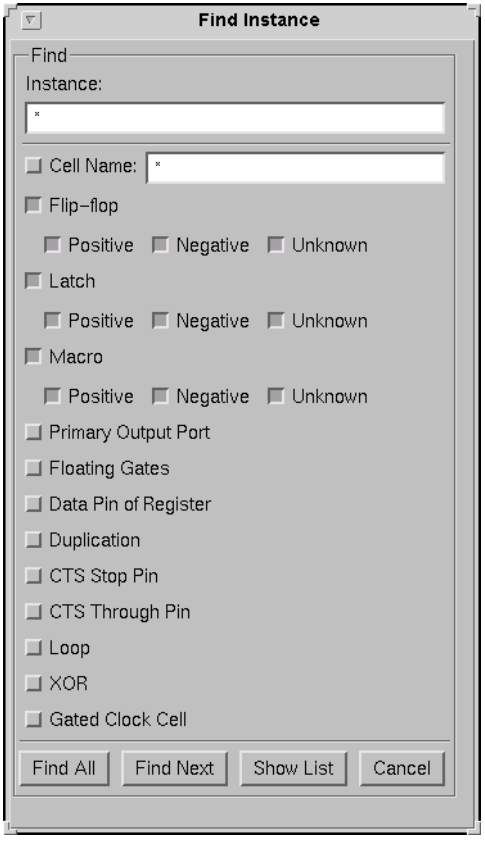


Figure: Find Instance Form

After entering an instance name in the **Instance** text field or dropping an instance dragged from *nSchema* or *nTrace* windows, click the **Find Next** button. The first instance matching the specification is highlighted in the **Clock Tree** table of the *Clock Tree Browser* window. Wildcard characters are supported. The default for the instance name is the asterisk (*) wildcard character. Additional instances are highlighted one by one.

By default, the **Find** function applies to all types. To narrow down the search, enable one or more of the **Cell Name**, **Flip-flop**, **Latch**, **Macro**, **Primary Output Port**, **Floating Gates**, **Data Pin of Register**, **Duplication**, **CTS Stop Pin**, **CTS Through Pin**, **Loop**, **XOR**, and/or **Gated Clock Cell** instance type options. For the **Flip-flop**, **Latch**, and **Macro** instance types, specify their trigger types by enabling the **Positive**, **Negative**, or **Unknown** options.

After the **Cell Name** option is enabled, type one or more module names (a space is used to separate multiple module names) in the *Cell Name* text field, or drop a module dragged from *nSchema* or *nTrace* windows.

NOTE: The relationship between these instance type options and **Instance** is AND.
The relationship between the instance type options is OR.
The relationship between the trigger type is OR.

- **Find All:** Click this button to highlight the instances that match the specifications on the *Find Instance* form in the *Clock Tree Browser* window.
- **Find Next:** Click this button to highlight the instances that match the specifications in the **Clock Tree** table of the *Clock Tree Browser* window one by one. Click the **Find Next** button again to locate the next instance.
- **Show List:** Click this button to view the results of CTS attributes of the instances that match the specifications from **Cell Name**, **FF (flip-flop)**, **Latch**, **Macro**, **Primary Output Port**, **Floating Gates**, **Data Pin of Register**, **Duplication**, **CTS Stop Pin**, **CTS Through Pin**, **Loop**, **XOR**, and/or **Gated Clock Cell** options in the *Find Nodes* tab of the *CTS Constraint Checker Result* form.

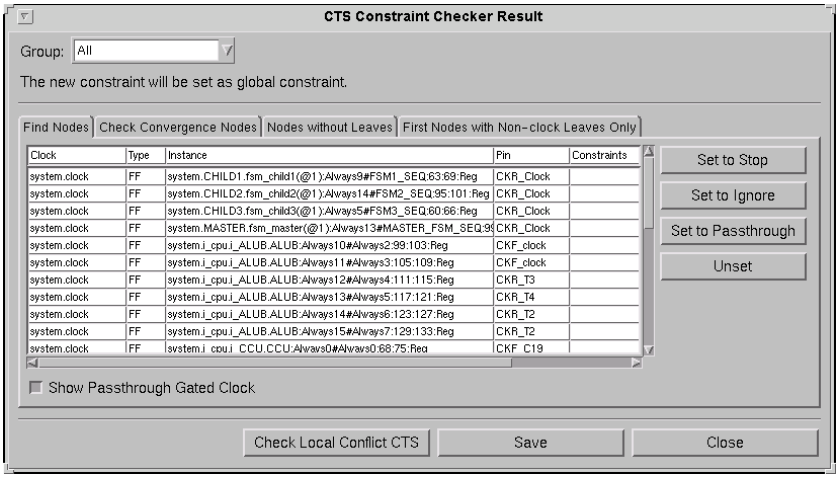


Figure: CTS Constraint Checker Result Form - Find Nodes Tab

The **Group** selection field displays all clock sources extracted during clock tree extraction. The default is **All**. Select a clock source and view the associated instances in the tree. If a desired clock source is selected from the **Group** selection field, the CTS constraint results are for local CTS settings; if **All** is selected in the **Group** selection field, the CTS constraint results are for global CTS settings.

- In the **Find Nodes** display area, **Clock**, **Type**, **Instance**, **Pin**, and **Constraint** columns can be sorted by clicking the triangle or upside-down triangle on the column headers. The **Type** column is shown on the **Comment** column as an instance type in **Stop Type**, **Ignore Type,** and/or **Passthrough Type** table of the **CTS Attributes** tab in the *Clock Extraction Settings* form.

  Select an instance and set its constraint type by clicking **Set to Stop**, **Set to Ignore**, or **Set to Passthrough** buttons. Remove the constraint by clicking the **Unset** button. The settings are shown in the **Constraints** column of the main display area.

  When the **Show Passthrough Gated Clock** option is turned *on*, all gated clocks are shown, even though the clocks are set as passthrough. When this option is turned *off*, gated clocks set as passthroughs are not shown in the *CTS Constraint Checker Result* form.

  The **Check Local Conflict CTS** button checks for setting conflicts between the local CTS settings. Refer to the **Check Local Conflict CTS** button description in the **Global CTS Subtab** of the *Clock Extraction Settings* form for details.

Click the **Save** button to open a form where the directory structure can be viewed and a file can be specified to save the CTS attributes.

- **Cancel:** Click this button to close the *Find Instance* form without making any changes.

Set Marker

Menu Bar: **Tools -> Set Marker**

Toolbar Icon: 

After an SDF file is loaded, select an instance on the main display area in the *Clock Browser Tree Window* and then invoke the **Set Marker** command under the **Tools** menu or click the  icon on the toolbar to specify the instance as a base point. The row specified as the source/base point is highlighted in yellow. After another non-summary row (an instance without 'Summary' in the folder description, such as ) is highlighted on the main display area in the *Clock Browser Tree Window*, delay and level differences between the two instances is calculated and displayed in the area next to the  icon. The delay and the level information is updated each time another non-summary row is selected.

Unset Marker

Menu Bar: **Tools -> Unset Marker**

This command removes the marker set previously from the **Set Marker** command. If the marker needs to be set for a new row, the current marker must be unset first and the marker must be set again.

New Schematic

Menu Bar: **Tools -> New Schematic**

Toolbar Icon: 

After selecting either a single clock domain (left-click) or multiple clock domains (Ctrl-left-click), invoke this command to display the selected clock domain(s) in a flattened schematic window.

New Insertion Delay Histogram

Menu Bar: **Tools -> New Insertion Delay Histogram**

Toolbar Icon: 

After an SDF file is loaded, select no more than two non-summary rows (an instance without 'Summary' in the folder description, such as 

) on the main display area in the *Clock Browser Tree* window and then invoke the **New Insertion Delay Histogram** command to open a statistics window where the insertion delay is shown in the histogram format. Delay values of leaf nodes under the selected node of the current clock tree are calculated and classified on the statistics window. Non-leaf and non-stop point nodes are not included.

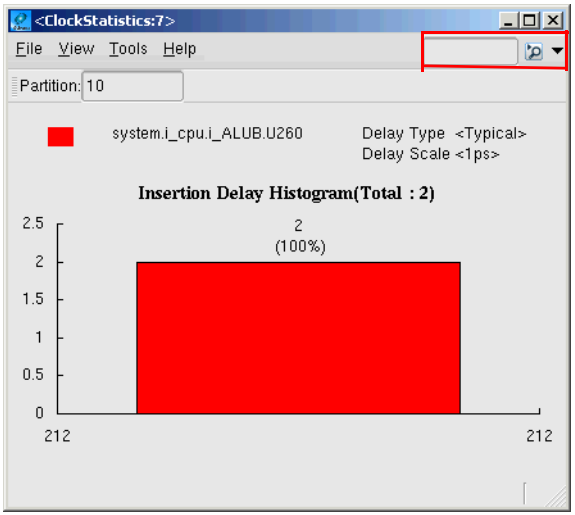


Figure: Statistics Window

The X-axis indicates the insertion delay value and the Y-axis indicates the flip-flop number. Bar(s) can be dragged and dropped into the *Clock Tree Browser* window or the *nSchema* window where the instances with insertion delay values in the bar are highlighted.

The *Statistics Window* frame can become a standalone window by clicking the **Undock** icon on the toolbar. Refer to the *Icons for Dockable Frames* section for details on the menu bar icons.

The *Statistics Window* opened by the **Tools -> New Insertion Delay Histogram** command includes the following commands:

File Commands

The **File** commands include the following subcommands:

Capture Window

Refer to the **File -> Capture Window** command description in the *nSchema* chapter for details.

Close

This command closes the *Statistics Window*.

View Commands

The **View** commands specify the preferred viewing setting for delay scale, delay type, and delay precision of the SDF in the X-axis of the statistics window. Default values of delay scale, delay type, and delay precision of the statistics window are from the SDF setting in the *Clock Tree Browser* window.

The **View** commands include the following subcommands:

Delay Scale

Use this command to select the SDF delay scale from one of the options: **Default**, **10fs**, **100fs**, **1ps**, **10ps**, **100ps**, **1ns**, or **10ns**.

Delay Type

Use this command to specify the SDF delay type from one of the **Minimum**, **Typical**, or **Maximum** options.

Delay Precision

Use this command to specify the SDF delay precision from one of the options: **0.1**, **0.01**, or **0.001**.

Insertion Type

This command provides two insertion types for delay value: **Delay from Source** and **Delay from Level**. **Delay from Source** calculates the delay value from the source node of the clock tree to each leaf node. **Delay from Level** calculates the delay value from the selected node of the clock tree to each leaf node. One of the two options must be selected. By default, **Delay from Source** is selected.

Histogram Title

This command turns the display of the histogram title display in the *Statistics Window on* or *off*. The number in parentheses represents the total number of flip-flops from the selected node to the left node.

Clock Source Label

This command turns the display of the clock source name on the Y-axis of the *Statistics Window on* or *off*.

Delay Type/Scale Label

This command turns the display of delay type and delay scale on the X-axis of the *Statistics Window on* or *off*.

List CTS Stop

When this command is turned *on*, CTS stop pins are treated as leaves of the specified clock source, and they are counted in the bar chart of the *Statistics Window*. When this command is turned *off*, CTS stop pins are filtered in the *Statistics Window*.

Tools Commands

The **Tools** commands include the following subcommands:

Color Preferences

This command opens the *Color Preference* form where the color for bars, the selected object, and the background for all statistics windows can be set. After the **OK** button is clicked in the *Color Preference* form, the new settings apply to all statistic windows immediately and are logged into the *novas.rc* resource file as the default value.

Customize Menu/Toolbar

Refer to the **Tools -> Customize Menu/Toolbar** command in the *nTrace* chapter for details.

The following toolbar icon is available for the *Statistics Window*:

- **Partition**  **:** Enter a value in this field to set the partition between instance bars on the X-axis, and the range of bars are calculated automatically. The default value is *10*.

New Level to Leaves Histogram

Menu Bar: **Tools -> New Level to Leaves Histogram**

Toolbar Icon: 

After one or two nodes are selected, invoke the **New Level to Leaves Histogram** command to open a statistics window where the X-axis shows level numbers (current level of leaves minus the current level of the selected nodes) and the Y-axis shows the total instance number from the leaf to the selected nodes.

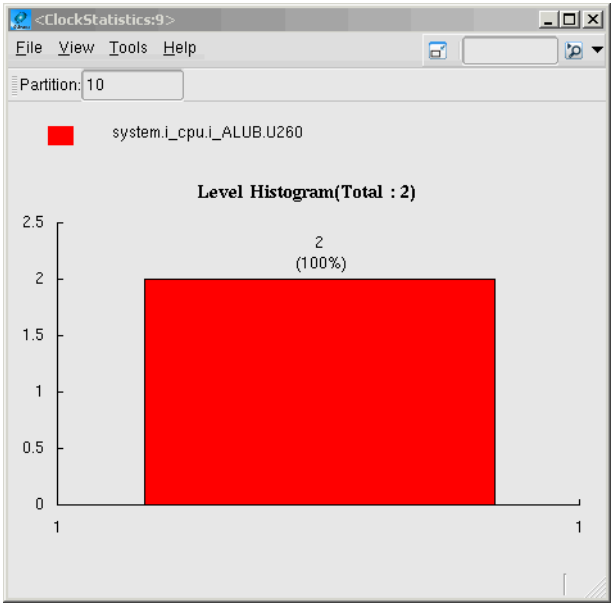


Figure: Statistics Window

For the pull-down commands in the *Statistics Window*, refer to the **New Insertion Delay Histogram** command description for details.

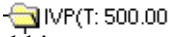
Clock Skew

Refer to the **Trace -> Clock Tree -> Clock Skew - nAnalyzer** command description in the *nSchema* chapter for details.

Set Insertion Delay Violation Threshold

Menu Bar: **Tools -> Set Insertion Delay Violation Threshold**

Toolbar Icon: 

This command specifies a threshold value for the insertion delay. After an SDF file is loaded, left-click to select a non-summary node (an instance without 'Summary' in the folder description, such as ) and invoke this command or click the  icon on the toolbar to open the *Set Insertion Delay Threshold* form where a value to set as the threshold can be entered. The input value of the threshold must be a positive integer or a floating point number greater than zero. After the **OK** button is clicked in the *Set Insertion Delay Threshold* form, parentheses including T and a number (such as ) is shown in the non-summary node. T indicates that the threshold is set successfully and the number indicates the value specified in the *Set Insertion Delay Threshold* form.

The threshold setting is reset after a clock tree is recreated.

Insertion Delay Violation Check

Menu Bar: **Tools -> Insertion Delay Violation Check**

Toolbar Icon: 

After the threshold is set using the **Set Insertion Delay Violation Threshold** command, invoke this command or click the  icon on the toolbar to find the violation points. The “Delay from Source” value for violation instances is compared to the base point; if the result is greater than the threshold delay, the instance is identified and highlighted in red in the *Clock Tree Browser* window.

Find Previous Violation Instance

Menu Bar: **Tools -> Find Previous Violation Instance**

Toolbar Icon: 

If any violation instances are found, this command is enabled. Invoke this command or click the toolbar icon to find the previous violation instance in the *Clock Tree Browser* window. The target violation instance is highlighted.

Find Next Violation Instance

Menu Bar: **Tools -> Find Next Violation Instance**

Toolbar Icon: 

If any violation instances are found, this command is enabled. Invoke this command or click the toolbar icon to find the next violation instance in the *Clock Tree Browser* window. The target violation instance is highlighted.

Show Path Information

Menu Bar: **Tools -> Show Path Information**

This command opens the *User-defined Level Information* form where the user-defined cells can be set.

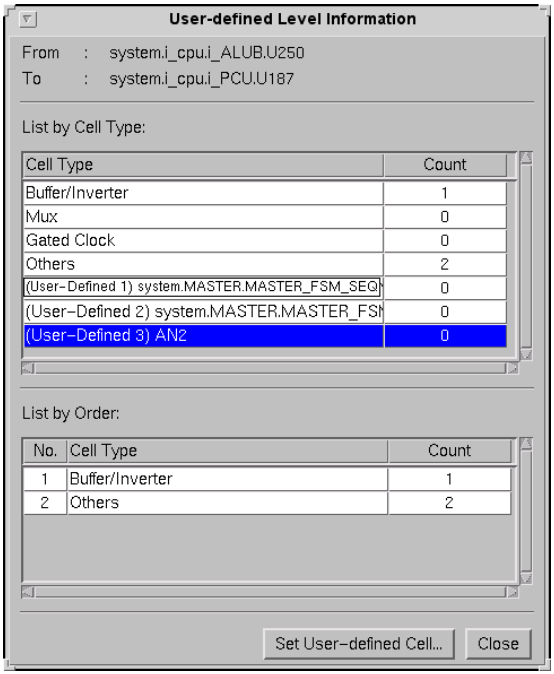


Figure: User-defined Level Information Form

In the *User-defined Level Information* form, **From** displays the source clock tree and **To** displays the instance name of the selected node.

The *List by Cell Type* display area shows the cell types and the total number of cell types found from the clock source to the selected node. The cell type includes **Mux**, **Buffer/Inverter**, **Gated Clock**, and **Others** (for the remaining cell types). Set the user-defined cells from the **Others** type. When the **Others** type is selected, the number in the **Others** type is decreased by one and the **User-defined** cell is added.

The **List by Order** display area shows detailed level information from the clock source to the selected node by order as each cell type is encountered in the path.

The **Set User-defined Cell** button opens the *Set User-defined Cell Form*.

The **Close** button closes the *User-defined Level Information* form without making any changes.

Set User-defined Cell Form

View and edit the user-defined cell in the display area of the *Set User-defined Cell* form.

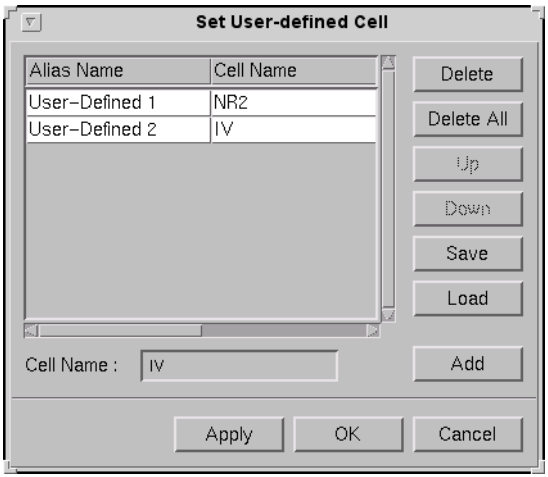


Figure: Set User-defined Cell Form

After dragging an instance from the *nSchema* window and dropping it to the *Cell Name* text field or entering the name manually, click the **Add** button to specify the user-defined cell. The cell name does not need to include a full hierarchical path. The specified cell is shown on the display area of the *Set User-defined Cell* form. After the cell is added, left-click in the **Alias Name** column to change the default alias name.

The following buttons are available to perform different actions in the *Set User-defined Cell* form:

- **Delete:** Click this button to remove the selected user-defined cell from the display area.
- **Delete All:** Click this button to delete all user-defined cells from the display area.
- **Up:** Click this button to move the selected cell up one row each time the button is clicked. The order in the user-defined cell table of the *Set*

User-define Cell form corresponds to the order in the *List by Cell Type* table of the *User-defined Level Information* form.

- **Down:** Click this button to move the selected cell down one row each time the button is clicked. The order in the user-defined cell table of the *Set User-define Cell* form corresponds to the order in the *List by Cell Type* table of the *User-defined Level Information* form.
- **Save:** Click this button to open the *Save File* form to view the directory structure and specify a file to save the current settings to.
- **Load:** Click this button to open the *Open File* form to view the directory structure to load a previously saved file to the display area.
- **Add:** Click this button to add the user-defined cell to the display area.

CTS Constraint Checker

Menu Bar: **Tools -> CTS Constraint Checker**

This command opens the *Checks CTS Constraint* form which includes **Find Nodes**, **Check Convergence Nodes**, **Nodes without Leaves**, and **First Nodes with Non-clock Leaves Only** tabs.

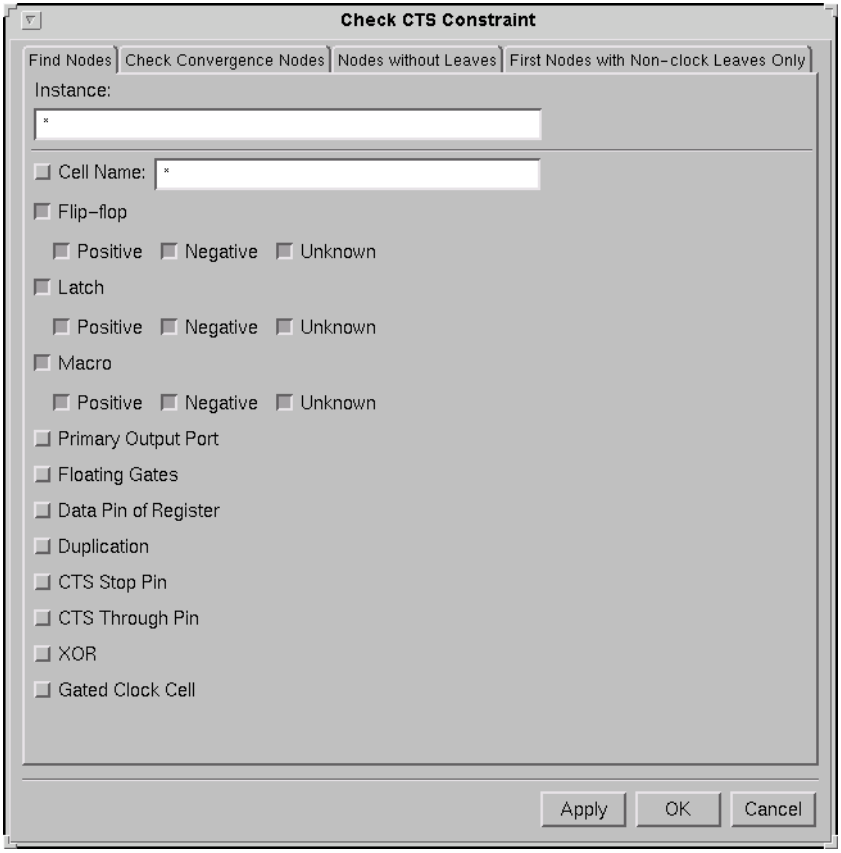


Figure: Check CTS Constraint Form - Find Nodes Tab

- **Find Nodes Tab**

  Find nodes in the specified clock source from settings in this tab.

  Type an instance name in the *Instance* text field (wildcard characters are supported) or drop an instance dragged from *nSchema* or *nTrace* windows, and then click **Apply** or **OK** button. The instance matching the specification is shown in the *Find Nodes* tab of the *CTS Constraint Checker Result* form. To narrow down the search, enable one or more of the **Cell Name** (after enabling this option, type a module name in the text field or drop a module dragged from the *nSchema* or *nTrace* windows), **Flip-flop**, **Latch**, **Macro**, **Primary Output Port**, **Floating Gates**, **Data Pin of Register**, **Duplication**, **CTS Stop Pin**, **Loop**, **XOR**, and/or **Gated Clock Cell** instance type options. For **Flip-flop**, **Latch**, and **Macro** instance types,

specify their trigger types by enabling **Positive**, **Negative**, or **Unknown** options.

NOTE: The relationship between the instance type options and **Instance** is AND.
The relationship between the instance type options is OR.
The relationship between the trigger type is OR.

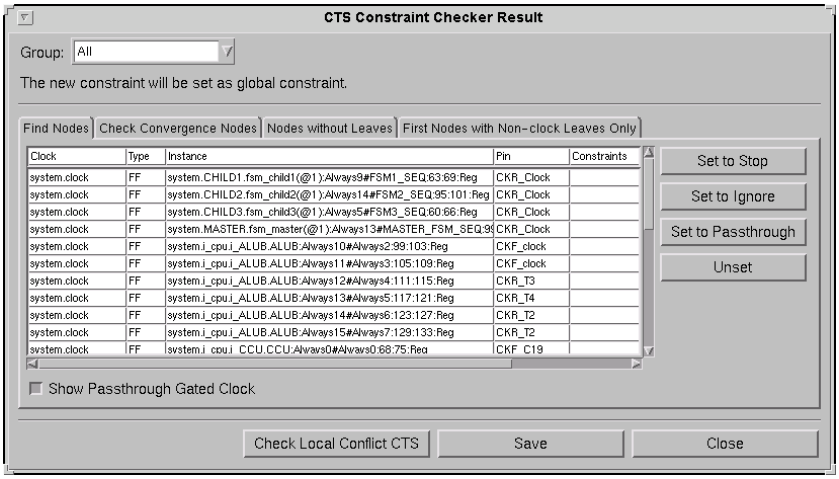


Figure: CTS Constraint Checker Result Form - Find Nodes Tab

The *Group* selection field lists all the extracted clock sources. Specify a clock source to view the found results by selecting a desired clock source from the *Group* selection field for local CTS settings or by selecting **All** in the *Group* selection field for global CTS settings.

Select an instance and set its constraint type by clicking **Set to Stop**, **Set to Ignore**, **Set to Passthrough**, or **Unset** button. The settings are shown in the *Constraints* column of the main display area.

When the **Show Passthrough Gated Clock** option is turned *on*, all gated clocks are shown, even though the clocks are set as passthrough. When this option is turned *off*, gated clocks set as passthroughs are not shown in the *CTS Constraint Checker Result* form.

- **Check Convergence Node Tab**

  When more than one clock tree is displayed in the *Clock Tree Browser* window (multiple clock sources must be selected in the *Clock Extraction Settings* form), this tab can check convergence node(s) for multiple clock roots.

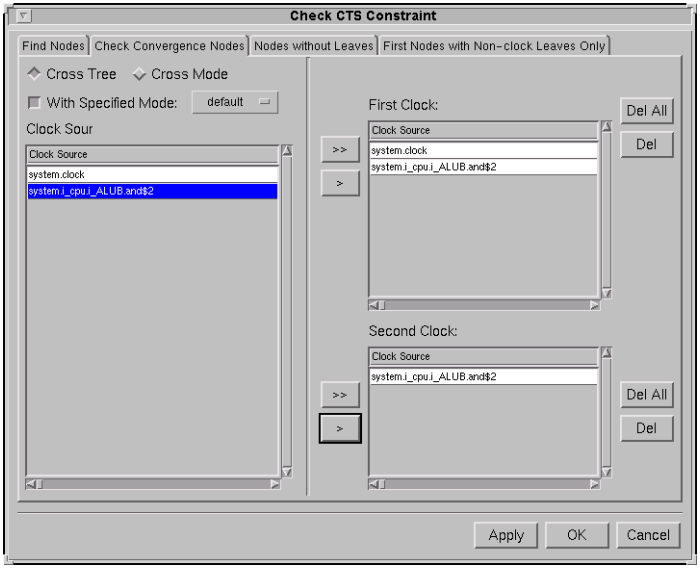


Figure: Cross Tree to Check Convergence Nodes

When the **Cross Tree** option is selected, the selected clock sources are displayed in the *Clock Source List* area. Highlight the desired clock source and input it to **First Clock** and **Second Clock** areas with the >> > buttons.

When the **Cross Tree** option is selected, the **With Specified Mode** option can also be turned *on* to select a mode, the specified mode applies to both the first clock and the second clock in the convergence check. The **With Specified Mode** option is *off* by default.

Refer to the *Mode Form* section in the *Clock Extraction Settings* form for details about the mode settings.

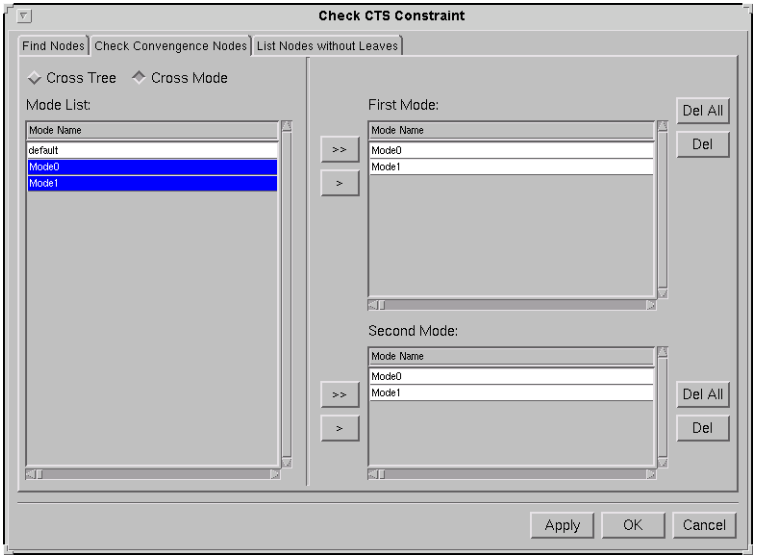


Figure: Cross Mode to Check Convergence Nodes

If the **Cross Mode** option is selected, the modes set with the **Mode** button on the *Options* tab in the *Clock Extraction Settings* form are listed on the *Mode List* area. Select the desired mode and input it to the **First Mode** and the **Second Mode** areas with the >> > buttons.

After **Apply** or **OK** buttons are clicked, convergence node results are shown in the *Check Convergence Nodes* tab of the *CTS Constraint Checker Result* form.

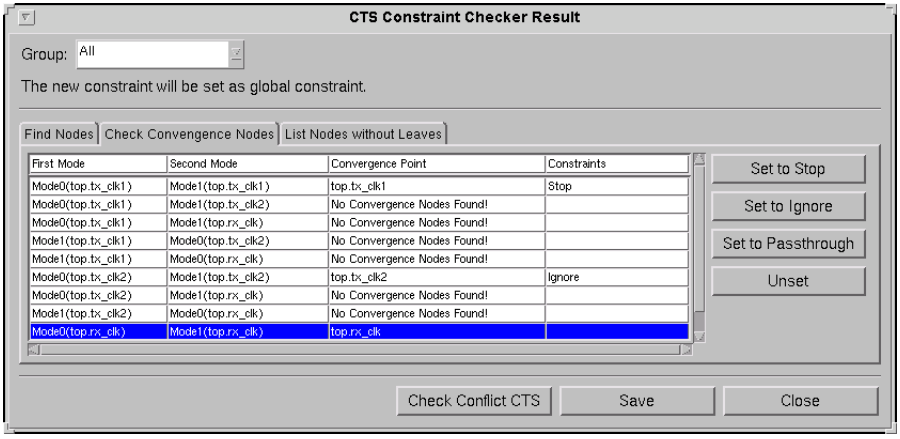


Figure: CTS Constraint Checker Result Form - Check Convergence Nodes Tab

Select a convergence point and use the middle mouse button to drag and drop it in the *nSchema* window to highlight the signal including the convergence point.

Alternatively, select a convergence point and set its constraint type by clicking **Set to Stop**, **Set to Ignore**, **Set to Passthrough**, or **Unset** buttons. The settings are shown in the *Constraints* column of the main display area.

- **Nodes without Leaves Tab**

  If any instances or ports are set as 'ignore' in the *CTS Attributes* tab of the *Clock Extraction Settings* form, and the **List All Nodes without Leaf Points Resulting from the Ignore Setting** option is turned *on*, clicking the **OK** button in the *Check CTS Constraint* form displays nodes without leaf points on the *Nodes without Leaves* tab of the *CTS Constraint Checker Result* form. By default, the **List All Nodes without Leaf Points Resulting from the Ignore Setting** option is *off*.

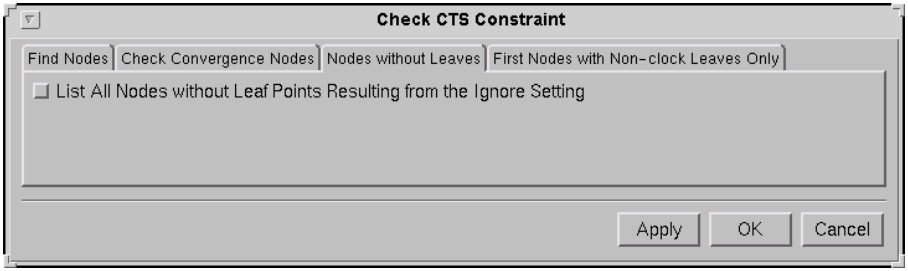


Figure: Check CTS Constraint Form - Nodes without Leaves Tab

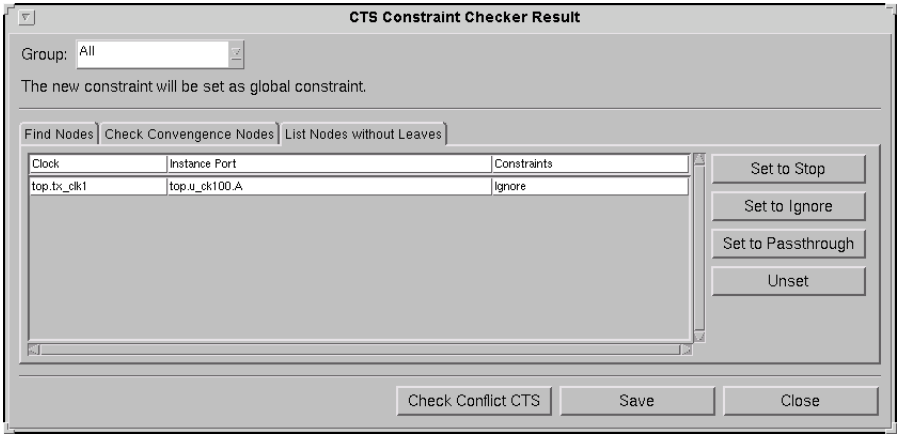


Figure: CTS Constraint Checker Result Form - Check Convergence Nodes Tab

On the *Nodes without Leaves* tab of the *CTS Constraint Checker Result* form, the table shows the ignored instance port information. The **Clock**

column shows the clock source of the selected instance port. The **Instance Port** column shows the ignored instance port name. Set the constraint type for the selected instance by clicking **Set to Stop**, **Set to Ignore**, **Set to Passthrough**, or **Unset** buttons. The settings are shown in the **Constraint** column of the table.

The selected instance port on the *Nodes without Leaves* table can be dragged and dropped into the *nSchema* window. If the instance port already exists in the *nSchema* window, the object is highlighted. If the instance port does not exist, it is added. If a selection is dragged from the table and dropped into the *Clock Source* table of the *Clock Extraction Settings* form, the instance is shown as an instance port name. Multiple objects can be selected for drag and drop.

- **First Nodes with Non-clock Leaves Only Tab**

  The options in this tab list the root nodes with leaf points that are not connected to the clock input of a sequential cell. “First nodes with non-clock leaves only” refers to the first cell in the clock tree that drives only to the non-clock leaves. When the **List First Nodes Driving Only to Non-clock Leaves** option is turned *on* and the **Apply** or **OK** button is clicked, instance ports matching the specification are listed in the display area of the **First Nodes with Non-clock Leaves Only** tab on the *CTS Constraint Checker Result* form.

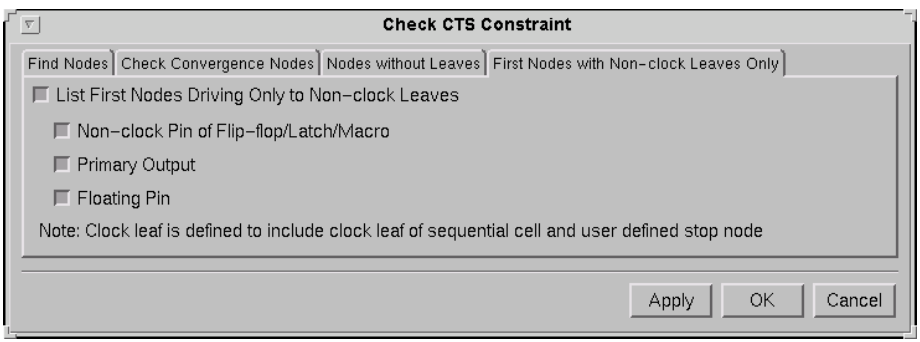


Figure: Check CTS Constraint Form - First Nodes with Non-clock Leaves Only Tab

The **List First Nodes Driving Only to Non-clock Leaves** option is *off* by default. **Non-clock pin of Flip-flop/Latch/Marco**, **Primary Output**, and **Floating Pin** options are for intended non-clock leave types.

Highlight Constraint Points

Menu Bar: **Tools -> Highlight Constraint Points**

This command opens the *Highlight Constraint Points* form where colors for the pin and its instance with stop points, through points, convergent points, and exception type points can be specified and highlighted in the extracted clock tree in the *nSchema* window. The default for all the **Enable Highlight** options of **Stop Points**, **Through Points**, **Convergent Points**, and **Exception Type Points** is *off*.

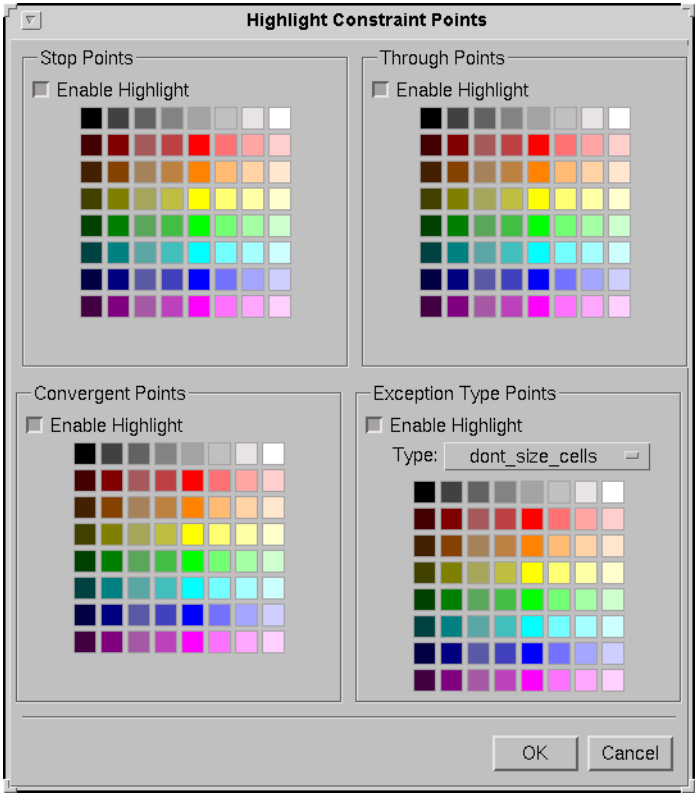


Figure: Highlight Constraint Points Form

Preferences

Menu Bar: **Tools -> Preferences**

This command opens the *Preferences* form with the **Schematics** page -> **Clock Folder** selected for specifying the highlight colors for marker, selected row, and violation points in the *Clock Tree Browser* window.

Customize Menu/Toolbar

Menu Bar: **Tools -> Customize Menu/Toolbar**

Refer to the **Tools -> Customize Menu/Toolbar** command in the *nTrace* chapter for details.

Clock Tree Browser Window Frame Right-Click Options

When the right mouse button is clicked anywhere in menu, toolbar icon, or frame banner area of the *Clock Tree Browser Window* frame or standalone window, a configuration option menu is displayed.This menu can be used to configure the available icons and frames.

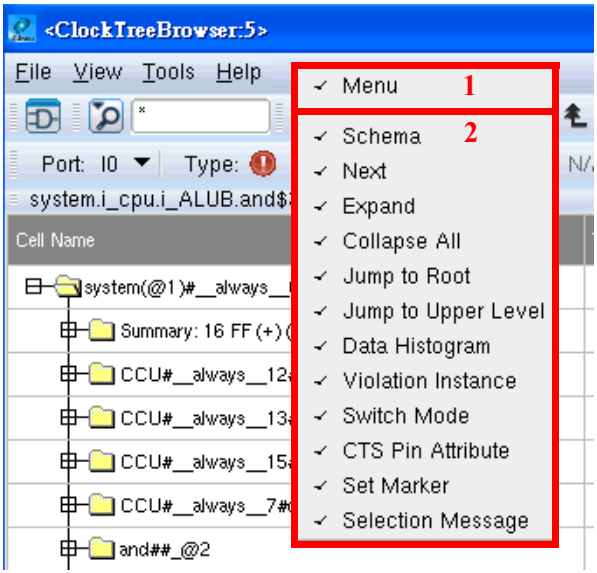


Figure: Configuration Option Menu

The **Menu** option (labeled 1 in the above figure) enables/disables the command menu bar. The options in the bottom section (labeled 2 in the above figure) enable/disable the toolbar icons for different functions.

Clock Tree Browser Window Body Right-Click Option

The right-click command in the *Clock Tree Browser* window body include the following:

Show Tip

When the right mouse button is clicked on the *Cell Name* column of the *Clock Tree Browser* window, the **Show Tip** option is displayed to view column details, including Cell Name, Type, Full Instance Name, Number of Registers, Total Number of Instances to Leaves, Number of Instances to Leaves, Total Current Level Number, Current Level, Total Min Level to Leaves, Min Level to Leaves, Total Max Level to Leaves, Max Level to Leaves, Delay from Source, and Delay to Leaves. The default value for the **Show Tip** option is *off*.

Clock Tree Browser Window Toolbar Icons and Fields

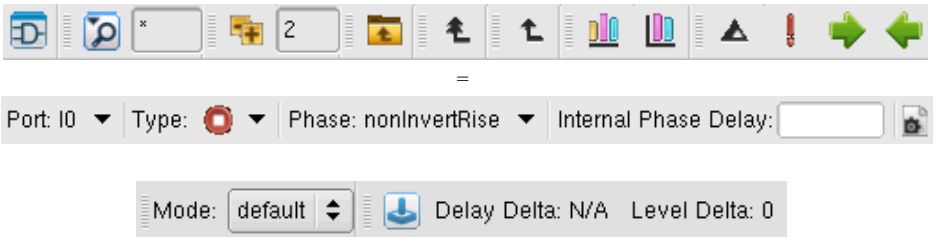


Figure: Toolbar Used in Clock Tree Browser Window

The available toolbar icons may be modified. Refer to the *Toolbars* section of the *User Interface* chapter in the *Verdi User Guide and Tutorial* manual for details.

Schema Category

New Schematic 

Refer to the **Tools -> New Schematic** command description for details.

Next Category

Find Next Instance 

Refer to the **Tools -> Find** command description for details.

Expand Category

Expand 

Enter a number in the text field to expand levels of the collapsed tree.

Collapse All Category

Collapse All Top Trees 

Refer to the **View -> Collapse All Top Trees** command description for details.

Jump to Root Category

Jump to Root 

Refer to the **View -> Jump to Root** command description for details.

Jump to Upper Level Category

Jump to Upper Level 

Refer to the **View -> Jump to Upper Level** command description for details.

Data Histogram Category

New Insertion Delay Data Histogram 

Refer to the **Tools -> New Insertion Delay Histogram** command description for details.

New Level to Leaves Data Histogram 

Refer to the **Tools -> New Level to Leaves Histogram** command description for details.

Violation Instance Category

Set Insertion Delay Violation Threshold 

Refer to the **Tools -> Set Insertion Delay Violation Threshold** command description for details.

Insertion Delay Violation Check 

Refer to the **Tools -> Insertion Delay Violation Check** command description for details.

Find Next Violation Instance 

Refer to the **Tools -> Find Next Violation Instance** command description for details.

Find Previous Violation Instance 

Refer to the **Tools -> Find Previous Violation Instance** command description for details.

CTS Pin Attribute Category

The CTS (Clock Tree Synthesis) pin attributes can be input in the CTS Pin Attribute section.

Port 

After a clock tree instance is selected in the clock tree table, all ports of the selected instance are listed in the **Port** selection field. Select the port of interest to specify its type.

Type 

Specify the type for the selected port. The type selections include **Stop**, **Ignore**, **Passthrough**, and **No Attribute**. See the **CTS Attributes Tab** section on the *Clock Extraction Settings* form for details about the Stop type, the Ignore type, and the Passthrough type. **No Attribute** indicates that no type is specified. The default value is **Stop**.

Phase 

Specify the phase when the type is **Stop**. The phase selections include **nonInvertRise**, **nonInvertFall**, **invertRise**, and **invertFall**. These options are not available with other pin attributes. The default value is **invertRise**.

Internal Phase Delay 

Enter an integer in the text field to specify the internal phase delay when the type is **Stop**.

Settings 

Click the **Settings** icon to open the *Clock Extraction Settings* form where all the clock sources in the current *Clock Tree Browser* window are loaded into the *Clock Sources* table on the *Clock Sources* tab.

Switch Mode Category

Mode 

Switch among different modes to see the clock tree results when different modes of constant settings are available.

Set Marker Category

Set Marker 

This icon and field is shown when a clock tree node is selected. Refer to the **Tools -> Set Marker** command description for details.

Selection Message Category

Selection Message

This field displays information about the selected object.

Crossing Paths Window

The *Crossing Paths Window* frame is displayed when **Tools -> Clock Analyzer -> Check Crossing Paths** or **Tools -> Clock Analyzer -> List Crossing Paths** commands are invoked from the *nTrace* window or the **Trace -> Clock Analyzer -> Check Crossing Paths** or the **Trace -> Clock Analyzer -> List Crossing Paths** commands are invoked from the *nSchema* frame. The **Tools -> Check Crossing Paths** or **Tools -> List Crossing Paths** commands in the *Clock Domains* window can also invoke the *Crossing Paths* window.

The *Crossing Paths* window displays the crossing path results and the corresponding CDC correlations and register pair details.

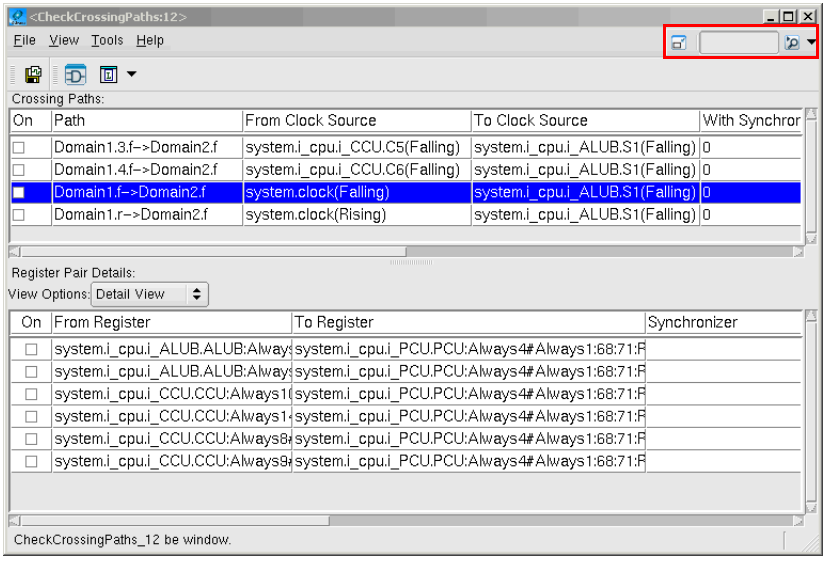


Figure: Check Crossing Paths Window

The *Crossing Paths* window includes three sections: **Crossing Paths**, **CDC Correlation Details**, and **Register Pair Details**.

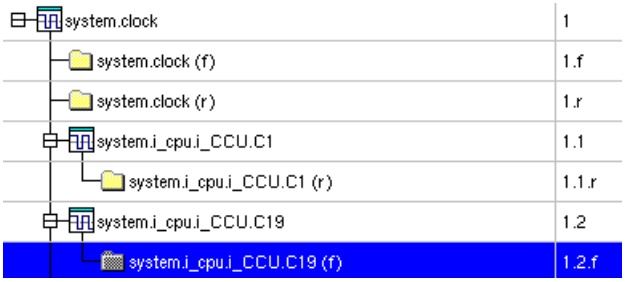
- **Crossing Paths Section**

  The table in the **Crossing Paths** section displays the crossing path results by clock domain pairs. You can change the way the results are sorted and displayed by clicking the column headers. The width of each column can be adjusted by placing the cursor over the vertical bar in the heading row and then pressing and holding the left mouse button. The column headings are summarized below.

- **On:** The check indicates if the selected crossing path is used or not. Use **View -> Hide Selection** or **View -> Show Selection** commands to view crossing paths of interest.
- **Path:** Indicates the crossing paths between domains.

NOTE: The entire clock tree (all nodes including sources and domains under the clock root) is defined as “clock domain.” All domains under the clock root are defined as “subclock domains.”

A number for all tree nodes is displayed. All domains and sub-domains are displayed in hierarchical layers where the source clock signal and generated clock signal can easily be found.



Using the blue highlights as an example, **1** indicates that it is clock source number **1** (based on the hierarchy order in the tree). **2**, attached to **1**, indicates that it is a subclock domain of the first clock source. **f** indicates trigger type falling. **r** indicates rising.

- **From Clock Source:** Indicates the 'from clock source' of each clock domain.
- **To Clock Source:** Indicates the 'to clock source' of each clock domain.

  The **To/From Clock Source** columns also indicate the trigger type (**Rising** or **Falling**) of each clock domain. **Rising** indicates rising edge trigger for flip-flops and level high for latches. **Falling** indicates falling edge trigger for flip-flops and level low for latches.
- **With Synchronizer:** Lists the number of register pairs with synchronizer for each crossing path in the *Crossing Paths* table.
- **Without Synchronizer:** Lists the number of register pairs without synchronizer for each crossing path in the *Crossing Paths* table.
- **Convergence:** Summarizes the total number of convergence problem types listed in the *CDC Correlation Details* table.
- **Divergence:** Summarizes the total number of divergence problem types listed in the *CDC Correlation Details* table.

- **Reconvergence:** Summarizes the total number of re-convergence problem types listed in the *CDC Correlation Details* table.
- **Filtered/Colored Register Pair Num.:** Shows the number of filtered register pairs before the forward slash (/) character and shows the number of colored register pairs after the forward slash. For example, after setting filter rules from the **Tools -> Filter** command, if five register pairs are filtered and four register pairs are highlighted, the *Filtered/Colored Register Pair Num.* column shows 5/4.

  To view this column, **Filtered/Colored Register Pair Num.** must be moved from **Hidden Columns** field to **Visible Columns** field in the *Configure Columns* form invoked by the **View -> Configure Crossing Path List Columns** command. This column is in the **Hidden Columns** list by default.

- **CDC Correlation Details**

  To view the *CDC Correlation Details* section, two conditions must be met. Firstly, a synchronizer must be specified in the **Edit** option on the *Synchronizer SCDC Correlation Detail Settings* tab of the *Check Crossing Paths Form*. Refer to the *Edit Form* section for details.

  Secondly, at least one of **Convergence**, **Divergence**, or **Reconvergence** options on the *Check Crossing Paths Form* must be turned *on*.

  The table displays the cross domain checking (CDC) correlation results for the crossing path(s) selected in the **Crossing Paths** table. You can change the way the results are sorted and displayed by clicking the column headers. Results for **Problem Type** and **Instance/Signal Name** are sorted alphabetically and **Path Number** is sorted in an increasing order.

  After left-clicking to select and highlight a row, invoke the **New Schematic** command or the  icon to display a schematic view.

  The column headings are summarized below.

  - **On:** The check indicates if the selected CDC correlation is used or not. Use **View -> Hide Selection** or **View -> Show Selection** commands to view register pair details of interest.
  - **Problem Type:** Identify the type of **Convergence** (convergence before or after synchronization), **Divergence** (divergence before synchronization) and **Reconvergence** (re-convergence after synchronization).
  - **Instance/Signal Name:** Indicates the convergence or re-convergence cell name if the problem type is **Convergence** or **Reconvergence**. Indicates the divergence signal name if the problem type is **Divergence**.

- **Path Number:** Indicates the crossing path number.

- **Register Pair Details**

  The table in the **Register Pair Details** section displays register pair details for each crossing path.

  The **View Options** selection menu provides three display modes:

  - **Detail View:** Shows all information, including *Synchronizer, Level, From Register, To Register, From Register Data Signal, and To Register Data Signal* columns (depending on the settings in the **View -> Configure Register Pair Columns** command) in the *Register Pair Details* section. The default is **Detail View**.
  - **Fan-in View:** Shows the fan-in number of all to-registers in the *Register Pair Details* section.

    If this mode is selected, the **View -> Configure Register Pair Columns** command is disabled.
  - **Fan-out View:** Shows the fan-out number of all from-registers in the *Register Pair Details* section.

    If this mode is selected, the **View -> Configure Register Pair Columns** command is disabled.

NOTE: The selection in the **View Options** menu simultaneously affects the settings in the *Filter Rule Settings* form (opened by clicking the **Add** option of the *Filter* form invoked by the **Tools -> Filter** command in the *Crossing Paths* window).
In the **Fan-in View**, only the **From Register** and **Fan-in Number** options are enabled in the *Filter Rule Settings* form; in the **Fan-out View**, only the **From Register** and **Fan-out Number** options are enabled in the *Filter Rule Settings* form.

NOTE: The **Tools -> New File Viewer -> Long List** and **Tools -> New File Viewer -> Short List** commands are not available in the **Fan-in View** and **Fan-out View**, because register pair(s) have not been selected. Select register pair(s) before enabling **Tools -> New File Viewer -> Long List** and **Tools -> New File Viewer -> Short List** commands.

After left-clicking to select and highlight a row, invoke the **New Schematic** command or the  icon to display a schematic view. Alternatively, double-click any row to automatically show the selected path in the *nSchema* window. To change the way the results are sorted and displayed, click the column headers. The width of each column can be adjusted by placing the cursor over the vertical bar in the heading row and then pressing

and holding the left mouse button. The column headings are summarized below.

- **On:** The check indicates if the selected register pair is used or not. Use **View -> Hide Selection** or **View -> Show Selection** commands to view register pair details of interest.
- **From Register:** Lists the register names for the source clock domains.
- **To Register:** Lists the register names for the destination clock domains.
- **Synchronizer:** Displays the synchronizer specified in the *Check Crossing Paths* form.
- **Level:** The total number of instances in the longest path.
- **From Register Data Signal:** Lists the output data signal of the **From Register**.
- **To Register Data Signal:** Lists the output data signal of the **To Register**. If no output ports exist for a **To Register**, use the input data signal of the **To Register** as its **To Register Data Signal**.

NOTE: When a selected register pair is dragged and dropped into *nSchema* or *nTrace* windows, the selected **From Register** is highlighted in *nSchema* or *nTrace*.

The *Crossing Paths Window* frame can become a standalone window by clicking the **Undock** icon on the toolbar. Refer to the *Icons for Dockable Frames* section for details on the menu bar icons.

The menu bar for the *Crossing Paths* window is shown below:

Figure: Crossing Paths Window Menu Bar

The pull-down menus for the *Crossing Paths* window are summarized and explained on the following pages.

File Commands

Save Clock DB

Menu Bar: File -> Save Clock DB

Toolbar Icon: 

This command opens the *Save Clock DB* form where the directory structure can be viewed and a file specified to save the clock extraction results to. The default extension (and format) is *.cdb* (Clock Database) file.

Exit

Menu Bar: **File -> Exit**

This command closes the *Crossing Paths* window.

View Commands

Configure Crossing Path List Columns

Menu Bar: **View -> Configure Crossing Path List Columns**

This command opens the *Configure Columns* form where the display in the **Crossing Paths** table can be customized.

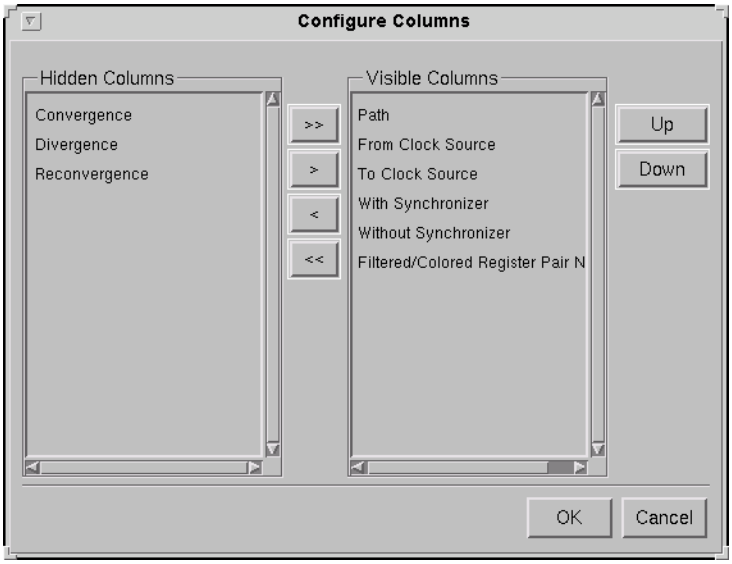


Figure: Crossing Path - Configure Columns Form

The *Configure Columns* form has two main columns: *Hidden Columns* which lists the columns not to be displayed and *Visible Columns* which lists the columns to be displayed. Use the >> > < << buttons to specify what to view in

the *Crossing Paths* table. The selections can be moved between the columns. By default, all sorting types are placed in the *Visible Columns*.

>>: This button moves all selections from *Hidden Columns* to *Visible Columns*.

>: This button moves a single selection from *Hidden Columns* to *Visible Columns*.

<: This button moves a single selection from *Visible Columns* to *Hidden Columns*.

<<: This button moves all selections from *Visible Columns* to *Hidden Columns*.

UP: This button moves the selected column to an upper display order. The column at the top of the list is the left-most column.

DOWN: This button moves the selected column to a lower display order. The column at the bottom of the list is the right-most column.

OK: This button applies the column setting to the *Crossing Paths* table immediately.

Cancel: This button closes the *Configure Columns* form. The previous settings remain in the form.

Configure CDC Correlation Columns

Menu Bar: **View -> Configure CDC Correlation Columns**

This command opens the *Configure Columns* form where the display in the *CDC Correlation Details* table can be customized.

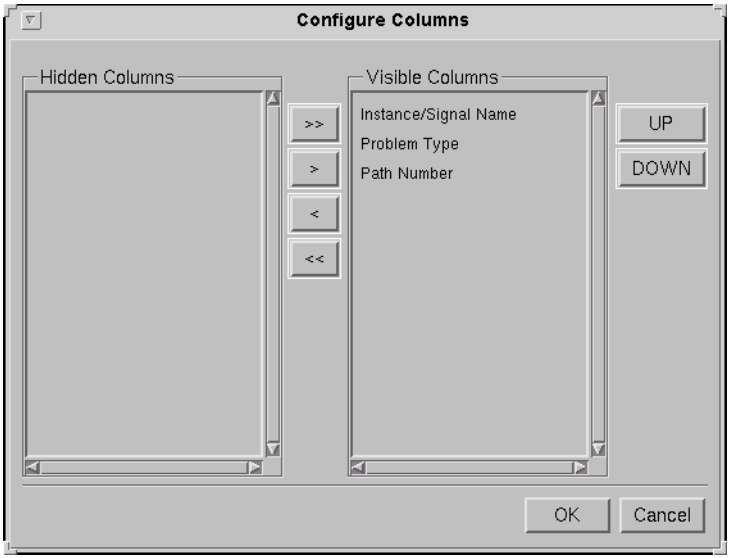


Figure: CDC Correlation - Configure Columns Form

The *Configure Columns* form has two main columns: *Hidden Columns* which lists the columns not to be displayed and *Visible Columns* which lists the columns to be displayed. Use the >> > < << buttons to specify what to view in the *CDC Correlation Details* table. The selections can be moved between the columns. By default, **Instance/Signal Name**, **Problem Type**, and **Path Number** are placed in the *Visible Columns*. For usage of column customization, refer to the **Configure Crossing Path List Columns** command description for details.

Configure Register Pair Columns

Menu Bar: **View -> Configure Register Pair Columns**

This command opens the *Configure Columns* form where the display in the *Register Pair Details* table can be configured.

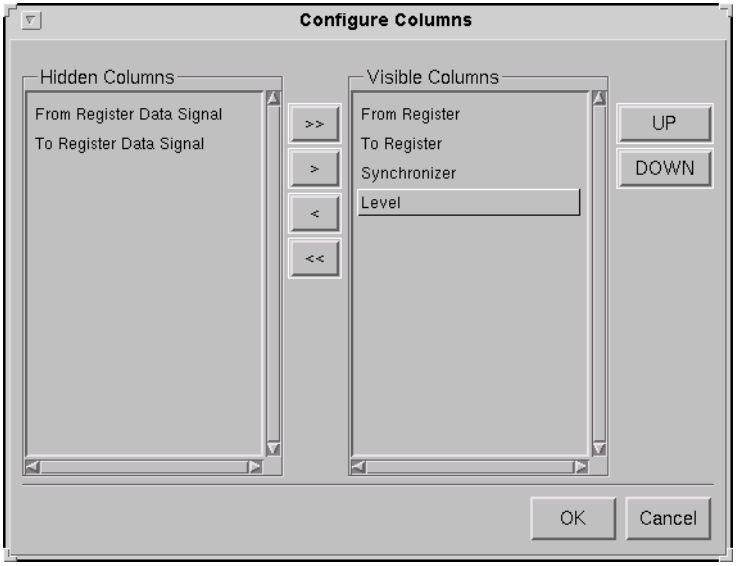


Figure: Register Pair - Configure Columns Form

The *Configure Columns* form has two main columns: *Hidden Columns* which lists the columns not to be displayed and *Visible Columns* which lists the columns to be displayed. Use the >> > < << buttons to specify what to view in the *Register Pair Details* table. The selections can be moved between the columns. By default, **From Register**, **To Register**, **Synchronizer**, and **Level** are placed in the *Visible Columns*. **From Register Data Signal** and **To Register Data Signal** are placed in the *Hidden Columns*. For column customization usage, refer to the **Configure Crossing Path List Columns** command description for details.

Hide Selection

Menu Bar: **View -> Hide Selection**

This command hides checked crossing path(s) and register pair(s).

Show Selection

Menu Bar: **View -> Show Selection**

This command displays hidden crossing path(s) and register pair(s).

Select All

Menu Bar: **View -> Select All**

This command checks all crossing paths and register pairs.

Clear Selection

Menu Bar: **View -> Clear Selection**

This command clears all the checked crossing paths and register pairs.

Tools Commands

Filter

Menu Bar: **Tools -> Filter**

This command opens the *Filter* form where filter rules can be set up to modify the display in *Crossing Paths*, *CDC Correlation Details,* and *Register Pair Details* tables in the *Crossing Paths* window.

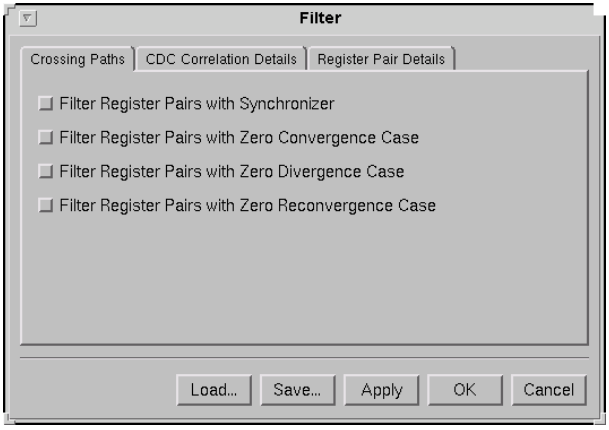


Figure: Filter Form - Crossing Paths Tab

- **Crossing Paths Tab**

Use the following four options to show or hide the results in **With Synchronizer**, **Convergence**, **Divergence**, and **Reconvergence** columns in the *Crossing Paths* table.

- **Filter Register Pairs with Synchronizer:** When this option is turned *on*, only paths without synchronizers are shown in the *Crossing Paths* table. When this option is turned *off*, all paths (with and without synchronizers) are shown.
- **Filter Register Pairs with Zero Convergence Case:** When this option is turned *on*, only paths with convergence cases are shown in the *Crossing Paths* table. When this option is turned *off*, all paths (with and without convergence) are shown.
- **Filter Register Pairs with Zero Divergence Case:** When this option is turned *on*, only paths with divergence cases are shown in the *Crossing Paths* table. When this option is turned *off*, all paths (with and without divergence) are shown.
- **Filter Register Pairs with Zero Reconvergence Case:** When this option is turned *on*, only paths with reconvergence cases are shown in the *Crossing Paths* table. When this option is turned *off*, all paths (with and without reconvergence) are shown.

NOTE: The relationship among the four options above is **AND**.

- **CDC Correlation Details Tab**

Use the following three options and the **Add** button to show or hide the results in the *Problem Type, Instance/Signal Name*, and *Path Number* columns in the *CDC Correlation Details* table.

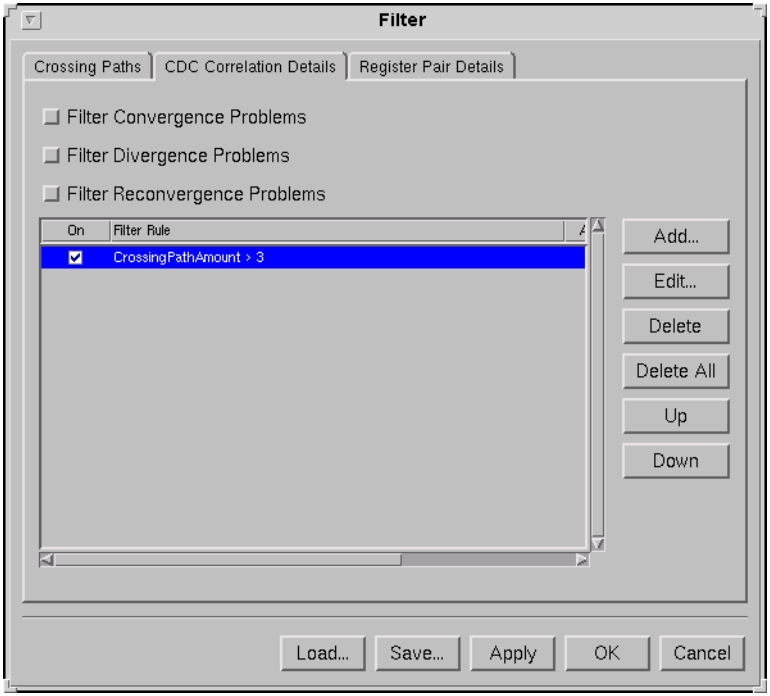


Figure: Filter Form - CDC Correlation Details Tab

- **Filter Convergence Problems:** When this option is turned *on*, convergence paths are not shown in the *CDC Correlation Details* table.
- **Filter Divergence Problems:** When this option is turned *on*, divergence paths are not shown in the *CDC Correlation Details* table.
- **Filter Reconvergence Problems:** When this option is turned *on*, reconvergence paths are not shown in the *CDC Correlation Details* table.

After additional filters are specified, the filter summary is shown in the main display area which consists of two columns: the *On* column specifies whether the filter rule is used; the *Filter Rule* column displays the filter rule(s).

- **Add:** Click this button to open the *CDC Correlation Details Filter Rule Setting Form* where the filter rules and actions can be customized.
- **Edit:** After selecting a row in the main display area, click the **Edit** button to open the *CDC Correlation Details Filter Rule Setting Form* for editing the filtering rule.
- **Delete:** After selecting one or more rows in the main display area, click the **Delete** button to remove the selected filter rule setting(s).

- **Delete All:** Click the **Delete All** button to remove all the filter rule settings.
- **Up/Down:** Click these buttons to modify the priority order of the filter rule setting by moving the selected row up or down one row at a time.

CDC Correlation Details Filter Rule Setting Form

Click the **Add** button in the **CDC Correlation Details** tab of the *Filter* form to open the *CDC Correlation Details Filter Rule Setting* form.

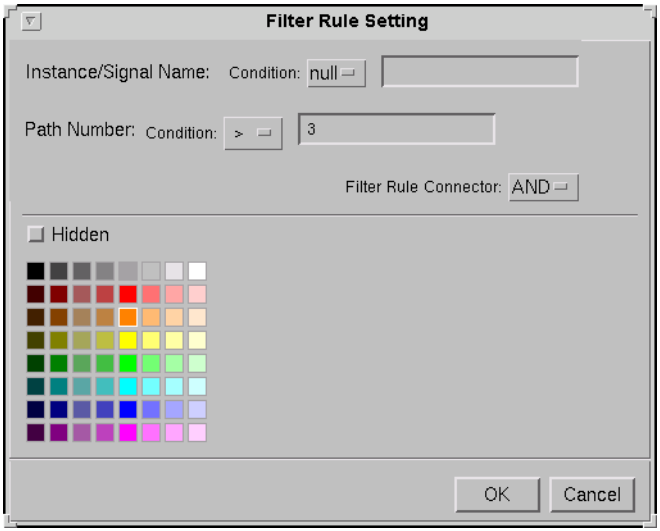


Figure: Filter Rule Setting Form - CDC Correlation Details

The *Filter Rule Setting* form is used to customize filter rules. The **Instance/ Signal Name** and **Path Number** options in the form correspond to columns in the *CDC Correlation Details* table of the *Crossing Paths* window.

The following conditions are used for specifying the filter rule setting. If a condition is selected but a value is not entered in the associated column, a filter rule is not created.

- **null:** If the condition is set to null, the associated column field is ignored and no filter rule is created.
- **"==":** The **CDC Correlation Details** table is filtered based on results equal to the value in the associated column field.
- **"!=":** The **CDC Correlation Details** table is filtered based on results not equal to the value in the associated column field.
- **">":** The *CDC Correlation Details* table is filtered based on results greater than the value in the *Path Number* column field.

- **"<":** The *CDC Correlation Details* table is filtered based on results smaller than the value in the *Path Number* column field.

Filter Rule Connector: Specifies the **Filter Rule Connector** with **AND** or **OR**. The default is **AND**.

Hidden Option/ **Color** Palate**:** If a CDC correlation meets the specification in the filter rule setting, turn the **Hidden** option *off* and select a preferred color for highlighting the CDC correlation in the *CDC Correlation Details* table. The mismatched CDC correlations are shown in the default color, black. When the **Hidden** option is turned *on*, the CDC correlation that meets the specifications in the filter rule setting becomes invisible in the *CDC Correlation Details* table.

OK: Click the **OK** button and the customized filter rule setting is added in the *CDC Correlation Details* tab of the *Filter* form. The *CDC Correlation Details Filter Rule Setting* form is closed.

Cancel: This button closes the *CDC Correlation Details Filter Rule Setting* form. The previous settings remain in the form.

- **Register Pair Details Tab**

  Set filter rule(s) with options in this tab to view register pairs of interest in the **Register Pair Details** table of the *Crossing Paths* window.

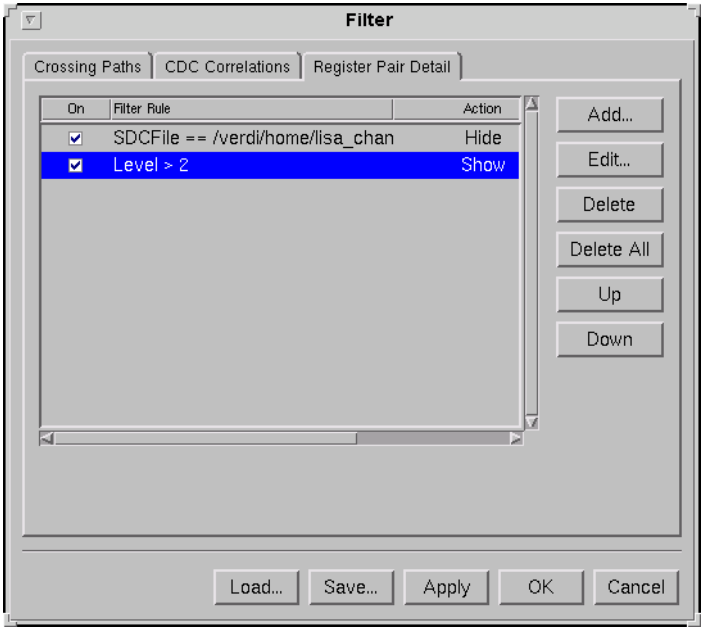


Figure: Filter Form - Register Pair Details Tab

The main display area shows the filter rules for register pairs. The **On** column specifies whether the filter rule is used or not. The *Filter Rule* column displays the filter rule summary. The *Action* column indicates if the filter rule is shown or hidden in the *Register Pair Details* table.

- **Add:** Click this button to open the *Register Pair Details Filter Rule Settings Form* where the filter rules and actions for the **Register Pair Details** table can be customized.
- **Edit:** After selecting a rule, click the **Edit** button to open the *Register Pair Details Filter Rule Settings Form* where the rule settings for the **Register Pair Details** table can be edited.
- **Delete:** After selecting one or more rules, click the **Delete** button to remove settings for the selected filter rule.
- **Delete All:** Click the **Delete All** button to remove all the filter rule settings.
- **Up/Down:** Click these buttons to modify the priority order of the filter rule settings by moving the selected rule up or down one row at a time.

Register Pair Details Filter Rule Settings Form

Click the **Add** button in the *Register Pair Details* tab of the *Filter* form to open the *Filter Rule Settings* form where the filter rules for the *Register Pair Details* tab in the *Crossing Paths* window can be customized. The *Filter Rule Settings* form includes the *Programatic Conditions* tab and the *False Path* tab. The filter settings in these two tabs are not related to each other.

- **Programatic Conditions Tab**

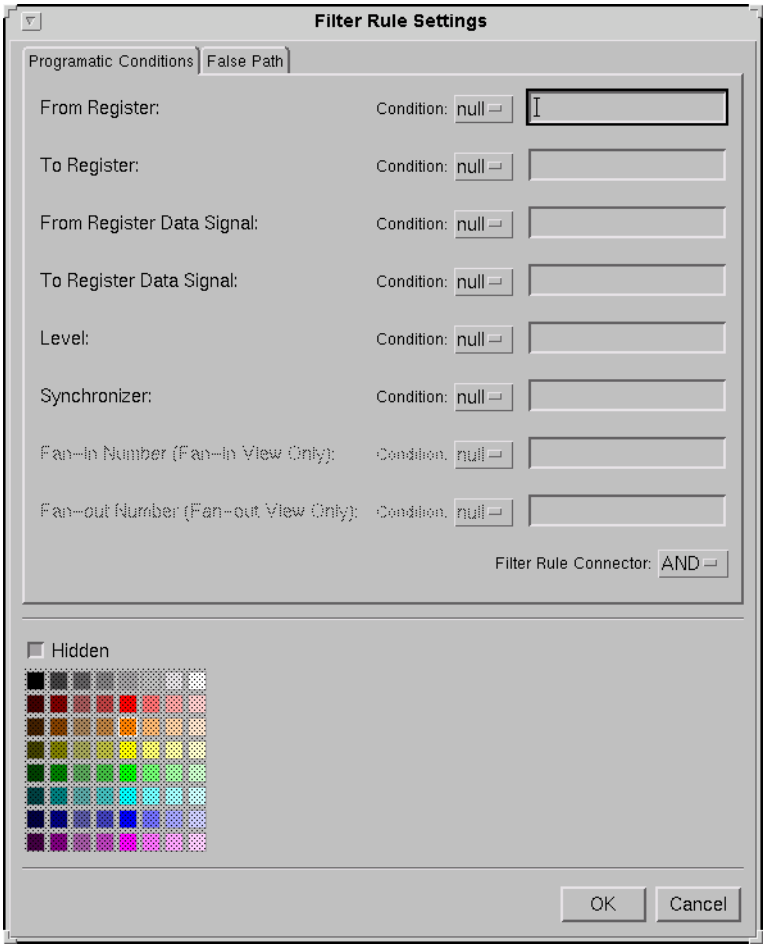


Figure: Filter Rule Settings Form - Programatic Conditions Tab

The **From Register**, **To Register**, **From Register Data Signal**, **To Register Data Signal**, **Level**, **Synchronizer**, **Fan-in Number (Fan-in View Only)**, and **Fan-out Number (Fan-out View Only)** options in this tab correspond to columns in the *Register Pair Details* table of the *Crossing Paths* window. Refer to the **Register Pair Details** description for details.

The following conditions are used for specifying the filter rule settings. If a condition is selected but a value is not entered in the associated column, a filter rule is not created.

- **null:** If the condition is set to null, the associated column field is ignored and no filter rule is created.

- **“==”:** The *Register Pair Details* table is filtered based on results equal to the value in the associated column field.
- **“!=”:** The *Register Pair Details* table is filtered based on results not equal to the value in the associated column field.
- **“>”:** The *Register Pair Details* table is filtered based on results greater than the value in the *Level* column field.
- **“<”:** The *Register Pair Details* table is filtered based on results smaller than the value in the *Level* column field.

Filter Rule Connector: Specifies the **Filter Rule Connector** with **AND** or **OR**. The default is **AND**.

- **False Path Tab**

  Load an SDC file which includes false path information and set the false path(s) as filter rule(s) for viewing the *Register Pair Details* table in the *Crossing Paths* window.

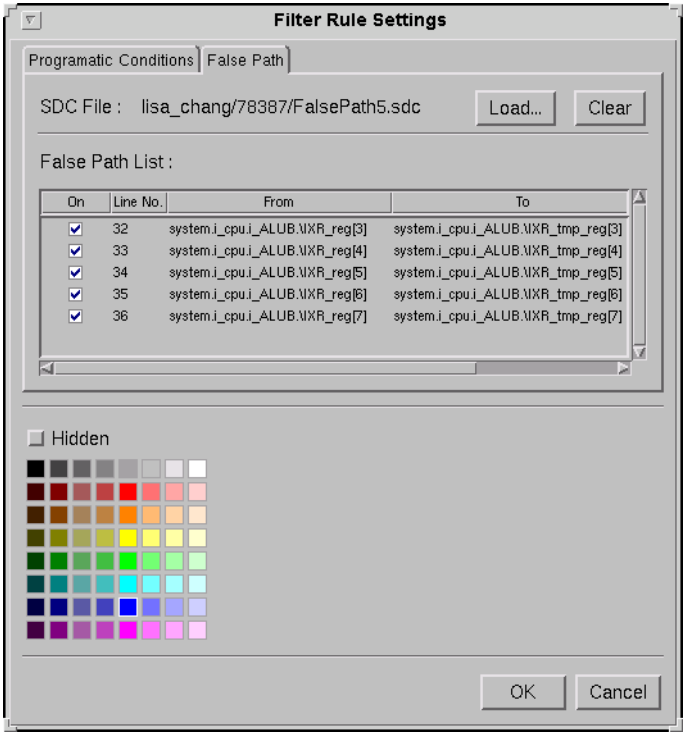


Figure: Filter Rule Settings Form - False Path Tab

SDC File: This field shows the path to the loaded SDC file.

Load: Click this button to open the *Load SDC* form where the directory structure can be viewed and an SDC file specified to load from.

Clear: Click this button to remove all loaded rules from the **False Path List**.

False Path List: The table shows false paths in the selected SDC file. All false paths are instance to instance. Double-click a rule and the SDC file opens at the line for the rule.

- **On:** Indicates if the filter rule is selected or not.
- **Line No:** Indicates the line number of the selected false path in the SDC file.
- **From:** Indicates the 'from register' of the false path or the 'from clock source', depending on the imported SDC file. If it is a clock source, the filter rule is applied to the *Crossing Paths* table of the *Crossing Paths* window.
- **To:** Indicates the 'to register' of the false path or the 'to clock source', depending on the imported SDC file. If it is a clock source, the filter rule is applied to the *Crossing Paths* table of the *Crossing Paths* window.

NOTE: After the **OK** button is clicked in the *Filter Rule Settings* form:
1. The filter rule for the false path settings is shown in italics in the *Filter* form. After the **Apply** button is clicked, the setting is applied to the *Register Pair Details* table and then the font is changed to plain text.

As the above example shows, the highlighted rule is a “not applied yet” rule, while the other rule is an “applied” rule.
2. The SDC file name with full path name is converted to a filter string and shown in the *Filter Rule* column of the *Register Pair Details* tab of the *Filter* form. The rule is displayed in an “SDCFile = = file path” (such as SDCFile = = /home/TEST.sdc) manner.

Hidden Option/**Color** Palate: If a register pair meets the specifications in the filter rule settings, turn the **Hidden** option *off* and select a preferred color to highlight the register pair in the **Filter Register Pairs** table. The mismatched register pairs are shown in the default color, black. When the **Hidden** option is turned *on*, the register pair that meets the specifications in the filter rule settings become invisible in the **Filter Register Pairs** table.

OK: Click this button and the customized filter rule setting is added in the *Register Pair Details* tab of the *Filter* form, while the *Filter Rule Settings* form is closed.

Cancel: Click this button to close the *Filter Rule Settings* form without changing the previous settings.

Highlight Crossing Path

Menu Bar: **Tools -> Highlight Crossing Path**

This command opens the *Highlight Clock Domains* form. In the **Cross Path Tab**, specify the preferred color in the color palate to highlight nets and instances on all crossing paths.

Set False Path

Menu Bar: **Tools -> Set False Path**

This command sets the path selected in the *Crossing Paths* table and/or the register path selected in the *Register Pair Details* table of the *Crossing Path* window as false path(s). After selecting a crossing path and/or register path(s), invoke this command and the selection is added to the table in the *Set False Path* form automatically. The selected path(s) can also be dragged from the *Crossing Paths* window and dropped into the *Set False Path* form to add the false path(s).

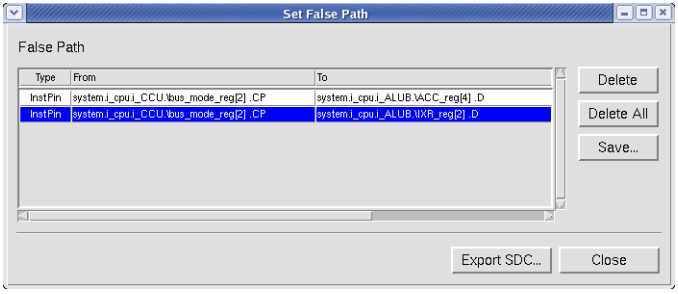


Figure: Set False Path Form

The *Type* column shows as **Clock**, if the selected path comes from the *Crossing Path* table. The *Type* column shows as **Cell**, if the selected path comes from the *Register Pair Details* table. The *From* and *To* columns refer to the start point and end point of the false path respectively.

The *Set False Path* form includes the following options:

- **Delete:** After a path in the table is selected, click this button to delete the false path.
- **Delete All:** Click this button to delete all false paths.
- **Save:** Click this button to save filter settings for the *Check Crossing Paths* form to a file that can be loaded in the *Filter* form opened by the **Tools -> Filter** command in the *Crossing Paths* window.
- **Export SDC:** Click this button to open the *Save SDC* form where a directory can be specified and the false path(s) exported to an SDC format file.
- **Close:** Click this button to close the *Set False Path* form without making any changes.

New Schematic

Menu Bar: **Tools -> New Schematic**

This command opens a flattened schematic window where the register pair(s) are displayed.

New File Viewer

This command displays the selected register pair(s) in the *Register Pair Details* form in text format. The 'from register' and 'to register' are listed as a text summary.

Short List

Menu Bar: **Tools -> New File Viewer -> Short List**

This command displays the 'from register' and 'to register' in the *Text Viewer* frame in the same format as the *Crossing Paths* window.

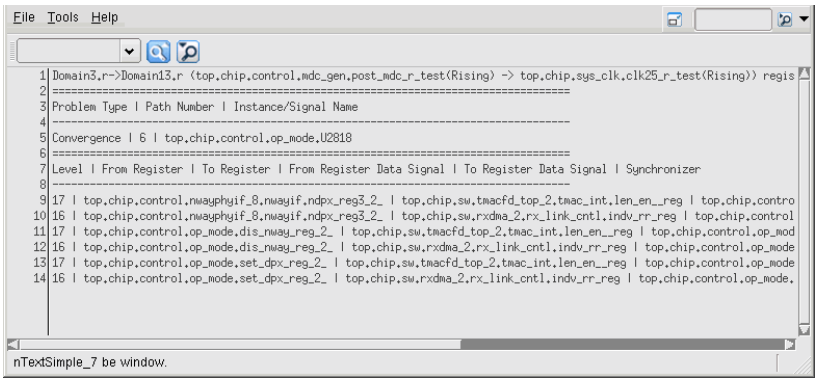


Figure: Text Viewer Frame - Short List View

Refer to the *Text Viewer Frame* section for details about the pull-down commands in the *Text Viewer* frame.

Long List

Menu Bar: **Tools -> New File Viewer -> Long List**

This command displays a detailed instance list for the selected crossing paths in the *Text Viewer* frame.

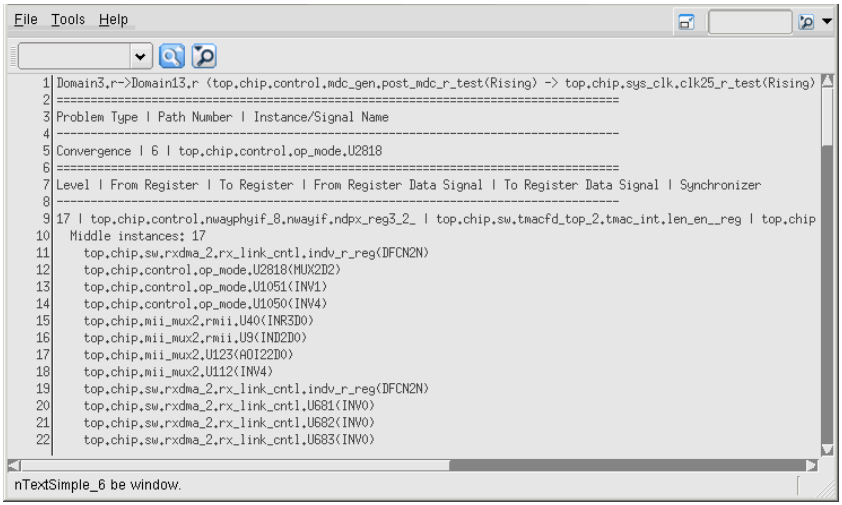


Figure: Text Viewer Frame - Long List View

Refer to the *Text Viewer Frame* section for details about the pull-down commands in the *Text Viewer* frame.

Show All

Menu Bar: **Tools -> New File Viewer -> Show All**

This command shows the register pair(s) of all path(s) listed on the **Crossing Paths** section of the *Crossing Paths* window in the *Text Viewer* frame.

Options

Menu Bar: **Tools -> Options -> Clear the Previous Results**

The **Clear the Previous Results** command is the only command under the **Tools -> Options** command. When this command is enabled, the crossing paths list is shown without its previous result. The default is disabled.

Customize Menu/Toolbar

Menu Bar: **Tools -> Customize Menu/Toolbar**

Refer to the **Tools -> Customize Menu/Toolbar** command in the *nTrace* chapter for details.

Crossing Paths Window Frame Right-click Options

When the right mouse button is clicked anywhere in the menu, toolbar icon, or frame banner area of the *Crossing Paths Window* frame or standalone window, a configuration option menu is displayed.This menu can be used to configure the available icons and frames.

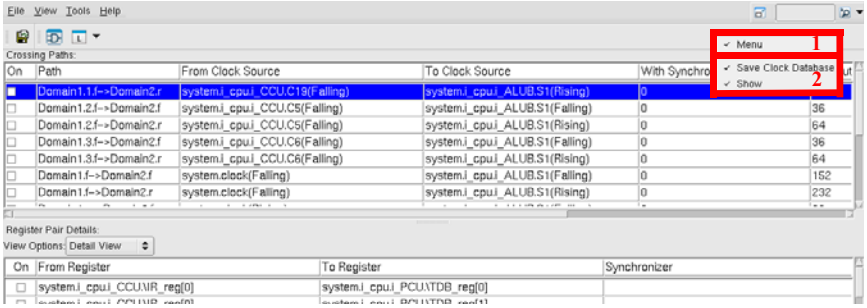


Figure: Configuration Option Menu

The Menu option (labeled 1 in the above figure) enables/disables the command menu bar. The options in the bottom section (labeled 2 in the above figure) enable/disable the toolbar icons for different functions.

Crossing Paths Window Toolbar Icons and Fields

The available toolbar icons may be modified. Refer to the *Toolbars* section of the *User Interface* chapter in the *Verdi User Guide and Tutorial* manual for details.

Save Clock Database Category

Save Clock DB 

Refer to the **File ->** **Save Clock DB** command description for details.

Show Category

New Schematic 

Refer to the **Tools ->** **New Schematic** command description for details.

New File Viewer

The **New File Viewer** command includes the **Long List**  , **Short List**  and **Show Disabled Gate**  commands. Refer to the **Tools ->** **New File Viewer** command description for details.

Assertion Debug

Overview

Assertion Debug support is enabled when the **Tools -> Property Tools -> Statistics** or **Tools -> Property Tools -> Analyzer** menu option is selected from the *nTrace* or *nWave* window. The Assertion Debug functions are docked to the existing window locations in the main *nTrace* window and are used to find, view, and debug the assertions in the design. By default, the *Statistics* pane is docked to the same pane location as the source code pane as a new tab and the *Analyzer* pane is docked to the same pane location as the *Message/nWave* panes as a new tab.

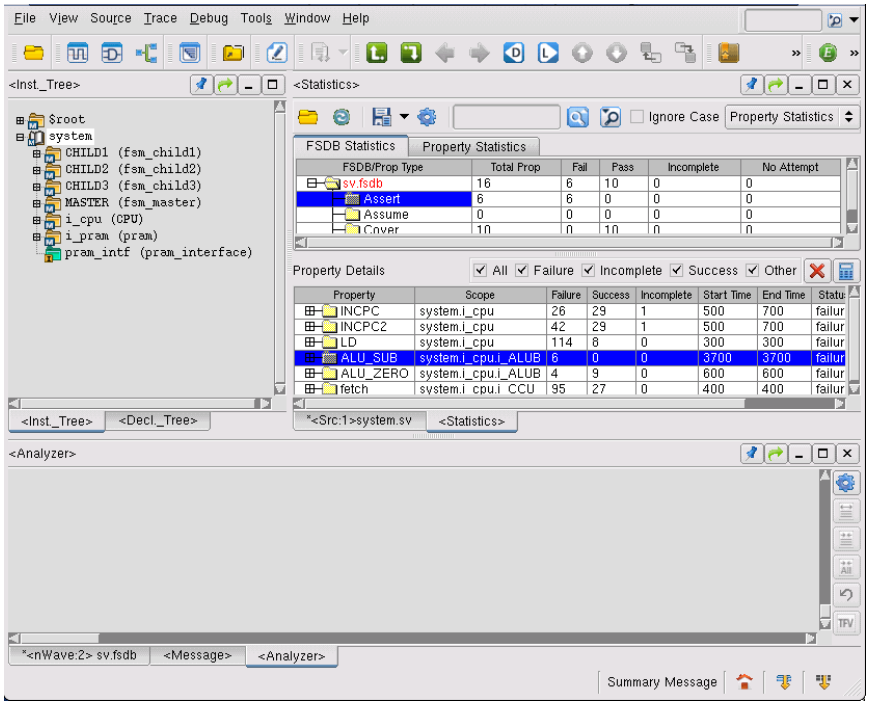


Figure: Default View for Assertion Debug

The panes, forms, and toolbar icons associated with Assertion Debug support are summarized and explained on the following pages.

Statistics Pane

The *Statistics* pane is enabled when the **Tools -> Property Tools -> Statistics** menu option is selected from the *nTrace* or *nWave* window. The *Statistics* pane is docked to the same pane location as the source code pane as a new tab.

The *Statistics* pane includes an upper and lower pane. The upper pane summarizes the **Waveform Statistics** and **Property Statistics** on separate tabs and the lower pane displays the **Property Details**.

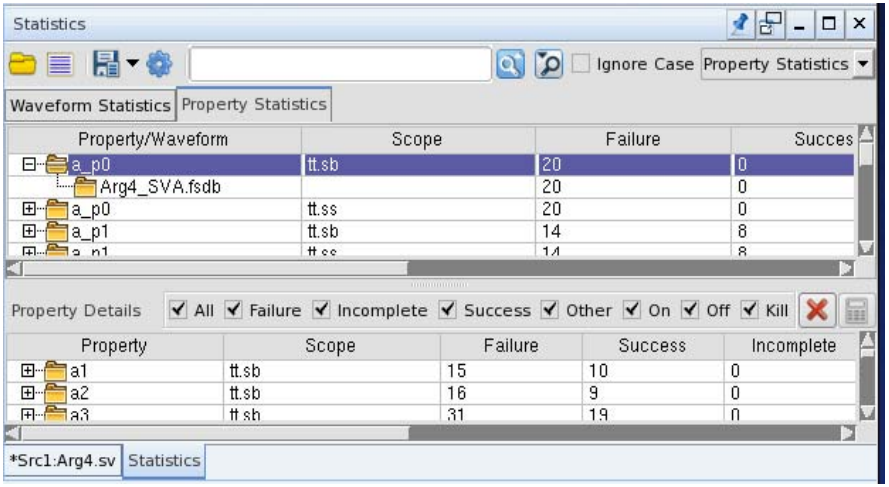


Figure: Statistics Pane

Waveform Statistics Tab

The **Waveform Statistics** tab displays the waveform-based statistics from one or more waveform files in a tabular format. It is initially empty until an waveform file is loaded.

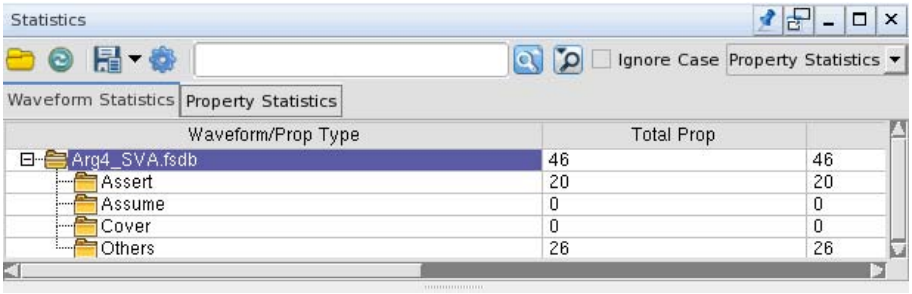


Figure: Statistics Pane - Waveform Statistics Tab

In the **Waveform Statistics** table, you can select multiple cells by holding the **Ctrl** key. Similarly, you can select a range by holding the **Shift** key. You can adjust the width of each column by placing the cursor over the vertical bar in the heading row and then pressing and holding the left mouse button. Click the + or - symbol to expand or collapse the folder.

The column headings are summarized as follows:

- **Waveform/Prop Type:** This column summarizes the property type in the selected waveform file.
- **Total Prop:** This column summarizes the total property count.
- **Fail:** This column summarizes the property count that has one or more failed attempts.
- **Pass:** This column summarizes the property count that has no failed attempts.
- **Incomplete:** This column summarizes the number of property evaluations that were not completed before the end of simulation.
- **No Attempt:** This column summarizes the number of properties that were not activated by the simulation run of the waveform file.
- **Not Checked:** This column summarizes the number of properties that were not checked during the simulation run to generate the waveform file. This column is enabled or disabled when the **Show Fields -> Not Checked** menu option is selected under the right-click option.

Waveform Statistics Tab Right-Click Options

The following right-click options are available for the **Waveform Statistics** tab of the *Statistics* pane.

Open Waveform File

This toolbar icon opens the *Open Dump File* window where you can select an FSDB file and add it to the **Waveform Statistics** table. Refer to the **File** -> **Open** option description in the *nWave* chapter for details.

Set Active Waveform File

This option sets the selected waveform file as the current active FSDB file of *nWave* and *nTrace*.

Close Waveform File

Bind Key: Shift+D

This option removes the selected waveform file from the **Waveform Statistics** tab.

Add to Details

This option adds the selected property data into the **Property Details** table.

Collapse All

This option collapses all folders in the **Waveform Statistics** table.

Show Fields

This menu includes the following option:

- **Not Checked:** When this toggle option in *on*, the **Not Checked** column in the **Waveform Statistics** pane is displayed. When this toggle option is *off*, the **Not Checked** column in the **Waveform Statistics** pane is hidden and the count in the **Total Prop** column does not include the count of **Not Checked**. The default is *off*.

Property Statistics Tab

The **Property Statistics** tab displays the property-based statistics from one or more FSDB files in a tabular format.

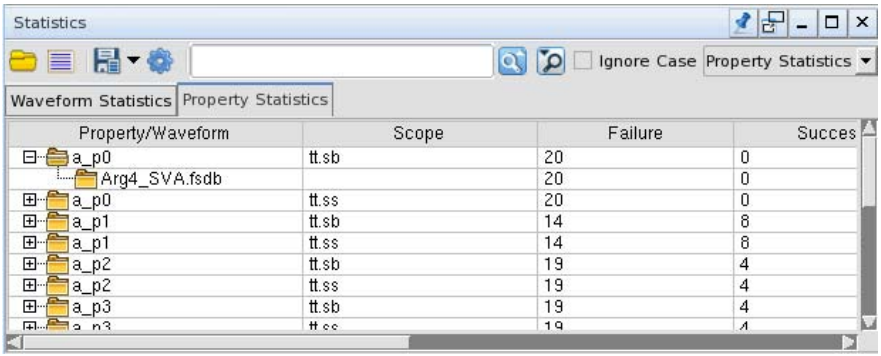


Figure: Statistics Pane - Property Statistics Tab

In the **Property Statistics** table, the property data may be selected with a left-click anywhere along the row (multiple rows can be selected at a time). The width of each column can be adjusted by placing the cursor over the vertical bar in the heading row and then pressing and holding the left mouse button. Click the "+" or "-" symbol to expand or collapse the folder.

The column headings are summarized as follows:

- **Property/Waveform:** This column displays the property and waveform file name.
- **Scope:** This column displays the scope for the property.
- **Failure:** This column displays the number of property attempts with failed results.
- **Success:** This column displays the number of property attempts with successful results.
- **Incomplete:** This column displays the number of property evaluations which were not completed before the simulation ended.
- **No Attempt:** This column displays the number of properties which were not activated by the simulation run of the FSDB file.
- **Not Checked:** This column displays the number of properties which were not selected for evaluation in the **Assertion Hierarchy View** pane.

Property Statistics Tab Right-Click Options

The following right-click menu options are available for the **Property Statistics** tab of the *Statistics* pane.

Open Waveform File

This toolbar icon opens the *Open Dump File* form where an waveform file can be selected and added to the **Waveform Statistics** table. Refer to the **File -> Open** option description in the *nWave* chapter for details.

Add to Details

This option adds the selected property data into the **Property Details** table.

Hide Properties

This option hides the selected properties from the **Property Statistics** tab.

Unhide All

This option shows all properties (including those that have been hidden) in the **Property Statistics** table.

Show Fields

This menu has the following options:

- **Module:** This option turns the display of the **Module** column *on* or *off*. The default is *off*.
- **Scope:** This option turns the display of the **Scope** column *on* or *off*.
- **Failure:** This option turns the display of the **Failure** column *on* or *off*.
- **Success:** This option turns the display of the **Success** column *on* or *off*.
- **Incomplete:** This option turns the display of the **Incomplete** column *on* or *off*.
- **Not Attempt:** This option turns the display of the **No Attempt** column *on* or *off*.
- **Not Checked:** This option turns the display of the **Not Checked** column *on* or *off*.
- **Type:** This option turns the display of the **Type** column *on* or *off*.

Options

This option opens the *Preferences* form where the **Property Statistics** page is displayed. Refer to the *Property Statistics Page* section in the *Preferences* chapter for details.

Property Details Pane

The **Property Details** table displays detailed property information.

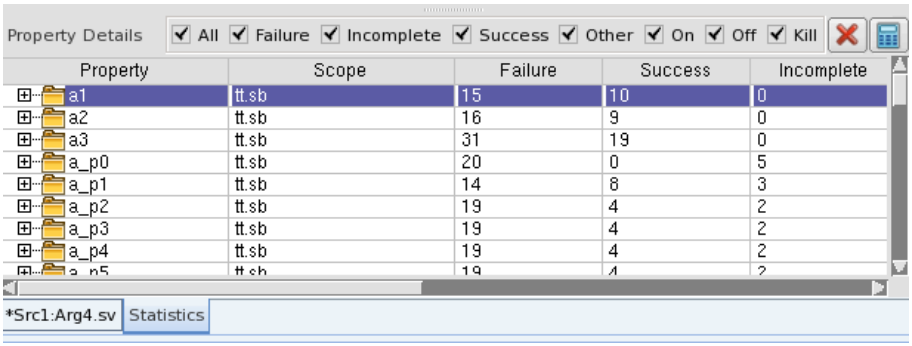


Figure: Statistics Pane - Property Details Table

In the **Property Details** table, the property data may be selected by left-clicking anywhere along the row (only one row can be selected at a time). The width of each column can be adjusted by placing the cursor over the vertical bar in the heading row and then pressing and holding the left mouse button. Choose the types of assertion results to be displayed in the **Property Details** table by enabling/disabling **All, Failure, Incomplete, Success,** and/or **Others**. Sort by any column by clicking the column headers, and click the "+" or "-" symbol to expand or collapse the folder. The **Property Details** table is refreshed when the time unit in *nWave* is changed.

The column headings are summarized as follows:

- **Property:** This column displays the property name.
- **Scope:** This column displays the scope for the property.
- **Failure:** This column displays the number of property attempts with failed results.
- **Success:** This column displays the number of property attempts with successful results.
- **Incomplete:** This column displays the number of property evaluations that were not completed before the simulation ended.
- **Start Time:** This column displays the trigger start time for the property result.
- **End Time:** This column displays the trigger end time for the property result.
- **Status:** This column displays the status for the property result (as listed previously).
- **Waveform:** This column displays the source FSDB file name for the property result.

- **Type:** This column displays the type name (For example, assert).
- **Index Info:** This column displays the index information for each failed assertion. The format is "<index1 name>=<index1 value>, <index2 name>=<index2 value>, ... <indexN name>=<indexN value>" (For example, "i=1, j=2, k=1).

Property Details Table Right-Click Options

The following right-click menu options are available for the **Property Details** table of the *Statistics* pane.

Display Thread by Count

Specify the number of assertion threads to be displayed in the **Property Details** table. The options include **All, 5, 20,** and **100**. {...} is used for the threads that exceed the specified number. After double-clicking {...}, the next specified number of threads are shown.

Analyze Property

Toolbar Icon:	
Bind Key:	Alt+Shift+A

When selecting the row with an assertion fail in the **Property Details** table and invoking this option, assertion debug is started in the *Analyzer* pane.

Delete Properties

Toolbar Icon:	
Bind Key:	Ctrl+Shift+D

This option removes one or more selected rows from the **Property Details** table. Multiple rows can be selected by holding the **Shift** key (selects a range) or the **Ctrl** key (selects non-contiguous items) and left-clicking.

Delete All Properties

This option removes all data from the **Property Details** table.

Show Fields

This menu has the following options:

- **Module:** This option turns the display of the **Module** column *on* or *off*. The default is *off*.
- **Scope:** This option turns the display of the **Scope** column *on* or *off*.
- **Failure:** This option turns the display of the **Failure** column *on* or *off*.
- **Success:** This option turns the display of the **Success** column *on* or *off*.
- **Incomplete:** This option turns the display of the **Incomplete** column *on* or *off*.
- **Start Time:** This option turns the display of the **Start Time** column *on* or *off*.
- **End Time:** This option turns the display of the **End Time** column *on* or *off*.
- **Status:** This option turns the display of the **Status** column *on* or *off*.
- **Waveform:** This option turns the display of the **Waveform** column *on* or *off*.
- **Type:** This option turns the display of the **Type** column *on* or *off*.
- **State:** This option turns the display of the **State** column *on* or *off*.
- **Index Info:** This option turns the display of the **Index Info** column *on* or *off*.

Options

This option opens the *Preferences* form where the **Property Details** page is displayed. Refer to the *Property Details Folder* section in the *Preferences* chapter for details.

Statistics Pane Toolbar Icons and Fields

Open Waveform File 

This toolbar icon opens the *Open Dump File* form where an FSDB file can be selected and added to the **Waveform Statistics** table (multiple FSDB files can be added to the table). Refer to the **File -> Open** option description in the *nWave* chapter for details on the options and usage of the *Open Dump File* form.

Coverage Report 

This toolbar icon opens the *Coverage Report* form where the coverage report can be displayed and saved.

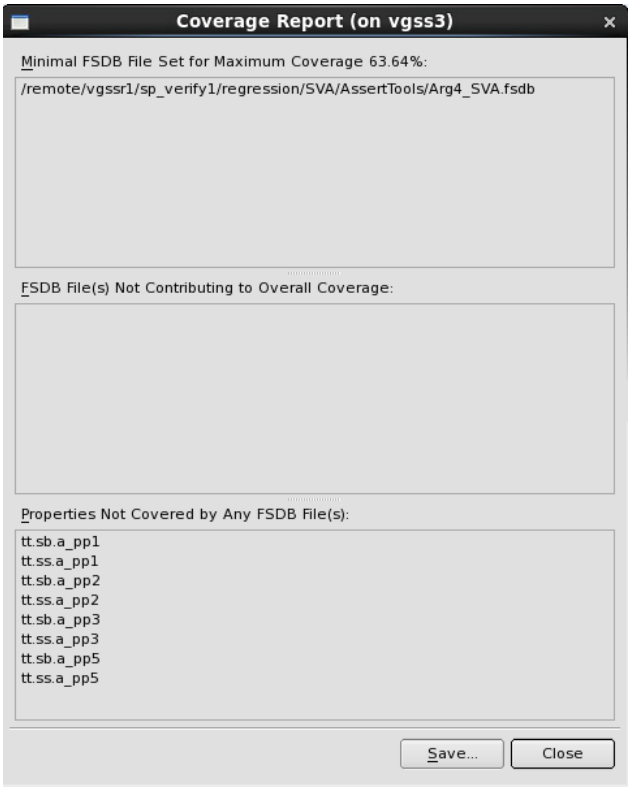


Figure: Coverage Report Form

The *Coverage Report* form includes the following results sections:

- **Minimal FSDB File Set for Maximum Coverage %**: This section shows the likely heuristically-determined minimal set of FSDB files that can achieve the maximum coverage possible.

  The following is an example of the heuristic solution for determining the minimum FSDB set for maximum coverage. For example, assume the following properties are included in the following files:

  f1={p1}, f2={p1, p2}, f3={p3}, f4={p3}, f5={p1, p4}

  a. Set min_fsdb_set = {}, and set covered_prop_set = {}
  b. Create two tables P and F. P is for properties that are covered by the listed files, and F is for files that cover the listed properties.

     + P = {p1={f1, f2, f5}, p2={f2}, p3={f3, f4}, p4={f5}}

     + F = {f1={p1}, f2={p1, p2}, f3={p3}, f4={p3}, f5={p1, p4}}
  c. Choose the property with the least number of covered files.

Both p2 and p4 are covered by only one file, p2 is chosen first.

Remove p2 from P, add p2 to covered_prop_set, and add its covered file f2 to min_fsdb_set:

P = {p1={f1, f2, f5}, p3={f3, f4}, p4={f5}}

covered_prop_set ={p2}, min_fsdb_set = {f2}

d. Remove additional covered property p1 of f2 from P, add p1 to covered_prop_set, and remove f2 from F:

P = {p3={f3, f4}, p4={f5}}

covered_prop_set ={p2, p1}

F = {f1={p1}, f3={p3}, f4={p3}, f5={p1, p4}}

e. Repeat steps 3 and 4 until all covered properties are removed from P.

i. Choose p4 and then remove p4 from P, add p4 to covered_prop_set, add f5 to min_fsdb_set, and remove f5 from F.

P = {p3={f3, f4}}

covered_prop_set ={p2, p1, p4}, min_fsdb_set = {f2, f5},

F = {f1={p1}, f3={p3}, f4={p3}}

ii. Choose p3 and then remove p3 from P, add p3 to covered_prop_set, and add f3 to min_fsdb_set (randomly choose f3 from covered files{f3,f4}).

P = { }

covered_prop_set ={p2, p1, p4, p3}, min_fsdb_set = {f2, f5, f3}

iii. All covered properties are removed from P; therefore, stop.

covered_prop_set ={p2, p1, p4, p3}, min_fsdb_set = {f2, f5, f3}

This means {f2,f5,f3} covers maximum properties {p2,p1,p4,p3}, and redundant files are {f1,f4} that can be removed from the regression cases.

- **FSDB File(s) Not Contributing to Overall Coverage**: This section shows the FSDB file(s) that do not add to the overall coverage.
- **Properties Not Covered by Any FSDB File(s)**: This section shows the properties that are not covered by any of the FSDB file(s).

Save 

This toolbar icon has two options for file format: **Save as Text** and **Save as CSV**. After the desired format is selected, the *Save Property Results* form opens with the **Filter** and file name set for this format. In the *Save Property Results* form, the directory structure can be viewed and a file name designated for saving the

property results to. Select the information to be saved by enabling one or more of the **Save FSDB Statistics**, **Save Property Statistics**, and/or **Save Property Details** options.

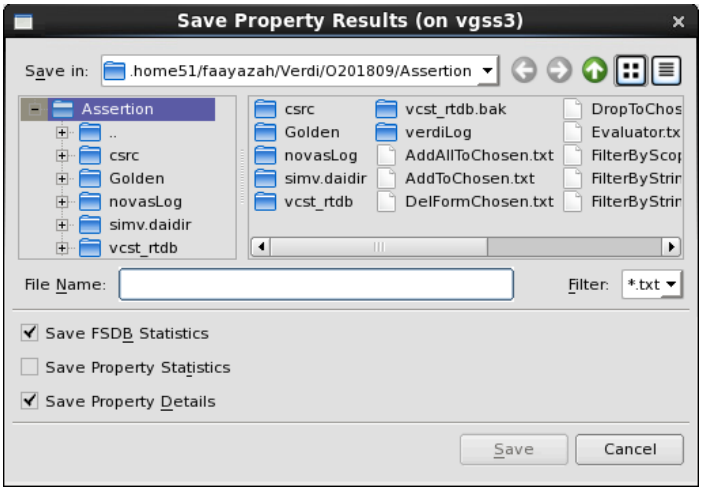


Figure: Save Property Results Form

Options

This toolbar icon opens the *Preferences* form with the **Property** folder displayed. Refer to the *Property Folder* section in the *Preferences* chapter for details.

Find Property

Enter the assert name to be searched for in the current file. Click the **Find Next** or **Find Previous** toolbar icons to find the next or previous assert name specified in the this text field.

Find Previous

This toolbar icon finds the previous assert name specified in the **Find Property** text field.

Find Next

This toolbar icon finds the next assert name specified in the **Find Property** text field.

Ignore Case 

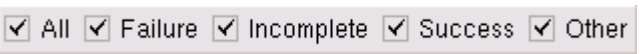
When this option is turned *on*, the case is ignored in **Find Property** text field. When this option is turned *off*, the case is considered in the **Find Property** text field. The default is *off*.

Find Property in Specified Area 

Specify the area in the *Statistics* pane to search for the assert name. The options include **Property Statistics** and **Property Details**.

Property Details Table Toolbar Icons and Fields

The following toolbar icons are available for the **Property Details** table. A property must be added to the table for these icons to provide value.

Display 

These options filter the type of properties displayed in the **Property Details** table.

Delete Properties 

This toolbar icon removes the selected property from the **Property Details** table.

Analyze Property 

This toolbar icon starts the assertion debug in the *Analyzer* pane for the selected assertion fail in the **Property Details** table.

Analyzer Pane

The *Analyzer* pane is enabled when the **Tools -> Property Tools -> Analyzer** option is selected from the *nTrace* or *nWave* window. The *Analyzer* pane can also be displayed by invoking the **Assertion Analyzer** option in the source code pane or double-clicking on the assertion failure or success point in *nWave* or clicking the **Analyze Property** icon on the *Statistics* pane. The *Analyzer* pane is docked to the same pane location as the *Message/nWave* window as a new tab.

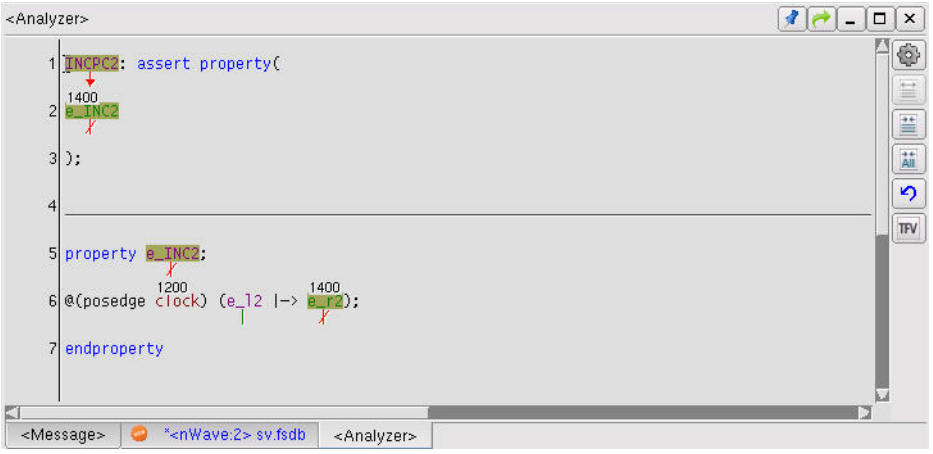


Figure: Analyzer Pane

The *Analyzer* pane displays the assertion information for the assertion being analyzed, including the assertion name and the current active FSDB file. Additionally, the expression label <-----> representing the text range for the first operation is displayed and the operation result is displayed under this label as active annotation. A red line indicates a failing expression and a green line indicates a passing expression. To debug a failure/success point, double-click the failure/success point in the *Analyzer* pane to locate the causing statement in *nTrace* and jump to the signal in *nWave*.

Analyzer Pane Right-Click Options

The following right-click options are available for the *Analyzer* pane.

Signal Value Radix

Refer to the **Waveform -> Signal Value Radix** option description in the *nWave* chapter for details.

Signal Value Notation

Refer to the **Waveform -> Signal Value Notation** option description in the *nWave* chapter for details.

Options

This option opens the *Preferences* form where the **Analyzer** page is displayed. Refer to the *Analyzer Page* section in the *Preferences* chapter for details.

Analyzer Pane Toolbar Icons

Open Options 

This toolbar icon opens the *Preferences* form where the **Analyzer** page is displayed. Refer to the *Analyzer Page* section in the *Preferences* chapter for details.

Shrink 

This toolbar icon shrinks all the expression labels of assertions, properties, let constructs, or sequences.

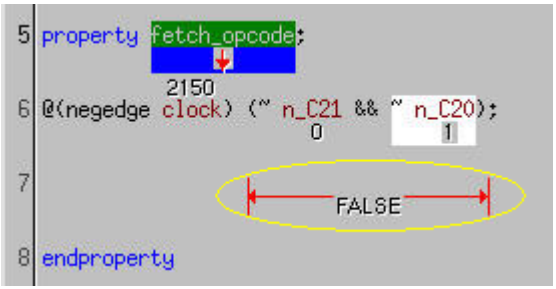


Figure: Shrink All Expression Labels

Expand 

This toolbar icon displays the expression labels of assertions, properties, let constructs, or sequences for the next lower level.

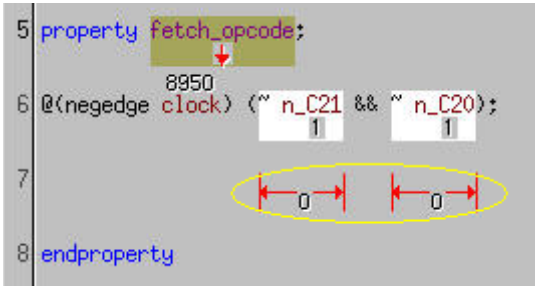


Figure: Expand Expression Labels

Expand All 

This toolbar icon displays the expression labels of assertions, properties, let constructs, or sequences for the lowest level.

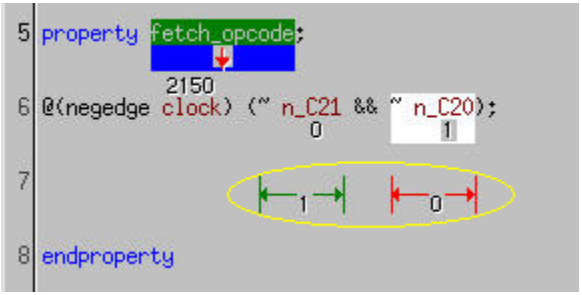


Figure: Expand All Expression Labels

Backward History 

This toolbar icon traces backwards through the most recent trace results.

Open TFV 

This toolbar icon creates a *Temporal Flow View* window for the selected signal. The *Setup Temporal Flow View* form is initially displayed to specify the analysis options. Refer to the **Tools -> Temporal Flow View ->** **Behavior Analysis** option description in the *nTrace* chapter for details.

Evaluate Properties Form

This form is displayed when the **Tools -> Property Tools -> Evaluator** option is invoked from the *nTrace* or *nWave* window. All available SystemVerilog Assertions are listed in the *Evaluate Properties* form and the assertions to be evaluated can be selected.

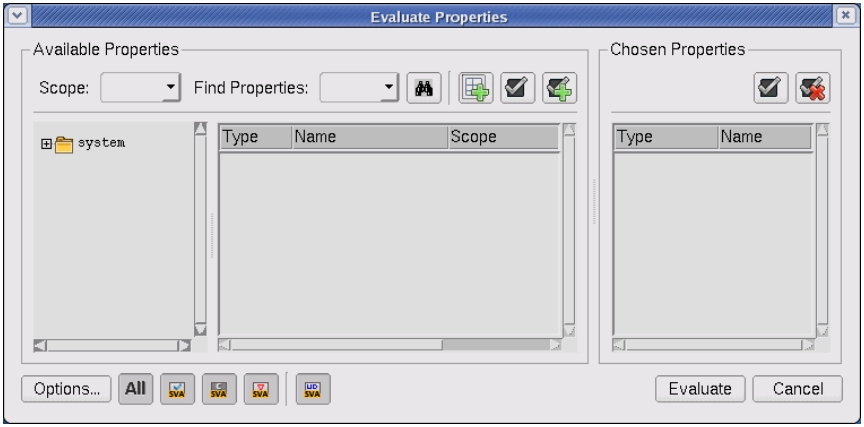


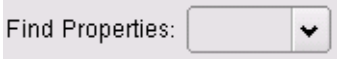
Figure: Evaluate Properties Form

In the *Evaluate Properties* form, the left pane displays the design hierarchies that have underlying SVA, the center pane lists all assertions for the current scope, and the right pane displays the assertions selected for evaluation.

The following fields and buttons are available in the **Available Properties** section:

Scope 

This text field displays the scope selected in the **Hierarchical Scope Tree** pane. A partial scope name can be entered in the **Scope** text field, and the matched scopes are highlighted. This text field is case-sensitive.

Find Properties 

This text field is used to filter the properties to display for the current selected scope. It is case-insensitive. For example, if "C*" is typed in the **Find Properties** text field and the **Find** button is clicked (or the **Enter** key on the keyboard pressed), the **Assertion List** is updated to show all the transactions that start with C or c.

Find 
This button finds the matching scopes and properties and filters the results in the right pane of the **Available Properties** section.

Add All Properties 
This button adds all assertions in the design to the **Chosen Properties** section.

Select/Deselect All Properties 
This button selects or deselects all assertions in the right pane of the **Available Properties** section.

Add Selected Properties 
This button adds the selected assertions from the right pane of the **Available Properties** section to the **Chosen Properties** section. Multiple assertions can be selected by holding the **Shift** key (selects a range) or the **Ctrl** key (selects non-contiguous items) and left-clicking. Alternatively, double-click an assertion in the right pane of the **Available Properties** section to add the assertion.

The following fields and buttons are available in the **Chosen Properties** section:

Select/Deselect All Properties 
This button selects or deselects all assertions in the **Chosen Properties** section.

Delete Selected Properties 
This button removes unwanted assertions from the **Chosen Properties** section.

The following buttons are also available on the *Evaluate Properties* form:

Options
This button opens the *Options* form where the options to be used in the *Transaction Evaluator* form can be configured after extraction. Refer to the *Options Form* section for details.

Signal Type 

Click these buttons to choose the selected signal types displayed in the right pane of the **Available Properties** section. The available types are **All**, **Assert**, **Cover**, **Assume** and respectively.

Show Temporary Assertions Only 

Click this button to view temporary assertions (such as runtime assertions) in the right pane of the **Available Properties** section. Temporary assertions are displayed in blue.

Evaluate

Click this button to evaluate the assertions in the **Chosen Properties** list and save the results to the FSDB file specified in the *Options* form.

Options Form

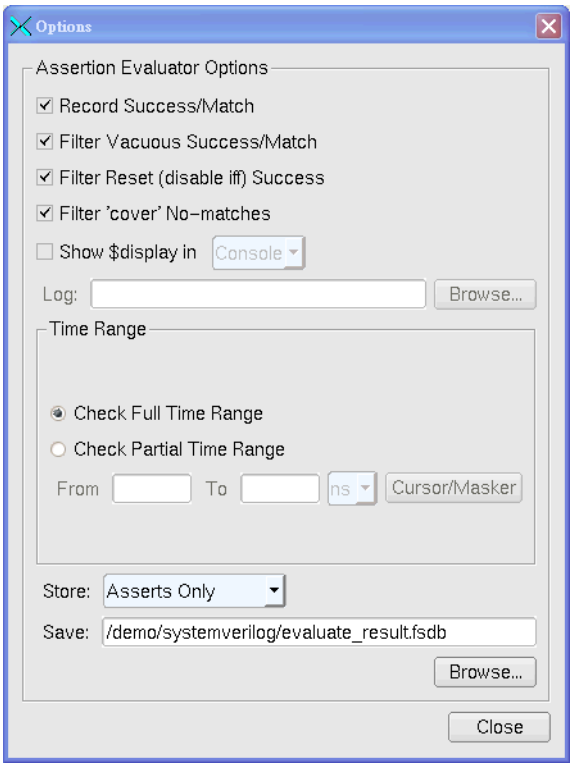


Figure: Options Form

The **Assertion Evaluator Options** section includes the following options and fields:

- **Record Success/Match:** When this option is turned *off*, only assertion failures are saved in the result file. When this option is turned *on*, assertion passes and failures are saved in the result file. The default is *on*.
- **Filter Vacuous Success/Match:** When this option is turned *off*, trivial (vacuous) successes or matches are saved in the result file. When this option

is turned *on*, trivial successes or matches are not saved in the result file. The default is *on*.

- **Filter Reset (disable iff) Success:** When this option is turned *off*, reset (disable iff) successes are saved in the result file. When this option is turned *on*, reset (disable iff) successes are not saved in the result file. The default is *on*.
- **Filter 'cover' No-Matches:** When this option is turned *off*, no-match cover results for cover is saved in the results file. When this option is turned *on*, only match cover results are saved in the results file. The default is *on*.
- **Show $display in:** When this option is turned *on*, $display contents can be saved in the console or the log file by selecting either **Console** or **Log** from the selection field. The default is *off*.
  When **Log** is selected, the **Browse** button is activated to open the *Save Log File* form where the directory structure can be viewed and a file name to save can be specified. The default file name is *evaluate_result_display.log*.
- The **Time Range** section includes the following options and fields. Only one option can be selected at a time.
  - **Check Full Time Range:** Specify the full time range for assertion evaluation. This option is the default.
  - **Check Partial Time Range:** Specify the partial time range for assertion evaluation.When this option is selected, the time range of interest can be manually entered in the **From** and **To** text fields and the time unit (options are *s, ms, us, ns, as,* or *as*) set or automatically synchronized with the *nWave* cursor or marker time by clicking the **Cursor/Marker** button. If a specific setting does not exist in the *novas.rc* resource file and an active FSDB file is not specified, the default time is set as *ns*.
- **Store:** Specify the data type to be stored in the FSDB file by selecting one of the following options: **Asserts Only** (assertions only, such as assert, assume, and cover without the dependent data, such as events, sequences), **No Local variables**, and **All** (all data types except SVA local variables).
- **Save:** Specify the FSDB file name to save the evaluation results to. The default name is *evaluate_result.fsdb*.

Add Temporary Assertions Form

This form is displayed when the **Tools -> Property Tools -> Add Temporary Assertions** option is invoked from the main *nTrace* window. Temporary assertions (such as runtime assertions) can be added during runtime without compiling and loading the design again.

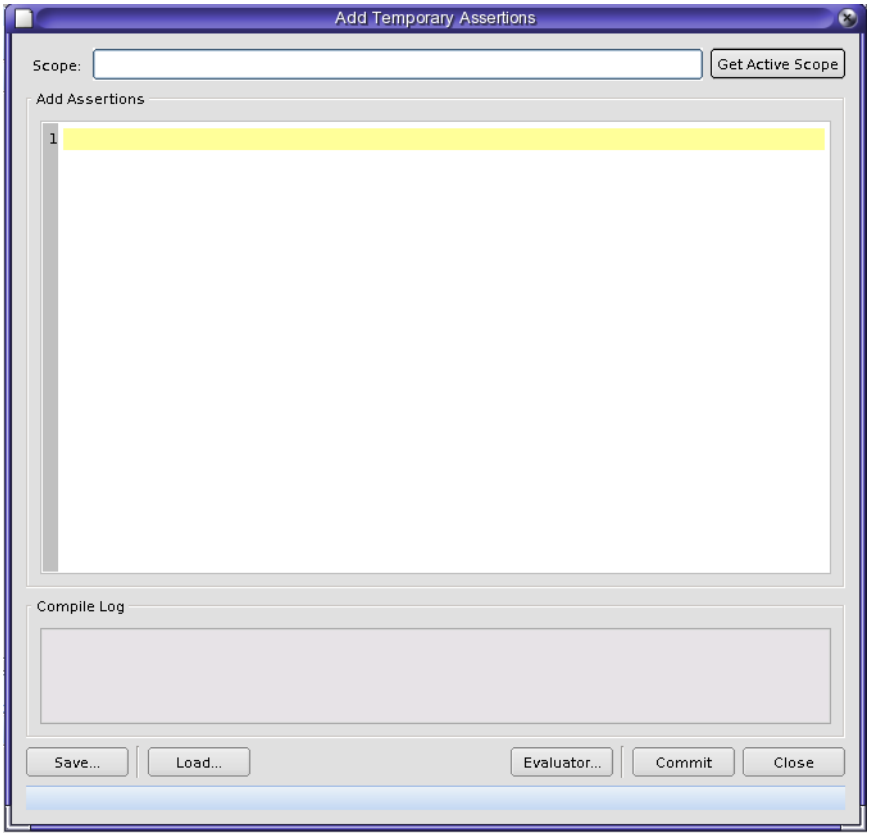
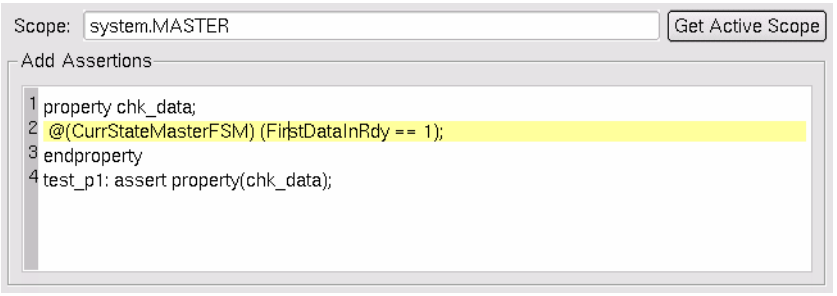


Figure: Add Temporary Assertions Form

The *Add Temporary Assertions* form includes the following buttons, options, and fields:

- **Scope:** Enter the scope name by typing in this text field or click the **Get Active Scope** button to add the active scope automatically.
- **Add Assertions:** Type assertion statements in this section as shown in the example.



NOTE: The right-click options in this section (such as **Undo**, **Redo**, **Cut**, **Copy**, **Paste**, **Delete**, **Select All**, **Select IM**, and **Insert unicode control character**) are editing options embedded in the Qt platform.

- **Compile Log:** After clicking the **Commit** button, this section displays the compilation results of the new assertion code.
- **Save:** Click this button to save the assertion and scope to a file with a .tsva extension.

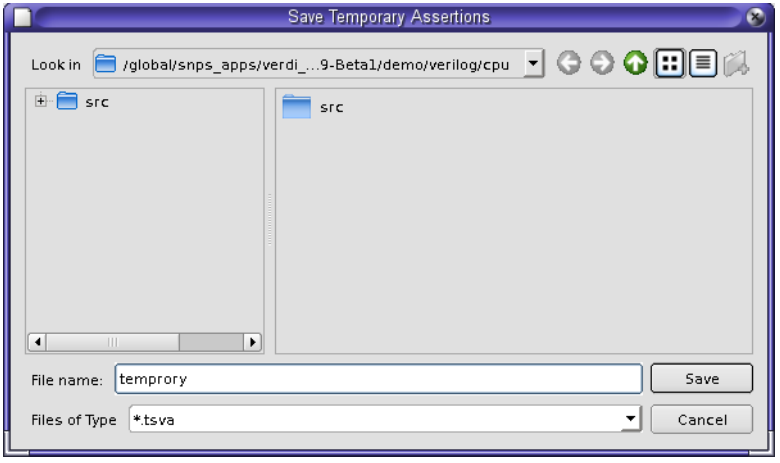


Figure: Save Temporary Assertions

- **Load:** Click this button to load the assertion and scope from a file with a .tsva extension.

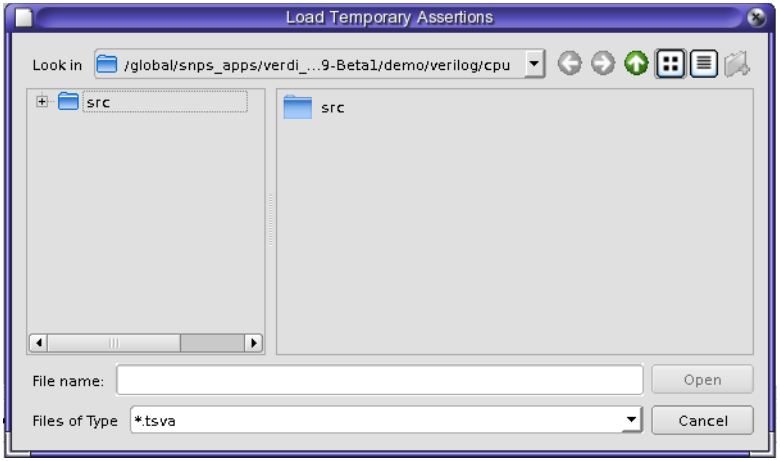


Figure: Load Temporary Assertions

- **Evaluator:** Click this button to open the *Evaluate Properties Form* where the results after the assertions are added successfully can be checked and debugged.
- **Commit:** After completing assertion statements in the **Add Assertions** section, click this button and the **Compile Log** section shows the compilation results of the new assertion code.

Assertion Attempt List Form

This form is opened when double-clicking the end time of an assertion in the waveform pane of *nWave*.

NOTE: The end time must have at least two results at the same time.

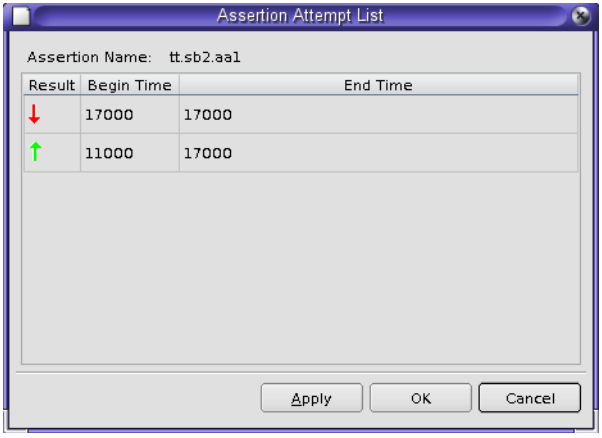


Figure: Assertion Attempt List Form

The *Assertion Attempt List* form lists results of the selected assertion in the Result, Begin Time, End Time, Index Info columns. After the result of interest is selected, click the **OK** button and the assertion analysis is shown in the *Analyzer* pane.

Power Aware Debug

Overview

Power Aware Debug support is integrated with the main *nTrace* window. When a CPF/UPF file is loaded through the command line and the **Window -> Power Debug Mode** menu option is selected, the panes and icons associated with Power Aware Debug and the **Power** pull-down menu are available in the *nTrace* window. The CPF/UPF files can also be loaded using the **File** -> **Import CPF/UPF Files** menu option from the *nTrace* window menu bar. The design and power aware source code can be browsed and interactive debugging can be performed.

The various Power Aware Debug functions are docked to existing pane locations. By default, the *Power Domain Tree* pane showing the power domain list or hierarchy is docked to the same pane as the design browser as either a **Flatten Power Domain** tab (CPF file only) or **Hierarchical Power Domain** tab (UPF file only). The *Power Mode Table* and *Power State Table* panes are docked to the same pane as the signal list pane (the middle pane) as new tabs after the **Power -> Show Power Mode Table** or the **Power -> Show Power State Table** toggle options are turned *on* (the defaults for **Show Power Mode Table** and **Show Power State Table** are *off*). The *Power Map* and *Partial Power Map* panes are docked to the same pane as the source code/schematic pane as a new tab after the related **Power -> New Power Map -> Full/Partial Power Map** options are invoked. The power source code pane named with the prefix <*UPF*> or <*CPF*> is incorporated as a separate tab in the source code pane.

NOTE: A power license is required when a CPF/UPF file is loaded.

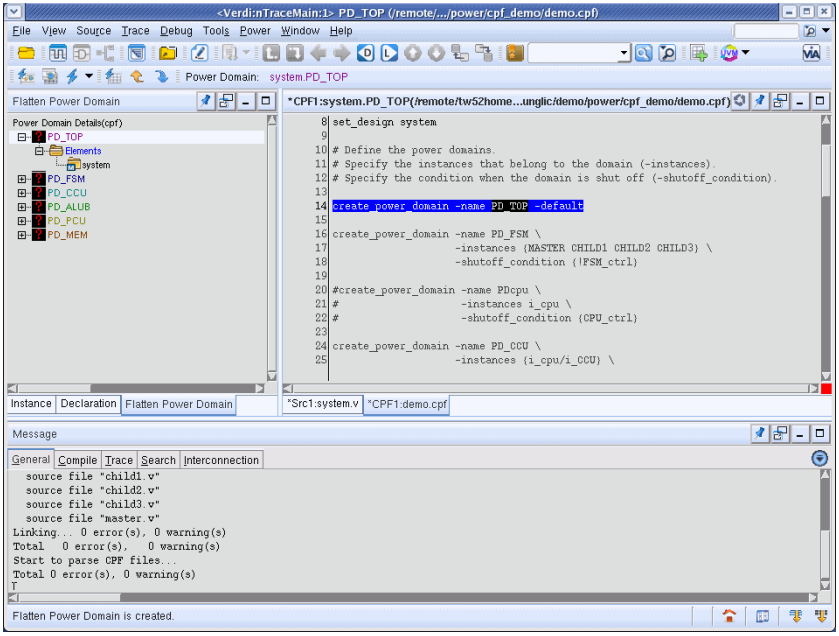


Figure: Default View for Power Aware Debug with CPF File Imported

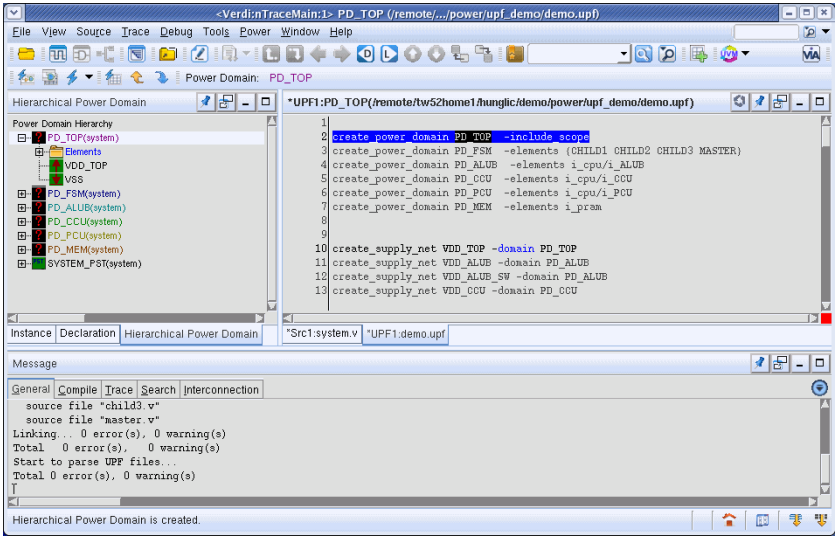


Figure: Default View for Power Aware Debug with UPF File Imported

The pull-down menus, toolbar icons, and panes associated with Power Aware Debug support are summarized and explained on the following pages.

Design and Power Source Code Panes

The design source code pane shows the code for the imported design. The power source code pane is a separate tab and shows the source code of the imported CPF/UPF power file. The contents are changed when different items are selected in the *Power Domain Tree* pane. Enabling the **Source** -> **Active Annotation** menu option annotates the FSDB value for signals under the source code if the FSDB file is loaded into the Verdi platform.

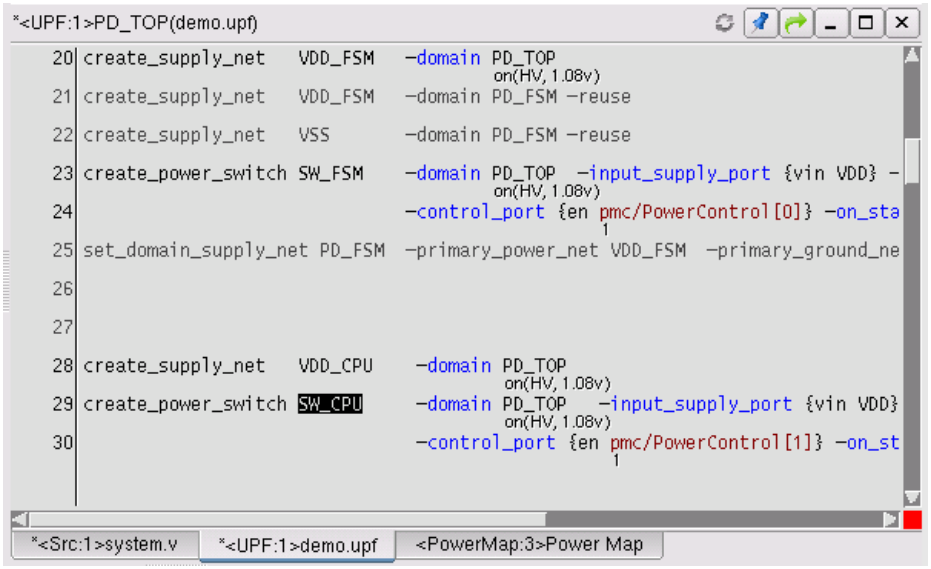


Figure: Source Code Tab for Power Code (UPF)

When the **Source -> Find String** menu option is invoked to search for any item in the power file, all searched results are listed in the **Search** tab of the *Message* pane. Double-clicking the search results message jumps to the corresponding line in the *Source Code* pane.

NOTE: All power library cells defined by library CPF commands can be displayed in *nTrace*/*nSchema* and highlighted with the color specified in the **General** -> **Power** page of the *Preferences* form (if the **Highlight Power Aware Design Object(s)** option is turned *on*). The library cells defined by library CPF commands can be annotated in the *Source Code* pane to indicate their power object type (if the **Annotate Power Aware Design Object(s)** option is turned *on* in the **General** -> **Power** page of the *Preferences* form).

NOTE: When the *nTrace* and *Power Domain* source panes are opened simultaneously, only one source window is activated. When the panes are opened in different areas, click the pane title to activate it (the pane title is highlighted). If the panes are opened in the same area, click the tab to activate it (the tab name includes an asterisk).

In *nTrace* source pane,

- If an HDL signal connects to an isolation, SRSN, or SPA cell directly, and is used as the output of the connected isolation, SRSN, or SPA cell, the signal is marked as `iso_out`, `srsn_out`, or `spa_out` signal, and is annotated with symbol in the *nTrace* source window.
- If an HDL signal connects to an isolation, SRSN, or SPA cell directly, and is used as the input of the connected isolation, SRSN, or SPA cell, the signal is marked as `iso_in`, `srsn_in`, or `spa_in` signal, and is annotated with symbol in *nTrace* source window.
- If an HDL signal connects to an isolation, SRSN, or SPA cell directly, and is used as both input and output of the connected isolation, SRSN, or SPA cells, the signal is marked as `iso_inout`, `srsn_inout`, or `spa_inout` signal, and is annotated with symbol in *nTrace* source window.

  If the HighConn or LowConn of an HDL signal is connected to the output and input ports of two different cells (for example, `srsn_out` + `iso_in` as shown in the following figure), the signal is annotated with symbol in *nTrace* source window.

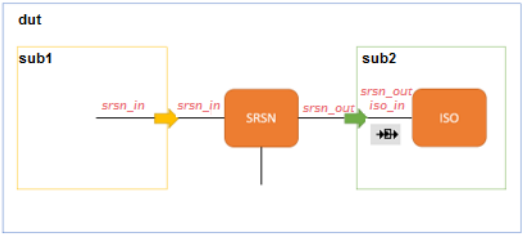
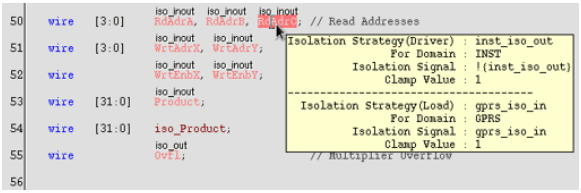


Figure: HDL Signal Connection to Two Different Cells

In *nTrace* source pane, tip information for the isolation, SRSN, and SPA signals (`iso_in`, `iso_out`, `iso_inout`, `spa_in`, `spa_out`, `spa_inout`, `srsn_in`, `srsn_out`, `srsn_inout`) is displayed for the previous and next isolation, SRSN, or SPA rules.



The tip information format of the Isolation, SRSN, and SPA signals is as follows:

Isolation Format:

```
Isolation Stratgey (Driver|Load) : <ISO_RULE_NAME>
                        For Domain : <POWER_DOMAIN_NAME>
                  Isolation Signal : <ISO_SIG_NAME>
                       Clamp Value : <CLAMP_VALUE>
```

SRSN Format:

```
SRSN Strategy (Driver|Load) : <SRSN_RULE_NAME>
                      Power Net : <SUPPLY_NET_NAME>
                     Ground Net : <SUPPLY_NET_NAME>
```

SPA Format:

```
SPA Strategy (Driver|Load) : <SPA_RULE_NAME>
                Driver Supply : <SUPPLY_SET_NAME>
             Receiver Supply : <SUPPLY_SET_NAME>
                 Clamp Value : <CLAMP_VALUE>
```

In *nTrace* signal pane, Verdi displays the information (`iso_in`, `iso_out`, `iso_inout`, `spa_in`, `spa_out`, `spa_inout`, `srsn_in`, `srsn_out`, or `srsn_inout`) of the signals which impact or are impacted by an isolation, SRSN, or SPA cell in the **Power Info** column as shown in the following figure:

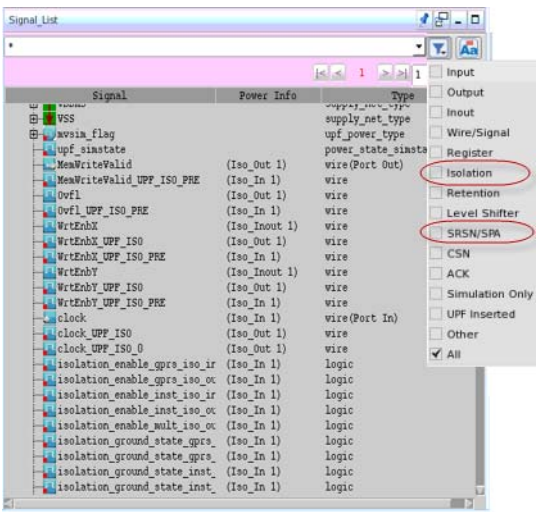


Figure: nTrace Signal Pane

You can use the **Isolation** and **SRSN/SPA** filter options to hide or display the isolation (`iso_in`, `iso_out`, `iso_inout`) and SRSN/SPA (`spa_in`, `spa_out`, `spa_inout`, `srsn_in`, `srsn_out`, or `srsn_inout`) signals in the **Power Info** column.

The following options are available with the source code panes after CPF/UPF files are imported into the Verdi platform.

File Options

The following Power Aware Debug options are available under the **File** menu in the *nTrace* window.

Import CPF/UPF Files

Menu Bar: File -> Import CPF/UPF Files

This option opens the *Import CPF/UPF Files* form where the CPF/UPF files can be imported.

NOTE: To import the IEEE SystemVerilog package for UPF, the Verdi installation includes the necessary UPF functions to support implementations for SystemVerilog simulations. To access these functions, the **-sv** SystemVerilog option must be added to the Verdi or *vericom* command line to invoke the UPF package. The UPF design can then be imported using SystemVerilog syntax, for example, "import UPF::*".

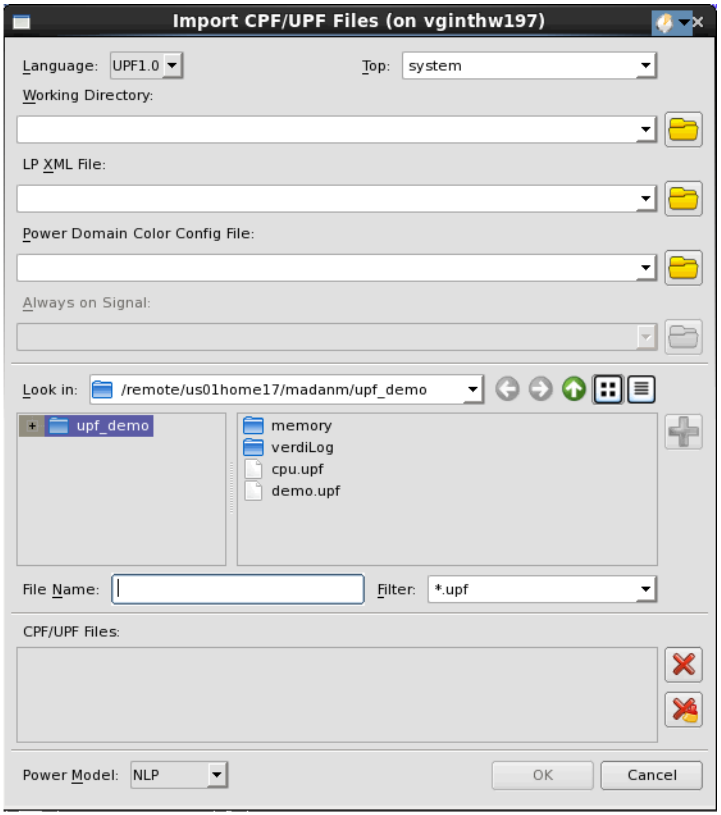


Figure: Import CPF/UPF Files Form

The *Import CPF/UPF Files* form includes the following options and fields:

- **Language:** This selection field specifies the design language to be imported from one of the following options: **UPF 1.0**, **UPF 2.0**, **CPF 1.0**, **CPF 1.1** or **CPF 2.0**.
- **Top:** Specify the design top. The default is the top scope of the design.
- **Working Directory:** Specify the working directory. Enter the path name in the text field directly or click the **Browse** button to open the *Working Directory* form where the directory structure can be viewed and a file name specified.
- **Power Library File:** Specify the power library file. Enter the file name in the text field directly or click the **Browse** button to open the *Power Library File* form where the directory structure can be viewed and a file name specified to set the power library.

- **Power Domain Color Config File:** Specify the power domain color configuration file. Enter the file name in the text field directly or click the **Browse** button to open the *Power Domain Color Config File* form where the directory structure can be viewed and a file name specified to set the power domain colors.
- **Always on Signal:** Specify the input file where the full hierarchy names of always-on signals are listed. Enter the file name in the text field directly or click the **Browse** button to open the *Always on Signal* form where the directory structure can be viewed and a file name specified to input the always-on signals.
- **CPF/UPF Files:** This list specifies the CPF/UPF files to be imported into the Verdi platform. Add files to the list by selecting the file in the lower right file list pane and clicking the **Add** button. Multiple files can be selected by holding the Shift key (selects a range) or the Ctrl key (selects non-contiguous items) and a click-left or drag-left.
- **Delete:** Remove the selected file in the **CPF/UPF Files** list.
- **Delete All:** Remove all files in the **CPF/UPF Files** list.
- **Add:** Put the file selected in the lower right file list pane in the **CPF/UPF Files** list.
- **Power Model:** Specify the simulator by selecting from one of the following options: **NLP**, **MVSim**, **ModelSim**, or **NCSim**. The default is **NLP**. Refer to the **-powerModel** option in the *verdi* utilities for details.
- **Filter:** This field is the file format filter for the File list. The default file extensions in the **Filter** field are *.*cpf*, and *.*upf*. Additional filters can be added in this selection field using a semicolon (;) as the separator.

After the selected CPF/UPF files are added and the **OK** button is clicked, the *Power Aware Debug* panes open and the *nTrace* window adds some power related information (for example, toolbar icons, fields, and the **Power** menu).

When the cursor is placed over a node in the design browser pane, a tip for the selected node containing the power domain name, power state, and shutoff condition is displayed.

When the cursor is placed over a power aware signal in the *Source Code* pane, (whose type is boundary port, Retention, Isolation, and/or Level-shifted) a tip for the selected object containing the rule name and conditions is displayed. The following is an example of a tip for a signal with the Isolation/Level-shifted type.

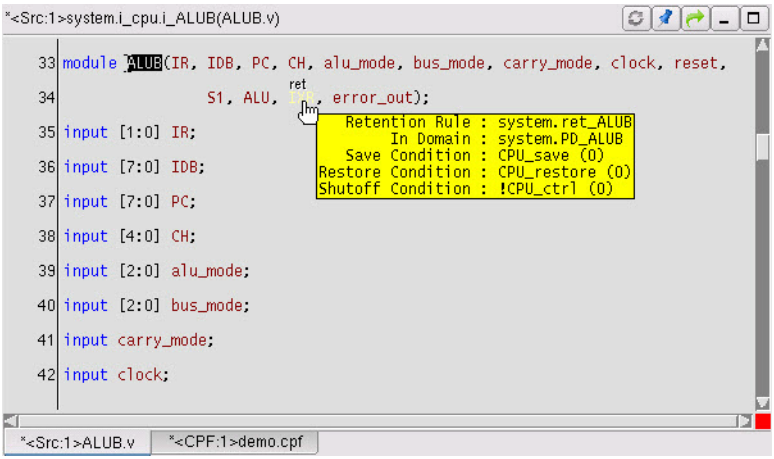


Figure: Tip Window in the Source Code Pane

In addition to the tip items in the figure, the secondary power domain is shown if it exists.

When an HDL signal is impacted by multiple power rules, the tip for this HDL signal separates the rules with a line.

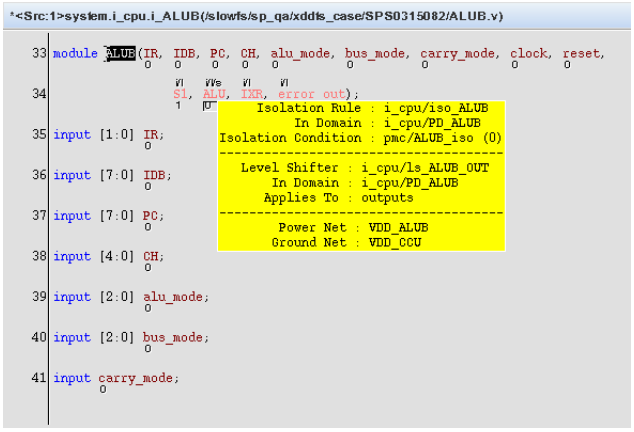


Figure: Example for Signal with Multiple Power Rules

Reload CPF/UPF Files

Menu Bar:	File -> Reload CPF/UPF Files
Bind Key:	Shift+R

NOTE: This option is available when CPF/UPF files have already been loaded.

This option reloads the previously opened CPF/UPF file.

Trace Options

The following Power Aware Debug option is available under the **Trace** pull-down menu in the *nTrace* window.

NOTE: This option is available when CPF/UPF files have already been loaded.

Trace across Power Design

Menu Bar: Trace -> Trace across Power Design

When a power signal is selected, and this toggle option is turned *on*, all the power results of the selected signal in the traced scopes in *nTrace* are traced and the results are displayed in the *Message* pane. Double-clicking the power results in the *Message* pane jumps to the corresponding source code line. This option is *on* by default.

Power Options

The **Power** pull-down menu in the *nTrace* window is available after a CPF/UPF file is loaded from the **File -> Import CPF/UPF Files** option or the Verdi command line.

Show Power State Table

Menu Bar: Power -> Show Power State Table

Toolbar Icon: 

NOTE: This option is enabled for UPF files only and is disabled when no power state table(s) exist in the imported UPF files.

This toggle option turns the display of the *Power State Table* pane *on* and *off*. Values for the power state include *on*, *off*, and *don't care* (presented as "*"). The default is *off*.

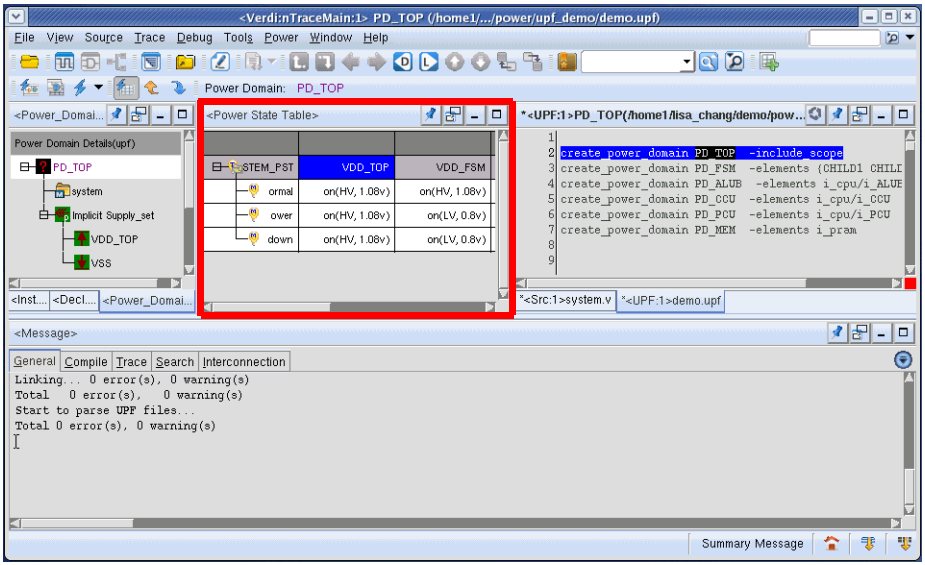


Figure: Power State Table

Refer to the *Power State/Mode Table Pane* section for details.

Show Power Mode Table

Menu Bar: Power -> Show Power Mode Table

Toolbar Icon:

NOTE: This option is enabled for the CPF files only and is disabled when no power mode table(s) exist in the imported CPF files.

This toggle option turns the display of *Power Mode Table* pane *on* and *off*. The default is *off*.

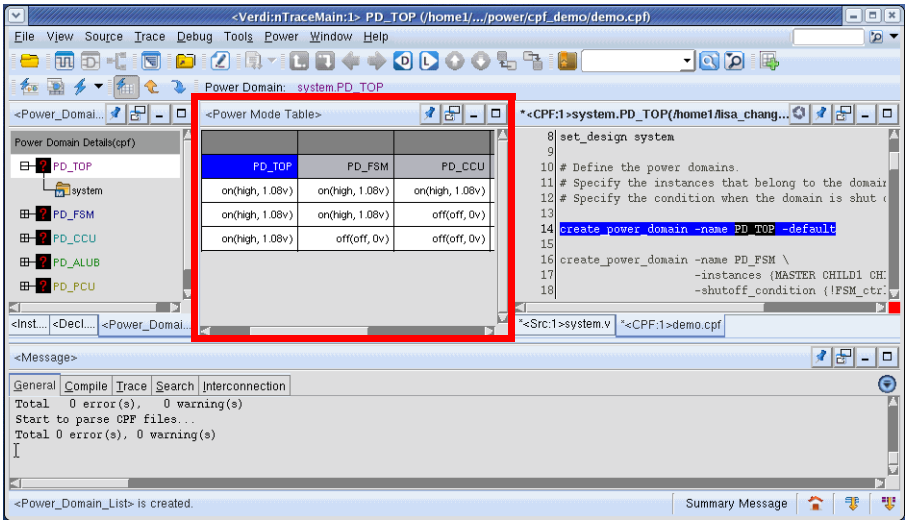


Figure: Power Mode Table

Refer to the *Power State/Mode Table Pane* section for details.

Tcl Variable Annotation

Menu Bar: Power -> Tcl Variable Annotation

This option enables the annotation for the Tcl Variables in the *Power Source Code* pane. The default is *off*.

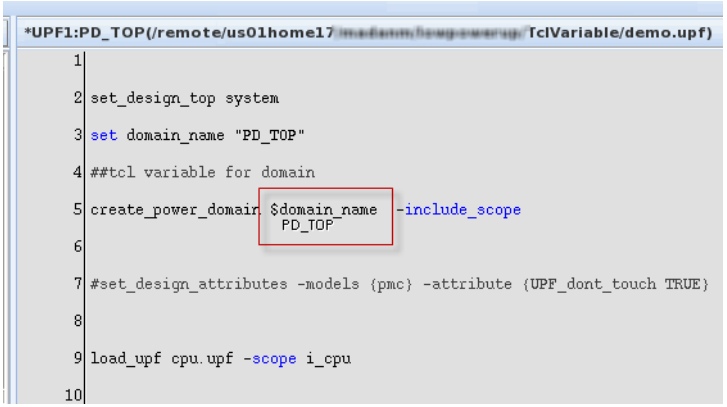


Figure: Tcl Variable Annotation

Show Tcl Variable

Menu Bar: Power -> Show Tcl Variable

NOTE: This option is displayed only when you select a Tcl variable object in the *Power Source Code* pane, else it is hidden.

This option opens a *Show Tcl Variable* form. This form reports the selected Tcl variable name, type, and value in tabular form.

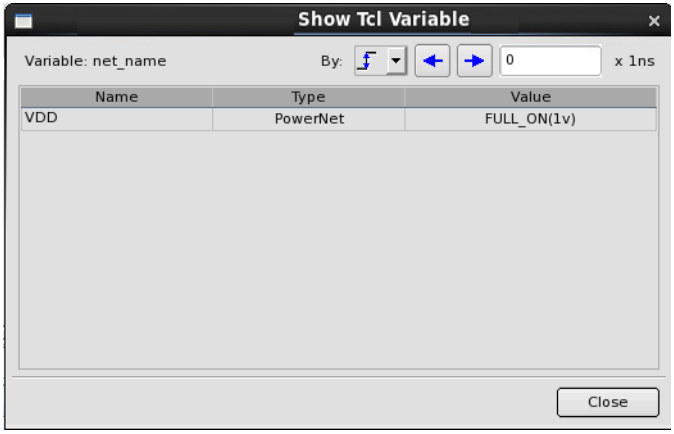


Figure: Show Tcl Variable Form

Show Tcl Variable Assignment

Menu Bar: Power -> Show Tcl Variable Assignment

NOTE: This option is displayed only when you select a Tcl variable object in the *Power Source Code* pane, else it is hidden.

This option shows the definition of the Tcl variable object selected in the *Power Source Code* pane.

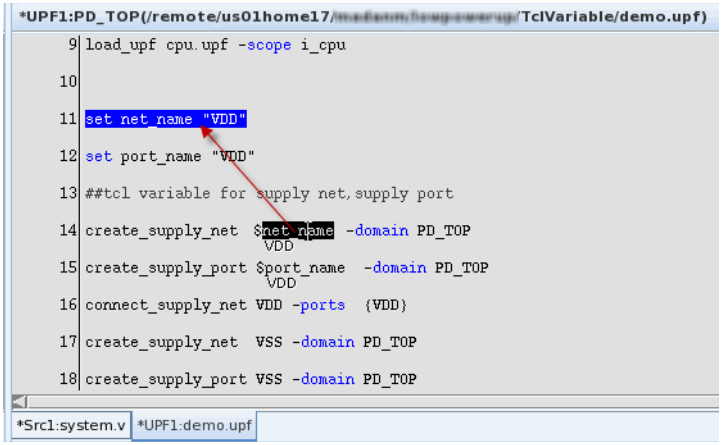


Figure: Show Tcl Variable Form

Full Scope Name

Menu Bar: Power -> Full Scope Name

This toggle option turns the display of the full power scope name in the *Power Domain Tree*, *Power Mode Table*, and *Power State Table* panes *on* and *off*. The default is *off*.

Report Impacted Signals by Scope

Menu Bar: Power -> Report Impacted Signals by Scope

Toolbar Icon: 

This option opens a *Scope Lists* form where scopes to report the power impacted signals should be specified.

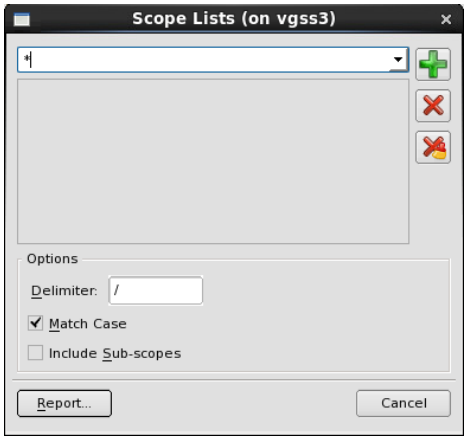
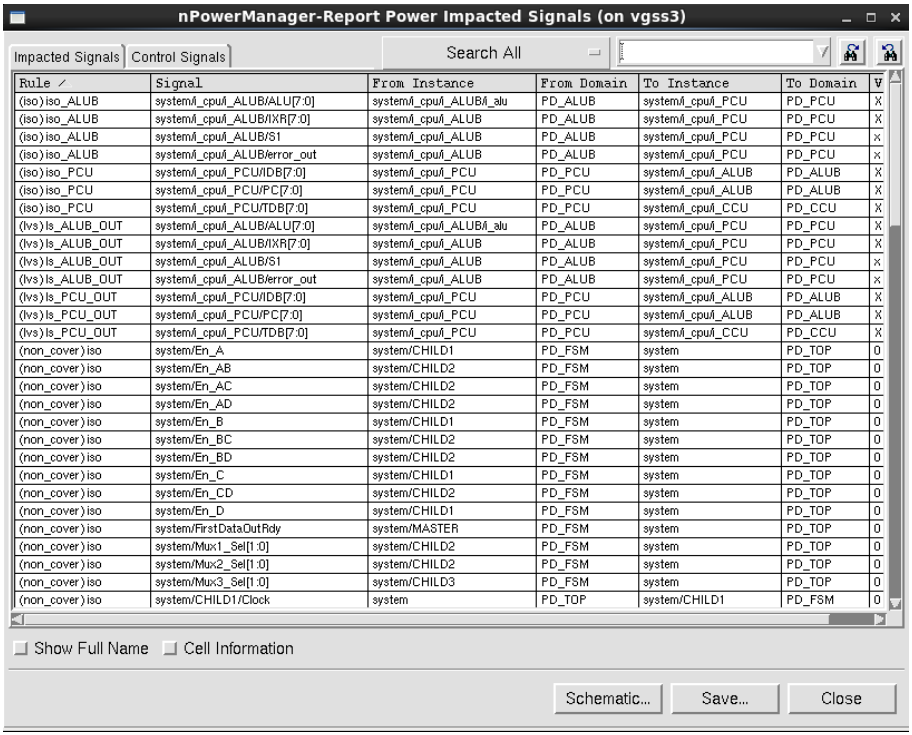


Figure: Scope Lists Form

The *Scope Lists* form includes the following options and fields:

- **Scope List:** The scope name can be entered by manually typing directly in the text field or by dragging a scope and dropping it in the text field. The wildcard characters (* and ?) are supported. When the scope is not specified, scopes in the whole design are reported after clicking the **OK** button.
- **Add:** After a scope is specified in the text field, click this button to add the specified scope to the scope list.
- **Delete Selected:** Click this button to delete the selected scope from the scope list.
- **Delete All:** Click this button to delete all scopes in the scope list.
- **Delimiter:** Specify the delimiter separating the hierarchy names. The default delimiter is the period (.) character.
- **Match Case:** When this option is turned *on*, the searching is case-sensitive. When this option is turned *off*, case is not considered during a search. The default is *on*.
- **Include Sub-scopes:** After clicking the **OK** button when this option is turned *on*, all sub-scopes under the selected scope are shown in the scope list. After clicking the **OK** button when this option is turned *off*, sub-scopes under the selected scope are not shown in the scope list. The default is *off*.

After clicking the **OK** button, the *nPowerManager - Report Power Impacted Signals* form opens and lists the details of the power-impacted and control signals for the selected scopes.



Rule	Signal	From Instance	From Domain	To Instance	To Domain	V
(iso) iso_ALUB	system/i_cpu/i_ALUB/ALU[7:0]	system/i_cpu/i_ALUB/i_alu	PD_ALUB	system/i_cpu/i_PCU	PD_PCU	X
(iso) iso_ALUB	system/i_cpu/i_ALUB/IXR[7:0]	system/i_cpu/i_ALUB	PD_ALUB	system/i_cpu/i_PCU	PD_PCU	X
(iso) iso_ALUB	system/i_cpu/i_ALUB/S1	system/i_cpu/i_ALUB	PD_ALUB	system/i_cpu/i_PCU	PD_PCU	X
(iso) iso_ALUB	system/i_cpu/i_ALUB/error_out	system/i_cpu/i_ALUB	PD_ALUB	system/i_cpu/i_PCU	PD_PCU	X
(iso) iso_PCU	system/i_cpu/i_PCU/IDB[7:0]	system/i_cpu/i_PCU	PD_PCU	system/i_cpu/i_ALUB	PD_ALUB	X
(iso) iso_PCU	system/i_cpu/i_PCU/PC[7:0]	system/i_cpu/i_PCU	PD_PCU	system/i_cpu/i_ALUB	PD_ALUB	X
(iso) iso_PCU	system/i_cpu/i_PCU/TDB[7:0]	system/i_cpu/i_PCU	PD_PCU	system/i_cpu/i_CCU	PD_CCU	X
(lvs) ls_ALUB_OUT	system/i_cpu/i_ALUB/ALU[7:0]	system/i_cpu/i_ALUB/i_alu	PD_ALUB	system/i_cpu/i_PCU	PD_PCU	X
(lvs) ls_ALUB_OUT	system/i_cpu/i_ALUB/IXR[7:0]	system/i_cpu/i_ALUB	PD_ALUB	system/i_cpu/i_PCU	PD_PCU	X
(lvs) ls_ALUB_OUT	system/i_cpu/i_ALUB/S1	system/i_cpu/i_ALUB	PD_ALUB	system/i_cpu/i_PCU	PD_PCU	X
(lvs) ls_ALUB_OUT	system/i_cpu/i_ALUB/error_out	system/i_cpu/i_ALUB	PD_ALUB	system/i_cpu/i_PCU	PD_PCU	X
(lvs) ls_PCU_OUT	system/i_cpu/i_PCU/IDB[7:0]	system/i_cpu/i_PCU	PD_PCU	system/i_cpu/i_ALUB	PD_ALUB	X
(lvs) ls_PCU_OUT	system/i_cpu/i_PCU/PC[7:0]	system/i_cpu/i_PCU	PD_PCU	system/i_cpu/i_ALUB	PD_ALUB	X
(lvs) ls_PCU_OUT	system/i_cpu/i_PCU/TDB[7:0]	system/i_cpu/i_PCU	PD_PCU	system/i_cpu/i_CCU	PD_CCU	X
(non_cover) iso	system/En_A	system/CHILD1	PD_FSM	system	PD_TOP	0
(non_cover) iso	system/En_AB	system/CHILD2	PD_FSM	system	PD_TOP	0
(non_cover) iso	system/En_AC	system/CHILD2	PD_FSM	system	PD_TOP	0
(non_cover) iso	system/En_AD	system/CHILD2	PD_FSM	system	PD_TOP	0
(non_cover) iso	system/En_B	system/CHILD1	PD_FSM	system	PD_TOP	0
(non_cover) iso	system/En_BC	system/CHILD2	PD_FSM	system	PD_TOP	0
(non_cover) iso	system/En_BD	system/CHILD2	PD_FSM	system	PD_TOP	0
(non_cover) iso	system/En_C	system/CHILD1	PD_FSM	system	PD_TOP	0
(non_cover) iso	system/En_CD	system/CHILD2	PD_FSM	system	PD_TOP	0
(non_cover) iso	system/En_D	system/CHILD1	PD_FSM	system	PD_TOP	0
(non_cover) iso	system/FirstDataOutRdy	system/MASTER	PD_FSM	system	PD_TOP	0
(non_cover) iso	system/Mux1_Sel[1:0]	system/CHILD2	PD_FSM	system	PD_TOP	0
(non_cover) iso	system/Mux2_Sel[1:0]	system/CHILD2	PD_FSM	system	PD_TOP	0
(non_cover) iso	system/Mux3_Sel[1:0]	system/CHILD3	PD_FSM	system	PD_TOP	0
(non_cover) iso	system/CHILD1/Clock	system	PD_TOP	system/CHILD1	PD_FSM	0

Figure: nPowerManager - Report Power Impacted Signals Form

The *nPowerManager - Report Power Impacted Signals* form includes the **Impacted Signals** tab and the **Control Signals** tab. The width of all columns in the **Impacted Signals** tab and the **Control Signals** tab can be changed by selecting the vertical line between column headers and dragging-left. This form includes the following option, text field, icons, and buttons:

- **Search Type:** In the **Impacted Signals** tab, select one search type option from **Search All, Search Rule, Search Signal, Search From Instance, Search From Domain, Search To Instance,** or **Search To Domain**. In the **Control Signals** tab, select one search type option from **Search All, Search Rule, Search Isolation Condition, Search Save Condition,** or **Search Restore Condition**. Then, click the **Find Previous** or **Find Next** icons to find the previous or next string in a specific column or all columns. The default search type is **Search All**.

NOTE: If the string that you search for does not exist in the columns under **Impacted Signals**/**Control Signals** tab, a warning message is issued to indicate that the specified string cannot be found.

- **Find Previous**  **/Find Next**  **:** Click these icons to find the previous or next string specified in the **Find String** text field.
- **Show Full Name:** When this option is turned *on*, the full name of rules and power domains are displayed. The default is *off*.
- **Cell Information:** When this option is turned *on*, **Iso/Lvs Instance**, **Cell Name**, and **Enable Pin** columns are displayed in the **Impacted Signals** tab. The default is *off*.

NOTE: This option is available in the **Impacted Signals** tab.

- **Schematic:** When a signal is selected and this button is clicked (or the **New Schematic** -> **Connectivity** option in the right-click menu is invoked or the bind key **Ctrl+S** is used), a flattened schematic window opens with the trace connectivity results displayed.

NOTE: This button is disabled in the **Control Signals** tab.

- **Save:** Click this button to open the *Save Power Impacted Signals* form and save the power impacted and control signal information in text format.
- **Close:** Click this button to close the *nPowerManager - Report Power Impacted Signals* form.

Impacted Signals Tab

The **Impacted Signals** tab displays the rules, from and to instances, and from and to domains in a tabular format. All data in the **Impacted Signals** tab can be dragged from the *nPowerManager - Report Power Impacted Signals* form and dropped to the *nTrace* or *nWave* windows, and vice versa.

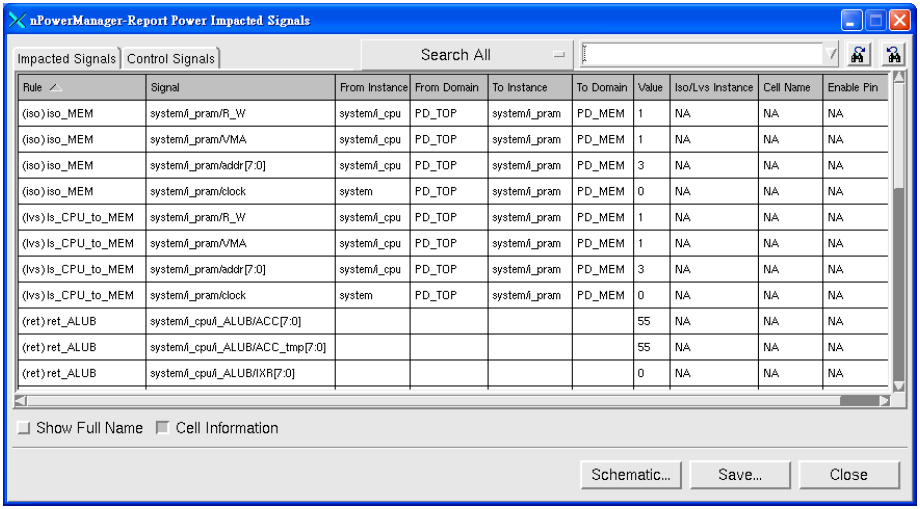


Figure: nPowerManager-Report Power Impacted Signals Form -Impacted Signals Tab

The columns in the **Impacted Signals** tab can be sorted by left-clicking the triangle or upside-down triangle on the column headers. The data can be selected by left-clicking anywhere along the row in the table. Multiple selections are supported by holding the **Shift** key (selects a range) or the **Ctrl** key (selects non-contiguous items) and left-clicking or dragging-left. The column headings are summarized as follows:

- **Rule:** Displays rules or expressions for rule-covered, cell-covered, RTL-covered or non-covered impacted signals:
  - **Cell-covered signals:** (iso_cell) and (lvs_cell) are displayed with the prefix to indicate Isolation cell rules and Level-shifter cell rules respectively.
  - **Non-covered signals:** (non_cover)iso and (non_cover)lvs are displayed with the prefix to indicate the non-covered Isolation rules and Level-shifter rules.
  - **RTL-covered signals:** (iso)*RTLexpression* is displayed with the prefix to indicate RTL-covered rules.
  - **Rule-covered signals:** (iso)*RuleName*, (lvs)*RuleName* and (ret)*RuleName* are displayed with the prefix to indicate the Isolation, Retention, and Level-shifter rules respectively.
  - **SPA signals:** (spa) is displayed to indicate that the signal is impacted by the SPA rule.

- **SRSN signals:** (srsn) is displayed to indicate the signal is impacted by the SRSN rule.

NOTE: The SPA rule has a higher priority than the SRSN rule. When a signal is specified using both the SPA rule and SRSN rule, the signal is shown to be impacted by the SPA rule only.

When you double-click a rule, the rule created command is highlighted in the *Power Source Code* pane.

- **Signal:** Displays the impacted signal name. When you double-click a signal, the definition of the signal is highlighted in the *Source Code* pane.

NOTE: Impacted signals refer to HDL signals whose simulation values are impacted by Isolation/Level-shifter/Retention settings in UPF/CPF files.

- **From Instance:** Displays the from instance of the impacted signal. When you double-click a from instance, the selected instance is set as the active scope in the design browser and *Source Code* panes.
- **From Domain:** Displays the from domain of the impacted signal. When you double-click a from domain, the selected power domain is set as the active power domain.
- **To Instance:** Displays the to instance of the impacted signal. When you double-click a to instance, the selected instance is set as the active scope in the design browser and *Source Code* panes.
- **To Domain:** Displays the to domain of the impacted signal. When you double-click a to domain, the selected power domain is set as the active power domain.
- **Value:** Displays the FSDB value of the impacted signal. If an FSDB file has not been loaded, 'NA' is shown.

NOTE: If a power domain is unknown, the **From Domain** and the **To Domain** columns in the *Report Power Impacted Signals* form displays *Undefined* string.

When the **Cell Information** option is turned *on*, the following columns are available:

- **Iso/Lvs Instance:** Displays the HDL Isolation/Level-shifted instance full name for the cell-covered signal. Double-clicking a cell in this column jumps to the corresponding module definition in the design browser pane and the *Source Code* pane. If the information is not available, 'NA' is shown.

- **Cell Name:** Displays the HDL Isolation/Level-shifter instance cell name for the cell-covered signal. If the information is not available, 'NA' is shown.
- **Enable Pin:** Displays the enable pin of the HDL Isolation/Level-shifter instance defined in the Isolation/Level-shifter cell command. If the information is not available, 'NA' is shown.

The following right-click options are available in the **Impacted Signals** tab of the *nPowerManager - Report Power Impacted Signals* form:

- **Show Definition in Source Window:** When an instance or a signal is selected, invoking this option shows the definition in the *Source Code* pane. When a rule or a domain is selected, this option shows the definition in the *Power Source Code* pane.
- **Add Signal(s) to Waveform:** When a signal is selected (multiple selections are supported), invoking this option adds the selected signal to the synchronized *nWave* pane. Alternatively, press the **Ctrl+W** bind key to add the selected signal(s) to the *nWave* window.
- **New Schematic -> Connectivity:** This option creates a new partial flattened schematic window based on the results from tracing the connectivity of the selected signal. Alternatively, press the **Ctrl+S** bind key to display the trace connectivity results of the selected signal in *nSchema*.

Control Signals Tab

The **Control Signals** tab displays the rules of Isolation and Retention signals and related control conditions in a tabular format.

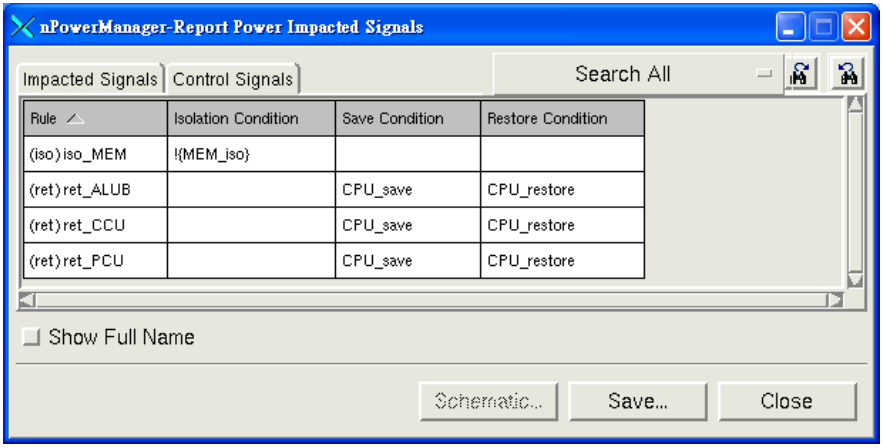


Figure: nPowerManager - Report Power Impacted Signals Form - Control Signals Tab

Each column can be sorted by clicking the triangle or upside-down triangle on the column headers. The column headings are summarized as follows:

- **Rule:** Displays Isolation or Retention rules with the (iso) or (ret) prefix
- **Isolation Condition:** Displays the Isolation condition of the signal
- **Save Condition:** Displays the Save condition of the signal
- **Restore Condition:** Displays the Restore condition of the signal

When a rule, isolation condition, save condition, or restore condition is double-clicked, the selection is highlighted simultaneously in the *Power Domain Tree*, *Power State Table/Power Mode Table*, and the *Power Source Code* panes.

Check Power Sequence

Menu Bar: Power -> Check Power Sequence

NOTE: This option is available when CPF/UPF files and the associated FSDB file are already loaded.

This option opens the *Check Power Sequence* form where the power rules to be checked may be selected and the check results may be reviewed. This option checks whether the results in the FSDB file matches the intent in the UPF/CPF file using typical power rules defined in the Verdi platform.

The *Check Power Sequence* includes the **Rules** tab, the **Report** tab, and the following buttons:

- **Save:** Click this button to open the *Save Power Sequence Report to File* form where the directory structure can be viewed and a file name to save the results to can be specified. This button is enabled after the rules have been checked. Refer to the *Check Power Sequence Report Format* section in *Appendix E* for details.
- **Restore:** Click this button to open the *Restore Power Sequence Report from File* form where the directory structure can be viewed and a file name to restore the results from can be specified.
- **Check:** Click this button to check the selected rules on the **Rules** tab and report the results on the **Report** tab.
- **Close:** Click this button to close the form.

Rules Tab

There are three categories of power rules: **Isolation**, **Power Domain**, and **Power Mode**. Toggle *on* or toggle *off* the check box before the rule to enable or disable the rule for checking. Checking a category enables all rules under the category.

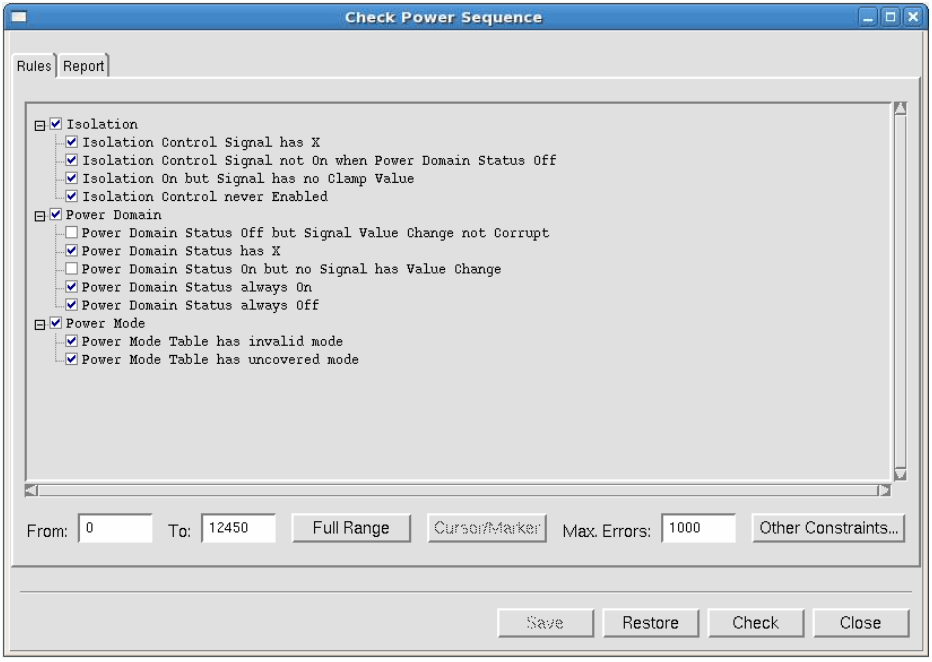


Figure: Check Power Sequence Form - Rules Tab

The **Rule** tab includes the following rules:

- **Isolation**

NOTE: For Isolation rules, only isolation strategies of the selected power domains in the *Other Constraints* form is checked. **Isolation Control Signal has X:** This error or warning is reported when the Isolation control signal has an unknown value. If the signal has multiple unknown values, only the first unknown value is shown.

NOTE: Before the value becomes stable, the initial unknown value is not included in the checking.

- **Isolation Control Signal not On when Power Domain Status Off:** This error or warning is reported when some unknown logic might be propagated.

- **Isolation On but Signal has no Clamp Value:** This error or warning is reported when some unknown logic might be propagated.
- **Isolation Control never Enabled:** This error or warning is reported when low testing coverage or a functional bug in Isolation cell(s) occurred.

- **Power Domain**
  - **Power Domain Status Off but Signal Value Change not Corrupt:** This error or warning is reported when the power status for some signals does not meet expectations.
    An information message is reported when checking the "Power Domain Status Off but Signal Value Change not Corrupt" rule, because it can take time to execute. This rule is toggled *off* by default.
  - **Power Domain Status has X:** This error or warning is reported when the Power Domain status has an unknown value. If the Power Domain has multiple unknown values, only the first unknown value is shown.

NOTE: Before the value becomes stable, the initial unknown value is not included in the checking.

  - **Power Domain Status On but no Signal has Value Change:** This error or warning is reported when some Power Domains are shutdown to save power consumption.
    An information message is reported when checking the "Power Domain Status On but no Signal has Value Change" rule, because it can take time to execute. This rule is toggled *off* by default.
  - **Power Domain Status always On:** This error or warning is reported when low testing coverage or a functional bug in a Power Domain occurred. If cells are defined as always-on, this warning is still reported.
  - **Power Domain Status always Off:** This error or warning is reported when low testing coverage or a functional bug occurs in a Power Domain.
- **Power Mode**

NOTE: For Power Mode rules, only Power Mode Tables in the corresponding scopes of the power domains selected in the *Other Constraints* form are checked.

  - **Power Mode Table has invalid mode:** This error or warning is reported when a bug exists or the Power Mode table needs to be modified to include more power modes. All invalid modes are reported.

- **Power Mode Table has uncovered mode:** This error or warning is reported when low testing coverage occurred.

The following text fields and options are included:

- **From/To:** Enter the start and end time respectively in the **From** and **To** text fields to specify the time duration. The time unit is the current Verdi time unit.
- **Full Range:** Click this button to obtain the full time range of the imported FSDB file.
- **Cursor/Marker:** Click this button to obtain the cursor and marker time in the primary waveform window.

NOTE: This option is enabled when a waveform window is opened.

- **Max Errors:** Specify the maximum number of errors to report. The default is *1000*.
- **Other Constraints:** Click this button to open the *Other Constraints* form where power domains of interest can be selected.

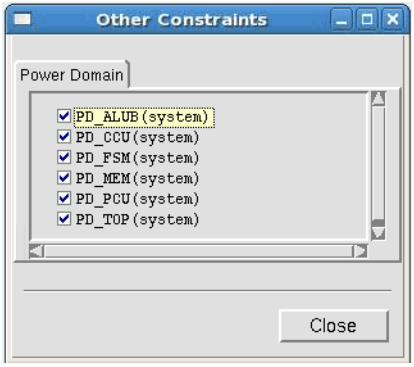


Figure: Other Constraints Form

To view power domains of interest, drag a power domain from the *Power Domain Tree* pane and drop it in the *Other Constraints* form. Multiple power domain selections are supported.

NOTE: For Isolation rules, only the isolation strategies in the selected power domains of the *Other Constraints* form are checked.
For Power Mode rules, only Power Mode Tables in the corresponding scopes of the power domains selected in the *Other Constraints* form are checked.

Report Tab

After the **Check** button is clicked, violations are reported on this tab using the same power rule categories as the **Rules** tab. Click the plus/minus symbol before the rule or double-click the row to expand/collapse the results.

To search a specific string, enter the string in the **Search** text field, and matched items in the report are highlighted. Searching with the wildcard characters (for example, "Power*", "No Erro?") is supported.

After expanding an error or warning rule grouping, an error or warning line may be dragged from this tab and dropped in the *nWave* or *nTrace* windows to display the associated signals or code.

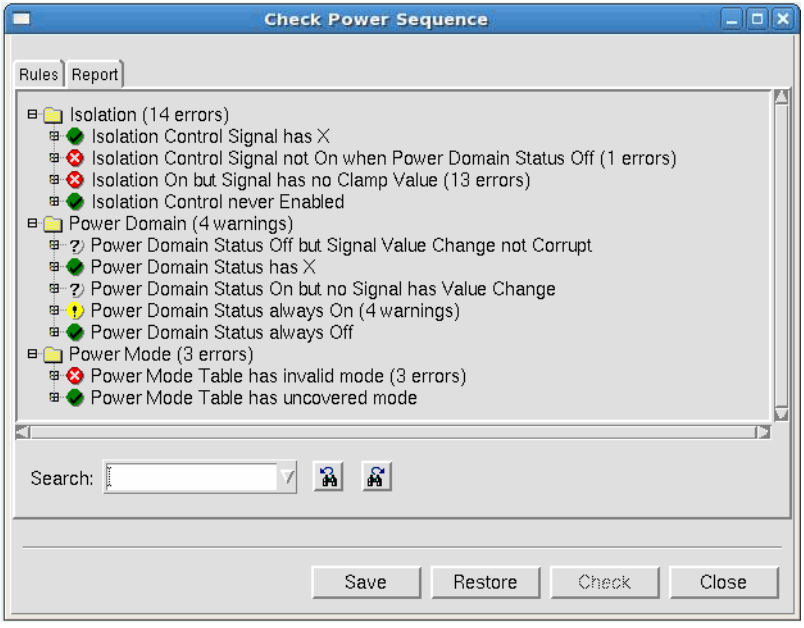


Figure: Check Power Sequence Form - Report Tab

The following icons are included on this tab:

- **No Error** : This icon indicates that no violations exist for the associated rule.
- **Error** : This icon indicates that at least one violation was found for the associated rule.
- **Not Checked** : This icon indicates that the associated rule was not enabled for checking.
- **Warning** : This icon indicates that at least one warning was found for the associated rule.

The following options are available on the right-click menu of a warning or error line:

- **Show Definition:** This option locates the definition for the selected line and displays the definition in the *Power Source Code*, the *Power Mode Table*/ *Power State Table*, and the *Power Domain Tree* pane. For Isolation control or power domain signals, the definition is shown in the Power panes. For impacted signals, the signal definition is shown in the *nTrace Source Code* pane, and the Isolation control definition is shown in the Power panes.

NOTE: The **Show Definition** right-click menu is not available for the "Power Mode Table has invalid mode" and "Power Mode Table has uncovered mode" rules.

- **Add Signal(s) to Waveform:** This option adds the waveform for the corresponding signals of the selected line to the *nWave* window. Alternatively, double-click the line to add the signals to the waveform.
- **Show Schema:** This option traces the connectivity signals for the selected signal and displays the results in the *nSchema* window.

NOTE: The **Show Schema** right-click menu is not available for the "Power Mode Table has invalid mode" and "Power Mode Table has uncovered mode" rules.

The following options are available on the right-click menu of blank row:

- **View by Rule:** Invoke this option to view reports in the Isolation, Power Domain, and Power Mode rule categories. The default is **View by Rule**.
- **View by Hierarchy:** Invoke this option to view reports in scope hierarchy. The hierarchy for each rule is defined as follows:

Rule Name	Scope Location
Isolation Control Signal has X	Where isolation control signal is located
Isolation Control Signal not On when Power Domain Status Off	Where isolation control signal is located
Isolation On but Signal has no Clamp Value	Where signal that has no clamp value is located
Isolation Control never Enabled	Where isolation control signal is located
Power Domain Status Off but Signal Value Change not Corrupt	Where signal that has non-corrupted value change is located

Power Domain Status has X	Where power domain signal is located
Power Domain Status On but no Signal has Value Change	Where power domain signal is located
Power Domain Status always On	Where power domain signal is located
Power Domain Status always Off	Where power domain signal is located
Power Mode Table has invalid mode	Where power mode signal is located
Power Mode Table has uncovered mode	Where power mode signal is located

- **Expand All:** Invoke this option to expand all sub-folders.
- **Collapse All:** Invoke this option to collapse all folders to the top-level categories.

List HDL Signals

Menu Bar: Power -> List HDL Signals

This option opens the *List HDL Signals* form where all HDL signals used in the CPF/UPF files are listed. Clicking the **Save** button opens the *Save HDL Signals* form where the HDL Signals can be saved into a text file.

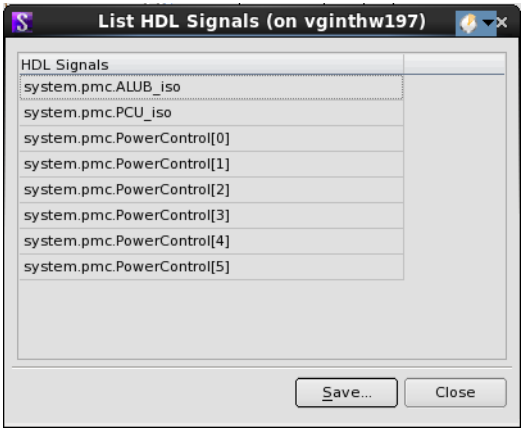


Figure: List HDL Signals Form

Highlight Power Domain

Menu Bar: Power -> Highlight Power Domain

This option opens the *Highlight Power Domain* form where all currently loaded power domains are listed and the colors for each power domain can be specified.

In the *Power Domain Tree* pane, sub-domains and their parent domains are displayed using the same colors. Instances in the design browser pane also displays the same colors as the power domains where these instances are located.

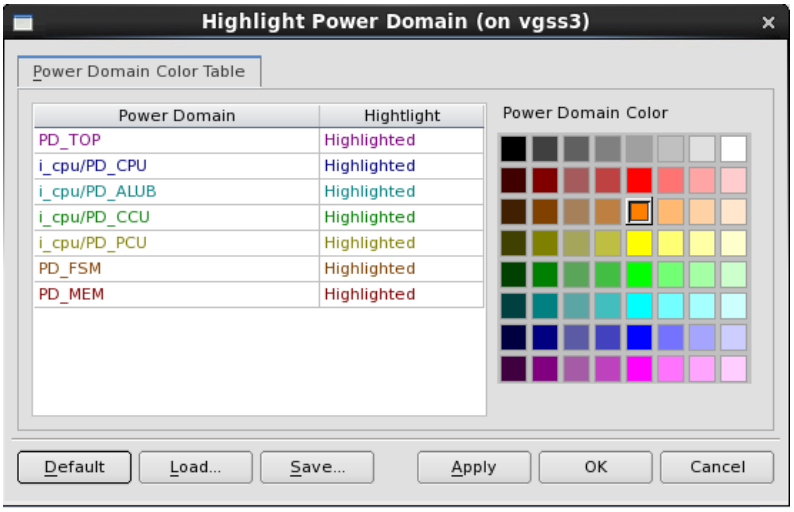


Figure: Highlight Power Domain Form

The **Power Domain Color** palate is a general color palate to specify the preferred color for each power domain.

Default: Click this button to return all power domains to their default colors.

Load: Click this button to open the *Load Power Domain Highlight Color* form and load previously saved power domain highlight information.

Save: Click this button to open the *Save Power Domain Highlight Color* form and save the power domain highlight information in text format.

New Power Map

Full Power Map

Menu Bar: Power -> New Power Map -> Full Power Map

Toolbar Icon: 

This option opens the *Power Map* pane displaying a schematic view of power aware objects.

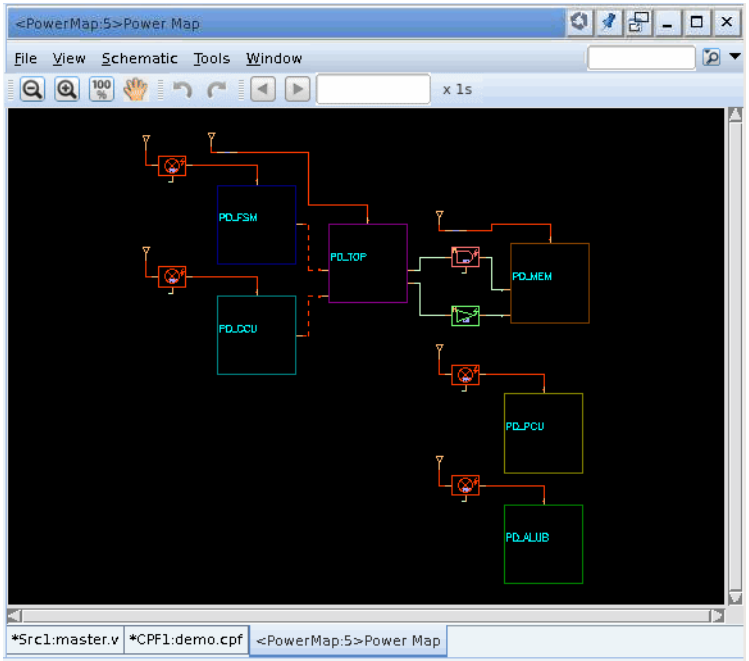
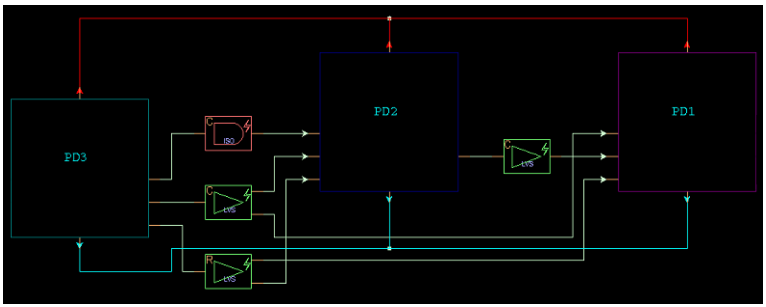


Figure: Power Map Pane

In the *Power Map* pane, the  icon represents an Isolation command (rule), the  icon represents a Level-shifted command (rule), and the  icon represents a power switch cell.



Refer to the *Power Map and Partial Power Map Panes* section for details.

Partial Power Map

Menu Bar: Power -> New Power Map -> Partial Power Map

This option opens the *Partial Power Map* pane displaying a schematic view of the selected power aware objects on the *Power Source Code* pane. Power aware objects include power domain, Isolation, Level-shifter rule, power supply net, or power switch (UPF only) objects.

Power Source Code Pane Right-click Options

Add Object(s) to Waveform

Bind Key: Ctrl+W

Refer to the **Add Object(s) to Waveform** option in *Power Domain Tree pane Right-click Options* section for details.

Set as Active Domain

NOTE: This option is enabled if an inactive power domain is selected. If the power domain that the option in the source code line belongs to is invalid, the right-click menu is not displayed.

This option is only valid when the specified line is a non-active line. When this option is invoked, the valid power domain that the option in the source code line belongs to is set as the active power domain.

Trace HDL Driver

Toolbar Icon: 

NOTE: This option is enabled if a signal is selected.

This option statically traces all possible drivers that exist in the design for the selected signal. Refer to the **Trace** -> **Driver** option in the *nTrace* chapter for details.

Trace HDL Load

Toolbar Icon: 

NOTE: This option is enabled if a signal is selected.

This option statically traces all possible loads that exist in the design for the selected signal. Refer to the **Trace** -> **Load** option in the *nTrace* chapter for details.

Connectivity

NOTE: This option is enabled if a signal is selected.

This option simultaneously gives the results of both the **Trace HDL Driver** and **Trace HDL Load** options with a single option.

HDL Active Trace

NOTE: This option is enabled if a signal is selected.

This option locates the active driver in the *Source Code* pane for the selected signal.

Display Liberty Definition

Refer to the **Display Liberty Definition** option description in the *nTrace* chapter for details.

Trace Power Driver

After an HDL signal is selected, this option statically traces all possible power drivers (including supply net, supply port, power switch, power domain, supply net function, or supply set handle object types) that exist in the design. Same nets of the traced signal's power driver results are also reported. The trace results are shown in two places:

1. In the *Power Source Code* pane indicator area with semicircles indicating driver results. The current selected result is highlighted with a color-filled semicircle  and additional results (if multiple results exist) are identified with a semicircle symbol  .
2. On the **OneTrace** tab in the *Message* pane which reports the location of each match. Double-click the line of a match to locate that match in the *Source Code* pane.

Trace Power Load

After an HDL signal is selected, this option statically traces all possible power loads (including supply net, supply port, power switch, power domain, supply net function, or supply set handle object types) that exist in the design. Same nets of the traced signal's power load results are also reported. The trace results are shown in two places:

1. In the *Power Source Code* pane indicator area with semicircles indicating load results. The current selected result is highlighted with a color-filled semicircle  and additional results (if multiple results exist) are identified with a semicircle symbol  .
2. On the **OneTrace** tab in the *Message* pane which reports the location of each match. Double-click the line of a match to locate that match in the *Source Code* pane.

Trace Power Connectivity

After an HDL signal is selected, this option simultaneously gives the results of both the **Trace Power Driver** and **Trace Power Load** options. The results are displayed in the same manner as described above.

New Flattened Schematic Window

NOTE: This option is enabled when a supply net is selected.

This option opens a flattened schematic window for the selected supply net to show the supply net, all connected HDL instances of this supply net, and connections between the supply net and HDL instances.

NOTE: *nSchema* does not support tracing for the HDL pin whose connection is defined in the **connect_supply_net** UPF command.

NOTE: *nSchema* does not show the supply net connection if the object selected to open the flattened window is an HDL object.

Trace Unknown Power State

NOTE: This option is available for CPF designs. It is enabled when the **Source -> Active Annotation** menu option is turned *on* and a power domain containing an unknown-value signal in the *Source Code* pane is selected.

When this option is invoked, source of the unknown power state is tracked and the trace results are displayed on the *Trace Unknown/Error Power State Results* form which is similar to the *Trace Active X Results* window opened by the **Trace X** option.

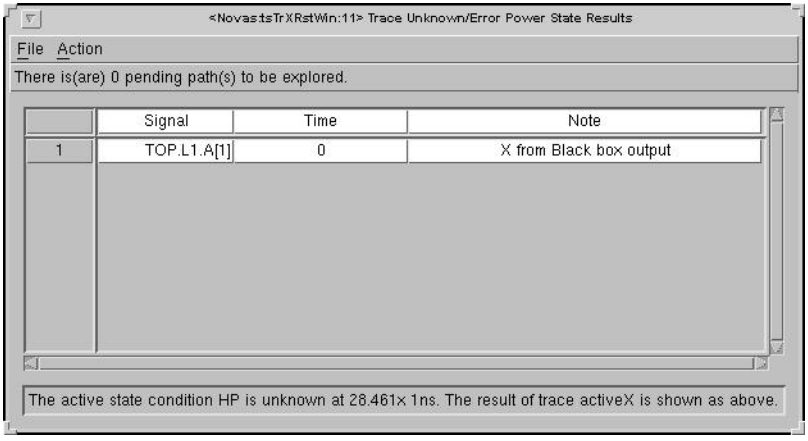


Figure: Trace Unknown/Error Power State Results Form

Refer to the *Trace Triggering X Results Window* section in the *Flow Views* chapter for details.

Trace Error Power State

NOTE: This option is available for UPF designs. It is enabled when the **Source -> Active Annotation** menu option is turned *on* and a power domain containing an error in the *Source Code* pane is selected.

When this option is invoked, source of the error power state is tracked and the corresponding state option is highlighted in the *Power Source Code* pane and the reason for the error power state and all active option expressions and issued supply nets are shown in the *Message* pane. Possible reasons for error power state include **-error_state** is true, **-control_port** is unknown, conflict between **-on_state** and **-off_state**, and none of **-on_state** and **-off_state** is active. Double-clicking a specific supply name jumps to the **create_power_switch** -**output_supply_port** whose output supply net is [supply_name] and highlight it in the *Source Code* pane.

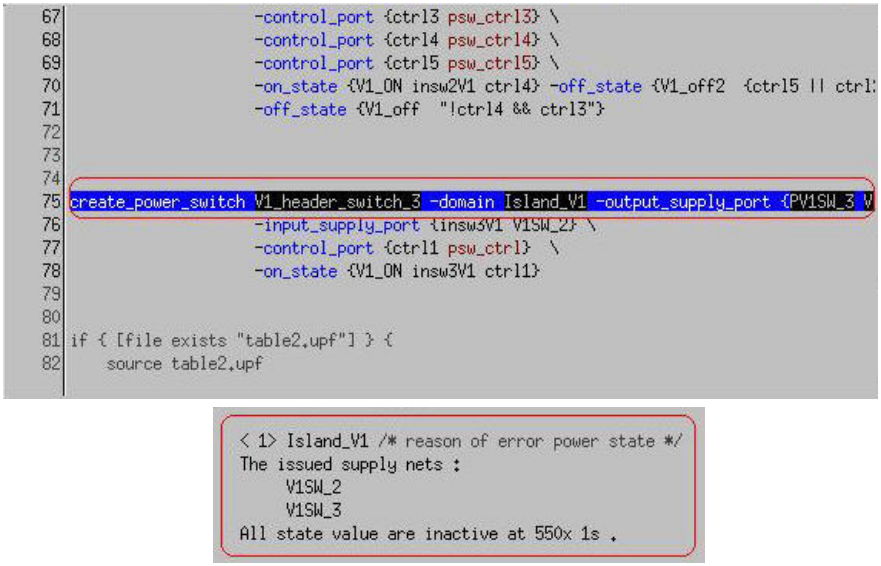


Figure: Trace Error Power State

If the reason for the error power state is caused by unknown **-control_port**, a message dialog is displayed to inform users to proceed with Behavior Analysis. The trace results are also displayed on the *Trace Unknown/Error Power State Results* form which is similar to the *Trace Active X Results* window opened by the **Trace X** option. Refer to the *Trace Triggering X Results Window* section in the *Flow Views* chapter for details.

Show Calling

Bind Key: Ctrl+G

NOTE: This option is available for the UPF designs when the power domain is created by the **load_upf** or **source** commands.

This option locates the **load_upf** or **source** command for the selected power domain and highlights the corresponding line in the *Power Source Code* pane. The power domain is also changed to the selected domain.

Show Tcl Variable

NOTE: This option is displayed only when you select a Tcl variable object in the *Power Source Code* pane.

This option opens a *Show Tcl Variable* form. This form reports the selected Tcl variable name, type, and value in tabular form.

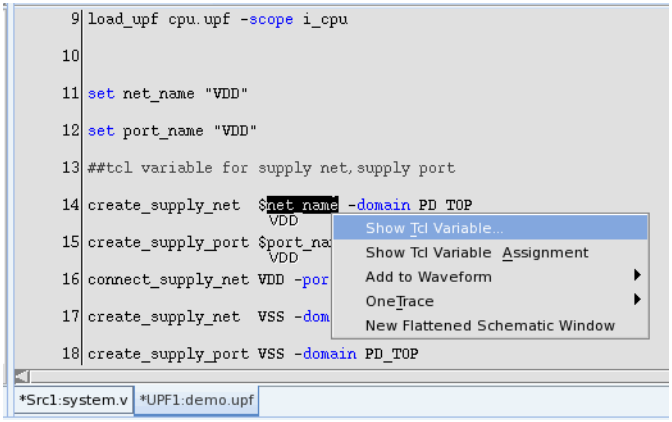


Figure: Show Tcl Variable

Show Tcl Variable Assignment

NOTE: This option is displayed only when you select a Tcl variable object in the *Power Source Code* pane.

This option shows the definition of the selected Tcl variable object selected in the *Power Source Code* pane.

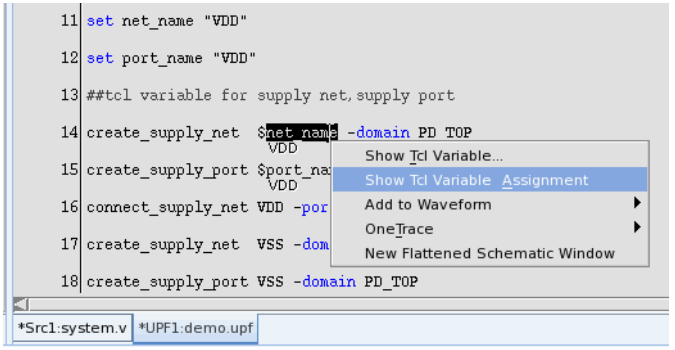


Figure: Show Tcl Variable Assignment

You can also double-click a Tcl variable object to view its definition in the *Power Source Code* pane.

Design Source Code Pane Right-click Options

Trace Power

The menu includes the following sub-options: **Driver**, **Load**, **Connectivity**, and **Active Trace**.

Driver

Bind Key: Shift+T

This option traces all of the options which impacts the HDL signal. The trace results are shown in two places:

1. In the *Power Source Code* pane indicator area with semicircles indicating driver results. The current selected result is highlighted with a color-filled semicircle and additional results (if multiple results exist) are identified with a semicircle symbol .
2. In the **OneTrace** tab of the *Message* pane which reports the name, instance, and location of each match. By double-clicking on the line of a match, the match can be easily located in the *Power Source Code* pane.

Load

This option traces all of the options which are impacted by the HDL signal. The trace results are shown in two places:

1. In the *Power Source Code* pane indicator area with semicircles indicating load results. The current selected result is highlighted with a color-filled semicircle  and additional results (if multiple results exist) are identified with a semicircle symbol  .
2. In the **OneTrace** tab of the *Message* pane which reports the name, instance, and location of each match. By double-clicking on the text line of a match, the match can be easily located in the *Power Source Code* pane.

Connectivity

This option simultaneously gives the results of both the **Driver** and **Load** options with a single command. The results are displayed in the same manner as described above.

Active Trace

This option locates the active driver in the *Source Code* pane for the selected power signal.

Add Instrumented Signals to Waveform

The menu includes the following sub-options:

Isolation Signals

This option adds the instrumented Isolation signals to *nWave*.

Retention Signals

This option adds the instrumented Retention signals to *nWave*.

SRSN Signals

This option adds the instrumented SRSN signals to *nWave*.

SPA Signals

This option adds the instrumented SPA signals to *nWave*.

Power

The menu includes the following sub-options: **Isolation Rules**, **Retention Rules**, **Level-shifted Rules**, **SRSN Rules**, **SPA Rules**, **CSN Rules**, and **ACK Rules**.

Isolation Rules

This option menu is available when you right-click an Isolation signal in the

design source code pane. The following sub-options are available:

- **Locate Iso. Command:** This option locates the Isolation command that applies to the Isolation signal in the *Power Source Code* pane.
- **Add Iso. Controls to Waveform:** This option adds control signals for the specified Isolation signal to the *nWave* window. The Isolation control signals are added to a group named *"Signal_Name#ISO#Rule_Name_Iso_Ctrl"* (*Signal_Name* is the selected signals' name) in the *nWave* signal pane.

Retention Rules

This option menu is available when right-clicking a Retention signal in the *design source code* pane. The following sub-options are available:

- **Locate Ret. Command:** This option locates the Retention command that applies to the Retention signal in the *Power Source Code* pane.
- **Add Ret. Controls to Waveform:** This option adds the control signals for the specified Retention signal to the *nWave* window. The save control signals are added to a group named "*Signal_Name#RET#Rule_Name_Save_Ctrl*" (*Signal_Name* is the selected signals' name) in the *nWave* signal pane. The restore control signals are added to a group named "*Signal_Name#RET#Rule_Name_Restore_Ctrl*" (*Signal_Name* is the selected signals' name) in the *nWave* signal pane.

Level-shifted Rules

This option menu is available when right-clicking a Level-shifted signal in the *design source code* pane. The following sub-options are available:

- **Locate Lvs. Command:** This option locates the Level-shifter command that applies to the Level-shifter signal in the *Power Source Code* pane.

SRSN Rules

This option menu is available when right-clicking an SRSN signal in the *design source code* pane. The following sub-options are available:

- **Locate SRSN Command:** This option locates the SRSN command that applies to the SRSN signal in the *Power Source Code* pane.
- **Add Supply Net to Waveform:** This option adds the supply net of the selected SRSN signals to the *nWave* window.

SPA Rules

This option menu is available when right-clicking an SPA signal in the *design source code* pane. The following sub-options are available:

- **Locate SPA Command:** This option locates the SPA command that applies to the SPA signal in the *Power Source Code* pane.
- **Add Supply Net to Waveform:** This option adds the supply net of the selected SPA signals to the *nWave* window.

CSN Rules

This option menu is available when right-clicking a CSN signal in the *design source code* pane. The following sub-options are available:

- **Locate CSN Command:** This option locates the CSN command that applies to the CSN signal in the *Power Source Code* pane.
- **Add Supply Net to Waveform:** This option adds the supply net of the selected CSN signals to the *nWave* window.

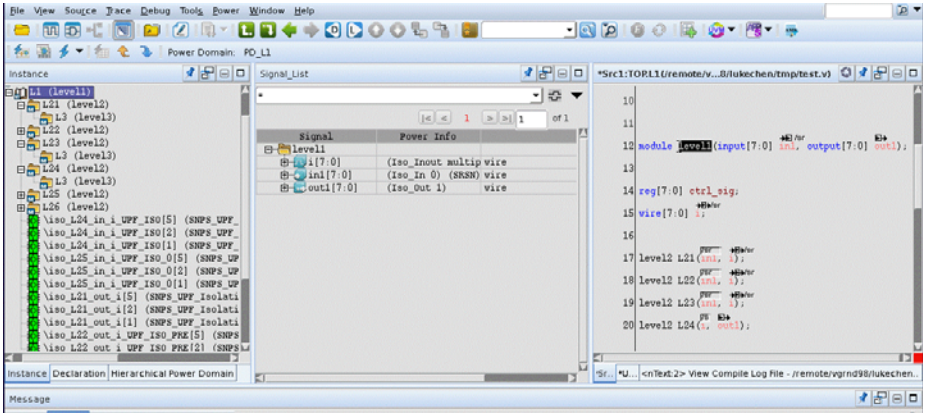
ACK Rules

This option menu is available when right-clicking a ACK signal in the *design source code* pane. The following sub-options are available:

- **Locate ACK Command:** This option locates the `create_power_switch` command that applies to the ACK signal in the *Power Source Code* pane.
- **Add Connected Power Switch to Waveform:** This option adds the related power switch object of the selected ACK signals to the *nWave* window.

Signal List Pane

In *nTrace* signal pane list, the hide/display action for the `iso_in`, `iso_out` and `iso_inout` signals is controlled by the **Isolation** filter type. The power information for `iso_in` signal is displayed as *Iso_In*. For `iso_out`, signal is displayed as *Iso_Out*, and for `iso_inout`, signal is displayed as *Iso_Inout*.



Signal List Pane Right-click Options

Add Instrumented Signals to Waveform

The menu includes the **Isolation Signals**, **Retention Signals**, **SRSN Signals**, and **SPA Signals** sub-options. Refer to the **Add Instrumented Signals to Waveform** option in the **Design Source Code Pane Right-click Options** section for details.

New Schematic

Refer to descriptions of the **Fan-in**, **Fan-out**, **Driver**, **Load**, **Connectivity** options under the **Tools -> New Schematic from Source** menu option for details.

nWave Window

The following options are available in the *nWave* window after CPF/UPF files are imported into the Verdi platform.

View Options

NOTE: These options are available when a CPF/UPF file is loaded.

Power

NOTE: Signals' dense blocks are not masked since the non-masking range may exist in dense blocks.

NOTE: When the **Mask Power Off**, **Mask Isolation**, **Mask Driving Power Off**, and **Mask Toggle** options are *on*, intersection ranges for multiple rules are masked.

Mask Power Off

Menu Bar: View -> Power -> Mask Power Off

Bind Key: Ctrl+M

When this toggle option is *on*, *nWave* masks the liberty-affected, set_port_attrbute (SPA), set_related_supply_net (SRSN), or power-off range for HDL signals.

Figure: Power Off Mask in the nWave Window

Mask Isolation

Menu Bar: View -> Power -> Mask Isolation

Bind Key: Ctrl+I

When this toggle option is *on*, *nWave* masks the isolation range for HDL signals according to the Isolation condition applied.

Figure: Isolation Mask in the nWave Window

Mask Retention

Menu Bar: View -> Power -> Mask Retention

When this toggle option is *on*, *nWave* masks the retention range for HDL signals according to the Retention condition applied.

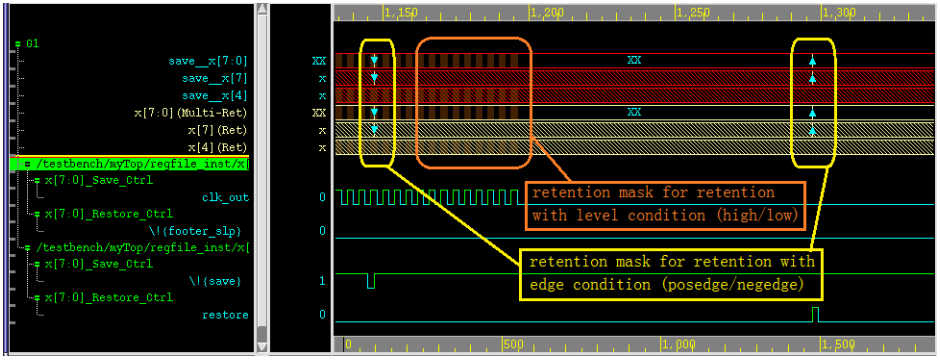


Figure: Retention Mask in the nWave Window

Mask Driving Power Off

Menu Bar: View -> Power -> Mask Driving Power Off

When this toggle option is *on*, *nWave* masks the driving power-off ranges. Driving power-off ranges are areas where the driver signal(s) of the selected signal are liberty-affected, set_port_attribute (SPA), set_related_supply_net (SRSN), or Power Off.

If a bus signal has multiple power domains post-fixed with *(Multi-PD)*, double-click the signal to expand the bus or use the **Power -> Add Power Domain** right-click option in the *nWave* signal pane to split the bus by power domain and check the precise mask in the expanded signals.

NOTE: In the Power Off situation where the power domain is unknown or not found, *nWave* does not mask the vacuous ranges.

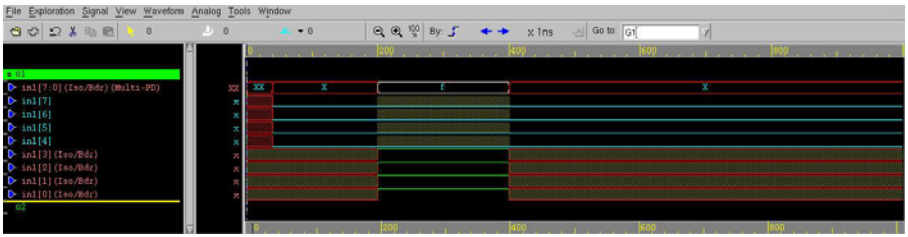


Figure: Vacuous Range Mask in the nWave Window

Mask Toggle

Menu Bar: View -> Power -> Mask Toggle

Bind Key: Ctrl+G

When this toggle option is *on*, *nWave* masks the range for HDL signals where the scope is already in a power-off corruption state and the input signals have value changes.

Signal Pane Right-click Options

Power

NOTE: These options are available when a CPF/UPF file is loaded.

The following sub-options are available:

Add Power Domain

This option adds the specified power domain signal(s) to the *nWave* window.

NOTE: If a bus signal has Isolation (Iso)/Retention (Ret)/Level-Shifter (Lvs)/ boundary port (Bdr) members, the signal along with the postfix (Iso/ Ret/Lvs/Bdr) is added in the *nWave* signal pane (for example, a[7:0](Iso)).
If there are multiple power domains or rules, a bus signal name along with the (Multi-) and/or (Multi-PD) postfix is shown in the *nWave* signal pane (for example, a[3:0](Multi-Ret), b[3:0](Multi-Lvs), in1[3:0](Multi-Iso), sig1[3:0](Multi-Bdr)(Multi-PD) where (Multi-Bdr) indicates multiple power domains exist and should be post-fixed with (Multi-PD)).

NOTE: The `-boundary_port` syntax is supported so that *nWave* can retrieve the correct power domain for the boundary ports.

Add Driving Power Info

This option creates a runtime message signal for the selected signal. As the time changes, the value change of this message signal dynamically indicates the full name of the driving power information. When multiple power information names exit, they are separated by "||".

Add Driver Supply Set

NOTE: This option is available when an SPA signal is selected.

This option adds the driver supply set of the selected SPA-affected signal to the *nWave* window.

Add Receiver Supply Set

NOTE: This option is available when an SPA signal is selected.

This option adds the receiver supply set of the selected SPA-affected signal to the *nWave* window.

Add Supply Net

This option adds the power net or ground net of the selected liberty-affected or SRSN-affected signal to the *nWave* window.

Add Power Mode

This option adds runtime signal(s) to display transitions of the corresponding power modes.

Retention Rules

This option menu is available when you right-click a Retention signal in the *nWave* signal pane. The following sub-options are available:

- **Locate Ret. Command:** This option locates the Retention command that applies to the Retention signal in the *Power Source Code* pane.
- **Add Ret. Controls:** This option adds the control signals for the specified Retention signal to the *nWave* window. The save control signals are added to a group named "*Signal_Name#RET#Rule_Name_Save_Ctrl*" (*Signal_Name* is the selected signals' name) in the *nWave* signal pane. The restore control signals are added to a group named "*Signal_Name#RET#Rule_Name_Restore_Ctrl*" (*Signal_Name* is the selected signals' name) in the *nWave* signal pane.

NOTE: If a control signal is dragged from another window to the *nWave* window and an existing signal in *nWave* has the same name, a new expression signal with a short expression and postfix (for example, (2), (3)) is created and added to the *nWave* signal pane. For example, "*\!!!rx_iso(2)*" is added to the *nWave* window whose logic expression is "*!!!system.i_cpu.rx_iso*".

- **Add Instrumented Signals:** This option adds the instrumented Retention signals to the *nWave* window.
- **Add Supply Net:** This option adds the supply net of the selected Retention signals to the *nWave* window.

Isolation Rules

This option menu is available when you right-click an Isolation signal in the *nWave* signal pane. The following sub-options are available:

- **Locate Iso. Command:** This option locates the Isolation command that applies to the Isolation signal in the *Power Source Code* pane.

NOTE: If the Isolation command is defined with a macro statement and this option is invoked, the ISO ASSIGN statement is automatically located in the RTL code.

- **Add Iso. Controls:** This option adds the control signals for the specified Isolation signal to the *nWave* window. The Isolation control signals are added to a group named *"Signal_Name#ISO#Rule_Name_Iso_Ctrl"* (*Signal_Name* is the selected signals' name) in the *nWave* signal pane.

NOTE: If the Isolation command is defined with a macro statement, the Isolation control signals are added to a group named *"Signal_Name#ISO#Extract_Iso_Ctrl"*.

- **Add Signals by Same Rules:** This option adds signals that are applied with the same Isolation command to the *nWave* window.

NOTE: This option is disabled if the Isolation command is defined with a macro statement.

- **Add Instrumented Signals:** This option adds the instrumented Isolation signals to the *nWave* window.
- **Add Supply Net:** This option adds the supply net of the selected Isolation signals to the *nWave* window.

Level-shifted Rules

This option menu is available when you right-click a Level-shifter signal in the *nWave* signal pane. The following sub-options are available:

- **Locate Lvs. Command:** This option locates the Level-shifter command that applies to the Level-shifter signal in the *Power Source Code* pane.
- **Add Signals by Same Rules:** This option adds the signals that are applied with the same Level-shifter command to the *nWave* window.

SRSN Rules

This option menu is available when you right-click an SRSN signal in the *nWave* signal pane. The following sub-options are available:

- **Locate SRSN Command:** This option locates the SRSN command that applies to the SRSN signal in the *Power Source Code* pane.
- **Add Instrumented Signals:** This command adds the instrumented SRSN signals to the *nWave* window.

- **Add Supply Net:** This option adds the supply net of the selected SRSN signals to the *nWave* window.

SPA Rules

This option menu is available when you right-click an SPA signal in the *nWave* signal pane. The following sub-options are available:

- **Locate SPA Command:** This option locates the SPA command that applies to the SPA signal in the *Power Source Code* pane.
- **Add Instrumented Signals:** This option adds the instrumented SPA signals to the *nWave* window.
- **Add Supply Net:** This option adds the supply net of the selected SPA signals to the *nWave* window.

Power Domain Tree Pane

The **Flatten Power Domain/Hierarchical Power Domain** tabs (also called the *Power Domain Tree* pane when referring to the pane as a whole) are docked to the same pane location as the design browser as a new tab. The **Flatten Power Domain** tab is displayed when a CPF file is loaded. The **Hierarchical Power Domain** tab is displayed when a UPF file is loaded.

Flatten Power Domain Tab

After the CPF file is loaded, the *Power Domain Tree* pane is docked to the design browser pane as a new **Flatten Power Domain** tab. The flattened power domain provides a module-based hierarchy view for the CPF design.

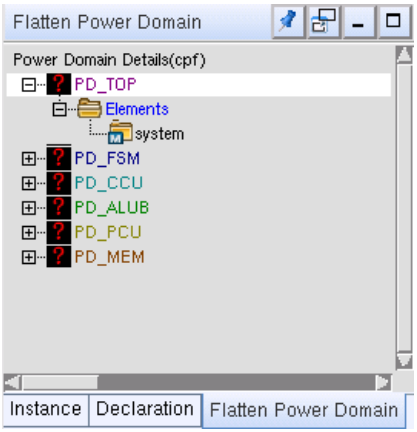


Figure: Flatten Power Domain Tab

This tab shows all power domains defined in the CPF design. Every domain is marked with a different color. These colors can be redefined by the **Power** -> **Highlight Power Domain** menu option or the **Highlight Power Domain** option (default is *on*) on the **General** -> **Power** page of the *Preferences* form (invoked with the **Tools** -> **Preferences** option in the *nTrace* window).

In the **Power Domain Details** pane, different power states are represented by different icons. Refer to the **Power Manager** tab of the *Legend* form opened by the **Help** -> **Legend** option in the *nTrace* chapter for details.

Hierarchical Power Domain Tab

After the UPF file is loaded, the *Power Domain Tree* pane is docked to the design browser pane as a new **Hierarchical Power Domain** tab. The power domain

hierarchy displays the power domain trees based on the UPF scope and the design hierarchy. At the end of the power domain hierarchy tree list, the power state table (indicated as PST in the hierarchy name) nodes and the supply list of the power state table nodes are built based on the UPF scope.

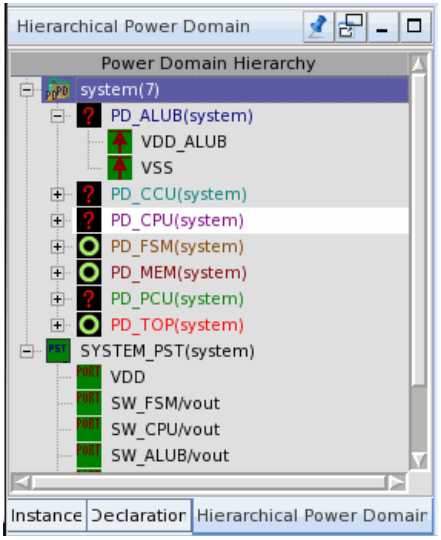


Figure: Hierarchical Power Domain Tab

In the **Power Domain Hierarchy** pane, different power states are represented by different icons. Refer to the **Power Manager** tab of the *Legend* form opened by the **Help** -> **Legend** option in the *nTrace* chapter for details.

Power Domain Tree Pane Right-click Options

The following right-click menu options are available for the *Power Domain Tree* pane.

Show Search/Filter

Bind Key: Ctrl+F

Toolbar Icon: 

This option provides filtering and searching features to navigate in the *Power Domain Tree* pane. By default, the **Filter** and **Search** fields are hidden. You can change this default setting using the **Power Domain Tree** preference option in the **Search/Filter** page (**General** -> **Search/Filter** -> **Enable Search/Filter on**: **Power Domain Tree**) of the *Preferences* form. The **Power Domain Tree** preference option is disabled by default.

Enter a string in the **Filter** field, select the filter type by clicking the **Filter Type** icon on the right, and press the **Enter** key. Tree nodes (including their parent nodes) matching the input string and the filter type has their power domain nodes displayed in the *Power Domain Tree* pane. Child nodes of the matched tree nodes are included and collapsed. The child nodes can be expanded as needed.

You can use the **Case Sensitive/Insensitive** icon on the right to enable or disable case-sensitivity for the string entered in the **Filter** and **Search** fields. By default, the searching and filtering mechanisms are case-sensitive. You can change this default setting using the **Match case** preference option in the **Search/Filter** page (**General** -> **Search/Filter** -> **Search/Filter**: **Match case**) of the *Preferences* form.

The following are the available filter types. Multiple selections of the filter types are supported.

- **Element**
- **Isolation**
- **Retention**
- **Switch**
- **Level Shifter**
- **Shut Off Condition** (available for CPF 1.0 and CPF 1.1 files)
- **Supply Set** (available for UPF 2.0 files)
- **Primary Power Net** (available for UPF 1.0 files)

- **Primary Ground Net** (available for UPF 1.0 files)
- **Internal Power Net** (available for CPF 1.1 files)
- **Internal Ground Net** (available for CPF 1.0 and CPF 1.1 files)
- **Secondary Domain** (available for CPF 1.1 files)
- **Mapping Domain** (available for CPF 1.1 files)
- **Power State Table** (available for UPF 1.0 files)
- **All**

Type a search string in the **Search** text field and press the **Enter** key to locate the first match. Click the **Previous Node** /**Next Node** icons to find the previous/next matching tree node.

In the **Filter** and **Search** fields, the wildcard characters (* or ?) are supported to specify the pattern-matching string.

The path for the selected node is automatically updated in the **Path** field. The vertical bar (|) character is the hierarchy delimiter.

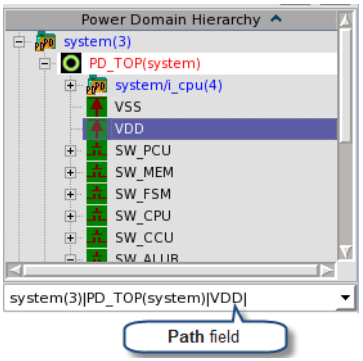


Figure: Path Field

Set as Active Domain

After a domain or a module is selected, this option sets the selected power domain in the *Power Domain Tree* pane as the current active power domain and all lines belonging to the power domain in the *Source Code* pane become active lines.

Show Definition

This option brings up the definition statement (shown in the *Source Code* pane) of the selected instance in the *Power Source Code* pane. An alternate method for determining the definition statement is to double-click an instance name in the *Source Code* pane or an instance node in the design browser pane. This locates

the definition statement (shown in the *Source Code* pane) of the selected instance.

Show Calling

Bind Key: Ctrl+G

Refer to the **Show Calling** option in the *Design and Power Source Code Panes* section for details.

Add Applied Signal(s) to Waveform

After an Isolation/no-Isolation command, a Retention/no-Retention command, or a Level-shifter/no-Level-shifter command is selected. This option adds the signals applied by the Isolation/no-Isolation command, Retention/no-Retention command, or Level-shifted/no-Level-shifted command to the synchronized *nWave* window.

Add Object(s) to Waveform

After a signal is selected, this option adds the selected power object(s) to the *nWave* window. The power objects include the power domain, supply net, supply port, supply set, power switch, and power state table.

Unhighlight All

Unhighlights the highlighted UPF policies when certain signals are dragged from the *Source Code* pane to *Power Domain Tree* pane.

Collapse to Parent Node

NOTE: This option is enabled when the selected node has a parent node.

This option collapses the selected node to its parent node.

New Flattened Schematic Window

NOTE: This option is enabled when a supply net is selected.

When the supply net is specified in the **connect_supply_net** UPF command, the new flattened schematic window only shows the connected HDL instance of this supply net.

When the supply is specified in commands that exclude the **connect_supply_net** UPF command, the new flattened schematic window shows all connected HDL instances of this supply net.

NOTE: *nSchema* does not support tracing for the HDL pin whose connection is defined in the **connect_supply_net** UPF command.

NOTE: *nSchema* does not show the supply net connection if the object selected to open the flattened window is an HDL object.

New Partial Power Map

Bind Key: Ctrl+N

After selecting a power domain, power switch, Isolation policy, no-Isolation policy, Level-shifter policy, no-Level-shifter policy, supply net, or supply set in the *Power Domain Tree* pane, invoke this option to open the *Partial Power Map* pane with the selected objects. Multiple selection is supported.

Refer to the **Partial Power Map** option in the *Power Map and Partial Power Map Panes* section for details.

Report Impacted Signal(s)

Bind Key: Ctrl+I

After a node of the Isolation, non-Isolation, Level-shifter, non-Level-shifter, Retention, or non-Retention rule is selected (multiple selection is supported), this option opens the *nPowerManager - Report Power Impacted Signals* form where the details of the power-impacted and control signals for the selected scopes are listed. Refer to the **Report Impacted Signals by Scope** option for details.

Expand All

This option expands all power domain tree nodes in the *Power Domain Tree* pane.

Collapse All

This option collapses all power domain tree nodes in the *Power Domain Tree* pane.

Full Domain Name

Refer to the **Power -> Full Scope Name** option description for details.

Power State/Mode Table Pane

The *Power State Table/Power Mode Table* panes are docked to the same pane location as the signal list pane as a new tab. The *Power State Table* pane is displayed when the **Power -> Show Power State Table** option is turned *on*. The *Power Mode Table* pane is displayed when the **Power -> Show Power Mode Table** option is turned *on*.

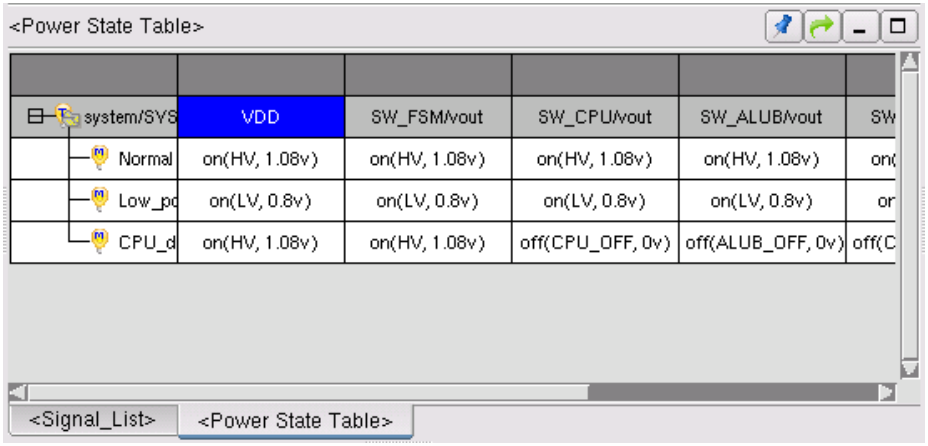


Figure: Power State Table Pane

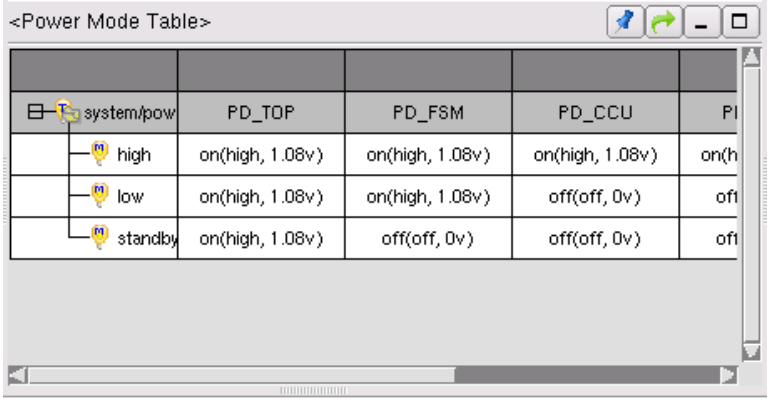


Figure: Power Mode Table Pane

Power State/Mode Table Pane Right-click Options

Show Find Box 

This option displays the **Find** text field and the **Find Previous** and **Find Next** icons in the *Power State Table* pane. To search for a string, type the string name in the **Find** text field and press **Enter**. The results are highlighted in the **Power State Table**. Click the **Find Previous** or **Find Next** icons to find the previous or next string specified in the **Find** text field.

Show Domain

NOTE: This option is available for the UPF files in the *Power State Table* pane.

When this option is invoked, the short domain name which is supplied primarily by the port is shown in the *Power State Table* pane. For example, if there is only one power domain 'PD_FSM' which is supplied by the net, (PD_FSM) is shown under the port. If there are two power domains 'PD_ALUB' and 'PD_FSM', (PD_ALUB,PD_FSM) is shown under the port. If there is no power domain, (none) is shown under the port.

Expand All

When this option is invoked, all rows in the power state/mode table are expanded.

Collapse All

When this option is invoked, all rows in the power state/mode table are collapsed.

Save as CSV File

This option opens the *Save as CSV File* form where the directory structure can be viewed and a file name can be specified to save the power state table in CSV file format. The default path is the current working directory.

Power Map and Partial Power Map Panes

The *Power Map* pane is displayed when the **Power -> New Power Map -> Full Power Map** menu option is invoked. The *Partial Power Map* pane is displayed when the **Power -> New Power Map -> Partial Power Map** menu option is invoked. The *Power Map* and *Partial Power Map* panes are docked to the same pane location as the source code/schematic as a new tab. These panes can become a standalone window by clicking the **Undock** icon on the toolbar. Refer to the *Icons for Dockable Panes* section for details on the menu bar icons.

After the initial *Partial Power Map* pane is created, additional power objects may be added by drag-and-drop from other panes.

NOTE: Only one power control signal can be selected at a time and dragged to another window. Selecting multiple control signals to drag-and-drop to another window is not supported in the *Power Map* pane.

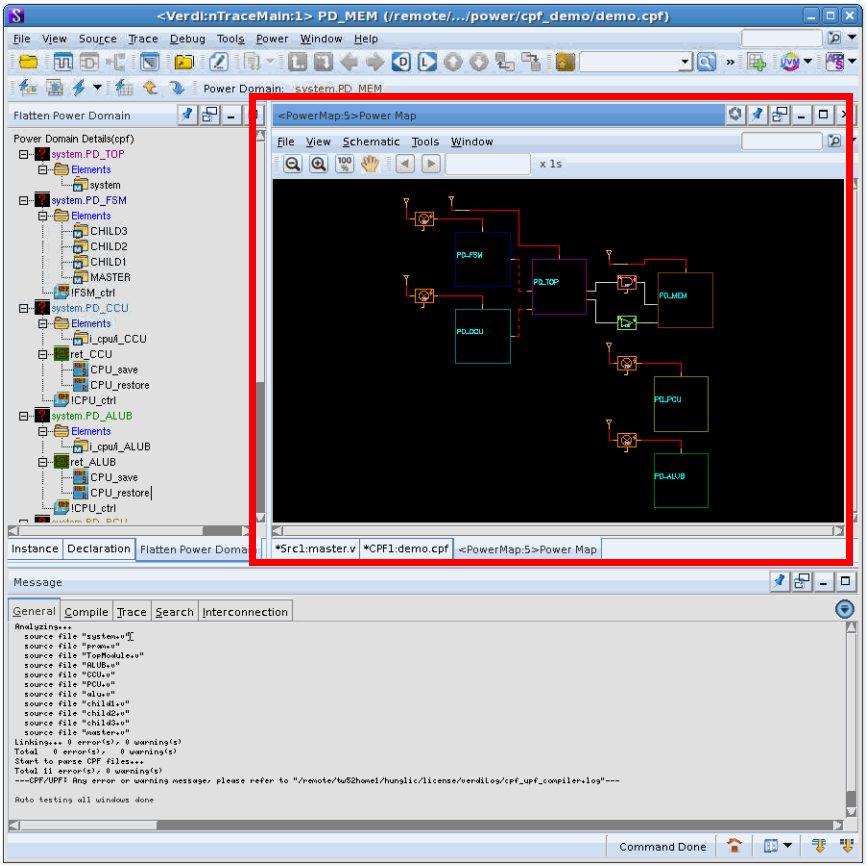


Figure: Power Map Pane

The menu bars for the *Power Map* and *Partial Power Map* panes are shown below:

Figure: Power Map Menu Bar

Figure: Partial Power Map Menu Bar

The menus for the *Power Map* and *Partial Power Map* panes are summarized and explained on the following pages.

File Options

Close Window

Menu Bar: File -> Close Window

This option closes the *Power Map* or *Partial Power Map* panes.

Edit Options

NOTE: These options are available in the *Partial Power Map* pane.

Undo

Menu Bar: Edit -> Undo

Toolbar Icon: 

This option undoes the previous action and removes whatever was added from the view.

Redo

Menu Bar: Edit -> Redo

Toolbar Icon: 

This option redoes the previous action and adds whatever was removed back to the view.

View Options

Full Domain Name

Menu Bar: View -> Full Domain Name

This toggle option turns the display of full power domain names *on* or *off*. The default is *off*.

Power Domain Name

Menu Bar: View -> Power Domain Name

Bind Key: Ctrl+D

This toggle option turns the display of power domain names *on* or *off*. The default is *on*.

Power Cell Name

Menu Bar: View -> Power Cell Name

Bind Key: E

This toggle option turns the display of power cell (Isolation/Level-shifted/power switch) names *on* or *off*. The default is *off*.

Net Name

Menu Bar: View -> Net Name

This toggle option turns the display of local net names *on* or *off*. The default is *off*.

Rule-covered Signals

Menu Bar: View -> Rule-covered Signals

Bind Key: Ctrl+R

NOTE: These options are available in the *Power Map* pane.

When this toggle option is turned *on*, the *Power Map* pane performs rule-covered checking and show the rule-covered signals in the *Power Map* pane. A rule-covered signal is shown with the letter "R" added on the upper-left corner such as  . The default is *on*.

Cell-covered Signals

Menu Bar: View -> Cell-covered Signals

Bind Key: Ctrl+O

NOTE: These options are available in the *Power Map* pane.

When this toggle option is turned *on*, the *Power Map* pane performs cell-covered checking and show the cell-covered signals in the *Power Map* pane. A cell-covered signal is shown with the letter "C" added on the upper-left corner such as  . The default is *off*.

Non-covered Signals

Menu Bar: View -> Non-covered Signals

NOTE: These options are available in the *Power Map* pane.

The *Power Map* pane can validate all signals in between power domains guarded with Isolation rules, if there is a possibility that for a domain the signal coming from ('from' domain) is *off* and the signal going to ('to' domain) is *on*. Unexpected results may propagate through such signals from the 'from' domain to the 'to' domain.

Similarly, signals should be guarded with Level-shifter rules if the signals' 'from' and 'to' domains are both in 'On'/'StandBy' states, but the domains' voltages are different.

For the DONT_CARE state of the domain, the Isolation/Level-shifter non-covered connections reported by the power mode table checking use the following rules:

Isolation Non-covered Power Mode Checking Rule

From Domain Status	To Domain Status	Report Isolation Non-covered
Off/DONT_CARE	On/StandBy/DONT_CARE	Yes
Off/DONT_CARE	XXXX	Yes
XXXX	On/StandBy/DONT_CARE/Off	Yes

Level-shifter Non-covered Power Mode Checking Rule

From Domain Status	To Domain Status	Report Level-shifted Non-covered
On/StandBy/ DONT_CARE	On/StandBy/DONT_CARE	Yes, if the voltage between domains is different.

On/StandBy/ DONT_CARE	XXXX	Yes
XXXX	On/StandBy/DONT_CARE/Off	Yes

The results are shown as non-covered nets in dotted lines (see the following figure for an example). A non-covered net contains all non-covered signals on the path from domain X to domain Y. The red dotted lines represent the non-covered Isolation nets; the green dotted lines represent the non-covered Level-shifter nets.

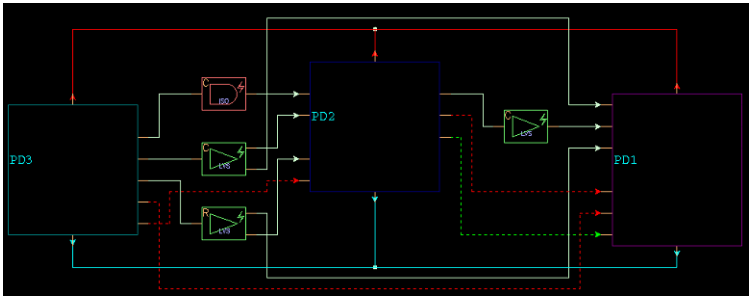


Figure: Non-Covered Signals in Power Map

Isolation Signals

Menu Bar: View -> Non-covered Signals -> Isolation Signals

Bind Key: Ctrl+I (Uppercase I)

When this toggle option is *on*, the *Power Map* pane performs Isolation non-covered checking. The default is *on*.

Level-shifter Signals

Menu Bar: View -> Non-covered Signals -> Level-shifter Signals

Bind Key: Ctrl+L

When this toggle option is *on*, the *Power Map* pane performs Level-shifter non-covered checking. The default is *on*.

Highlight Power Aware Object

Power Domain Color

Menu Bar:	View -> Highlight Power Aware Object -> Power Domain Color
Bind Key:	D

After the CPF/UPF file is loaded and the **Highlight Power Domain** option is turned *on* (located on the **General -> Power** page of the *Preferences* form), invoke this toggle option to highlight the power domain color. The default is *on*.

The highlight color can be changed by selecting the **Isolation Object** in the **Power Aware Object** section and specifying a preferred color on the **General -> Power** page of the *Preferences* form (invoked with the **Tools -> Preferences** option).

Isolation Cell Color

Menu Bar:	View -> Highlight Power Aware Object -> Isolation Cell Color
Bind Key:	S

After the CPF/UPF file is loaded and the **Highlight Power Aware Design Object(s)** option is turned *on* (located on the **General -> Power** page of the *Preferences* form), invoke this toggle option to highlight the Isolation cell color. The default is *on*.

The highlight color can be changed by selecting the **Isolation Object** in the **Power Aware Object** section and specifying a preferred color on the **General -> Power** page of the *Preferences* form (invoked with the **Tools -> Preferences** option).

Level-shifter Cell Color

Menu Bar:	View -> Highlight Power Aware Object -> Level-shifter Cell Color
Bind Key:	V

After the CPF/UPF file is loaded and the **Highlight Power Aware Design Object(s)** option is turned *on* (located on the **General -> Power** page of the

Preferences form), invoke this toggle option to highlight the Level-shifter cell color. The default is *on*.

The highlight color can be changed by selecting the **Level Shifted Object** in the **Power Aware Object** section and specifying a preferred color on the **General -> Power** page of the *Preferences* form (invoked with the **Tools -> Preferences** option).

Power Switch Cell Color

Menu Bar:	View -> Highlight Power Aware Object -> Power Switch Cell Color
Bind Key:	W

After the CPF/UPF file is loaded and the **Highlight Power Aware Design Object(s)** option is turned *on* (located on the **General -> Power** page of the *Preferences* form), invoke this toggle option to highlight the power switch cell color. The default is *on*.

The highlight color can be changed by selecting the **Power Switch Object** in the **Power Aware Object** section and specifying a preferred color on the **General -> Power** page of the *Preferences* form (invoked with the **Tools -> Preferences** option).

High Contrast

Menu Bar:	View -> High Contrast

When this toggle option is *on*, the schematic view is dimmed out except for the selected and traced set. The default is *off*.

Zoom

Zoom In

Menu Bar:	View -> Zoom -> Zoom In
Toolbar Icon:	
Bind Key:	Shift+Z

This option provides a close-up view of the content in the *Power Map* pane. The magnification of the viewing area is changed to half the magnification of the previous view.

NOTE: A specific area can also be zoomed by dragging-left to form a rectangle around the area to be zoomed.

Zoom Out

Menu Bar: View -> Zoom -> Zoom Out

Toolbar Icon: 

Bind Key: Z

This option enables more of the content to be seen in the *Power Map* pane at a reduced size. The magnification of the viewing area is changed to two times the magnification of the previous view from the center point in both the horizontal and vertical directions.

Zoom All

Menu Bar: View -> Zoom -> Zoom All

Toolbar Icon: 

Bind Key: F

This option shows all of the content of the *Power Map* pane.

Fit Selected Set

Menu Bar: View -> Zoom -> Fit Selected Set

This option fits the selected objects in the *Power Map* pane.

Pan

Toolbar Icon: 

Click this toolbar icon to move the schematic in the *Power Map* pane to the left, right, up, or down.

Pan Left

Menu Bar: View -> Pan -> Pan Left

Bind Key: Left Arrow

This option pans the *Power Map* pane to the left.

Pan Right

Menu Bar: View -> Pan -> Pan Right

Bind Key: Right Arrow

This option pans the *Power Map* pane to the right.

NOTE: Panning left and right can also be achieved by moving the horizontal scroll bar in the *Power Map* pane.

Pan Up

Menu Bar: View -> Pan -> Pan Up

Bind Key: Up Arrow

This option pans the *Power Map* pane up.

Pan Down

Menu Bar: View -> Pan -> Pan Down

Bind Key: Down Arrow

This option pans the *Power Map* pane down.

NOTE: Panning up and down can also be achieved by moving the vertical scroll bar in the *Power Map* pane.

Schematic Options

Find in Current Scope

Menu Bar: Schematic -> Find in Current Scope

Bind Key: A

This option opens the *Find Power Object* form which lists all power objects in the *Power Map* pane. Select one of the **Power Domain**, **Power Rule**, **Supply Net**, or **Library Power Cell** options to filter objects that are not of interest. Use the **Find** text field to find the desired power objects. Wildcard characters are supported.

The selected power domain or power rule in the *Find Power Object* form is highlighted in the *Power Map* pane.

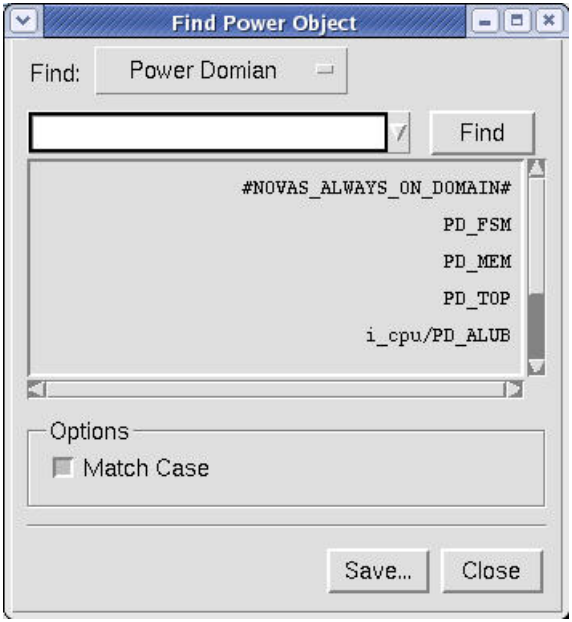


Figure: Find Power Object Form

When the **Match Case** option is selected, only those instances in which the capitalization matches the text typed in the **Find** text field are found.

Click the **Save** button to save search results to a text file.

Report Total Impacted Signals

Menu Bar: Schematic -> Report Total Impacted Signals

NOTE: This option is available in the *Power Map* pane.

This option is enabled when the selected object(s) (or the first selected object if there are multiple selections) are Isolation/Level-shifter cells/nets or non-covered nets. This option opens the *nPowerMap - Report Power Impacted Signals* form where details of the selected impacted power signals are listed. Refer to the *nPowerManager - Report Power Impacted Signals* form invoked through the **Power** -> **Report Impacted Signals by Scope** option for details.

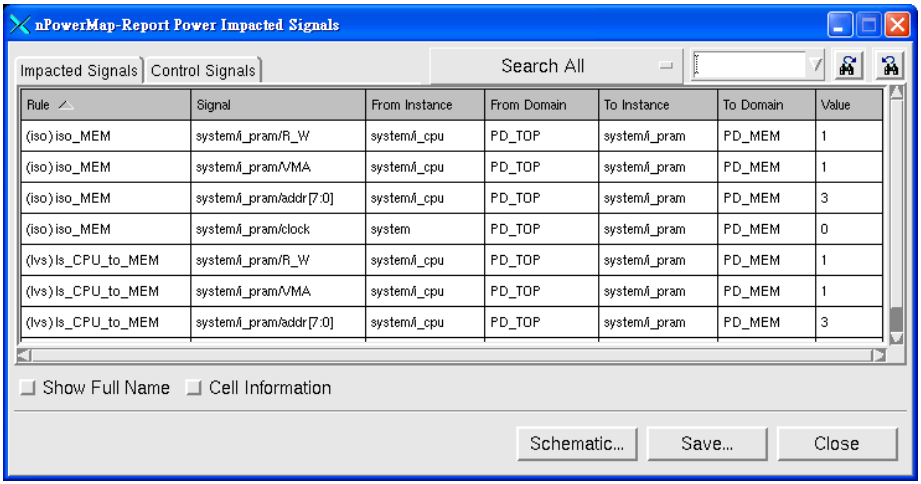


Figure: nPowerMap - Report Power Impacted Signals Form

Report Selected Impacted Signals

Menu Bar: Schematic -> Report Selected Impacted Signals

Bind Key: I (Uppercase I)

This option is enabled when the selected object(s) (or the first selected object if there are multiple selections) are Isolation/Level-shifted cells/nets or non-covered nets. This option opens the *nPowerMap - Report Power Impacted Signals* form which lists the details of the selected impacted power signals.

When selecting an Isolation/Level-shifted rule in the *Power Map* and invoking the **Report Selected Impacted Signals** option, the impacted signals of the selected rule are listed in the *nPowerMap - Report Power Impacted Signals* form.

When selecting non-covered net(s) in the *Power Map* and invoking the **Report Selected Impacted Signals** option, all the non-covered signals belonging to the selected non-covered net are listed in the *nPowerMap - Report Power Impacted Signals* form.

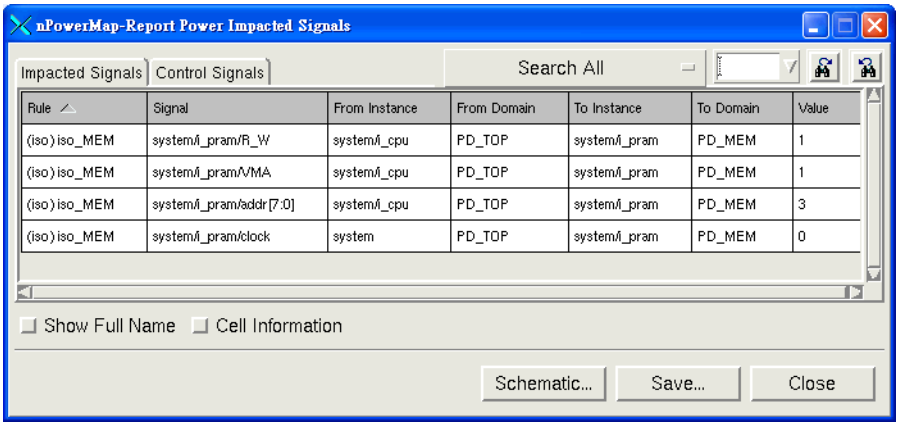


Figure: nPowerMap - Report Power Impacted Signals Form

Refer to the **Power -> Report Impacted Signals by Scope** and **Schematic -> Report Total Impacted Signals** option descriptions for details.

Change Color

Menu Bar: Schematic -> Change Color

Bind Key: C

This option opens the *Change Selection Color* form where the color of the selected power object can be changed.

This form shows the selected power object with its current color in the color map. The color can be changed instantly by clicking another color on the color map. Set the selected signal color as the default by enabling the **Default** option.

All Objects to Default Color

Menu Bar: Schematic -> All Objects to Default Color

This option returns all power objects to their default colors.

Active Annotation

Menu Bar: Schematic -> Active Annotation

Bind Key: X

This toggle option is enabled after a simulation results file is loaded either from *nTrace* or *nWave*. When **Active Annotation** is *on*, the signal values appear in the *Power Map* pane and change simultaneously when the current simulation time changes. Refer to the **Active Annotation** option in the *nTrace* chapter for details.

Tools Options

Open Partial Power Map

Menu Bar: Tools -> Open Partial Power Map

Bind Key: Ctrl+S

This option opens a *Partial Power Map* pane displaying a partial schematic view for the selected power aware objects. Multiple objects may be selected by holding the **Ctrl** key while left-clicking.

Preferences

Menu Bar: Tools -> Preferences

Refer to *PowerMap Folder* description in the *Preferences* chapter for details. The preference settings are saved in the *novas.rc* resource file. The Verdi platform loads the resource file the next time it is invoked.

Customize Menu/Toolbar

Menu Bar: Tools -> Customize Menu/Toolbar

Refer to the **Tools -> Customize Menu/Toolbar** option in the *nTrace* chapter for details.

Power Map Pane and Partial Power Map Pane Right-click Options

When the right mouse button is clicked anywhere in the menu, toolbar icon, or pane banner area of the *Power Map* or *Partial Power Map* panes or their standalone windows, a configuration option menu is displayed. This menu can be used to configure the available icons and panes.

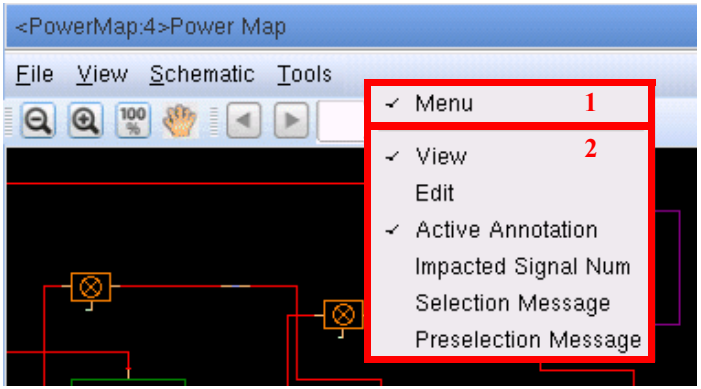


Figure: Configuration Option Menu

The **Menu** option (labeled 1 in the figure above) enables/disables the menu bar. The options in the bottom section (labeled 2 in the figure above) enable/disable the toolbar icons for different functions.

Drag

Bind Key: Ctrl+C

The selected signal can be dragged to another location and dropped. This is similar to using the middle mouse button to select and drag.

Drop

Bind Key: Ctrl+V

The selected signal dragged from another location can be dropped in this window. This is similar to releasing the middle mouse button for a drop.

Report Selected Impacted Signals

Bind Key: I

Refer to the **Report Selected Impacted Signals** option description for details.

Open Partial Power Map

Bind Key: Ctrl+S

Refer to the **Open Partial Power Map** option description for details.

Show Power Network for Selected Domain(s)

Bind Key: N

This option shows the power network for the selected power domains. The power network instances are added to the display and then highlighted.

Power Map and Partial Power Map Panes Mouse Operations

The following table lists mouse actions in the *Power Map* pane:

Mouse Action	Result
Double-click a power cell in the *Power Map* pane.	Find and highlight the corresponding source code of the cell.
Drag & Drop a power cell from the *Power Map* pane to the *Source Code* pane.	Find and highlight the corresponding source code of the cell.

The following table lists mouse action in the *Partial Power Map* pane:

Mouse Action	Result
Double-click (non-covered) Isolation or Level-shifter input/output pins.	Trace power domain driver/load.

Power Map and Partial Power Map Panes Toolbar Icons and Fields

Figure: Toolbar Used in Power Map Pane

Figure: Toolbar Used in Partial Power Map Pane

The available toolbar icons may be modified. Refer to the *Toolbars* section of the *User Interface* chapter in the *Verdi User Guide and Tutorial* manual for details.

The different toolbar categories and available icons are described below:

View Category

Zoom Out

Refer to the **View -> Zoom -> Zoom Out** option description for details.

Zoom In

Refer to the **View -> Zoom -> Zoom In** option description for details.

Zoom All

Refer to the **View -> Zoom -> Zoom All** option description for details.

Pan

Refer to the **Pan** option description for details.

Edit Category

Undo

Refer to the **Edit -> Undo** option description for details.

Redo

Refer to the **Edit -> Redo** option description for details.

Active Annotation Category

Previous Event 

Click this toolbar icon to search backward in time to locate the previous power change for the selected power domain. This icon is active only when an FSDB file is loaded and the **Schematic -> Active Annotation** option is enabled.

Next Event 

Click this toolbar icon to search forward in time to locate the next power change for the selected power domain. This icon is active only when an FSDB file is loaded and the **Schematic -> Active Annotation** option is enabled.

Cursor Time Entry/Display 

This text field displays the cursor time and also sets the cursor time. The field updates all power values at the specified cursor time. This field is active only when an FSDB file is loaded and the **Schematic -> Active Annotation** option is enabled.

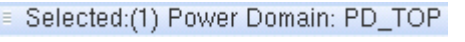
Impacted Signal Num Category 

This field reports the total number of impacted signals for the selected Isolation/ Level-shifter cell/net.

When multiple cells are selected, the impacted signal number shows the number of impacted signals without any duplication. For example, the rule Isolation impacted signal is *a[0:7]* and the rule Level-shifter impacted signal is also *a[0:7]*. When selecting these Isolation and Level-shifter cells, the number shown in this field is 1. The rule Isolation impacted signal is *a[0:4]* and the rule Level-shifter impacted signal is *a[4:7]*. When selecting these Isolation and Level-shifter cells, the number shown in this field is 2.

NOTE: If the impacted signal number of the selected rules exceeds 1000, the number shown in this field is 1000+, instead of the accurate impacted signal number.

Refer to the **Schematic -> Report Selected Impacted Signals** option description for details.

Selection Message 

This field displays the details about the currently selected object.

Preselection Message 

This field displays the details about the object the cursor is over.

Power Aware Toolbar Icons and Fields

Figure: Toolbar Used in nTrace Window with Power Aware Debug

The available toolbar icons may be modified. Refer to the *Toolbars* section of the *User Interface* chapter in the *Verdi User Guide and Tutorial* manual for details.

The different toolbar categories and available icons are described below.

Power Category

Power Map

Refer to the **Power -> Full Power Map** option in the *Power Aware Debug* chapter for details.

List HDL Signals

Refer to the **Power -> List HDL Signals** option in the *Power Aware Debug* chapter for details.

Trace Power

Click the arrow icon to open the **Power Trace** menu which includes four sub-options: **Driver**, **Load**, **Connectivity**, and **Active Trace**. Refer to the **Trace Power** option description for details.

Power Manager Category

Show Power State Table

Refer to the **Power -> Show Power State Table** option in the *Power Aware Debug* chapter for details.

Show Previous Command

Click this toolbar icon to find the previous power option.

Show Next Command

Click this toolbar icon to find the next power option.

Power Domain Information Category

Power Domain

This field is available when a CPF/UPF file is loaded. This field displays the power domain name with the specified highlight color. The color can be redefined by the **Power** -> **Highlight Power Domain** option in *nTrace* or the **Highlight Power Domain** option (default value is *on*) on the **General** -> **Power** page of the *Preferences* form (invoked with **Tools** -> **Preferences**).

Moving the cursor over the **Power Domain** field displays a tip containing the power domain, power state, and shutoff condition.

Power Status Information Category

Power State

This field is available when a CPF/UPF file is loaded. This field displays the power state information.

Testbench Debug

Overview

Testbench Debug support is integrated with the main *nTrace* window. The panes associated with testbench debug are available when SystemVerilog testbench code is loaded into the Verdi platform. The design and testbench code can be browsed and interactive debugging can be performed. All declared modules, classes, interfaces, and programs are sorted and displayed in a tree view for easy browsing.

The various testbench functions are docked to existing pane locations. By default, the declaration tree pane is docked to the same location as the design browser pane as a new tab, the constraint, inheritance, and FSDB message panes are docked to the middle pane (the same as the signal list) and the testbench code is incorporated as part of the source code pane. Refer to the *Icons for Dockable Panes* section for details on the menu bar icons.

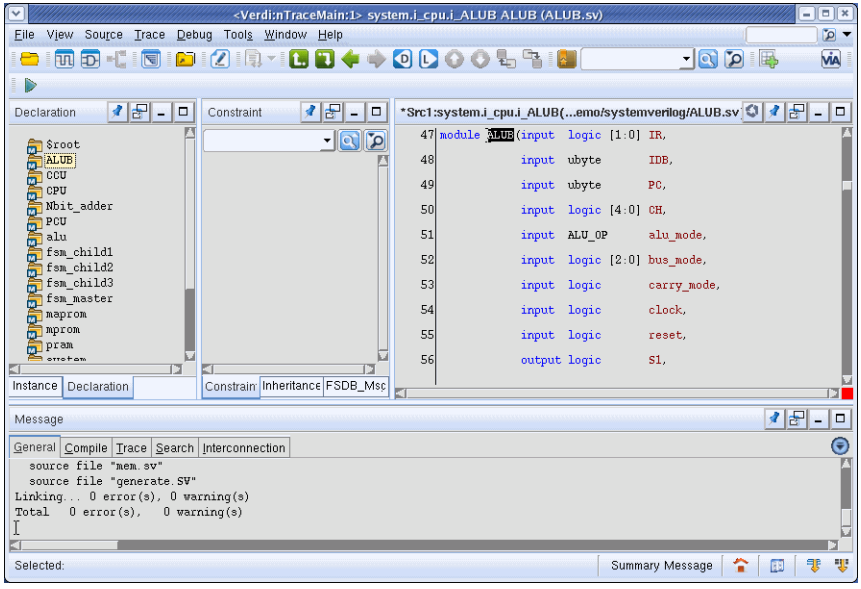


Figure: Default View for Testbench Browsing

The panes associated with Testbench Debug support are summarized and explained on the following pages.

Declaration Tree Pane

The *Declaration Tree* pane is docked to the same pane location as the design browser pane as a new **Declaration** tab. The declaration tree provides a module/class based view for the design and testbench. All of the declared modules, entities, classes, tasks, functions, packages, interfaces, and programs are sorted and displayed in a tree view.

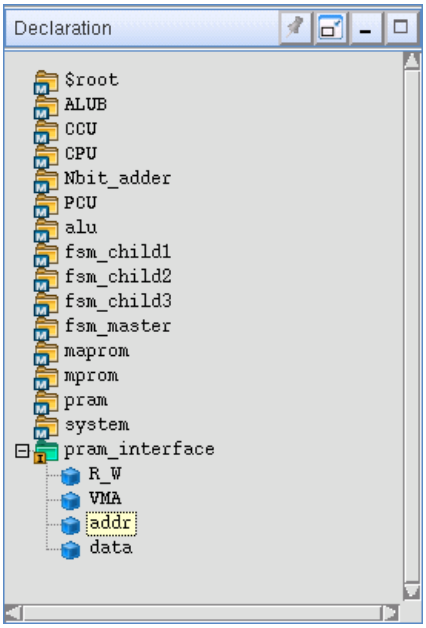


Figure: Declaration Tree Pane

Declaration Tree Pane Right-click Options

The following right-click menu options are available for the declaration tree pane. Some selections are available from the right mouse button menu. Other selections have equivalent selections available from the toolbar menus.

Show Navigation Text Field

Refer to the **Show Search/Filter** option description in the *nTrace* chapter for details.

Show File Information

Refer to the **Show File Information** option description in the *nTrace* chapter for details.

Show Instantiation

This option identifies the instantiation location(s) for the selected object and summarizes the results on the **Interconnection** tab of the *Message* pane.

Expand Tree by Level

Refer to the **View -> Hierarchy Tree by Level** option description in the *nTrace* chapter for details.

Collapse All

This option collapses all expanded tree nodes in the declaration tree pane.

Display Source Code in New Tab

Refer to the **View -> Source Tab -> New Source Tab** option description in the *nTrace* chapter for details.

Constraint Pane

The *Constraint* pane is docked to the middle pane as another tab. This is the same pane location as the *Inheritance* and *FSDB_Msg* panes for testbench code and the *Signal List* pane for design code. The *Constraint* pane displays all classes that contain constraints in a tree view. Under each class node, the constraints defined in the class are listed.

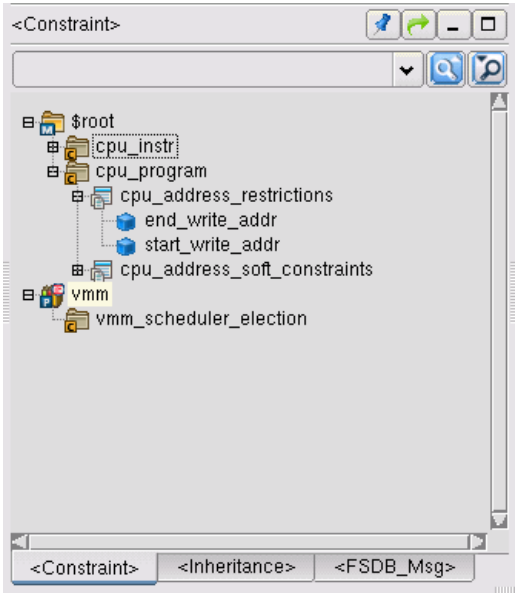


Figure: Constraint Pane

Double-clicking any constraint or variable node automatically displays the associated code in the source code pane. The search text field can be used to search for constraints.

The *Constraint* pane can be displayed or hidden by enabling/disabling the **View -> Constraint View** option.

Constraint Pane Right-click Options

The following right-click menu option is available for the constraint view pane.

Show Constraint Source Code

This option displays the source code for the selected constraint as a **Constraint Source Code** tab in the source code pane. Multiple constraints can be selected

by clicking the left mouse button on a constraint and then holding the Ctrl key while left-clicking another constraint.

Constraint Pane Toolbar Icons and Fields

The available toolbar icons may be modified. Refer to the *Toolbars* section of the *User Interface* chapter in the *Verdi User Guide and Tutorial* manual for details.

Search 

Specify the constraint to search for in this text field.

Previous Node 

Click this icon to find the previously matched string.

Next Node 

Click this icon to find the next matched string.

Inheritance Pane

The *Inheritance* pane is docked to the middle pane as another tab. This is the same pane location as the *Constraint* and *FSDB_Msg* panes for testbench code and the *Signal List* pane for design code. The *Inheritance* pane displays the inheritance relationship for the selected class. If a class node (object initiated by module or program block) is clicked in the **Declaration** or **Instance** tabs, the inheritance hierarchy of the class is shown in the *Inheritance* pane.

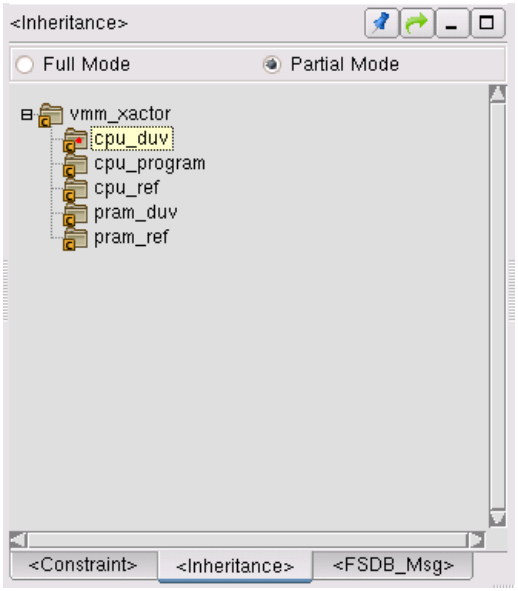


Figure: Inheritance Pane - Partial Mode

Two display modes are available in the *Inheritance* pane: **Full Mode** and **Partial Mode**. When **Full Mode** is selected, all classes from which the selected class extends are displayed. When **Partial Mode** is selected, the sibling classes that are in the **ovm_pkg** package with the prefix "ovm_", "avm_", or "vmm_" are filtered and not displayed. The default is **Partial Mode**.

The *Inheritance* pane can be displayed or hidden by enabling/disabling the **View -> Inheritance View** option.

FSDB Message Pane

The FSDB message (*FSDB_Msg*) pane is docked to the middle pane as another tab. This is the same pane location as the *Constraint* and *Inheritance* panes for testbench code and the *Signal List* pane for design code. The *FSDB_Msg* pane is initially empty until a signal with message information is added (by dragging the transaction/message from the *nWave* or the *Transaction/Message Analyzer* panes and dropping in the source code or *FSDB_Msg* panes). The logged message information, including stream name, label name, time, and all call stacks are shown on the *FSDB_Msg* pane.

Message streams recorded using the *$fsdbLog* dumping option and transactions automatically recorded from the OVM/UVM library with call stack logging enabled are supported.

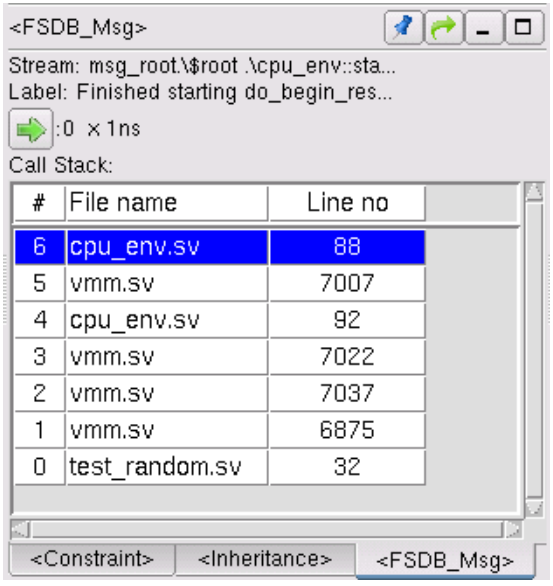


Figure: FSDB Message Pane with Message Displayed

Double-clicking on a specific call stack displays the corresponding source code in the source code pane. Selecting a specific call stack and then invoking the right-click option **Show Source Code** shows the corresponding source code in the source code pane.

The *FSDB_Msg* pane can be displayed or hidden by enabling/disabling the **View -> FSDB Message View** option.

FSDB Message Pane Right-click Options

The following right-click menu options are available for the FSDB message pane. Some selections are available from the right mouse button menu. Other selections have equivalent selections available from the toolbar menus.

Show Source Code

This option shows the corresponding source code of the selected call stack in the source code pane.

Set Breakpoint

This option adds a breakpoint for the selected thread into the source code pane. This option is disabled if a breakpoint was set previously on the same thread.

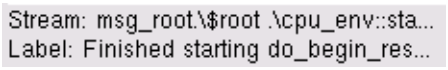
Manage Breakpoints

Refer to the **Debug -> Manage Breakpoints** option description for details.

FSDB Message Pane Toolbar Icons and Fields

The available toolbar icons may be modified. Refer to the *Toolbars* section of the *User Interface* chapter in the *Verdi User Guide and Tutorial* manual for details.

Stream/Label



These fields display the stream name and selected label.

Simulate and break at specified message 

Click this icon to start the simulation and break the simulation at the selected message.

Source Code Pane

The options summarized here are located on the menu bar of the main *nTrace* window and are associated with testbench code.

Source Options

Show -> Reference

Menu Bar: Source -> Show -> Reference

NOTE: This option is only available for SystemVerilog.

This option displays the source code references for the selected signal or variable on the **Interconnection** tab of the *Message* pane. Click the desired result in the *Message* pane to display the corresponding source code.

The reference types can be filtered by clicking the arrow to the right of the **Show Reference** icon. After changing the type selection on the toolbar, the filter is applied to all reference options (toolbar icon, right-click option, and main menu). Available types are: **Read**, **Write**, **Pass**, **Define**, and **Call**. By default, all types are enabled. The type of reference (R, W, P, D, or C) is also indicated at the start of each line in the *Message* pane.

The types are defined as follows:

- Assignment (for example, `"r=1'b1"`, `continue_assign`, `<=`, `force L=R`, `lhs=[#delay] rhs`, `L=new(R1,R2)`)
  - LHS -> **Write**
  - RHS -> **Read**
- Function/Task/instance {`inst1.func(arg1,arg2)`, `inst(.port1(net1))`}
  - In funcName(inst1) -> **Call**
  - In argument -> **Pass** (not including expression, for example, `"func(a+b)"`, a and b are **Read**)
- Definition (for example, A `ia`; `expr inside {...}`, `dist`) -> **Define**
- unary op (for example, `++a`, `a--`) -> **Write**

NOTE: The types only work on variables, signals, and instances. Class types and function types are not supported.

Trace Options

Interface Mapping

Menu Bar: Trace -> Interface Mapping

This option extracts the hardware and testbench interface mapping results for the imported design and summarizes the results on the **Interconnection** tab of the *Message* pane.

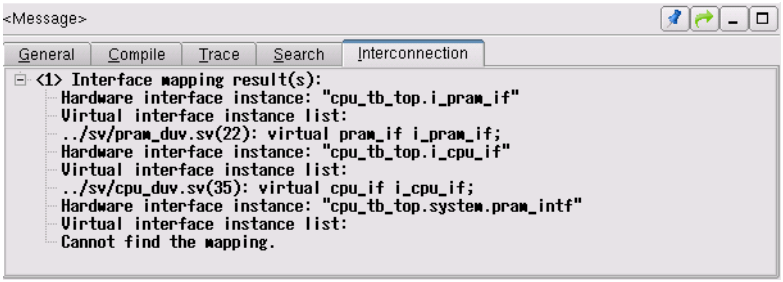


Figure: Interconnection Tab - Interface Mapping Results

For each hardware interface, the hardware interface result is listed on one line and the connected virtual interface results are listed on the following lines. Double-click any virtual interface results to display the declaration of the virtual interface in the source code pane.

From HDL to HVL Interface

After selecting a hardware interface or interface members in the HDL code, this option locates and displays the line where the signal is used in a virtual interface or where the virtual interface is declared. The results are also summarized on the **Interconnection** tab of the *Message* pane.

From HVL to HDL Interface

After selecting a virtual interface or virtual interface members in the HDL code, this option locates and displays the line where the signal/object is declared in the hardware interface. The results are also summarized on the **Interconnection** tab of the *Message* pane.

Source Code Pane Right-click Options

The following right-click menu options are available for the source code pane when testbench code is displayed. Some selections are available from the right mouse button menu. Other selections have equivalent selections available from the toolbar menus.

Show Declaration

This option identifies the declaration location(s) for the selected class or class member reference.

Show Definition

This option identifies the location where the selected object is defined.

Show Reference

Refer to the **Source -> Show -> Reference** option description for details.

From HDL to HVL Interface

Refer to the **Trace -> From HDL to HVL Interface** option description for details

From HVL to HDL Interface

Refer to the **Trace -> From HVL to HDL Interface** option description for details.

Set as Active Scope

This option sets the current scope as the active scope.

Drag/Drop

Refer to the **Drag/Drop** option descriptions in the *nTrace* chapter for details.

Copy Instance's/Signal's Full Path

Refer to the **Copy Full Path** option description in the *nTrace* chapter for details.

Interactive Simulation Debug

Overview

Interactive Simulation Debug support is enabled when the **Simulation** -> **Run Simulation** menu option or the  toolbar icon is selected. The various Interactive Simulation Debug functions are docked to existing pane locations in the main *nTrace* framework. By default, the *Interactive Debug* panes are docked to the same pane locations as the design browser (left pane) as the **Class**, **Object**, and **Stack** tabs and the signal list (middle pane) as **Member** and **Local** tabs. In addition, the **Interactive_Console** tab (simulation control) is docked to the same pane location as the **Message** tab as a new tab and the *Watch* pane with one or more **Watch** tabs is docked to the right of the **Interactive_Console** tab as a new pane.

NOTE: The default layout for the *Interactive Debug* panes invoked by the **Simulation** -> **Run Simulation** menu option is the same as the **Window -> Interactive Debug Mode** option.

NOTE: Before using SVTB Interactive Simulation Debug, make sure that VCS 2014.03 or a later version is used.

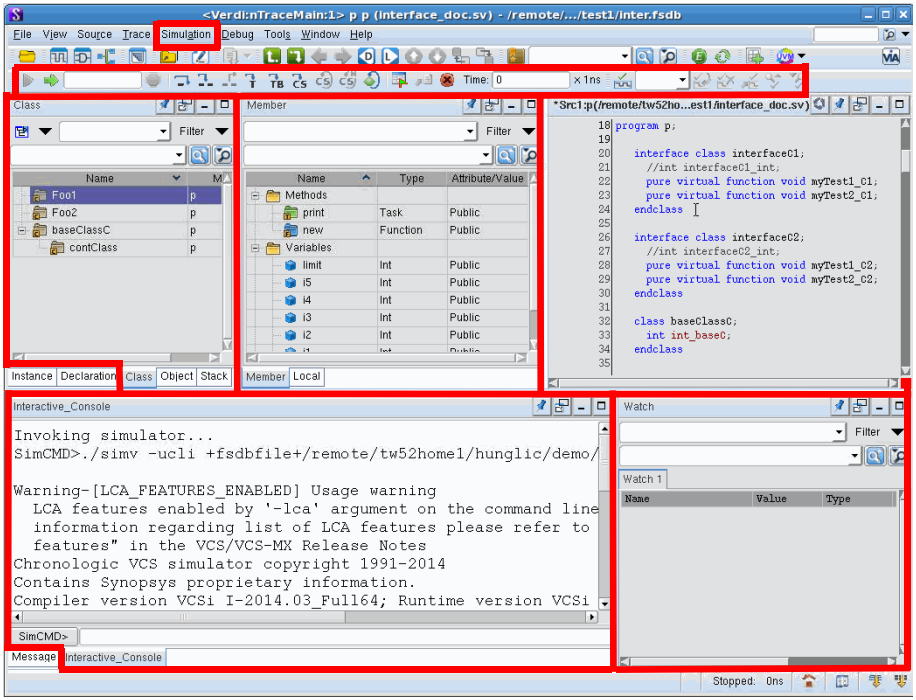


Figure: Default View for Interactive Debug

The pull-down menus, toolbar icons, and panes associated with Interactive Simulation Debug support are summarized and explained on the following pages.

Source Option

Active Annotation

Menu Bar: Source -> Active Annotation

Bind Key: X

This option displays annotation values of the variables on the source code pane. When a macro or protect code is in the current active scope and is expanded, the annotation values are also displayed. The following table summarizes the display format for different data types:

Data Type	Example	Display Format
Associate array	logic [63:0] assoc[*], idx = 1; repeat(64) begin assoc[idx] = idx; idx = idx << 1; end	Size(64)
Dynamic array	Class Foo; int array[]; … bar.array = new[limit];	Dynamic array is empty: empty Dynamic array is not empty: size(2)
Event	Event gen_done;	Event is null: null Event is triggered: triggered
Integer	Int limit = 'h10;	'h10
Integer array	int ascend[4] = '{0,1,2,3}	size(4)
Logic	bit one_bit = 1;	'b1
Logic array	bit [3:0][7:0] bytes	size(4)
Mailbox	mailbox s_mbx; s_mbx = new;	Mailbox is null: null Mailbox is not null: @1
Multiple dimension array (packed arrays)	Interface arb_if(input bit clk) Logic [1:0][2:0] grant; arbif.grant[0] <= 2'b00;	Whole values of the MDA
Multiple dimension array (unpacked arrays)	Interface arb_if(input bit clk) Logic [1:0][2:0]; arbif.grant[0] <= 2'b00;	size(2)
Object	Foo1 bar1;	Object id has value: @1 Object id is null: null

Queue	Int q[$] = {0,2,5}; Int j = 1; q.insert(1,j);	size(n)
Semaphore	semaphore sm; sm = new;	Semaphore is null: null Semaphore is not null: @1
String	String s1 = "ABC"	"ABC"("" for empty string)
String array	String s2[2]	size(2)
String queue	typedef string string_queue_t [$]; string_queue_t q={1,2}	size(2)
Struct	Struct { Bit [0:3] A; Bit[0:3] B; }abc	struct

If the variables belong to a testbench scope, the values of the variables are obtained from the simulator. When the variables have been executed, if the variables belong to the active pane or belong to the member variables of the current object, the values of the variables are displayed in the source code pane. The radix of the variable value can be changed by right-clicking the **Set Radix** option.

If variables belong to an HDL scope, the values of the variables are obtained from the active FSDB file first and then the simulator. The annotation values are based on the cursor time in the active *nWave* window. The previous values can be shown when an FSDB file is loaded by moving the cursor in the nWave window as follows:

- If the cursor time in the *nWave* window is equal to the current simulation time, the values of the variables are obtained from the FSDB file first and then the simulator.
- If the cursor time in the *nWave* window is not equal to the current simulation time, the values of the variables are obtained from the active FSDB file (that is, the variables need to be dumped first, and then the values can be annotated to the variables on the source code pane).

The radix of the variable value can be changed by right-clicking the **Signal -> Signal Value Radix** option.

Simulation Options

Simulation menu options control the simulation progress. After the **Tools -> Run Simulation** option is invoked, the following options appears and the Interactive Simulation Debug toolbar is displayed.

Restart

Menu Bar:	Simulation -> Restart
Toolbar Icon:	
Bind Key:	Ctrl+F5

This option restarts the simulation run. After invoking this option, the following actions are also restarted:

- Set breakpoints.
- Dumped and added signals in the *nWave* window.
  The signals previously dumped into the inter.fsdb file is dumped automatically after restarting the simulation.
- Issued force options for HDL signals.

NOTE: Only the issued options for HDL signals are applied. The force options from the script files or the **Interactive_Console** tab are ignored.

Attach to Simulation

Menu Bar: Tools -> Attach to Simulation

NOTE: The required settings for this feature are as follows:
1. The simulation must start with the **-ucli2Proc** VCS runtime option.
2. The **-debug_access/-debug_pp/-debug/-debug_all** VCS elaboration options must be added to enable the VCS debug features.
3. The design must be loaded in Verdi before attaching a simulation.

NOTE: If the simulation is running in GUI mode (DVE or Verdi) or specman, the process cannot be attached.

NOTE: The **Restart** and **Restore Session** options are not supported in Verdi after a simulation is attached.

This option opens the *Attach to Simulation* form where all running simulation processes are listed. After a simulation process is selected (only one simulation process can be attached at one time), click the **Attach** button and then in the UCLI terminal, press the **Enter** key on the keyboard to accept the attached simulation.

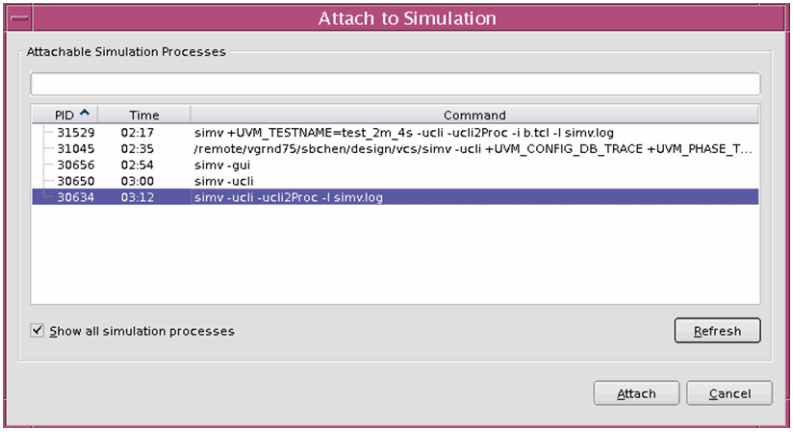


Figure: Attach to Simulation Form

The *Attach to Simulation* form includes the following field and options:

- **Attachable Simulation Process** text field: Type text in the text field to filter simulation process. The wildcard characters are supported.
- **Show all simulation processes:** When this option is turned *on*, all running processes are displayed. The default is *off*.

- **Refresh:** Click this button to refresh processes in the current machine.

Detach Simulation

Menu Bar: Tools -> Detach Simulation

This option opens the *Attach to Simulation* form where the attached process is marked with the icon. Select the connected process and click the **Detach** button to detach the simulation.

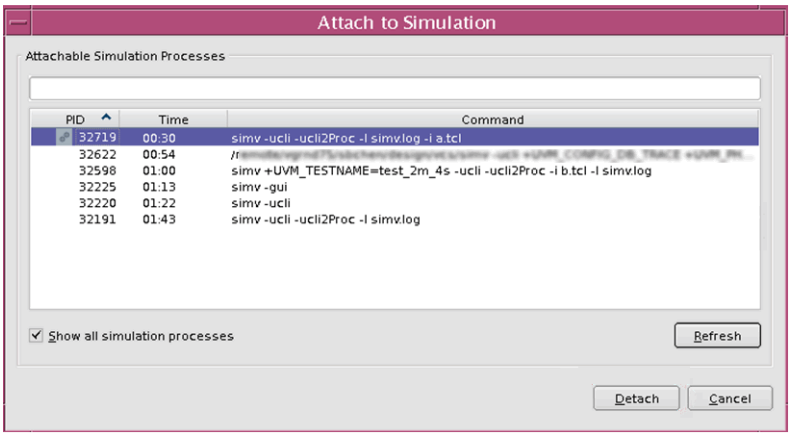


Figure: Attach to Simulation Form for Detaching

Rebuild and Restart

Menu Bar: Simulation -> Rebuild and Restart

This option opens the *Rebuild and Restart* form where the configurations of rebuild and restart features can be specified. After setting the configurations, click the **OK** button and Verdi starts to rebuild the KDB and the simv simulator executable files and load the KDB.

After the simulation is rebuilt and restarted, the GUI layouts and the following data in Interactive Simulation Debug mode is restored:

- The objects shown in the **Watch** tab.
- The objects dumped in the FSDB file.
- The breakpoint options, force and release options, and signals shown in the *nWave* window.

This feature is controlled in the **Preferences -> Interactive Debug -> Restart** page. Refer to the **Restart Page** for details.

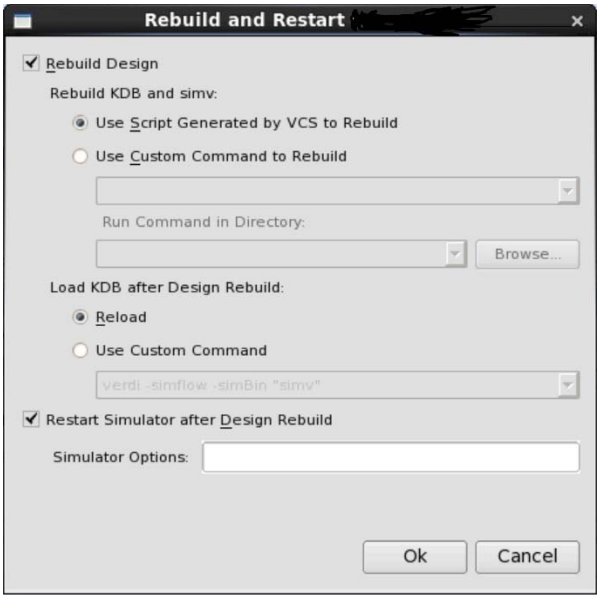


Figure: Rebuild and Restart Form

The *Rebuild and Restart* form includes the following options and fields:

- **Rebuild KDB and simv:** Specify one of the following methods to rebuild the KDB and the simv simulator executable files. The default is **Use Custom Command to Rebuild**.
  - **Use Script Generated by VCS to Rebuild:** When this option is turned *on*, commands recorded in the vcs_rebuild rebuild script file are used to rebuild the KDB and the simv simulator executable files. VCS automatically generates the vcs_rebuild rebuild script (located in the *simv.daidir/vcs_rebuild* directory) when VCS is invoked each time.
  - **Use Custom Command to Rebuild:** When this option is turned *on*, type the rebuild command in the field to rebuild the KDB and the simv simulator executable files in the current working directory. To run the rebuild commands in another directory, specify the path in the **Run Command in Directory** field.
    For a custom script file, use the **Use Custom Command to Rebuild**

option and the script file to perform the rebuild. It is recommended to include only the compile and elaborate commands in the script file.

NOTE: The **Use Script Generated by VCS to Rebuild** method is recommended for Verilog-only single-VCS compilation flows. It does not support UUM, MX, slave mode, or multi-step builds. For complex compilation script, UUM, MX or multi-step builds, use the **Use Custom Command to Rebuild** method.

- **Load KDB after Design Rebuild:** Specify one of the following methods to load the KDB after the design is rebuilt. The default is **Reload**.
  - **Reload:** When this option is turned *on*, the options previously used to load the KDB are used to reload the KDB.
  - **Load KDB from 'simv.kdb' Generated by VCS:** When this option is turned *on* and if the KDB file is generated by the Unified Compiler front-end flow, the KDB is loaded with the options generated in the simv.kdb file.
  - **Use Custom Command:** When this option is turned *on*, custom commands or a script file are used to load the KDB.
    If files are added from the design, new import commands or a new script file should be used to import these files.
- **Restart Simulator after Design Rebuild:** When the option is turned *on*, Verdi restarts the simulation in Interactive Simulation Debug mode after the design is rebuilt and loaded. The default is *on*.
  Enter the specified simulator options in the **Simulator Options** field. The simulator options used when invoking the simulator are listed in this field by default.

Run/Continue

Menu Bar: Simulation -> Run/Continue

Toolbar Icon: 

Bind Key: F5

This option starts the simulator. If the simulator is invoked and stopped at a preset breakpoint (a flag and arrow combination  indicates the current breakpoint and simulator position), invoke this option again to continue the simulation run.

NOTE: All VCS simulator options are supported and can be executed in the **Interactive_Console** tab at the bottom of the main *nTrace* window by typing the option and arguments and pressing **Enter**. The simulator output is displayed in the **Interactive_Console** tab. Refer to the VCS simulator document for details on the option list and usage.

Step/Next

The **Step/Next** option includes the following sub-options: **Step**, **Next**, **Step in Thread**, **Next in Thread**, **Step in Testbench**, **Step Out**, **Step in Constraint Solver**, **Redo Randomize Call and Step in Constraint Solver**, **Do Distribution Re-randomize and Step in Constraint Solver**, **and Next in C/C++**.

Step

Menu Bar: Simulation -> Step

Toolbar Icon: 

Bind Key: F11

This option stops at the statement where a function is called in the source code pane. The **Local** and **Stack** tabs are refreshed accordingly.

Refer to the *Interactive Debug Panes* section for details on these tabs.

Next

Menu Bar: Simulation -> Next

Toolbar Icon: 

Bind Key: Ctrl+F10

This option jumps to the next executed statement. A solid blue arrow  in the source code pane indicates the current simulator position (clicking the **Jump to Current Debug Position**  icon in the toolbar also jumps to the current position).

Step in Thread

Menu Bar: Simulation -> Step in Thread

Toolbar Icons: 

Bind Key: F12

After a thread is selected in the **Stack** tab, the simulator jumps to a new location as follows:

- If the simulator is not in the specified thread, this option changes to the specified thread.
- If the simulator is in the specified thread, this option changes to the next executed statement in the specified thread. This option can step into tasks and functions.

If a thread is not selected, this option stops at the current thread.

Next in Thread

Menu Bar: Simulation -> Next in Thread

Toolbar Icons: 

Bind Key: Ctrl+F9

After a thread is selected in the **Stack** tab, the simulator jumps to a new location as follows:

- If the simulator is not in the specified thread, this option changes to the specified thread.
- If the simulator is in the specified thread, this option changes to the next executed statement in the specified thread. This option steps over tasks and functions.

If a thread is not selected, this option stops at the current thread.

Step in Testbench

Menu Bar:	Simulation -> Step in Testbench
Toolbar Icon:	
Bind Key:	Ctrl+F11

NOTE: This option is only for testbench code.

For SystemVerilog testbench, this option stops at the next executable line in the testbench.

Step Out

Menu Bar:	Simulation -> Step Out
Toolbar Icon:	
Bind Key:	Ctrl+F12

This option steps out to the previous stack in the source code pane.

Step in Constraint Solver

Menu Bar:	Simulation -> Step in Constraint Solver
Toolbar Icon:	

When the simulation stops before a randomize call (see **Constraint Tab** for the settings), this option steps in the constraint solver engine and the simulation enters Constraint Debug mode.

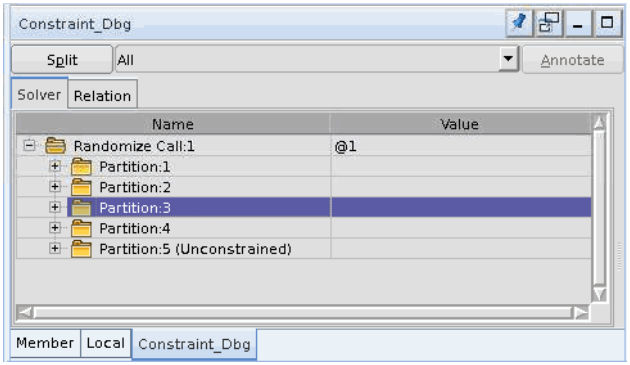


Figure: Constraint Debug with Solver and Relation Subtabs

Details for the solved results are shown in the **Constraint Debug** tab. Refer to the **Solver** and **Relation** subtabs in **Constraint Debug Tab** for details.

Redo Randomize Call and Step in Constraint Solver

Menu Bar:	Simulation -> Redo Randomize Call and Step in Constraint Solver
Toolbar Icon:	

When the simulation is already in Constraint Debug mode, this option rerandomizes and steps in the constraint solver engine. After the rerandomization is complete, the simulation enters Constraint Debug mode and stop before the post-randomized call. In addition, another set of **Solver** and **Relation** subtabs with the **Solver_R** and **Relation_R** tab names are added in the **Constraint Debug** tab to display the rerandomized solved results.

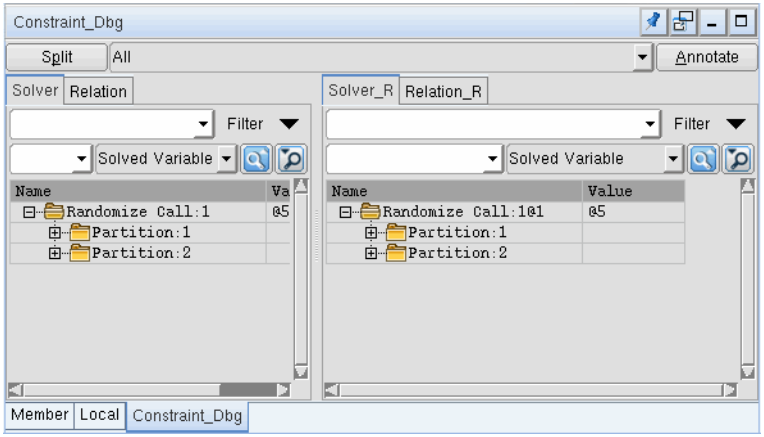


Figure: Constraint Debug with Solver/Solver_R and Relation/Relation_R Subtabs

The **Solver** and **Relation** subtabs show the original value after the randomize call. The **Solver_R** and **Relation_R** subtab show the new value after rerandomize call.

Click the **Annotate** button in the upper right corner to annotate the solved value from the original randomize tree to the rerandomize tree. The new **Original** column is added and value differences are highlighted in red.

To view desired types in the **Constraint Debug** tab, select **Original**, **Rerand** or **All** from the selection field to show the original **Solver/Relation** subtabs, the **Solver_R/Relation_R** subtabs or both the **Solver/Relation** and **Solver_R/ Relation_R** subtabs respectively.

Do Distribution Re-randomize and Step in Constraint Solver

Menu Bar:	Simulation -> Do Distribution Re-randomize and Step in Constraint Solver
Toolbar Icon:	

NOTE: The **Distribution Debug** toolbar icon is enabled when the simulation is already in Constraint Debug mode, same as the normal rerandomize.

This option opens the *Distribution Rerandomize* form to perform multiple re-randomizations.

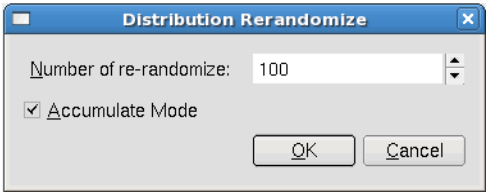


Figure: Distribution Rerandomize Form

The *Distribution Rerandomize* form includes the following field and option:

- **Number of re-randomized:** Specify a number of rerandomizations in this field. A valid number is from 2 to 99999 and the default is **100**.
- **Accumulate Mode:** When this option is turned *on*, the results of the previous n distribution randomizations are accumulated for the current randomize call. The default is *on*.

After clicking the **OK** button, Verdi performs randomization for the specified number of times, and the simulation is stopped in the constraint solver engine. Another set of **Distribution** and **Histogram** subtabs with the **Dist_R** and **Histogram_R** tab names are added in the *Constraint Debug* tab to display the distribution of rerandomized analysis. Refer to **Distribution_R Subtab** and **Histogram_R Subtab** for details.

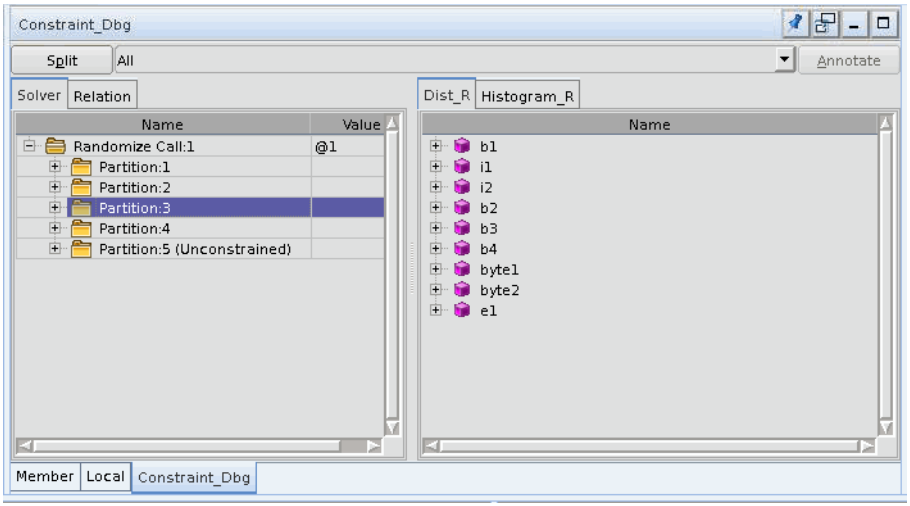


Figure: Constraint Debug with Distribution_R and Histogram_R Subtabs

Next in C/C++

Menu Bar: Simulation -> Next in C/C++

Toolbar Icon: 

Stop

Menu Bar: Simulation -> Stop

Toolbar Icon: 

This option stops the simulator.

Quit

Menu Bar: Simulation -> Quit

Toolbar Icon: 

This option terminates the simulator. The Interactive Simulation Debug toolbar and **Simulation** menu options are removed. A message indicating the simulator process has been completed is printed to the **General** tab of the *Message* pane.

Kill Simulator Process

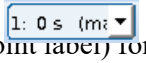
Menu Bar: Simulation -> Kill Simulator Process

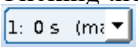
This option kills the halted simulator process. The Interactive Simulation Debug toolbar and **Simulation** menu options are removed. A message indicating the simulator process has been completed is printed to the **General** tab of the *Message* pane.

Add Checkpoint

Menu Bar: Simulation -> Add Checkpoint

Toolbar Icon: 

This option adds a checkpoint that records the data based on the current simulation time by clicking the  icon and the checkpoint is added in the **Checkpoint List**  field with the *index: simulation time* (user-specified checkpoint label) format. Multiple checkpoints can be added at the same time.

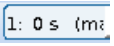
Clicking the upside-down triangle on the right of the **Checkpoint List**  field to specify the checkpoint label. The following three settings for checkpoint labels are provided. The default checkpoint label is the active simulation scope.

- **Use Scope as Label:** When this option is selected, the active simulation scope is added automatically as the label for checkpoints.
- **Specify Label:** When this option is selected and then the  icon is clicked, the *Add Checkpoint* form is opened to specify the label for the new checkpoint.
- **No Label:** When this option is selected, labels are added for checkpoints.

Rewind to Checkpoint

Menu Bar: Simulation -> Rewind to Checkpoint

Toolbar Icon: 

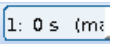
After a checkpoint is selected in the **Checkpoint List**  field, this option rewinds the simulation to the selected checkpoint from the current simulation time. When the simulation is rewound to a previous time, all checkpoints after the selected checkpoint are removed.

The data shown in all Interactive-related panes are updated accordingly. In the *nWave* window with loaded *inter.fsdb* file, the FSDB file is read-only now. A new waveform cursor highlighted in dashed blue line can be viewed. So is the 'future' waveforms which are later than the current simulation time.

Delete Checkpoint

Menu Bar: Simulation -> Delete Checkpoint

Toolbar Icon: 

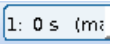
After a checkpoint is selected in the **Checkpoint List** 

field, this option deletes the selected checkpoint.

Go to Source Position

Menu Bar: Simulation -> Go to Source Position

Toolbar Icon: 

After a checkpoint is selected in the **Checkpoint List** 

field, this option goes to the source code where the checkpoint was set during Interactive Simulation Debug mode. The source code pane indicates the line of source code that the checkpoint has been set to.

Command History

The **Command History** option includes the following sub-options: **Undo** and **Redo**.

Undo

Menu Bar: Simulation -> Command History -> Undo

Toolbar Icon: 

This option cancels the effect of the most recent simulation execution option(s) (for example, **Run**, **Next**, or **Step**, and so on). For example, assuming the **Run** option was executed and the simulation stopped at a certain breakpoint. The **Undo** option can be used to go back to the simulation state, as it was before the **Run** option.

The **Undo** option is not fully supported in following situations:

- When execution is in VHDL code
- When simulation is stopped inside certain breakpoints (for example, constraint solver or assertion breakpoint). In this case, the simulation goes

to the most recently executed source code line before the desired location. The **Undo** option cannot go back to the desired execution location exactly.

To ensure the **Undo** option work in the above situations, enable automatic checkpoint creation at each simulation stop with the following step:

1. Select the **Interactive Debug -> Reverse Debug** page from the *Preferences* form opened by the **Tools -> Preferences** option.
2. Enable the *Provides Checkpoints for Last ___ Simulation Stops* option. The default is *5*. This option can undo five most recent simulation control options.

Redo

Menu Bar: Simulation -> Command History -> Redo

Toolbar Icon: 

The **Redo** option can redo the corresponding option.

Go to Value Assignment

The **Go to Value Assignment** option includes the following sub-options: **Previous** and **Next**.

Previous

Menu Bar: Simulation -> Go to Value Assignment -> Previous

Toolbar Icon: 

This option goes to the previous assignment for a signal or variable for the selected object in the Watch, Local, or source code panes to reverse the simulation to the assignment point.

Next

Menu Bar: Simulation -> Go to Value Assignment -> Next

Toolbar Icon: 

This option goes to the next assignment for the selected signal or variable which advances the simulation to the assignment point.

Discard Future

Menu Bar: Simulation -> Discard Future

The **Discard Future** option allows you to destroy future simulator status (including FSDB truncation).

Following is the syntax to use the **Discard Future** option from the *Interactive Console*:

```
srcTBDiscardFuture
```

Set Window Time Unit

Menu Bar: Simulation -> Set Window Time Unit

Toolbar Icon: 

This option opens the *Set Window Time Unit* form where the time scale and time unit display is specified.

After the time scale is changed, the time scale in the primary waveform window, Interactive Simulation Debug toolbar, other Interactive Simulation Debug components, and message bar is changed accordingly.

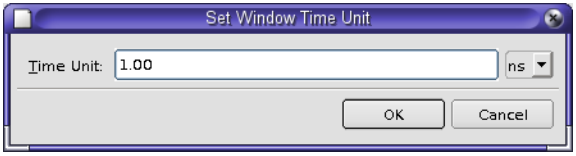


Figure: Set Window Time Unit Form

The *Set Window Time Unit* form includes the following fields and options:

- **Time Scale:** Enter a time in the text field to set the time scale.
- **Time Unit:** Select one of the options from **fs**, **ps**, **ns**, **us**, **ms**, and **s** to set the time unit. The default time scale shown on the toolbar and the *Set Window Time Unit* form is based on the setting provided in the **Miscellaneous** page (invoked using **Tools -> Preferences -> General** folder).

Go to Active File/Line

Menu Bar: Simulation -> Go to Active File/Line

Toolbar Icon: 

This option goes to the next active source code file or line in the source code pane.

Save State

Menu Bar: Simulation -> Save State

This option opens the *Save State* form where the file name for saving the current simulation status (including the time scale) and the dumped FSDB file name can be specified. By default, the file is saved with a **.bin* file extension.

Restore State

Menu Bar: Simulation -> Restore State

This option opens the *Restore State* form where the file with the saved simulation status can be selected for restoration. The FSDB file is also restored and the dumped signals continue to be dumped to the FSDB file.

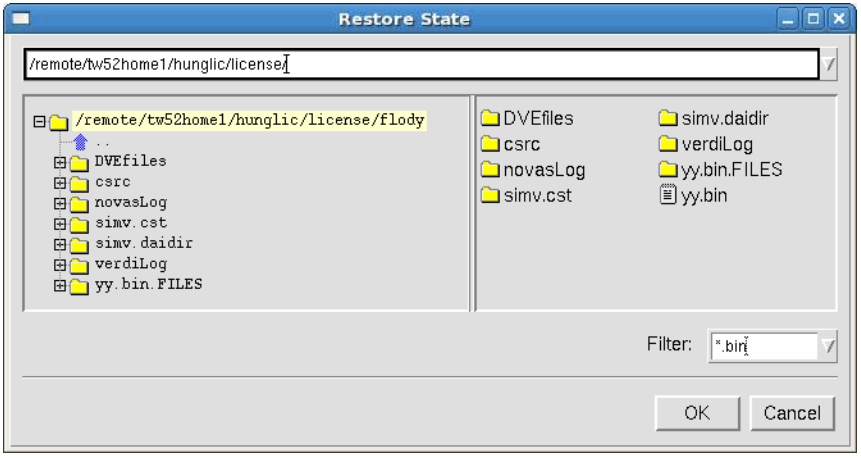


Figure: Restore State Form

Manage Breakpoints

Menu Bar: Simulation -> Manage Breakpoints

Toolbar Icon: 

NOTE: This option is available in Interactive Simulation Debug mode.

This option opens the *Manage Breakpoints* form where breakpoints can be edited.

The **Manage Breakpoint** list includes the **Enable**, **ID**, **Type**, and **Breakpoint** columns. The width of each column can be adjusted by placing the cursor over the vertical bar in the heading row and then pressing and holding the left mouse button. The column headings are summarized as follows:

- **Enable:** Indicates the status for each of the added breakpoints. When the check box is selected, the breakpoint is enabled and a green flag appears next to the line number. When the check box is not selected, the breakpoint is disabled and an empty flag appears next to the line number. The **Enable** check box is selected by default.
- **ID:** Indicates the ID number for each of the added breakpoints. The number is assigned in numeric order, but a deleted number is not added back.
- **Type:** Indicates the breakpoint type for each of the added breakpoints.
- **Breakpoint:** Indicates the UCLI commands for the breakpoint settings for each of the added breakpoints.

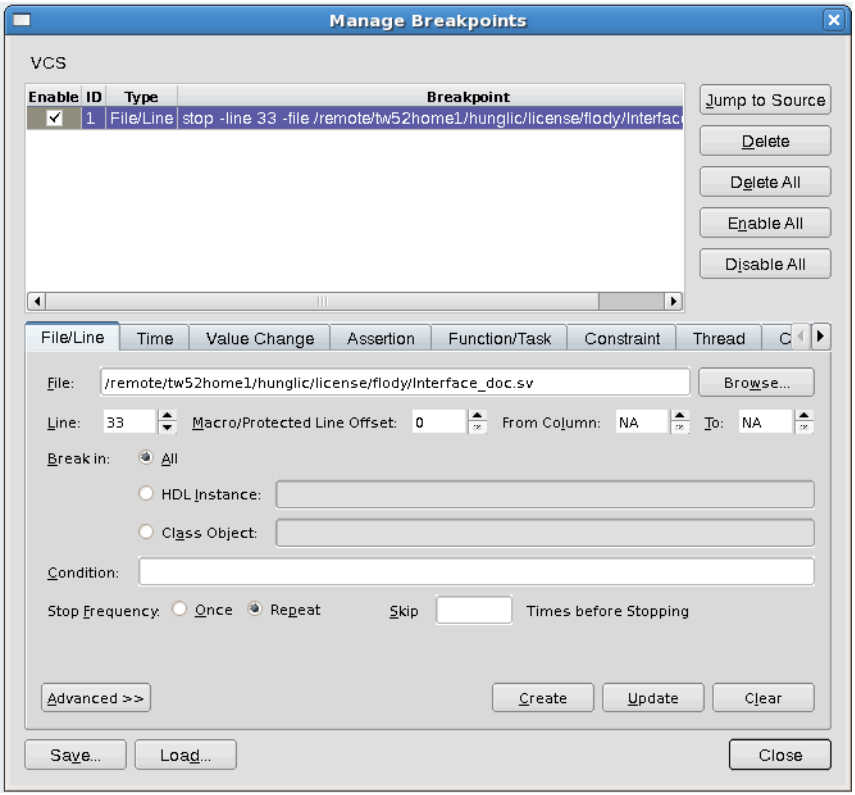


Figure: Manage Breakpoints Form

The *Manage Breakpoints* form includes the following buttons, options, and fields:

- **Simulation Mode:** This field shows the simulator name currently specified in the **Simulator** field of the **Simulation** page in the *Preferences* form.
- **Jump to Source:** Click this button to jump to the line of source code where the selected breakpoint is located. The line of source code where the breakpoint is located is highlighted.
- **Delete:** Click this button to delete the selected breakpoint from the **Breakpoint** list.
- **Delete All:** Click this button to delete all breakpoints from the **Breakpoint** list.
- **Enable All:** Click this button to enable all breakpoints in the **Breakpoint** list.
- **Disable All:** Click this button to disable all breakpoints in the **Breakpoint** list.

- **File/Line**, **Time**, **Value Change**, **Assertion**, **Function/Task**, **Constraint**, and **Thread:** Refer to the **File/Line Tab**, **Time Tab**, **Value Change Tab**, **Assertion Tab**, **Function/Task Tab**, **Constraint Tab**, and **Thread Tab** for details about the breakpoint types.

  In addition, the following fields and buttons are repeated on each of the tabs:

  - **Condition:** Specify an expression to evaluate if a breakpoint should be triggered. This option is equivalent to the UCLI **-cond** <*expression*> option. The  icon is shown in the indicator area of the source code pane when a file/line-based condition breakpoint is enabled.
  - **Stop Frequency:** Specify the stop frequency of the breakpoint by selecting one of the following options. The default is **Repeat**.
    Once: When this option is selected, the breakpoint stops one time. This option is equivalent to the UCLI **-once** option.
    Repeat: When this option is selected, the breakpoint stops every time.
    Skip 'n' Times before Stopping: Enter a number in the **Skip 'n' Times before Stopping** field to specify the skip number. The default skip number is *0*. This field is equivalent to the UCLI **-skip** <*number*> option.
  - **Advanced >>:** Click this button to display additional UCLI options.

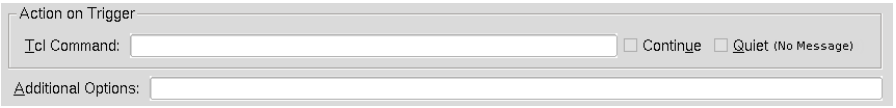


Figure: Advanced UCLI Options

The advanced UCLI options include the following options and fields:

Tcl Command: Define Tcl commands to execute when a breakpoint is triggered. This field is equivalent to the UCLI **-command** option.

Continue: Set to prevent stopping when a breakpoint is triggered. This option is equivalent to the UCLI **-continue** option.

Quiet: Set to disable printed messages when a breakpoint is triggered. This option is equivalent to the UCLI **-quiet** option.

Additional Options: Specify user-defined UCLI options.

- **Create:** Click this button to add the specified breakpoint to the **Breakpoint** list.

- **Update:** Click this button to update the specified breakpoint in the **Breakpoint** list. The old breakpoint ID number is deleted and changed to a new ID number.
- **Clear:** Click this button to clean up previous breakpoint settings in the active tabs.

- **Save:** Click this button to open the *Save Breakpoints* form where the directory structure can be viewed and a file name (*.*bp*) to save the breakpoint results to can be specified.
- **Load:** Click this button to open the *Load Breakpoints* form where the directory structure can be viewed and a previously saved breakpoint file (*.*bp*) can be loaded.

File/Line Tab

Use this tab to add a breakpoint to a line by specifying the line number in the selected file. When a line contains multiple breakpoints, the  icon is shown in the indicator area of the source code pane.

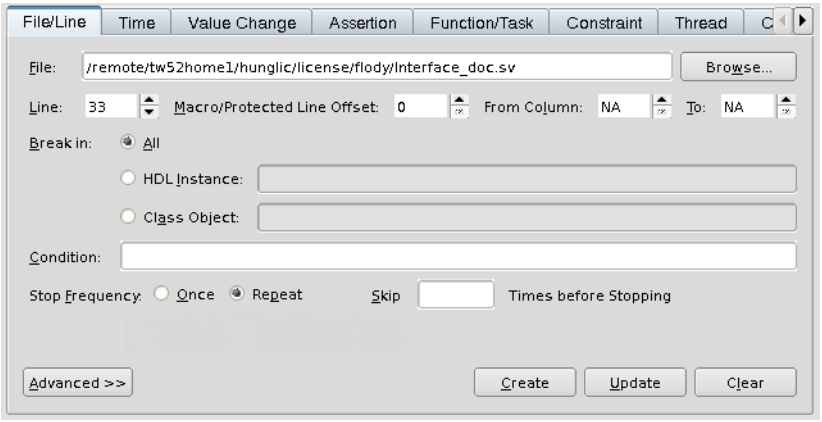


Figure: File/Line Tab

The **File/Line** tab includes the following options and fields:

- **File:** Specify the source code file for the breakpoint by typing the path and file name in the field or clicking the **Browse** button. This field is equivalent to the UCLI **-file** <*filename*> option.
- **Line:** Specify the line number for the breakpoint. This field is equivalent to the UCLI **-line** <*line number*> option.
- **Macro/Protect Line Offset:** Specify an offset line number for a macro or protected code. This field is equivalent to the UCLI **-moffset** option.

- **From Column: __ To: __:** Specify the start column and the end column for the statement breakpoint. These fields are respectively equivalent to the UCLI **-start_col** and **-end_col** options. The  icon is shown in the indicator area of the source code pane when a statement breakpoint is enabled.
- **Break in:** Specify a scope for the breakpoint by selecting one of the following options. The default is **All**.
  - **All:** When this option is selected, the specified breakpoint stops at the specified file/line of HDL instances or class objects.
  - **HDL Instance:** When this option is selected, the specified breakpoint stops at the specified file/line of the HDL instance. This option is equivalent to the UCLI **-instance** <*instance id*> option.
  - **Class Object:** When this option is selected, the specified breakpoint stops at the specified file/line of the class object. This option is equivalent to the UCLI **-object** <*variable*> or **-object_id** <*object id*> options.
- **Condition:** Refer to the **Condition** option for details.
- **Stop Frequency:** Refer to the **Stop Frequency** option for details.
- **Advanced >>:** Refer to the **Advanced** option for details.

Time Tab

Use this tab to add a breakpoint by specifying the simulation time for the breakpoint.

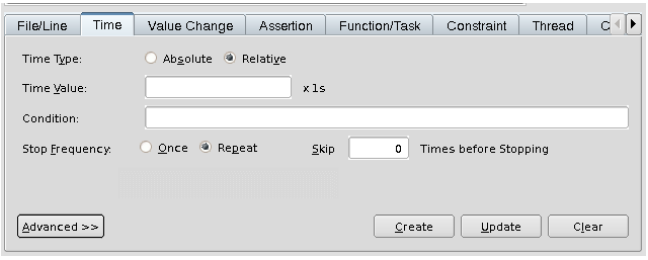


Figure: Time Tab

The **Time** tab includes the following options and fields:

- **Time Type:** Set type of the simulation time by selecting one of the following options. The default is **Relative**.
  - **Absolute:** When this option is selected, the breakpoint stops at the specified absolute simulation time. This option is equivalent to the UCLI **-absolute** <*time*> option.

  - **Relative:** When this option is selected, the breakpoint stops at the specified relative simulation time, counting from the current time. This option is equivalent to the UCLI **-relative** <*time*> option.
- **Time Value:** Enter a number to specify the time for a breakpoint. The time unit is synchronized with the simulator time scale and time unit.
- **Condition:** Refer to the **Condition** option for details.
- **Stop Frequency:** Refer to the **Stop Frequency** option for details.
- **Advanced >>:** Refer to the **Advanced** option for details.

Value Change Tab

Use this tab to add a breakpoint by specifying an event when the variable value changes in a signal or a dynamic object.

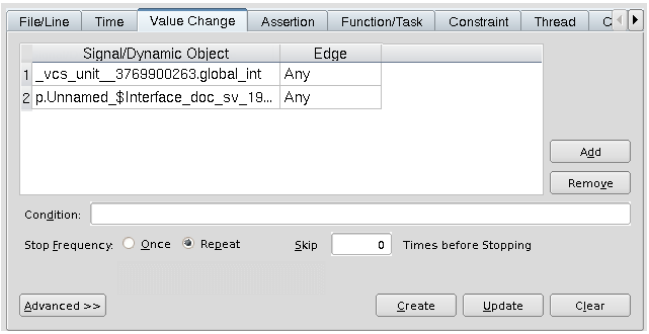


Figure: Value Change Tab

The **Value Change** tab includes the following options and fields:

- **Add:** Click this button to add a row under the **Signal/Dynamic Object** and **Edge** columns to stop/pause the simulation.

  For an HDL signal, specify the full hierarchy name of the signal in the **Signal/Dynamic Object** column, and double-click the **Edge** column to select the **Any** (equivalent to the UCLI **-change** <*id*> option), **Rising** (equivalent to the UCLI **-posedge** <*id*> option), or **Falling** (equivalent to the UCLI **-negedge** <*id*> option) option.

  For a dynamic object, specify the member data of the object's full hierarchy name in the **Signal/Dynamic Object** column (the object must belong to the current simulation scope and a testbench scope), and double-click the **Edge** column to select the **Any** option (equivalent to the UCLI **-change** <*id*> option).

- **Remove:** Click this button to remove the selected row under the **Signal/Dynamic Object** and **Edge** columns.

- **Condition:** Refer to the **Condition** option for details.
- **Stop Frequency:** Refer to the **Stop Frequency** option for details.
- **Advanced >>:** Refer to the **Advanced** option for details.

Assertion Tab

Use this tab to add a breakpoint at an assertion event by specifying the event type to stop the simulation. The assertion name should be specified with a full hierarchical name.

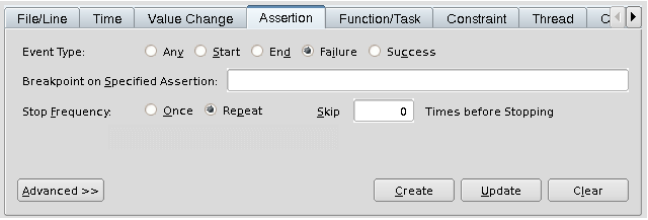


Figure: Assertion Tab

The **Assertion** tab includes the following options and fields:

- **Event Type:** Specify an event type for the breakpoint by selecting one of the following options. The default is **Failure**.
  - **Any:** When this option is selected, the assertion event includes starting, ending, failing and succeeding. This option is equivalent to the UCLI **-any** option.
  - **Start:** When this option is selected, the simulation is stopped at the start of the assertion events. This option is equivalent to the UCLI **-start** option.
  - **End:** When this option is selected, the simulation is stopped at the end of the assertion events. This option is equivalent to the UCLI **-end** option.
  - **Failure:** When this option is selected, the simulation is stopped at the failure of the assertion events. This option is equivalent to the UCLI **-failure** option.
  - **Success:** When this option is selected, the simulation is stopped at the success of the assertion events. This option is equivalent to the UCLI **-success** option.
- **Breakpoint on Specified Assertion:** Specify a valid OVA/SVA identifier. This option is equivalent to the UCLI **-assert** <*assert id*> option.
- **Stop Frequency:** Refer to the **Stop Frequency** option for details.

- **Advanced >>:** Refer to the **Advanced** option for details.

Function/Task Tab

Use this tab to add a breakpoint at the beginning or ending of the specified function/task. The  icon is shown in the indicator area of the source code pane when a function/task breakpoint is enabled.

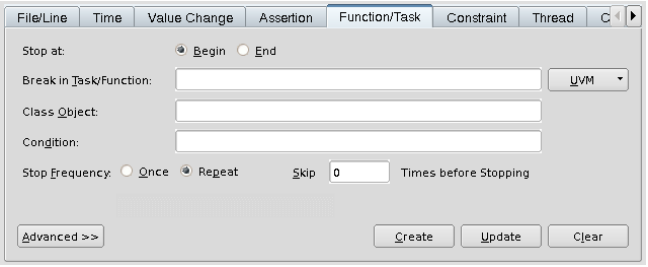


Figure: Function/Task Tab

The **Function** tab includes the following options and fields:

- **Stop at:** Specify where to stop the simulation within the task/function call by selecting one of the following options. The default is **Begin**.
  - **Begin:** When this option is selected, the simulation stops at the beginning of the task/function call
  - **End:** When this option is selected, the simulation stops at the ending of the task/function call.
- **Break in Task/Function:** Specify the function or task name where the simulation is stopped. This field is equivalent to the UCLI **-in** <*name*> option.
  Click the **UVM** button to open the UVM drop-down menu. The menu divides the UVM component breakpoints to the **uvm_root::phase_started**, **uvm_root::build_phase**, **uvm_root::run_phase**, **Build**, **Run**, **Cleanup**, and **Report** phases.
  Select the **uvm_root:phase_started**, **uvm_root::build_phase**, or **uvm_root::run_phase** task to stop the simulation execution in the specified task.
  Select a task from the **Build**, **Run**, **Cleanup**, or **Report** phase to set a breakpoint in the selected phase.
- **Class Object:** Specify a class object by typing the name in this field or dropping an object dragged from the **Local** tab, **Watch** tab, or the source code pane. The specified function must be within the given class. This field is equivalent to the UCLI **-object** <*variable*> or **-object_id** <*object id*> option.

- **Condition:** Refer to the **Condition** option for details.
- **Stop Frequency:** Refer to the **Stop Frequency** option for details.
- **Advanced >>:** Refer to the **Advanced** option for details.

Constraint Tab

Use this tab to stop the simulation when hitting all or specified randomize calls.

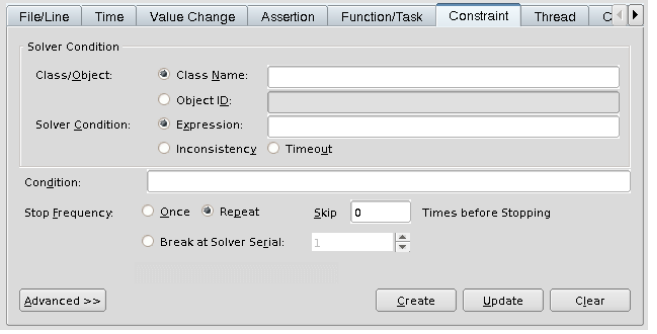


Figure: Constraint Tab

The **Constraint** tab includes the following options and fields:

- **Class/Object:** Specify the class or object of the constraint to by selecting one of the following options. The default is **Class Name**.
  - **Class Name:** When this option is selected, enter the class name of the built-in randomize method. This field is equivalent to the UCLI **stop -solver** [**-class** <*class name*>] option.
  - **Object ID:** When this option is selected, specify the object ID of the randomize method. This field is equivalent to the UCLI **-object_id** <*object id*> option.
- **Solver Condition:** Specify the constraint condition by selecting one of the following options. The default is **Expression**.
  - **Expression:** When this option is selected, enter an expression to evaluate if a breakpoint should be triggered. This field is equivalent to the UCLI **-solver_cond** {<*expression*>}option.
  - **Inconsistency:** When this option is selected, the simulator stops when constraint conflicts occur. This option is equivalent to the UCLI **-inconsistency** option.
  - **Timeout:** When this option is selected, the simulator stops when constraints cannot be solved within the time limit. This option is equivalent to the UCLI **-timeout** option.
- **Condition:** Refer to the **Condition** option for details.

- **Stop Frequency:** Refer to the **Stop Frequency** option for details.
- **Advanced >>:** Refer to the **Advanced** option for details, except the **Break at Solver Serial** option.

  Break at Solver Serial: Specify the valid serial number of a randomize call. This option is equivalent to the UCLI **-serial** <*number*> option.

Thread Tab

Use this tab to add a breakpoint at the specified thread.

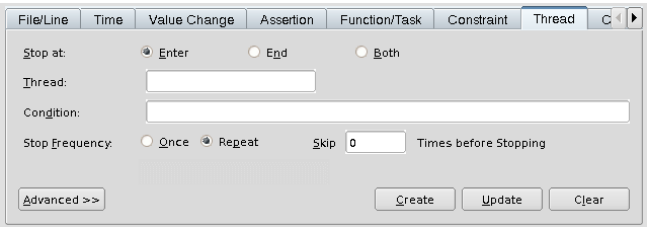


Figure: Thread Tab

The **Thread** tab includes the following options and fields:

- **Stop at:** Specify where to stop the simulation by selecting one of the following options. The default is **Enter**.
  - **Enter:** When this option is selected, the simulation stops at the beginning of the thread. This option is equivalent to the UCLI **-enter** option.
  - **End:** When this option is selected, the simulation stops at the ending of the thread. This option is equivalent to the UCLI **-end** option.
  - **Both:** When this option is selected, the simulation stops at the beginning and ending of the thread.
- **Thread:** Specify a valid tool thread identifier. This field is equivalent to the UCLI **-thread** <*thread id*> option.
- **Condition:** Refer to the **Condition** option for details.
- **Stop Frequency:** Refer to the **Stop Frequency** option for details.

Advanced >>: Refer to the **Advanced** option for details.

Capture Delta Cycle Values

Menu Bar: Simulation -> Capture Delta Cycle Values

Verdi allows you to enable or disable delta cycle dumping at current simulation time in interactive mode using the **Simulation** -> **Capture Delta Cycle Values** option. The following sections describe the use model of the delta cycle dumping in interactive mode:

- Prerequisite
- Use Model
- Limitations

Prerequisite

To enable delta cycle dumping capability in interactive mode, use the `+fsdb+delta=1|2` option at runtime.

Following is the syntax:

`+fsdb+delta=1|2`

`1`: Enables delta cycle dumping capability

`2`: Enables delta cycle dumping, but disables cycle dumping at time `0`

Use the `+fsdb+delta=0` option at runtime to disable the delta cycle dumping capability.

Use Model

Perform the following steps to use this feature:

1. Compile your design with the `-debug_access` option. For example, as shown below:
   ```
   %vcs -sverilog test.v -fsdb -debug_access+all -kdb
   ```
2. Run the design using the `+fsdb+delta=1|2` option. For example, as shown below:
   ```
   %./simv -verdi +fsdb+delta=2
   ```
   This step opens `simv` in GUI mode.
3. Perform one of the following methods to enable delta cycle dumping:
   - Click **Simulation** -> **Capture Delta Cycle Values**

Capture Delta Cycle Values is a check box option; you can toggle it on or off. By default, this option is off. Toggling this option turns on/off delta cycle dumping starting at the current simulation time. This increases the FSDB size. You can try to limit the time range for dumping delta cycle values.

You can use the **Capture Delta Cycle Values** option in the *Preferences -> Simulation* page to enable this feature after simulation restart. This option is disabled by default.

- Use the following dumping options to enable/disable delta cycle dumping:

  `fsdbDumpon +delta` - Enable delta cycle dumping in interactive mode.

  `fsdbDumpoff +delta` - Disable delta cycle dumping in interactive mode.

4. Add signals to the waveform and run the simulation.
5. Enable Region Mode and then expand time.

Limitations

Following is the limitation:

- This feature is not supported if you are executing in past (FSDB Read-Only mode).

C/C++ Debugging

The **C/C++ Debugging** option includes the following sub-options: **Enable C/C++ Debugging** and **Show External Functions**.

Enable C/C++ Debugging

Menu Bar: Simulation -> Enable C/C++ Debugging

This option enables the C/C++ debugging features and loads the symbolic information.

Show External Functions

Menu Bar: Simulation -> Show External Functions

This option shows external functions of the C/C++ debugging results.

Watch View

The **Watch View** option includes the following sub-options: **Add New Watch**, **Delete Watch**, and **Rename Watch**.

Add New Watch

Menu Bar: Simulation -> Add New Watch

NOTE: This option is available in Interactive Simulation Debug mode.

This option adds additional **Watch** subtabs in the **Watch** tab with subtab names of *Watch 1* up to *Watch 5*. A minimum of one and a maximum of five **Watch** subtabs can be added. Refer to the *Watch Tab* section for additional details.

Delete Watch

Menu Bar: Simulation -> Delete Watch

NOTE: This option is available in Interactive Simulation Debug mode.

This option removes the selected **Watch** subtab. At least one **Watch 1** subtab should remain.

Rename Watch

Menu Bar: Simulation -> Rename Watch

NOTE: This option is available in Interactive Simulation Debug mode.

This option opens the *Rename Watch* form where the **Watch** subtabs that are currently being used are listed and can be renamed.

Figure: Rename Watch Form

Interactive Debug Panes

The *Interactive Debug* panes include the **Class**, **Object** tab, **Stack** tab, **Member** tab, **Local** tab, **Constraint Debug** tab, **Source Code** pane, **Interactive_Console** tab, and **Watch** tab.

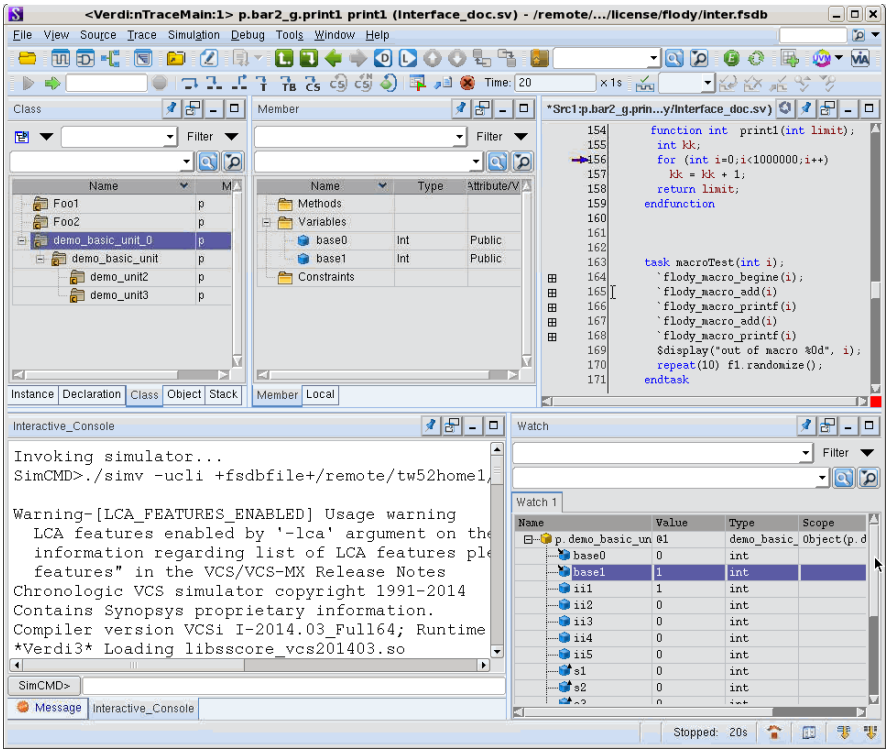


Figure: Interactive Debug Pane

The tabs and right-click options for the *Interactive Debug* panes are summarized and explained on the following pages.

Class Tab

The **Class** tab displays all the defined classes and the instantiations in a hierarchical view.

The **Class** tab displays class object instances only when the simulation (VCS) is compiled with the object related options such as the **debug_access+object** option.

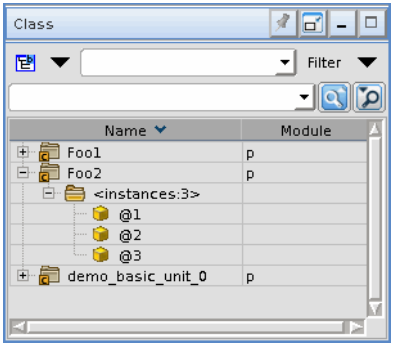


Figure: Class Tab

The order of the rows can be sorted by clicking the column headers. By default, the rows are sorted by **Name**. The **Class** tab includes the following columns:

- **Name:** Shows the defined classes.
- **Module:** Shows the class scopes.

Class Tab Right-click Options

The following right-click menu options are available for the **Class** tab when the testbench code is loaded and Interactive Simulation Debug mode is enabled. The following options are only available from the right mouse button menu.

Show Navigation Text Field

Bind Key: Ctrl+S

Toolbar Icon: 

This option provides three functions for navigating classes in the **Class** tab: viewing modes, filtering, and searching. The default is *on*.

Viewing Modes

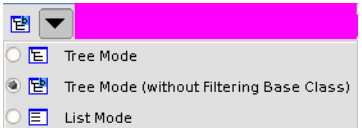


Figure: Viewing Modes for Class Tab

Specify how to view classes by selecting one of the following modes. The default is **Tree Mode**.

- **Tree Mode:** This mode shows the inheritance relationship between all the defined classes with an inheritance view. The root node classes are not derived from any class, the base classes are parent nodes, and the derived classes are child nodes.
- **Tree Mode (without Filtering Base Class):** Similar to **Tree Mode**, but this mode always shows the base library classes (that is, VMM/OVM/UVM library classes) if derived user-defined classes are shown, regardless of the filter setting.
- **List Mode:** This mode shows flattened defined classes.

Filtering

The **Filter** text field hides classes that do not need to be observed. Enter a string in the field or select a history string from the upside-down triangle on the right. The wildcard characters (* or ?) are supported.

The filter types include **VMM Classes**, **RVM Classes**, **UVM Classes**, **OVM Classes**, **Objects**, and **All** library classes to be filtered. Select the filter type by clicking the upside down triangle on the right. Multiple selections are supported. The default is **Objects**.

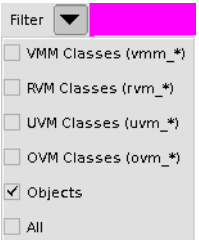


Figure: Filter Types for Class Tab

Searching

The **Search** text field searches classes in the **Class** tab. Enter the target string in the field or select a history string from the upside-down triangle on the right. When strings match, the class is selected and highlighted. The wildcard characters (* or ?) are supported and searching is case-sensitive.

Show Tip

This toggle option turns the tip display *on* or *off*. When this option is turned *on* and the cursor is placed over an object, the tip for the object is displayed. The default is *off*.

Show Reference Count

This option shows the total number of reference counts in parentheses after the object ID name. The default is *off*.

Show Total Memory

This toggle option turns the display of the **Memory** column in the **Class** tab *on* or *off*. The **Memory** column displays the memory usage of classes and objects. The default is *off*.

Show File Information

This option shows the file information of the selected class or instanced object in the **Interconnection** tab of the *Message* pane.

Show Create Location

This option shows the location where the selected object was instantiated in the source code pane.

View Object References

This option shows all objects that reference the selected objects.

Go to Object Creation

This option allows you to go back when an object is created, and is enabled only when a valid object id is selected.

Add to Watch Tab

This option adds the selected object ID to the specified **Watch** tab.

Set Constraint Breakpoint

This option opens the *Breakpoints* form to add the constraint breakpoint of the selected class.

Dump Object to FSDB File

Refer to the **Dump Object to FSDB File** option under the **Local** tab right-click options for details.

Add Object to Waveform

Refer to the **Add Object to Waveform** option under the **Local** tab right-click options for details.

Expand Tree by Level

This option expands the hierarchical tree to the specified levels, **All**, **2 Levels**, **3 Levels**, **4 Levels,** and **5 Levels**.

Collapse All

This option collapses the hierarchical tree in the **Class** tab.

Display Source Code in New Tab

This option displays the source code of the selected class in a newly opened tab of the source code pane.

Object Tab

The **Object** tab displays the hierarchical and dynamic changes of objects in the instances trees. The root nodes are the modules or packages in the imported design that contain objects or references to objects.

Double-click an object in the **Object** tab or drag the object and drop it to the source code pane, and then the debug cursor jumps to the location where the object is declared and the declaration is highlighted in the source code pane.

The **Object** tab displays object information only when the simulation (VCS) is compiled with the object related options such as the **debug_access+object** option.

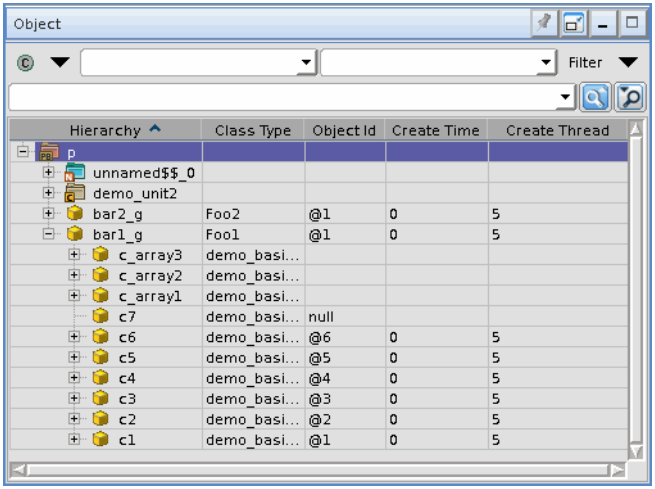


Figure: Object Tab

The order of the rows can be sorted by clicking the column headers. By default, the rows are sorted by **Hierarchy**. The **Object** tab includes the following columns:

- **Hierarchy:** Shows the *p* root node, the instances of classes, and interface classes within the instances tree.
- **Class Type:** Shows the class names of the instances.
- **Object ID:** Shows the values of the active object IDs. Before executing the simulation, the object IDs listed in this column are null.
- **Create Time:** Shows the creation time of each object.
- **Crate Thread:** Shows the execution thread of each object.

Object Tab Right-click Options

The following right-click menu options are available for the **Object** tab when the testbench code is loaded and Interactive Simulation Debug mode is enabled. The following options are only available from the right mouse button menu.

Show Navigation Text Field

Bind Key: Ctrl+S

Toolbar Icon:

This option provides three functions for navigating classes in the **Object** tab: viewing modes, filtering and searching. The default is *on*.

Viewing Modes

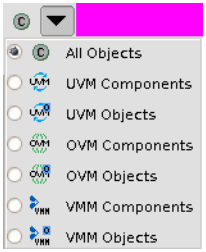


Figure: Viewing Modes for Object Tab

The viewing modes show the library group of interest by selecting one of the following modes. The default is **All Objects**.

- **All Objects:** Shows objects under all scopes and these objects belong to all types.
- **UVM Components:** Shows objects under the *uvm_pkg.uvm_top.top_levels* scope and these objects are derived from uvm_component.
- **UVM Objects:** Shows objects under the *uvm_pkg.uvm_top.top_levels* scope and these objects are derived from uvm_object.
- **OVM Components:** Shows objects under the *ovm_pkg.ovm_top_levels* scope and these objects are derived from ovm_component.
- **OVM Objects:** Shows objects under *ovm_pkg.ovm_top_levels* scope and these objects are derived from ovm_object.
- **VMM Components:** Shows objects under the *_vcs_vmm.vmm_object.roots* scope and these objects are derived from vmm_group, vmm_env, and vmm_simulation.
- **VMM Objects:** Shows objects under the *_vcs_vmm.vmm_object.roots* scope and these objects are derived from vmm_object and vmm_ral_block_or_sys.

Filtering

The **Filter** text fields hide objects that do not need to be observed. Enter a string in the left field to filter the hierarchy name or the right field to filter the class type. The upside-down triangle on the right of the field is for selecting a history search string. The wildcard characters (* or ?) are supported.

The filter types include the **Package Class Variables**, **Module/Program Class Variables**, **Class Static Variables**, **Task/Function Automatic Variables**, **All Variables**, **Base Members**, **Null References**, and **Empty Arrays** object types to be filtered. Select the filter type by clicking the upside down triangle on the right. Multiple selections are supported. The default is **All**.

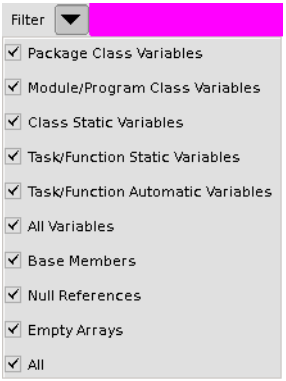


Figure: Filter Types for Object Tab

Searching

The **Search** text field searches objects in the **Object** tab. Enter the target string in the field or select a history string from the upside-down triangle on the right. When strings match, the object is selected and highlighted. The wildcard characters (* or ?) are supported and searching is case-sensitive.

Show Tip

Refer to the **Show Tip** option under the **Class** tab right-click options for details.

Show Total Memory

Refer to the **Show Total Memory** option under the **Class** tab right-click options for details.

Show File Information

Refer to the **Show File Information** option under the **Class** tab right-click options for details.

Show In Class Browser

This option shows the class of the selected object in the **Class** tab.

Show Create Location

Refer to the **Show Create Location** option under the **Class** tab right-click options for details.

View Object References

This option opens the *References* form where all objects that reference the selected object are shown. Double-click a reference to see the line where the object is declared in the source code.

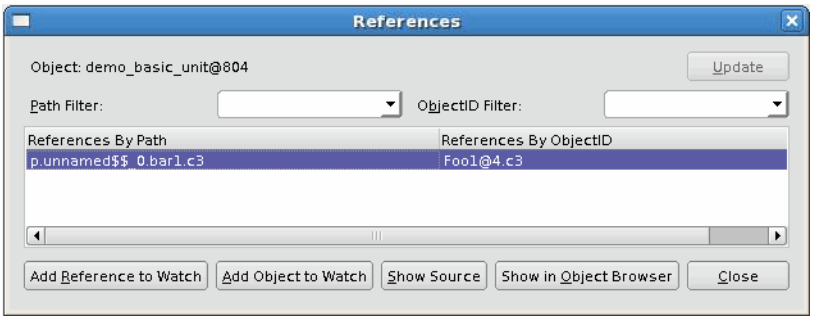


Figure: References Form

The *References* form includes the following options and fields:

- **Update:** Click this button to refresh and retrieve the references again after the simulation is advanced.
- **Path Filter:** Enter a string to filter references by path.
- **Object ID Filter:** Enter a string to filter references by object ID.
- **Add Reference to Watch:** Click this button to add the selected reference to the **Watch** tab.
- **Add Object to Watch:** Click this button to add the selected object to the **Watch** tab.
- **Show Source:** Click to show the line where the object is declared in the source code pane.
- **Show in Object Browser:** Click this button to highlight the selected references in the **Object** tab.
- **Close:** Click this button to close the *References* form.

Go to Object Creation

This option allows you to go back when an object is created, and is enabled only when a valid object id is selected.

Add to Watch Tab

Refer to the **Add to Watch Tab** option under the **Class** tab right-click options for details.

Add Reference to Watch Tab

This option adds the reference path of the object to the specified **Watch** tab.

Dump Object to FSDB File

Refer to the **Dump Object to FSDB File** option under the **Local** tab right-click options for details.

Add Object to Waveform

Refer to the **Add Object to Waveform** option under the **Local** tab right-click options for details.

Expand Tree by Level

Refer to the **Expand Tree by Level** option under the **Class** tab right-click options for details.

Collapse All

Refer to the **Collapse All** option under the **Class** tab right-click options for details.

Display Source Code in New Tab

Refer to the **Display Source Code in New Tab** option under the **Class** tab right-click options for details.

Stack Tab

The **Stack** tab shows the current testbench stack and thread tree.

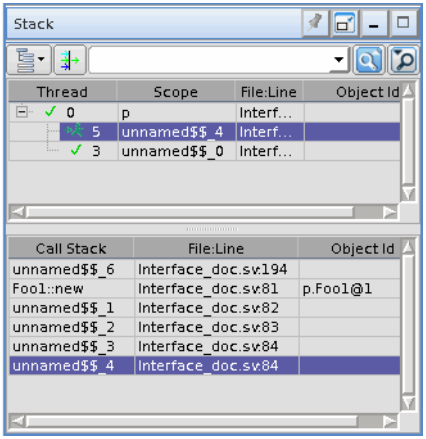


Figure: Stack Tab in Thread Mode

The **Stack** tab in Thread Mode includes the following columns:

- **Thread:** Shows the thread ID.
- **Scope:** Shows the scope name.
- **File: Line:** Shows the file name and line number.
- **Object ID:** Shows the object ID.

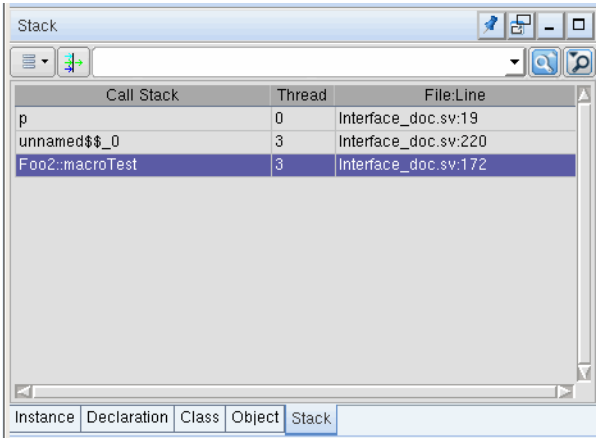


Figure: Stack Tab in Active Mode

The **Stack** tab in Active Mode includes the following columns:

- **Call Stack:** Shows the scope name.
- **Thread:** Shows the thread ID.
- **File: Line:** Shows the file name and line number.

Stack Tab Right-click Options

The following right-click menu options are available for the **Stack** tab when the testbench code is loaded and Interactive Simulation Debug mode is enabled. The following options are only available from the right mouse button menu.

Show Navigation Text Field

Bind Key: Ctrl+S

Toolbar Icon: 

This option provides two functions for navigating threads in the **Stack** tab: viewing modes and searching. The default is *on*.

Viewing Modes

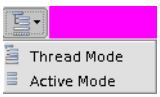


Figure: Viewing Modes for Stack Tab

The **Stack** tab includes **Thread Mode** and **Active Mode** to view all the threads of the program execution. The default is **Active Mode**.

When **Thread Mode** is enabled, the call stack information of all current threads is shown in a hierarchical view. In **Thread Mode**, the full thread tree is displayed in the upper pane. Select a thread in the upper pane to show its call stack in the lower pane. The thread with the  icon indicates that the thread is executing. The thread with the  icon indicates that the thread is ready to be executed.

When **Active Mode** is enabled, only active stacks are shown. In **Active Mode**, the call stack of the active thread is shown. The debug cursor jumps to the line in the source code pane when double-clicking the row in the **Stack** tab. To check the execution sequences of the program, repeat the operations. The local variables are shown in the **Local** tab simultaneously.

To hide internal subscope names of threads, click the  icon. This icon also renames the unnamed blocks to more meaningful names (for example, initial/fork).

Searching

The **Search** text field searches threads in the **Stack** tab. Enter the target string in the  field or select a history string from the upside-down triangle

on the right. When strings match, the thread is selected and highlighted. The wildcard characters (* or ?) are supported and searching is case-sensitive.

Show Tip

Refer to the **Show Tip** option under the **Class** tab right-click options for details.

Show File Information

Refer to the **Show File Information** option under the **Class** tab right-click options for details.

Show In Class Browser

Refer to the **Show In Class Browser** option under the **Class** tab right-click options for details.

Show Create Location

Refer to the **Show Create Location** option under the **Class** tab right-click options for details.

Add Object to Watch

This option is enabled when the current scope of selected **Stack** tab has an object ID. Refer to the **Add Object to Watch Tab** option under the **Local** tab right-click options for details.

Go to Task/Function Invocation

This option allows you to go to next or previous function/task invocation, and is enabled only when a valid method is selected.

Step into Thread

This option advances the simulation until entering the selected thread.

Set Task/Functions Breakpoints

This option sets task/functions breakpoints. Refer to the **Function/Task Tab** of the **Manage Breakpoints** option for details.

Set Thread Breakpoints

This option sets thread breakpoints. Refer to the **Thread Tab** of the **Manage Breakpoints** option for details.

Expand Tree by Level

Refer to the **Expand Tree by Level** option under the **Class** tab right-click options for details.

Collapse All

Refer to the **Collapse All** option under the **Class** tab right-click options for details.

Display Source Code in New Tab

Refer to the **Display Source Code in New Tab** option under the **Class** tab right-click options for details.

Member Tab

The **Member** tab shows the member data depending on the selection. When a class is selected in the **Class** tab, the member methods, member variables, and constraints/constraint expression of the selected class are shown automatically in the **Member** tab. When an object is selected in the **Object** tab, the creation information, member methods, member variables, and constraints/constraint expression of the selected object are shown automatically in the **Member** tab.

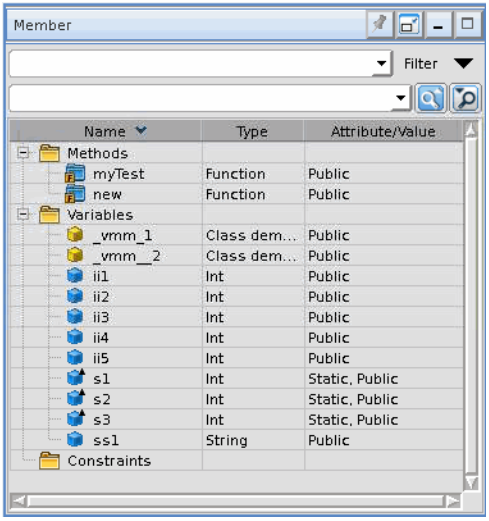


Figure: Member Tab for Class

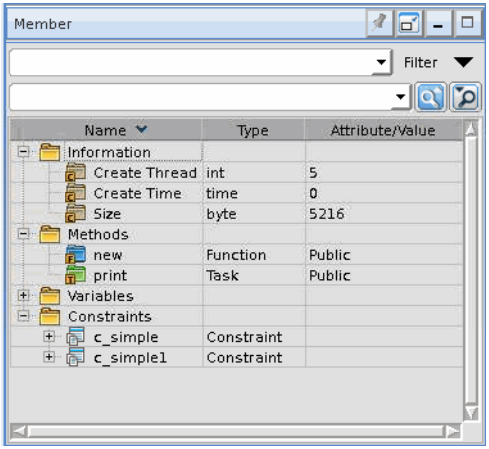


Figure: Member Tab for Object

The order of the rows can be sorted by clicking the column headers. By default, the rows are sorted by **Name**.

The **Member** tab includes the following categories:

- **Methods:** Shows the task and function of the selected class/object.
- **Variables:** Shows the defined data type of the selected class/object.
- **Constraints:** Shows the constraint expression type of the selected class/object.
- **Interface Classes:** Shows the interface classes if a class implements interface classes.
- **Information:** Shows the information for the thread and creation time of the selected object.

Member Tab Right-click Options

The following right-click menu options are available for the **Member** tab when the testbench code is loaded and Interactive Simulation Debug mode is enabled. The following options are only available from the right mouse button menu.

Show Navigation Text Field

Bind Key: Ctrl+S

Toolbar Icon:

This option provides two functions for navigating member data in the **Member** tab: filtering and searching. The default is *on*.

Filtering

The **Filter** text fields hide member data that do not need to be observed. Enter a string in the left field or select a history string from the upside-down triangle on the right. The wildcard characters (* or ?) are supported.

The filter types include the **Base Member**, **Internal Member**, **Static Member**, **Automatic Member**, **Local Member**, **Protected Member**, **Public Member**, **Virtual Member**, **Rand Member**, **Interface Class Member**, **Constraint / Distribution**, **Constraint / Implication**, **Constraint / If**, **Constraint / If-else**, **Constraint / Foreach**, **Constraint / Ordering**, **Constraint / Regular**, and **All** types to be filtered. Select the filter type by clicking the upside down triangle on the right. Multiple selections are supported.

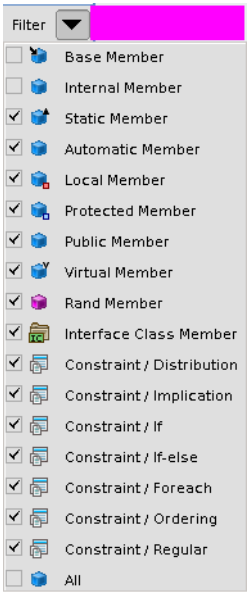


Figure: Filter Types for Member Tab

Searching

The **Search** text field searches members in the **Member** tab. Enter the target string in the field or select a history string from the upside-down triangle on the right. When strings match, the variable is selected and highlighted. The wildcard characters (* or ?) are supported and searching is case-sensitive.

Show Tip

Refer to the **Show Tip** option under the **Local** tab right-click options for details.

Show File Information

Refer to the **Show File Information** option under the **Class** tab right-click options for details.

Show Definition

Refer to the **Show Definition** option under the **Local** tab right-click options for details.

Show in Class Browser

Refer to the **Show In Class Browser** option under the **Object** tab right-click options for details.

View Object References

Refer to the **View Object References** option under the **Class** tab right-click options for details.

Go to Object Creation

This option allows you to go back when an object is created, and is enabled only when a valid object id is selected.

Go to Task/Function Invocation

This option allows you to go to next or previous function/task invocation, and is enabled only when a valid method is selected.

Add to Watch Tab

Refer to the **Add to Watch Tab** option under the **Class** tab right-click options for details.

Add Reference to Watch Tab

Refer to the **Add to Watch Tab** option under the **Class** tab right-click options for details.

Change Value

Refer to the **Change Value** option under the **Local** tab right-click options for details.

Set Breakpoints

Refer to the **Set Breakpoint** option under the **Local** tab right-click options for details.

Stop/Continue Coverage Collecting

This option stops or continues coverage collecting for the selected covergroup instance.

Re-Sample

This option calls the sample() method again for the selected covergroup instance.

Dump Object to FSDB File

Refer to the **Dump Object to FSDB File** option under the **Local** tab right-click options for details.

Add Object to Waveform

Refer to the **Add Object to Waveform** option under the **Local** tab right-click options for details.

Expand Tree by Level

Refer to the **Expand Tree by Level** option under the **Class** tab right-click options for details.

Collapse All

Refer to the **Collapse All** option under the **Class** tab right-click options for details.

Display Source Code in New Tab

Refer to the **Display Source Code in New Tab** option under the **Class** tab right-click options for details.

Local Tab

When the scope of the current simulation belongs to a testbench scope or defined class object, the current simulation scope's name, value, and type are shown automatically in the **Local** tab.

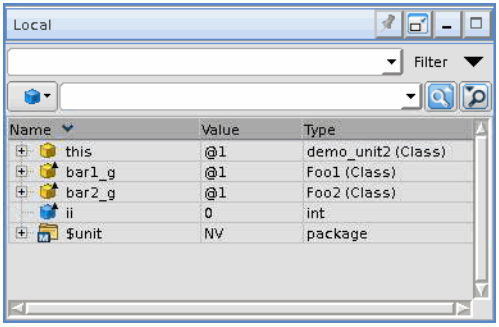


Figure: Local Tab

NOTE: The **Local** tab supports testbench code only.

The order of the rows can be sorted by clicking the column headers. By default, the rows are sorted by **Name**. The **Local** tab includes the following columns:

- **Name:** Shows the variable or signal name.
- **Value:** Shows the value of the variable or signal.
- **Type:** Shows the type of the variable or signal.

Local Tab Right-click Options

The following right-click menu options are available for the **Local** tab when the testbench code is loaded and Interactive Simulation Debug mode is enabled. The following options are available from the right mouse button menu.

Show Navigation Text Field

Bind Key: Ctrl+S

Toolbar Icon: 

This option provides two functions for navigating the current tree in the **Local** tab: filtering and searching. The default is *on*.

Filtering

Enter a string in the **Filter** 

field and select the filter type by clicking the upside-down triangle on the right to display object names matching the input string and filter type in the current tree. The wildcard characters (* or ?) are supported. Multiple selections of the filter types are supported.

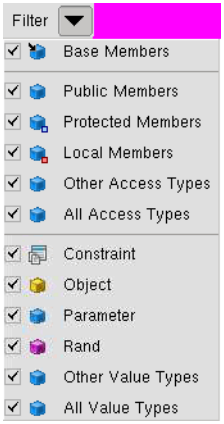


Figure: Filter Types for Local Tab

The filter types are grouped as base member, access types and value types. By default, all filters are turned *on*.

If the attributes of a variable belong to multiple filter options, the variable is shown only after all matched filter options are turned *on*. For example, if a public member variable of 'this' is an object, and it is defined in a base member, then after the **Base Members**, **Public Members**, and **Object** options are all turned *on*, the member variable is shown in the 'this'.

Base Members Group
Select **Base Members** to filter class member variables defined in the base classes of the parent object.

NOTE: If the parent item does not match the selected base member type, the child items are not shown in the **Local** or **Watch** pane.

Access Type Group
This group includes **Public Members**, **Protected Members**, **Local Members**, **Other Access Types**, and **All Access Types**.

- **Public Members:** Select to filter public members in class members.
- **Protected Members:** Select to filter protected members in class members.
- **Local Members:** Select to filter local members in class members.
- **Other Access Types:** Select to filter variables that are not class members (for example, local variables in the current scope or parameters under packages).

- **All Access Types:** When this option is not enabled, select this option to select all in this group. When this option is enabled, select this option to deselect all.

NOTE: If the parent item does not match the selected access types, the child items are not shown in the **Local** or **Watch** pane.

Value Type Group

This group includes **Constraint**, **Object**, **Parameter**, **Rand**, **Other Value Types**, and **All Value Types**.

- **Constraint:** Select to filter constraint blocks and constraint expressions.
- **Object:** Select to filter variables of an object.
- **Parameter:** Select to filter parameters and constants.
- **Rand:** Select to filter variables with rand attribute.
- **Other Value Types:** Select to filter value types that are not included above.
- **All Value Types:** Select to filter all value types in this group.

NOTE: If the parent item does not match the selected value types, and the child item matches the item both in Base Members and access type, the parent item is shown in the **Local** or **Watch** pane.

Searching

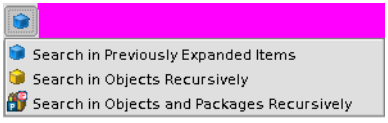


Figure: Searching Modes for Local Tab

The searching modes searches for items, objects or packages of interest by selecting one of the following modes. The default is **Search in Previously Expanded Items**.

- **Search in Previously Expanded Items:** When this mode is selected, only the tree that has been expanded previously is searched. If an item is previously shown, and then its parent is collapsed to hide the item, this hidden item can still be searched. If the **Local** tab flushes (for example, new simulation stops or switch panes), the hidden item may be removed, so it cannot be searched.
- **Search in Objects Recursively:** When this mode is selected, search into objects recursively to find the match in the subtree. This mode searches into

objects that have not been expanded previously. If an object's object ID is the same as any of its parent, the search stops at this object.

- **Search in Objects and Packages Recursively:** When this mode is selected, the search is the same as **Search in Objects and Packages Recursively**, but this mode searches into all packages and their objects recursively shown in the **Local** tab. Type a search string in the **Search**  text field and press the **Enter** key to locate the first match. Click the **Previous Node**/**Next Node** icons to find the previous/next matching tree node. The wildcard characters (* or ?) are supported and searching is case-sensitive.

Show Tip

This toggle option turns the tip display *on* or *off*. When this option is turned *on* and the cursor is placed over a row in the **Local** and **Watch** tabs, the tip for the selected data is displayed.

Set/Unset Marker

Bind Key: A

This option sets the selected tree nodes in the **Watch**/**Local** tab as the base point. The rows specified as the base point are highlighted in yellow. To clear previously set markers, left-click to select markers and invoke this option again.

Unset All Markers

This option removes all markers set previously from the **Set/Unset Marker** option.

Search Previous Marker

Bind Key: [

This option searches for and jumps to the previous marker node in the current **Watch**/**Local** tab.

Search Next Marker

Bind Key:]

This option searches for and jumps to the next marker node in the current **Watch**/**Local** tab.

Collapse All Trees

This option collects all trees in the **Local** tab.

Expand Tree

This option expands all selected tree nodes with one level. If the selected nodes have already been expanded, no change occurs. This option can be applied to multiple selections.

Set Breakpoint

This option sets value-change breakpoints. Refer to the **Value Change Tab** of the **Manage Breakpoints** option for details.

Go to Object Creation

This option allows you to go back when an object is created, and is enabled only when a valid object id is selected.

Set Radix

This option includes six sub-options: **Binary**, **Octal**, **Hexadecimal**, **Decimal**, **Unsigned**, and **Signed 2's Complement**. Refer to the **Signal Value Radix** option in the *nTrace* chapter for details.

Multiple selections are supported in the **Set Radix** menu - one basic radix format among **Binary**, **Octal**, **Decimal**, or **Hexadecimal** and one signed radix format between **Unsigned** and **Signed 2's Complement** can be selected at the same time. When a basic radix is specified, all selected variables are changed to the specified basic radix, but the signed radix is not changed. Similarly, when a signed radix is specified, all selected variables are changed to the specified signed radix, but the basic radix is not changed. For example, if **Signed 2's Complement** is *on* and the radix is **Decimal**, the value is the signed value with 2's complement.

Add to Watch Tab

This option adds the selected nodes to the specified or newly created tab in the **Watch** tab. Invoking the **New Watch Tab** sub-option creates a new **Watch** tab if the number of **Watch** tabs is less than 5.

Add Object to Watch Tab

NOTE: This option is enabled when a tree node of a class object has been selected.

This option adds the selected object to a **Watch** tab for observing the value even when the simulation exits in the living scope of the object. Refer to the **Watch Tab** section for details about **Watch** tab features.

Show Declaration

This option shows the declaration of the selected node in the source code pane. Alternatively, double-clicking the node shows the result in the source code pane.

Show Definition

This option shows the definition of the selected node in the source code pane.

Show Reference

This option lists all references of the selected node in the **Interconnection** tab of the *Message* pane.

Show in Class Browser

NOTE: This option is enabled when a tree node of a class object has been selected.

This option finds the selected object in the **Class** tab to display the content of this class. The class is highlighted in the **Class** tab. Refer to the **Class Tab** section for details about Class Browser features.

View Object References

Refer to the **View Object References** option under the **Object** tab right-click options for details.

Dump to FSDB File

NOTE: This option is enabled when a variable has been selected.

This option dumps HDL signals to the current active FSDB file. Selecting multiple variables to be dumped is supported.

The Verdi platform calls the **fsdbDumpvars depth {signal full-path name}** Tcl dumping option for the simulator to start recording the values for the specified signals and call the **fsdbDumpflush** Tcl dumping option to write signals to the FSDB file. The values of the dumped signals are recorded from the current time. Signals can be dumped at any simulation time during Interactive Simulation Debug mode.

Refer to the **$fsdbDumpvars** and **$fsdbDumpflush** dumping options in the *Linking Novas Files with Simulators and Enabling FSDB Dumping* manual for details.

Dump Object to FSDB File

NOTE: This option is enabled when a dynamic object has been selected.

This option dumps a dynamic object to the current active FSDB file. Selecting multiple dynamic objects to be dumped is supported.

For the supported Tcl dumping options, refer to the **Dump to FSDB File** option description for details.

Dump

NOTE: This option is enabled when an HDL signal has been selected.

This option opens the *Dump* form where dumping options for the selected HDL signal can be set. After clicking the **Apply** or **OK** buttons, signals are dumped to the FSDB file according to the settings. Options in this form have equivalent options in the **$fsdbDumpvars** option. Refer to the **$fsdbDumpvars** dumping options in the *Linking Novas Files with Simulators and Enabling FSDB Dumping* manual for details.

The *Dump* form includes the following fields and options:

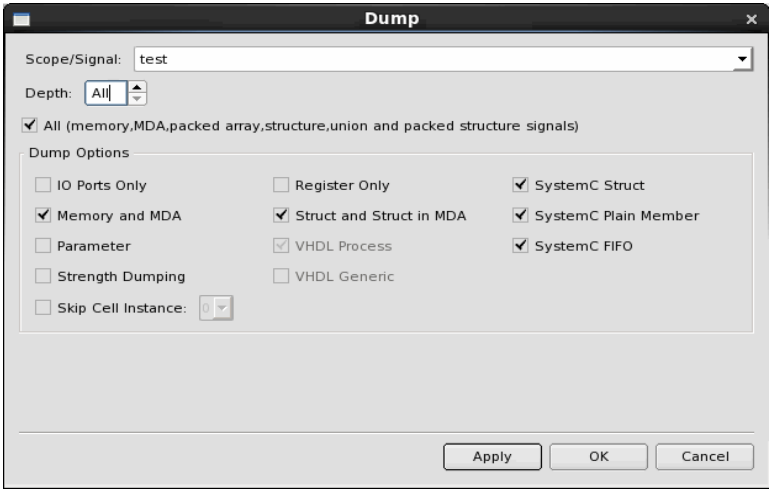


Figure: Dump Form

- **Scope/Signal:** This field shows the name of the selected signal or scope. The name can also be specified by typing in this field.

NOTE: The signal or scope name is based on the active scope where the signal or instance is selected in the source code pane.

- **Depth:** Specify the hierarchy level of subsequent scopes to be dumped. If the subsequent arguments are signals, only these signals are dumped. Level values have the following meanings:
  - All: All signals in all scopes.
  - 1: All signals in the current scope.
  - 2: All signals in the current scope and all scopes one level below.
  - N: All signals in the current scope and all scopes N-1 levels below.
- **All:** This option is turned *on* by default. This option dumps all signals including memory, MDA, packed array, structure, union, and packed structure in all scopes specified in **$fsdbDumpvars** or the entire design if no scope is specified. This option is equivalent to the **+all** option.

The **Dumping Options** section includes the following options:

- **IO Port Only:** Select this option to dump input/output port signals only. This option is equivalent to the **+IO_Only** option. This option is turned *off* by default.

  When this option is turned *on*, all the other options in the **Dumping Options** section and the **All** option is turned *off*.
- **Memory and MDA:** This option dumps all memory and MDA signals. This option is equivalent to the **+mda** option. This option is turned *on* by default.
- **Parameter:** Select this option to dump parameters. This option is equivalent to the **+parameter** option. This option is turned *off* by default.
- **Strength Dumping:** Select this option to enable strength dumping. This option is turned *off* by default.
- **Register Only:** Select this option to dump register signals only. This option is equivalent to the **+Reg_Only** option. This option is turned *off* by default.

  When this option is turned *on*, all other options in the **Dumping Options** section and the **All** option is turned *off*.
- **Struct and Struct in MDA:** This option dumps the structs in all specified scopes. This option is equivalent to the **+struct** option. This option is turned *on* by default.

- **VHDL Process:** This option dumps VHDL processes. This option is equivalent to the **+trace_process** option. This option is turned *on* by default.
- **VHDL Generic:** This option dumps VHDL generics. This option is equivalent to the **+generic** option. This option is turned *on* by default.
- **Skip Cell Instance:** Select this option to dump cell instances with the specified mode. This option is turned *off* by default.
  - 0: Disable dumping of cell instances. This is the default mode.
  - 1: Skip all cell information.
  - 2: Dump all ports of cell instances.

  This option is equivalent to the **+skip_cell_instance=mode** option.
- **SystemC Struct:** This option dumps default items and variables of user-defined structs. This option is equivalent to the **+sc_struct** option. This option is turned *on* by default.
- **SystemC Plain Member:** This option dumps default items and plain members. This option is equivalent to the **+sc_plain** option. This option is turned *on* by default.
- **SystemC FIFO:** This option dumps default items and SystemC sc_fifo and tlm_fifo. This option is equivalent to the **+sc_fifo** option. This option is turned *on* by default.

Add to Waveform

This option dumps and adds HDL signals or dynamic objects to the *nWave* window. When HDL signals are selected, the selected signals are added. When dynamic objects are selected, the selected first-level object (for example, "this") or the object member (for example, "this.addr") is added.

The following are sub-commands of the **Add to Waveform** command.

New Waveform

This command enables you to create a new waveform window, open an annotation file, and add signals.

Add to Wave[n]

Bind Key Ctrl+4

This command enables you to add selected objects to the last adding signal waveform or the new created waveform.

Add to New Group

This command creates a new group in the waveform and adds selected objects to this group.

*Wave[k]

Bind Key Ctrl+W

This command adds selected objects to the waveform window. For example, *Wave[k], k is the primary waveform window Id.

Wave[m]

This command adds selected objects to waveform window. For example, Wave[m], m is the window id.

Add Object to Waveform

This option adds the object ID of the selected object to the *nWave* window. The members of the object are dumped and can be expanded in the waveform by double-clicking the object.

Change Value

This option opens the *Force Signal Value* form where the value for the selected signal can be changed. Refer to the **Force Signal Value** option for details.

Set Constraint Mode

NOTE: This option is available when a constraint block has been selected.

This option includes two sub-options: **On** and **Off**. The sub-options control the constraint block status. When the **Set Constraint Mode -> On** option is invoked, constraint mode is enabled. When the **Set Constraint Mode -> Off** option is invoked, constraint mode is disabled.

Set Rand Mode

NOTE: This option is available when a variable has been selected.

This option includes two sub-options: **On** and **Off**. The sub-options control the randomization of the random variables. When the **Set Rand Mode -> On** option is invoked, the randomization of the variable is enabled. When the **Set Rand Mode -> Off** option is invoked, the randomization of the variable is disabled.

Enable Constraint

This option enables the constraint after a disabled leaf statement in a constraint block tree is selected.

Disable Constraint

This option disables the constraint after an enabled leaf statement in a constraint block tree is selected.

Add Constraint

This option adds constraints for a runtime constraint block or a class object. When a runtime constraint block is selected, this option opens the *Add Constraints* form where a new expression can be added by entering the expression first and then clicking the **Add** button.

When a class object is selected, this option opens the *Add Constraints* form where runtime constraints can be created by entering the constraint block name and its expression first and then clicking the **Add** button.

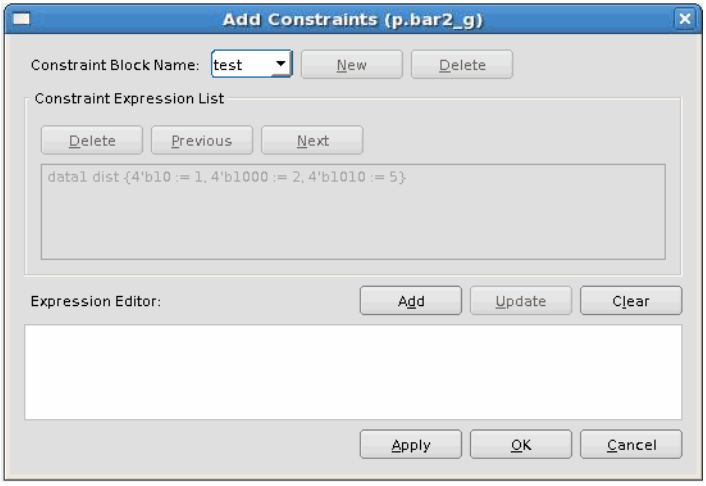


Figure: Add Constraints Form

The *Add Constraints* form includes the following options:

- **Constraint Block Name:** Enter a name for a constraint block name in this field.

  New: Click this button to clear current settings of the runtime constraint block.

Deleted: Click this button to delete settings of the selected runtime constraint block.

- **Expression Editor:** Type the expression for the constraint block in this field (for example, *solve byte1 before byte2*).

  Add: Click this button to add a new expression to the new constraint block and the new statement is shown in the **Constraint Expression List** section.

  Update: After a constraint expression is selected in the **Constraint Expression List** section, click this button to replace the original statement with the new statement.

Edit Constraint

NOTE: This option is enabled after a constraint is created.

After a runtime constraint expression is selected, this option opens the *Change Constraint Expression* form where the current constraint expression can be modified. After the new expression is input, click the **OK** button and the new expression is shown in the **Local** tab.

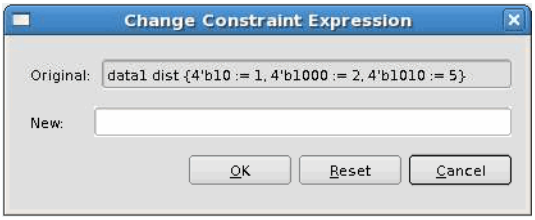


Figure: Change Constraint Expression Form

Delete Constraint

After a runtime constraint block or a runtime constraint expression is selected, this option removes the constraint block or the runtime constraint expression from the **Local** tab. When the only constraint expression in the constraint block is deleted, the constraint block is deleted automatically. Pressing the **Delete** key also deletes the constraint.

Constraint Debug Tab

The Verdi platform provides the constraint solver engine for randomization. This tab shows details about how the constraints were solved in the **Solver** subtab and shows the relationship between the variables in the **Relation** subtab. After using the **Redo Randomize Call and Step in Constraint Solver** option and the **Do**

Distribution Re-randomize and Step in Constraint Solver options for rerandomization and distribution debug, the **Distribution_R** and **Histogram_R** subtabs are shown.

The **Merge** or **Split** option provides different viewing layouts. For the **Solver** and **Relation** subtabs, when the **Split** option is enabled, the **Solver** subtab is placed on the left side and the **Relation** subtab is placed on the right side. When the **Merge** option is enabled, these two subtabs are combined. The **Distribution_R** and **Histogram_R** subtabs can also be merged or split.

Solver Subtab

The **Solver** subtab can be invoked to display the randomize calls. A randomize call is the root node and can be expanded to view the partitions of the randomize call.

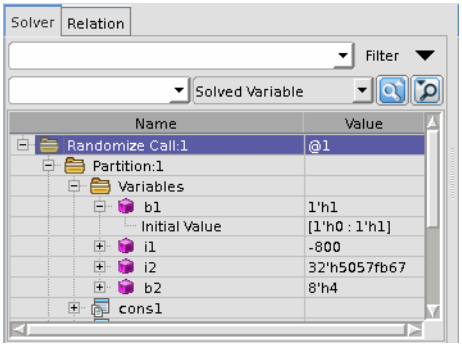


Figure: Solver Subtab

The **Solver** subtab includes the following columns:

- **Name:** Shows the variable or constant block names of randomize calls.
- **Value:** Shows the value of the variable or constant block.

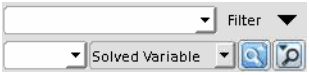
Solver Subtab Right-click Options

The following right-click menu options are available for the **Solver** subtab when Interactive Simulation Debug mode is enabled. The following options are available from the right mouse button menu.

Show Navigation Text Field

This option provides two functions for navigating the current tree in the **Solver** subtab: filtering and searching. The default is *on*.

Bind Key: Ctrl+S

Toolbar Icon: 

Filtering

The **Filter** text field shows variables, constraint blocks, and constraint statements that need to be observed. Enter a string in the  field or select a history string from the upside-down triangle on the right. The wildcard characters (* or ?) are supported.

The filter types include the **On Constraint Block**, **Off Constraint Block**, **Soft Dropped Constraint**, **Soft Honored Constraint**, **Not Soft Constraint**, and **All** constraint types to be filtered. Select the filter type by clicking the upside down triangle on the right. Multiple selections are supported. The default is **All**.

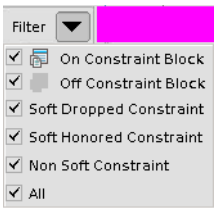


Figure: Filter Types for Solver Subtab

Searching

The **Search** text field searches solved variables or constraint blocks in the **Solver** subtab. Enter the target string in the  field or select a history string from the upside-down triangle on the right. Search for the solved variables or constraint blocks by selecting the type in the **Search Type**  field. When strings match, the solved variables or constraint blocks are selected and highlighted. The wildcard characters (* or ?) are supported and searching is case-sensitive.

Show Tip

This toggle option turns the tip display *on* or *off*. When this option is turned *on* and the cursor is placed over an item, the tip for the item is displayed.

Different items show different information types in the tip. When the cursor is placed on the **Name** column, the tip shows detailed information about the current item. When the cursor is placed on the **Value** column, the tip only shows the contents in the **Value** column for the current item.

Display Source Code in New Tab

NOTE: This option is enabled when a randomize call, function call, constraint block, constraint expression, or variable is selected.

This option highlights the definition line of the selected item in the new source code pane.

Show in Class Browser

NOTE: This option is enabled when a constraint block or variable is selected.

This option shows the definition of the selected item in the **Class** tab. The selected item and its class is highlighted respectively in the **Member** and **Class** tabs.

Show in Local Tab

NOTE: This option is enabled when a root item is selected.

This option shows the selected item in the **Local** tab. If the selected item is a variable, this option displays the selected variable and related variables in the **Relation** subtab.

Show Relation

NOTE: This option is enabled when a variable is selected.

This option shows the selected variable and related variables in the **Relation** subtab.

Show Dropped Reason

NOTE: This option is enabled when a soft dropped constraint is selected. Multiple selections are not supported.

This option shows the dropped reason for the selected dropped constraint in the **Dropped Reason** subtab.

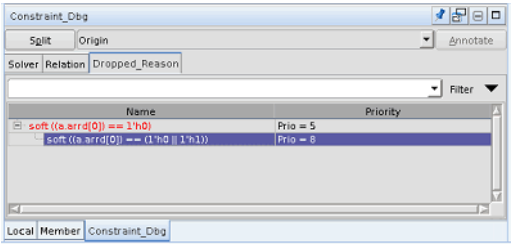


Figure: Dropped Reason Subtab

Set Radix

NOTE: This option is enabled when a variable or constraint expression is selected.

When a variable is selected, this option changes the radix on the solved value and the initial value. When a constraint expression is selected, this option changes the radix on all qualified values in the expression.

Extract Testcases

NOTE: This option is enabled when a randomize call or partition is selected.

When a randomize call is selected, this option opens the *Extract All* form where testcases from all partitions in the selected randomize call is extracted. When a partition is selected, this option opens the *Extract Partition* form where a testcase from the selected partition is extracted.

The directory and the file name of the extracted testcase can be specified in the *Extract All* or *Extract Partition* forms.

Save Changes

This option opens the *Save Constraint Changes* form where changes in Constraint Debug mode can be saved. The default directory is *<working_dir>/<simulator_name>.cst*. The default file name is *constraint_changes.txt*.

Relation Subtab

The **Relation** subtab can be invoked to display how the variable selected in the **Solver** subtab is related to its corresponding variable and other related variables in the **Relation** subtab.

To check the source code of a random variable or constraint statement declaration, double-click the row where the random variable or constraint

statement declaration is located in the **Relation** subtab and the debug cursor jumps to the line in the source code pane with the line highlighted.

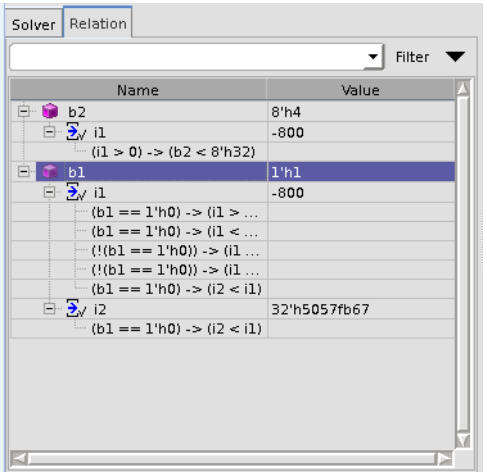


Figure: Relation Subtab

The **Relation** subtab includes the following columns:

- **Name:** Shows the name of the variable selected in the **Solver** subtab, related variables of the selected variable, and constraint expressions of the selected variable and its related variables.
- **Value:** Shows the value of the selected variable and its related variables.

Relation Subtab Right-click Options

The following right-click menu options are available for the **Relation** subtab when Interactive Simulation Debug mode is enabled. The following options are available from the right mouse button menu.

Show Navigation Text Field.

Bind Key: Ctrl+S

Toolbar Icon: 

This option provides a filtering function for navigating the current tree in the **Relation** subtab. The default is *on*

The **Filter** text field shows variables that need to be observed. Enter a string in the  field or select a history string from the upside-down triangle on the right. The wildcard characters (* or ?) are supported.

The filter types include **Solved Together**, **Not Solved Together**, and **All** variables to be filtered. Select the filter type by clicking the upside down triangle on the right. Multiple selections are supported. The default is **All**.

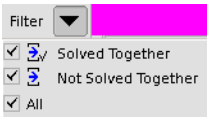


Figure: Filter Types for Relation Subtab

Show Tip

This toggle option turns the tip display *on* or *off*. When this option is turned *on* and the cursor is placed over an item, the tip for the item is displayed.

When the cursor is placed on the **Name** column of a variable, the tip shows the name, type, random type, and random mode of the variable. When the cursor is placed on the **Name** column of a statement, the tip shows the expression. When the cursor is placed on the **Value** column, the tip shows the contents in the column for the current item.

Display Source Code in New Tab

NOTE: This option is enabled when a constraint expression or variable is selected.

This option highlights the definition line of the selected item in the new source code pane.

Show in Class Browser

NOTE: This option is enabled when a variable is selected.

Refer to the **Show in Class Browser** option under the **Solver** subtab right-click options for details.

Show in Solver View

NOTE: This option is enabled when a constraint expression is selected.

This option shows the selected expression in the **Solver** subtab.

Show Relation

NOTE: This option is enabled when a variable from the **Relation** or **Solver** subtabs is selected.

When a variable is selected, this option appends the related variables to the **Relation** subtab.

Set Radix

Refer to the **Set Radix** option under the **Solver** subtab right-click options for details.

Delete

NOTE: This option is enabled when a root variable is selected.

This option deletes the selected variable and the related variables in the **Relation** subtab.

Distribution_R Subtab

To check the value distribution of a random variable, specify the number of rerandomizations, and the **Distribution_R** and **Histogram_R** subtabs are shown. All the solved variables in the specified number of randomize-calls are displayed in the **Distribution_R** subtab.

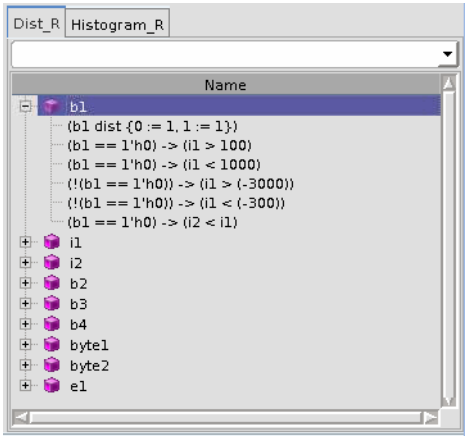


Figure: Distribution Subtab

Distribution Subtab Right-click Options

The following right-click menu options are available for the **Distribution** subtab when Interactive Simulation Debug mode is enabled. The following options are available from the right mouse button menu.

Show Tip

Refer to the **Show Tip** option under the **Solver** subtab right-click options for details.

Display Source Code in New Tab

Refer to the **Display Source Code in New Tab** option under the **Solver** subtab right-click options for details.

Show in Class Browser

Refer to the **Show in Class Browser** option under the **Solver** subtab right-click options for details.

Show Histogram

This option checks the distribution of the selected variable. After a variable is selected in the **Distribution_R** subtab, invoke this option to show the value distributions of the variable as a histogram in the **Histogram_R** subtab.

The **Histogram_R** subtab is blank before this option is invoked for the selected variable.

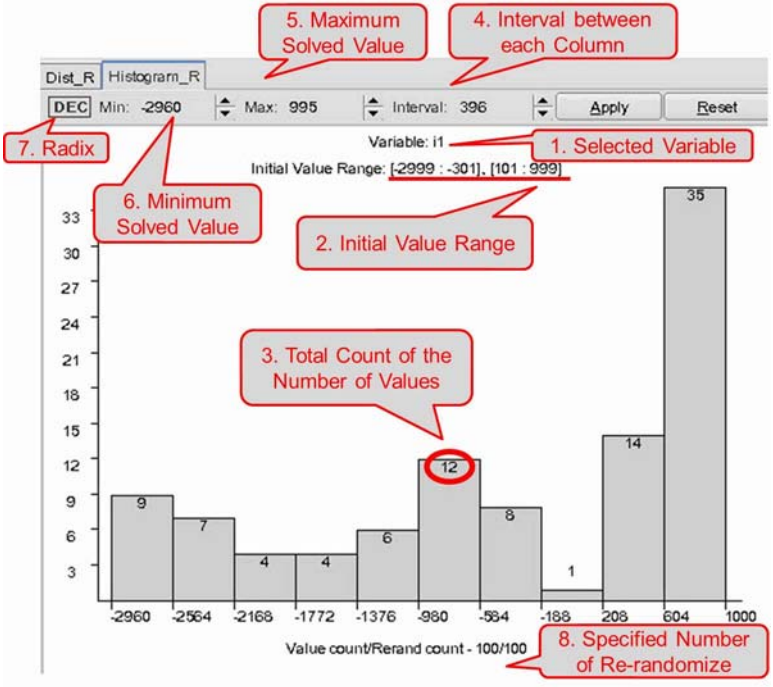


Figure: Distribution Histogram

The following descriptions provide further information for the corresponding item in the *Distribution Histogram* figure.

1. The variable name is displayed. Its initial value is listed under the variable name.
2. The initial value range is displayed. The range of the initial value is the union of the initial value ranges from the specified number of rerandomizations.
3. The total number of values during each interval is displayed. If the number is zero, the number is highlighted in red.
4. **Interval:** The interval for each column can be set.
5. **Max:** The maximum solved value can be set. The maximum value equals to:

   ```
   min_value + interval * number_of_columns
   ```
6. **Min:** The minimum solved value can be set. The minimum value is the smallest value in the solved values.
7. The current radix is displayed.
8. The specified number of rerandomizations is displayed.

After specifying the minimum, maximum and interval respectively in the **Min**, **Max** and **Interval** fields, click the **Apply** button and the histogram is updated.

To replot the default histogram, click the **Reset** button.

Histogram_R Subtab

To check the distribution of the random variable values in a histogram, invoke the **Show Histogram** right-click option in the **Distribution_R** subtab to show the **Histogram_R** subtab.

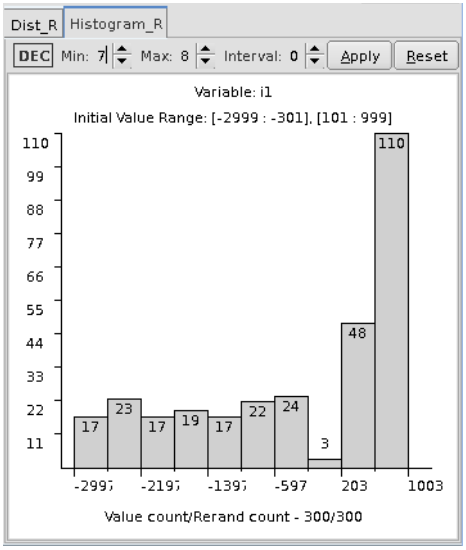


Figure: Histogram_R Subtab

Source Code Pane

Source Code Pane Right-click Options

The following right-click menu options are available for the source code pane when Interactive Simulation Debug mode is enabled. The following options are available from the right mouse button menu.

Run to Source Line

This option stops the simulation at the specified line after an item in the line is selected. Dump to FSDB File

NOTE: This option is available in HDL scopes and not available in testbench scopes.

Refer to the **Dump to FSDB File** option under the **Local** tab right-click options for details.

Dump

NOTE: This option is available in HDL scopes and not available in testbench scopes.

Refer to the **Dump** option under the **Local** tab right-click options for details.

Force Signal Value

NOTE: This option is available when a variable belongs to a block under a testbench scope (except initial blocks) is selected in the source code pane.

This option opens the *Force Signal Value* form where the signal value can be specified and forced during Interactive Debug. Selecting multiple signals for which to change signal value is also supported.

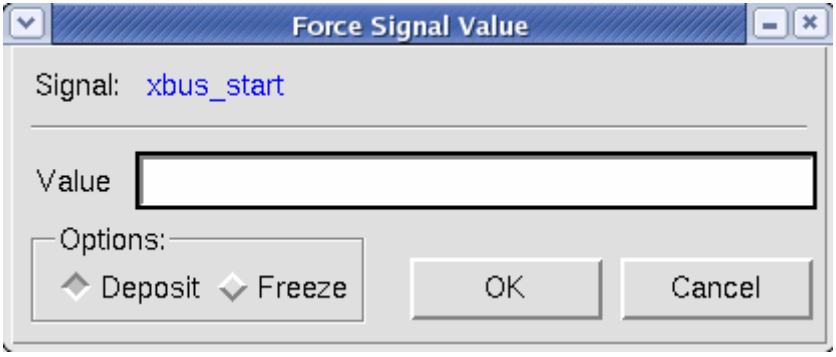


Figure: Force Signal Value

The *Force Signal Value* form includes the following options and fields:

- **Signal:** This field shows the name of the selected signal.
- **Value:** Specify a value in this field.

- **Options:** Specify to deposit or freeze the forced value by selecting one of the following options. The default is **Deposit**.
  - **Deposit:** When this option is selected, the signal value is forced and the changed value is overwritten by a subsequent driver.
  - **Freeze:** When this option is selected, the signal value is always the specified value.

Force to 0

NOTE: This sub-option is available when a variable belonging to an initial block under a testbench scope is selected in the source code pane.

This option forces the selected signal value to *0*. Selecting multiple signals for which to change the signal value is also supported.

Force to 1

NOTE: This sub-option is available when a variable belonging to an initial block under a testbench scope is selected in the source code pane.

This option forces the selected signal value to *1*. Selecting multiple signals for which to change the signal value is also supported.

Force to X

NOTE: This sub-option is available when a variable belonging to an initial block under a testbench scope is selected in the source code pane.

This option forces the selected signal value to *X*. Selecting multiple signals for which to change the signal value is also supported.

Force Release

NOTE: This sub-option is available when a variable belonging to an initial block under a testbench scope is selected in the source code pane.

This option releases the forced value.

Set Force

NOTE: This sub-option is available when a variable belonging to an initial block under a testbench scope is selected in the source code pane.

This option opens the *Force Values* form where complex options for the forced value can be set. The selected signals are displayed in the **Signals** column of the *Force Values* form. The forced valued changes are applied to the listed signals. Selecting multiple signals for which to change the signal value is also supported.

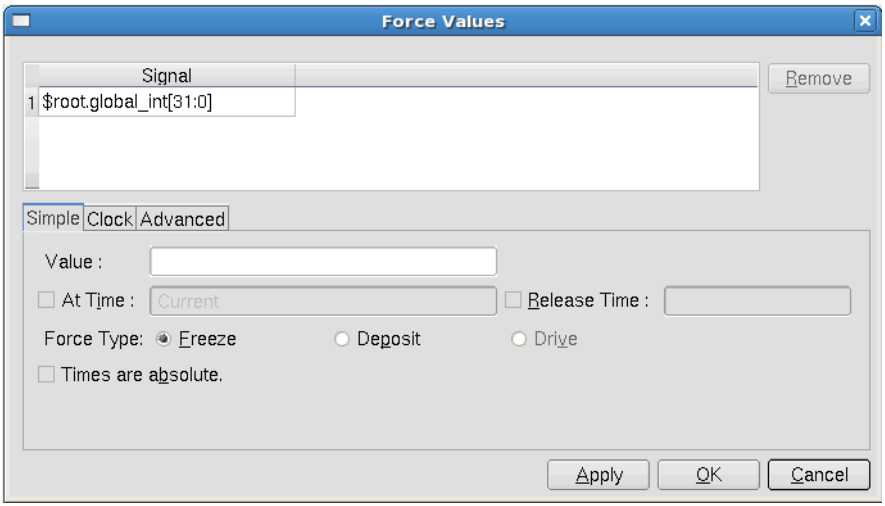


Figure: Force Values Form

The *Force Values* form includes the **Simple**, **Clock** and **Advanced** tabs.

Simple Tab
Use this tab for basic settings.

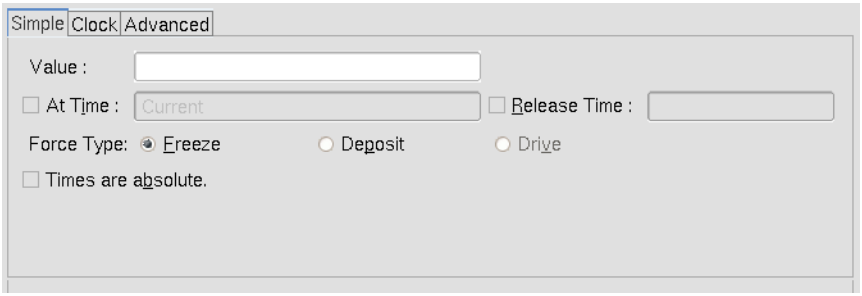


Figure: Force Values Form - Simple Tab

The **Simple** tab includes the following options and fields:

- **Value:** Enter a value for the listed signals.
- **At Time:** Turn *on* this option and enter a time to specify when the forcing time is applied. The time unit is optional and the default is the time

precision of the tool. The unit can be s, ms, us, ns, ps or fs.
When the **Times are absolute** option is turned *off*, the time entered in this field refers to the time expression relative to the current simulation time.

- **Release Time:** Turn *on* this option and enter a time to release the forced value.
- **Force Type:** Specify a force type by selecting one of the following options. The default is **Freeze**.
  - **Freeze:** When this option is selected, the value to the forced signal is frozen.
  - **Deposit:** When this option is selected, the value can be overwritten by a subsequent driver transaction.
  - **Drive:** When this option is selected, a new driver is attached to the signal. This option can only be turned *on* in VHDL designs.
- **Times are absolute:** When this option is turned *on*, the specified times are absolute. The default is *off*.

Clock Tab

Use this tab to force the oscillation value for the clock signals.

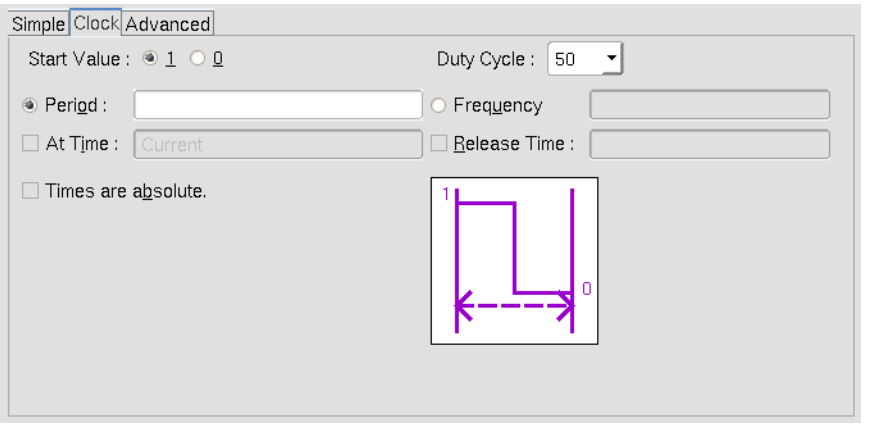


Figure: Force Values Form - Clock Tab

The **Clock** tab includes the following options and fields:

- **Start Value:** Specify the start value by selecting from either 1 or 0. The default is **1**.
- **Duty Cycle:** Specify the duty cycle percentage of the start value by selecting one of the following values: 25, 50, or 75, or entering an integer

number between 0 to 100. A corresponding figure for the duty cycle waveform is shown simultaneously. The default is **50**.

- **Period/Frequency:** Specify the period or frequency of the clock by selecting either the **Period** or **Frequency** option and entering a value in the associated field.
- **At Time:** Turn *on* this option and enter a time to specify when the forcing clock value is applied. When this **At Time** option is turned *off*, the current time is applied to start forcing clock values.
- **Release Time:** Turn *on* this option and enter a time to release clock value.
- **Times are absolute:** When this option is turned *on*, the specified times are absolute. The default is *off*.

Advanced Tab

Use this tab to force values with other methods.

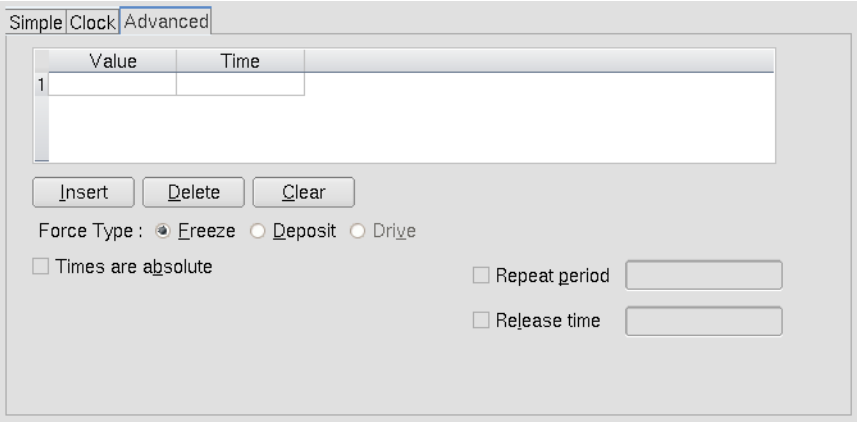


Figure: Force Values Form - Advanced Tab

The **Advanced** tab includes the following options and fields:

- **Value** and **Time:** Enter value changes at different times in these fields.
- **Insert:** Click to add a row of **Value** and **Time** fields.
- **Delete:** Click to delete the selected row of **Value** and **Time** fields.
- **Clear:** Click to clear all rows of **Value** and **Time** fields.
- **Force Type:** Specify a force type by selecting one from the following options. The default is **Freeze**.
  - **Freeze:** When this option is selected, the value to the forced signal is frozen.

  - **Deposit:** When this option is selected, the value can be overwritten by a subsequent driver transaction.
  - **Drive:** When this option is selected, a new driver is attached to the signal. This option can only be turned *on* in VHDL designs.
- **Times are absolute:** When this option is turned *on*, the specified times are absolute. The default is *off*.
- **Repeat period:** When this option is turned *on*, enter a value to repeat the signal waveform. After the sequence of force values is complete and the specified time interval is reached, the sequence is restarted. The default is *off*.
- **Release time:** When this option is turned *on*, enable to enter a time to release values. The default is *off*.

Watch Tab

The **Watch** tab is used to observe the current values of variables in the current simulation scope. Both the values and current scopes are updated as the simulation time changes.

The **Watch** tab is initially empty until a variable is dragged from the source code pane, or **Class**, **Object**, and **Member** tabs and dropped to a **Watch** tab. The supported data types of variables are objects, member data of objects, local variables, and signals that belong to a testbench scope.

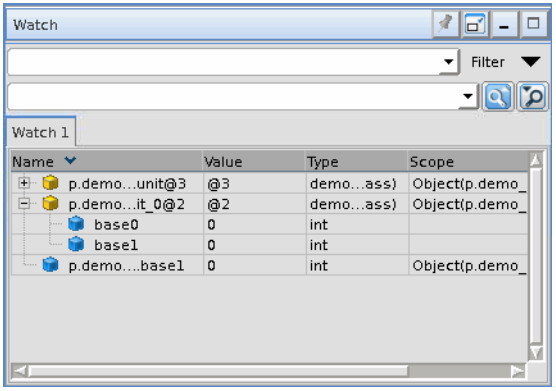


Figure: Watch Tab

NOTE: The **Watch** tabs support both testbench and design code.

NOTE: The function and task blocks, constraint statements, or class templates cannot be added to a **Watch** tab individually.

The order of the rows can be sorted by clicking the column headers. By default, the rows are sorted by **Name**. The **Watch** tab includes the following columns:

- **Name:** Shows the variable or signal name.
- **Value:** Shows the value of the variable or signal.
- **Type:** Shows the type of the variable or signal.
- **Scope:** Shows the full hierarchical path for signals in the HDL scope.

Watch Tab Right-click Options

The following right-click menu options are available for the **Watch** tab when Interactive Simulation Debug mode is enabled. The following options are available from the right mouse button menu.

Show Navigation Text Field

Refer to the Filtering and Searching sections in the **Show Navigation Text Field** option under the **Local** tab right-click options for details.

Show Tip

Refer to the **Show Tip** option under the **Local** tab right-click options for details.

Delete Selected Tree

Bind Key: Delete

This option removes the selected data from the current **Watch** tab.

Delete All

This option removes all data from the current **Watch** tab. Multiple rows can be selected by holding the **Shift** key (selects a range) or the **Ctrl** key (selects individual rows) and a left-click.

Set/Unset Marker

Refer to the **Set/Unset Marker** option under the **Local** tab right-click options for details.

Unset All Markers

Refer to the **Unset All Markers** option under the **Local** tab right-click options for details.

Search Previous Marker

Refer to the **Search Previous Marker** option under the **Local** tab right-click options for details.

Search Next Marker

Refer to the **Search Next Marker** option under the **Local** tab right-click options for details.

Collapse All Trees

This option collects all trees in the **Watch** tab.

Expand Tree

Refer to the **Expand Tree** option under the **Local** tab right-click options for details.

Set Breakpoint

Refer to the **Set Breakpoint** option under the **Local** tab right-click options for details.

Go to Object Creation

This option allows you to go back when an object is created, and is enabled only when a valid object id is selected.

Set Radix

Refer to the **Set Radix** option under the **Local** tab right-click options for details.

Add to Watch Tab

Refer to the **Add to Watch Tab** option under the **Local** tab right-click options for details.

Add Object to Watch Tab

Refer to the **Add Object to Watch Tab** option under the **Local** tab right-click options for details.

Show Declaration

Refer to the **Show Declaration** option under the **Local** tab right-click options for details.

Show Definition

Refer to the **Show Definition** option under the **Local** tab right-click options for details.

Show Reference

Refer to the **Show Reference** option under the **Local** tab right-click options for details.

Show in Class Browser

Refer to the **Show in Class Browser** option under the **Local** tab right-click options for details.

View Object References

Refer to the **View Object References** option under the **Local** tab right-click options for details.

Dump to FSDB File

Refer to the **Dump to FSDB File** option under the **Local** tab right-click options for details.

Dump Object to FSDB File

Refer to the **Dump Object to FSDB File** option under the **Local** tab right-click options for details.

Dump

Refer to the **Dump** option under the **Local** tab right-click options for details.

Add to Waveform

Refer to the **Add to Waveform** option under the **Local** tab right-click options for details.

Add Object to Waveform

Refer to the **Add Object to Waveform** option under the **Local** tab right-click options for details.

Change Value

Refer to the **Change Value** option under the **Local** tab right-click options for details.

Interactive Simulation Debug Toolbar Icons and Fields

Figure: Toolbar Used in nTrace Window with Interactive Simulation Debug Mode

The available toolbar icons may be modified. Refer to the *Toolbars* section of the *User Interface* chapter in the *Verdi User Guide and Tutorial* manual for details.

The available icons are described below.

Run/Continue

Refer to the **Simulation -> Run/Continue** option description for details.

Stop

Refer to the **Simulation -> Stop** option description for details.

Next

Refer to the **Simulation -> Next** option description for details.

Step

Refer to the **Simulation -> Step** option description for details.

Step Out

Refer to the **Simulation -> Step Out** option description for details.

Step in Thread

Refer to the **Simulation -> Step in Thread** option description for details.

Next in Thread

Refer to the **Simulation -> Next in Thread** option description for details.

Step in Testbench

Refer to the **Simulation -> Step in Testbench** option description for details.

Step in Constraint Solver 

Refer to the **Simulation -> Step in Constraint Solver** option description for details.

Redo Randomize Call and Step in Constraint Solver 

Refer to the **Simulation -> Redo Randomize Call and Step in Constraint Solver** option description for details.

Do Distribution Re-randomize and Step in Constraint Solver 

Refer to the **Simulation -> Do Distribution Re-randomize and Step in Constraint Solver** option description for details.

Restart 

Refer to the **Simulation -> Restart** option description for details.

Manage Breakpoints 

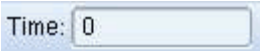
Refer to the **Simulation -> Manage Breakpoints** option description for details.

Go to Current Source Position 

Refer to the **Simulation -> Go to Source Position** option description for details.

Quit 

Refer to the **Simulation -> Quit** option description for details.

Simulation Time 

This field displays the current simulation time.

Set Time Scale Display 

Refer to the **Simulation -> Set Window Time Unit** option description for details.

UVM Interactive Simulation Debug

Overview

The UVM Interactive Simulation Debug support is integrated with Interactive Simulation Debug mode in the Verdi platform. To enter Interactive Simulation Debug mode, invoke the **Tools -> Run Interactive Mode** menu option after the design and the UVM testbench are loaded in the Verdi platform and the simulation configuration is completed.

Before entering the Interactive Simulation Debug mode, the **UVM Debug** option must be enabled in the **Simulation** page of the *Preferences* form or the **-uvmDebug** option is specified on the Verdi command line.

After entering Interactive Simulation Debug mode, to select the desired UVM debug panes, invoke the **Tools -> UVM/OVM/VMM Debug** menu option with associated sub-option or click the  icon with associated UVM menu.

The UVM related panes are opened automatically and docked as new panes to the right of the **Interactive Console** tab as new panes. After selecting the **All Views** option in the UVM menu, all the UVM Interactive Simulation Debug panes, including the *Resource View Pane*, *Factory View Pane*, *Phase View Pane*, *Sequence View Pane*, and *Register View Pane* are opened.

NOTE: To see the sequence history in the *Sequence View* pane, the `+UVM_TR_RECORD` option should be specified to enable the transaction recording.

NOTE: If you want to view the OVM/UVM hierarchy tree, add the `+UVM_VERDI_TRACE=HIER` option to enable OVM/UVM hierarchy tree and TLM port connectivity dumping.

NOTE: If you want to view the register read/write access history in the *Register View*, add the `+UVM_VERDI_TRACE=RAL` option to enable register history dumping.

OVM/UVM Hierarchy Tree

Verdi supports the *OVM/UVM Hierarchy Tree* pane to display the component hierarchy in a test bench. The pane is docked at the upper left corner of the main window, when invoked. Along with the other UVM-Aware windows, the *OVM/UVM Hierarchy Tree* pane can be invoked from the  *UVM/OVM/VMM Debug Windows* icon.

The hierarchy tree is built during runtime. Therefore, the hierarchy tree data has to be complete before any hierarchy tree is displayed in this pane.

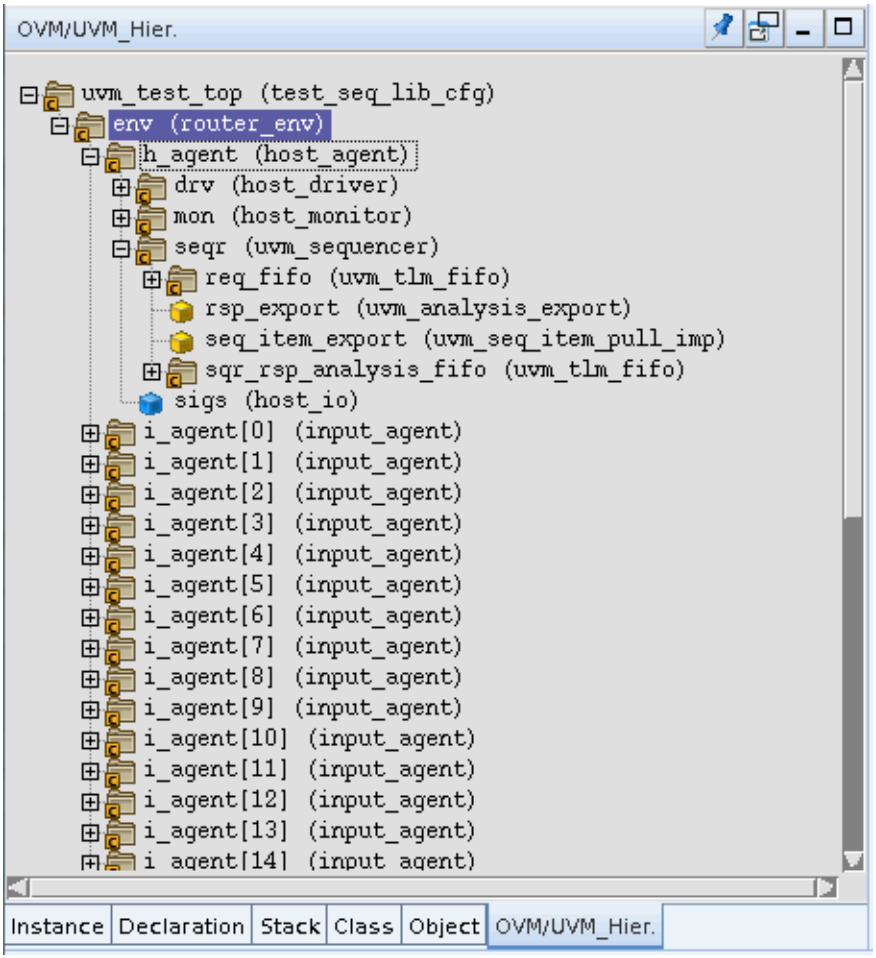


Figure: OVM/UVM Tree Hierarchy Tree pane

OVM/UVM Hierarchy Tree Pane Right-click Options

Show Navigation Text Field

This option opens the Search Bar.

NOTE: This option is available when you right-click on a selected component in the *OVM/UVM Hierarchy Tree* pane.

Show Detail Information

This option displays the call stack for component creation, port connection, and the path of the connected ports of the component.

NOTE: This option is available when you right-click on a selected component in the *OVM/UVM Hierarchy Tree* pane.

Show Definition

This option shows the definition code of the selected item in the *Source Code* pane.

NOTE: This option is available when you right-click on a selected component in the *OVM/UVM Hierarchy Tree* pane.

Show Interface

This option shows the interface code of the selected item in the *Source Code* pane.

NOTE: This option is available when you right-click on a selected component in the *OVM/UVM Hierarchy Tree* pane.

Show Creation

This option shows the creation code of the selected item in the *Source Code* pane.

NOTE: This option is available when you right-click on a selected component in the *OVM/UVM Hierarchy Tree* pane.

Show Connections

This option shows the line that contains the connect function call for a port with a single connection. If a port is connected to multiple ports, it shows the corresponding source code for all connections in the message window and allows you to select a specific port.

NOTE: This option is available when you right-click on a selected component in the *OVM/UVM Hierarchy Tree* pane.

Show Resource Data

This option opens the *Resource View* pane and displays the all available configuration or resource information of the selected component.

NOTE: This option is available when you right-click on a selected component in the *OVM/UVM Hierarchy Tree* pane.

Show Sequence Data

This option opens the *Sequence View* pane and displays the sequence data generated from the selected component.

NOTE: This option is available when you right-click on a selected component in the *OVM/UVM Hierarchy Tree* pane.

Show Original Type

This option opens the *Factory View* pane to display the original type of the selected component.

NOTE: This option is available when you right-click on a selected component in the *OVM/UVM Hierarchy Tree* pane.

Show in Object Browser

This option displays the selected component in the *Object Browser*.

NOTE: This option is available when you right-click on a selected component in the *OVM/UVM Hierarchy Tree* pane.

Show in Class Browser

This option displays the class of the selected component in the *Class Browser*.

NOTE: This option is available when you right-click on a selected component in the *OVM/UVM Hierarchy Tree* pane.

Add to Waveform

This option displays the waveform of the selected component in the *nWave* window.

New Waveform

This command enables you to create a new waveform window, open an annotation file, and add signals.

Add to Wave[n]

Bind Key Ctrl+4

This command enables you to add selected objects to the last adding signal waveform or the new created waveform.

Add to New Group

This command creates a new group in the waveform and adds selected objects to this group.

*Wave[k]

Bind Key Ctrl+W

This command adds selected objects to the waveform window. For example, *Wave[k], k is the primary waveform window Id.

Wave[m]

This command adds selected objects to waveform window. For example, Wave[m], m is the window id.

NOTE: This option is available when you right-click on a selected component in the *OVM/UVM Hierarchy Tree* pane.

Resource View Pane

The *Resource View* pane is a configuration interface that displays all available configuration or resource information in the design. The resources and configurations enable the sharing of information across different parts of a testbench. The configuration or resource information is retrieved from the simulator and the FSDB file. For the history of a configuration item, it is retrieved from the FSDB file.

NOTE: The UVM 1.0p1 library does not include the resource information. If UVM1.0p1 is used in the Verdi platform, the *Resource View* pane appears empty.

The *Resource View* pane include the following panes:

- **Resource View:** Shows the **Resource View** tab. You can sort the data in each column by clicking the column name.

  The *Resource View Pane Layout* has the following columns:

  - **Name:** Shows the name of the resource or configuration.
  - **Scope:** Shows the scope covered by the resource or configuration.
  - **Value:** Shows the value of the resource or configuration.
  - **Type:** Shows the data type of the resource or configuration.
- **Resource History View:** The *Resource History* pane displays the `set` or `get` history of the resource/configuration item selected in the *Resource View* pane. This pane contains five columns and you can sort the data in each column.

  The *Resource History View* pane includes the following columns:

  - **Action:** Shows the `Set/Get` actions of the selected resource or configuration in the *Resource View*.
  - **Scope:** Shows the action applied on the resource or configuration.
  - **Value:** Shows the resource or configuration value related to the action.
  - **Time:** Shows the time at which the action is performed.
  - **Accessor:** Shows the object that performs the action.

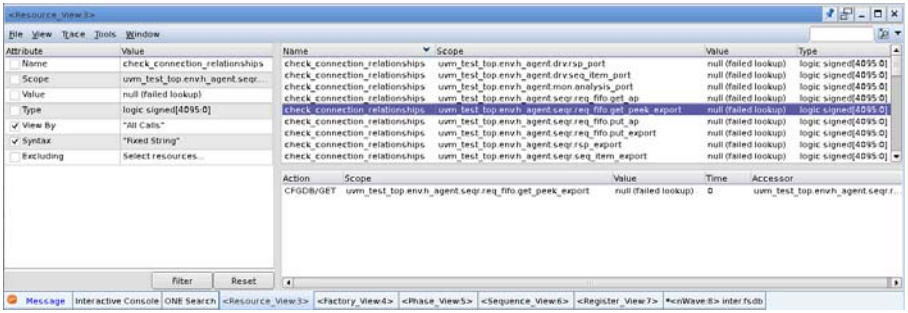


Figure: Resource and History Panes of Resource View

File Options

Export Set/Get Calls

Menu Bar: File -> Export Set/Get Calls

This option opens the **Export UVM Resource Set/Get Calls** file browser where you can specify the name and location of the text file to which the resource call details must be exported to.

View Options

Show Quick Filter

Menu Bar: View -> Show Quick Filter

This option opens the *Quick Filter* pane where the attributes of an item selected in the *Resource View* pane are shown. Choose the desired attributes as the filtering input and click the **Filter** button to show the results in the *Resource View* pane. The default for the **Show Quick Filter** option is *on*.

The *Quick Filter* pane includes the following columns:

- **Name:** Enter the desired name in this field.
- **Scope:** Enter the desired scope name in this field.
- **Value:** Enter the desired value in this field.
- **Type:** Enter the desired type in this field.

- **View By:** Select a value from **All Calls**, **Set Calls**, **Set Calls without Get**, **Get Calls without Set**, and **Multiple Set Calls**. The default is **All Calls**. This option is checked by default and cannot be unchecked.
- **Syntax:** Select a value from **Fixed String**, **Reg Exp**, and **Wildcard**. The default is **Fixed String**. This option is checked by default and cannot be unchecked.
- **Excluding:** Select a resource from **"default_sequence"**, **"recording_detail"**, **"check_connection"** and **Others**. Multiple selections are supported.

After clicking the **Reset** button, all resources are displayed in the *Resource View* pane and all attributes in the *Quick Filter* pane are deselected.

Show Resource Access History

Menu Bar: View -> Show Resource Access History

Toolbar Icon: 

This option shows the *Resource History* pane in the bottom while the *Resource View* pane is shown on the top. The *Resource History* pane displays the `set`/`get` history of the resource/configuration item selected in the *Resource View* pane.

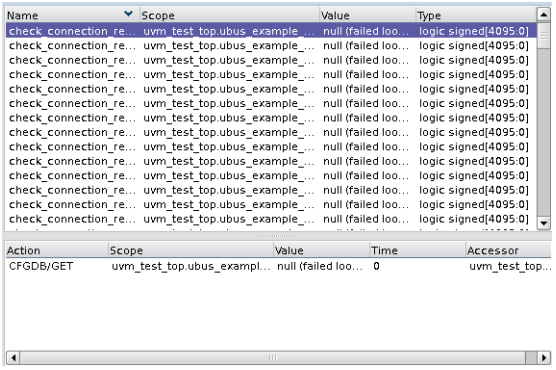


Figure: Resource View Pane (Top) and Resource History Pane (Bottom)

The *Resource History* pane contains the following columns (the data in each column can be sorted):

- **Action:** Shows the SET or GET actions of the selected resource or configuration item.
- **Scope:** Shows the scope where the selected resource or configuration item is applied to.

- **Value:** Shows the resource or configuration value related to the action.
- **Time:** Shows the action time when the resource or configuration is selected.
- **Accessor:** Shows the object that performs the action.

Set Radix

NOTE: This option is enabled after a resource is selected in the *Resource View* pane.

This option checks the different radixes of the selected resource value. The available options are Binary, Octal, Decimal, and Hexadecimal.

Trace Options

Show Source for Virtual Interface

Menu Bar: Trace -> Show Source for Virtual Interface

NOTE: This option is enabled after an item with the Virtual Interface type is selected in the *Resource View* pane.

This option shows the definition code of the selected item in the source code pane.

Show Call in Source Window

Menu Bar: Trace -> Show Call in Source Window

This option shows the code where the `set`/`get` function is called in the source code pane.

Show Accessor in Object Browser

Menu Bar: Trace -> Show Accessor in Object Browser

This option shows the accessor object of the selected function in the *Object Browser* pane after a function is selected in the *Resource History* pane.

Show Accessor in Class Browser

Menu Bar: Trace -> Show Accessor in Class Browser

This option shows the accessor class of the selected function in the *Class Browser* pane after a function is selected in the *Resource History* pane.

Resource View Pane Right-click Options

Show Source for Virtual Interface

Refer to the **Show Source for Virtual Interface** option under the **Trace** options for details.

Set Radix

Refer to the **Set Radix** option under the **View** options for details.

Export Set/Get Calls

This option opens the *Export UVM Resource Set/Get Calls* form where `set`/`get` calls can be export to a specified file.

Resource History Pane Right-click Options

Show Call in Source Window

Refer to the **Show Call in Source Window** option under the **Trace** options for details.

Show Accessor in Object Browser

Refer to the **Show Accessor in Object Browser** option under the **Trace** options for details.

Show Accessor in Class Browser

Refer to the **Show Accessor in Class Browser** option under the **Trace** options for details.

Factory View Pane

The *Factory View* pane provides an interface to view the list of overridden classes that define the UVM factory facility.

By default, the following data types are shown in the *Factory View* pane:

- The original class types that are overridden and the data is obtained from the FSDB file.
- The class types that are registered in the factory and the data is obtained from the simulator.

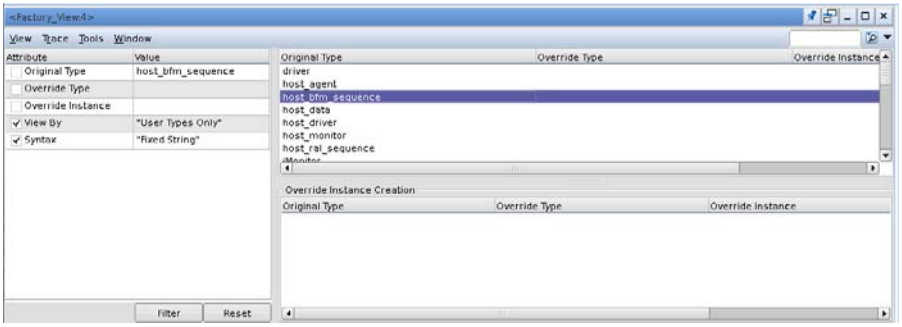


Figure: Factory View Pane

The *Factory View* pane contains the following columns (the data in each column can be sorted):

- **Original Type:** Shows the list of original types.
- **Override Type:** Shows the type of instance overrides.
- **Override Instance:** Shows instance overrides.

View Options

Show Quick Filter

Menu Bar: View -> Show Quick Filter

This option opens the *Quick Filter* pane where the attributes of an item selected in the *Factory View* pane are shown. Choose the desired attributes as the filtering input and click the **Filter** button to show the results in the *Factory View* pane. The default for the **Show Quick Filter** option is *on*.

The *Quick Filter* pane includes the following columns:

- **Original Type:** Enter the desired original type in this field.
- **Override Type:** Enter the desired override type in this field.
- **Override Instance:** Enter the desired override instance name in this field.
- **View By:** Select a value from **All Types**, **User Types Only**, and **Override Types Only**. The default is **Override Types Only**. This option is checked by default and cannot be unchecked.
  - **All Types:** Shows all classes in the *Factory View* pane.
  - **User Types Only:** Shows user-defined class types in the *Factory View* pane. When one of the following definitions is met, the type is considered as a user type:
    1. Override types that do not exist in the original types.
    2. The original types are types without the "uvm_" and "ovm_" prefixes. Any original types with these prefixes are considered as UVM/OVM predefined class types unless these original types are override.
  - **Override Types Only:** Shows overridden classes in the *Factory View* pane.

NOTE: The default type in the **View By** filter has changed from **Override Types Only** to **User Types Only** since the **Override Types Only** setting masks all the user defined factory types.

- **Syntax:** Select a value from **Fixed String**, **Reg Exp**, and **Wildcard**. The default is **Fixed String**. This option is checked by default and cannot be unchecked.

  Only the following syntaxes are supported for wildcard characters:

  - * equals to .* in regular expressions
  - + equals to .+ in regular expressions
  - ? equals to . in regular expressions

After clicking the **Reset** button, all class types are displayed in the *Factory View* pane and all attributes in the *Quick Filter* pane is deselected.

Show Override Instance Creation

Menu Bar: View -> Show Override Instance Creation

This option shows the *Override Instance Creation* pane in the bottom while the *Factory View* pane is shown on the top. The *Override Instance Creation* pane

shows class instances generated by the override class selected in the *Factory View* pane and is enabled by default. The **Time** column in the *Override Instance Creation* pane shows the time when the action is performed.

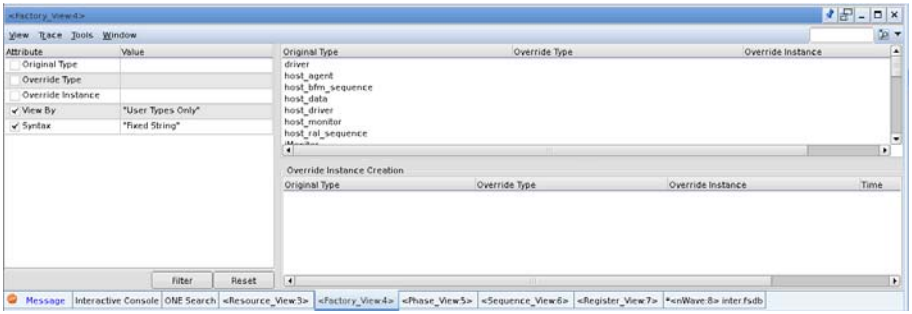


Figure: Factory View Pane (Top) and Override Instance Creation Pane (Bottom)

Trace Options

Show Original type in Class Browser

Menu Bar: Trace -> Show Original Type in Class Browser

This option displays the original class of the selected class in the *Class Browser* pane after a class is selected in the *Factory View* pane.

Show Override type in Class Browser

Menu Bar: Trace -> Show Override Type in Class Browser

This option displays the overridden class of the selected class in the *Class Browser* pane after a class is selected in the *Factory View* pane.

Show Declaration of Original Type

Menu Bar: Trace -> Show Declaration of Original Type

This option displays the source line of the selected original class in the source code pane after a class is selected in the *Factory View* pane.

Show Declaration of Override Type

Menu Bar: Trace -> Show Declaration of Override Type

This option displays the source line of the selected override class in the source code pane after a class is selected in the *Factory View* pane.

Show Reference of Instance Creation

Menu Bar: Trace -> Show Reference of Instance Creation

NOTE: This option is enabled after an FSDB file is imported.

This option displays the source line of the selected instance creation in the source code pane after a class is selected in the *Factory View* pane.

Show Definition of Override Instance

Menu Bar: Trace -> Show Definition of Override Instance

NOTE: This option is enabled after an FSDB file is imported.

This option displays the source line of the selected overridden instance in the source code pane after a class is selected in the *Factory View* pane.

Show Reference of Override Setup

Menu Bar: Trace -> Show Reference of Override Setup

NOTE: This option is enabled when a class with reference is selected.

This option displays the source line of the selected override instance in the source code pane after a class is selected in the *Factory View* pane.

Factory View Pane Right-click Options

Show Original type in Class Browser

Refer to the **Show Original type in Class Browser** option under the **Trace** options for details.

Show Override type in Class Browser

Refer to the **Show Override type in Class Browser** option under the **Trace** options for details.

Show Declaration of Original Type

Refer to the **Show Declaration of Original Type** option under the **Trace** options for details.

Show Declaration of Override Type

Refer to the **Show Declaration of Override Type** option under the **Trace** options for details.

Show Reference of Override Setup

Refer to the **Show Reference of Override Setup** option under the **Trace** options for details.

Override Instance Creation Pane Right-click Options

Show Reference of Instance Creation

Refer to the **Show Reference of Instance Creation** option under the **Trace** options for details.

Show Definition of Override Instance

Refer to the **Show Definition of Override Instance** option under the **Trace** options for details.

Phase View Pane

UVM provides the objection mechanism to control simulation phases. A UVM component may raise an objection when the component is busy, and drop the objection for the simulation to advance to the next phase. Use the *Phase View* pane to see the states and history of each phase and the objections in the selected phase.

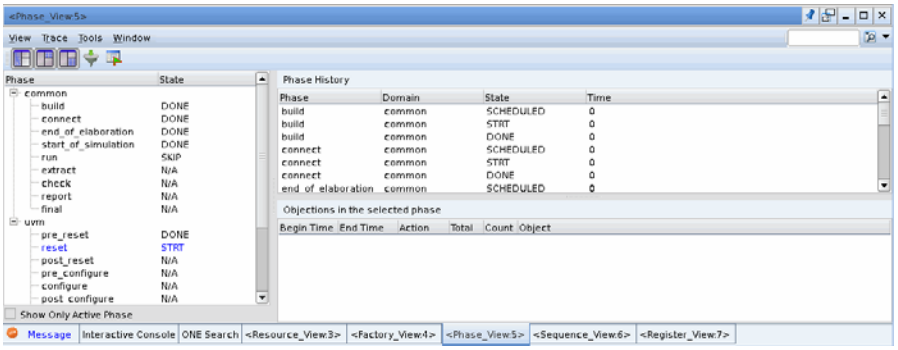


Figure: Phase View Pane

The *Phase View* pane includes the following panes:

- **Phase View:** Shows all the predefined phases of common domain and UVM domain.

  For phases in the *Phase View* pane, the data is retrieved from the simulator. As for current state of the phases, the data is retrieved from the simulator if the FSDB file is unavailable.

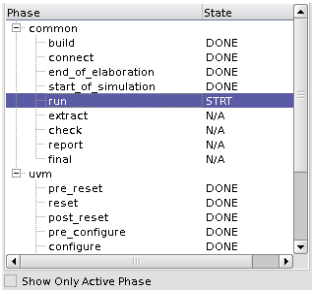


Figure: Phase View Pane

The **State** column displays the status of a phase. Available statuses of a phase include:

- **N/A:** Represents not started.
- **STRT:** Represents executing.
- **DONE:** Represents completed.

Enable the **Show Only Active Phase** option to view only the executing phases.

- **Objection History:** Shows the objection history of the selected phase. For each objection action, the time, action, source object, total count and the top count are displayed.

  For the objection history, the data is retrieved form the FSDB file.

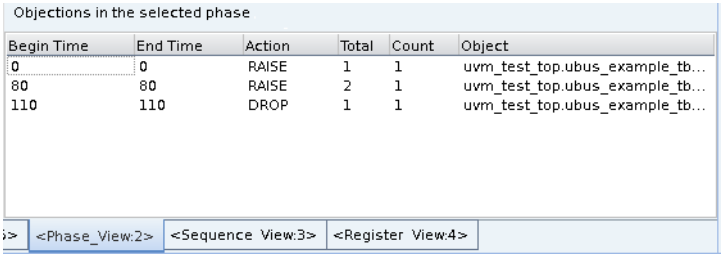


Figure: Objection History Pane

The *Objection History* pane includes the following columns:

- **Begin Time:** Shows the time that the objection is raised by a component object.
- **End Time:** Shows the time that the objection is propagated to the top scope.
- **Action:** Shows the action of the objection.
- **Total:** Shows the total number of objections that are currently observed by the top scope.
- **Count:** Shows the number of objections raised by an object.
- **Object:** Shows the path of the source object.

The **Quick Filter** is enabled by default. The attributes of an item selected in the *Objection History* pane are shown. Choose the desired attributes as the filtering input and click the **Filter** button to show the results in the *Objection History* pane.

The *Quick Filter* pane includes the following columns:

- **Begin Time:** Enter the desired begin time in this field.
- **End Time:** Enter the desired end time in this field.
- **Action:** Enter the desired action in this field.

- **Total:** Enter the desired total number of objections that are currently observed by the top scope in this field.
- **Count:** Enter the desired number of objections raised by an object in this field.
- **Object:** Enter the path of the desired source object in this field.
- **Syntax:** Select a value from **Fixed String**, **Reg Exp**, and **Wildcard**.

- **Phase History:** Shows the status history of the phase selected in the *Phase View* pane.

  For the phase history, the data is retrieved form the FSDB file.

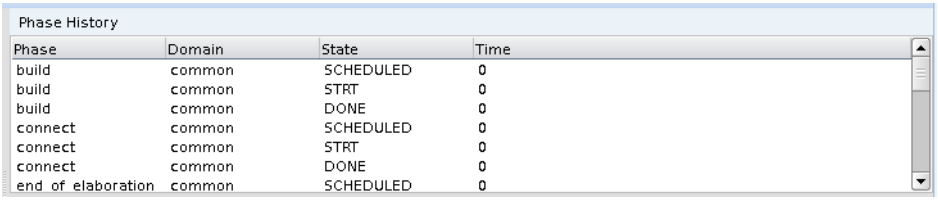


Figure: Phase History Pane

The *Phase History* pane includes the following columns:

- **Phase:** Shows the phase name.
- **Domain:** Shows the domain of the phase.
- **State:** Shows the state of the phase.
- **Time:** Shows the time of the phase.

Click the **Show Quick Filter** icon to open the *Quick Filter* pane where the attributes of an item selected in the *Phase History* pane are shown. Choose the desired attributes as the filtering input and click the **Filter** button to show the results in the *Phase History* pane. The default for the **Show Quick Filter** option is *off*.

The *Quick Filter* pane includes the following columns:

- **Phase:** Enter the desired phase in this field.
- **Domain:** Enter the desired domain name in this field.
- **State:** Select a value from **DONE**, **SKIP**, **START**, **NA**, **N/A**, **STOPPED**, **FINISHED**, **JUMP**, **TIMEOUT**, **WAITPRED OF SUCC**, **DORMANT** and **TO WAIT**.
- **Time:** Enter the desired time in this field.
- **Syntax:** Select a value from **Fixed String**, **Reg Exp**, and **Wildcard**.

View Options

Show Phase History

Menu Bar: View -> Show Phase History

Toolbar Icon: 

This option shows the history of all executed phases and the executing phase. Select a phase in the *Phase View* pane and rows with the information of the selected phase is highlighted in the *Phase History* pane. The domain, state, and time of this phase can be checked in the *Phase History* pane.

Trace Options

Manage Breakpoints

Menu Bar: Trace -> Manage Breakpoints

Toolbar Icon: 

This option opens the *Manage Breakpoints* form where breakpoints can be edited. Refer to the **Manage Breakpoints** option in the *Interactive Simulation Debug* chapter for details.

Set Breakpoint for Phase

Menu Bar: Trace -> Set Breakpoint for Phase

Toolbar Icon: 

NOTE: This option or icon is available when a phase is selected in the *Phase View* pane.

This option opens the *Manage Breakpoints* form where a breakpoint can be set for the selected phase.

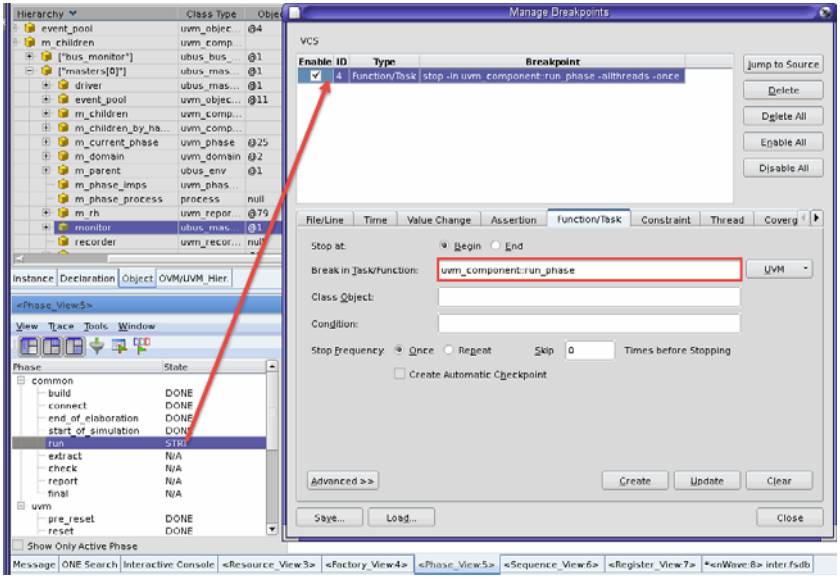


Figure: Manage Breakpoints Form for Phase Settings

The corresponding phase function of the selected phase is entered in the **Break in Task/Function** field automatically and the breakpoint is also added on clicking the **Create** button.

Set Checkpoint for Phase

Menu Bar: Trace -> Set Checkpoint for Phase

NOTE: This option or icon is available when a phase is selected in the *Phase View* pane.

This option opens the *Manage Breakpoints* form where a breakpoint for the selected phase can be added by enabling the **Create Automatic Checkpoint** option.

Show Source

Menu Bar: Trace -> Show Source

NOTE: This option is enabled after an object is selected in the *Objection History* pane.

This option shows the source code of the selected item in the *Source Code* pane. Double-click a selected item can also show the source code.

Show Object in Object Browser

Menu Bar: Trace -> Show Object in Object Browser

NOTE: This option is enabled after an object is selected in the *Objection History* pane.

This option shows the selected objection in the *Object Browser* pane.

Show Object in Class Browser

Menu Bar: Trace -> Show Object in Class Browser

NOTE: This option is enabled after an object is selected in the *Objection History* pane.

This option shows the selected objection in the *Class Browser* pane.

Show Object in Source Window

Menu Bar: Trace -> Show Object in Source Window

NOTE: This option is enabled after an object is selected in the *Objection History* pane.

This option shows the declaration of the selected objection in the source code pane.

Phase View Pane Right-click Options

Manage Breakpoints

Refer to the **Manage Breakpoints** option under the **Trace** options for details.

Set Breakpoint for Phase

Refer to the **Set Breakpoint for Phase** option under the **Trace** options for details.

Set Checkpoint for Phase

Refer to the **Set Checkpoint for Phase** option under the **Trace** options for details.

Objection History Pane Right-click Options

Show Source

Refer to the **Set Checkpoint for Phase** option under the **Trace** options for details.

Show Object in Object Browser

Refer to the **Show Object in Object Browser** option under the **Trace** options for details.

Show Object in Class Browser

Refer to the **Show Object in Class Browser** option under the **Trace** options for details.

Show Object in Source Window

Refer to the **Show Object in Object Browser** option under the **Trace** options for details.

Sequence View Pane

The *Sequence View* pane shows data related to the sequences and sequence items in the UVM flow. The sequences are displayed in a tree structure that represents the parent/child relationship of sequences and sequence items.

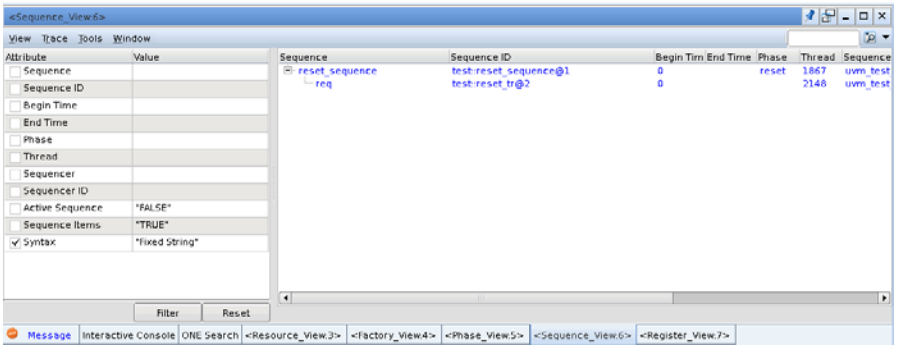


Figure: Sequence View Pane

This *Sequence View* pane contains the following columns (the data in each column can be sorted):

- **Sequence:** Shows the list of sequence names.
- **Sequence ID:** Shows the object ID of a sequence.
- **Start Time:** Shows the start time of a sequence.
- **Finish Time:** Shows the end time of a sequence.
- **Phase:** Shows the phase name and the domain of a sequence.
- **Thread:** Shows the thread executing the sequence.

NOTE: The **Thread** column is only available for data that is still in the simulator. If the sequence no longer exists in the simulator, the thread number may not be seen.

- **Sequencer:** Shows the sequencer that initiated the sequence.
- **Sequencer ID:** Shows the object ID of a sequencer.

NOTE: The **Sequence ID**, **Sequencer ID**, and **Phase** columns require previous sequence data from the Verdi library. This data is currently available for VCS. If the data still exists in the simulator, the data is retrieved. If the data does not exist in the simulator, the Verdi UVM library is required.

View Option

Show Quick Filter

Menu Bar: View -> Show Quick Filter

This option opens the *Quick Filter* pane where the attributes of an item selected in the *Sequence View* pane are shown. Choose the desired attributes as the filtering input and click the **Filter** button to show the results in the *Sequence View* pane. The default for the **Show Quick Filter** option is *on*.

The *Quick Filter* pane includes the following columns:

- **Sequence:** Enter the desired sequence name in this field.
- **Sequence ID:** Enter the desired sequence ID in this field.
- **Start Time:** Enter the desired start time of a sequence in this field.
- **Finish Time:** Enter the desired end time of a sequence in this field.
- **Phase:** Enter the desired phase of a sequence in this field.
- **Thread:** Enter the desired thread executing the sequence in this field.
- **Sequencer:** Enter the desired sequencer that initiated the sequence in this field.
- **Sequencer ID:** Enter the object ID of a sequencer in this field.
- **Active Sequence:** Select the **TRUE** value to view all the sequences in the simulator at the current simulation time. The default is **FALSE**.
- **Sequence Items:** Select the **TRUE** value to view all sequence items in the design. The default is **FALSE**.
- **Syntax:** Select a value from **Fixed String**, **Reg Exp**, and **Wildcard**. The default is **Fixed String**.

Trace Options

Show Class Definition in Source Window

Menu Bar: Trace -> Show Class Definition in Source Window

This option displays the class definition of the selected sequence in the *Source Code* pane. Double-click a sequence also shows its class definition in the *Source Code* pane.

Show Invocation in Source Window

Menu Bar: Trace -> Show Invocation in Source Window

NOTE: This option is enabled after the following conditions are both met:
1. The **+UVM_TR_RECORD** runtime option is specified in the **Simulation** page of the *Preferences* form.
2. The selected sequence is finished.

This option displays the source code where the selected sequence is started in the *Source Code* pane.

Show Sequence in Class Browser

Menu Bar: Trace -> Show Sequence in Class Browser

This option displays the class instance of the selected sequence in the *Class Browser* pane.

Show Sequencer in Class Browser

Menu Bar: Trace -> Show Sequencer in Class Browser

This option displays the class instance of the selected sequencer in the *Class Browser* pane.

Show Sequence in Stack Pane

Menu Bar: Trace -> Show Sequence in Stack Pane

This option displays the execution call stack of the selected sequence in the *Stack* pane.

View Reference Path for Sequence

Menu Bar: Trace -> View Reference Path for Sequence

This option opens the *References* form where the current references of the selected sequence are displayed.

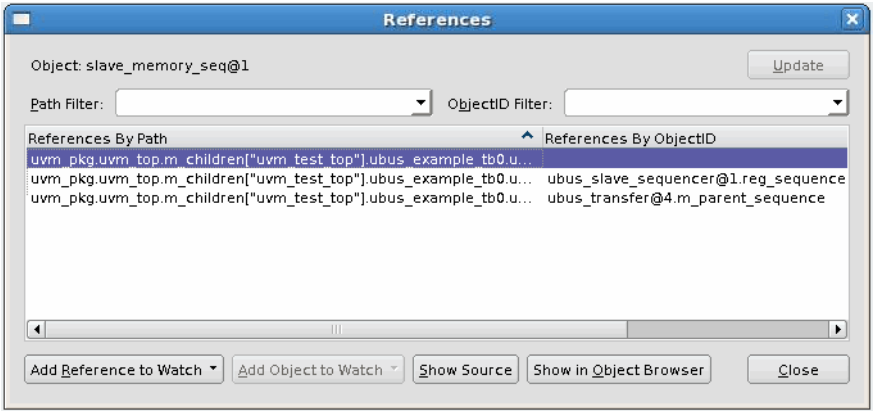


Figure: References Form

The *References* form includes the following fields and options:

- **Path Filter:** Enter the string to view the desired filter path(s) in the **References by Path** column.
- **Object ID Filter:** Enter the string to view the desired object(s) in the **References by Object ID** column.
- **Add Reference to Watch:** Click this button to add the selected reference to the specified **Watch** Tab.
- **Add Object to Watch:** Click this button to add the selected object to the specified **Watch** tab.

NOTE: This option is enabled when a sequence with an object is selected.

- **Show Source:** Click this button to add the selected reference to the source code pane.
- **Show in Object Browser:** Click this button to add the selected reference to the *Object Browser* pane.

Set Breakpoint on Sequence Method

Menu Bar: Trace -> Set Breakpoint on Sequence Method

This option opens the *Manage Breakpoints* form where a breakpoint on the specified sequence method can be created. The available sequence methods include: pre_start, start, post_start, start_item, pre_body, body, post_body, pre_do, mid_do, and post_do.

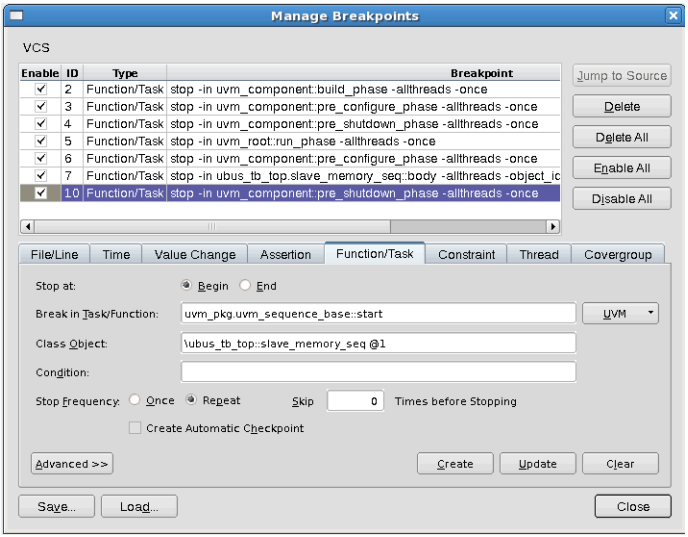


Figure: Manage Breakpoints Form for Sequence Settings

The corresponding sequence task of the selected sequence is entered in the **Break in Task/Function** field automatically. The selected Sequence ID is also entered in **Class Object** field automatically.

Add Sequence to Watches

Menu Bar: Trace -> Add Sequence to Watches

This option adds the selected sequence object to the specified **Watch** tab.

Add Sequencer to Watches

Menu Bar: Trace -> Add Sequencer to Watches

This option adds the selected sequencer object to the specified **Watch** tab.

Add Sequence to Waveform

Menu Bar: Trace -> Add Sequence to Waveform

This option adds the selected sequence object to the *nWave* window.

Add Sequence to Transaction Browser

Menu Bar: Trace -> Add Sequence to Transaction Browser

This option adds the selected sequence object to the *Transaction and Protocol Analyzer* pane. However, if the `INVOKE_UTB_BROWSER` environment variable is set to 0, this command adds the selected sequence object to the *Transaction Browser.*

Sequence View Pane Right-click Options

Show Class Definition in Source Window

Refer to the **Show Class Definition in Source Window** option under the **Trace** options for details.

Show Invocation in Source Window

Refer to the **Show Invocation in Source Window** option under the **Trace** options for details.

Show Sequence in Class Browser

Refer to the **Show Sequence in Class Browser** option under the **Trace** options for details.

Show Sequencer in Class Browser

Refer to the **Show Sequencer in Class Browser** option under the **Trace** options for details.

Show Sequence in Stack Pane

Refer to the **Show Sequence in Stack Pane** option under the **Trace** options for details.

View Reference Path for Sequence

Refer to the **View Reference Path for Sequence** option under the **Trace** options for details.

Set Breakpoint on Sequence Method

Refer to the **Set Breakpoint on Sequence Method** option under the **Trace** options for details.

Add Sequence to Watches

Refer to the **Add Sequence to Watches** option under the **Trace** options for details.

Add Sequencer to Watches

Refer to the **Add Sequencer to Watches** option under the **Trace** options for details.

Add Sequence to Waveform

Refer to the **Add Sequence to Waveform** option under the **Trace** options for details.

Add Sequence to Transaction Browser

Refer to the **Add Sequence to Transaction Browser** option under the **Trace** options for details.

Register View Pane

The *Register View* pane provides an interface to focus on the register modeling at the abstract level and how the register operations (for example, read and write) affect their values. The register hierarchy of each block, the attributes of each corresponding register/memory, and the history of the register access are displayed in the *Register View* pane in the following figure.

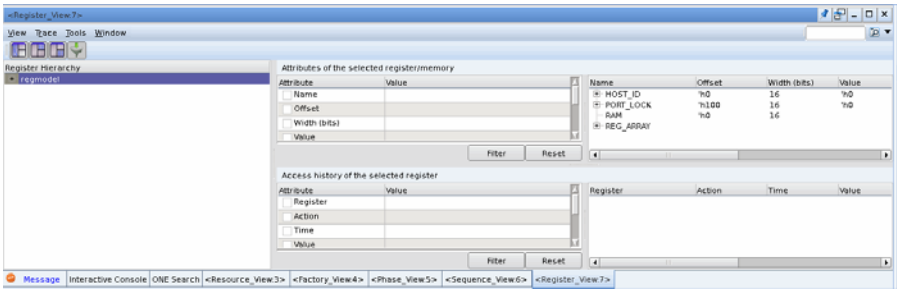


Figure: Register View Pane

The *Register View* pane includes the following panes:

- **Register Hierarchy:** Shows the entire register hierarchy tree.

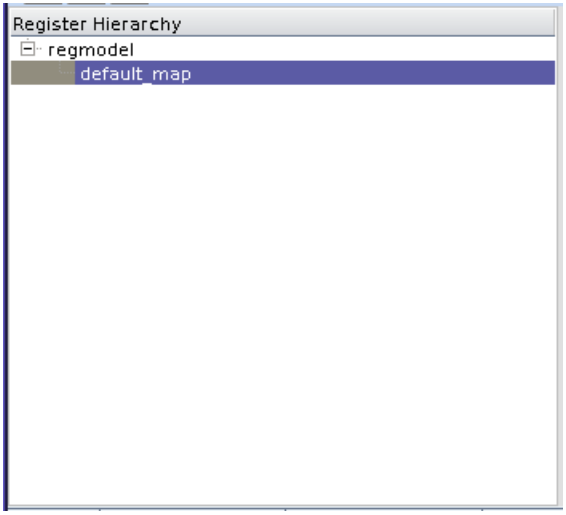


Figure: Register Hierarchy Pane

- **Register Attribute:** Shows the register models which are the first-level children of the selected register model.

The *Register Attribute* pane includes the following columns:

- **Name:** Shows the name of the register
- **Offset:** Shows the offset value
- **Width (bits):** Shows the value of the width in bits
- **Value:** Shows the value in the register
- **Desired:** Shows the desired value expected in the register
- **Mirrored:** Shows the mirrored value of the register
- **Access:** Shows the access rights

The **Quick Filter** option opens the *Quick Filter* pane where the attributes of an item selected in the *Register Attribute* pane is shown. Select the desired attributes as the filtering input and click the **Filter** button to view the results in the *Register Attribute* pane.

- **Attribute:** Shows the Attribute name of the register
- **Value:** Shows the Attribute value of the register

The *Quick Filter* pane includes the following columns:

- **Name:** Shows the name of the register
- **Offset:** Shows the offset of the register
- **Width (bits):** Shows the width of the register
- **Value:** Shows the value in the register
- **Desired:** Shows the desired value expected in the register
- **Mirrored:** Shows the mirrored value of the register
- **Access:** Shows the access rights
- **Value vs. Desired Value:** Select the Equal value to view all register items in the design. The default is UnEqual
- **Syntax:** Select a value from Fixed String, Reg Exp, and Wildcard. The default is Fixed String.

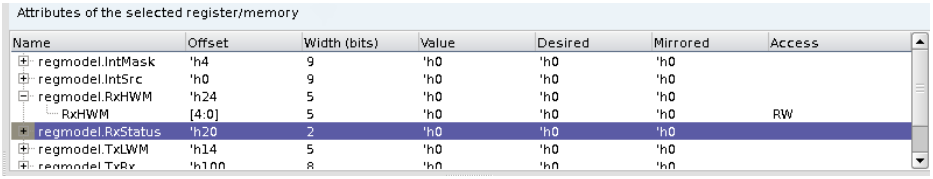


Figure: Register Attribute Pane

- **Register Access History**

Shows the accessed history of the selected register as shown in the following figure.

The *Register Access History* pane includes the following columns:

- **Register**: Shows the name of the register
- **Action:** Shows the READ/WRITE access to the register
- **Time:** Shows the access time to the register
- **Value:** Shows the value in the register
- **Desired:** Shows the desired value expected in the register
- **Mirrored:** Shows the mirrored value of the register

Figure: Show Register Access History Pane

Click the **Show Quick Filter** option to open the *Quick Filter* pane where the attributes of an item selected in the *Register Access History* pane is shown. Select the desired attributes as the filtering input and click the **Filter** button to view the results in the *Register Access History* pane.

The *Quick Filter* pane has the following columns**:**

- **Register:** Shows the name of the register
- **Action:** Shows the READ/WRITE access to the register
- **Time:** Shows the access time to the register
- **Value:** Shows the value in the register
- **Desired:** Shows the desired value expected in the register
- **Mirrored:** Shows the mirrored value of the register
- **Active Objection Only:** TRUE or FALSE
- **Syntax:** Select a value from Fixed String, Reg Exp, and Wildcard. The default is Fixed String.

NOTE: The *Register Access History* pane gets updated automatically when there is a change in the *Register Hierarchy* pane.

View Options

Show Register Hierarchy

Menu Bar: View -> Show Register Hierarchy

Toolbar Icon: 

This option displays the register hierarchy of each block.

Show Register Attribute

Menu Bar: View -> Show Register Attribute

Toolbar Icon: 

This option displays the register models which are the first-level children of the selected register model in the *Register Attribute* pane.

Show Register Access History

Menu Bar: View -> Show Register Access History

Toolbar Icon: 

This option displays the accessed history of the selected register in the *Register Access History* pane.

Change Desired Value

Menu Bar: View -> Change Desired Value

NOTE: This option is available when you select a register attribute under a register name in the *Register Attribute* pane.

This option forces the desired value of the selected register mode and the **Force Signal Value** form.

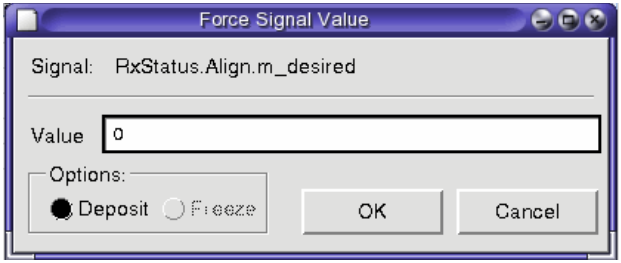


Figure: Force Signal Value

Set Radix

Menu Bar: View -> Set Radix

NOTE: This option is available when you select a register attribute under a register name in the *Register Attribute* pane.

This option checks the different radixes for different register models. The options available are: Binary, Octal, Decimal, and Hexadecimal.

Trace Options

Show Definition

Menu Bar: Trace -> Show Definition

This option displays the definition of the selected register model in the source code pane. Double-clicking an item has the same effect.

Show Class Definition

Menu Bar: Trace -> Show Class Definition

This option displays the class declaration of the selected register model in the source code pane.

Show in Object Browser

Menu Bar: Trace -> Show Object Browser

This option displays the selected register model in the *Object Browser* pane.

Show in Class Browser

Menu Bar: Trace -> Show Class Browser

This option displays the selected register model in the *Class Browser* pane

Show HDL Path

Menu Bar: Trace -> Show HDL Path

This option displays the related HDL of DUT in the source code pane. In the message pane, a warning message is displayed if the HDL path cannot be found for the register model.

Add to Waveform

NOTE: This option is available when you select a register attribute in the *Register Attribute* pane.

Menu Bar: Trace -> Add to Waveform

This option invokes *nWave* and display the waveform of the register attribute.

New Waveform

This command enables you to create a new waveform window, open an annotation file, and add signals.

Add to Wave[n]

Bind Key Ctrl+4

This command enables you to add selected objects to the last adding signal waveform or the new created waveform.

Add to New Group

This command creates a new group in the waveform and adds selected objects to this group.

*Wave[k]

Bind Key Ctrl+W

This command adds selected objects to the waveform window. For example, *Wave[k], k is the primary waveform window Id.

Wave[m]

This command adds selected objects to waveform window. For example, Wave[m], m is the window id.

Set Break Point

Menu Bar: Trace -> Set Break Point

This option creates a breakpoint on a register model and specify the condition for the breakpoint. The available values are Value, Desired Value, and Mirrored Value. The available methods are Read, Write, Peek, Poke, Mirror, and Update.

Register Attribute Pane Right-click Options

The following options are found on the right-click menu of the *Register Attribute* pane:

Show Definition

Refer to the **Show Definition** option under the **Trace** options for details.

Show Class Definition

Refer to the **Show Class Definition** option under the **Trace** options for details.

Show in Object Browser

Refer to the **Show Class Definition** option under the **Trace** options for details.

Show in Class Browser

Refer to the **Show in Class Browser** option under the **Trace** options for details.

Show HDL Path

Refer to the **Show HDL Path** option under the **Trace** options for details.

Add to Waveform

Refer to the **Add to Waveform** option under the **Trace** options for details.

Change Desired Value

Refer to the **Change Desired Value** option under the **View** options for details.

Set Break Point

Refer to the **Set Break Point** option under the **Trace** options for details.

Set Radix

Refer to the **Set Radix** option under the **View** options for details.

Register Hierarchy Pane Right-click Options

The following right-click options are found on the *Register Hierarchy* pane.

Show Definition

Refer to the **Show Definition** option under the **Trace** options for details.

Show Class Definition

Refer to the **Show Class Definition** option under the **Trace** options for details.

Show in Object Browser

Refer to the **Show in Object Browser** option under the **Trace** options for details.

Show in Class Browser

Refer to the **Show in Class Browser** option under the **Trace** options for details.

Show HDL Path

Refer to the **Show HDL Path** option under the **Trace** options for details.

Add to Waveform

Refer to the **Add to Waveform** option under the **Trace** options for details.

Coverage Debug

Overview

The Verdi Coverage function in the Verdi® Automated Debug Platform provides visualization for coverage data. After you have loaded the coverage of simulation results for a Verilog or VHDL design, you can easily use the Verdi Coverage capability to check the coverage report in the Verdi platform.

Visualization for coverage data in Verdi Coverage is a comprehensive visualization environment that integrates with the design hierarchy display and can be used to get a summary of the coverage results or details of various types of coverage. Verdi Coverage can display the following coverage information:

- **Code coverage** includes the following:
  - **Line Coverage:** The statements that are executed during simulation.
  - **FSM Coverage:** The blocks of code that make up FSMs are identified. FSM coverage reports on the states, transitions, and sequences of states during simulation.
  - **Toggle Coverage:** Whether the signals in the design toggle from 0 to 1 and 1 to 0 during simulation.
  - **Condition Coverage:** Conditions are expressions and sub-expressions that control the execution of code or the assignment of values to signals. Condition coverage tests whether both true and false states of these conditions are covered during simulation.
  - **Branch Coverage:** Analyzes how if, case statements, and the ternary operator (?:) establish branches of execution in your Verilog design. It shows vectors of signal or expression values that enable or prevent simulation events.
- **Covergroup Coverage:** Coverage of SystemVerilog testbench coverage groups.
- **Assertion Coverage:** Coverage of SystemVerilog cover properties or assert properties. Low Power Coverage: Coverage of UPF low power coverage groups.

Furthermore, you can use the **Exclusion Manager** of Verdi Coverage to show all exclusions and unresolved exclusions and edit exclusions. You can also use Verdi Coverage to create a verification plan or verification plan components (`.hvp`

files) and to allow `.hvp` files to be linked to `.pdf` files. The interaction between the Verdi platform and Verdi Coverage is also provided to debug with designs and the coverage report.

NOTE: Currently, the supported simulator is VCS with version 2013.06 and later.

Invoking Verdi Coverage

To invoke Verdi Coverage, you need to add the `-cov` option to the `verdi` command line. You can also specify the directory name in the command line while invoking Verdi Coverage as follows:

```
> verdi -cov &
> verdi -cov -covdir simv.vdb &
```

NOTE: It is recommended to merge or map multiple `.vdbs` via URG before invoking Verdi. For details on this, see Chapter, "Merging Coverage Metrics" of the *Coverage Technology User Guide*.

To see the command line options for Verdi Coverage, use the following command:

```
> verdi -cov -help
```

Alternatively, you can use the `Tools à Coverage` command in the main window to invoke Verdi Coverage from a previously opened Verdi platform.

NOTE: When the **-**`cov` option is mixed with non-coverage options for the Verdi platform (for example, `verdi -cov -lib work`), Verdi Coverage is invoked. The Verdi platform is not invoked until the **Tools → Debug** command is used in **Verdi Coverage**. Unaffected options are shown in the **Message** frame as a warning message.

If you change the location of a source file, you can map the path to the relocated file using the `-pathmap <pathmap_file>` option. The rule specified in the file relocates source files. For example,

```
> verdi -cov -covdir simv.vdb -pathmap <pathmap_file> &
```

The syntax of the path mapping file is as follows:

```
<old_directory_path_name>:<new_directory_path_name>
```

For example,

```
## the content of the path mapping file
/home/work/src: ~local/files/source
```

X-Propagation Debug

Overview

The SystemVerilog standard defines an X as an unknown that means X can be 1, 0, or Z. This provides flexibility and optimization in simulation. Because the simulator can convert the unknown to a known, indeterminate states can be masked in simulation. Unfortunately, sometimes this conversion leads to a dangerous situation in hardware, such as an unexpected control flow or event trigger. In this case, lot of effort is required by the engineer to find out the cause and tracing back X in gates can be time consuming.

There are many ways to detect or avoid X problems. For example, you can use two-state value type to replace four-state value type or use a lint tool to find the potential issues. The VCS simulator provides an easy and effective way to let you find the x issue by adding the `-xprop` VCS compile-time option. This document describes how to quickly debug VCS simulator results (FSDB file) and a log file (`xprop.log`) in the Verdi platform,.

The Verdi platform provides many functions to support X-Propagation (Xprop) debugging. The **Active Trace** option can be used to trace the X value. Active trace not only supports statement but also supports function evaluation, which means that you can trace an X value if it comes from a function. Due to simulator limitations, the Xprop mechanism is blocked, such as the delay assignment. The Verdi platform provides a way to let you check the blocked section in the *source code* pane and quickly understand the flow in the design.

Requirements

To debug Xprop in the Verdi platform, you need the following data:

- The FSDB file generated from VCS version 2013.06 or higher. The Xprop version should be 1.3.
- The Xprop log file (`xprop.log`) generated from VCS.

In addition, you need to compile the design in the Verdi platform using the `-xprop` and `-xproplog` options.

NOTE: Because the `xprop.log` file records relative file names, you should compile the design for VCS and the Verdi platform in the same path, so that the correct files are found.

Usage

Command Line Options

You can use the following Xprop compile-time options on the Verdi command line:

- `-xprop= [ tmerge | xmerge | <configfile> ]`
  The option specifies the Xprop format. The default is `tmerge`.
- `-xproplog <xprop.log> <xprop_vhdl.log>`
  The option specifies the Xprop log file.

For example:

```
verdi -f run.f -ssf myfile.fsdb -xprop=xmerge
-xproplog xprop.log
```

NOTE: If the `tmerge` and `xmerge` arguments are specified multiple times for `-xprop` (in any order), only the last specification is recognized and takes effect. If the `configfile` argument is specified with any other argument, an error is generated.

Source Code Functions

In the Verdi platform, the **Active Trace** option and the *source code* pane are extended to support the Xprop information.

In the main window, the **View** -> **X-propagation Instrument** toggle option is added to mark the information in the source code according to the Xprop file. In addition, a new **Xprop** page and a new **Show X-propagation Instrument Information** option is added to the *Preferences* form (invoked with **Tools** -> **Preferences** menu option).

When the toggle option is turned on, colors are added to the *source code* pane to represent Xprop information.

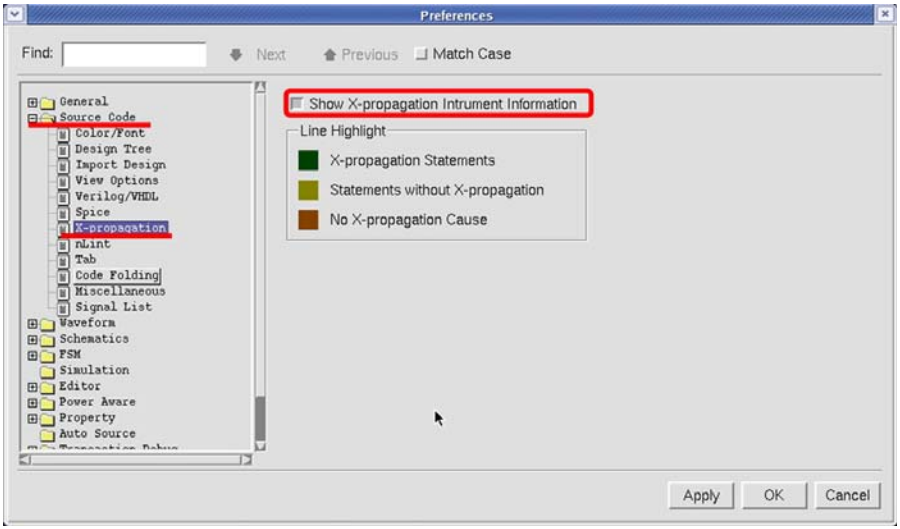


Figure: Preferences, Source Code Folder, Code Folding Page

The following figure shows an example of the colorization. Only the lines blocking the Xprop are highlighted in the *source code* pane. The other colors are shown in the line number column only.

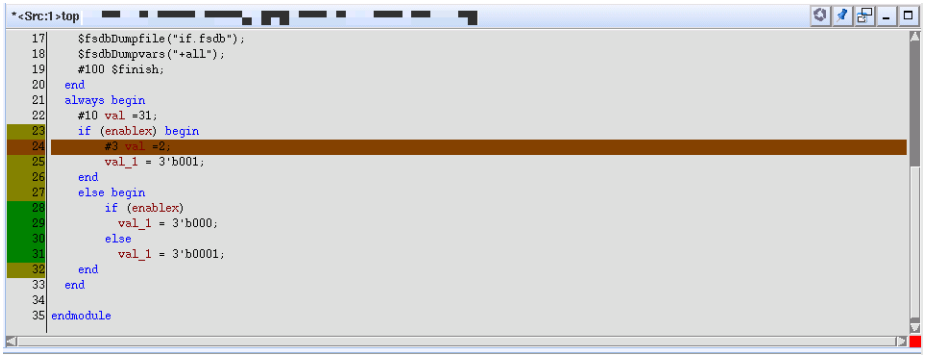


Figure: Colorization in Source Code

NOTE: The **View -> Line Number** toggle menu option or the **Line Number** option on the **View Options** page in the **Source Code** folder of the *Preferences* form must be turned on when the Xprop is turned on.

Active Trace

Active Trace indicates the actual driver in the current time and shows the result in the **Trace** tab of the *Message* pane. In order to distinguish the original active trace results from the Xprop results, a `T` suffix is used to indicate that the result is from `tmerge` while an `X` suffix is used to indicate that the result is from `xmerge`.

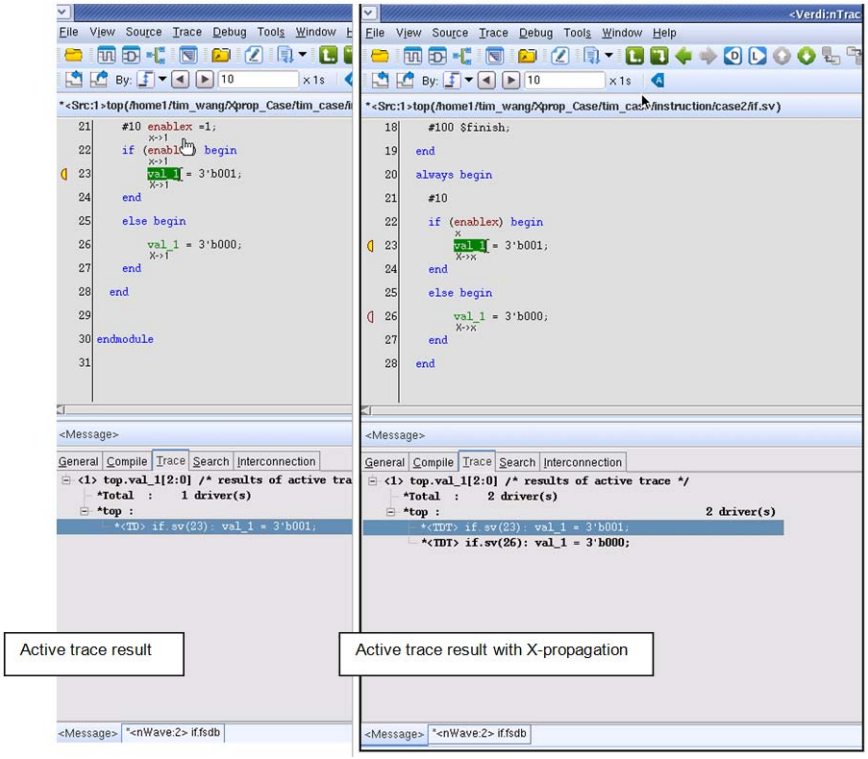


Figure: Active Trace Results Differences

Message Pane

The Xprop log file read by the Verdi platform is available on the new **X-Propagation** tab in the *Message* pane. Depending on the size of the log file, it might take some time for the log file to be displayed. After the log file is displayed, click a message to go to the corresponding file and line in the *source code* pane.

The **X-propagation** tab includes a search function that helps you o quickly locate the file instrument information.

Figure: Search Function

Temporal Flow View

The *Temporal Flow View* also supports the Xprop results by following the instrumented source code when the `xprop.log` file is loaded. This eliminates any mismatches between the FSDB file and the original source code.

Instance Tree

In the *Instance Tree* pane, instances that are applied with Xprop instrument are shown with the  icon.

You can turn on or turn off the function by selecting the **View** -> **X-propagation Instrument** menu option. The default is *off*.

FFT Pane

Overview

The *FFT* pane is displayed when the **Analog -> FFT** option is invoked in *nWave* window. The *FFT* pane is docked to the right of the source code frame. Click the **Undock** icon on the toolbar to make *FFT* pane a standalone window.

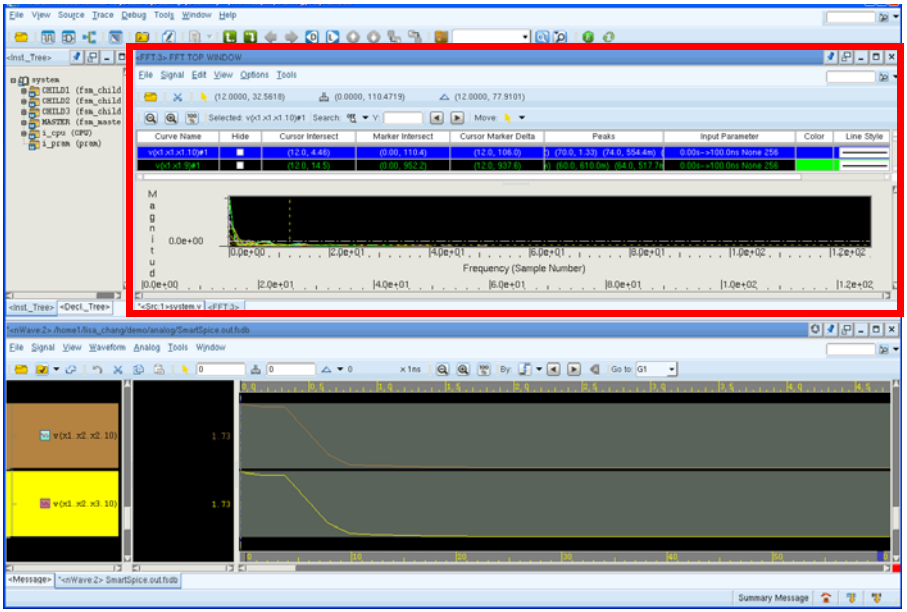


Figure: FFT Frame

The **Curve** pane (located in the lower section of the *FFT* pane) displays the FFT curves after adding FFT signal(s) to the *FFT* pane by invoking the **File -> Restore Signals** or the **Signal -> Add FFT** signals menu option, as illustrated in the following figure.

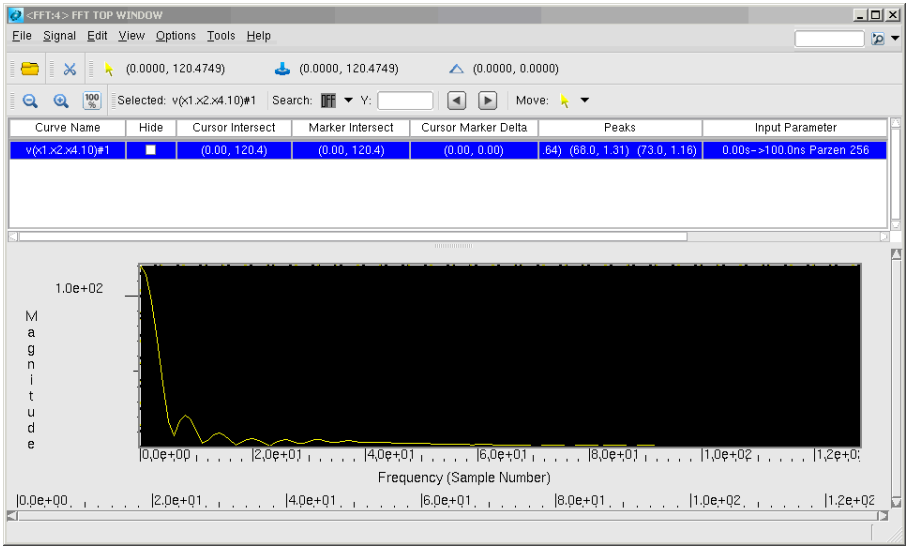


Figure: FFT Curves Displayed in the Curve Pane

The **Legend** pane, located in the upper section of the *FFT* pane, includes nine standard columns: **Curve Name**, **Hide**, **Cursor Intersect**, **Marker Intersect**, **Cursor Marker Delta**, **Peaks**, **Input Parameter**, **Color**, and **Line Style**. The width of each column can be adjusted by selecting the vertical line between column headers and dragging-left. The standard column headings are summarized as follows:

- **Curve Name:** Indicates the FFT signal name. **Curve Name** includes signal name and index (such as `A#1`). A new FFT curve calculated from analog signal A is displayed as `A#1` in the **Curve Name** column in the **Legend** pane. If more than one FFT curve is generated from analog signal A, the index part of the **Curve Name** increases accordingly (such as `A#2`, and so on).
- **Hide:** The check indicates if the selected FFT curve is to be shown in the **Curve** pane.
- **Cursor Intersect:** Indicates the intersecting points of the cursor X-dimension value and the FFT curve.
- **Marker Intersect:** Indicates the intersecting point of the marker X-dimension and the FFT curve.
- **Cursor Mark Delta:** Indicates the delta value of the **Cursor Intersect** point and the **Marker Intersect** point.
- **Peaks:** Indicates the peak points of the FFT curve.

- **Input Parameters:** Indicates the information specified in the *FFT Input Parameters* form. For example, `10.6ns->24.0ns Hanning 1024` represents the start time, stop time, weighted type, and sampling points.
- **Color:** Displays the color of the curve.
- **Line Style:** Displays the line style of the curve.

The menu bars for the *FFT* pane are as shown in the following figure:

Figure: FFT Pane Menu Bar

The menus are summarized and explained on the following pages. Refer to the *Icons for Dockable Panes* section for details on the menu bar icons.

Menu Summary

The *FFT* pane menu options are summarized in the following sections.To jump to the corresponding option, double-click all options description. For right-click options, refer to the Right-click Options section for details.

File Menu Options

Save Signals
Restore Signals
Output Data to a File
Print
Close Window

Signal Menu Options

Add FFT Signals
Recalculate
Recalculate and Append

Edit Menu Options

Calculate Selected
Cut
Hide Selected
Show Hidden
Change Color/Pattern

View Menu Options

Zoom In X
Zoom In Y
Zoom In X/Y
Zoom Out X
Zoom Out Y
Zoom Out X/Y
Zoom All X
Zoom All Y
Zoom All X/Y
Zoom Area
Pan Left
Pan Right
Pan Up
Pan Down
Go To
Last View

Options Menu Options

Snap
Preferences

Tools Menu Option

Customize Menu/Toolbar

File Menu Options

Save Signals

Menu Bar: **File -> Save Signals**

This option saves the current FFT signals in a file in the *Save FFT Signals* form. It is recommended to save the file with the `fft` file extension (such as `signal.fft`).

Restore Signals

Menu Bar: **File -> Restore Signals**

Toolbar Icon: 

This option restores the signals from an FFT file saved previously by the **Save Signals** option in the *Restore FFT Signals* form where the FFT curves file can be selected. The default file filter is **.fft*.

.

Output Data to a File

Menu Bar: **File -> Output Data to a File**

This option saves the current **Curve** pane information, including the curve name, start time, stop time, weighted type, sample point, and all points. When the **Peaks Only** option is turned on, peaks instead of all points is shown on the file. The sample output data file is as follows:

```
Curve Name : I(vro5#1
Start time(s) : 0.00
Stop time(s) : 10.0n
Weighted Type : None
Sample Point : 256

Peaks
-----------------------------------
Frequency(Hz)    : Magnitude
-----------------------------------
8.00             : 2.26p
26.0             : 1.24p
```

```
76.0              : 125.2f
102.0             : 44.7f
124.0             : 68.2f

Curve Name : I(vvcc#1
Start time(s) : 0.00
Stop time(s) : 10.0n
Weighted Type : None
Sample Point : 256

Peaks
-----------------------------------
Frequency(Hz)     : Magnitude
-----------------------------------
100.0             : 26.6p
```

Print

Menu Bar: **File -> Print**

This option opens the *Print* form where print options, paper, and print destination can be specified.

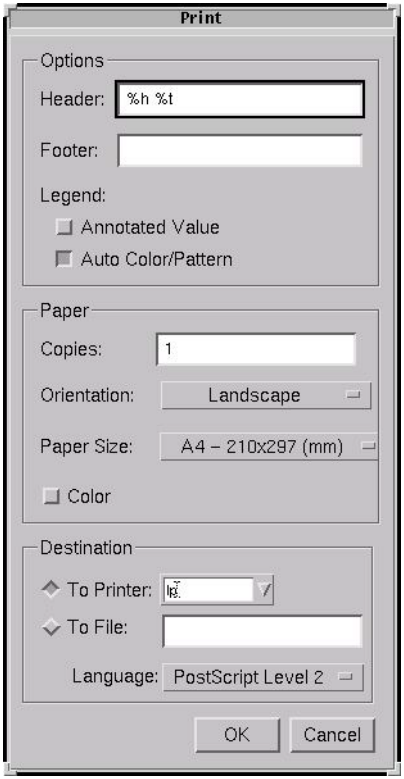


Figure: Print Form

The **Options** section includes the following options and fields:

- **Header:** Specify the header in the text field. The default macro, *%h %t* can be used for the header.
- **Footer:** Specify the footer in the text field.
- **Annotated Value:** When this option is turned *on*, the printout shows the values of **Cursor Value**, **Marker Value**, **Input Parameters**, **Cursor Intersect**, **Marker Intersect**, and **Cursor Marker Delta** for each curve.
- **Auto Color/Pattern:** When this option is turned *on*, the FFT curves in the **Curve** pane are assigned with different line styles in the printout.

The **Paper** section includes the following options and fields:

- **Copies:** Enter the number of copies to be printed.
- **Orientation:** Select the page orientation from the **Landscape** option or the **Portrait** option.
- **Paper Size:** Select one of the supported paper sizes of the printer from **Letter**, **A4**, **A3**, **A2**, **A1**, **A0**, **B**, **C**, **D**, **E**, or **User-defined**.

- **Color:** Turn this option *on* to print multiple colors on a color printer. Turn this option off to print in black and white on a color printer.

The **Destination** section includes the following options and fields:

- **To Printer:** When this option is turned *on*, the printed file goes to the specified printer.
- **To File:** Turn this option *on* to save the printed file to the working directory. The file name must be specified in the text field.
- **Language:** Specify the language from the **PostScript Level 2** option or the **PostScript Level 1** option.

Close Window

Menu Bar: **File -> Close Window**

This option closes the current *FFT* pane.

Signal Menu Options

Add FFT Signals

Menu Bar: **Signal -> Add FFT Signals**

This option opens the *FFT Input Parameters* form where analog signals can be added to the *FFT* pane. You can also double-click the column heading row of the *Legend* pane to open the form.

Click the **OK** button to calculate the FFT signals on the selected analog signal. After calculation, the result for the FFT signal appears in the *Legend* and *Curve* panes. Different colors are assigned to each FFT curve.

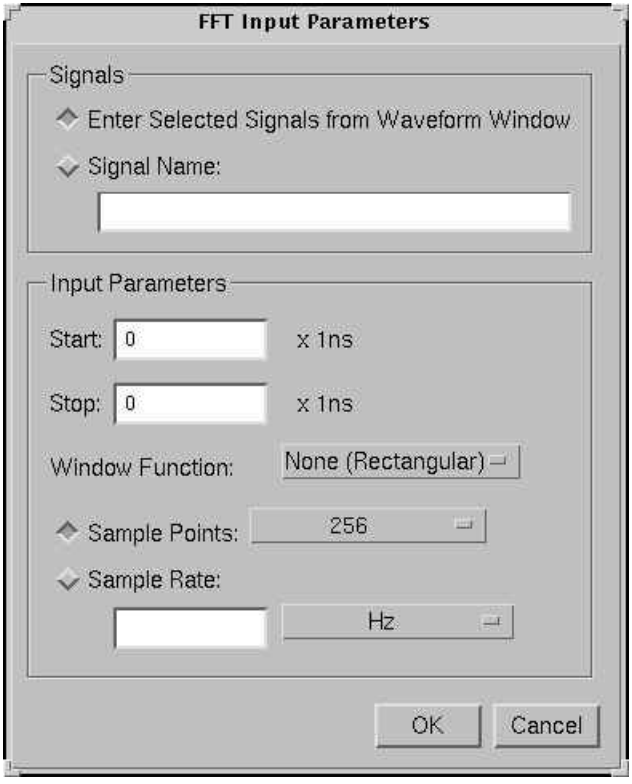


Figure: FFT Input Parameters Form

The **Signals** section includes the following options and field to add analog signal in the *FFT* pane.

- **Enter Selected Signals from Waveform Window:** Select this option to add the analog signals from the selected signals in the signal pane of *nWave* window.
- **Signal Name:** Drag the analog signal from other windows (such as from the signal pane of *nWave* window) or type the signal name in the text field.

The **Input Parameters** section includes the following options and fields:

- **Start:** This parameter sets the begin time of the FFT sample for the set of analog signals. By default, this value is the cursor value in the *nWave* window, but the start value can also be specified manually. When the **Start** and **Stop** values are the same, a warning message "The length of the sampling segment cannot be zero!" is displayed.
- **Stop:** This parameter sets the end time of the FFT sample for the set of analog signals. By default, this value is the marker value in the *nWave* window. The **Start** and **Stop** time change automatically if the cursor or the marker position in the waveform pane of the *nWave* window is updated.
- **Window Function:** Specify an option for the window type by selecting **None (Regular)**, **Blackman**, **Hamming**, **Hanning**, **Parzen**, **Triangular**, or **Welch**.
- **Sample Points:** Specify an option for the number of sample points within the begin/end time segment by selecting **256**, **512**, **1024**, **2048**, **4096**, **8192**, **16384**, **32768**, **65536**, **131072**, **262144**, **524288**, **1048576**, **2097152**, or **4194304**.
- **Sample Rate:** Type in the rate and select a rate unit type of Hz or MHz.

Recalculate

Menu Bar: **Signal -> Recalculate**

This option recalculates the selected FFT curve with the original signal and the same input parameters in the **Legend** pane.

Recalculate and Append

Menu Bar: **Signal -> Recalculate and Append**

This option recalculates the selected FFT curve in the **Legend** pane and appends the results to the existing curves.

Edit Menu Options

Calculate Selected

Menu Bar: **Edit -> Calculate Selected**

This option calculates the selected FFT signal by specifying different input parameters for the selected FFT signal in the *FFT Input Parameters [2]* form. The Input Parameters section is similar to the *FFT Input Parameters* form opened by the **Signal ->** **Add FFT Signals** option. You can also double-click the **Input Parameters** column heading of the **Legend** pane to open the form.

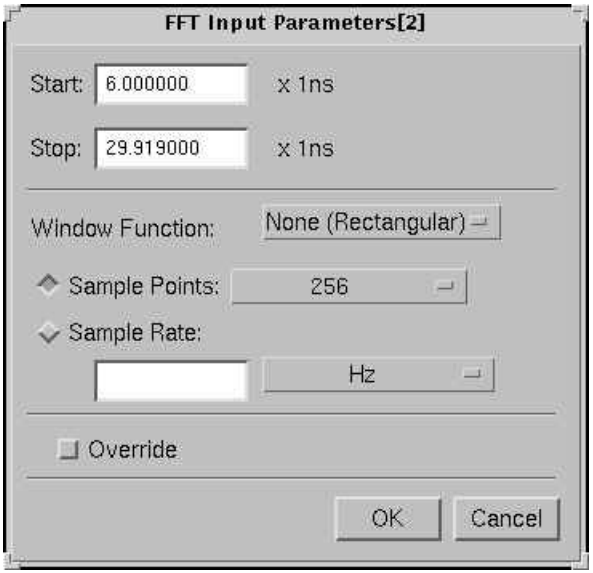


Figure: FFT Input Parameters Form

Cut

Menu Bar: **Edit -> Cut**

Toolbar Icon:

This option removes the selected FFT signal from the **Legend** pane.

Hide Selected

Menu Bar: **Edit -> Hide Selected**

When the **Hide** box is checked in the **Legend** pane, this option hides the selected FFT signal.

Show Hidden

Menu Bar: **Edit -> Show Hidden**

When the **Hide** box is checked in the **Legend** pane, this option shows the selected FFT signal.

Change Color/Pattern

Menu Bar: **Edit -> Change Color/Pattern**

This option changes the color, line width, and line style for the selected FFT signal in the *Change Color/Pattern* form. You can also double-click the **Color** or **Line Style** column heading of each signal in the **Legend** pane to invoke option.

View Menu Options

Zoom In X

Menu Bar: **View -> Zoom In X**

This option provides a close-up view of the content in the X direction in the *FFT* pane. The magnification of the viewing area is changed to half the magnification of the previous view.

Zoom In Y

Menu Bar: **View -> Zoom In Y**

This option provides a close-up view of the content in the Y direction in the *FFT* pane. The magnification of the viewing area is changed to half the magnification of the previous view.

Zoom In X/Y

Menu Bar: **View -> Zoom In X/Y**

Toolbar Icon: 

Bind Key: Shift+Z

This option provides a close-up view of the content in both the X and Y directions in the *FFT* pane. The magnification of the viewing area is changed to half the magnification of the previous view.

Zoom Out X

Menu Bar: **View -> Zoom Out X**

This option enables more of the content to be seen in the X direction in the *FFT* pane at a reduced size. The magnification of the viewing area is changed to two times the magnification of the previous view.

Zoom Out Y

Menu Bar: **View -> Zoom Out Y**

This option enables more of the content to be seen in the Y direction in the *FFT* pane at a reduced size. The magnification of the viewing area is changed to two times the magnification of the previous view.

Zoom Out X/Y

Menu Bar: **View -> Zoom Out X/Y**

Toolbar Icon: 

Bind Key: Z

This option enables more of the content to be seen in both the X and Y directions in the *FFT* pane at a reduced size. The magnification of the viewing area is changed to two times the magnification of the previous view.

Zoom All X

Menu Bar: **View -> Zoom All X**

This option shows the full range of the X-axis in the **Curve** pane.

Zoom All Y

Menu Bar: **View -> Zoom All Y**

This option shows the full range of the Y-axis in the **Curve** pane.

Zoom All X/Y

Menu Bar: **View -> Zoom All X/Y**

Toolbar Icon: 

Bind Key: F

This option shows the full range of both the Y-axis and X-axis in the **Curve** pane.

Zoom Area

Menu Bar: **View -> Zoom Area**

Toolbar Icon: 

This option zooms the area between the **Cursor Intersect** and the **Marker Intersect**. The bracket next to the  icon indicates the difference between the **Marker Intersect** and the **Cursor Intersect** (**Marker Intersect - Cursor Intersect**) (X-axis, Y-axis) in the **Curve** pane. If the area between the cursor and marker is too small, this option does not work.

Pan Left

Menu Bar: **View -> Pan Left**

Bind Key: Left Arrow

This option moves the **Curve** pane viewing area to the left.

Pan Right

Menu Bar: **View -> Pan Right**

Bind Key: Right Arrow

This option moves the **Curve** pane viewing area to the right.

NOTE: Moving the horizontal scroll bar in the **Curve** pane also allows panning left and right.

Pan Up

Menu Bar: **View -> Pan Up**

Bind Key: Up Arrow

This option moves the **Curve** pane viewing area up.

Pan Down

Menu Bar: **View -> Pan Down**

Bind Key: Down Arrow

This option moves the **Curve** pane viewing area down.

NOTE: Moving the vertical scroll bar in the **Curve** pane also allows panning up and down.

Go To

Menu Bar: **View -> Go To**

This option opens the *FFT Go To* form where the value for X and Y can be entered in the **X Value** and **Y Value** text fields to indicate the X/Y value for cursor.

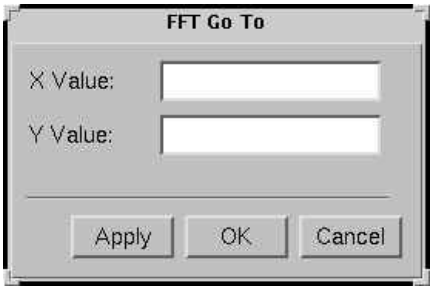


Figure: FFT Go To Form

Last View

Menu Bar: **View -> Last View**

Bind Key: L

This option shows the view of the last waveform pane. This option only displays ONE last viewing. When this option is invoked more than once, it switches between the current view and the last view.

Options Menu Options

Snap

Menu Bar: **Options -> Snap**

When this option is invoked, the cursor or marker always jumps to the closest FFT result point.

Preferences

Menu Bar: **Options -> Preferences**

This option opens the *Preferences* form where display and grid options can be specified.

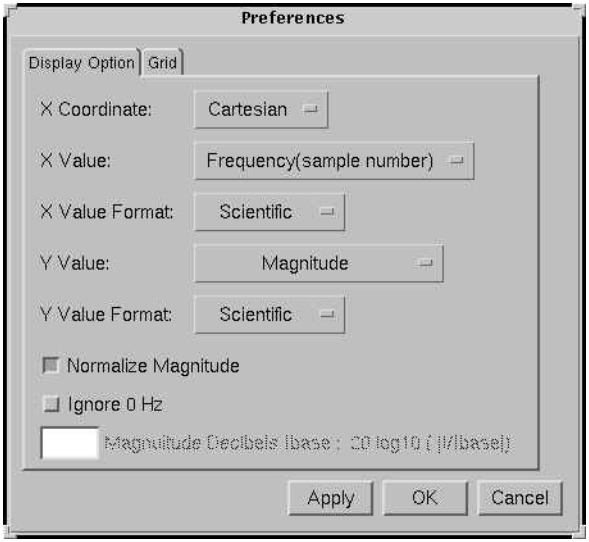


Figure: Preferences Form

Display Option Tab

The **Display Option** tab includes the following options and fields:

- **X Coordinate:** Select an X coordinate in the **Curve** pane from the **Cartesian** option or the **Logarithm** option.

- **X Value:** Specify the X value in the **Curve** pane from the **Frequency (Sample Number)** option or the **Frequency** option.
- **X Value Format:** Specify the X value format from the **Engineering** option or the **Scientific** option.
- **Y Value:** Specify the Y value in the **Curve** pane from the **Magnitude**, **Magnitude in decibels (dB)**, **Real Part**, or **Imaginary Part** options.
- **Y Value Format:** Specify the Y value format from the **Engineering** option or the **Scientific** option.
- **Normal Magnitude:** When this option is turned on, the normalized magnitude is displayed. When this option is turned off, the abnormalized magnitude is displayed.
- **Ignore 0 Hz:** When this option is turned on, the zero Hz component of the FFT result is ignored. When this option is turned off, the zero Hz component is displayed in the spectrum.
- **Magnitude Decibels Ibase: 20 log10 (|I/Ibase|):** When the **Y Value** is set to **Magnitude in decibels (dB),** this option is turned on. The Ibase value can be specified in the text field in front for decibel calculation.

Grid Tab

The **Grid** tab includes the following options and fields:

- **X Grids:** Select this option to set the horizontal grid lines in the background of the **Curve** pane.
- **Y Grids:** Select this option to set the vertical grid lines in the background of the **Curve** pane.

Tools Menu Option

Customize Menu/Toolbar

Menu Bar: **Tools -> Customize Menu/Toolbar**

Refer to the **Tools -> Customize Menu/Toolbar** option description in the *nTrace* chapter for details.

Right-click Options

Many of the previously described options can also be selected from the right-click option in the *FFT* pane.

FFT Pane Right-click Options

When you right-click anywhere in the menu or toolbar icon, a configuration option menu is displayed. This menu can be used to configure the available icons.

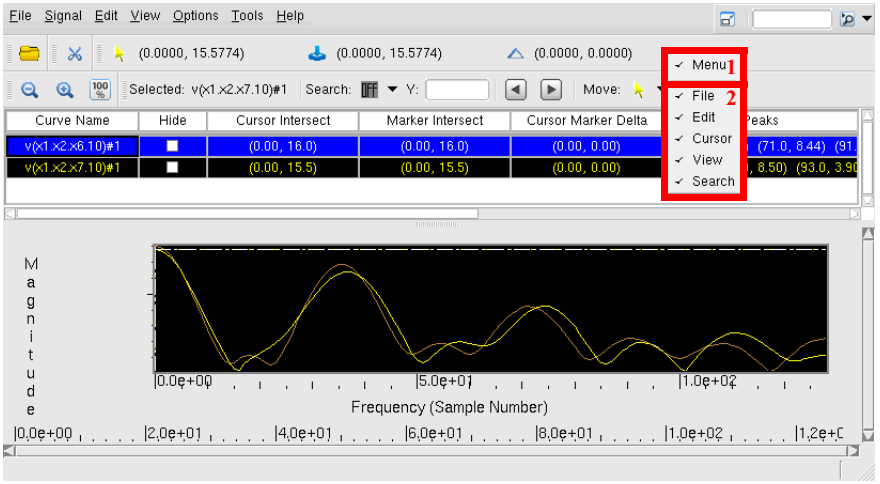


Figure: Configuration Option Menu

The Menu option (labeled 1 in the above figure) enables or disables the option menu bar. The options in the bottom section (labeled 2 in the above figure) enable or disable the toolbar icons for different functions.

Curve Pane Right-click Options

The right-click options in the **Curve** pane include the following:

Print

Refer to the **File -> Print** option description for details.

View

The **View** option includes the following suboptions:

- **Zoom In X**
  Refer to the **Zoom In X** option description for details.
- **Zoom In Y**
  Refer to the **Zoom In Y** option description for details.
- **Zoom In X/Y**
  Refer to the **Zoom In X/Y** option description for details.
- **Zoom Out X**
  Refer to the **Zoom Out X** option description for details.
- **Zoom Out Y**
  Refer to the **Zoom Out Y** option description for details.
- **Zoom Out X/Y**
  Refer to the **Zoom Out X/Y** option description for details.
- **Zoom All X**
  Refer to the **Zoom All X** option description for details.
- **Zoom All Y**
  Refer to the **Zoom All Y** option description for details.
- **Zoom All X/Y**
  Refer to the **Zoom All X/Y** option description for details.
- **Zoom All X**
  Refer to the **Zoom All X** option description for details.
- **Zoom All Y**
  Refer to the **Zoom All Y** option description for details.
- **Zoom All X/Y**
  Refer to the **Zoom All X/Y** option description for details.

- **Zoom Area**
  Refer to the **Zoom Area** option description for details.
- **Pan Left**
  Refer to the **Pan Left** option description for details.
- **Pan Right**
  Refer to the **Pan Right** option description for details.
- **Pan Up**
  Refer to the **Pan Up** option description for details.
- **Pan Down**
  Refer to the **Pan Down** option description for details.
- **Last View**
  Refer to the **Last View** option for details

Toolbar Icons and Fields

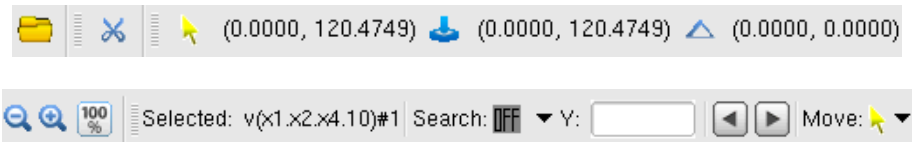


Figure: Toolbar Used in FFT Pane

The available toolbar icons may be modified. Refer to the *Toolbars* section of the *User Interface* chapter in the *Verdi User Guide and Tutorial* manual for details.

File Category

Restore Signals 

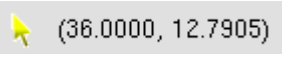
Refer to the **File -> Restore Signals** option description for details.

Edit Category

Cut 

Refer to the **Edit -> Cut** option description for details.

Cursor Category

Cursor Jump 

Click the **Cursor Jump** icon to show the **Cursor Intersect** at a visible position in the **Curve** pane. The bracket next to the cursor button indicates the current **Cursor Intersect** (X-axis, Y-axis) in the **Curve** pane.

Marker Jump 

Click the **Marker Jump** icon to show the **Marker Intersect** at a visible position in the **Curve** pane. The bracket next to the marker button indicates the current **Marker Intersect** (X-axis, Y-axis) in the **Curve** pane.

Zoom Area 

Refer to the **View -> Zoom Area** option description for details.

View Category

Zoom Out X/Y 

Refer to the **View -> Zoom Out X/Y** option description for details.

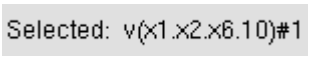
Zoom In X/Y 

Refer to the **View -> Zoom In X/Y** option description for details.

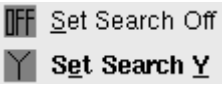
Zoom All X/Y 

Refer to the **View -> Zoom All X/Y** option description for details.

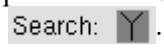
Search Category

Select 

This field shows the selected signal name.

Search 

There are two options for the **Search** selection fields: **Search: OFF** 

and **Search**: **Y** 

. The **Search: OFF** option searches for the closest FFT result point of the selected FFT signal. The **Search: Y** option searches for a Y value of the selected FFT signal by specifying a value in the Y value box.

Y Value Box 

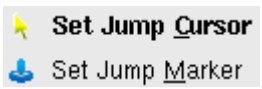
This field is for setting the Y value for the **Search: Y** option.

Search Backward 

When the **Search: OFF** option is selected, click this icon to jump to the left FFT result point closest to the selected FFT signal. When the **Search: Y** option is selected, click this icon to move the **Cursor Intersect** or the **Marker Intersect** (depending on the **Move** selection fields) to the previous closest FFT result point of the selected FFT signal.

Search Forward 

When the **Search: OFF** option is selected, click this icon to jump to the next closest FFT result point of the selected FFT signal. When the **Search: Y** option is selected, click this icon to move the **Cursor Intersect** or the **Marker Intersect** (depending on the **Move** selection fields) to the next closest FFT result point of the selected FFT signal.

Move 

When the **Search: OFF** option is selected, select the **Set Jump Cursor** option or the **Set Jump Marker** option to move the **Cursor Intersect** or the **Marker Intersect** for the **Search Backward** and **Search Forward** options.

Language Support and Compile/Import Methods

Overview

The Verdi platform can compile, import, view, debug, and dump waveform data for several Hardware Description Languages (HDLs) and Hardware Verification Languages (HVLs). This chapter gives a brief introduction to language support focusing on compiling, importing, and viewing the supported design, assertion, and testbench languages.

The following design and verification languages are supported:

- Verilog and Verilog-2001
- VHDL
- SystemVerilog, SystemVerilog-2005, SystemVerilog-2009, and SystemVerilog-2012

The following assertion language is supported:

- SVA (SystemVerilog Assertions)

The following power languages are supported:

- UPF (Unified Power Format)
- CPF (Common Power Format)

The following netlist language is supported:

- HSPICE

The following SystemVerilog class libraries are supported:

- OVM (Open Verification Methodology)
- UVM (Universal Verification Methodology)

What is Supported?

The following sections summarize where to locate the reference material for the various languages and describes the support available.

NOTE: For the list of constructs for any languages not currently supported, refer to the product release notes.

Verilog/VHDL

All constructs, as described in the appropriate language reference manual at http://www.ieee.org, are supported.

Specify the **-v95** language option on the Verdi or *vericom* command lines for Verilog language support. Specify the **-93** language option on the Verdi or *vhdlcom* command lines for VHDL language support.

SystemVerilog/SVA

All constructs, as described in the SystemVerilog IEEE 1800-2005 reference manual at http://www.ieee.org, are supported.

Specify the **-sv** language option on the Verdi or *vericom* command lines for the language support.

SystemVerilog 2005

All constructs, as described in the Verilog IEEE 1364-2005 reference manual at http://www.ieee.org, are supported.

Specify the **-2005** language option on the Verdi or *vericom* command lines for the language support.

SystemVerilog 2009

All constructs, as described in the SystemVerilog IEEE 1800-2009 reference manual at http://www.ieee.org, are supported.

Specify the **-2009** language option on the Verdi or *vericom* command lines for the language support.

SystemVerilog 2012

All constructs, as described in the SystemVerilog IEEE 1800-2012 reference manual at http://www.ieee.org, are supported.

Specify the **-2012** language option on the Verdi or *vericom* command lines for the language support.

NOTE: When the -**2012** option is specified on the Verdi or *vericom* command lines, interface class is supported for SystemVerilog 2012 data type in the compilation phase.

UPF

UPF versions 1.0 and 2.0 constructs, as described in the IEEE 1801-2009 UPF standard at http://www.ieee.org, are supported.

CPF

CPF versions 1.1 and 1.0, as described in the specifications at http://www.si2.org, are supported.

OVM

OVM versions 2.1.2, as described in the specifications at http://verificationacademy.com, are supported.

UVM

OVM versions 1.0p1 and 1.1d, as described in the specifications at http://www.accellera.org, are supported.

Dump HVL/HDL Simulation Results to FSDB

Refer to the *Linking Novas Files with Simulators and Enabling FSDB Dumping* document for details on linking Novas object files to simulators for FSDB dumping and FSDB dumping commands available for dumping HDL/HVL waveform data to FSDB.

Debug HDL/HVL Designs

After the HDL/HVL design languages have been compiled and imported, and the FSDB file created (optional - required for simulation debug), the Verdi platform provides a unified debug environment for viewing, static tracing (driver, load), annotating simulation results, dynamic tracing (active trace), and correlating different views, along with other features. Multiple languages can be imported into the same Verdi session with debug supported across languages.

Compile HDL Source Files

This section gives a brief introduction to compiling HDL source code to a design library. Each language is discussed individually. It is possible to compile multiple languages using a combination of the steps described.

Refer to the *Design Libraries* section of *Appendix B: Customizing Verdi* for details on basic design library mapping for Verilog/VHDL designs, and on using an NC-like mapping scheme. Without the library map information in the *novas.rc* resource file, compile errors and undefined modules are reported when a pure VHDL or mixed-language design is loaded.

Compile Verilog

Use the *vericom* utility to compile the Verilog design files into different libraries. For example, first specify the Verilog source files in a run file (*run.f*) similar to the following:

```
system.v
pram.v
TopModule.v
ALUB.v
CCU.v
PCU.v
alu.v
```

Then invoke the *vericom* utility with the following command:

```
% vericom -lib <libName> -f run.f
```

NOTE: If the Verilog source files include 2001 constructs, add the **-2001** option to the Verdi command line.

Alternatively, the **+verilog2001ext+.v2k+** option can be used to specify the preferred Verilog 2001 file extension for automatic recognition. This applies to both the run file and command line.

The file extension specification is a requirement if the design contains a mix of Verilog-95, Verilog-2001, and SystemVerilog. In this situation, the command line options (**-sv**, **-2001**) are not recommended as only the last option takes effect.

Alternatively, each design file can be listed individually on the *vericom* command line. For example:

```
% vericom -lib <libName> system.v pram.v TopModule.v
% vericom -lib <libName> ALUB.v CCU.V PCU.v alu.v
```

In either case, the *vericom* utility compiles the specified Verilog design files into the designated library according to the basic library mapping scheme in the Verdi platform. Note that design units are incrementally compiled leaving design units previously compiled to the <*libName*> unchanged. To remove/replace specified modules from a designated library and keep the remaining Verilog design units previously compiled to the <*libName*> unchanged, the *vericom* replace option must be specified. For example:

```
% vericom -lib <libName> -rep <verilog_design_files>
```

If **-lib <libName>** is specified in the *vericom* utility, *vericom* checks whether the lowercase **<libName>** is an existing logical library name according to the basic library mapping scheme in the Verdi platform.

- If it is, then the specified design files are compiled into the mapped <*physical_location*>.
- If it is not, then the <*libName*> is the real physical library into which the specified design files are compiled.

If no libraries were specified for the *vericom* utility, design units are compiled into the default library, *work.lib++*.

To use an NC-like library mapping scheme, refer to the *Compile Verilog with an NC-like Library Mapping Scheme* section later in this chapter.

Compile VHDL

Before compiling VHDL, library mapping must be added to the *novas.rc* resource file, similar to how library mapping information is specified for simulation with the ModelSim **.ini* or Cadence *cds.lib* files. Refer to the *Design Libraries* section of *Appendix B: Customizing Verdi* for details on design library mapping for VHDL designs.

NOTE: Before VHDL compilation, the *novas.rc* resource file must be copied to the local working directory and then manually modified according to the library mapping.

For example, if a VHDL file includes the following information:

```
...
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.
library foo;
```

```
use foo.all;
library goo;
use goo.all;
...
```

The following must be added in the *novas.rc* resource file, `Library` section:

```
...
foo = /home/myvhdl/foo
goo = goo
...
```

The *ieee*, *std*, *synopsys* and *vital2000* standards have precompiled libraries that are included with the Verdi platform; therefore, specifying the mapping for these libraries is not necessary.

After the mapping information is defined in the *novas.rc* resource file, the *vhdlcom* utility can be used to compile the VHDL design files into different libraries. For example:

```
% vhdlcom -lib <libName> <vhdl_design_files>
```

With this command, the specified VHDL design files can be compiled into a designated library. Note that the existing design units in the <*libName*> are not overwritten. For example:

```
% vhdlcom -lib /home/myvhdl/foo foo1.vhd foo2.vhd ...
% vhdlcom -lib goo goo1.vhd goo2.vhd ...
```

If **-lib <libName>** is specified on the *vhdlcom* command line, the *vhdlcom* utility checks whether the lowercase <*libName*> is an existing logical library name according to the basic library mapping scheme in the Verdi platform.

- If it is, then the specified design files are compiled into the mapped <*physical_location*>.
- If it is not, then the lowercase <*libName*> is the real physical library into which the specified design files are compiled.

If no libraries were specified for the *vhdlcom* utility, design units are compiled into the default library, *work.lib++*.

If a large number of VHDL source files exist in a VHDL or a mixed-language design, these files must be organized in the right order. Otherwise, compiling them for the Verdi platform may be difficult. The *vhdlcom* utility provides two command line options, **-smartorder** and **-smartscript**, to solve problems associated with order:

- **-smartorder** compiles the VHDL files in an order independent of node (similar to <*smartorder*> in NC). The output is a precompiled library.

- **-smartscript** generates a compilation script with the proper order.

Compile SystemVerilog

Just like Verilog, the *vericom* utility can be used to compile the SystemVerilog design files into different libraries using the *vericom* utility with the additional **-sv** language switch. For example, first specify the SystemVerilog source files in a run file (*run.f*) and then invoke the *vericom* utility with the following command:

```
% vericom -lib <libName> -sv -f run.f
```

The **+systemverilogext+.sv+.SV+** option can be used to specify the preferred SystemVerilog file extension for automatic recognition. This applies to both the run file and command line.

NOTE: The file extension specification is a requirement if the design contains a mix of Verilog-95, Verilog-2001, and SystemVerilog. In this situation, the command line options (**-sv**, **-2001**) are not recommended as only the last one takes effect.

Alternatively, each SystemVerilog design file can be listed individually on the *vericom* command line. For example:

```
% vericom -lib <libName> <systemverilog_design_files>
```

In either case, the *vericom* utility compiles the specified Verilog design files into the designated library according to the basic library mapping scheme in the Verdi platform. Note that design units are incrementally compiled leaving design units previously compiled to the <*libName*> unchanged.

If **-lib <libName>** is specified on the *vericom* command line, the *vericom* utility checks whether the lowercase <*libName*> is an existing logical library name according to the basic library mapping scheme in the Verdi platform.

- If it is, then the specified design files are compiled into the mapped <*physical_location*>.
- If it is not, then the <*libName*> is the real physical library into which the specified design files are compiled.

If no libraries were specified for the *vericom* utility, design units are compiled into the default library, *work.lib++*.

SystemVerilog Pragmas

If SystemVerilog pragmas (//sv_pragma or /*sv_pragma ... */) are used, **-sv_pragma** must be specified in addition to the **-sv** option on the *vericom* command line. For example:

```
% vericom -lib <libName> -sv -sv_pragma -f run.f
```

The **-sv_pragma** option does not automatically turn on the **-sv** option; it must be explicitly specified.

NOTE:
1. sv_pragma must be the first non-white space token after // or /*
2. sv_pragma must be in the same line as /*

Compile Mixed-Language - Verilog/VHDL

Before compiling the mixed-language design, library mapping must be added to the *novas.rc* resource file, similar to how library mapping information is specified for simulation with the ModelSim *.*ini* or Cadence *cds.lib* files. Refer to the *Design Libraries* section of *Appendix B: Customizing Verdi* for details on basic design library mapping for Verilog/VHDL designs.

NOTE: Before Verilog/SystemVerilog/VHDL compilation, the *novas.rc* resource file must be copied to the local working directory and then manually modified according to the library mapping.

For mixed-language designs, the mixed-language source code must be compiled into a Verdi library format. The recommendation is to compile the Verilog, VHDL, and SystemVerilog design source code into separate libraries using the methods described previously.

Since various simulators apply different approaches to compile and elaborate mixed-language designs, the Verdi platform offers approaches similar to those provided by ModelSim and Cadence NC simulators to import mixed-language designs.

1. Compile flow for the ModelSim simulator:
   - Use the *vhdlcom* utility to compile all VHDL source code into libraries.

- Use the *vericom* utility to compile all Verilog source code into libraries.

NOTE: For Verilog instantiation of VHDL design units, the approach of the library explicitly named in the special escaped identifier instance name is used. For more detailed information, see the ModelSim's user manual.

2. Compile flow for the Cadence NC simulator:
   - Use the *vhdlcom* utility to compile all VHDL source code and VHDL shell code that is generated for applied Verilog modules.
   - Use the *vericom* utility to compile all Verilog source code and Verilog shell code that is generated for applied VHDL entities.

NOTE: VHDL shell code is used for VHDL instantiation of Verilog design units and Verilog shell code is used for Verilog instantiation of VHDL design units. For more information, see the Cadence *ncshell* utility.

Compile Mixed Verilog/SystemVerilog

Before compiling a mixed Verilog/SystemVerilog design, the language used in design files must be identified by either using a different file extension for each language or by splitting the design files with different languages into separate run files. Then, use the *vericom* utility to compile the design files into the same library. For example, first specify the Verilog source files with different file extensions in a run file (*run.f*) similar to the following:

```
system.v
pram.sv
TopModule.v
ALUB.v2k
CCU.SV
PCU.v
alu.v2k
```

Then execute the *vericom* utility with the following command:

```
% vericom -lib <libName> -f run.f +verilog2001ext+.v2k+
+systemverilogext+.sv+.SV+
```

Alternatively, the different language design files can be separated into different run files (regardless of the file extension) and the *vericom* utility used to compile them into the same library. For example:

```
% vericom -f run_verilog_files.f
% vericom -2001 -f run_v2k_files.f
% vericom -sv -f run_sv_files.f
```

All design files are compiled into a library called *work*.

Logical Library Name vs. Physical Library Path

The Logical Library Name (the <libName> in the above sections) can be mapped to a Physical Library Path (the <physical_location> in the above sections) in the *novas.rc* resource file.

For example, in the *novas.rc* resource file, a library mapping is defined as follows:

```
[Library]
logicalName = ./path1/physical.lib++
```

In this mapping, "logicalName" is the Logical Library Name and "./path1/physical.lib++" is the Physical Library Path.

If no mapping exists in the *novas.rc* resource file, the Verdi platform obtains the mapping automatically from the current directory by default. This mechanism remains the same as the *novas.rc* resource file as the following mapping shows:

```
logicalName = ./logicaNamel.lib++
```

The mechanism assumes that all **.lib++* directories under the current directory are Physical Library Paths and obtains the name before "*.lib++*" as the Logical Library Name.

NOTE: The library specified for **-lib**, **-L**, and **-Lf** options are all Logical Library Names.

Import HDL/HVL/Power Source Files

This section is a brief introduction to importing HDL/HVL/power source code. Each language is discussed individually. It is possible to import multiple languages using a combination of the described steps.

Refer to the *Compile HDL Source Files* section for details on compiling Verilog/VHDL/Mixed-language designs into a design library. The following information is a general introduction.

Import Verilog Files

Verilog source code can be loaded from the command line (run file or precompiled library) or from the GUI.

Import Verilog on the Command Line from a Run File

Specify the Verilog source files in a run file similar to the following:

```
system.v
pram.v
TopModule.v
ALUB.v
CCU.v
PCU.v
alu.v
-v lib.v
-y ../special
```

Invoke the Verdi platform with the following command:

```
% verdi -f run.f
```

NOTE: To not tag modules in the library file (specified with **-v**) or in the library directory (specified with **-y**) as library cells, specify the **-ssv** and/or **-ssy** options as well. For example:

```
% verdi -f run.f -ssv -ssy
```

causes modules specified in *lib.v* or in files located in *../special* to be treated as normal modules instead of library cells.

The Verdi platform compiles and elaborates the design simultaneously. Alternatively, the files can be listed on the command line individually:

```
% verdi system.v ... alu.v -v lib.v -y ../special
```

NOTE: If the Verilog source files include 2001 constructs, add the **-2001** option to the Verdi command line.

```
% verdi -f run.f -2001
```

Alternatively, the +**verilog2001ext+.v2k**+ option can be used to specify the preferred Verilog 2001 file extension for automatic recognition. This applies to both the run file and command line.

```
% verdi -f run.f +verilog2001ext+.v2k+
```

The file extension specification is a requirement if the design contains a mix of Verilog-95, Verilog-2001, and SystemVerilog. In this situation, the command line options (**-sv**, **-2001**) are not recommended as only the last one takes effect.

Import Verilog on the Command Line from a Library

As described previously, Verilog can be precompiled into a Verdi library using the *vericom* utility.

```
% vericom -lib mylib -f run.f
```

NOTE: To not tag the modules in the library file (specified with **-v**) or in the library directory (specified with **-y**) as library cells, specify the **-ssv** and/or **-ssy** options as well. For example:
`% vericom -lib mylib -f run.f -ssv -ssy`

After the design is compiled, import it into the Verdi platform with:

```
% verdi -lib mylib -top mytop
```

NOTE: If the Verilog design is compiled to multiple libraries, the libraries can be included using a ModelSim- or NC-like binding scheme. Refer to the *Import Verilog Design - Multiple Libraries* section.

NOTE: Verilog modules are case-sensitive.

Import Verilog from the GUI

The Verdi GUI can be used to import Verilog source code. First, invoke the GUI with:

```
% verdi
```

Then, choose **File -> Import Design** from the *nTrace* menu to open the *Import Design* form and select the **From File** tab.

Finally, select Verilog or Verilog-2001 as the **Language**, select the desired files, click **Add**, and then click **OK** to import the files into the Verdi platform.

Alternatively, select the **From Library** tab and select a compiled library to load.

Import VHDL Files

VHDL source code can be loaded from the command line (run file or precompiled library) or from the GUI.

Import VHDL on the Command Line from a Run File

Specify the VHDL source files in the correct compilation order in a run file (*run.f*) and invoke the Verdi platform with the following command:

```
% verdi -vhdl -f run.f -top <TopDesign>
```

The Verdi platform compiles and elaborates the design simultaneously.

Import VHDL on the Command Line from a Library

As previously described, VHDL can be precompiled into a Verdi library using the *vhdlcom* utility.

```
% vhdlcom -lib mylib a.vhd b.vhd c.vhd ...
```

NOTE: Before VHDL compilation, the *novas.rc* resource file must be copied to the local working directory and then manually modified according to the library mapping.

After the design is compiled, import it into the Verdi platform with:

```
% verdi -lib mylib -top mytop
```

When VHDL is loaded from a library, the *novas.rc* resource file with the library mapping information used during the compile must be referenced. See the example under the *Compile VHDL* section and refer to the *Design Libraries* section of *Appendix B: Customizing Verdi* for details on basic design library mapping for VHDL designs.

Import VHDL from the GUI

The Verdi GUI can be used to import VHDL source code. First, invoke the GUI with:

```
% verdi
```

Then, choose **File -> Import Design** from the *nTrace* menu to open the *Import Design* form and select the **From Library** tab.

Finally, select the compiled library, select the desired design top, click **Add**, and then click **OK** to import the design library into the Verdi platform.

Alternatively, select the **From File** tab, select VHDL (several versions are available) as the **Language**, select the desired files, click **Add**, and then click **OK** to import the files into the Verdi platform. The proper compile order can be manually selected.

Import SystemVerilog Files

SystemVerilog files can be imported from the command line (run file or precompiled library) or from the GUI.

Import SystemVerilog on the Command Line from a Run File

Specify the SystemVerilog source files in a run file (*run.f*) and then add the **-sv** option to the Verdi command line when the design files are imported.

```
% verdi -sv -f run.f
```

Alternatively, use the +**systemverilogext+.sv+.SV**+ option to specify the preferred SystemVerilog file extension for automatic recognition. This applies to both the run file and command line.

NOTE: The file extension specification is a requirement if the design contains a mix of Verilog-95, Verilog-2001, and SystemVerilog. In this situation, the command line options (**-sv**, **-2001**) are not recommended as only the last one takes effect.

Import SystemVerilog on the Command Line from a Library

As described previously, SystemVerilog can be precompiled into a Verdi library using the *vericom* utility and the **-sv** language switch.

```
% vericom -lib svlib -sv -f run.f
```

After the design is compiled, import it into the Verdi platform by specifying the top implicitly:

```
% verdi -lib svlib
```

Import SystemVerilog from the GUI

The Verdi GUI can be used to import SystemVerilog source code. First, invoke the GUI with:

```
% verdi
```

Then, choose **File -> Import Design** from the *nTrace* menu to open the *Import Design* form and select the **From File** tab.

Finally, select SystemVerilog as the **Language**, select the desired files, click **Add**, and then click **OK** to import the files into the Verdi platform.

Alternatively, select the **From Library** tab and select a compiled library to load.

Import SystemVerilog Pragmas

If SystemVerilog pragmas (//sv_pragma or /*sv_pragma ... */) are used, **-sv_pragma** must be specified in addition to the **-sv** option on the Verdi command line. For example:

```
% verdi -sv -sv_pragma -f run.f
```

The **-sv_pragma** option does not automatically turn on the **-sv** option, it must be explicitly specified.

NOTE:

1. sv_pragma must be the first non-white space token after // or /*
2. sv_pragma must be in the same line as /*

Import Mixed HDL Language - Verilog/VHDL

After compiling the mixed-language source code into a Verdi library format, mixed-language designs can be imported and elaborated from the command line as follows:

```
% verdi -lib <VHDLlibName1> -lib <VHDLlibName2>
-top <designTop> -Lf <VerilogLib1> -L <VerilogLib2>
```

When a mixed-language design is loaded, the same *novas.rc* resource file with the library mapping information used during the compile must be referenced.

Refer to the *Design Libraries* section of *Appendix B: Customizing Verdi* for details on basic design library mapping for VHDL/Verilog designs.

After the library mapping is specified in the *novas.rc* resource file, for example:

```
[Library]
VerilogLib1 = ./VerilogLib1
VerilogLib2 = ./VerilogLib2
VHDLlib1 = ./VHDLlib1
VHDLlib2 = ./VHDLlib2
```

the Verdi platform can be invoked as follows:

```
verdi -nclib -top system // Import the design into the Verdi
platform.
```

For ModelSim users, the Verdi platform provides the same options and library search scheme for Verilog source files that were compiled in different libraries as ModelSim's for design unit instantiation as described in the *ModelSim-like Library-Binding Scheme* section.

For Cadence NC users, the **-nclib** Verdi command line option is also required. Refer to the *NC-like Library-Binding Scheme* section for various methods to include the design top.

After importing precompiled mixed-language designs into the Verdi platform, the usage is the same as for pure VHDL or pure Verilog designs.

Import Mixed Verilog/SystemVerilog Files

Mixed Verilog/SystemVerilog files can be imported from the command line (run file or precompiled library).

Import Mixed Verilog/SystemVerilog on the Command Line from a Run File

Before importing a mixed Verilog/SystemVerilog design, language used in design files must be identified by using a different file extension for each language. Specify the source files with different file extensions in a run file (*run.f*) similar to the following:

```
system.v
pram.sv
TopModule.v
ALUB.v2k
CCU.SV
PCU.v
alu.v2k
```

Then add the file extensions to the Verdi command line when the design files are imported:

```
% verdi -f run.f +verilog2001ext+.v2k+
+systemverilogext+.sv+.SV+
```

Files with extension name *.sv* or *.SV* are recognized as SystemVerilog, files with *.v2k* are recognized as Verilog-2001, and files with *.v* are recognized as Verilog-95.

NOTE: The file extension specification is a requirement if the design contains a mix of Verilog-95, Verilog-2001, and SystemVerilog. In this situation, the command line options (**-sv**, **-2001**) are not recommended as only the last one takes effect.

Import Mixed Verilog/SystemVerilog on the Command Line from a Library

As described previously, mixed Verilog/SystemVerilog can be precompiled into a Verdi library using the *vericom* utility and the file extensions or by separating the different language design files into different run files (regardless of the file extension) and using the *vericom* utility to compile them into the same library. For example:

```
% vericom -f run_verilog_files.f
% vericom -2001 -f run_v2k_files.f
% vericom -sv -f run_sv_files.f
```

After the design is compiled, import it into the Verdi platform by specifying the top implicitly:

```
% verdi -lib work
```

or (when multiple top levels exist):

```
% verdi -lib work -top '$root TopModule'
```

Import HSPICE Files

HSPICE files can be imported from the command line (run file or precompiled library). All HSPICE content is treated as a comment except for *.include*. A *.include* statement opens the file specified whether it is HSPICE, Verilog, or VerilogAMS.

NOTE: When the **+spiceext+.<file_suffix>** option is specified on the command line or the suffix of the imported file is *.sp*, HSPICE constructs are recognized. This means the content between *.subckt* and *.ends* is treated as a comment and a node is added to the design browser to bind the instance.

Import HSPICE on the Command Line from a Run File

Specify the HSPICE source files in a run file (*run.f*) and then add the **+spiceext+** option to the Verdi command line for automatic recognition of HSPICE files when the design files are imported.

```
% verdi -f run.f -ams +spiceext+.sp+.SP+.hsp+
```

NOTE: The **+spiceext+** option is case-insensitive so **+SpiceExt+** is also acceptable; however, the included extensions are case-sensitive. For example, if the HSPICE files have a mix of *.sp* and *.SP* file extensions, both **+.sp** and **+.SP** must be specified.

Alternatively, list the HSPICE files on the command line individually:

```
% verdi -f run.f -ams spice1.sp spice2.SP spice3.hsp
+spiceext+.sp+.SP+.hsp+
```

Import HSPICE on the Command Line from a Library

HSPICE can be precompiled into a Verdi library using the *vericom* utility and the **+spiceext+** file extension designator.

```
% vericom -lib hspicelib -f run.f -ams +spiceext+.sp+.SP+.hsp+
```

where the *run.f* file contains a combination of Verilog and HSPICE files.

After the design is compiled, import it into the Verdi platform by specifying the top implicitly:

```
% verdi -lib hspicelib
```

Import Power Files

Power source files can be loaded from the GUI or from the command line.

Import CPF/UPF Files from the GUI

After the HDL/HVL design is loaded into the Verdi platform, use the **File -> Import CPF/UPF Files** option in *nTrace* to open the *Import CPF/UPF Files* form.

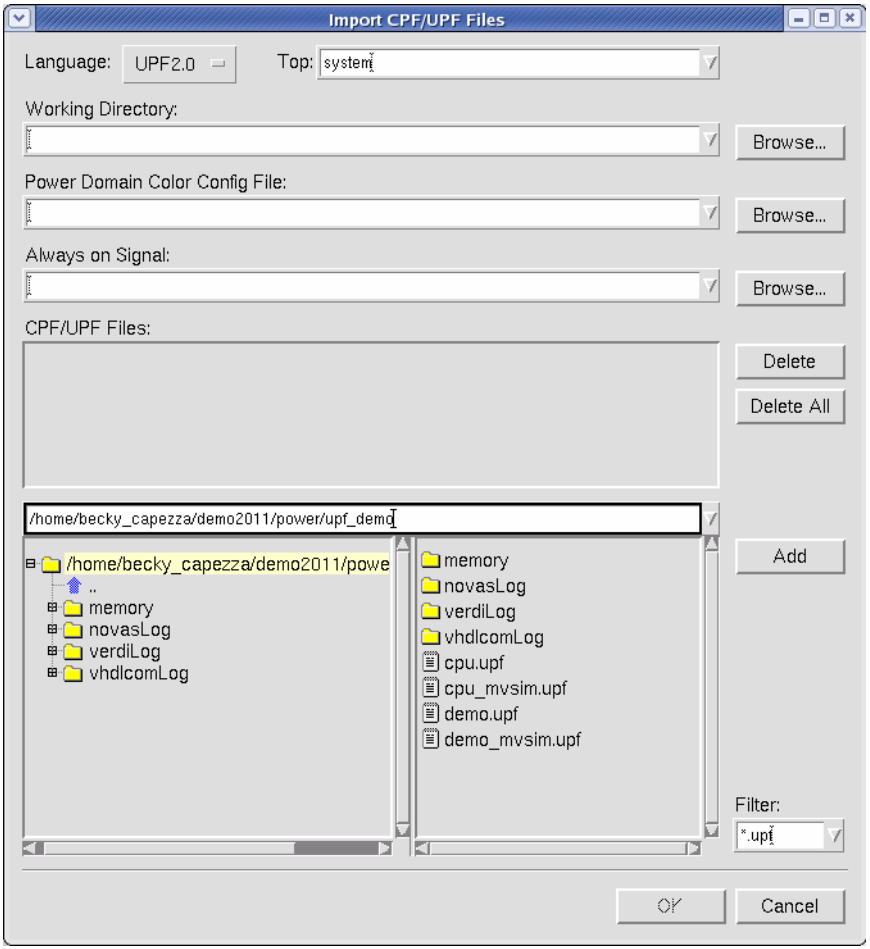


Figure: Import CPF/UPF Files Form

To import UPF files, select from either the **UPF 1.0** or **UPF2.0** options in the **Language** field. Then select one or more existing UPF files to load into the Verdi platform and click **OK**. To import CPF files, select from either the **CPF1.0** or

CPF 1.1 options in the **Language** field. Then select one or more existing CPF files to load into the Verdi platform and click **OK**.

Import UPF from the Command Line

To import the UPF file(s) from the Verdi command line, use the **-upf1.0 <filename>** or **-upf2.0** <**filename>** command line options to import the UPF file(s) and the **-upfTop <top>** option to specify the top scope for the imported power specification.
For example:

```
%> verdi -sv -f run.f upf_package.sv power_test.v -ssf
rtl.vcd -upf1.0 supply_package_1.upf -upfTop system
```

When more than one UPF file exists, specify the **-upf1.0**/**-upf2.0** option multiple times. For example:

```
%> verdi -sv -f run.f -upf2.0 power_1.upf -upf2.0 power_2.upf
-upfTop system
```

NOTE: Instead of using the **-upfTop** option, the **set_design_top** command can be specified at the beginning of the loaded UPF file. The Verdi platform sets the UPF power specification's top scope according to either the **-upfTop** option or the **set_design_top** command setting. If the top scope is not set using one of these methods, the Verdi platform picks an arbitrary top scope for the UPF power specification.

When the **set_design_top** command and the **-upfTop** option are both used, the specified design top should be the same otherwise an error message is reported.

Import CPF from the Command Line

To import the CPF file(s) from the Verdi command line, use the **-cpf1.0 <filename>**, **-cpf1.1 <filename>**, or **-cpf2.0 <filename>** command line options to import the CPF file. For example:

```
%> verdi nano_defines.v testbench.v nano.v -cpf1.0 nano.cpf
```

When more than one CPF file exists, specify the **-cpf1.0**, **-cpf1.1** or **-cpf2.0** option multiple times. For example:

```
%> verdi nano_defines.v testbench.v nano.v -cpf1.1 nano.cpf
-cpf1.1 nano1.cpf -cpf1.1 nano2.cpf
```

There are four options provided that are compatible with simulator (IUS) commands:

- **lps_cpf:** Specifies the name of the CPF file to be used with the design.

- **lps_iso_off:** Turns port Isolation off.
- **lps_off:** Turns all low-power simulation off.
- **lps_rtn_off:** Turns Retention state off.

NOTE: Unlike UPF, the top scope setting option is not required for CPF on the Verdi command line.

If the **set_cpf_version** command is specified in the loaded CPF file and the specified version is higher than the specified language version in the *Import CPF/UPF Files* form or the Verdi command line, the Verdi platform applies the higher version specified by the **set_cpf_version** command.

Browse Entities in the Design Browser

After HDL, assertion, and power source files are loaded, the design hierarchy is shown in the *nTrace* design browser pane. Each language provides various tree nodes to indicate different types of code functions. Refer to the **Help -> Legend** command for the complete list.

Special Compile/Import Topics

This section provides some additional details for special compile and import topics.

Compile Verilog with an NC-like Library Mapping Scheme

To compile Verilog design files with an NC-like library mapping scheme, the **-nclib** option must be specified on the *vericom* command line to tell the *vericom* utility to use the NC binding scheme. With this option, the Verilog design units are compiled into a *library.cell (view)*. The cell name is always set to the name of the design unit. The library locations where the *vericom* utility stores compiled objects and the view names that the *vericom* utility assigns to them can be controlled.

To specify the library and view, use variables defined in the *[NC_Sim]* section of the Verdi resource file, command line options, or compiler directives of Verilog source files.

The order of precedence (from high to low) is as follows:

1. The *`worklib* and *`view* compiler directives specified in Verilog source files.

   Refer to the *`worklib* and *`view* Compiler Directives in the NC Verilog documentation for details.

2. The **-work** and **-view** command line options:
   - **-view view_name**

     Provides the same function as *ncvlog -view view_name* and uses the specified view name for the design units of the input Verilog source files. For example:

     ```
     vericom -nclib -view novas ../src/novas.v
     ```

     All design units in *../src/novas.v* have a view name *novas*. Note that using the command line option, **-view**, overrides the *defaultView* and *viewMap* variables in the *novas.rc* resource file.
   - **-work library_name**

     Provides the same function as *ncvlog -work library_name* and uses the specified library name for the design units of the input Verilog source files. For example:

     ```
     vericom -nclib -work novas -view novas -f ../src/novas.f
     ```

All design units in the file *../src/novas.f* are compiled into the library *novas* and have a view name *novas*. Note that using the command line option, **-work**, overrides the *workLibrary* and *libMap* variables in the *novas.rc* resource file.

3. The definitions of the *workLibrary* and *defaultView* variables in the `[NC_Sim]` section of the Verdi resource file (*novas.rc*).
4. The definitions of the *libMap* and *viewMap* variables in the `[NC_Sim]` section of the Verdi resource file (*novas.rc*).
5. Use the default library, *work.lib++*, as the library name.
6. Use view name *module* for modules and macro modules.
7. Use view name *udp* for UDPs.

Import Verilog Design - Multiple Libraries

Refer to the *Import Verilog Files* section for details on various methods of importing Verilog source code from a single design library. The methods below describe how to import from multiple design libraries using either a ModelSim or NC binding scheme.

ModelSim-like Library-Binding Scheme

For complex designs, modules can be compiled in various libraries for better organization. The Verdi platform supports importing Verilog designs from multiple libraries so design files can be easily organized and analyzed. Since instantiation bindings are determined during elaboration, modules can be compiled into different libraries and then the Verdi platform can search libraries when loading the design.

After compiling the Verilog design into different libraries with the *vericom* utility, all Verilog instantiations are resolved in the following order:

1. Search libraries specified with **-Lf** options in the order that appears on the Verdi command line.
2. Search the library specified in the `` `uselib lib=<library> `` directive in the Verilog source code.
3. Search the library as specified in design configurations if written for binding any specific module(s) or specifying a default liblist for a module or design.
4. Search the library explicitly named in the special escaped identifier instance name (for example, `\mylib.mod1 mod(a, b, c);`).

5. Search libraries specified with **-L** options in the order that appears on the Verdi command line. **-L** searches the current library instead of searching the "work" library.
6. Search the library where the parent module was found.
7. Search all libraries as listed in the *novas.rc* [`Library`] section.

NOTE:

a. The above order is suitable for ModelSim users.
b. All logical libraries of Verilog are case-insensitive and are converted to lowercase.
c. All physical libraries of Verilog are case-sensitive.

NC-like Library-Binding Scheme

After compiling a Verilog design into different libraries with the **-nclib** option on the *vericom* command line, the design can be imported according to an NC-like library binding scheme.

During elaboration, the Verdi platform resolves the design units according to the following rules:

1. The default binding mechanism. The binding rules are as follows:
   - For a particular module, if an instance of the same design unit has already been bound during the current elaboration, use the same binding.
   - Using the libraries that are listed with the *libMap* variable of *novas.rc*, and beginning with the library where the parent was bound, search the view names that are listed with the *viewMap* variable in order, beginning with the view that has the same view name as the parent. If a view that has the same view of the parent is found, use that binding.
   - If no binding is found, continue with the next view in the *viewMap* variable. Continue searching the views in order, wrapping around to the first view.
   - If no binding is found, move on to the next library listed in the *libMap* variable and repeat the view search using the same steps.
   - If no binding is found, search the library where the parent module was found to determine a possible binding and use the first candidate.
   - If no binding is found, search all libraries defined in the [`Library`] section and use the first candidate.
   - If no binding exists, the message form reports an error.
2. The **-binding** option:

Use the **-binding** option to force the binding of a cell (design unit) to a particular library and view. Using this option overrides the default view and library search mechanism. The specified cell is bound to the library and view specified. After the first instance of the design unit has been resolved, all the instances of the same design unit use the same binding except the *`uselib* directive is applied to force different bindings for design units with the same name.

3. The *`uselib* directive:
   - Overrides the default search mechanism and the **-binding** command line options.
   - Each *`uselib* directive explicitly defines a library or view that resolves the instances that follow it until the elaboration encounters another *`uselib* directive, which redefines the search. An empty *`uselib* directive or a *`nouselib* directive makes the preceding `uselib directives ineffective.

NOTE: The **-nclib** option globally searches all libraries if it is unable to search the library specified with *`uselib*.

To use an NC-like library-binding scheme, the **-nclib** command line option must be used on the Verdi command line. With this option, the Verdi platform elaborates the precompiled design based on the NC elaboration scheme and validates the following options for the Verdi command.

1. The **-binding [library.]cell:view** option:

   This option implements the same function as the NC command:

   ```
   ncelab -binding...
   ```

   This forces the binding of a cell to the designated library and view. For example:

   ```
   verdi -nclib -top novas.top:rtl -binding src.cpu:behav
   ```

   The Verdi platform elaborates *novas.top:rtl* and uses **-binding** to force the binding of the design unit of *cpu* to the *src* library and the *behav* view.

   ```
   verdi -nclib -top novas.top -binding cpu:behav
   ```

   The Verdi platform elaborates top and uses **-binding** to force the binding of the design unit of *cpu* to the *behav* view. If the *library* or *cpu* are not specified, the Verdi platform searches all possible libraries for it.

   If the library or view is not specified in **-binding**, the Verdi platform uses the same library-binding scheme as NC.

In the Verdi platform, **-binding** must use one of the following argument types:

- **-binding library.cell:view**
- **-binding library.cell**
- **-binding cell:view**

NOTE: -binding cell is not supported.

2. The **-top [library.]cell:view** option:

   This option implements the same function as the NC command:

   ```
   ncelab [library.]cell[:view]
   ```

   For example:

   ```
   verdi -nclib -top novas.top:rtl
   ```

   The Verdi platform elaborates *top* in the library *novas* and the view *rtl*.

   If the library is not specified, then search *work.lib++* for the top design unit. For example:

   ```
   verdi -nclib -top top
   ```

   The above command is the same as:

   ```
   verdi -nclib -lib work -top top
   ```

   or

   ```
   verdi -nclib -top work.top
   ```

Debug Protected Code

This section provides information on typical debug support for protected code. Users are able to debug and understand the protected code in a similar manner to their RTL code.

Support Scope

The source code falls into the following categories

- Supported

  Normal statements without protected pragmas are fully supported.

- Supported Protected and Accessible

  The Synopsys VIP protection method using the `//vcs_vip_protect` pragma. The directive is:

  ```
  //vcs_vip_protect
  `protected
  ...
  `endprotected
  ```

  Available in both *vericom* and UFE.

- Supported Protected:
  - **+protect**
    120bit encryption for Verilog only. Output file is suffixed with .vhdp or .vp or .svp. The directives are `` `protected `` and `` `endprotected ``.
    Available in both *vericom* and UFE.
  - **-protect128**
    128bit AES encryption for both Verilog and VHDL. Output file is suffixed with .vhdp or .vp or .svp. The directives are `` `protected128 `` and `` `endprotected128 ``.
    Available in *vericom*, *vhdlcom* and UFE.
  - **+protect +pli_unprotect**
    120 bits encryption for Verilog only. Output file is suffixed with .v or .svp.
    Available in both *vericom* and UFE.
- Not Supported:
  - **-gen_ip [-dbg_ip]**
    DesignWare compatible for both Verilog and VHDL. Output file is suffixed with .vhd.e or .v.e.
    Available in UFE.

- **-ipprotect [-ipkey -ipopt]**
  IEEE encryption for both Verilog and VHDL. Output file is suffixed with .vhdp or .vp or .svp.
  Available in UFE.

NOTE: Unsupported protected code is treated as comments.

The following table summarizes the level of support for each category in the main Verdi features.

Level of Support	Description
Not Supported	Protected objects cannot be decrypted or accessed for debug. The modules containing protected code are black boxed by Behavior Analysis. The parser skips encrypted codes. However, when normal statements use any objects in the protected code, an error message is shown. Debug functions are disabled.
Supported Protected	The protected code can be successfully compiled into the Verdi platform without errors. The encrypted content is not viewable (are not decrypted) in the Verdi platform.
Supported Protected and Accessible	Protected objects are decrypted and can be accessed for debug. Behavior Analysis inferences protected objects. The parser skips encrypted codes. However, when normal statements use any objects in the protected code, an error message is shown. Most *nTrace*, Active Annotation, Interactive Debug, Behavior Analysis, Temporal Flow View, Trace X, Switching Analysis, *nSchema*, *nWave*, and VIA functions are available other than the exceptions listed below.
Supported Normal	Fully supported.

The following are exceptions for the Supported Protected and Accessible category:

- Copying, pasting, dumping, or printing of source code containing decrypted code is not supported.
- Exporting or dumping the hierarchy tree is not supported.

- Dumping trace results or log files with decrypted code is not supported. The decrypted code is replaced.
- Saving protected source code with the **File** -> **Save in HDL** option in *nSchema* or the **File** -> **Save Netlist** and **File** -> **ECO** -> **Save ECO Netlist** options in *nECO* are not supported. In these cases, the code between `'protected` and `'endprotected` is skipped.
- Correlation does not insert protected signals, instances, or modules into the CRDB.
- Data Expansion does not compute protected signals inside protected blocks. The signals are 'NF'.
- For VIA, Cont. Assignment, Atomic Stmt, and Operation objects in the Language Model and DM Model are not supported. These objects do not have any properties that can be dumped or methods to be traversed.

Source Code Visibility

The Synopsys VIP protection method uses the pragma `//vcs_lic_vip_protect` to support source code visibility in Verdi.

The directive for `vcs_vip_protect` is:

```
//vcs_vip_protect
`protected
...
`endprotected
```

The directive for`//vcs_lic_vip_protect is:`

```
// module dut   vip-protect with license check

module dut(output int out, input int i1, input int i2, input
int i3);
// vcs_lic_vip_protect
`pragma protect begin
// vip_check_license some_license
……
`pragma protect end
endmodule
```

With `// vcs_lic_vip_protect` pragma, there should be a `// vip_check_license ${license_name}` in the protected block and `-vip_licensed_source_visibility` option added to the command line.

Available in both *vericom* and UFE.

Usage of Supported Protected and Accessible

The following usage details are only for the Supported Protected and Accessible category.

Source Code and Related Functions

In the *nTrace* main window, decrypted code is treated like a normal macro except for the copying, pasting, dumping, printing and editing functions. The **Source -> Expand Decrypted Code** option shows or hides decrypted code when the design contains protected code. In addition, the Automatic Encrypted Folding option on the **Code Folding** page on the *Preferences* form (invoked with **Tools -> Preferences**) can be used to collapse or expand the protected code. Refer to the *nTrace* and *Preferences* chapters for details.

You can set breakpoints for Interactive Debug inside the decrypted protected code and execute line by line, just like macro-debugging.

NOTE: When entering Interactive Debug mode, the **Source -> Expand Decrypted Code** option is turned *on* and then disabled. The setting for this option cannot be changed in Interactive Debug mode.

Behavior Analysis

The **Ignore Protected Code** option on the **Black Box** page of the *Preferences* form (invoked with **Tools -> Preferences**) is used to specify how the Behavior Analysis engine should handle protected code.

- For general VIP protected code without the `//vcs_vip_protect` pragma, the following applies:
- When this option is turned *off*, the whole module or scope that contains the encrypted protected code is black boxed by Behavior Analysis.
- When this option is turned *on*, Behavior Analysis inferences the module or scope that contains the encrypted protected code but objects inside the protected code are black boxed.

Refer to the *Preferences* chapter for details.

Preferences

Overview

These preference settings are saved in the *novas.rc* resource file. The Verdi platform loads the resource file the next time it is invoked.

The *Preferences* form can be invoked from the **Tools** menu of *nTrace, nSchema, nState, nWave,* and *Flow View* panes.

The following field, buttons, and option are available in the *Preferences* form:

- **Find:** Specify a text string in this field to search for an option.
- **Next:** Click this button to locate the next occurrence of the string.
- **Previous:** Click this button to locate the previous occurrence of the string.
- **Match Case:** When this option is turned *on*, the search is case-sensitive. When this option is turned *off*, case is not considered during a search. The default value of this option is *off*.

General Folder

The **General** folder includes *General*, *Automatic Save*, *Docking and Framework, Font and Size, Search/Filter,* and *Miscellaneous* pages.

General Page

Use this page to set global Verdi actions.

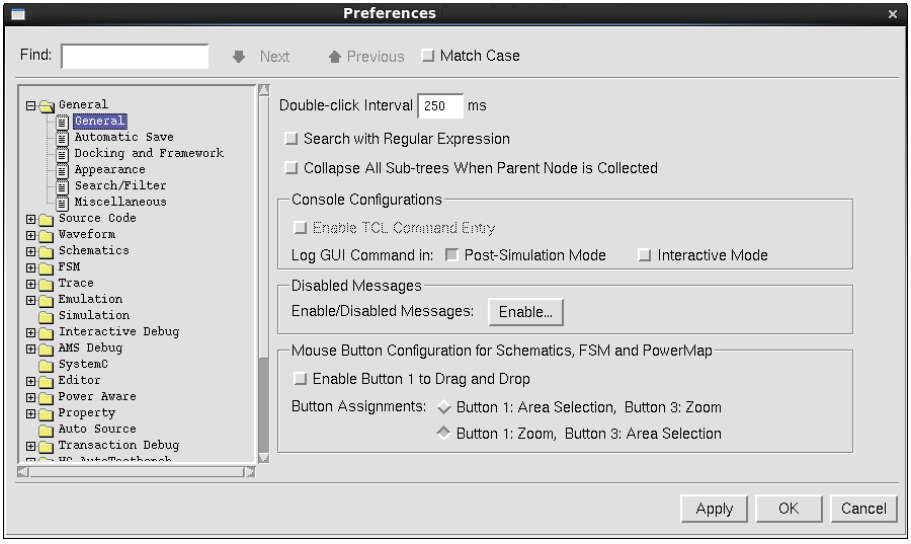


Figure: General Page

The following options are available:

- **Double-click Interval:** Specify the settings for the double-click interval in milliseconds in this field.
- **Search with Regular Expression:** When this option is turned *on*, regular expressions can be used while searching signals, instances, strings, and current scopes.
- **Enable TCL Command Entry:** This option enables *Console* window. This window can be used to enter Tcl/UCLI commands and pass the commands to the Tcl interpreter. For example, use source *<tcl command file>* to load and execute a command file. The command file can be previously recorded and/or hand-coded. The result from the command is returned to the display area on the *Console*. When this option is turned *on*, the command entry form is displayed each time the Verdi platform is invoked. By default, this option is disabled in the Post-Processing mode. In Interactive Debug mode,

the *Console* window is created (the option is grayed out) by default for monitoring simulation messages or to enter commands interactively.

- **Log GUI Command:** This option enables you to print the GUI operating Tcl commands in the *Console* window. The **Post-Simulation Mode** option is enabled by default and the **Interactive Mode** option is disabled by default.
- **Collapse All Sub-trees when Parent Node is Collected:** When this option is turned *on*, all subtree nodes are collapsed if their parent node is collapsed. When this option is turned *off*, the nodes are not collapsed. The default value of this option is *off*.
- **Disabled Messages:** The **Disabled Messages** section shows and enables messages that have already been turned *off*. Only messages that are turned off appear in the form by clicking the **Enable** button. Some messages can be disabled by checking the **Don't show this message again** check box that appears on message forms (for example, the Verdi *Exit* form displays this message). When turned *off*, the Verdi platform does not display this message again and immediately exits. To enable the message again, select the message from the form and click the **Enable** button. If a message has not been disabled, it does not appear in the list.
- **Mouse Button for Schematics, FSM and PowerMap**
  - **Enable Button 1 to Drag and Drop:** When this option is turned *on*, after you select one or more objects in *nSchema* and *nState* window, you can drag left mouse button from the selected object to do drag-and-drop operation. When this option is turned *off,* you can zoom or perform other mouse operations.The default value of this option is *off*.
- **Button Assignments:** Assign mouse button functions for *nSchema* and *nState* from **Button Assignments**. Button 1 is identified as the left mouse button and button 3 is identified as the right mouse button.

Automatic Save Page

Use this page to specify automatic saving options.

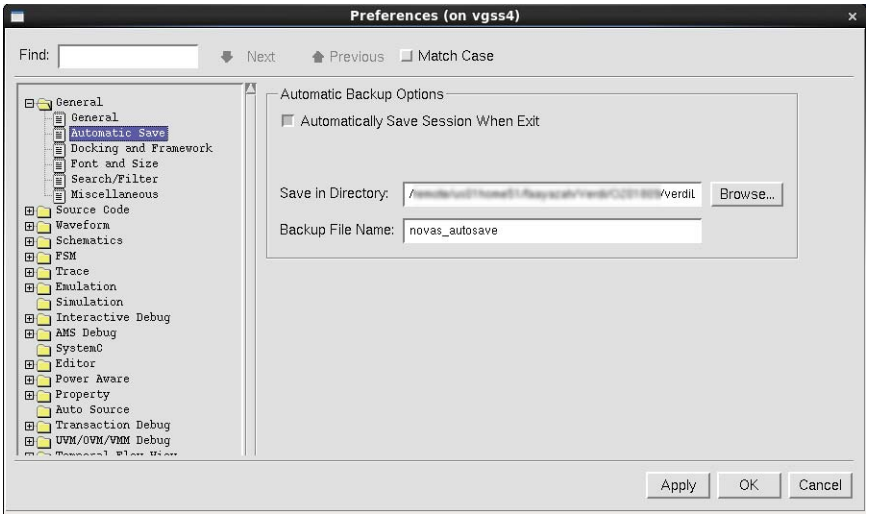


Figure: Automatic Save Page

The following options are available:

- **Automatically Save Session When Exit:** When this option is turned *on*, the last session gets automatically saved when the GUI is closed. The default value of this option is *on*.
- **Save in Directory:** Specify the directory to save the session files to. Enter the full path manually or click the **Browse** button to view the directory structure and select a location.
- **Backup File Name:** Specify the file name for the backup file. The default value in this field is *novas_autosave*.

Docking and Framework Page

Use this page to configure Docking and Framework options.

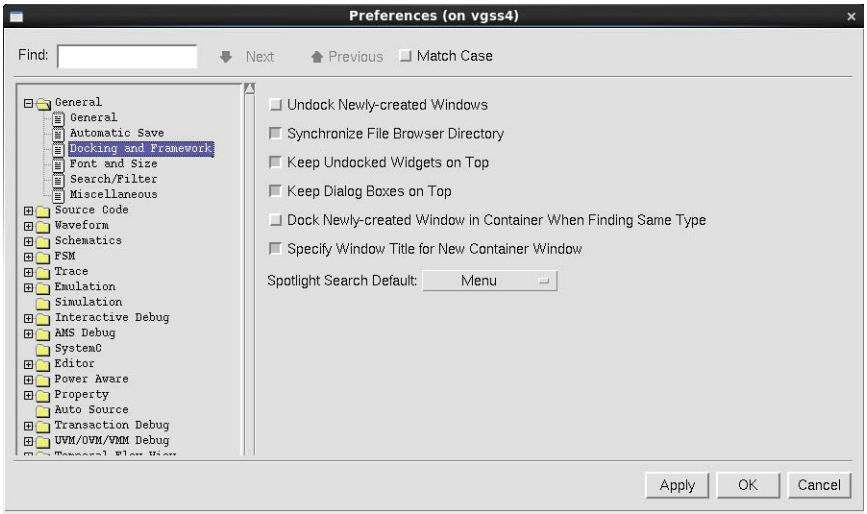


Figure: Docking and Framework Page

The following options are available:

- **Undock Newly-created Windows:** When this option is turned *on*, a newly created frame is displayed as a standalone window. Click the **Dock** icon to attach the frame to the main window. When this option is turned *off*, newly created frames are automatically docked to the main window in the default location. The default value of this option is *off*.
- **Synchronize File Browser Directory:** When this option is turned *on*, the file browser directory is synchronized to the previously opened file browser directory. When this option is turned *off*, the file browser directory in each form operates independently. The default value of this option is *on*.

NOTE: When this option is turned *off*, *Open Dump File*, *Save Signal*, and *Restore Signals* forms in *nWave* maintain their own file directory browser path in the *novas.rc* resource file.

In a standalone *nWave* pane (that is opened by `verdi -nWave` or `nWave` command lines), the **Synchronize File Browser Directory** option is displayed in the **Waveform -> General** page.

- **Keep Undocked Widgets on Top:** When this option is turned *on*, undocked widgets are placed in front of the main window. When this option is turned

off, undocked widgets are placed behind the main window after a random click on the main window. The default value of this option is *on*.

- **Keep Dialog Boxes on Top:** When this option is turned *on*, the specified dialog boxes are placed on top. When this option is turned *off*, the specified dialog boxes are not placed on top. The default value of this option is *on*.

Font and Size Page

Use this page to configure Font and Size options.

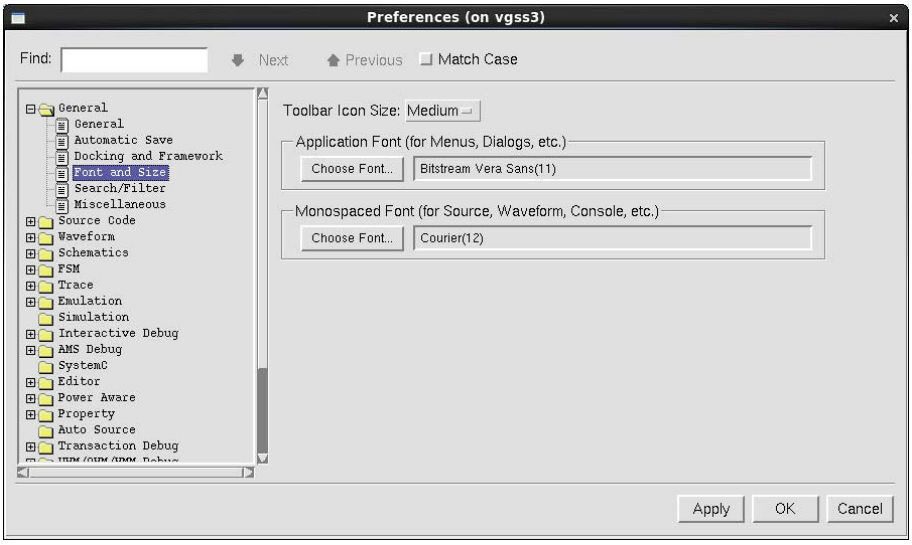


Figure: Font and Size Page

The following font and size options are available:

- **Toolbar Icon Size:** Specify the settings for the size of the toolbar icons by selecting from **Large, Medium**, or **Small**.
- **Application Font (For Menus, Dialogs, etc.):** Specify the fonts available in the system. By default, the font available is **Bitstream Vera Sans**, size **11**. This field cannot be edited directly. You need to click **Choose Font** to open the *Select Application Font* form which provides the flexibility to set the font, size, and so on, as illustrated in the following figure:

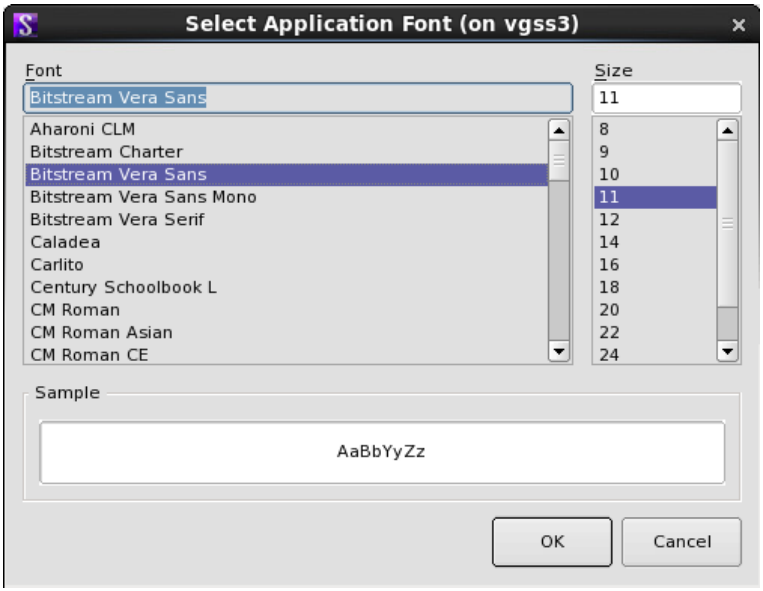


Figure: Select Application Font

The following options and fields are available in the *Select Application Font* form:

- **Font:** Specify the font by selecting any one of the font listed. By default, the font selected is *Bitstream Vera Sans*.
- **Size:** Specify the font size. By default, *11* is selected. You can select size from 8 to 26.
- **Sample:** Displays an example of Font and the Size selected.

- **Monospaced Font (for Source, Waveform, Console, etc.):** Specify all the fonts and sizes of the monospaced font used in the console, reports, and so on. By default, the font available is **Courier(12)**. This field cannot be edited directly. You need to click **Choose Font** to open the *Select Monospaced Font* form which provides the flexibility to set up the font and font size, as illustrated in the following figure:

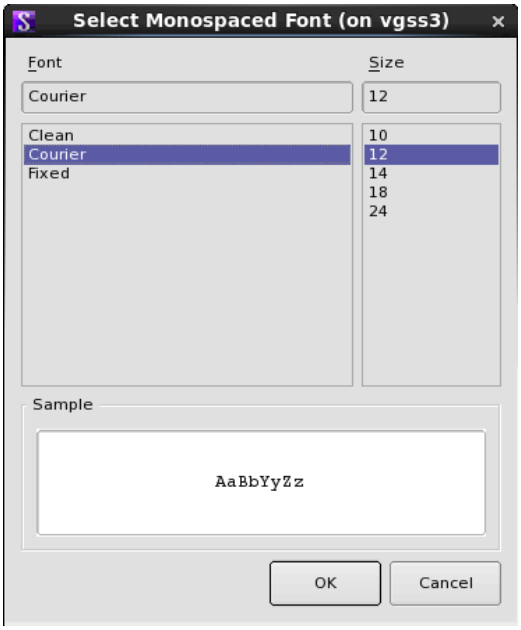


Figure: Select Monospaced Font Form

The following options and fields are available in the *Select Monospaced Font* form:

- **Font:** Specify the font by selecting any one of the font listed. By default, the font selected is *Courier*.
- **Size:** Specify the font size. By default, *12* is selected.
- **Sample:** Displays an example of Font and Font Size selected.

NOTE: The monospaced fonts needs to have equal width for each letter, therefore the selection of monospaced fonts differs with the Application fonts.

In the *nWave* window, to enlarge the icon's size and adjust the distance automatically between the **Language Type**, **Signal Type**, and **Signal Name**, specify the font size in the *Monospaced Font* form. For example, select *24* as the font size. The **Language Type**, **Signal Type**, and **Signal Name** are enlarged and adjusted as illustrated in the following figure:

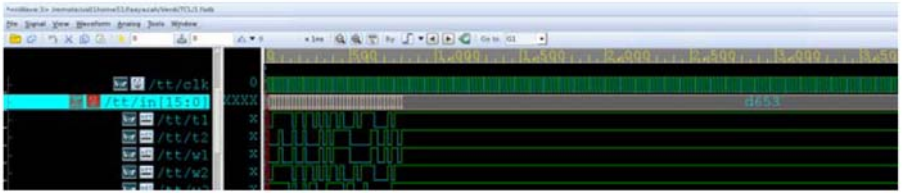


Figure: Enlarged Language Type, Signal Type, and Signal Name

Search/Filter Page

Use this page to specify the search and filter options.

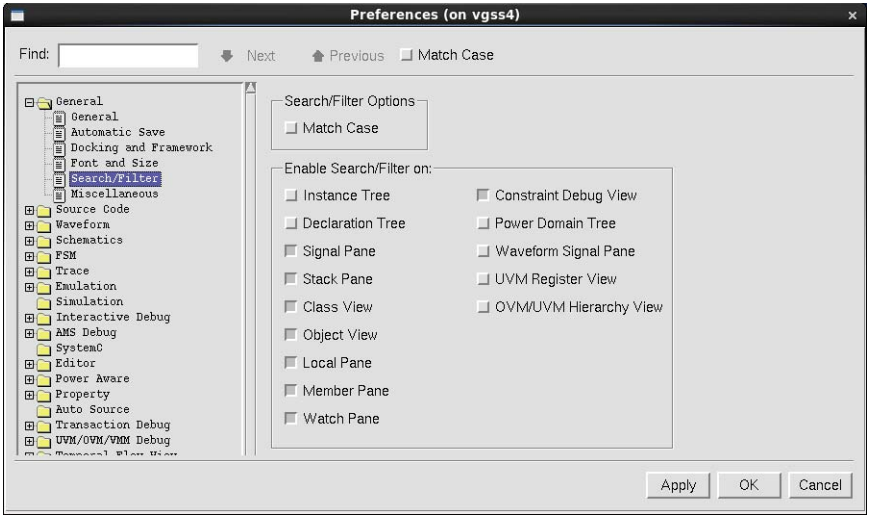


Figure: Search/Filter Page

The following option is available in the **Search/Filter** section:

- **Match Case:** When this option is turned on, the search and filter is case-sensitive. When this option is turned off, case is not considered during a search. The default value of this option is *off*.
- **Enable Search/Filter on:** Select the tabs/panes from the list in which the search and filter must be enabled.

Miscellaneous Page

Use this page to specify hierarchical path copy settings.

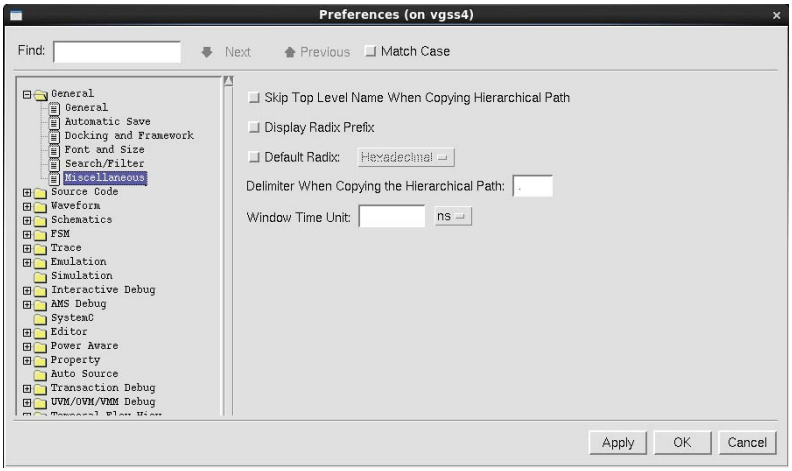


Figure: Miscellaneous Page

The following options are available:

- **Skip Top Level Name When Copying Hierarchical Path:** When this option is turned *on*, the top level name is ignored when the full hierarchical path is copied from *nSchema* to other windows. When this option is turned *off*, the top level name is included. The default value of this option is *off*.
- **Display Radix Prefix:** This option allows you to control the display of radix base specifier for all views (*nSchema*, *nTrace*, signal list, *nWave*, Local pane, Watch pane, Member pane, UVM Register view, and UVM Resource view).

  The default value of this option is *off*. When this option is turned *on*, the signal value format begins with preceding apostrophe (‘), followed by radix base specifier, and then followed by signal value, as shown in the following figure:

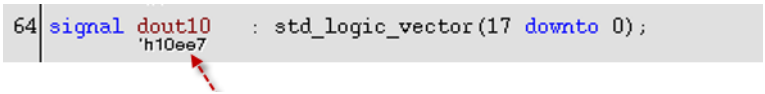


Figure: Signal Value With Radix Base Specifier

- **Default Radix:** When this option is turned *on*, it allows you to set global radix for the data types in all views (*nSchema*, *nTrace*, signal list, *nWave*, Local Pane, Watch Pane, Member Pane, UVM Register view, and UVM Resource view).
- **Delimiter When Copying the Hierarchical Path:** Specify the delimiter of the full hierarchical name in this field.

- **Window Time Unit:** Specify the window time unit in **fs**, **ps**, **ns**, **us**, **ms**, or **s**. The time unit in the toolbar icon for *nWave*, *nTrace*, *TFV*, and *Console* changes accordingly. By default, the value is empty.

Source Code Folder

The Verdi platform configures the default text color settings for language syntax to make the source code easy to read. Both the text colors and the text font are configurable. These default settings are saved upon exiting and used automatically the next time the Verdi platform runs.

Color Page

Use this page to associate different colors and fonts with constructs in the *nTrace* source code pane.

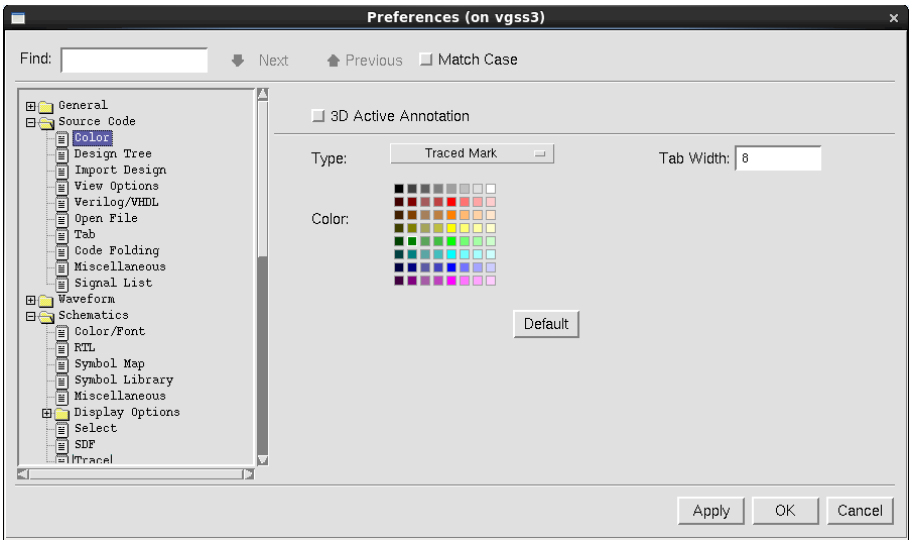


Figure: Color Page

The following color options are available:

- **3D Active Annotation:** When the **Active Annotation** command (**Source -> Active Annotation**) is enabled and the **3D Active Annotation** option is also enabled, the simulation results (signal and bus values) are displayed in 3D in the source code pane.
- **Type:** Select the type of object for which you need to change the color attributes. Following are the options available (only one of these options can be selected at a time):
  - **3D Light:** The color of the 3D light. Default = ID_WHITE.
  - **3D Shadow:** The color of the 3D shadow. Default = ID_BLACK.

- **Alias:** The color of the alias. Default = ID_BLUE5.
- **All:** The color of all variables.
- **Annotation:** The color of the annotation. Default = ID_BLACK.
- **Annotation Shadow:** The color of the annotation shadow. Default = ID_WHITE.
- **Architecture:** The color of the architecture objects. Default = ID_BLACK.
- **Assertion Identifier**: The color of the assertion identifier. Default = ID_PURPLE2.
- **Background:** The color of the background. Default = ID_GRAY5.
- **Code in Parentheses:** The color of the source code included within the selected parentheses in the *Source Code* pane or the *Watch Expression* form. The default is cyan.
- **Comment:** The color of regular comment types, such as // or /* */. The default is ID_GRAY1.
- **ComputedSignal:** The color of computed signals. Default = ID_GREEN5.
- **Constant:** The color of constants. Default = ID_BLACK.
- **Entity:** The color of entities. Default = ID_BLACK.
- **Folded Line Number:** The color of the folded line number. Default = ID_RED.
- **Function Annotation:** The color of the function annotation. Default = ID_BLUE2.
- **Generic:** The color of generic objects. Default = ID_BLUE5.
- **HVL Drive:** The color of the HVL drive. Default = ID_GREEN3.
- **HVL Drive and Load:** The color of the HVL drive and load. Default = ID_BLUE3.
- **HVL Identifier:** The color of the HVL identifier. Default = ID_BLACK.
- **HVL Load:** The color of the HVL load. Default = ID_YELLOW3.
- **InOut Signal:** The color of input/output signals. Default = ID_RED2.
- **Input Signal:** The color of input signals. Default = ID_RED2.
- **Instance:** The color of instances. Default = ID_BLACK.
- **Key Word:** The color of keywords. Default = ID_BLUE5.
- **Line Number:** The color of the line number. Default = ID_BLACK.
- **Macro Value:** The color of macro values. Default = ID_BLACK.

- **Matching Parenthesis:** The color of the corresponding parenthesis for the parenthesis at the current caret cursor position in the *Source Code* pane or the *Watch Expression* form. Default = ID_YELLOW5.
- **Module:** The color of modules. Default = ID_BLACK.
- **Operator:** The color of operators. Default = ID_BLACK.
- **Others:** The color of other variables. Default = ID_BLACK.
- **Output Signal:** The color of output signals. Default = ID_RED2.
- **Package:** The color of packages. Default = ID_BLUE5.
- **Parameter:** The color of parameters on the source code frame. Default = ID_BLACK.
- **Parameter Annotation:** The color of the parameter annotation. Default = ID_BLACK.
- **Port**: The color of ports. Default = ID_BLACK.
- **Signal:** The color of signals. Default = ID_RED2.
- **Special Comment:** The color of special comments, such as:

  //synopsys_translate_off //synopsys_translate_on
  //novas_translate_off //novas_translate_on
  //psl
  //0in
  Default = ID_GREEN2.

- **Traced Mark:** The color of the traced mark. Default = ID_GREEN2.

- **Color:** Select (left-click) the target color in the color matrix.
- **Default:** Returns the selected *Type* to its default *Color* and *Font*.
- **Tab Width:** Customizes the number of tab characters for the Source Code and *nEditor* panes.

Design Tree Page

Use this page to set the text font and the tree node types for the design browser pane.

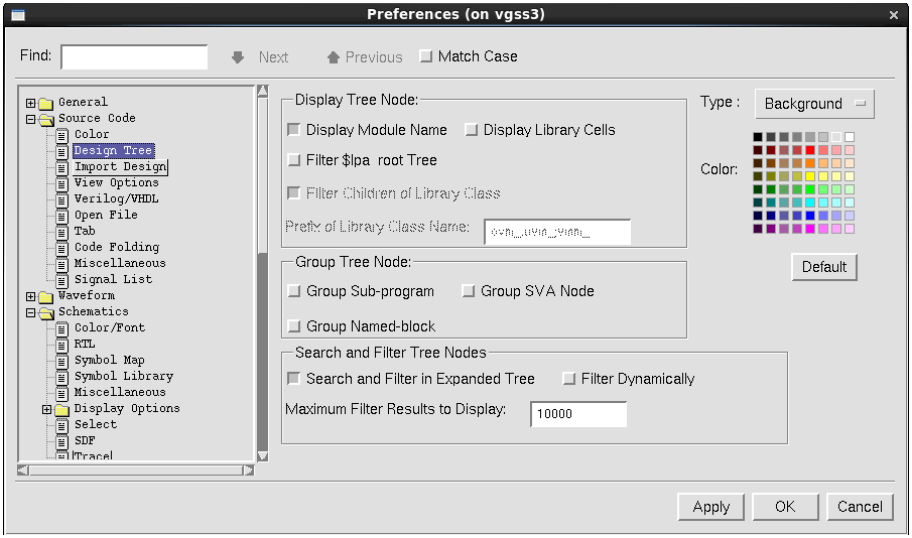


Figure: Design Tree Page

The following options are available in the **Display Tree Node** section:

- **Display Module Name:** When this option is turned *on*, the module name is displayed. The default value of this option is *on*.
- **Display Library Cells:** When this option is turned *on*, all library cells for every scope are displayed in the design browser pane. When this option is turned *off*, the **Display Library Cells** command is enabled in the design browser so library cells can be displayed for selected scopes. The default value of this option is *off*.
- **Filter $lpa_root Tree:** When this option is turned *on*, the Low Power Assertion (LPA) root tree is hidden. When this option is turned *off*, the Low Power Assertion (LPA) root tree is displayed. The default value of this option is *off*.
- **Filter Children of Library Class:** When this option is turned *on*, the children of library classes are not displayed in the design browser pane. When this option is turned *off*, children of library classes are displayed in the design browser. The default value of this option is *on*.
- **Prefix of Library Class Name:** Specify the prefix for library class names.

The following options are available in the **Group Tree Node** section to control grouping of specific types of nodes in the *nTrace* design browser pane:

- **Group SVA Node:** When this option is turned *on*, SVA assertion, cover, or assume related nodes are collected into the SVA Set node. The default value of this option is *off*.
- **Group Sub-program:** When this option is turned *on*, function or task related nodes are collected into the Subprogram Set node. The default value of this option is *off*.
- **Group Block:** When this option is turned *on*, named block nodes are collected into the Name block Set node. The default value of this option is *off*.

The following options are available in the **Search and Filter Tree Nodes** section:

- **Search and Filter in Expanded Tree:** When this option is turned *on*, searching and filtering can be performed on the current expanded tree. When this option is turned *off*, searching and filtering can be performed on the complete tree. The default value of this option is *on*.
- **Filter Dynamically:** When this option is turned *on*, the design browser pane is dynamically filtered as the search string is entered. The default value of this option is *off*. This option is enabled only when the **Search and Filter in Expanded Tree** option is turned *on*.
- **Maximum Filter Results to Display:** The maximum number of filtered results to be displayed can be entered. The default value is *10000*.
- **Type:** Select the type of object for which the color attributes need to be changed. The option is as follows:
  - **Background:** the background color.
- **Color:** Select (left-click) the target color in the color matrix.
- **Default:** Click this button to return the selected *Type* to its default *Color* and *Font*.

Import Design Page

Use this page to specify the global settings associated with library directories and library files using **Library Directories (-y)**, **Library Files (-v)**, and **Other Options** fields when importing a Verilog design file by selecting **File -> Import Design**, clicking the *From File* page, and selecting **Verilog** in the **Language** list.

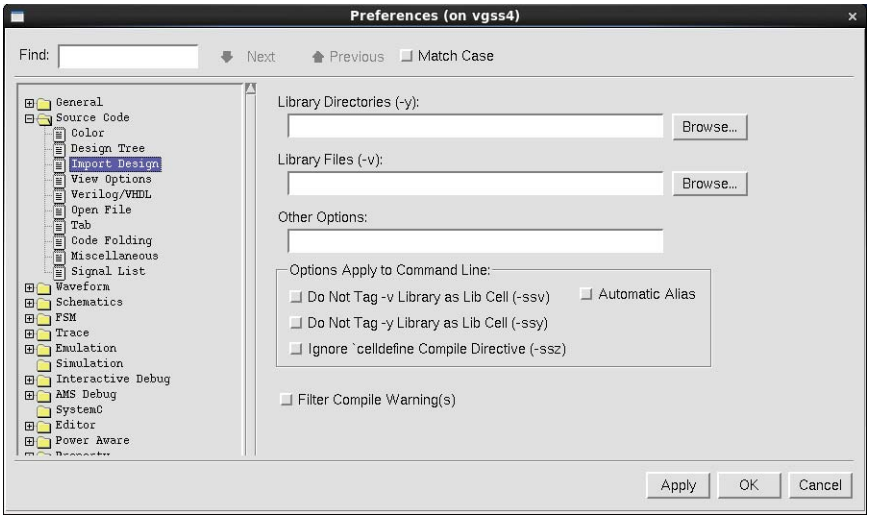


Figure: Import Design Page

The following options are available in the **Options Apply to Command Line** section:

- **Do Not Tag -v Library as Lib Cell (-ssv):** When this option is turned *on*, *nTrace* does not automatically tag the library modules in the library file (-v) as library cells. When this option is turned *off*, *nTrace* automatically tags the library modules in the library file (-v) as library cells. The default value of this option is *off*.
- **Do Not Tag -y Library as Lib Cell (-ssy):** When this option is turned *on*, *nTrace* does not automatically tag the library modules in the library directory (-y) as library cells. When this option is turned *off*, *nTrace* automatically tags the library modules in the library directory (-y) as library cells. The default value of this option is *off*.
- **Ignore 'celldefine Compile Directive (-ssz):** When this option is turned *on*, *nTrace* ignores '*celldefine* compiler directives. When this option is turned *off*, *nTrace* does not ignore '*celldefine* compiler directives. The default value of this option is *off*.
- **Automatic Alias:** When this option is turned *on*, *nTrace* executes automatic aliasing or specifies the predefined alias file. When this option is turned *off*, *nTrace* does not execute automatic aliasing. The default value of this option is *off*.

Filter Compile Warning(s): When this option is turned *on*, warning messages are not listed on the *Compile* tab of the *Message* pane. When this option is turned

off, warning and error messages are listed in the *Message* pane. The default value of this option is *off*.

View Options Page

Use this page to specify viewing options in the *nTrace* source code pane.

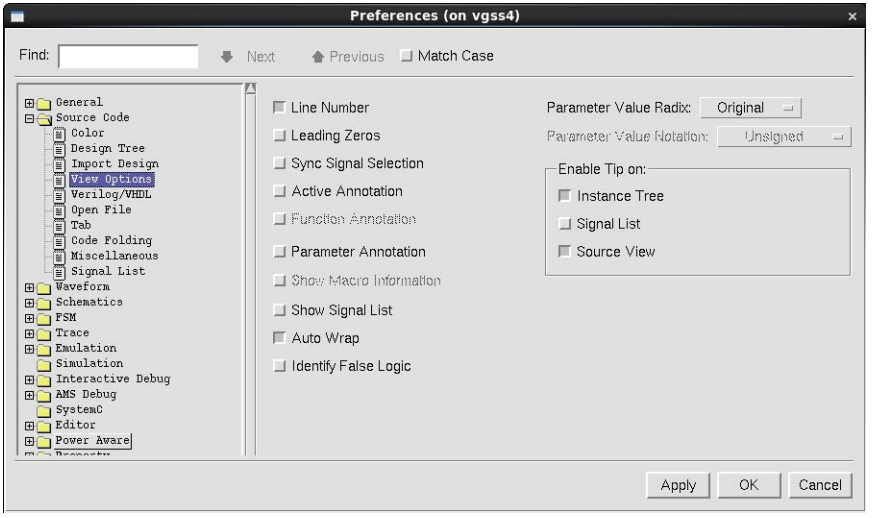


Figure: View Options Page

The following view options are available:

- **Line Number:** When this option is turned *on*, line numbers are displayed to the left of the *nTrace* source code pane.
- **Leading Zeros:** When this option is turned *on*, leading zeros are displayed for signal values in the *nTrace* source code pane when Active Annotation is enabled. The number of leading zeros depends on the format specified for the signal value. For changes to take immediate effect, select the **Source -> Leading Zeros** command in the *nTrace* window.
- **Sync Signal Selection:** When this option is turned *on*, the signal selection among windows can be synchronized. For example, if a signal is selected in waveform, source code, or schematic window, the associated signal in the other windows is also selected.
- **Active Annotation:** When this option is turned *on*, simulation results are annotated in *nTrace* by default.
- **Function Annotation:** After the **Active Annotation** option is turned *on*, the **Function Annotation** option can be turned *on*. When the **Function**

Annotation option is turned *on*, the return values of a given function are annotated in blue under the signal in the *nTrace* source code pane (unless the signal is in a testbench scope or a class scope). Refer to the **Source -> Function Annotation** command description in the *nTrace* chapter for details.

- **Parameter Annotation:** When this option is turned *on*, parameters are automatically annotated from the source code.

NOTE: When **-pvalue** or **-parameters** option is specified on the Verdi command line, the parameter annotation follows the value set by the command line option.

- **Show Macro Information:** When the **Parameter Annotation** option is turned *on*, the **Show Macro Information** option is enabled. When the **Show Macro Information** option is turned *on*, macro information is shown. When this option is turned *off*, macro information is not shown. The default value of this option is *on*.
- **Show Signal List:** When this option is turned *on*, the *Signal List* pane is shown automatically in the *nTrace* window. When this option is turned *off*, the *Signal List* pane is not shown automatically in the *nTrace* window. The default value of this option is *off*.
- **Auto Wrap**: When this option is enabled, the code in the *Source Code* pane is moved to the next line based on the view of the *Source Code* pane to improve the source code readability. That is, if the length of a single line of source code exceeds the view of the *Source Code* frame, then the code is moved to the next line based on the view of the *Source Code* pane. By default, this option is enabled.

NOTE: This option is dependent on the *Source Code* pane size. If the *Source Code* pane size is changed, then the code is moved to the next line automatically based on the pane size. The length of a line does not exceed the width of the *Source Code* pane.

- **Parameter Value Radix:** Set the parameter value to either **Binary, Octal, Decimal, Hexadecimal,** or **Original**. The default value of this option is **Original**.
- **Parameter Value Notation:** Set the parameter value notation to either **Unsigned, 1's Complement, 2's Complement,** or **Signed Magnitude**. The default value of this option is **Unsigned**.

The following options are available in the **Enable Tip on** section:

- **Instance Tree:** When this option is turned *on*, tips are shown automatically when you hover the mouse pointer on the instances in the *Instance* frame. The default value of this option is *on*.
- **Signal Pane:** When this option is turned *on*, tips are shown automatically when you hover the mouse pointer on the signals in the *Signal List* frame. The default value of this option is *off*.
- **Source View:** When this option is turned *on*, tips are shown automatically when you hover the mouse pointer on the source code in the *Source Code* pane. The default value of this option is *on*.

Verilog/VHDL Page

Use this page to set the reference simulator for naming styles in a Verilog or a VHDL design.

NOTE: If the `-ssf` command line option is used to specify an FSDB file and the simulator type is specified in the FSDB, the simulator type is set automatically. If the simulator type is not specified by the `-ssf` option, `-vhdlSimType` or `-veriSimType` command line options can be used to set VHDL or Verilog simulator type. If none of the `-ssf`, `-vhdlSimType`, or `-veriSimType` options are specified, ModelSim (for VHDL) or NC-SIM (for Verilog) is set as the default.

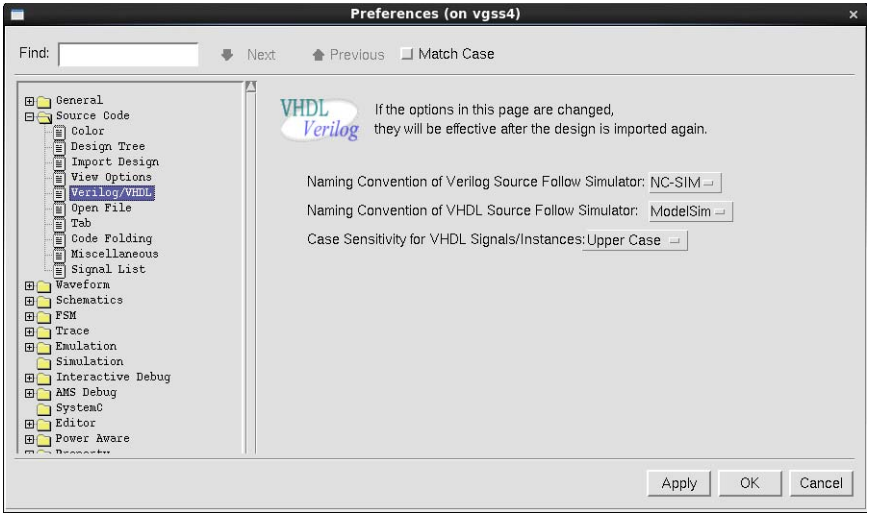


Figure: Verilog/VHDL Page

The following options are available:

- **Naming Convention of Verilog Source Follow Simulator:** Apply the naming style of the specified simulator on the Verilog source code by selecting **NC-SIM** or **VCS**. The default value of this option is **NC-SIM**.
- **Naming Convention of VHDL Source Follow Simulator:** Apply the naming style of the specified simulator on the VHDL source code by selecting **ModelSim** or **NC-VHDL**. The default value of this option is **ModelSim**. The purpose is to support different VHDL GENERATE naming conventions between **ModelSim** and **NC-VHDL**. For example,

  In *ModelSim*, the naming of the generated block is shown as follows:

  ```
  system(blk)
        |
        +- gen_block__0
         +- gen_block__1
  ```

  In *NC-VHDL*, the naming of the generated block is shown as follows:

  ```
  system(blk)
        |
        +- gen_block(1)
         +- gen_block(1)
  ```

  GENERATE Declaration:

  ```
  entity system is
  end system;
  architecture blk of system is
  ...
  begin
    gen_block: For I in 0 To 1 Generate
    ...
    end Generate;
  end blk;
  ```

- **Case Sensitivity for VHDL Signals/Instances:** Specify the case for VHDL signals and instances in the design by selecting **Upper Case** or **Original Case**. When **Original Case** is selected, the VHDL design is displayed in the original case. When **Upper Case** is selected, upper case for the VHDL design is displayed. The default value of this option is **Upper Case**. When the setting is changed, it is effective after the design is loaded or imported again.

Open File Page

Use the *Open File* page to specify the C/C++ file types, such as *.c; *.cpp; *.cxx; *.cc; *.h; *.hpp; *.hxx; *.hh files. If the C/C++ file type is not specified in the

Preferences page, the files are displayed without the syntax color in the *Source Code* window, although you import a design.

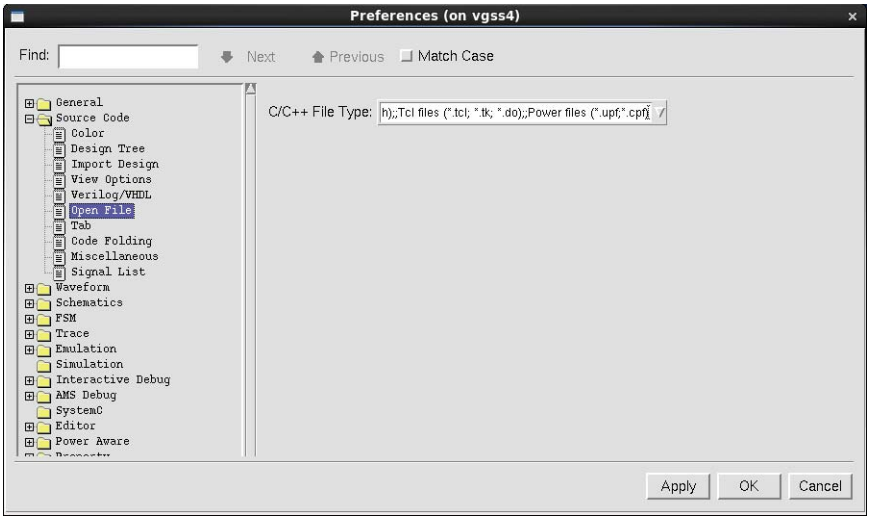


Figure: Open File Page

X-propagation Page

Use this page to specify the settings for X-propagation.

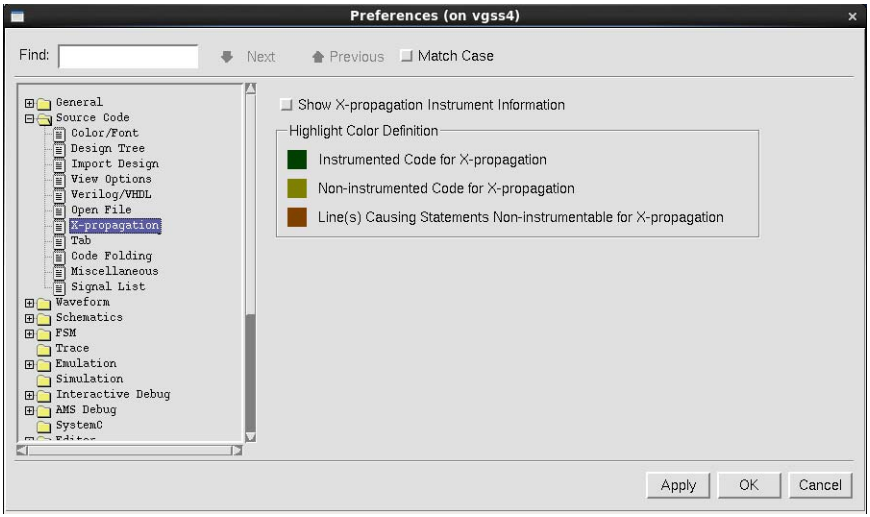


Figure: X-propagation Page

The following option is available:

- **Show X-propagation Instrument Information:** When this option is turned *on*, colors are added to the source code frame to represent X-propagation information as indicated in the **Highlight Color Definition** section. Only the lines blocking X-propagation are highlighted in the source code pane. The other colors are shown in the line number column only. The default value of this option is *off*.

NOTE: The **View -> Line Number** toggle command on the *nTrace* menu or the **Line Number** option on the **Source Code -> View Options** page of the *Preferences* form must be turned *on* before turning *on* the **Show X-propagation Instrument Information** option.

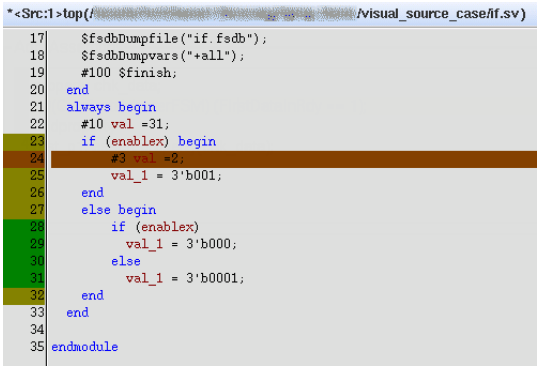


Figure: Example of Colorization

Tab Page

Use this page to specify the source code pane tab settings.

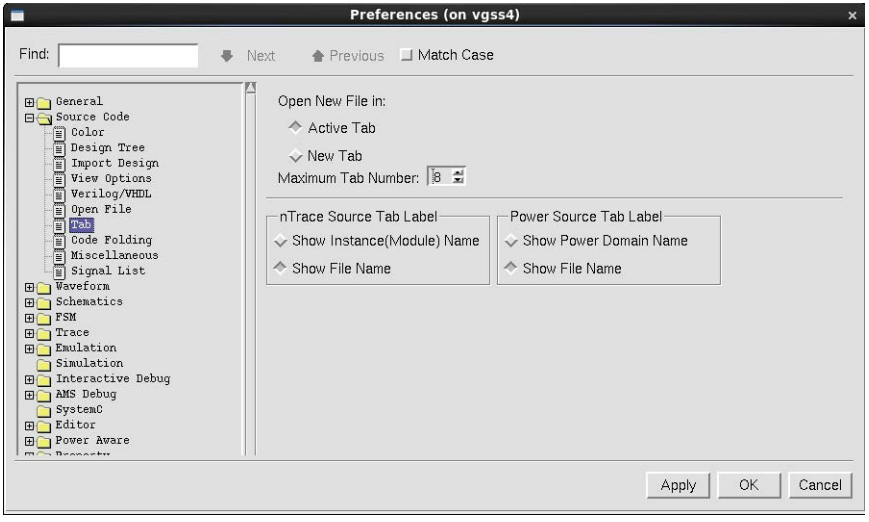


Figure: Tab Page

Select one of the following options for the **Open New File in** option (only one option may be selected at a time):

- **Active Tab:** When this option is turned *on*, source files that do not exist in the current display are opened in the current active tab. The default value of this option is *on*.

- **New Tab:** When this option is turned *on*, source files that do not exist in the current display are opened in a new tab. The default value of this option is *off*.

Maximum Tab Number: Specify the maximum number of tabs that may be opened in a session. The default value of this option is 8.

The following options are available in the **nTrace Source Tab Label** section (only one option may be selected at a time):

- **Show Instance (Module) Name:** When this option is turned *on*, the source code tabs are named with the active instance (module) name. The default value of this option is *off*.
- **Show File Name:** When this option is turned *on*, the source code tabs are named with the current file name. The default value of this option is *on*.

The following options are available in the **Power Source Tab Label** section (only one option may be selected at a time):

- **Show Power Domain Name:** When this option is turned *on*, the power source code tabs are named with the active power domain name. The default value of this option is *off*.
- **Show File Name:** When this option is turned *on*, the power source code tabs are named with the current file name. The default value of this option is *on*.

Code Folding Page

Use this page to specify source code folding settings.

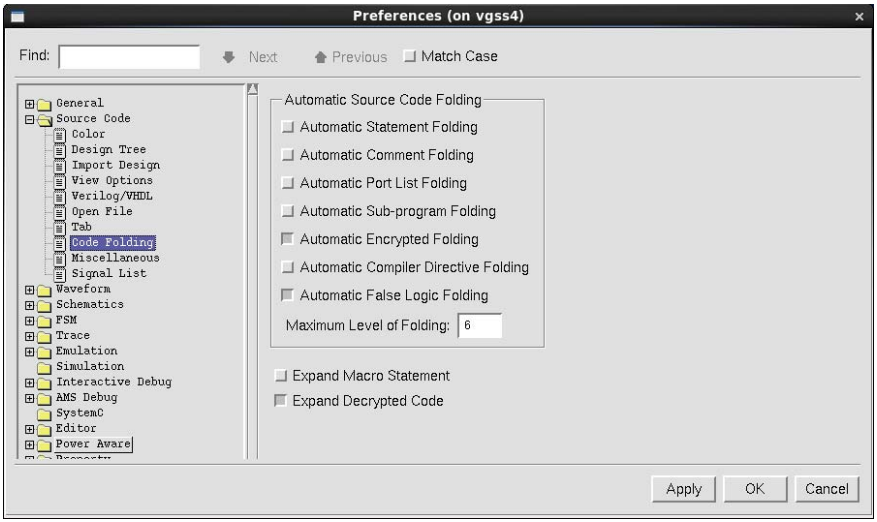


Figure: Code Folding Page

The following options are available in the **Automatic Source Code Folding** section:

- **Automatic Statement Folding:** When this option is turned *on*, the **Statement Folder** command and its subcommands under the **View** menu in *nTrace* appear and the expanded code symbol is created and shown automatically in the *nTrace* source code pane indicator area for certain code statements (including `if`, `else`, `case`, `always`, `for`, `while`, `loop`, and `process`). Click the expanded code symbol to collapse the code and display the collapsed code symbol . The default value of this option is *off*.

NOTE: Automatic statement folding is not supported for VHDL and Verilog generate block.

- **Automatic Comment Folding:** When this option is turned *on*, the **Comment Folder** command and its subcommands under the **View** menu in *nTrace* appear and the expanded code symbol is created and displayed automatically in the *nTrace* source code frame indicator area for comments. Click the expanded code symbol to collapse the code and display the collapsed code symbol . The default value of this option is *off*.
- **Automatic Port List Folding:** When this option is turned *on*, the **Port List Folder** command and its subcommands under the **View** menu in *nTrace* appear and the expanded code symbol is created and displayed automatically in the *nTrace* source code pane indicator area for ports or port

instance lists. Click the expanded code symbol  to collapse the code and display the collapsed code symbol  . The default value of this option is *off*.

- **Automatic Sub-program Folding:** When this option is turned *on*, the **Source Code Folder** command and its subcommands under the **View** menu in *nTrace* are added and the expanded code symbol  is created and displayed automatically in the *nTrace* source code pane indicator area for functions, tasks, or procedures. Click the expanded code symbol  to collapse the code and display the collapsed code symbol  . The default value of this option is *off*.
- **Automatic Encrypted Folding:** When this option is turned *on*, encrypted code is folded with the collapsed code symbol  displayed in the line number pane. When this option is turned *off*, encrypted code is fully displayed. The default value of this option is *on*.
- **Automatic Compiler Directive Folding:** When this option is enabled, the **Compiler Directive Folder** and its sub-commands under the **View - > Source Code Folder** menu in *nTrace* are added. This option is enabled to fold unmatched compiler directive branches with multiple lines. Following are the conditional compilation options that are supported:
  - `ifdef
  - `else
  - `elsif
  - `endif
  - `ifndef
- **Automatic False Logic Folding:** The **False Logic Folder** command and its sub-commands appear in the **View - > Source Code Folder** menu only when the **Automatic False Logic Folding** option in the **Source Code -> Code Folding** page of the *Preferences* form (invoked using **Tools -> Preferences**) and Identify False Logic (invoked using **View -> Identify False Logic** command) is enabled; the source code folding occurs for false logic in *nTrace Source Code* frame. The **Automatic False Logic Folding** option is enabled by default and **Identify False Logic** option is disabled by default.

NOTE: When the false branch changes in the source, then the folder is updated automatically. The folder appears collapsed by default.

- **Maximum Level of Folding:** Specify the maximum level of folding in the text field. The default value of this option is 6.

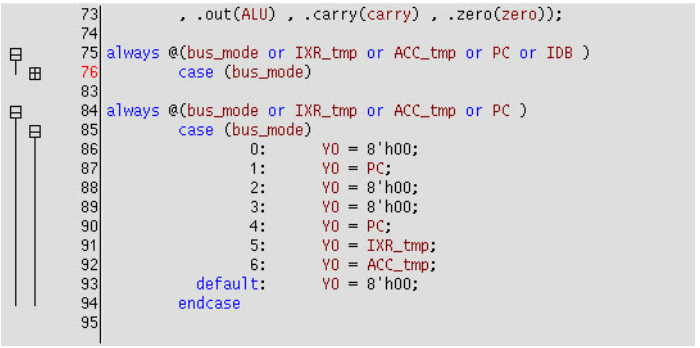


Figure: Example of Collapsed and Expanded Source Code Folding

Expand Macro Statement: When this option is turned *on*, all macro statements in the *nTrace* source code frame are expanded. When this option is turned *off*, the macro statements are collapsed. This option is synchronized with the **Source -> Expand Macro** command in *nTrace*.

Miscellaneous Page

Use this page to specify options for miscellaneous searching actions in *nTrace*.

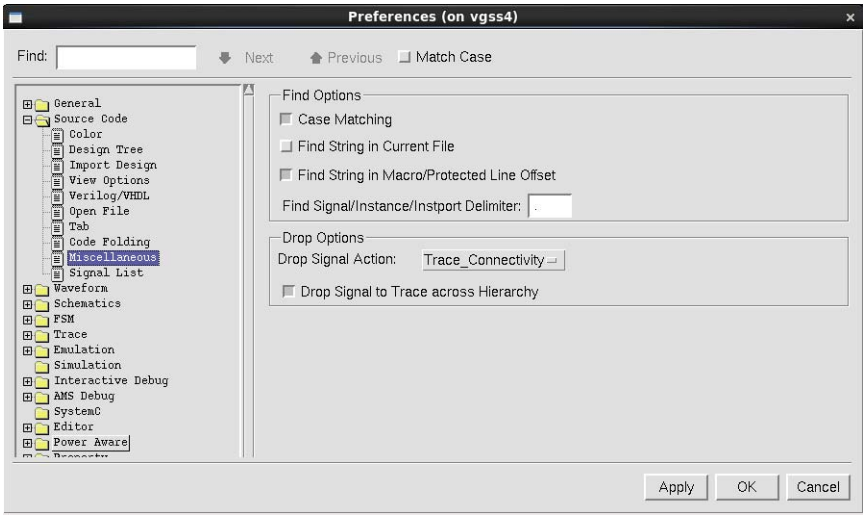


Figure: Miscellaneous Page

The following options are available in the **Find Options** section:

- **Case Matching:** When this option is turned *on*, only those instances in which the capitalization matches the text typed in the *Pattern* text field are

found. If the **Case Matching** option is turned *off*, case-sensitivity is not relevant for the text typed. The default value of this option is *on*.

This option affects the default search type of the **Find String** window and the search type of the **Find Previous/Next** toolbar commands in *Text Viewer* and *nEditor* panes.

NOTE: Changing the **Match Case** option in the **Source -> Find String** form does not change the preference setting.

- **Find String in Current File:** When this option is turned *on*, the search string behavior in *nTrace* is limited to only the current file. The searching does not go across other active scopes.
- **Find String in Macro/Protected Line Offset:** When this option is turned *on*, *nTrace* searches through the expanded macro and protected code. When this option is turned *off*, *nTrace* searches from the design file and skips the results in the expanded macro and protected code. The default value of this option is *on*.

NOTE: The source code files should be kept even if the source code files are compiled to a library because the search starts from the original source file.

- **Find Signal/Instance/Instport Delimiter:** Specify the delimiter to use when searching for signals, instances, or instance ports in *nTrace* and *nSchema* windows. The delimiter setting is saved to or loaded from the *.rc* file.

The following option is available in the **Drop Options** section:

- **Drop Signal Action:** Configure the drop settings for signals that are dropped in source code frame from other Verdi frames; *nTrace* does the drag and drop action accordingly. Four types of actions are available: **Trace Connectivity, Trace Driver, Trace Load,** and **Jump Declaration**. The default value of this option is **Trace Connectivity**.
- **Drop Signal to Trace across Hierarchy:** When this option is turned *on* and a signal is dropped in *nTrace* from any other window, tracing driver, load, or connectivity (as specified in the Drag and Drop Signal Action option) crosses hierarchies. When this option is turned *off*, tracing occurs in the current scope and does not cross hierarchies.

Signal List Page

Use this page to specify fonts and colors in the *Signal List* pane of the *nTrace* window.

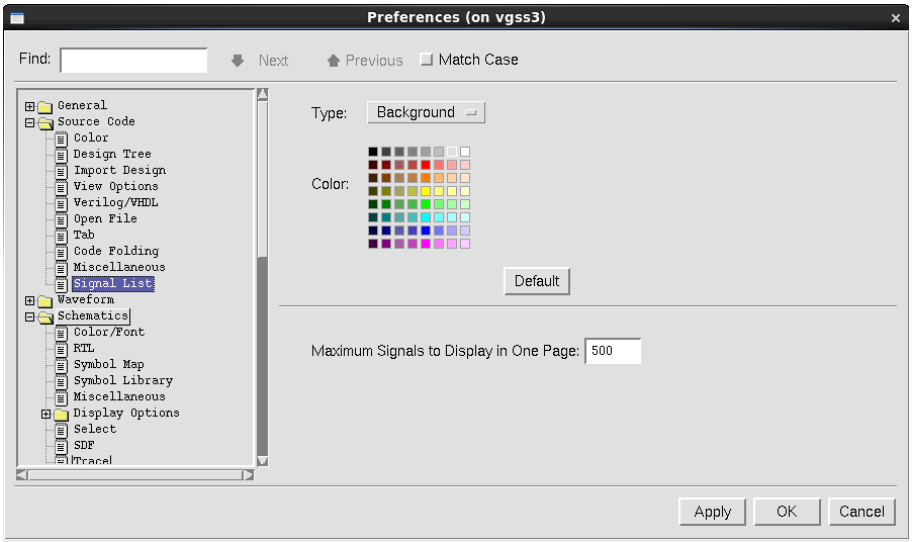


Figure: Signal List Page

The **Signal List** page includes the following options and field:

- **Type:** Select the type of object in this field for which you need to change the color attributes. The following options are available (only one of these options can be selected at a time):
  - **Foreground:** The foreground color of the current *Signal List* pane of the *nTrace* window. The default is ID_BLACK.
  - **Background:** The background color of the current *Signal List* pane of the *nTrace* window. The default is ID_GRAY.
  - **Highlighted:** The highlighted color of the current *Signal List* pane of the *nTrace* window. The default is ID_BLUE.
- **Color:** Select by left-clicking the target color in the color matrix.
- **Default:** When this option is enabled, the selected **Type** returns to its default *Color* and *Font*.
- **Maximum Signals to Display in One Page:** Specify the maximum number of signals to be displayed in each page of the *Signal List* frame. The default value is *500*. The minimum value is *50*.

Waveform Folder

The **Waveform** folder includes the **General** page, the **View Options** folder, the **Value System** page, the **Color/Pattern** folder, the **Default Value** folder, and the **Extended VCD** page.

General Page

Use this page to set various loading options for *nWave*.

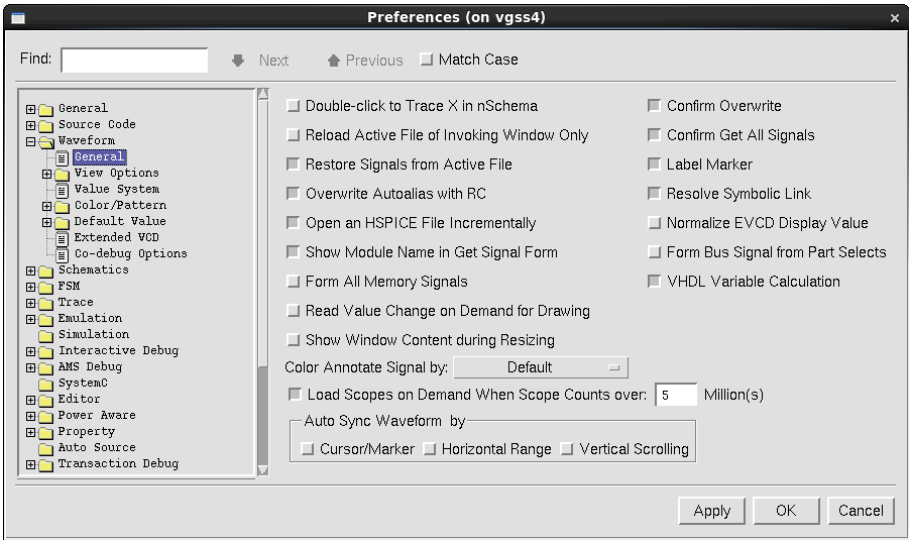


Figure: General Page

The following options are available:

- **Double-click to Trace X in nSchema:** When this option is turned *on*, double-clicking an X value in the waveform pane invokes the schematic window. When this option is turned off, the schematic is traced depending on the **Active Driver Detention** option.The default value of this option is *off*.
- **Reload Active File of Invoking Window Only:** When this option is turned *on*, only the active FSDB file of the *nWave* frame that invoked the **File -> Reload** command is reloaded. When this option is turned *off*, all active FSDB files in all *nWave* frames are reloaded.
- **Restore Signals from Active File:** When this option is turned *on*, waveforms are restored from the active file. When this option is turned *off*,

all files specified in the signal file are restored. The default value of this option is *on*.

- **Overwrite Autoalias with RC:** When this option is turned *on*, it overwrites the autoalias of the restored signals included with the resource file (`*.rc`). The default value of this option is *on*.
- **Open an HSPICE File Incrementally:** When this option is turned *on*, a HSPICE format file is opened incrementally.
- **Show Module Name in Get Signals Form:** When this option is turned *on*, the module name is displayed in the design tree pane of the *Get Signals* form.
- **Form All Memory Signals:** When this option is turned *on*, all 2D-arrays are formed as signals. The default is *off*.
- **Read Value Change on Demand for Drawing:** When this option is turned *on*, signal value changes are loaded on demand instead of loading all signal value changes simultaneously. When the FSDB file is opened and signals are selected, only dense blocks and necessary value changes in the waveform pane are loaded. This reduces memory and time consumption.The default value of this option is *off*.
- **Show Window Content during Resizing:** When this option is turned *on*, the *nWave* pane is always redrawn when the window is resized. When this option is turned *off*, the *nWave* frame is only redrawn when the mouse button is released. The default value of this option is *on*.
- **Color Annotate Signal by:** Annotate signals by different colors. Signals may be categorized by language (**Verilog/VHDL/SystemC**), IO type (**Input/Output/InOut**), or Net type (**Net/Register**). If the signal does not belong to any types, **Default** color is applied.
- **Load Scopes on Demand When Scope Counts over 'n' Million(s):** When this option is turned *on*, it loads the FSDB scope on demand to improve the performance of opening the FSDB file when the opened file has more than n millions scopes. By default, the option is enabled with five million scopes. A message appears mentioning that some GUI operations at runtime may be impacted and may slow-down.

  Select the **Don't show this message again** option if you do not want to see this message again.
- **Auto Sync Waveform by:** Specify the option to synchronize the views of multiple *nWave* frames by selecting one of the following options:
  - **Cursor/Marker:** When this option is turned *on*, the window's view and the **Cursor**/**Marker** movement are reflected in the other windows. The default value of this option is *off*.

  - **Horizontal Range:** When this option is turned *on*, the current visible range and the zoom ratio of multiple windows are synchronized.The default value of this option is *off*.
  - **Vertical Scrolling:** When this option is turned *on*, the vertical scrolling between the current window and other windows are synchronized. The default value of this option is *off*.
- **Confirm Overwrite:** When this option is turned *on,* a warning message is issued when a simulation file is being overwritten. When this option is turned *off*, *nWave* overwrites the simulation file without issuing any warning messages.
- **Confirm Get All Signals:** When this option is turned *on*, a *Confirmation* form is displayed when selecting the **Signal -> Get All Signals** command in *nWave*. When this option is turned *off*, the *Confirmation* form is not displayed and *nWave* gets all signals immediately when the **Get All Signals** command is invoked.
- **Label Marker:** When this option is turned *on*, the user-defined marker name is displayed in the waveform pane.
- **Resolve Symbolic Link:** When this option is turned *on,* an FSDB file is opened with the symbolic link resolved. The default value of this option is *on*.

  For example, when *verilog.fsdb -> 12345.fsdb* is loaded under */workdir* directory and this option is turned *on*, the file name is resolved to *12345.fsdb*; otherwise, it is resolved to *verilog.fsdb*.
- **Normalize EVCD Display Value:** When this option is turned *on*, the display values of EVCD are normalized.
- **Form Bus Signal from Part Selects:** When this option is turned *on*, if part-selects of a bus (such as, A[0:2], A[3:7]) are dumped to the FSDB file, *nWave* automatically reconstructs and displays the entire bus (such as, A[0:7]). When this option is turned *off*, the bus is not automatically reconstructed. This option is applied only for newly opened files. Overlapped (such as, A[0:3], A[3:7]), reversed continuity (such as, A[0:3], A[7:4]), and discontinuous (such as, A[0:3], A[5:7]) buses are not reconstructed. The default value of this option is *on*.
- **VHDL Variable Calculation:** When this option is turned *off*, the Verdi platform does not calculate VHDL variables. The default value of this option is *on*.

View Options Folder

The **View Options** folder includes the **Signal Pane** and **Value Pane** pages.

Signal Pane Page

Use this page to set the alignment and display preferences of signals in the *nWave* signal pane.

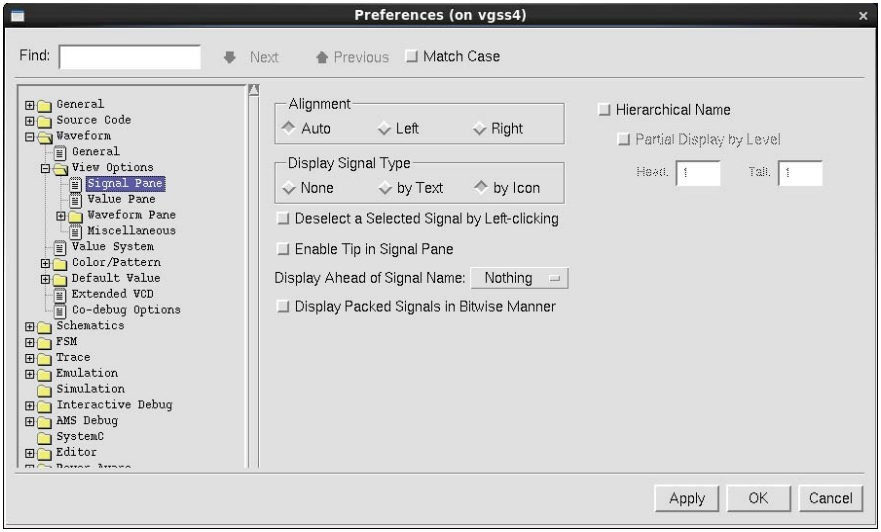


Figure: Signal Pane Page

The following options are available:

- **Alignment:** Specify the signal alignment in the signal pane by selecting one of the following options: **Auto**, **Left**, or **Right**. The default is **Auto**.

  The criteria for **Auto Alignment** are:

  a. With VHDL (or Mixed) designs, **Left Alignment** is applied.

  b. With Verilog designs, **Right Alignment** is applied.

  c. Without a design, **Left Alignment** is applied for VHDL (or Mixed mode) format dump files and **Right Alignment** is applied for other dump files.

- **Display Signal Type:** Specify the display mode of the signal type before each signal name in the signal pane by selecting one of the following options: **None**, **by Text**, or **by Icon**. The supported icons that can be displayed are: Input, Output, InOut, Net, Register, Message, and Transaction. The default value of this option is **None**.

- **Deselect a Selected Signal by Left-clicking:** When this option is turned *on*, a selected signal can be unselected by clicking the left mouse button. The default value of this option is *off*.
- **Enable Tip in Signal Pane:** When this option is turned *on*, tips are shown in the signal pane.
- **Display Ahead of Signal Name:** Specify the information to include with the signal name in the signal pane by selecting one of the following options: **Nothing, File Name,** or **File Number**.

  The default value of this option is **Nothing**.
- **Hierarchical Name:** When this option is turned *on*, each signal name appears with its full hierarchical path in the signal pane (that is, *CYCLE* is displayed as */system/CYCLE*). To view a partial signal name in the signal pane, turn the **Partial Display by Level** option *on* and specify how much of the beginning and end of the signal to display in **Head** and **Tail** text fields. The default value of this option is *off*.
- **Display Packed Signals in Bitwise Manner:** When this option is turned *on*, the packed structure signals appear in bitwise manner in both the Waveform pane and the Value pane as follows:

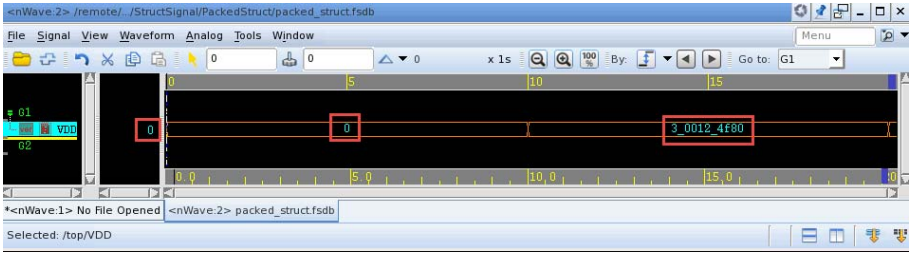


Figure: Packed Structure Signals in Bitwise Manner

The default value of this option is *off*.

Value Pane Page

Use this page to set the alignment and display preferences of values in the *nWave* value pane.

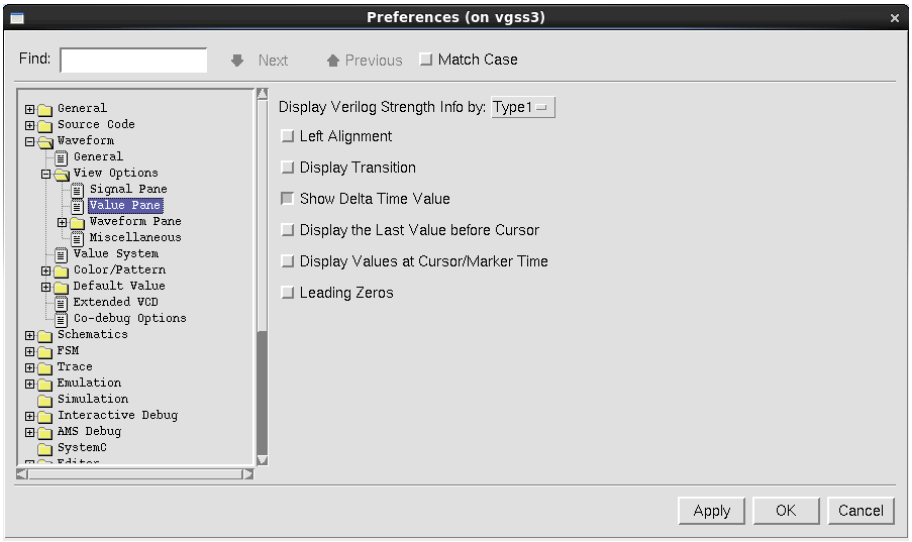


Figure: Value Pane Page

The following options are available:

- **Left Alignment:** When this option is turned *on*, the values are displayed on the left-hand side of the value pane. Otherwise, values are displayed on the right-hand side. The default value of this option is *off*.
- **Display Transition:** When this option is turned *on,* signal values with transition mode (1->0) is displayed in the value pane. The default value of this option is *on*.
- **Show Delta Time Value:** When this option is turned *on,* the delta time is displayed on the waveform toolbar. The default value of this option is *on*.
- **Display the Last Value before Cursor:** When this option is turned *on,* the signal values before the transition point of the current cursor are displayed. When this option is turned *off,* the signal values after the transition point of the current cursor are displayed. The default value of this option is *off*.
- **Display Values at Cursor/Marker Time:** By default, the signal value displayed in the value pane is the value at the cursor position. When this option is turned *on*, the value at the marker position is also shown next to (on the right of) the cursor value in the value pane. The default value of this option is *off*.
- **Leading Zeros:** When this option is turned *on*, leading zeros are displayed for signal values (including signals within transactions) in both value pane and waveform pane. The number of leading zeros depends on the format set

for the signal value. For changes to take immediate effect, select the **View -> Leading Zeros** command in the *nWave* window. The default is *off*.

- **Display Verilog Strength Info by:** Specify a strength display format for the Verilog value system by selecting either **Type1** (the original display rule) or **Type2** (the new display rule applied by most simulators). The default value of this option is **Type1**.

Waveform Pane Folder

This folder includes **General** and **Transaction/Message/Method** pages.

General Page

Use this page to set the display preferences of values in the waveform pane.

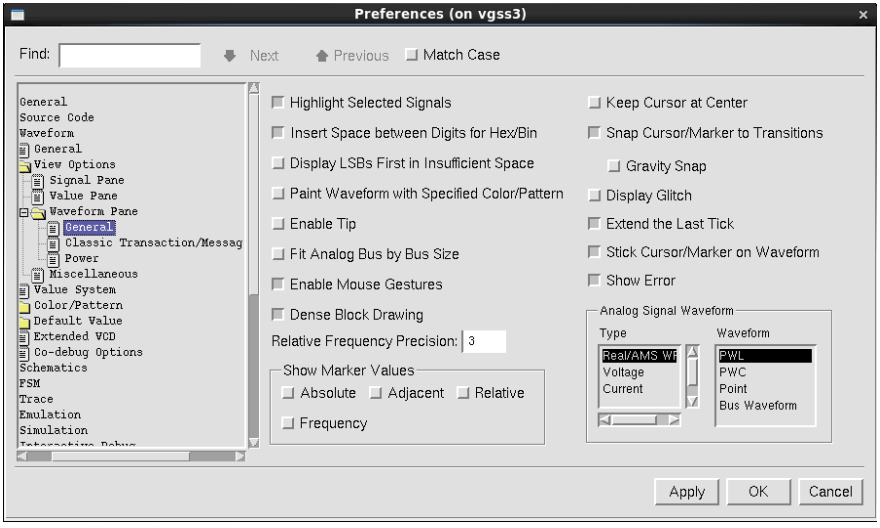


Figure: General Page

The following options are available:

- **Highlight Selected Signals:** When this option is turned *on*, selecting a signal in the signal pane highlights its waveform in the waveform pane. When this option is turned *off*, the waveform of a selected signal is not highlighted. The default value of this option is *on*.
- **Insert Space between Digits for Hex/Bin:** When this option is turned *on*, one underscore line is inserted between each group of 16 bits. The default value of this option is *on*.

- **Display LSBs First in Insufficient Space:** When this option is turned *on*, the LSBs are displayed first if the value box in a waveform pane does not have enough space to show the entire bus value. The default value of this option is *off*.
- **Paint Waveform with Specified Color/Pattern:** When this option is turned *on*, the setting from the **Waveform -> Color/Pattern** command is applied to the waveform pane. When this option is turned *off*, then the setting (**Waveform -> Color/Pattern**) is only applied to the signal and value panes. The default value of this option is *off*.
- **Enable Tip:** When this option is turned *on*, tips are displayed in the waveform pane. The default value of this option is *off*.
- **Fit Analog Bus by Bus Size:** When this option is turned *on*, analog buses are fit by bus size. The default value of this option is *off*.
- **Enable Mouse Gestures:** When this option is turned *on*, different mouse gestures can be used to specify the zoom ratio and direction for the waveform pane. The default value of this option is *on*.

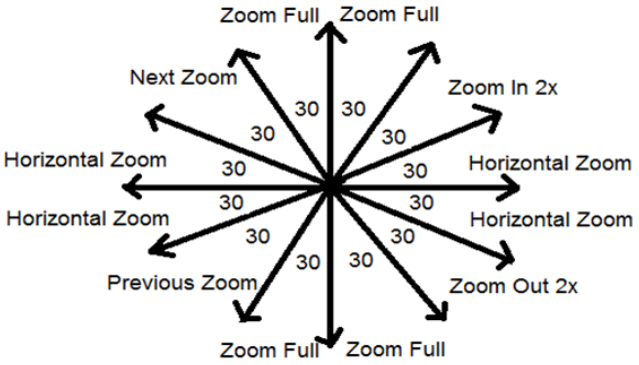


Figure: Mouse Gestures for Zoom Ratio and Direction

The following zoom ratios and directions can be specified with corresponding mouse gestures:

- **Zoom Full:** Click on the waveform pane and drag upwards or downwards.

NOTE: If an analog waveform in the waveform pane is clicked and dragged upwards or downwards, the mouse gesture runs vertical zoom rather than zoom full.

- **Zoom In 2x:** Click on the waveform pane and drag towards the up-right direction.

  - **Horizontal Zoom:** Click on the waveform pane and drag towards the left or right direction.
  - **Zoom Out 2x:** Click on the waveform pane and drag towards the down-right direction.
  - **Previous Zoom:** Click on the waveform pane and drag towards the down-left direction.
  - **Next Zoom:** Click on the waveform pane and drag towards the up-left direction.
- **Dense Block Drawing:** When this option is turned *on*, the dense block drawing signals for fast drawing of VHDL records are displayed in the waveform pane. When this option is turned *off*, displaying the dense block drawing signals for fast drawing of VHDL records is stopped in the waveform pane. The default value of this option is *on*.
- **Relative Frequency Precision 'n':** Specify the number of significant digits after the decimal that includes frequency between user markers and reference markers, and frequency between primary cursor and primary marker (displayed on the toolbar). The default value of this option is 3.

NOTE: If you uncheck the **Label Marker** option (invoked by the **Preference -> Waveform -> General** command), the marker information that is, name, time, adjacent, relative, and so on are not displayed on the Waveform.

- **Show Marker Values:** Select one of the following marker values to set the default value of the **Show Cursor/Marker Values**.

  Absolute, **Adjacent**, **Relative**, or **Frequency**.
- **Keep Cursor at Center:** When this option is turned *on*, if the cursor is not seen in the waveform pane, the cursor is made viewable in the center of the *nWave* window while it redraws the waveform (such as, searching signal values). The default value of this option is *off*.
- **Snap Cursor/Marker to Transitions:** When this option is turned *on* and the cursor or marker is placed in the waveform pane, the cursor or marker automatically snaps to the nearest value change position. Otherwise, the cursor or marker moves to the nearest value change on the signal selected in the waveform pane. This option only affects the default value of the **Waveform -> Snap Cursor/Marker to Transitions** command in the newly opened *nWave* window. The default value of this option is *on*.

NOTE: Clicking the **Zoom Scale Ruler** (at the top of the waveform pane) or **Full Scale Ruler** (at the bottom of the waveform pane) does not affect the snap option.

- **Gravity Snap:** When this option is turned *on*, and the **nWave -> Waveform -> Snap Cursor/Marker to Transitions** command is turned *on*, if the cursor/marker is *close enough* to the transition, snap to the transition occurs. If not, the cursor/marker is placed at the position where the mouse is pointing. When this option is turned *off,* and the **nWave -> Waveform -> Snap Cursor/Marker to Transitions** command is turned *on*, the cursor or marker *always* automatically snaps to the nearest value change position. The default is *off*.

- **Display Glitch:** This option shows glitches in the waveform pane if any exist in the FSDB file. Glitches are saved in the FSDB file by setting the `NOVAS_FSDB_ENV_MAX_GLITCH_NUM` environment variable to a value of 0 or greater than 1.

NOTE: If the value of `NOVAS_FSDB_ENV_MAX_GLITCH_NUM` is 0, all glitches are dumped.

When this option is turned *on*, and if one of the opened FSDB files has a glitch, then the **View -> Display Glitch** pull-down menu is enabled and turned *on*. Otherwise, the **View -> Display Glitch** menu option is disabled.

When this option is turned *off*, and if one of the opened FSDB files has a glitch, then the **View -> Display Glitch** pull-down menu is enabled. Otherwise, the **View -> Display Glitch** menu option is disabled.

- **Extend the Last Tick:** When this option is turned *on*, the last time tick in the waveform pane is displayed by extending 1 percent of the viewing range. The last value of a signal can be seen by zooming at the end. The default value of this option is *on*.
- **Stick Cursor/Marker on Waveform:** When this option is turned *on*, the cursor/marker can only be dragged in the marker area. When this option is *off*, the cursor/marker can be dragged in both the waveform area and the marker area. The default value of this option is *on*.
- **Show Error:** When this option is *on*, the show error indicator in *nWave* window is automatically shown in the Interactive Mode.

Classic Transaction/Message/Method Page

Use this page to set the display preferences of transactions or messages in the waveform pane.

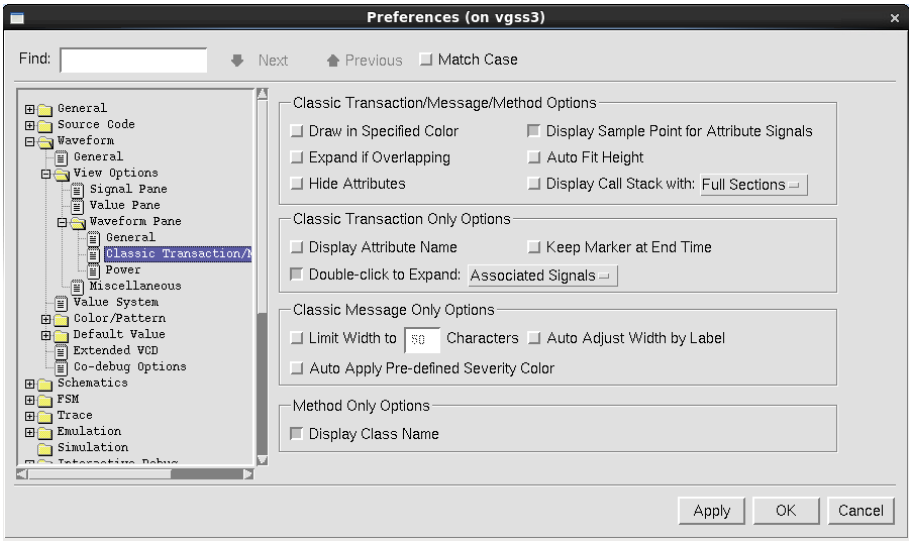


Figure: Classic Transaction/Message/Method Page

The following options are available in the **Classic Transaction/Message/Method Options** section:

- **Draw in Specified Color:** When transactions or messages in the FSDB file contain a color attribute and this option is turned *on*, transaction or message label in *nWave* displays the color that was defined in the attribute. When this option is turned *off*, the default color scheme is used even if a color attribute is defined for transactions or messages. The default value of this option is *off*.
- **Expand if Overlapping:** When this option is turned *on*, the overlapped transaction or message streams in the waveform pane are expanded to multiple layers. When this option is turned *off*, the expanded transactions or messages in the waveform pane are shrunk to one layer. The default value of this option is *off*.
- **Hide Attributes:** When this option is turned *on*, the attributes of selected transaction or message streams are not displayed in the waveform pane. The default value of this option is *on*.
- **Display Sample Point for Attribute Signals:** When this option is turned *on*, the sample points (green triangles) for expanded attributes of transaction

or message streams are displayed in the waveform pane. The default value of this option is *on*.

- **Auto Fit Height:** When this option is turned *on*, the transaction height is automatically adjusted to fit the number of attributes. When this option is turned *off*, the transaction height fits four attributes. The default value of this option is *off*.
- **Display Call Stack with:** When this option is turned *on*, full or last selected call stacks are displayed in the waveform pane. The default value of this option is *off*.

The following options are available in the **Classic Transaction Only Options** section:

- **Display Attribute Name of Transaction:** When this option is turned *on*, the transaction attribute name is displayed in the waveform pane. The default value of this option is *off*.
- **Keep Marker at End Time of Transaction:** When this option is turned *on*, the cursor is automatically set to the start time of the selected transaction and the marker is set to the end time of the transaction. The default value of this option is *off*.
- **Double-click to Expand:** When this option is turned *on*, select the type of signals to expand on double-click. The options are **Associated Signals** and **Attribute Signals**. The default value of this option is *on* and **Associated Signals**.

The following options are available in the **Classic Message Only Options** section:

- **Limit Width to Characters:** When this option is turned *on*, the width of messages shown in the waveform pane is limited to the specified number of characters. The default value of this option is *off* and *50*.
- **Auto Apply Predefined Severity Color:** When this option is turned *on*, the predefined severity color is applied to (1) the newly added message streams and (2) the message streams that have not been applied with the user-defined color. The default value of this option is *on*.
- **Auto Adjust Width by Label:** When this option is turned *on*, the message width is automatically adjusted by the label. The default value of this option is *off*.

The following option is available in the **Method Only Options** section:

- **Display Class Name:** When this option is turned *on*, the class name is displayed for methods. The default value of this option is *on*.

Power Page

Use this page for power mask settings in the waveform pane.

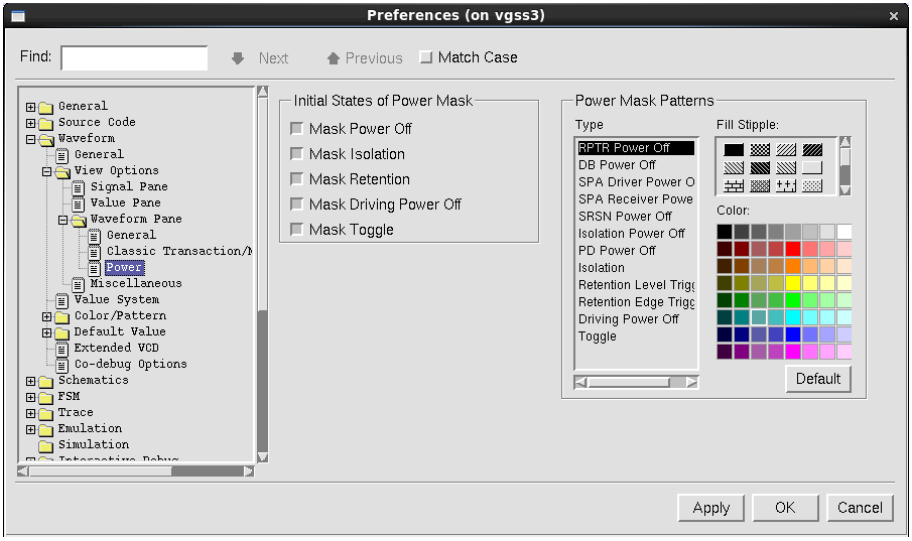


Figure: Power Page

The following options are available in the **Initial States of Power Mask** section:

- **Mask Power Off:** When this option is turned *on*, *nWave* masks the liberty-affected, set_port_attribute (SPA), set_related_supply_net (SRSN), or power-off range for HDL signals. The default value of this option is *off*. Refer to the **Mask Power Off** command for details.
- **Mask Isolation:** When this option is turned *on*, *nWave* masks the isolation range for HDL signals according to the Isolation condition applied. The default value of this option is *off*. Refer to the **Mask Isolation** command for details.
- **Mask Retention:** When this option is turned *on*, *nWave* masks the retention range for HDL signals according to the Retention condition applied. The default value of this option is *off*. Refer to the **Mask Retention** command for details.
- **Mask Driving Power Off:** When this option is turned *on*, *nWave* masks the driving power-off ranges. Driving power-off ranges are areas where the driver signal(s) of the selected signal are liberty-affected, set_port_attribute (SPA), set_related_supply_net (SRSN), or Power Off. The default value of this option is *off*. Refer to the **Mask Driving Power Off** command for details.

- **Mask Toggle:** When this option is turned *on*, *nWave* masks the range for HDL signals where the scope is already in a power-off corruption state and the input signals have value changes. The default value of this option is *off*. Refer to the **Mask Toggle** command for details.

The pattern for the power mask types can be changed in the **Power Mask Patterns** section. After a power mask type is selected in the **Type** field, select the desired fill style and color for this type to change its pattern. To reset the selected types to the default settings, click the **Default** button. The power mask type includes **DB Power Off**, **SPA Driver Power Off**, **SPA Receiver Power Off**, **SRSN Power Off**, **Isolation Power Off**, **Power Domain Power Off**, **Isolation**, **Retention Level Trigger**, **Retention Edge Trigger**, **Driving Power Off**, and **Toggle**.

NOTE: The icons of the power masks types on the *Legend* form in the **nWave** tab are not changed even if the pattern for the power mask types are changed in the *Preferences* form.

Miscellaneous Page

Use this page to specify several miscellaneous layout options for *nWave*.

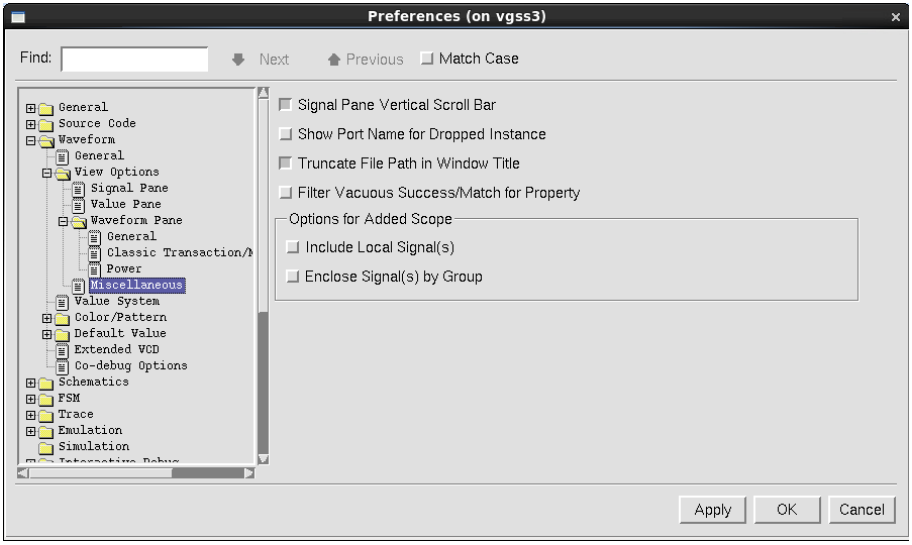


Figure: Miscellaneous Page

The following options are available:

- **Signal Pane Vertical Scroll Bar:** When this option is turned *on*, a vertical scroll bar is shown along the signal pane. When the mouse button is pressed and held to scroll up or down on the vertical scroll bar in the signal pane, the corresponding items are not shown in the waveform pane until the mouse button is released and the scrolling action is stopped. The default value of this option is *on*.
- **Show Port Name for Dropped Instance:** When this option is turned *on*, all instance port names for instances dragged from *nTrace*, *nSchema*, or *Temporal Flow View* or all port names for instance ports dragged from *nSchema* or *Temporal Flow View* and dropped in *nWave* are displayed. When this option is turned *off*, *nWave* displays net name(s) for dropped instance(s) or instance port(s) instead. The default value of this option is *off*.
- **Truncate File Path in Window Title:** When this option is turned *on*, all waveform window titles are truncated in the middle. When it is *off*, all waveform window titles are truncated at the end. The default value of this option is *on*.

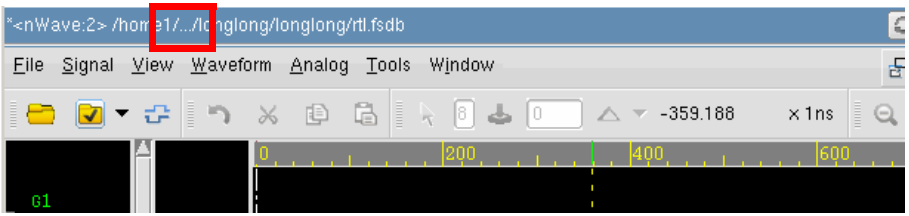


Figure: Truncate File Path in Window Title - On

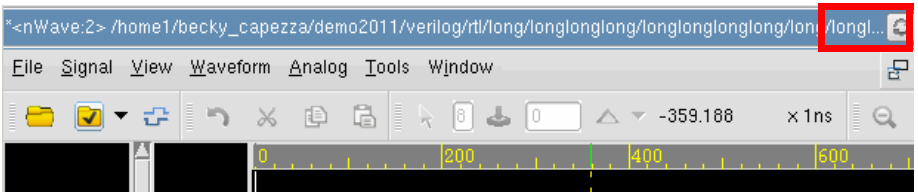


Figure: Truncate File Path in Window Title - Off

- **Filter Vacuous Success/Match for Property:** When this option is turned *on*, the threads with the value string "vacuous-success" or "vacuous-match" of the property signals are not displayed on the waveform window. The default value of this option is *off*.

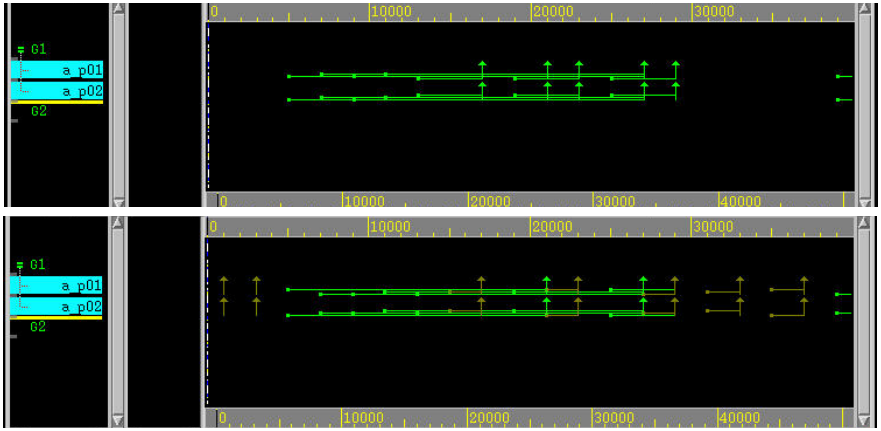


Figure: Filter Vacuous Success/Match for Property Option On and Off

- **Options for Added Scope:** This section specifies the following options for the dropped scope:
  - **Include Local Signal(s):** When this option is turned *on*, local signals that are under the selected scope by one level are added to *nWave* after an instance is dragged from the hierarchy tree and dropped to *nWave*. When this option is turned *off*, only the input and the output port are added to *nWave*. The default value of this option is *off*.

NOTE: If the option is *off*, and a scope which does not have in/output port is dragged and added to *nWave*, a warning message appears which mentions to turn *on* the option and try again.

  - **Enclosed Signal(s) by Group:** When this option is turned *on*, all signals under the selected scope are added to a new sub-group after an instance is dragged from the hierarchy tree and dropped to *nWave*. The sub-group is created with the name of the dropped instance by *nWave*. The default value of this option is *off*.

Value System Page

Use this page to specify the pattern for signals displayed in the *nWave* window according to **Data Type** and **Logic State**.

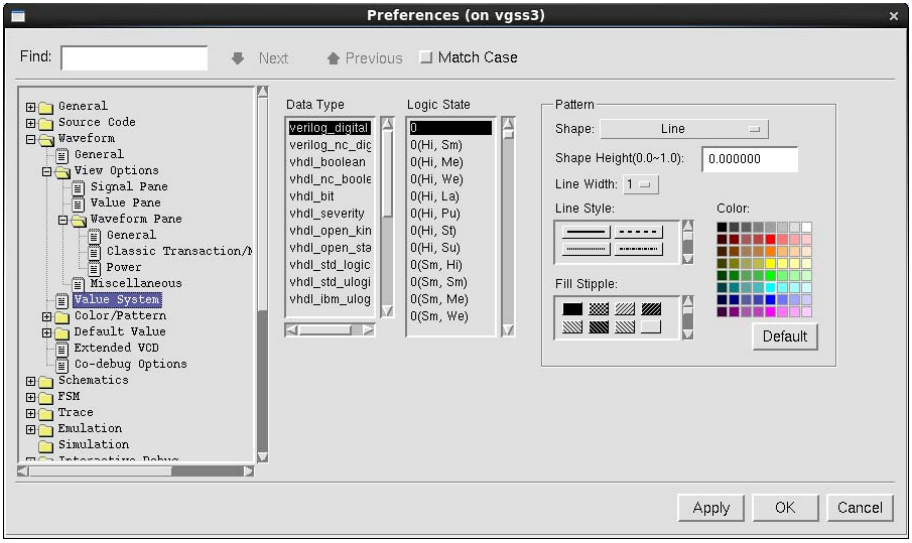


Figure: Value System Page

In the **Pattern** section, the pattern **Shape** from **Line**, **Rectangle** (with outline), or **Rectangle** (without outline) can be changed, as well as **Shape Height** (0.0~1.0) and **Line Width** (from 1 to 5).

For **Line Style** and **Fill Stipple**, 10 and 23 styles are available respectively.

Click the **Default** button to reset these options back to their original default settings.

Color/Pattern Folder

The **Color/Pattern** folder includes the **Color** and **Pattern** pages.

Color Page

Use this page to specify the color of attributes in the *nWave* window.

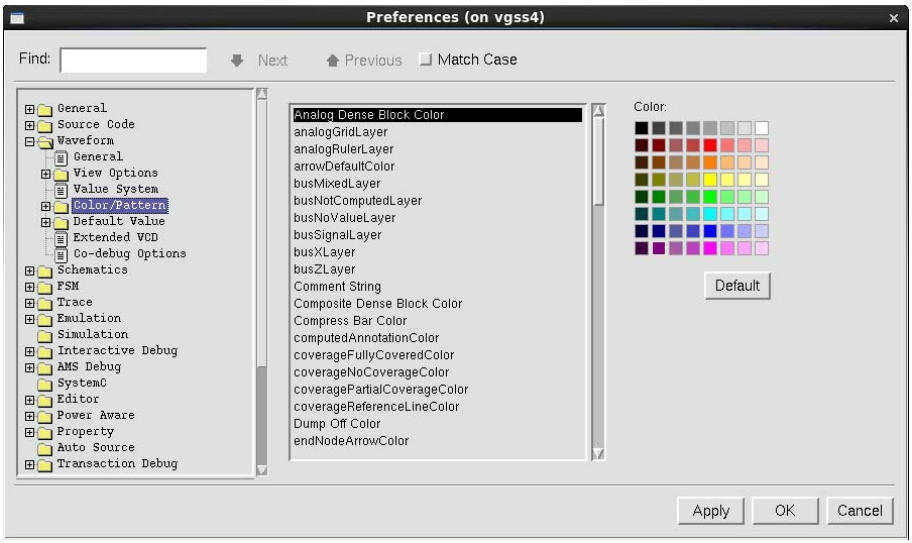


Figure: Color Page

The colors of the following attributes can be changed:

- **Analog Dense Block Color:** The color of analog dense block (scalar, vector, and composite) signals in *nWave*. The default color is ID_PURPLE2.
- **analogGridLayer:** The foreground color of the grid for the analog signal. The default color is ID_GRAY6.
- **analogRulerLayer:** The foreground color of the view ruler for the analog signal. The default color is ID_GRAY6.
- **arrowDefaultColor:** The color of the arrow with both a start node and an end node in a waveform viewable range. The default color is ID_ORANGE6.
- **busMixedLayer:** The waveform color of a bus signal whose partial members have the X value. The default color is ID_GRAY5.
- **busNotComputedLayer:** The color of an incomputable range for a bus signal in a waveform. For Data Expansion in Siloti only. The default color is ID_GRAY1.

- **busNoValueLayer:** The color of the not-yet-computed range for a bus signal in a waveform. For Data Expansion in Siloti only. The default color is ID_BLUE2.
- **busSignalLayer:** The line color of bus signals. The default color is ID_ORANGE8.
- **busXLayer:** The line color of the X value. The default color is ID_RED5.
- **busZLayer:** The line color of the Z value. The default color is ID_ORANGE6.
- **Comment String:** The color of the comment.
- **Composite Dense Block Color:** The color of drawn dense composite (record, struct, and MDA) signals in *nWave*. The default color is ID_RED5.
- **Compress Bar Color:** The color of the dense bar and the label for the compress time range. The default color is ID_YELLOW4.
- **computedAnnotationColor:** The color of computed signals. The default color is ID_ORANGE5.
- **coverageFullyCoveredColor:** The color of the full coverage. The default color is ID_GREEN5.
- **coverageNoCoverageColor:** The color of the coverage when no coverage exists. The default color is ID_RED5.
- **coveragePartialCoverageColor:** The color of the coverage being partially covered. The default color is ID_YELLOW5.
- **coverageReferenceLineColor:** The color of the coverage reference line. The default color is ID_GRAY4.
- **Dump Off Color:** The color of the dump-off section. The default color is ID_BLUE2.
- **endNodeArrowColor:** The color of an arrow with only an end node in the waveform viewable range. The default color is ID_YELLOW5.
- **GetSignal SignalList Background:** The background color of the signal list in the *Get Signals* form. The default color is ID_GRAY5.
- **GetSignal SignalList Default Signal Color:** The color of the unselected signal in the *Get Signals* form. The default color is ID_BLACK.
- **GetSignal SignalList Loaded-Signal Color:** The color of the loaded signal in the *Get Signals* form. The default color is ID_BLUE5.
- **Glitch Symbol:** The symbol color of the glitch indicator. The default color is ID_RED5.
- **Group Signal:** The color of the group signal.

- **High Light:** The color of the highlighted signal in the waveform pane. The default color is ID_GRAY2.
- **InOut Signal:** The color of inout (bidirectional) port type signals for color annotation. The default color is ID_BLUE5.
- **Input Signal:** The color of input port type signals for color annotation. The default color is ID_RED5.
- **keywordLayer:** The color of the *Get Signals* form *LOGIC_HIGH*, *LOGIC_LOW*, and *BLANK* generic signals.
- **messageBackgroundColor:** The color of the message background. The default color is ID_PURPLE1.
- **messageForegroundColor:** The color of the message foreground. The default color is ID_YELLOW4.
- **messageHighLightColor:** The color of the message when it is selected. The default color is ID_CYAN6.
- **messageInformationColor:** The color of the message information. The default color is ID_RED5.
- **Net Signal:** The color of net data type signals for color annotation. The default color is ID_YELLOW5.
- **Output Signal:** The color of output port type signals for color annotation. The default color is ID_GREEN5.
- **propertyAssertFailureColor:** The color of the assert property when the result is "failure". The default color is ID_RED5.
- **propertyAssertSuccessColor:** The color of the assert property when the result is "success". The default color is ID_GREEN5.
- **propertyBooleanFailureColor:** The color of the boolean property when the result is "failure". The default color is ID_RED5.
- **propertyBooleanSuccessColor:** The color of the boolean property when the result is "success". The default color is ID_CYAN5.
- **propertyEventMatchColor:** The color of the event property when the result is "success". The default color is ID_CYAN5.
- **propertyEventNoMatchColor:** The color of the event property when the result is "failure". The default color is ID_RED5.
- **propertyForbidFailureColor:** The color of the forbid property when the result is "failure". The default color is ID_RED5.
- **propertyForbidSuccessColor:** The color of the forbid property when the result is "success". The default color is ID_GREEN5.

- **propertyVacuousSuccessMatchColor:** The color of the vacuous success or match when the result is “success”. The default color is ID_YELLOW2.
- **qdsBrkMarkerLayer:** The color of the break point marker for the Quickturn waveform. The default color is ID_GREEN5.
- **qdsCurMarkerLayer:** The color of the current cycle marker for the Quickturn waveform. The default color is ID_BLUE5.
- **qdsTrgMarkerLayer:** The color of the trigger position marker for Quickturn waveform. The default color is ID_RED5.
- **Region (Active) Background:** The color of region (active) background. The default color is ID_YELLOW1.
- **Region (Dump-Off) Background:** The color of region (dump-off) background.
- **Region (NBA) Background:** The color of region (NBA) background. The default color is ID_RED1.
- **Region (Re-active) Background:** The color of region (re-active) background. The default color is ID_YELLOW3.
- **Region (Re-NBA) Background:** The color of region (re-NBA) background. The default color is ID_RED3.
- **Region (VHDL-Delta) Background:** The color of the VHDL-Delta background.
- **Register Signal:** The color of register data type signals for color annotation. The default color is ID_PURPLE5.
- **Ruler Background:** The background color of the view ruler (upper one) and full ruler. The default color is ID_GRAY3.
- **Ruler Foreground:** The foreground color of the view ruler (upper one) and full ruler. The default color is ID_YELLOW5.
- **Scalar Dense Block Color:** The color of drawn dense scalar signals in *nWave*. The default color is ID_GREEN6.
- **Signal Pane:** The color of the signal pane.
- **Signal Pane Background:** The background color of the signal pane. The default color is ID_BLACK.
- **startNodeArrowColor:** The color of the arrow with only start node in waveform viewable range. The default color is ID_WHITE.
- **SystemC Signal:** The color of the SystemC signal.
- **transactionBackgroundColor:** The color of the transaction background. The default color is ID_BLACK.

- **transactionErrorTypeColor:** The color of the transaction error type. The default color is ID_RED5.
- **transactionForegroundColor:** The color of the transaction foreground. The default color is ID_YELLOW8.
- **transactionHighLightColor:** The color of the selected transaction. The default color is ID_CYAN6.
- **transactionRelationshipColor:** The color of transactions related to the selected transaction. The default color is ID_PURPLE6.
- **Value Pane:** The color of the value pane. The default color is .
- **Value Pane Background:** The background color of the value pane. The default color is ID_BLACK.
- **Vector Dense Block Color:** The color of drawn dense vector signals in nWave. The default color is ID_ORANGE8.
- **Verilog Signal:** The color of Verilog file type signals for color annotation. The default color is ID_CYAN5.
- **VHDL Signal:** The color of VHDL file type signals for color annotation. The default color is ID_ORANGE5.
- **Waveform Pane:** The color of the waveform pane. The default color is ID_BLACK.
- **Waveform Pane Background:** The background color of the waveform pane. The default color is ID_BLACK.

Click the **Default** button to reset these options back to their original default settings.

Pattern Page

Use this page to specify width, style, and color for different line types in the *nWave* window.

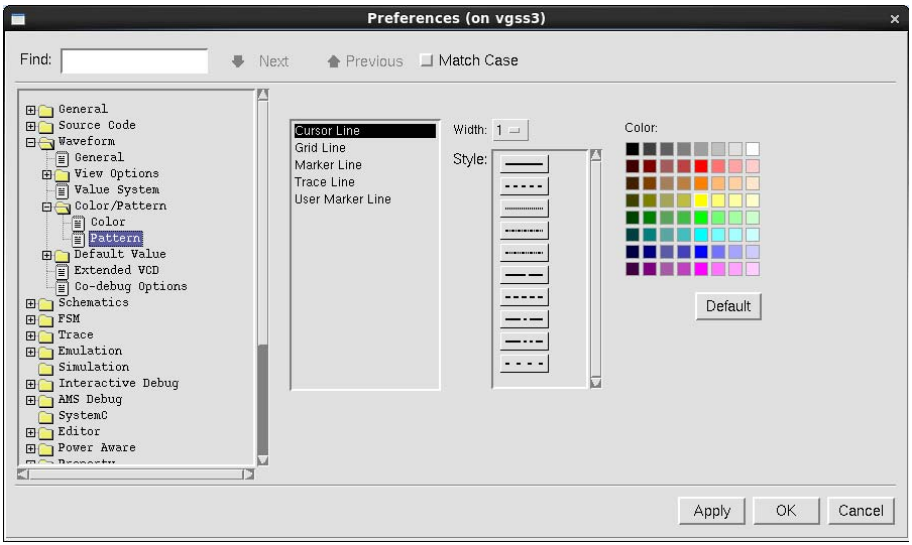


Figure: Pattern Page

Select either **Cursor Line, Grid Line, Marker Line, Trace Line,** or **User Marker Line** and specify the **Width** (from 1 to 5), **Style**, and color.

Click the **Default** button to reset these options back to their original default settings.

Default Value Folder

The **Default Value** folder includes **Displayed Signal**, **Conversion Attributes**, and **Miscellaneous** pages.

Display Signal Page

Use this page to specify various signal display options.

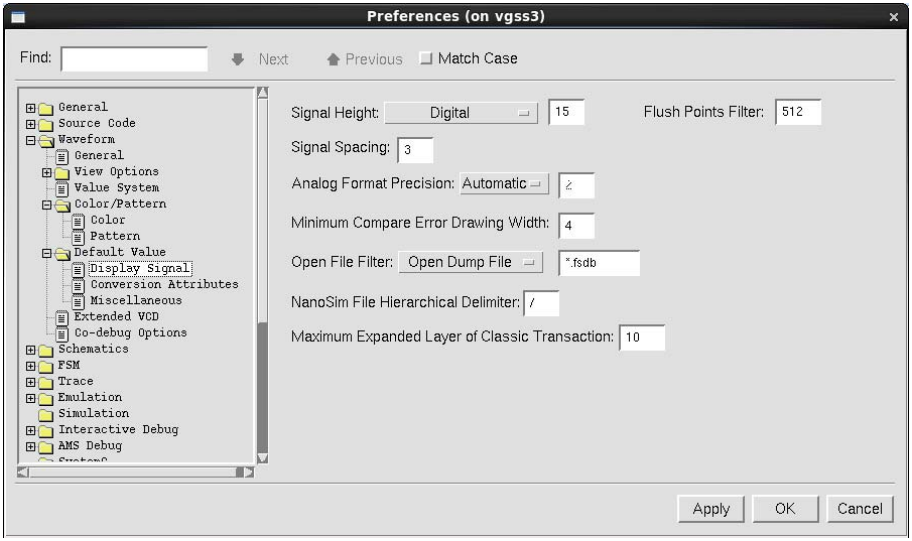


Figure: Display Signal Page

The following options are available:

- **Signal Height:** Select the displayed signal type as **Digital**, **Analog**, **Comment**, **Transaction**, or **Message** and set the height of the displayed signal in pixels in the waveform pane.
- **Signal Spacing:** Specify the spacing between signals with the height measured in pixels. The default value of this optionis *5*.
- **Analog Format Precision:** Define the value format displayed in *nWave* either as **Scientific**, **Engineering**, or **Automatic** (automatically decides which format is best suited for the signal).
- **Minimum Compare Error Drawing Width:** Configure the minimum width of compare error drawing. The default value of this option is *4*.
- **Open File Filter:** Specify the default file format filter for the **File** list by selecting one of the following options: **Open Dump File** (*.*fsdb*;*.*vf*), **Dump File** (*.*vcd*), **Restore Signal File** (*.*rc*), **Save Signal File** (*.*rc*),

Alias File (**.alias*), **Compare File** (**.err*), and **Convert File** (**.vcd;*.out;*.tr0;*.xp;*.raw;*.wfm*). If more options are added to the default filter, they must be separated by a semi colon (such as, **.vf;*.fsdb;*.vcd)*.

- **Maximum Expanded Layer of Transaction:** Set the maximum value for expanded layers of overlapped transactions. The default value of this option is *10*.
- **Flush Points Filter:** *nWave* buffers any geometric drawing points that must be updated. When the buffer is full, *nWave* flushes the buffer contents and draws the waveform in the *nWave* window. The larger is the buffer size the faster is the drawing process. The maximum buffer size is 2048. A VCD file is automatically converted to an FSDB file if the simulation results are in VCD format.

Conversion Attributes Page

Use this page to specify the attributes for VCD to FSDB conversion.

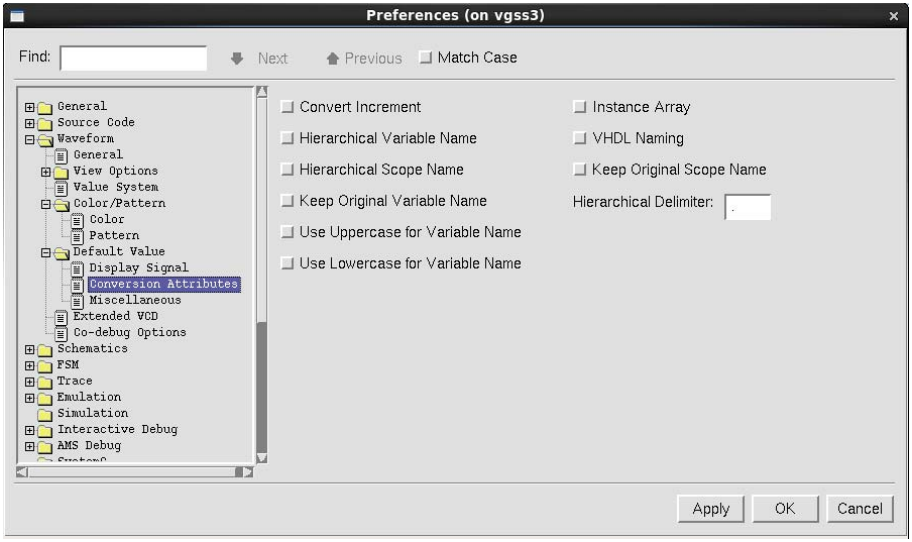


Figure: Conversion Attributes Page

The following options are available:

- **Convert Increment:** When this option is turned *on*, the file is incrementally parsed while it is still growing. The default value of this option is *off*.

- **Hierarchical Variable Name:** When this option is turned *on*, hierarchical variable names are parsed and the corresponding scope(s) created. The default value of this option is *off*.
- **Hierarchical Scope Name:** When this option is turned *on*, hierarchical scope names are parsed and the corresponding scope(s) created. The default value of this option is *off*.
- **Keep Original Variable Name:** When this option is turned *on*, the original variable name is kept and a conversion is not performed on the variable name. The default value of this option is *off*.
- **Use Uppercase for Variable Name:** When this option is turned *on*, all signal definitions are converted to uppercase. The default value of this option is *off*.
- **Use Lowercase for Variable Name:** When this option is turned *on*, all signal definitions are converted to lowercase. The default value of this option is *off*.
- **Instance Array:** When this option is turned *on*, an instance array is created when converting a VCD file. The default value of this option is *off*.
- **VHDL Naming:** When this option is turned *on*, VHDL naming rules are used. The default value of this option is *off*.
- **Keep Original Scope Name:** When this option is turned *on*, the original scope name is kept and a conversion is not performed on the scope name. The default value of this option is *off*.
- **Hierarchical Delimiter:** Specify the hierarchical delimiter for parsing and separating the full path signal names of analog type dump files. The forward slash (/) character is the default hierarchical delimiter in *nWave*. This option is only valid for dump files of the following types: *spice*, *xp*, *rawfile*, or *wfm*.

Miscellaneous Page

Use this page to specify several miscellaneous waveform format options.

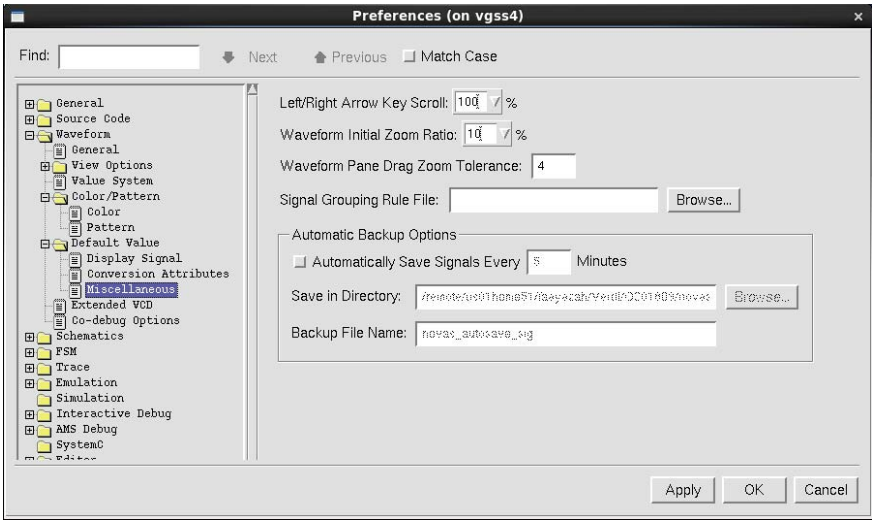


Figure: Miscellaneous Page

The following options and fields are available:

- **Left/Right Arrow Key Scroll:** Set the percentage for scrolling a waveform with the left or right arrow keys by **100%**, **75%**, **50%**, or **25%**.
- **Waveform Initial Zoom Ratio:** Change the initial zoom ratio of the waveform pane by setting the percentage as **100%**, **50%**, or **10%** respectively. The next time an FSDB file is opened after executing **File -> Close -> All** or if a new waveform pane is opened, the zoom ratio of the waveform pane is changed.
- **Waveform Pane Drag Zoom Tolerance:** Specify the drag zoom tolerance value in the waveform pane.
- **Signal Grouping Rule File:** Specify a grouping rule file by either inputting the file name or clicking the **Browse** button to view the directory structure and locate the grouping rule file. Automatic bus grouping is performed on future sessions.
- **Default Radix:** When this option is turned *on*, select the radix for bus signals as **Binary, Octal, Hexadecimal, Decimal, ASCII,** or **Floating Point**.
- **Automatic Backup Options:**

- **Automatically Save Signals Every 'n' Minutes:** When this option is turned *on*, all automatic backup options and fields are enabled and signals can be saved to a specified *rc* signal file and location at the specified time interval (only integer values are allowed). The saved files can be restored through the **File -> Restore Signal** command in *nWave*. The default value of this option is *off* and *5*.
- **Save in Directory:** Specify the file path by clicking the **Browse** button to open the *Select Directory* form where the directory structure can be viewed and a scope can be selected. The text field cannot be specified manually.
- **Backup File Name:** Specify the backup file name. The default backup file name is *novas_autosave_sig*. The naming rule for the automatically saved file is *BackupFileName_<nWave_winID>.rc* (such as, *novas_autosave_sig_nWave1.rc, novas_autosave_sig_nWave2.rc*, and so on).

Extended VCD Page

Use this page to specify the default settings for an extended VCD file.

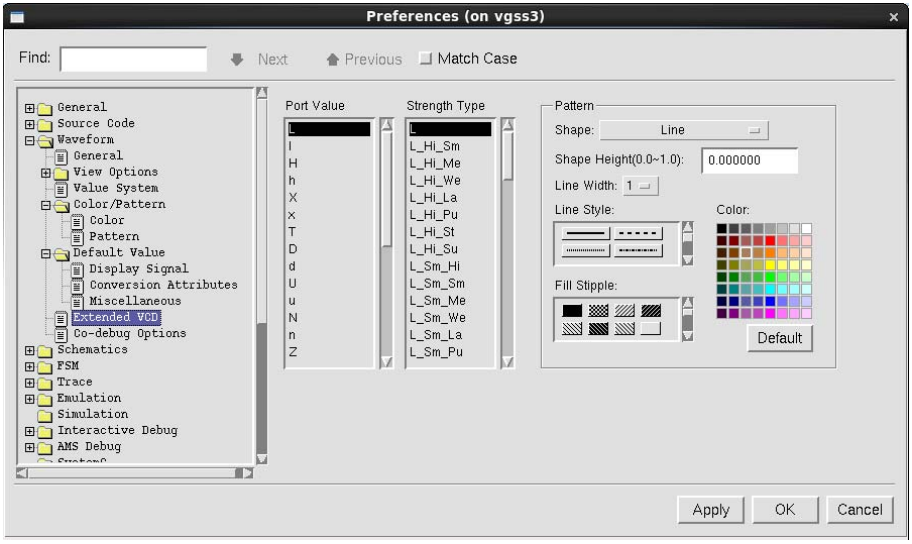


Figure: Extended VCD Page

Select **Port Value** (25 options) and **Strength Type** (64 options) and then in the **Pattern** section, the pattern **Shape** from **Line**, **Rectangle (with outline)**, or

Rectangle (without outline) can be changed, as well as **Shape Height** (0.0~1.0), **Line Width** (from 1 to 5), **Line Style** (10 options), and **Fill Stipple** (23 options).

Click the **Default** button to reset these options back to their default settings.

Schematics Folder

The **Schematics** folder includes the **Color/Font** page, the **RTL** page, the **Symbol Map** page, the **Symbol Library** page, the **Miscellaneous** page, the **Display Options** folder, the **Select** page, the **SDF** page, the **Trace** page, the **PowerMap** folder, the **Clock** folder, the **nECO** folder, and the **EditSchematic** folder.

Color/Font Page

Use this page to associate different colors and fonts with objects in the main display area for all schematic windows.

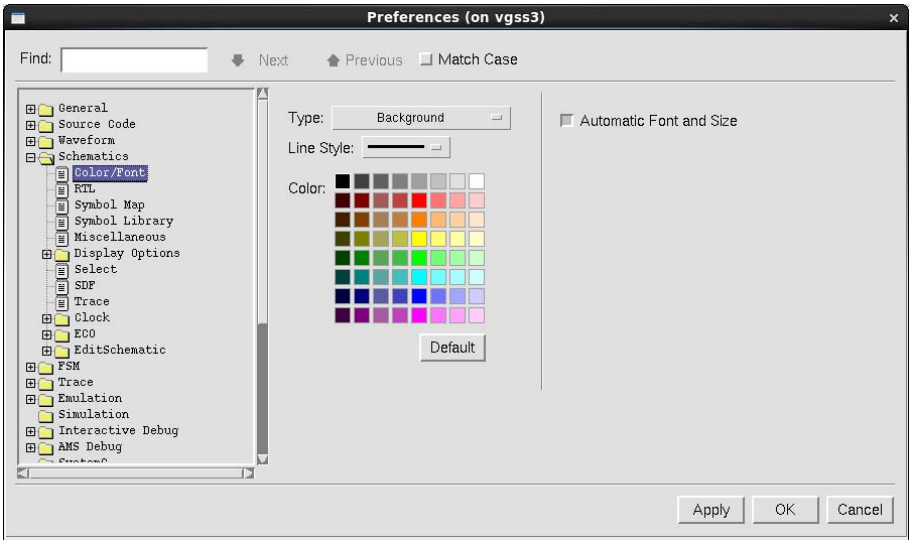


Figure: Color/Font Page

- **Type:** Select the type of object for which the color attributes need to be changed. Options are as follows (only one of these options can be selected at a time):
  - **Auto Merge Group Net:** The color for the net of merged cells.
  - **Back Annotation:** The color for annotation of simulation values.
  - **Background:** The background color in the *nSchema* window.
  - **Block Symbol:** The color for block symbols.
  - **Clock Signal:** The color for clock signals.
  - **Connected Pin:** The color for the connected pin.

- **Constant Net 0:** The color for the constant net with value set as 0.
- **Constant Net 1:** The color for the constant net with value set as 1.
- **Dim Color:** The line color when the **View-> High Contrast** command is enabled in the *nSchema* window.
- **FSM Block:** The color for FSM blocks.
- **Flip-flop Black Box:** The color for flip-flop compressed register views (black boxes).
- **First Selected:** The color for the first selected object.
- **Global Signal:** The color for global signals.
- **Latch Black Box:** The color for the latch compressed register views (black boxes).
- **Lvs Inactive Net:** The color of the net that connects the level shifter and to domain when the level shifter is inactive.
- **Macro Black Box:** The color for the macro compressed register views (black boxes).
- **Normal Signal:** The color for normal signals.
- **Preselect:** The color for preselect objects.
- **PureView Memory:** The color for PureView Memories.
- **RTL Symbol:** The color for RTL symbols.
- **Rubber Banding:** The color for rubber banding.
- **Selected Set:** The highlight color for selected objects.
- **Symbol Name:** The color for the symbol name.
- **Symbol Port:** The color for symbol ports.
- **Tip Background:** The background color for tips.
- **Traced Set:** The color for traced sets.
- **Value 0:** The annotation color for signals with value '0' when annotate in color is enabled.
- **Value 1:** The annotation color for signals with value '1' when annotate in color is enabled.
- **Value x:** The annotation color for signals with value 'x' when annotate in color is enabled.
- **Value z:** The annotation color for signals with value 'z' when annotate in color is enabled.
- **VDD Signal:** The color for the VDD signal.
- **VSS Signal:** The color for the VSS signal.

- **Line Style:** Select the line style of the object selected in the **Type** selection field.
- **Color:** Select (left-click) the target color in the color matrix.
- **Default:** Return the selected **Type** to its default color.
- **Automatic Font and Size:** When this option is turned *off*, the font and size can be manually specified. The available combinations are: **Helvetica 8, Helvetica 10, Helvetica 12, Helvetica 14, Courier 18,** and **Courier 24**.

  Enable **Automatic Font and Size** and the font of the viewing region is adjusted automatically in the *nSchema* window. Enable this option for the best visual effect without zooming in or zooming out.

RTL Page

Use this page to set the default value for the module or the entity to control the RTL schematic display.

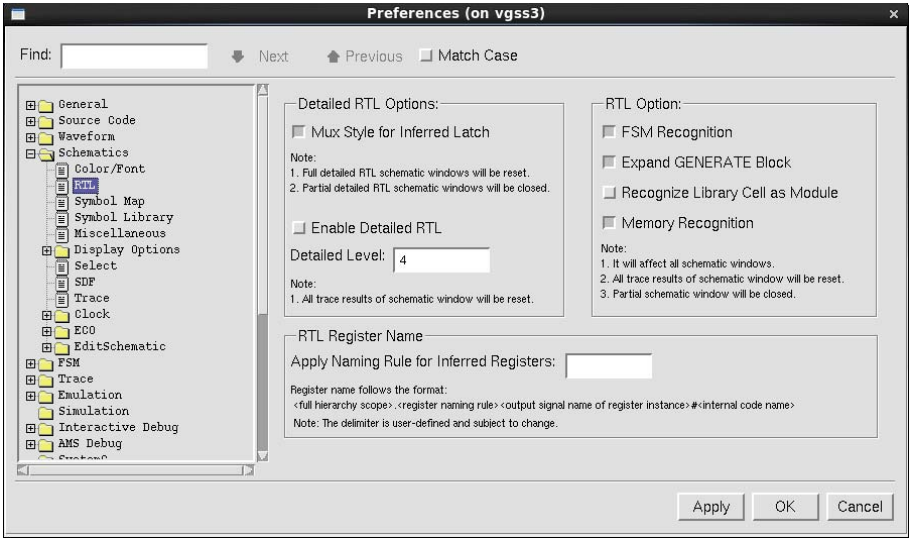


Figure: RTL Page

The following options are available in the **Detailed RTL Options** section:

- **Mux Style for Inferred Latch:** When this option is turned *on*, inferred latches are extracted as Mux style in the detailed RTL schematics. The default value of this option is *on*.
- **Enable Detailed RTL:** When this option is turned *on*, module or entity is displayed in detailed view (boolean equivalent logic). When this option is

turned *off*, module or entity is displayed as an abstract view (function symbol). Changes to this setting is reflected in the *nSchema* real-time. The default value of this option is *off*.

Detailed Level: After the **Enable Detailed RTL** option is turned *on*, a positive integer can be entered in this field and the specified integer shows that level of modules or entities in the detailed view.

The following options are available in the **RTL Option** section:

- **FSM Recognition:** When this option is turned *on*, *nSchema* recognizes an FSM for the selected module or entity. When this option is turned *off*, *nSchema* does not recognize an FSM for the selected module or entity. The default value of this option is *on*.
- **Expand GENERATE Block:** When this option is turned *on* and the GENERATE block is at the architecture level, *nSchema* expands the block and displays all individual instances in it. If the block is at the RTL level, this option has no effect. The default value of this option is *on*.
- **Recognize Library Cell as Module:** When this option is turned *on* and symbol libraries do not exist, then cells defined with `'celldefine` or `'endcelldefine`, or cells imported with the `-v` or `-y` options are recognized as modules. When this option is turned *on* and symbol libraries exist, cells are defined in symbol libraries, behaving the same as when this option is turned *off*. The default value of this option is *off*.

  When this option is switched from *on* to *off* or *off* to *on*, former netlist data is cleared.
- **Memory Recognition:** When this option is turned *on* then the cells that are recognized as memory type are displayed as a rectangle with a text MEMORY inside it. When this option is turned off, then the cells that may be of memory type are also displayed as general cells.

The following option is available in the **RTL Register Name** section:

- **Apply Naming Rule for Inferred Registers:** Enter a string in this text field to set the naming rule for RTL registers and then click **Apply** or **OK** buttons to apply the naming rule. In the non-detailed RTL view, the applied naming rule for an RTL register instance is only shown in the message bar of the schematic view. In the detailed RTL view, in addition to the message bar of the schematic view, the applied naming rule for an RTL register instance is also shown in the *Find* form opened by the **Schematic -> Find in Current Scope** command in *nSchema*, in the flattened schematic window, and *nSchema* frames opened by the **Tools -> New Schematic** commands in the *Clock Domains Window*, *Clock Tree Browser Window*, and *Crossing Paths*

Window. By default, the naming rule field is empty and RTL register instances display original names.

The naming rule for register follows the format of *RTL Register Name Option + Output signal name + #(internal code name)*. For example, if the current scope is `system.i_cpu`, the register output signal is `C0`, the RTL naming rule is `Reg_`, and the internal code name is `Always:3:0`, then the resulting name is `system.i_cpu.Reg_C0#Always:3:0`.

Symbol Map Page

Use this page to specify the settings for symbol-mapped files. These take effect when the design is reloaded or when the Verdi platform is invoked again.

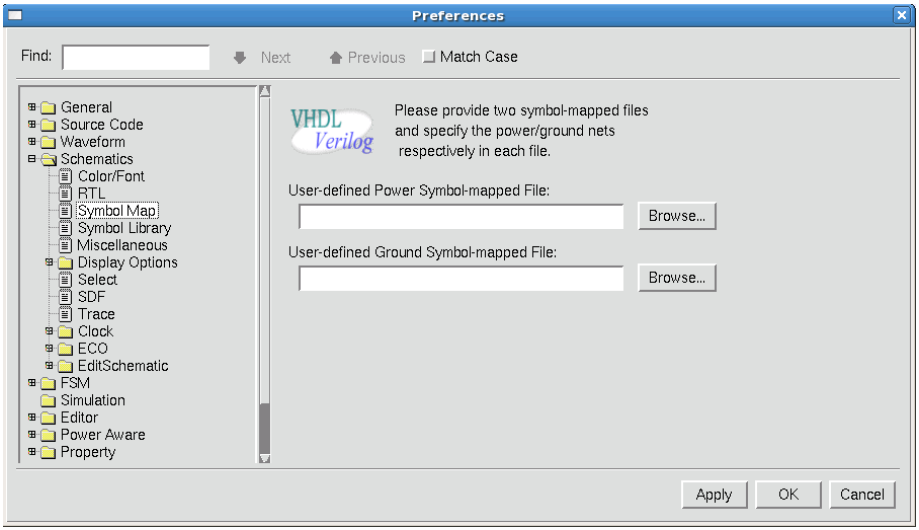


Figure: Symbol Map Page

The following fields are available:

- **User-defined Power Symbol-mapped File:** Specify the mapping file name in this text field or click the **Browse** button to open the *Select User-defined Symbol-mapped File* form to view the directory structure and specify a symbol-mapped file. In the mapping file, signal names that are recognized as power nets can be specified. The strings “VDD”, “VCC”, “VEE”, “POWER”, or “PWR” are regarded as power nets by default.
- **User-defined Ground Symbol-mapped File:** Specify the mapping file name in this text field or click the **Browse** button to open the *Select User-defined Symbol-mapped File* form to view the directory structure and specify a symbol-mapped file. In the mapping file, signal names that are

recognized as ground nets can be specified. The strings “VSS”, “GND”, or “GROUND” are regarded as ground nets by default.

The format of a symbol-mapped file:

- The file should be in ASCII format.
- The specified signal name should meet HDL syntax.
- The separator of names can be a character " "(space), "\t", or ",".
- The comment line is denoted by the character ";".
- The maximum length of one line is 1024.

An example of a symbol-mapped file is shown below:

---myPower.txt---

myPowerNet1 myPowerNet2

myPowerNet3,myPowerNet4

myPowerNet5

;myPowerNet6

Symbol Library Page

Use this page to specify the settings for symbol library files.

NOTE: The design and the liberty file with power symbol information are reloaded automatically after clicking **Apply** or **OK** button.

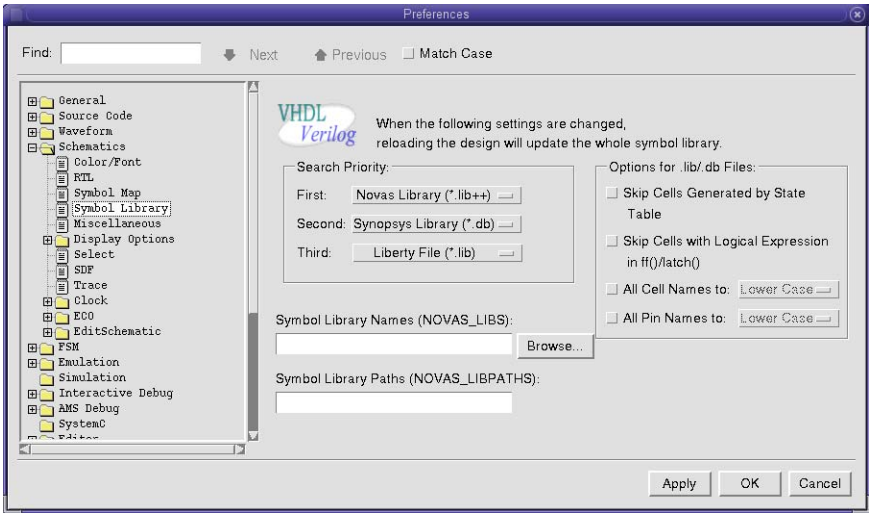


Figure: Symbol Library Page

The following options and fields are available:

- **Search Priority:** This section sets the priority for searching the symbol library source files by selecting the file type from **First**, **Second**, and **Third** selection fields. Each of these selection fields includes **Novas Library** (*.*lib*++), **Synopsys Library** (*.*db*), and **Liberty File** (*.*lib*) file types. By default, the .*lib*++ file has higher priority than .*db* and .*lib* files.
- **Symbol Library Names (NOVAS_LIBS):** Type the symbol library name(s) in this text field or click the **Browse** button to open the *Select Symbol Library* form where the directory structure can be viewed and a symbol library file (*.*lib*++), a liberty file (*.*lib*), or gzipped liberty files (*.*lib.gz)* can be selected as the symbol library (multiple selections are allowed). Separate multiple library names with either a space or the colon (:) character.
  When the symbol library file or the liberty file is selected in the *Select Symbol Library* form, the path of the selected file is added to the **Symbol Library Paths (NOVAS_LIBPATHS)** field automatically.

NOTE: Verdi Power Aware license is not required if you specify Symbol Libraries with power symbol information in **Symbol Library Names** and **Symbol Library Paths** fields.

- **Symbol Library Paths (NOVAS_LIBPATHS):** Specify the symbol library path(s) in this text field. Separate multiple path strings with either a space or the colon (:) character.

The following options are available in the **Options for Liberty File** section:

- **Skip Cells Generated by State Table:** When this option is turned *on*, cells generated by the power state table are skipped.
- **Skip Cells with Logical Expression in ff()/latch():** When this option is turned *on*, cells that have a logic expression in their definition of ff() or latch() definition in a liberty file are skipped.
- **All Cell Names to:** When this option is turned *on*, all cell names can be converted to either lowercase or uppercase. The default value of this option is *off*.
- **All Pin Names to:** When this option is turned *on*, all pin names can be converted to either lowercase or uppercase. The default value of this option is *off*.

Miscellaneous Page

Use this page to specify several miscellaneous viewing options for *nSchema*.

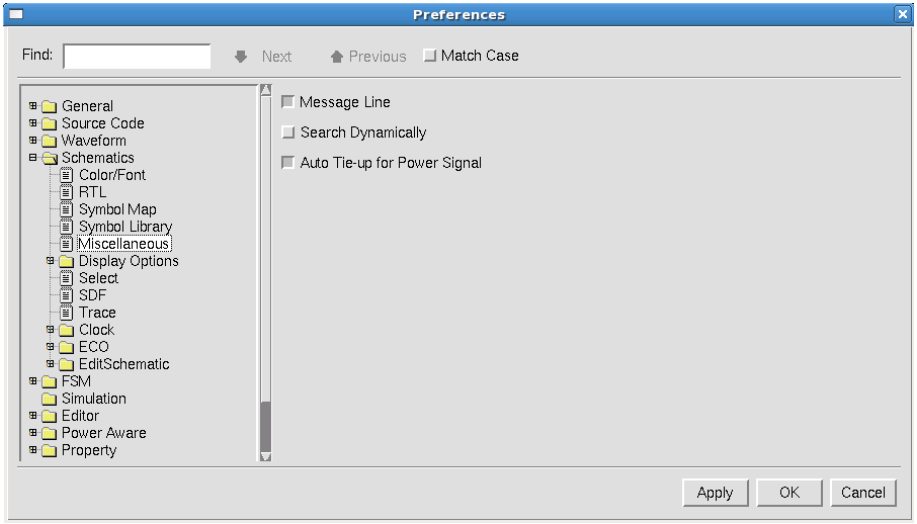


Figure: Miscellaneous Page

The following options are available:

- **Message Line:** When this option is turned *on*, the message line is displayed. The default value of this option is *on*.

- **Search Dynamically:** When this option is turned *on* and a letter is typed in the **Find** text field of the *Find in Current Scope* form (invoked with the **Schematic -> Find in Current Scope** command in the *nSchema* window), dynamic searching finds all the matched instances or signals. When this option is turned *off*, **Enter** must be pressed in the **Find** text field to complete the search.
- **Auto Tie-up for Power Signal:** When this option is turned *on*, VDD, VCC, VEE, VSS, GND, POWER, and PWR signals are tied up automatically. When this option is turned *off*, these signals are not tied up. The default value of this option is *on*.

Display Options Folder

The **Display Options** folder includes **View** and **Schematic** pages.

View Page

Use this page to set the display of objects under the **View** pull-down menu in the *nSchema* window.

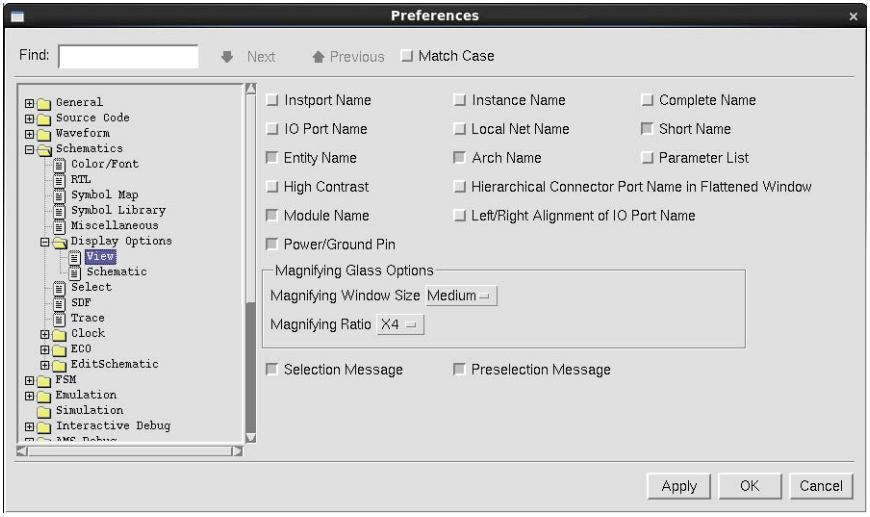


Figure: View Page

The following options are available:

- **Instport Name:** When this option is turned *on*, instance port names are displayed. When this option is turned *off*, instance port names are not displayed. The default value of this option is *off*.
- **IO Port Name:** When this option is turned *on*, input/output port names are displayed. When this option is turned *off*, input/output port names are not displayed. The default value of this option is *off*.
- **Entity Name:** When this option is turned *on*, entity names are displayed. When this option is turned *off*, entity names are not displayed. The default value of this option is *on*.
- **High Contrast:** When this option is turned *on*, the schematic view is dimmed except for the selected objects or results from a trace. The default value of this option is *off*.

- **Module Name:** When this option is turned *on*, module names are displayed. When this option is turned *off*, module names are not displayed. The default value of this option is *on*.
- **Power/Ground Pin:** When this option is turned *on*, power/ground pins are displayed. When this option is turned *off*, power/ground pins are not displayed. The default value of this option is *on*.
- **Instance Name:** When this option is turned *on*, instance names are displayed. When this option is turned *off*, instance names are not displayed. The default value of this option is *off*.
- **Local Net Name:** When this option is turned *on*, local net names are displayed. When this option is turned *off*, local net names are not displayed. The default value of this option is *off*.
- **Arch Name:** When this option is turned *on*, architecture names are displayed. When this option is turned *off*, architecture names are not displayed. The default value of this option is *on*.
- **Hierarchical Connector Port Name in Flattened Window:** When this option is turned *on*, module port names for hierarchical connectors are displayed in flattened schematic windows. When this option is turned *off*, module port names for hierarchical connectors are not displayed in flattened schematic windows. The default value of this option is *off*.
- **Left/Right Alignment of IO Port Name:** When this option is turned *on*, input/output port names are displayed on the left or right side. When this option is turned *off*, input/output port names are displayed on the top or the bottom side. The default value of this option is *off*.
- **Complete Name:** When this option is turned *on*, complete net and module names are displayed. When this option is turned *off*, complete net and module names are not displayed. If the net and the module names are too long and this option is turned *off*, only a partial name is displayed. The default value of this option is *off*.
- **Short Name:** When this option is turned *on*, short names are displayed. When this option is turned *off*, short names are not displayed. The default value of this option is *on*.
- **Parameter List:** When this option is turned *on*, parameter lists are displayed on the module block (note that the parameter list must be defined in the design file). When this option is turned *off*, parameter lists are not displayed. The default value of this option is *off*.

The following options are available in the **Magnifying Glass Options** section to specify the magnifying window size and the magnifying ratio:

- **Magnifying Window Size:** Select either Small, Medium, or Large option as the magnifying window size. By default, Medium is selected as the window size.
- **Magnifying Ratio:** Select either X2, X4, X8, or X16 as the magnifying ratio. By default, X4 is selected as the magnifying ratio.

- **Selection Message:** When this option is turned *on*, it displays information related to the selected object. The default value of this option is *off*.
- **Preselection Message:** When this option is turned *on* and the cursor is moved over an object, it displays information related to the preselected object. The default value of this option is *off*.

Schematic Page

Use this page to set the display of objects under the **Schematic** pull-down menu in the *nSchema* window.

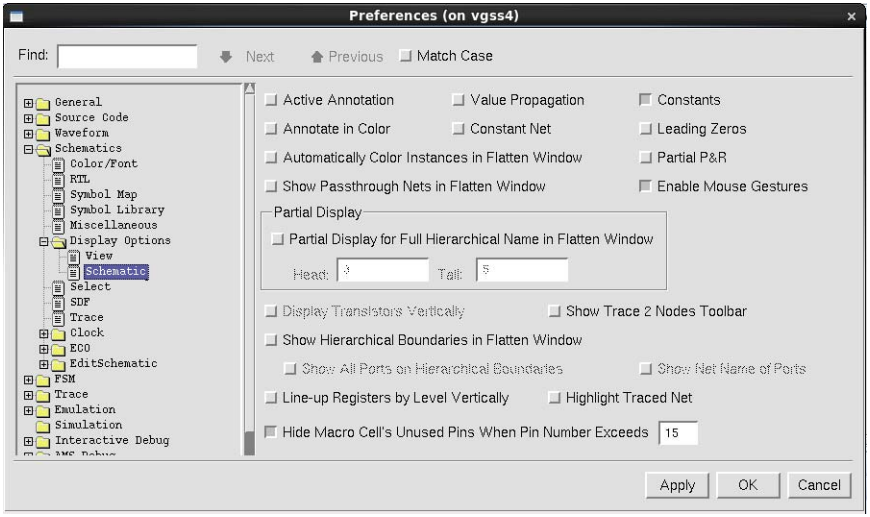


Figure: Schematic Page

The following options are available:

- **Active Annotation:** When this option is turned *on*, active annotation is displayed. When this option is turned *off*, active annotation is not displayed. The default value of this option is *off*.
- **Annotate in Color:** When this option is turned *on*, annotate in color is displayed. When this option is turned *off*, annotate in color is not displayed. The default value of this option is *off*.

- **Automatically Color Instances in Flatten Window:** When this option is turned *on*, *nSchema* automatically highlights instances of different scopes in different colors. When this option is turned *off*, all instances from all scopes have the same color. The default value of this option is *off*.

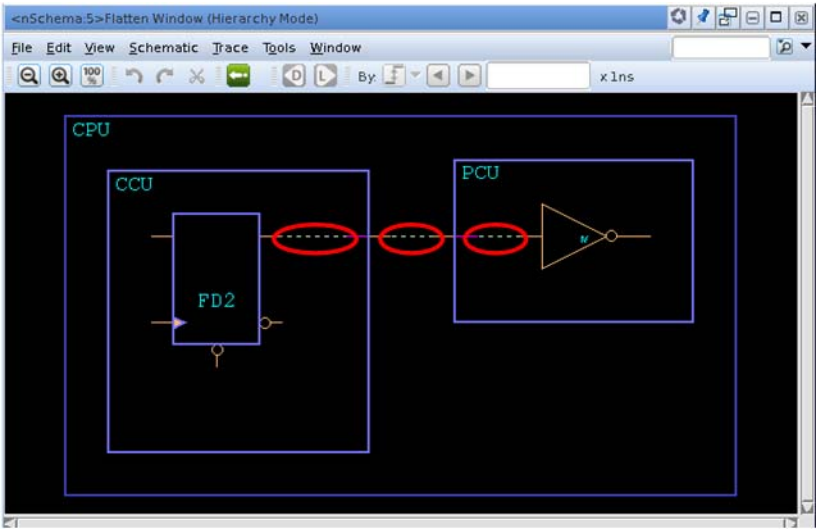
  The following 16 colors are used to highlight instances. These colors are repeated if the number of hierarchical scopes exceeds 16.

  ID_GRAY3, ID_RED3, ID_ORANGE3, ID_YELLOW3, ID_GREEN3, ID_CYAN3, ID_BLUE2, ID_PURPLE2, ID_GRAY6, ID_RED6, ID_ORANGE6, ID_YELLOW6, ID_GREEN6, ID_CYAN6, ID_BLUE6, ID_PURPLE6, ID_UNDEFINED_COLOR.

  If additional viewing objects are added into a flattened schematic window (such as, from trace fan-in or fan-out), the colors are still assigned automatically.

- **Show Passthrough Nets in Flatten Window:** When this option is turned *on*, all the different scopes of passthrough nets are displayed in a flattened schematic window. This option only affects newly opened flattened schematic windows; existing flattened schematic windows are not effected. The default value of this option is *off*.
- **Value Propagation:** When this option is turned *on*, the **Schematic** -> **Propagated Value** command of the *nSchema* window is enabled.
- **Constant Net:** When this option is turned *on*, the **Schematic** -> **Constant Net** command of the *nSchema* window is enabled.
- **Constants:** When this option is turned *on*, constants connected to gates or block symbols in the schematic view are displayed.
- **Leading Zeros:** When this option is turned *on*, leading zeros are displayed on *nSchema* when **Active Annotation** is enabled. When this option is turned *off*, leading zeros are not displayed. The default value of this option is *off*. For changes to take immediate effect, select the **Schematic -> Leading Zeros** command in the *nSchema* window.
- **Partial P&R:** When this option is turned *on*, partial place and route is displayed. When this option is turned *off*, partial place and route is not displayed. The default value of this option is *off*.
- **Enable Mouse Gestures:** When this option is turned *off*, the schematic is zoomed in if any mouse gesture is used. When this option is turned *on*, different mouse gestures can be used to specify the zoom ratio (Zoom Fit/ Zoom In/Zoom Out) and pop-up view of the schematic window. You can view the schematic in a pop-up window, if you drag the schematic to the upper left. The default value of this option is *on*.

- **Partial Display for Full Hierarchical Name in Flatten Window:** When this option is turned *on*, a partial hierarchical name is displayed in a flattened schematic window based on the values set for *Head* and *Tail* display. For example, if *3* is input in *Head* and 5 in *Tail*, the first 3 letters and the last 5 letters of a hierarchy are displayed in the flattened schematic window. When this option is turned *off*, the full hierarchical name is displayed. The default value of this option is *off*.
  - The partially visible net has connections that are not complete in current view. For this net, *nSchema* indicates it with dash-styled lines as shown in the following figure. When you double-click on the partial net, the connectivity of the net expands. You must enable the **Highlight Incomplete Net** from **Schematic** -> **Highlight Incomplete Net** command to enable the display of all the incomplete nets in the current window by highlighting them as dash-lines.



- **Display Transistors Vertically:** When this option is turned *on*, transistor (that is, PMOS/RPMOS/NMOS/RNMOS) symbols are displayed vertically as illustrated in the following figure on right side. When this option is turned *off*, transistor symbols are displayed horizontally as illustrated in the following figure on left side. The default value of this option is *off*.

NOTE: This option can be activated before the design is loaded in *nTrace*. After the design is loaded, this option is disabled.

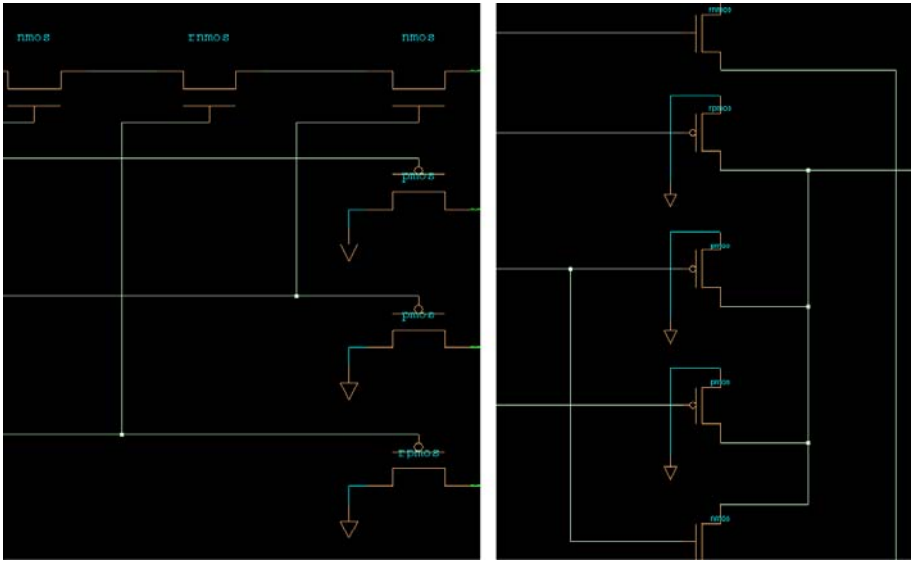


Figure: Example of Turning Display Lengthwise Transistor On (Right) and Off (Left)

- **Show Trace 2 Nodes Toolbar:** When this option is turned on, the Trace 2 Nodes toolbar is displayed in the current *nSchema* window.
- **Show Hierarchical Boundaries in Flatten Window:** When this option is turned *on*, hierarchical boundaries are displayed in the flattened schematic window that shows the *Hierarchy Mode* in the window title. When this option is turned *off*, hierarchical connectors are displayed in the flattened schematic window. The default value of this option is *off*.
  During netlist debugging with a flattened schematic format, displaying the hierarchies and the scopes between cell connections can help to understand the design hierarchy and the scope of the tracing results. Also, understanding the hierarchy boundaries that a path passes through is one of the considerations for looking for the optimal path and layout aspects.

NOTE: It is not supported to display the hierarchical boundaries in the flattened schematic windows used by Clock Analyzer (invoked by the **Clock Analyzer** commands or **Clock Tree** commands in *nTrace* or *nSchema*).

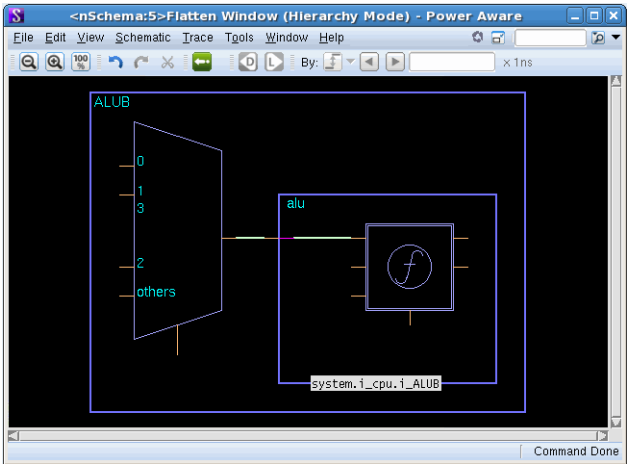


Figure: Results when Show Hierarchical Boundaries in Flatten Window is On

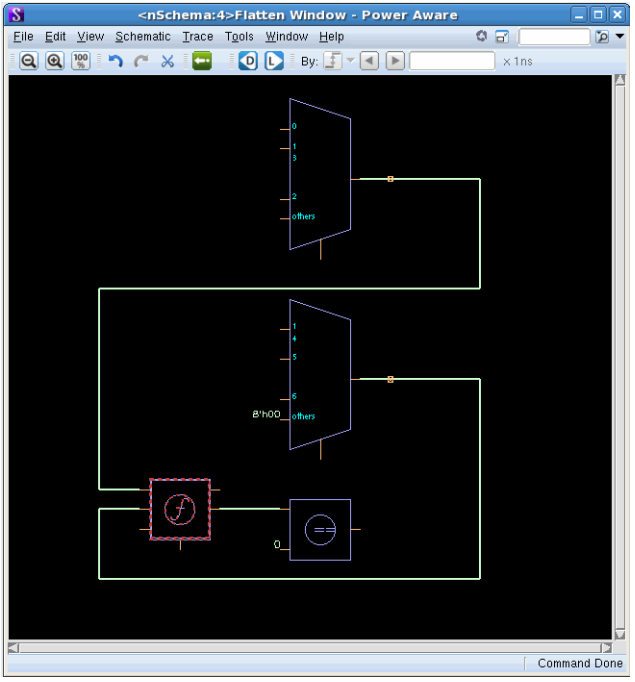


Figure: Results when Show Hierarchical Boundaries in Flatten Window is Off

- **Show All Ports on Hierarchical Boundaries:** After the **Show Hierarchical Boundaries in Flatten Window** option is turned *on*, the **Show All Ports on Hierarchical Boundaries** option is enabled. When the **Show All Ports on Hierarchical Boundaries** option is turned *on*,

all the input/output ports on hierarchical boundaries are displayed in the hierarchical flattened schematic window. When this option is turned *off*, the connected input/output ports are displayed in the hierarchical flattened schematic window. The default value of this option is *off*.

- **Show Net Names of Ports:** After the **Show Hierarchical Boundaries in Flatten Window** option is turned *on*, the **Show Net Names of Ports** option is enabled. When the **Show Net Names of Ports** option and the **Show All Ports on Hierarchical Boundaries** option is turned *on*, the net name which is connected to the input/output ports on hierarchical boundaries appear in the hierarchical flattened schematic window, as illustrated in the following figure. The default value of this option is *off*.

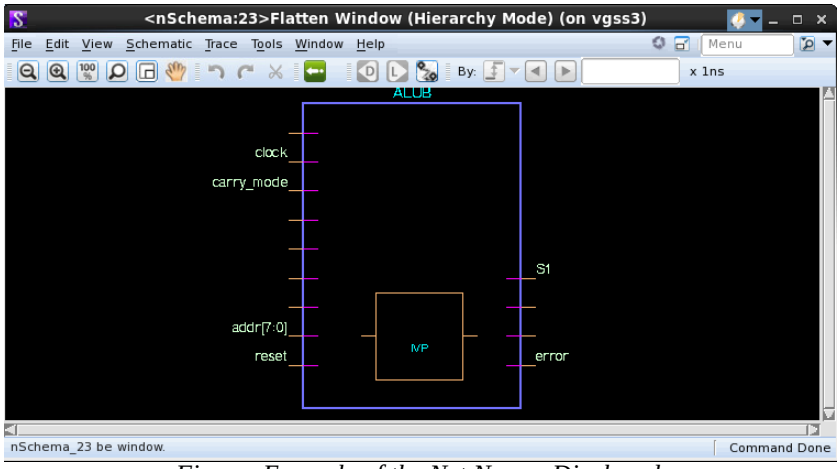


Figure: Example of the Net Names Displayed

- **Line-up Registers by Level Vertically:** When this option is turned *on*, the newly opened *nSchema Flattened Window* performs a vertical line-up for all the register instances by level, counting from right to left of the window. The option is turned *off* by default to ensure backward compatibility.

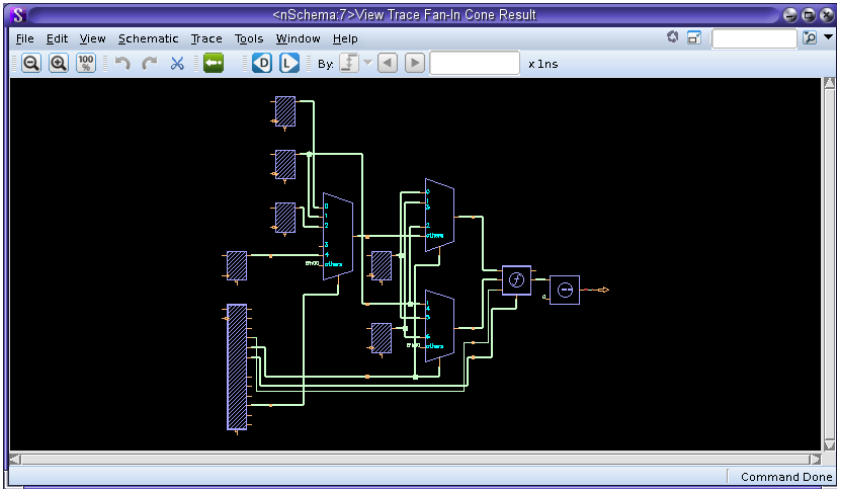


Figure: nSchema Display when Line-up Registers by Level Vertically is Off

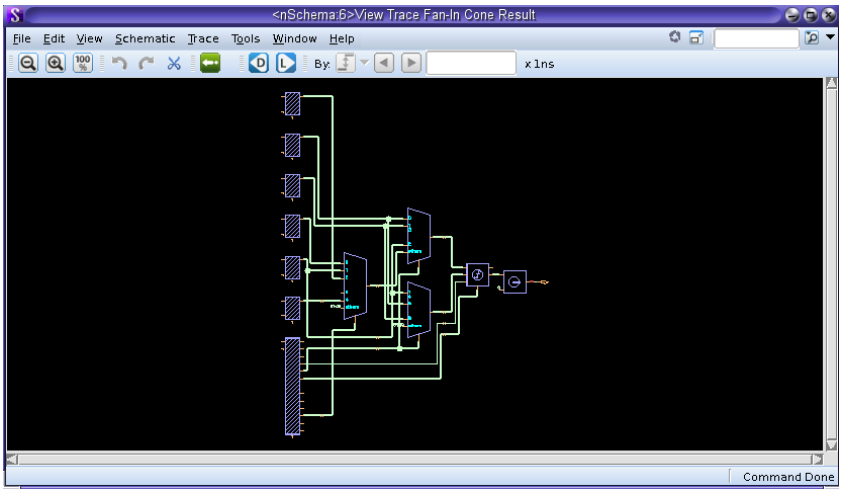


Figure: nSchema Display when Line-up Registers by Level Vertically is On

Select Page

Use this page to specify the selection settings for the *nSchema* window.

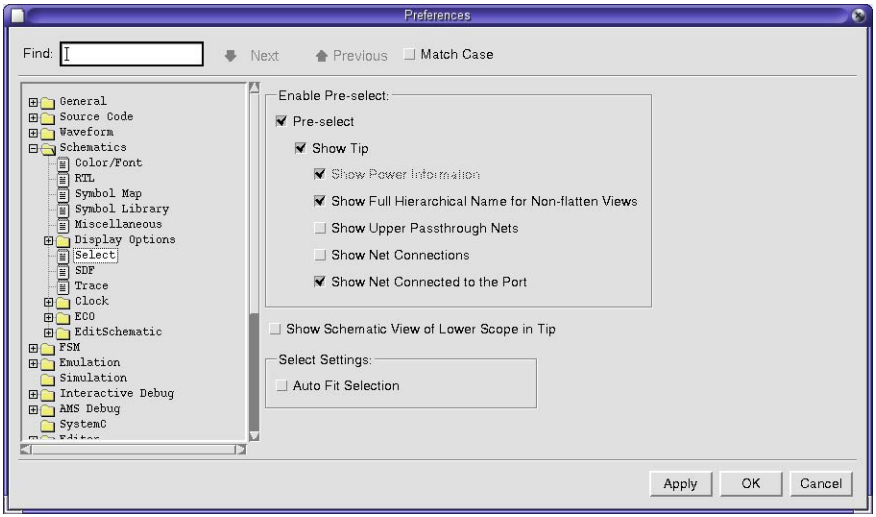


Figure: Select Page

When the **Pre-select** option is enabled, signals or blocks are highlighted when the cursor moves over them in the schematic window. The default value of this option is *on*.

The following options are available when the **Pre-select** option is turned *on*:

- **Show Tip:** When this option is turned *on* and a cursor moves over signals or blocks (for example, instances, ports, and instance pins), the signals or blocks are highlighted and a tip is displayed in the *nSchema* window. If signals or blocks include SDC information, SDC information is shown in a tip. When this option is turned *off*, a tip is not displayed. The default value of this option is *off*.

NOTE: If the **View -> Preselect** toggle command is not enabled, tips for the preselect objects are not shown in the current window.

The following options are enabled when the **Show Tip** option is turned *on*:

- **Show Power Information:** After the CPF/UPF file is loaded and the **Show Tip** option is turned *on*, the **Show Power Information** option is enabled. When this option is turned *on* and a cursor moves over a schematic object that belongs to a power domain or a signal, where the signal type is Isolation, Retention, or Level-shifted, the object is highlighted and a tip is displayed. If the preselect object is a schematic

object, the tip shows power domain, status, and shut-off condition. If the preselect object is a power aware signal, the tip shows the rule name and conditions. If the preselect object is a supply port in the *Power Map* frame, the tip shows the port name, the port direction (that is, input, output, or inout), and the port state. If the preselect object is a power switch output pin in the *Power Map* frame, the tip shows port name and port state. The default value of this option is *on*.

- **Show Full Hierarchical Name for Non-flatten Views:** After the **Show Tip** option is turned *on*, the **Full Hierarchical Name for Non-flatten Views** option is enabled. When the **Full Hierarchical Name for Non-flatten Views** option is turned *on,* the full hierarchical name of instances are displayed in a tip. When this option is turned *off*, the full hierarchical name of instances are not displayed in a tip. The default value of this option is *off*.
  Hierarchical name up to a maximum of six lines is displayed in the tip. When the full name is too long, the tip fails to display the complete hierarchical name.
- **Show Upper Passthrough Nets:** After the **Show Tip** option is turned *on*, the **Show Upper Passthrough Nets** option is enabled. When the **Show Upper Passthrough Nets** option is turned *on*, upper passthrough nets are displayed in a tip with the following format:

  Current port short name
  (upN) short instance name.portN (CELL: moduleN name) (net: net name passed into the port when initializing the instance)

  When this option is turned *off*, upper passthrough nets are not displayed in a tip. The default value of this option is *off*.

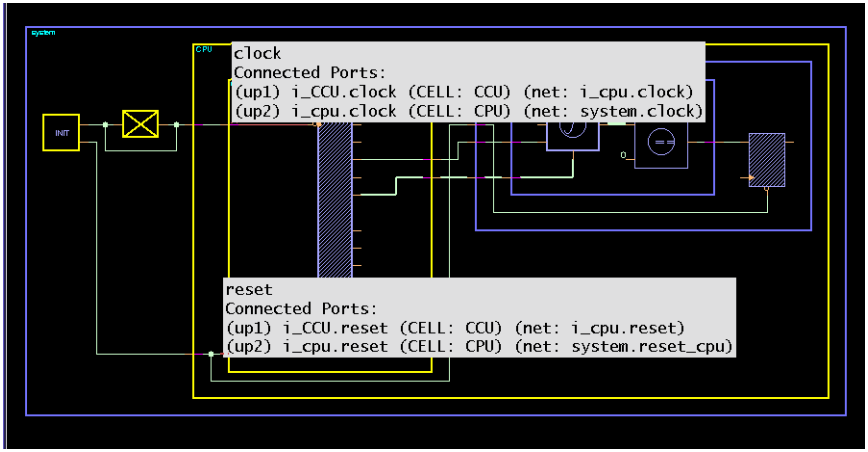


Figure: Example of Tip Display for Upper Passthrough Nets

- **Show Net Connected to the Port**: You must enable the **Show Tip** option to enable the **Show Net Connected to the Port** option. When this option is enabled, the net which is connected to the selected port appears in the tooltip. The port name appears in the first line and the connected net name appears in the second line, as illustrated in the following figure:

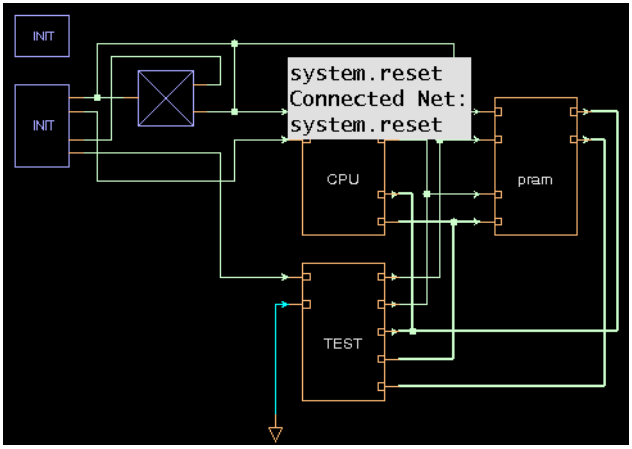


Figure: Example of the Tooltip Displayed for Show Net Connected to the Port Option

---

NOTE: To display the complete name in the tooltip, enable the **Show Full Hierarchical Name for Non-flatten Views** option.

To display the short name in the tooltip, invoke the **Short Name -> View** command in the flattened window.

- **Show Schematic View of Lower Scope in Tip:** When this option is turned *on* and the mouse cursor is moved over an instance in the *nSchema* window, a tip displays the netlist for the instance. The default value of this option is *off*.
- **Auto Fit Selection:** When this option is turned *on*, the selected sets of signals or blocks are automatically fit in the *nSchema* window. The default value of this option is *off*.

SDF Page

Use this page to set the default SDF settings for **Delay Scale**, **Delay Type**, and **Delay Precision** in the *nSchema* window.

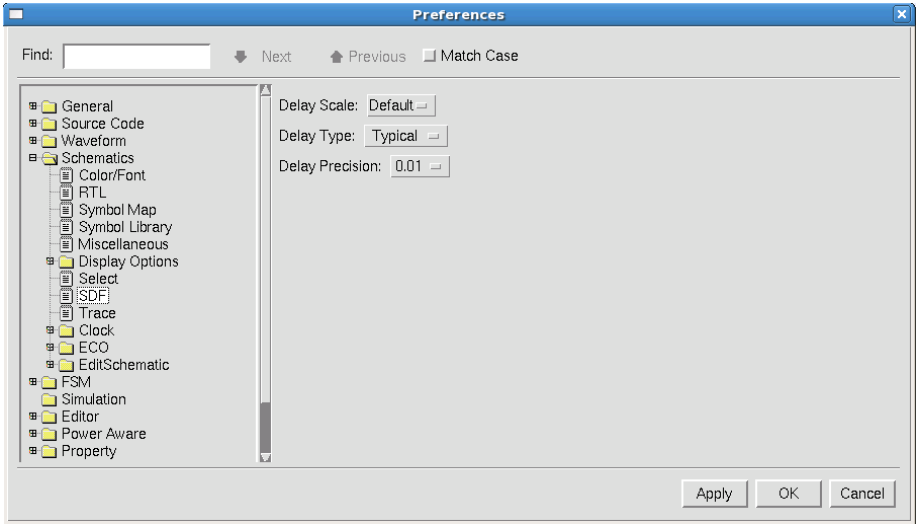


Figure: SDF Page

- **Delay Scale:** Refer to the **Schematic -> Delay Scale - nAnalyzer** command in the *nSchema* chapter for details.
- **Delay Type:** Refer to the **Schematic -> Delay Type - nAnalyzer** command in the *nSchema* chapter for details.
- **Delay Precision:** Refer to the **Schematic -> Delay Precision - nAnalyzer** command in the *nSchema* chapter for details.

Trace Page

Use this page to specify the options for the trace action in *nSchema*.

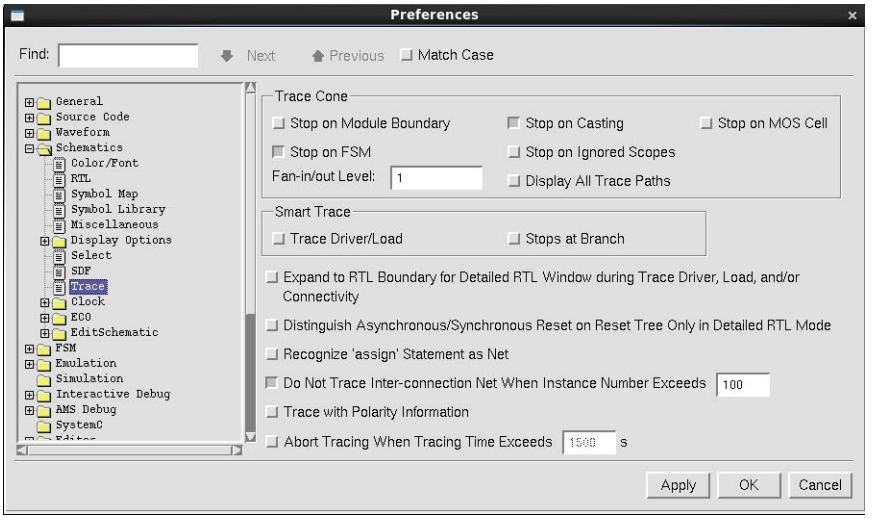


Figure: Trace Page

The following options are available in the **Trace Cone** section:

- **Stop on Module Boundary:** When this option is turned *on*, *nSchema* stops on the module boundary when tracing fan-in cones, fan-out cones, driver, load, or connectivity commands. By default, the option is *off*.
- **Stop on Casting:** When this option is turned *on*, *nSchema* stops at a casting instance when tracing driver, load, or connectivity instances. When this option is turned *off*, *nSchema* passes through casting instances when tracing driver, load, or connectivity instances. The default value of this option is *on*.
- **Stop on MOS Cell:** When this option is turned *on*, *nSchema* stops on MOS cells when tracing fan-in or fan-out cones. The default value of this option is *off*.
- **Stop on FSM:** When this option is turned *on*, *nSchema* stops on FSM when tracing fan-in or fan-out cones. The default value of this option is *on*.
- **Stop on Ignored Scopes:** When this option is turned *on*, the specified scopes (using **Trace -> Options -> Ignore Scopes during Tracing** in *nSchema*) are ignored when tracing schematics. When this option is turned *off*, all scopes are included when tracing schematics. The default value of this option is *off*.

- **Fan-in/out Level:** Enter a number in the **Fan-in/out Level** text field to specify a fan-in or a fan-out level. The default value of this option is *1*.
- **Display All Trace Paths:** When this option is turned *on*, the following three behaviors occur:
  a. If no further driver load exists for the tracing result, *nSchema* stops at the last traced instance pin.
  b. If an instance (cell) pin or port exists at the end of the tracing result, *nSchema* stops at the instance.
  c. If the color of a signal is changed via the **Schematic -> Change Color** command of *nSchema* and the selected signal for the color change is a full-bus or struct signal, the color for partial-bus or struct-member is changed. If the selected signal for the color change is a partial-bus or a struct-member signal, the color for partial-bus or struct-member signals is not changed.

  When this option is turned *off* and no further driver load result exists, the tracing path is not shown. The default value of this option is *off*.

The following options are available in the **Smart Trace** section:

- **Trace Driver/Load:** When this option is turned *on*, *nSchema* automatically traces through single input/output cells (such as, buffers) while tracing a signal's driver or load. See the trace results (cells with yellow highlights) in the figure below. Tracing stops on the first multi-input/output cells in the path.

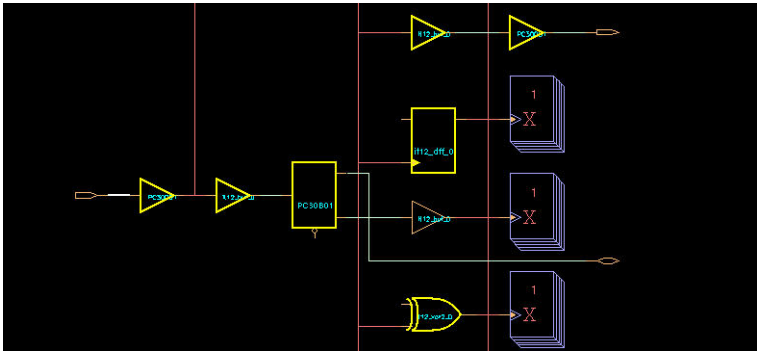


Figure: Output of Turning Trace Driver/Load Option On

When this option is turned *off*, *nSchema* traces to the first driver or load in the path. See the trace result (cell with yellow highlight) in the figure below. The default value of this option is *off*.

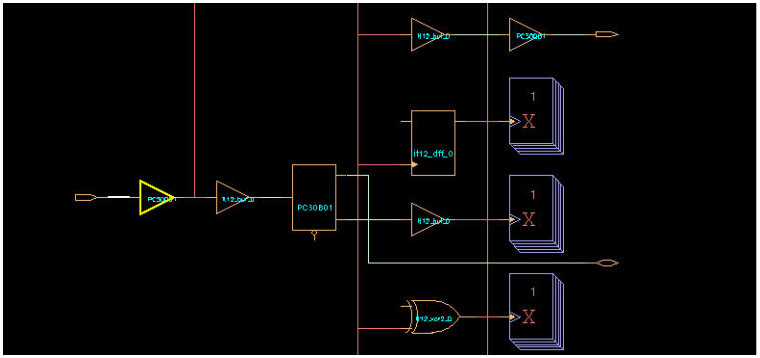


Figure: Output of Turning Trace Driver/Load Option Off

- **Stops at Branch:** This option only works when the **Smart Trace Driver/ Load** option is turned *on*. When this option is turned *on*, *nSchema* stops tracing when multiple driver or load instances exist. When this option is turned *off*, *nSchema* traces a signal's driver or load directly. The default value of this option is *off*.

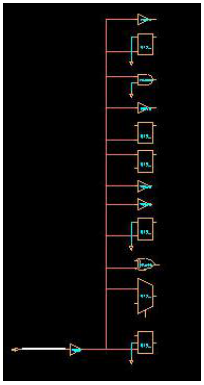


Figure: Output of Turning Stop at Branch Option On

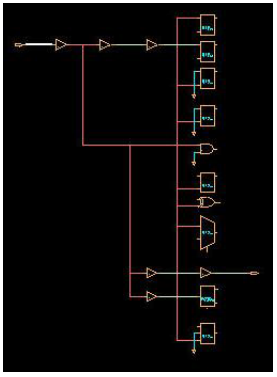


Figure: Output of Turning Stop at Branch Option Off

The following options are also available on the **Trace** page:

- **Expand to RTL Boundary for Detailed RTL Window during Trace Driver, Load and/or Connectivity:** When this option is turned *on*, the net is automatically expanded to the code boundary. The *nSchema* window is not affected until **OK** or **Apply** button is clicked.
- **Distinguish Asynchronous/Synchronous Reset on Reset Tree Only in Detailed RTL Mode:** When there are synchronous and asynchronous resets in a design, this option distinguishes between synchronous and asynchronous resets when tracing. When this option is turned *on*, asynchronous resets and synchronous resets are grouped in different black boxes (identified with ASYNC or SYNC) in the *Reset Tree* window. Cells (RTL cells and symbol register cells) that have less than four port types defined on the cell, are grouped into different black boxes for reset tree results, that is,
  - Asynchronous reset and asynchronous set are grouped in the same black box.
  - Synchronous reset and synchronous set are grouped in the same black box.

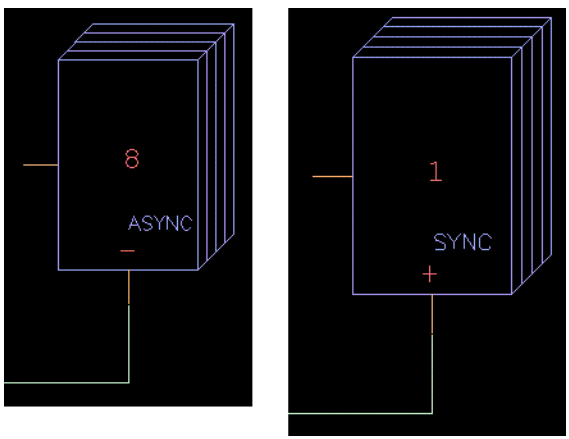


Figure: Compressed Register Symbols Annotated as 'ASYNC' for Asynchronous Reset (Left) and 'SYNC' for Synchronous Reset (Right)

When the option is turned *off*, asynchronous resets, synchronous resets, asynchronous sets, and synchronous sets are all grouped in the same black box. The default value of this option is *off*.

NOTE: If the option is changed from *on* to *off*, existing *nSchema* windows are not affected. However, new *nSchema* windows created by the reset tree command follow the grouping rule.

- **Recognize 'assign' Statement as Net:** When this option is turned *on*, an assign statement is treated as a net while tracing either driver, load, connectivity, fan-in cone, or fan-out cone. When this option is turned *off*, the RTL assignment block in *nSchema* is treated as an instance (with an input and output) during tracing. The default value of this option is *off*.
- **Do Not Trace Inter-connection When Instance Number Exceeds:** When a signal is selected for tracing driver, load, or connectivity, instances of the result set may have inter-connections. An inter-connection means the primary results (first level of gates) may have additional paths to other primary results causing additional connections to be made.

  When this option is turned *off*, the Verdi platform traces the inter-connection nets between instances of trace-result (as shown in the first figure below). When this option is turned *on* and the instance number of trace results (total three) is less than or equal to the value in the text field (a value of three or more), the Verdi platform traces the inter-connection nets between instances of the trace-result. The output is the same as turning this option *off* (as shown in the first figure below). When this option is turned *on* and the instance number of trace results (total three) is greater than the value in the text field (a value of one or two), the Verdi platform does not trace the inter-connection between instances of the trace-result (as shown in the second figure below). The default value of this option is *off*.

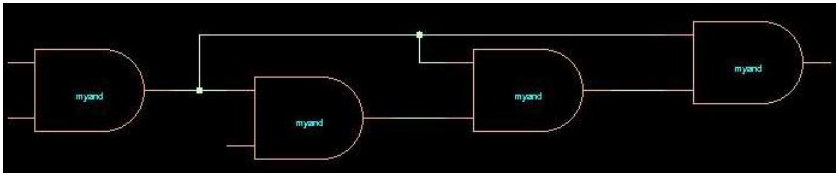


Figure: Results when Do Not Trace Inter-connection Net Option is Off

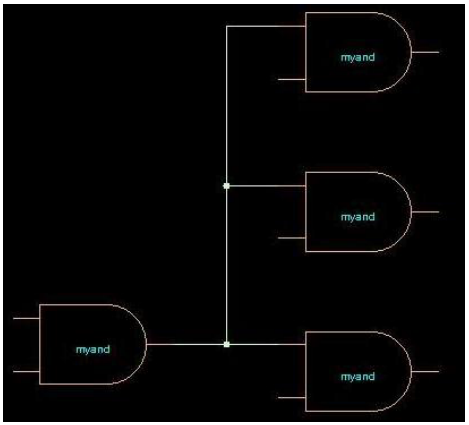


Figure: Results when Do Not Trace Inter-connection Net Option is On

- **Trace with Polarity Information:** When this option is turned *on*, tracing in *nSchema* only includes paths that contain polarity information that is defined from an input port to an output port of one instance. Following is an example of paths in the trace result that have polarity information:

```
[
   CELL HMACRO1 MACRO
   I1  INPUT L D H
   I2  INPUT L D H
   OUT OUTPUT R COM H
   POLARITY I1 OUT +
 ]
```

  For cell `HMACRO1`, it has input `I1`, input `I2` and output `OUT`. If `I1->OUT` has polarity and `I2->OUT` does not have polarity, the path that includes `I1->OUT` can be in the trace result, but the path that includes `I2->OUT` cannot be in the trace result.

  When this option is turned *off*, *nSchema* traces without polarity information. The default value of this option is *off*.

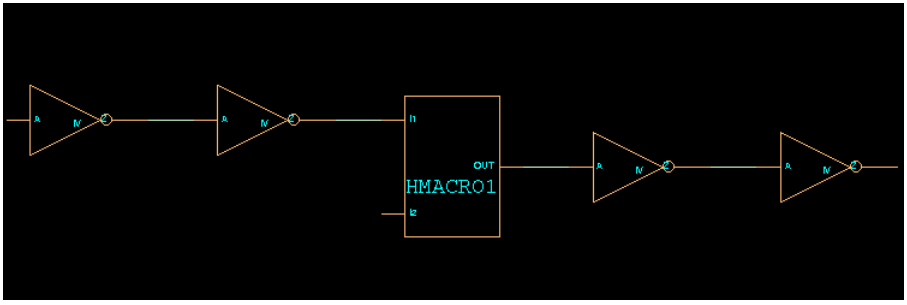


Figure: Results When Trace with Polarity Information Option is On

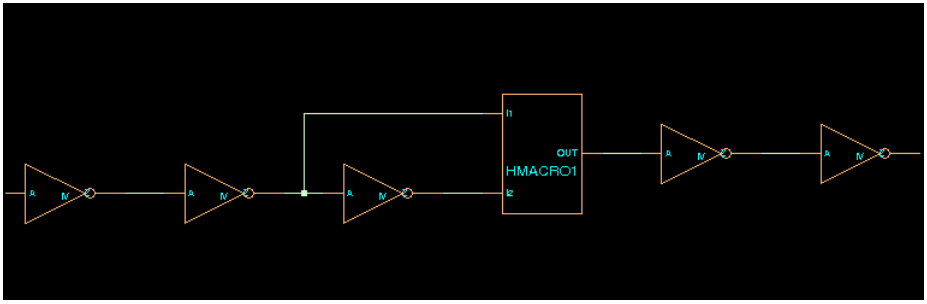


Figure: Results When Trace with Polarity Information Option is Off

- **Abort Tracing When Tracing Time Exceeds:** When this option is *on*, then the tracing automatically stops when the tracing exceeds the time out range provided. A warning message appears mentioning that the tracing is terminated. For example, if **1500 s** is provided as the timeout range then after **1500 s** the tracing is stopped.

  Also, if there are five paths between two points and three paths are traced within the time out range, then the three paths appear. If the traced path is not completed, then no partial result is displayed. By default, this option is *off*.

Clock Folder

The **Clock** folder includes **Color**, **Auto Merge**, and **Trace** pages.

Color Page

Use the color palate to specify the highlight colors for marker, selected row, and violation points in the *Clock Tree Browser* window.

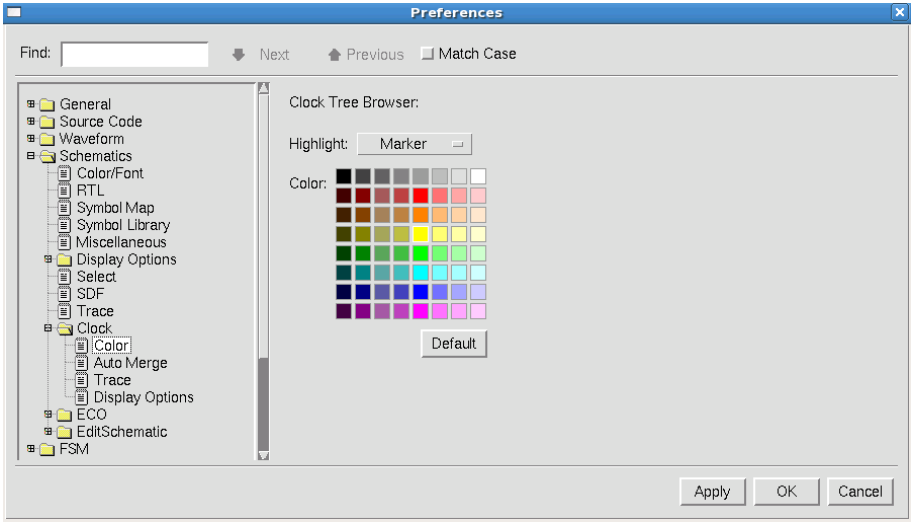


Figure: Color Page

Highlight: Select the type of object to change color attributes for. Options are as follows (only one of these options may be selected at a time):

- **Marker:** Specify the marker color in the *Clock Tree Browser* window. The default is *yellow*.
- **Selected Row:** Specify the color of the selected row in the *Clock Tree Browser* window. The default is *blue*.
- **Violation Point:** Specify the color of the violation point in the *Clock Tree Browser* window. The default is *red*.

Auto Merge Page

Use this page for settings of the merged cell group.

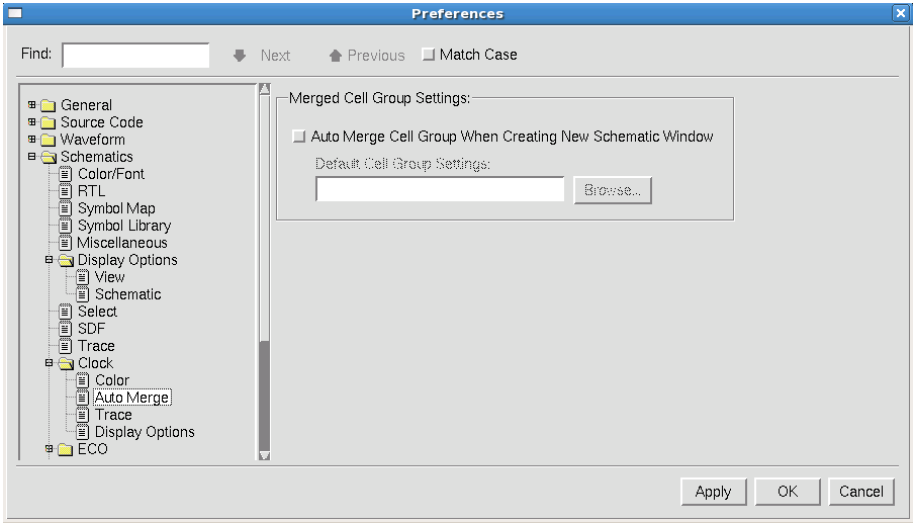


Figure: Auto Merge Page

Auto Merge Cell Group When Creating New Schematic Window: When this option is turned *on*, the merged rule file can be specified in the **Default Cell Group Settings** field. The merged rule is automatically applied to the newly opened clock tree schematic windows the next time a clock tree schematic window is opened.

The syntax of the merged setting log is listed below.

```
AutoMergeCellList:
Cell Name 1
Cell Name 2
Cell Name 3
...
ExcuInstList:
Full Instance Name 1
Full Instance Name 2
Full Instance Name 3
```

Trace Page

Use this page to set the trace behavior.

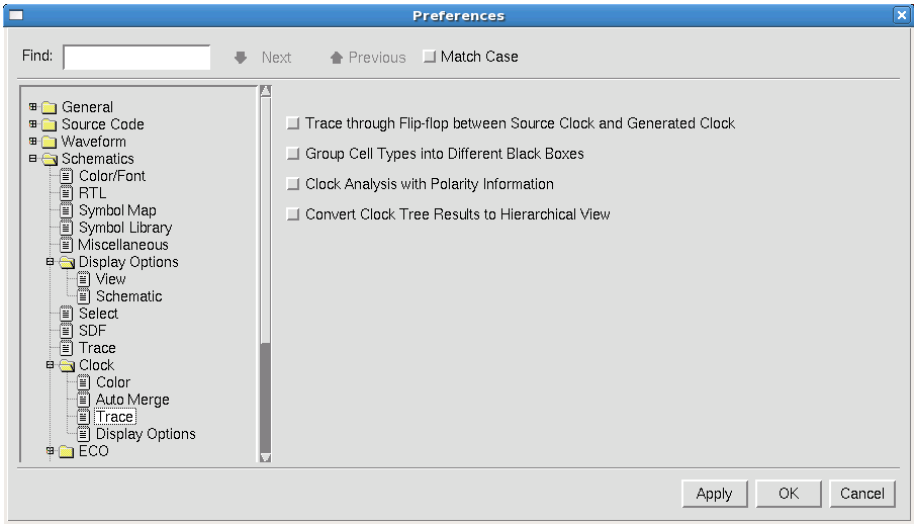


Figure: Trace Page

The following options are available:

- **Trace through Flip-Flop between Source Clock and Generated Clock:** When this option is turned *on*, flip-flops between the source clock and its generated clock can be traced through. When this option is turned *off*, flip-flops between the clock source and its generated clock cannot be traced through. The default value of this option is *off*.
- **Group Cell Types into Different Black Boxes:** When this option is turned *on*, it shows group cell types in different black boxes in the clock tree view. The default value of this option is *off*.
- **Clock Analysis with Polarity Information:** When this option is turned *on*, the tracing results of creating a clock tree, extracting clock domain, and checking the crossing path are affected by the polarity information in the symbol library. With polarity information, the connection between pins (in particular IO pins) can easily be determined. The default value of this option is *off*.
- **Convert Clock Tree Results to Hierarchical View:** This option provides the ability to view clock tree results in a flattened view or in a hierarchical view. After selecting a clock tree in the *Clock Tree Browser* window or the *Clock Domains* window, invoke the **Tools -> New Schematic** command or click the  icon to view the selection in the *nSchema* window. When this

option is turned *on*, the clock tree results are shown in the top scope view (module instance block(s) with full-hierarchical instance name, flip flop number, and the trigger type: '+' indicates rising; '-' indicates falling; 'x' indicates unknown). Double-click the module instance block to expand and view instances in the lower scope. To collect the expanded instances to the upper scope, invoke the **Collect Black Box by Scope** command from the right-click command menu. When dragging the instance block(s) from the *nSchema* window and dropping it (them) to the *Clock Tree Browser* or *Clock Domains* window, the flip-flop (not including combinational logic) in the selected block(s) is highlighted in *Clock Tree Browser* or *Clock Domains* window. When dragging the selected clock tree from the *Clock Tree Browser* window or the *Clock Domains* window to the *nSchema* window, if the clock tree can be found in the *nSchema* window, the block of the clock tree is highlighted in the *nSchema* window. When this option is turned *off*, clock tree results are displayed in a flattened view (in black box format). The default value of this option is *off*.

NOTE: This option must be set before clock extraction.

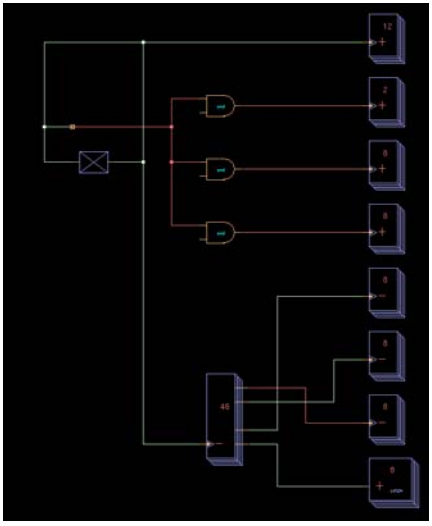


Figure: Clock Tree Results in Flattened View

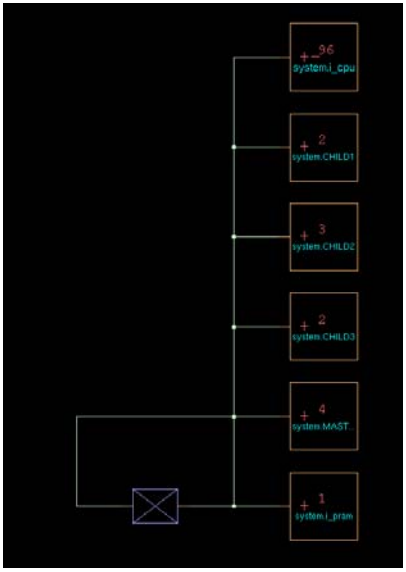


Figure: Clock Tree Results in Hierarchical View

Display Options Page

Use this page for display settings.

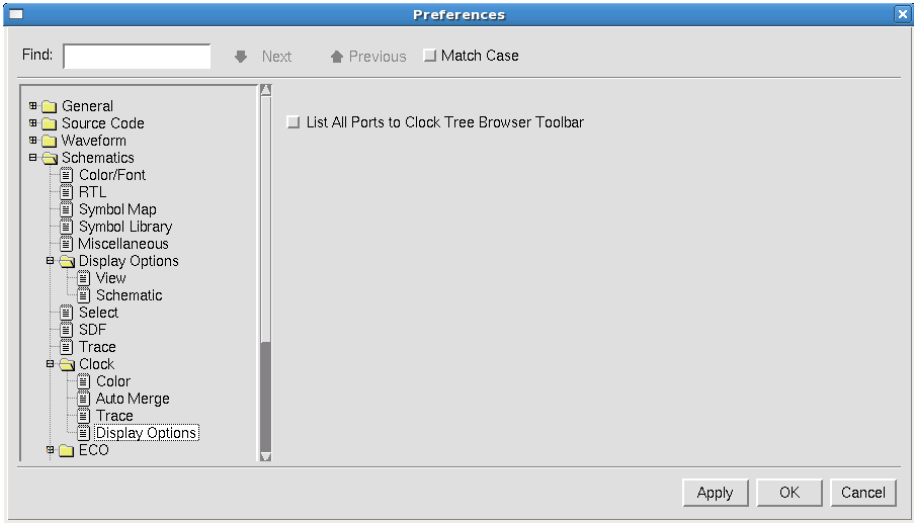


Figure: Display Options Page

List All Ports to Clock Tree Browser Toolbar: When this option is turned *on*, all ports of the selected instance node are displayed in the **Port** selection field in the *Clock Tree Browser* window. When this option is turned *off*, ports of the

selected instance node that the clock tree has traced are displayed in the **Port** selection field. The default value of this option is *off*.

ECO Folder - nECO

The **ECO** folder includes **General, Freeze Silicon, Color, Script File**, and **Netlist** pages.

General Page

Use this page to specify the comment format and naming rules.

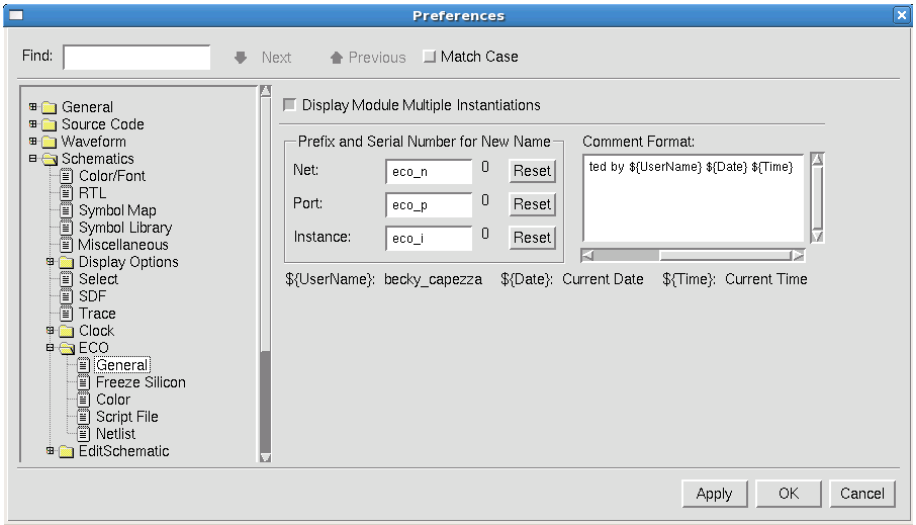


Figure: General Page

The following option and fields are available:

- **Display Module Multiple Instantiations:** When this option is turned *on*, a message appears when the *nECO* window is opened indicating that the module has multiple instantiations.
- **Net:** Enter the text to be used as the prefix for new nets. The *nECO* predefined variables that are described in the **Comment Format** section can also be specified.
- **Port:** Enter the text to be used as the prefix for new ports. The *nECO* predefined variables that are described in the **Comment Format** section can also be specified.

- **Instance:** Enter the text to be used as the prefix for new instances. The *nECO* predefined variables that are described in the **Comment Format** section can also be specified.
- **Comment Format:** Specify the fixed content for comments. *${UserName}*, *${Date}*, and *${Time}* can also be used in the comment field for user name, date, and time.

Freeze Silicon Page

Use this page to specify the options for *nECO* Freeze Silicon mode.

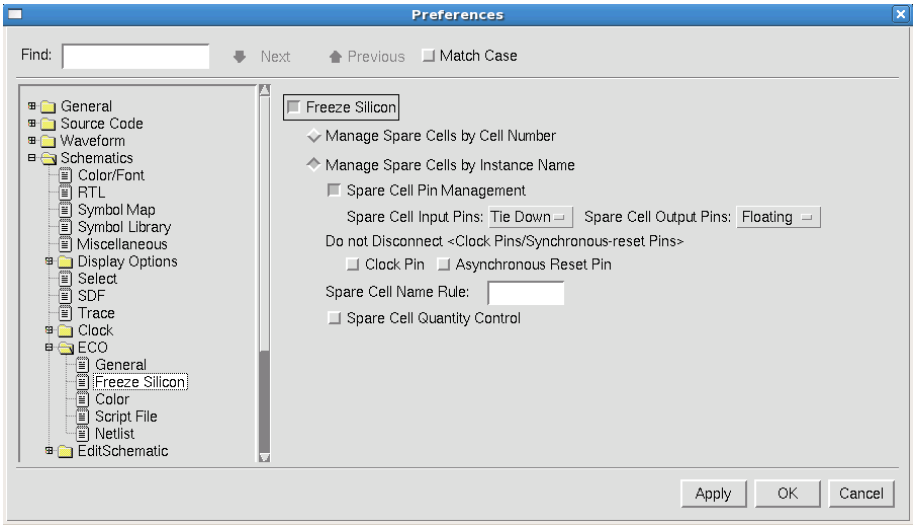


Figure: Freeze Silicon Page

When the **Freeze Silicon** option is turned *on*, only spare cells can be used for either managing spare cell numbers or managing new instances. When this option is turned *off*, any cells can be used for ECO changes.

The following options are available for **Freeze Silicon** mode:

- **Manage Spare Cells by Cell Number**

  When this option is selected, instances by cell names can be added. *nECO* assigns instances using the settings for adding instances in Non-Freeze Silicon mode. **Spare Cell** and **Count Spare Cell** commands update the number of the available spare cells. The deleted instances are set as spare cells for future use.

  This option is provided because some physical placement and route tools can add new instances under Freeze Silicon mode. When the updated netlist

is loaded into the tools, the tools can then replace the new instances with existing spare cell instances properly.

- **Manage Spare Cells by Instance Name**

  When this option is selected, existing spare cells for design changes can be used. The deleted instances are set as spare cells for future use.

  - **Spare Cell Pin Management:** When this option is turned *on*, the following additional options are available:

    Spare Cell Input Pins: Specify the spare cell input pins as **Tie Up, Tie Down,** or **Floating**. The selected input is displayed in the menu command. The default value of this option is **Tie Down**.

    Spare Cell Output Pins: Specify the spare cell output pins as **Tie Up, Tie Down,** or **Floating**. The selected output is displayed in the menu command. The default value of this option is **Floating**.
  - **Do Not Disconnect <Clock Pins/Sync-reset Pins>:** Specify whether to connect or disconnect clock and asynchronous reset pins. Two options are available: **Clock Pin** and **Async-Reset Pin.** When the **Clock Pin** option is turned *on*, clock pins stay connected after instance-related operations. When the **Async-Reset Pin** option is turned *on*, asynchronous reset pins stay connected after instance-related operations. When these options are turned *off*, the related pins are disconnected.
  - **Spare Cell Name Rule:** Enter the text to be used as the prefix for new spare cells or to search the spare cell list.
  - **Spare Cell Quantity Control:** Use this option to control spare cells by number.

    When this option is turned *on* and a spare cell is selected, the right-click command menu includes three sub-options: **Add Buffer -> Add Buffer by Name**, **Add Instance -> Add Instance by Name**, and **Replace Instance -> Replace Instance by Name**. When this option is turned *off*, there are no sub-options in **Add Instance** and **Replace Instance** commands on the right-click command menu and the **Add Buffer** command is disabled.

    The option also affects the display results for the *Spare Cell* form, which is opened by invoking the **Edit** -> **Spare Cell** command or by the equivalent right-click command. When this option is turned *on*, the **Number List** section is shown on the *Spare Cell* form. More information about the **Number List** is available in the **Spare Cell** command. When this option is turned *off*, the **Number List** section is not displayed. See the following figure for the display differences.

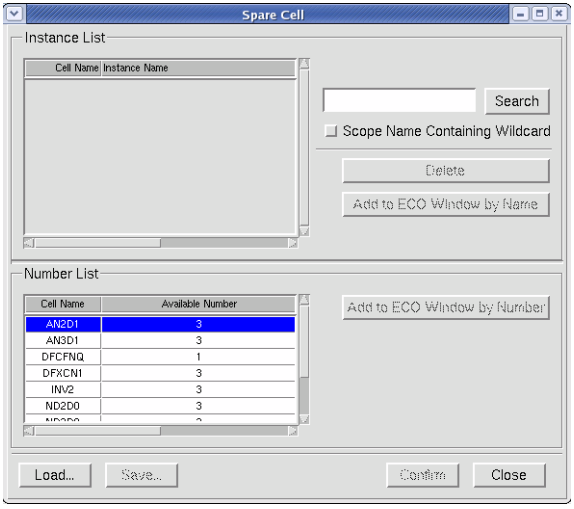


Figure: Display of the Spare Cell Form When Spare Cell Quantity Control is On

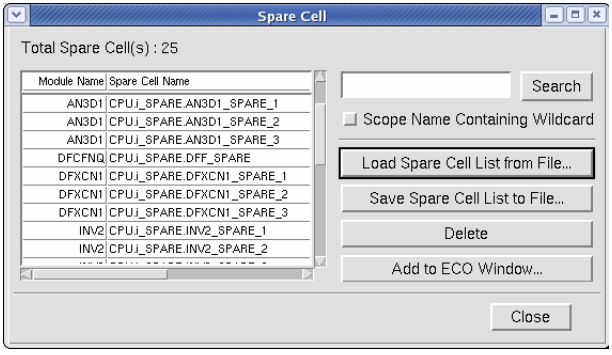


Figure: Display of the Spare Cell Form When Spare Cell Quantity Control is Off

Color Page

Use this page to specify the highlight color for changed components.

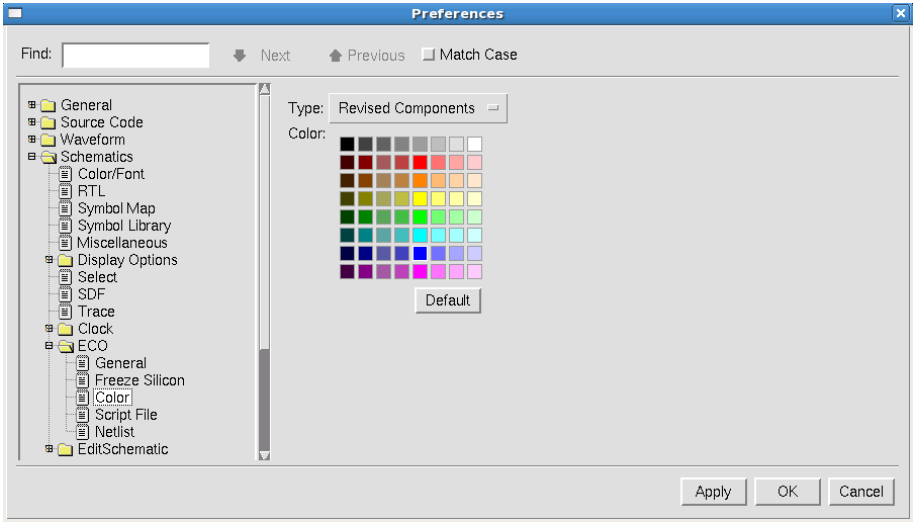


Figure: Color Page

- **Type:** Change the color for the following types:
  - **Revised Components:** Indicates instances, nets, and pins that have committed changes.
  - **Spare Cell:** Indicates the spare cell.
- **Default:** Reset to the original colors.

Script File Page

Use this page to specify the ECO output script file formats. The settings must be completed before making changes in the *nECO* window.

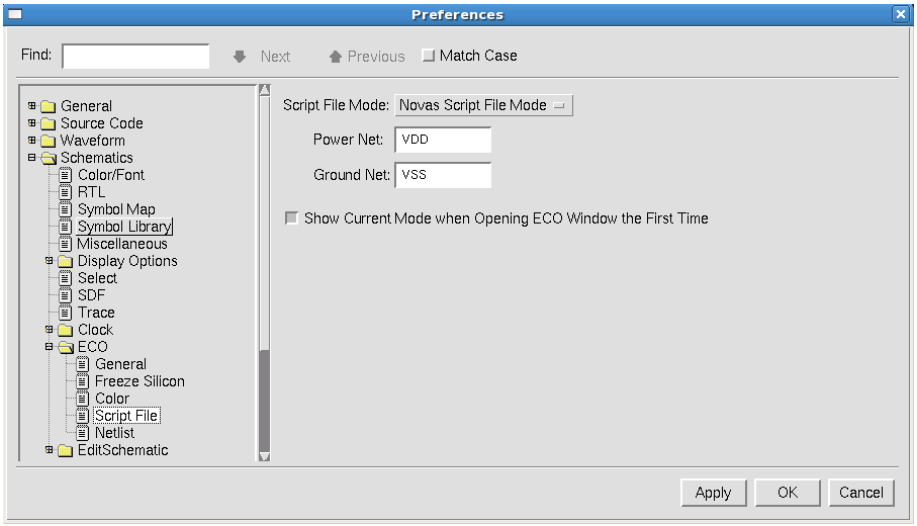


Figure: Script File Page

The following options and fields are available:

- **Script File Mode:** Select the ECO output script format from one of the following options:
  - **Novas Script File Mode:** Outputs the ECO script to Novas format.
  - **ICC Script File Mode:** Saves the ECO script to ICC (IC Compiler) format.
  - **FE Script File Mode:** Saves the ECO script to First Encounter format.

  Power Net/ Ground Net: In these two text fields, enter the power and ground net names to be substituted when producing the specified script format.

  In the **Novas Script File Mode**, **Power Net** and **Ground Net** fields are disabled.

- **Show Current Mode when Opening ECO Window the First Time:** When this option is turned *on*, a prompting window shows the current mode when an *nECO* window is opened. When this option is turned *off*, the current mode is not indicated.

Netlist Page

Use this page to specify the netlist settings.

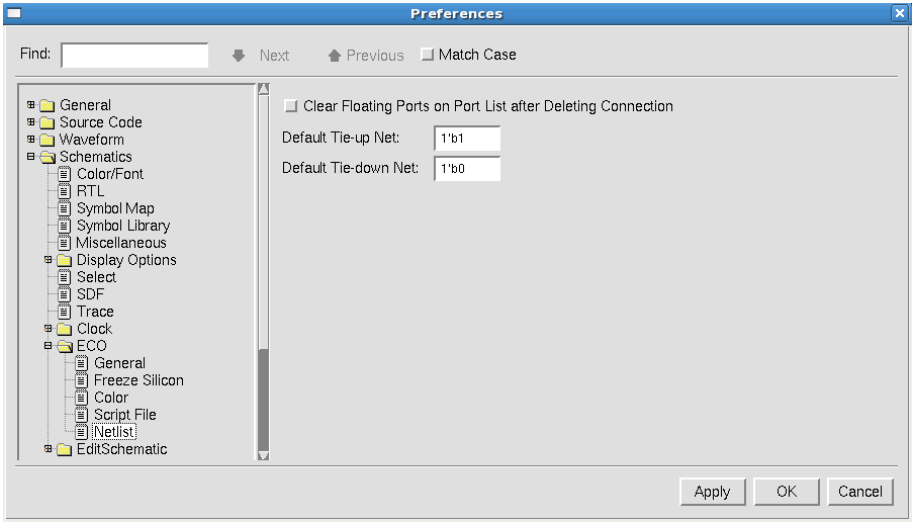


Figure: Netlist Page

The following options are available:

- **Clear Floating Ports on Port List after Deleting Connection:** When this option is turned *on* and a connection is deleted, *nECO* does not show the description for the floating ports in the source code pane. When this option is turned *off*, *nECO* displays the pin connection description in the source code frame. The default is *off*.
- **Default Tie-Up Net:** This text field is used to specify local scope tie-up nets. The default value is *1'b1*.
- **Default Tie-Down Net:** This text field is used to specify local scope tie-down nets. The default value is *1'b0*.

Edit Schematic Folder

The **EditSchematic** folder includes **Color, Font, Pin/Port,** and **Page View** pages.

NOTE: The **EditSchematic** folder includes *Color* and *Font* pages and hence, does not support global font.

Color Page

Use this page to specify the color options for the selected type.

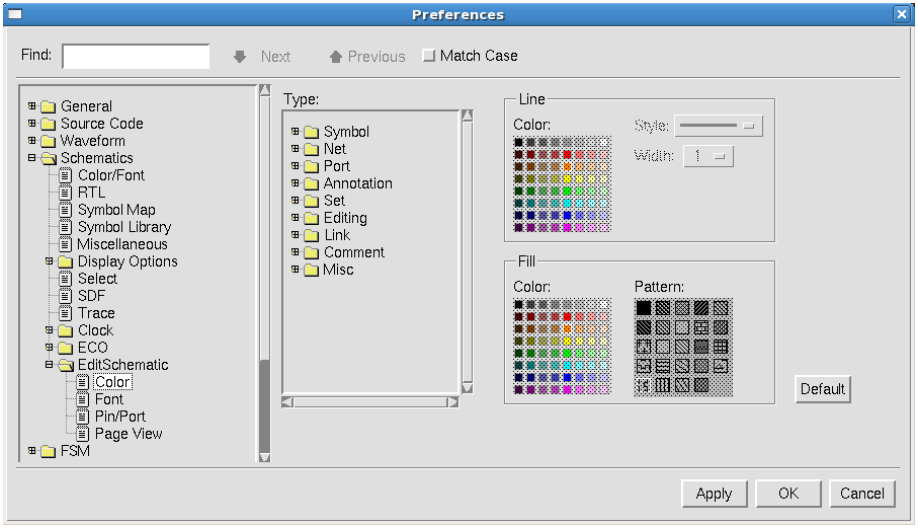


Figure: Color Page

For the selected type in the **Type** field, **Line Color**, **Style**, and **Width** can be changed as well as **Fill Color** and **Pattern**.

Type: Select the type. Click the plus (+) symbol in front of the folder to expand the folder and see the available options.

After a type is selected, **Line** and **Fill** sections are enabled and the options can be changed.

Font Page

Use this page to specify the font type and the color for the selected object.

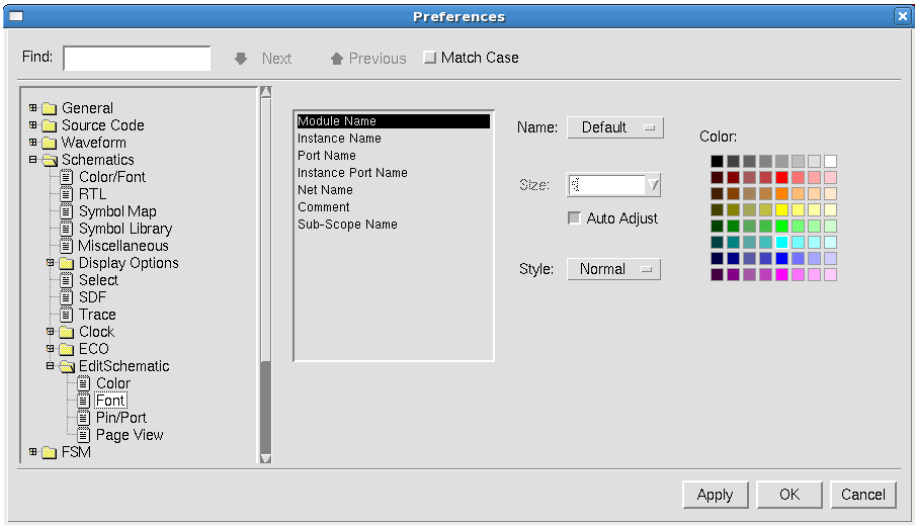


Figure: Font Page

Type: Specify the type of object for which to change the color attributes by selecting one of the following options: **Module Name, Instance Name, Port Name, Instance Port Name, Net Name, Comment**, and **Sub-Scope Name**.

Name: Specify a font style by selecting one of the following options: **Default, AGoth, ARoman,** or **ASlab**. The default value is **Default**.

Size: Specify the font size. This field is selectable when the **Auto Adjust** option is turned *off*.

Auto Adjust: When this option is turned *on*, the font size is automatically adjusted. When this option is turned *off*, the desired font size can be selected. The default value of this option is *on*.

Style: Specify the font style by selecting one of the following options: **Normal, Bold, Italic**, or **Bold Italic**. The default value of this option is **Normal**.

Color: Select (left-click) the target color in the color matrix.

Pin/Port Page

Use this page to specify the pin and the port settings.

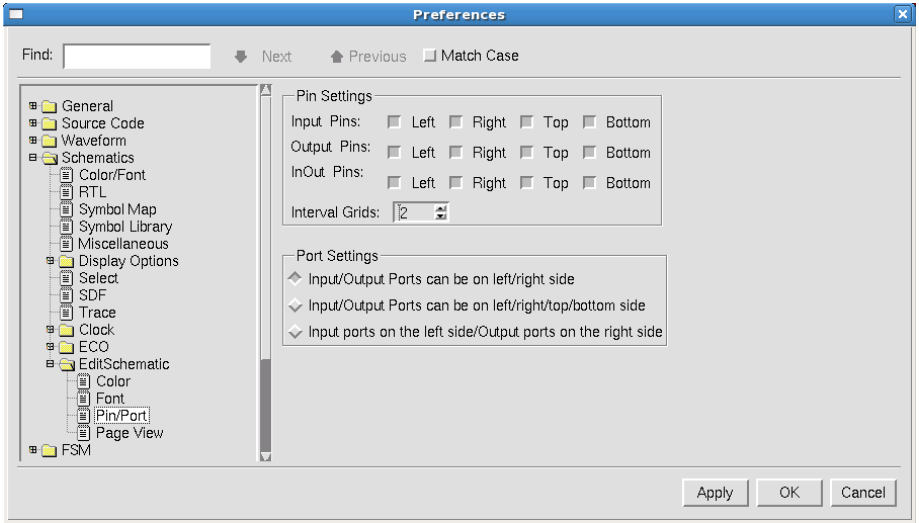


Figure: Pin/Port Page

The following options are available in the **Pin Settings** section:

- **Input Pins:** Specify the available sides for input pins by turning **Left, Right, Top,** and **Bottom** options *on*. All sides are enabled by default.
- **Output Pins:** Specify the available sides for output pins by turning **Left, Right, Top,** and **Bottom** options *on*. All sides are enabled by default.
- **InOut Pins:** Specify the available sides for input/output pins by turning **Left, Right, Top,** and **Bottom** options *on*. All sides are enabled by default.
- **Interval Grids:** Specify the distance between pins.

The following options are available in the **Port Settings** section (only one option can be selected at a time):

- **Input/Output Ports can be on left/right side:** When this option is selected, the input and the output ports can only be on left and right sides of the instance.
- **Input/Output Ports can be on left/right/top/bottom side:** When this option is selected, the input and the output ports can be on any side of the instance.
- **Input ports on the left side/Output ports on the right side:** When this option is selected, the input ports can only be on the left side and the output ports can only be on the right side of the instance.

Page View Page

Use this page to specify the page view for the editable schematic.

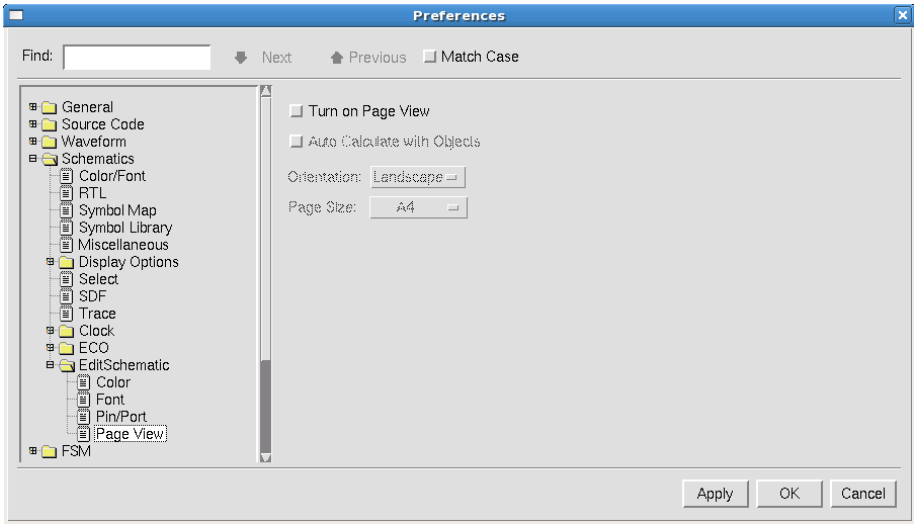


Figure: Page View Page

The following options are available:

- **Turn on Page View:** When this option is turned *on*, the page view is enabled and the following options can be specified. The default value of this option is *off*.
  - **Auto Calculate with Objects:** When this option is turned *on*, the number of pages are automatically calculated based on the objects. The default value of this option is *off*.
  - **Orientation:** Specify the orientation by selecting either **Landscape** or **Portrait**.
  - **Page Size:** Specify the page size by selecting from one of the following options: **Letter, A4, A3, A2, A1, B, C, D**, or **E**. The default value of this option is **A4**.

FSM Folder

The **FSM** folder includes **Color** and **View Options** pages.

Color Page

Use this page to define line color, line width, fill, and text for each object type in the *nState* window.

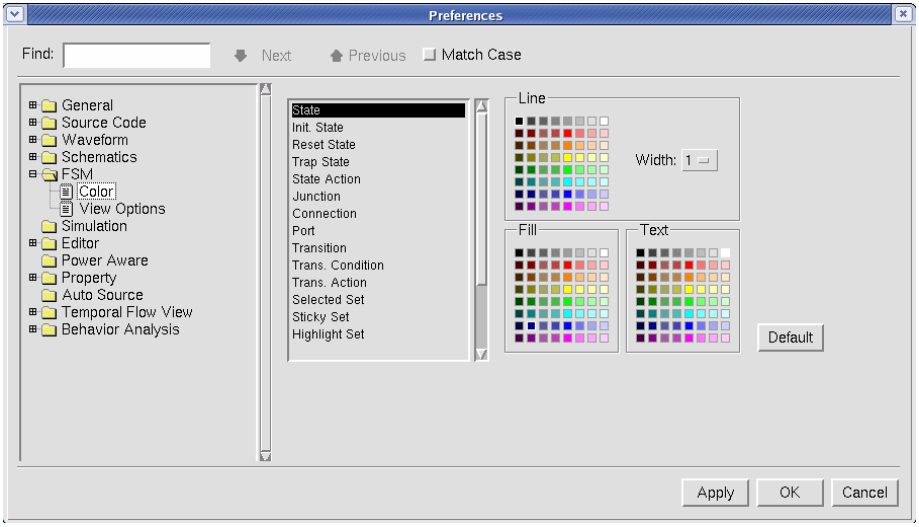


Figure: Color Page

View Options Page

Use this page to specify the viewing options in *nState*.

NOTE: Setting changes in this page are NOT applied immediately after clicking **Apply** or **OK** buttons. Only after a new *nSchema* window is created and the *nState* window is opened from it, the preference changes are applied.

The **Using Vector Font** option is available and hence does not support global font

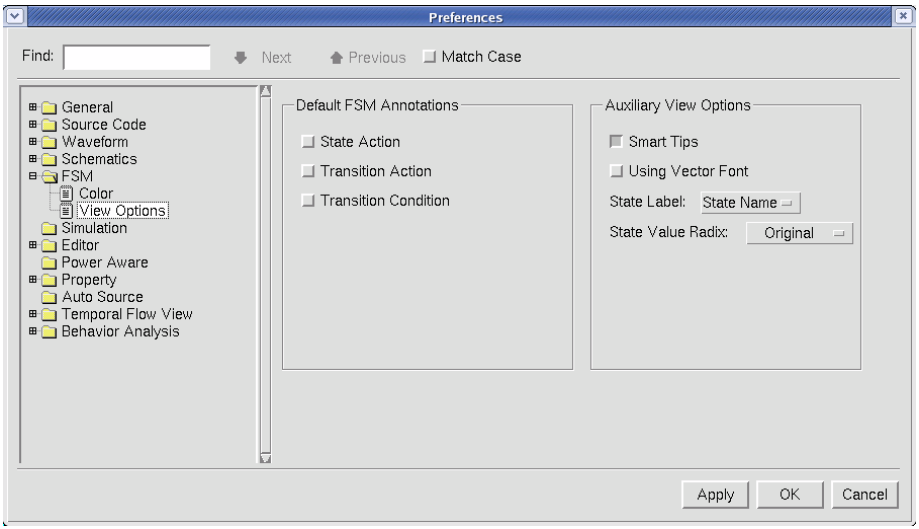


Figure: View Options Page

The following options are available in the **Default FSM Annotations** section:

- **State Action:** When this option is turned *on*, *State Actions* are shown in the state diagram. The default value of this option is *off*.
- **Transition Action:** When this option is turned *on*, *Transition Actions* are shown in the state diagram. The default value of this option is *off*.
- **Transition Condition:** When this option is turned *on*, *Transition Conditions* are shown in the state diagram. The default value of this option is *off*.

The following options are available in the **Auxiliary View Options** section:

- **Smart Tips:** When this option is turned *on* and the mouse cursor is placed on a state or a transition, a smart tip shows the name for the state and the

transition conditions when they are invisible or blurred in the *nState* window. The default value of this option is *on*.

- **Using Vector Font:** When this option is turned *on*, state action and condition labels, and transition action and condition labels are enlarged to the maximum size. The default value of this option is *off*.
- **State Label:** Specify what is displayed in a state in the *nState* window by selecting one of the following options: **State Name**, **State Value**, or **Both**. The display of state values is controlled by the **State Value Radix** setting. The format for the display of both state name and state value is State Name(State Value) (such as, ST3(‘h3)). The default value of this option is **State Name**.
- **State Value Radix:** Specify the state value format to display in a state in the *nState* window by selecting one of the following options:
  - **Original:** The state value format is displayed the same as it is in the *Machine Properties* form. This is the default option.
  - **Binary:** The state value is displayed in Binary format.
  - **Octal:** The state value is displayed in Octal format.
  - **Decimal:** The state value is displayed in Decimal format.
  - **Hexadecimal:** The state value is displayed in Hexadecimal format.

Trace Folder

The **Trace** folder includes **General** and **Active Driver** pages.

General Page

Use this page to specify options for tracing actions in *nTrace*.

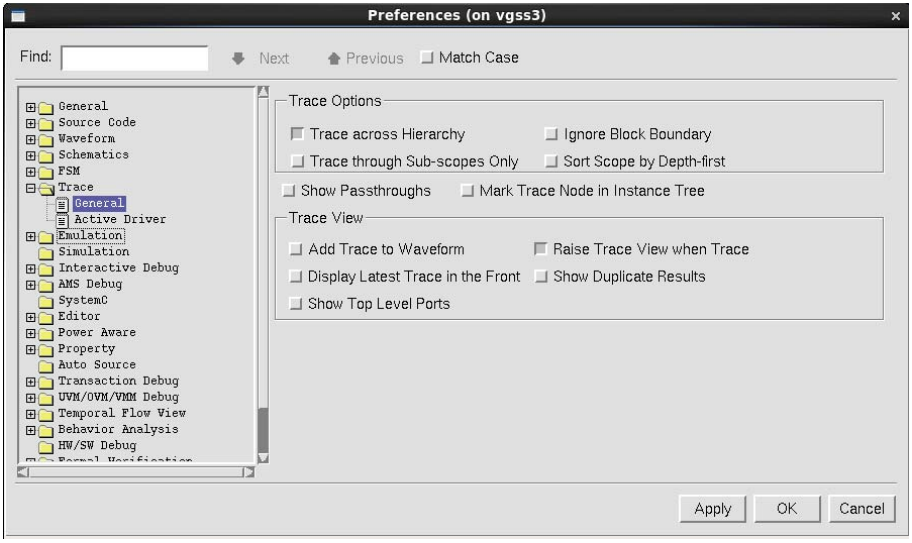


Figure: Trace Page

The following options are available in the **Trace** options section:

- **Trace across Hierarchy:** When this option is turned *on*, tracing a signal through the hierarchy level by level is enabled and **Trace Drive** and **Trace Load** commands behave exactly the same in *nTrace*. When this option is turned *off*, **Trace Drive** and **Trace Load** commands trace a signal level by level. The default value of this option is *on*.
- **Ignore Block Boundary:** When this option is enabled, then the last block scope(s) of the trace result(s) are skipped when sorting the results. By default, this option is *off*.
- **Trace through Sub-scopes Only:** When this option is turned *on*, the tracing scope is limited to the located scope of the selected signal and all its sub-scopes. When this option is turned *off*, all hierarchies are traced. This option applies to trace driver, trace load, and trace connectivity functions. The default value of this option is *off*.

- **Sort Scope by Depth-first:** When this option is turned *off*, the displayed scopes of the trace result are sorted width-first. If not, the scopes are sorted depth-first. The display order impacts the **Show Next/Previous in Hierarchy** command accordingly. By default, this option is *off*.
- **Show Passthroughs:** Displays the passthroughs when tracing drivers and loads.
- **Mark Trace Node in Instance Tree:** When this option is enabled, then the scopes including the trace result are displayed in the orange color in the **Instance** tab. If some scope is set as an active scope once, then *nTrace* updates the tree node to green.

The following options are available in the **Trace View** section:

- **Add Trace to Waveform:** This option adds all the selected signals to the synchronized *nWave* window. You can enable this option in the *Preferences* form or click the  con in the toolbar. By default, this option is *off*.
- **Raise Trace View when Trace:** This option activates the *Message* tab when trace results are added in the trace view. By default, this option is *on*.
- **Display Latest Trace in the Front:** This option adds the latest traced result at the top of tree in the *OneTrace* tab and the oldest traced result is added at the bottom of the *OneTrace* tab.
- **Show Duplicate Results:** This option shows the duplicate trace results, that is, the results that point to the same source line. By default, this option is *off*.
- **Show Top Level Ports:** This option shows/hides the top level port trace results. By default, this option is *off*.
- Click **Apply** or **OK** to apply the settings.

Active Driver Page

Use this page to specify options for tracing active driver settings in *nTrace*.

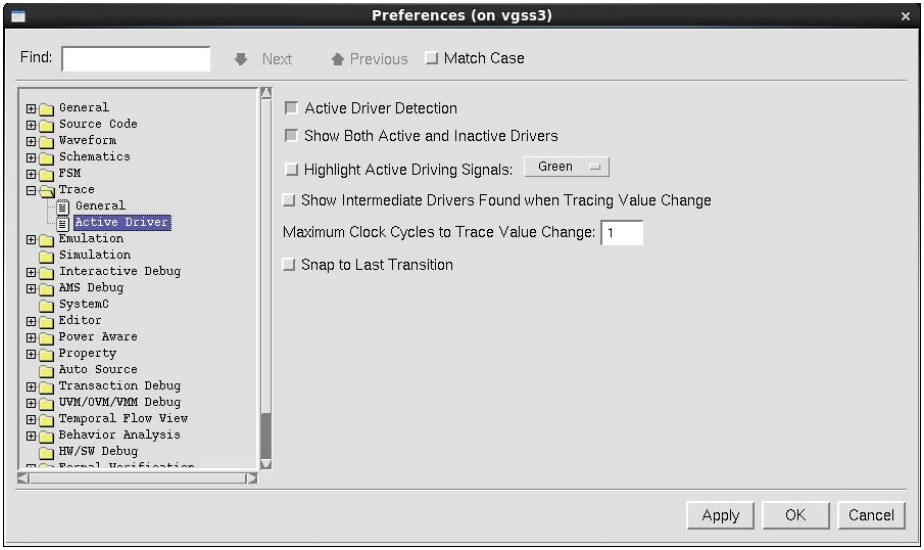


Figure: Active Driver Page

The following options are available in the **Active Driver** section:

- **Active Driver Detection:** This option traces active drivers for the selected signal in the design (multiple drivers in different instances are allowed). The trace results are displayed in the *Source Code* pane and the *OneTrace* tab in the *Message* pane. By default, this option is *on*.
- **Show Both Active and Inactive Drivers:** This option traces and displays all active and inactive drivers in the waveform and the *OneTrace* tab in the *Message* pane. By default, this option is *off*.
- **Highlight Active Driving Signals:** This option highlights all the active drivers in the selected color. The default color is *Green*. You can change the default color of the driving signal by selecting the colors specified. By default, this option is *off*.
- **Show Immediate Drivers Found when Tracing Value Change:** This option displays the last driver encountered when tracing the signal. You can enable this option either in the *Preferences* form or the Show Immediate Drivers right-click menu option. By default, this option is *off*.
- **Maximum Clock Cycles to Trace Value Change:** This option allows you to set the maximum number of clock cycles to trace the value changes. The default value is *0*.
- **Snap to Last Transition:** When this option is turned *on* and if the current cursor time has no transitions, then the cursor snaps to the nearest value

change position. If not, the cursor does not snap to the last transition when the active driver is traced. The default value of this option is *off*.

- Click **Apply** or **OK** to apply the settings.

NOTE: When the following trace options are changed, then the primary *nWave* window updates the trace results accordingly.
- Show Duplicate Results
- Show Both Active and Inactive Drivers
- Show Intermediate Drivers Found when Tracing Value Change

Emulation Folder

The **Emulation** folder includes **General**, **Color** and **Miscellaneous** pages.

General Page

The following option is available:

- **Display Reason Code for Signal Value**: This option enables the reason code display in value annotation. In *nTrace*, *nSchema*, or *Temporal Flow View*, if the **Show Tip** and **Active Annotation** options are turned *on*, a detailed reason code description is showed when the cursor is placed on a signal. The default is *on*.

Simulation Folder

The **Simulation** folder sets the simulation preferences for Interactive Simulation Debug mode invoked by the **Tools -> Run Simulation** command.

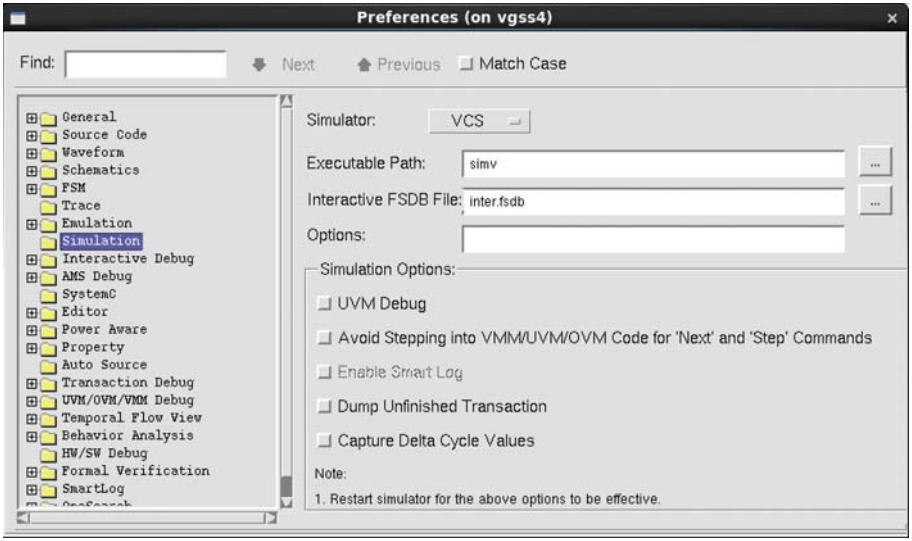


Figure: Simulation Folder

The following options and fields are available:

- **Simulator:** Select the preferred simulator from the list of supported simulators. At this time, only the VCS simulator is supported.
- **Executable Path:** After selecting the simulator, this field is automatically completed with a typical default for the simulator. The default value can be changed to match the executable.

  The VCS simulator can be run interactively through the Verdi platform with the `simv` executable.
- **Interactive FSDB File:** Specify the name of the interactive FSDB file to dump the results to in this field. Specifying `+fsdbfile+file_name` in the **Options** field also works.

  Options: Specify additional simulation options in this field.
- **UVM Debug:** When this option is turned *on*, runtime options are automatically turned on to enable different UVM message types in Interactive Simulation Debug Mode.The default is *on*. The runtime options include:

- `UVM_CONFIG_DB_TRACE`**:** Enables configuration database trace messages.
- `UVM_RESOURCE_DB_TRACE`**:** Enables resource database trace messages.
- `UVM_PHASE_TRACE`**:** Enables Phase trace messages.
- `UVM_OBJECTION_TRACE`**:** Enables Objection trace messages.
- `UVM_FACTORY_TRACE`**:** Enables Factory trace messages.

NOTE: Factory trace messages are instrumented in the Verdi UVM library only.

- `UVM_VERDI_TRACE=UVM_AWARE`**:** Indicates that the enabled trace messages are used for Verdi UVM-aware debug and are not output to the user log file.
- `UVM_LOG_RECORD:` Enables the message catcher to record trace messages into the FSDB file.

NOTE: This option is enabled automatically when the `-uvmDebug` option is specified on the Verdi command line.

- **Avoid Stepping into VMM/UVM/OVM Code for 'Next' and 'Step' Commands:** When this option is turned *on*, the UCLI `config stepintotblib off` command is sent to the simulator and the invoked **Step** and **Next** commands do not step into the VMM/UVM/OVM library code (that is, the testbench code under *$VCS_HOME* or *$UVM_HOME* directory). The default value of this option is *off*.
  When the `config stepintotblib on|off` command is entered in the Verdi console, the simulation behavior is changed and the setting for this option is updated.

NOTE: The option only takes effect after the simulation starts or restarts. If this option setting is changed after the simulation starts, this option does not take effect until the simulation starts again.
If a saved simulation state is restored, the simulation takes the saved setting of the **Avoid Stepping into VMM/UVM/OVM Code for 'Step' and 'Next' Commands** option when the **Save State** command is executed, instead of the current option setting.

- **Enable Smart Log:** This option is turned *on* by default, which means that Verdi turns *on* the Smart Log feature in *Interactive Console* by default.
- **Dump Unfinished Transaction:** When this option is turned *on*, the `+fsdb+event_dump_unfinished` FSDB option is automatically

appended when the Verdi platform starts the `simv` binary. This option is turned *off* by default.

NOTE: Enabling the **Dump Unfinished Transaction** option may impact performance if there are many unfinished transactions. On enabling this option for the first time, a message dialog opens indicating that dumping unfinished transactions may impact performance.

- **Capture Delta Cycle Values**: When this option is turned *on*, Verdi automatically turns *on* delta cycle dumping after invoking interactive debug. This option is turned *off* by default.

NOTE: To apply the above preference options, you must restart the simulator.

Interactive Debug

The **Interactive Debug** folder includes the **Display** page and the **Restart** page.

Display Page

Use this page to set the display preferences.

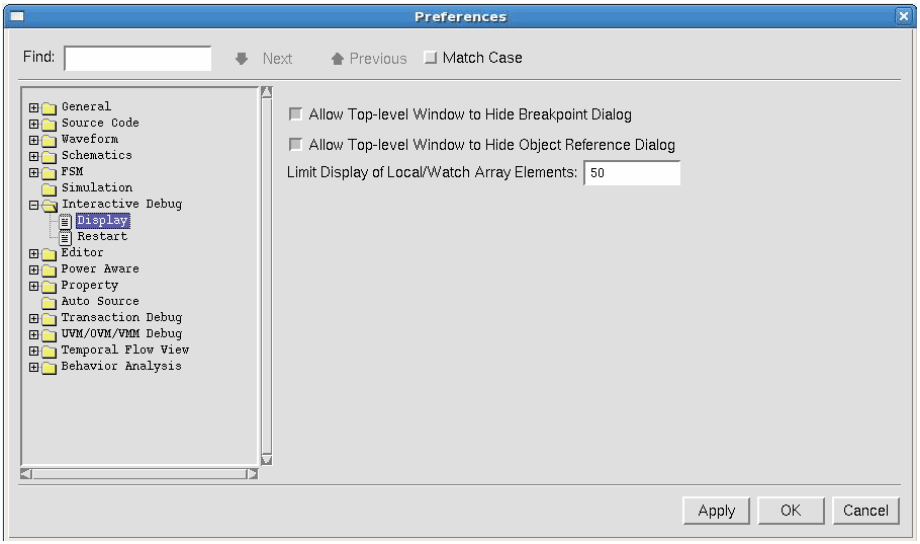


Figure: Display Page

The following options and field are available:

- **Allow Top-level Window to Hide Breakpoint Dialog:** When this option is turned *on*, the top-level window is placed in front of the *Breakpoints* form that is opened by the **Manage Breakpoints** command. When this option is turned *off*, the *Breakpoints* form is placed on top of other windows. The default value of this option is *on*.
- **Allow Top-level Window to Hide Object Reference Dialog:** When this option is turned *on*, the top-level window is placed in front of the *References* form that is opened by the **View Object References** command. When this option is turned *off*, the *References* form is placed on top of other windows. The default value of this option is *on*.
- **Limit Display of Local/Watch Array Elements:** Specify a number (from 1 to 1,000,000) for the maximum number of array elements displayed in **Local** and **Watch** tabs. Searching in the **Show Navigation Text Field** only applies to elements in the display. The default value of this option is *50*.

When array elements are truncated because the array limit is reached, the final array is shown as (Array truncated). To view more array elements, modify the number again. After the specified number is applied, all the expanded arrays are updated immediately.

Restart Page

Use this page to specify options for reissuing commands and breakpoints when the **Simulation -> Restart** command is invoked.

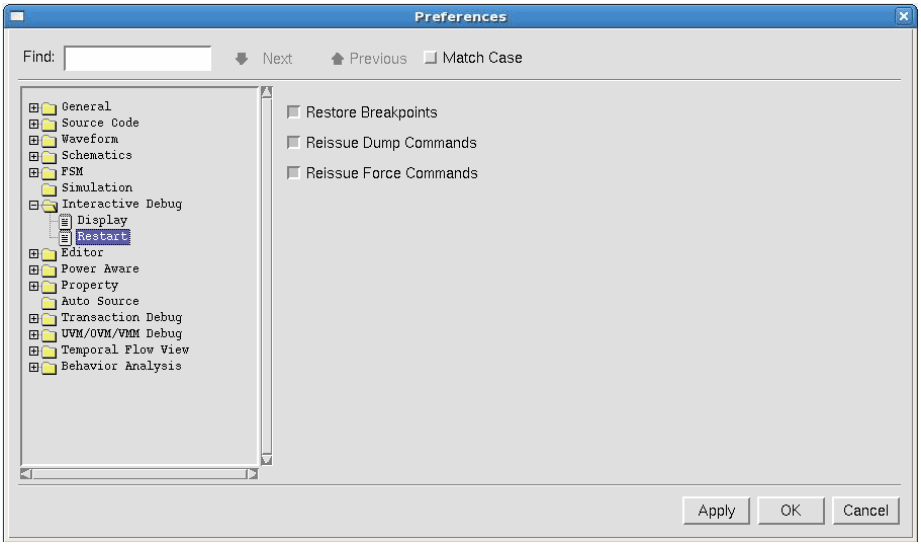


Figure: Restart Page

The following options are available:

- **Restore Breakpoints:** When this option is turned *on*, breakpoint commands are issued again when restarting the simulation. The default value of this option is *on*.
- **Reissue Dump Commands:** When this option is turned *on*, dumping commands are issued again when restarting the simulation. The default value of this option is *on*.

- **Reissue Force Commands:** When this option is turned *on*, force commands are issued again when restarting the simulation. The default value of this option is *on*.

NOTE: The Verdi platform only supports the reissue of force and release commands triggered from the GUI for DUT signals. For force and release commands from the script or from the console, they are not reissued upon restart. If force and release commands are for testbench signals, they are not reissued.

Reverse Debug Page

To use the **Reverse Debug** option, the options in the *Reverse Debug* page must be operated in the *Interactive* mode and the *Keep Future* mode must be *on*.

If the driver is from testbench scope, double-click the hardware boundary signal in the *Source Code* pane to trace the driver signal in the testbench scope using the **Reverse Debug** option.

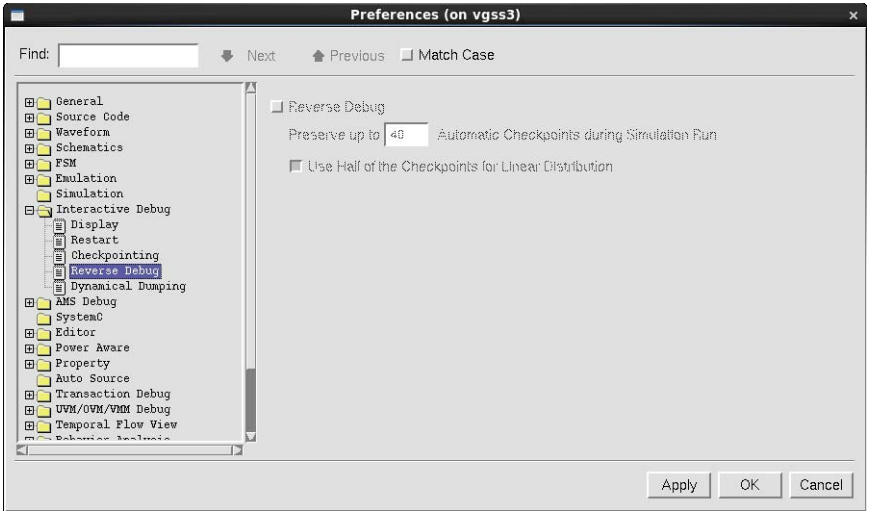


Figure: Reverse Debug Page

1. Use any of the following options that are available in the **Reverse Debug** section:
   - **Preserve up to 40 Automatic Checkpoints during Simulation Run:** This option allows you to set the checkpoints for the simulation stops. The default value is *40*. This option can undo five most recent simulation controls.

- **Use Half of the Checkpoints for Linear Distribution:** This option allows you to make reverse execution faster when you need to go to the execution time far away from the current time. If you need to go to the recent simulation execution time, you can *disable* this option to reduce memory consumption. This option is enabled by default.

NOTE: To enable the **Reverse Debug** option from the GUI, the **+reverse** option must be added during the VCS compilation (for example, `vcs -sverilog t1.v -full64 -lca -debug_access+all+reverse` ...).

---

2. Click **Apply** or **OK** to apply the settings.

AMS Debug Folder

The **AMS Debug** folder includes the **Schematic Color/View** page and the **Miscellaneous** page.

Schematic Color/View Page

Use this page to specify the color settings and viewing options for AMS objects.

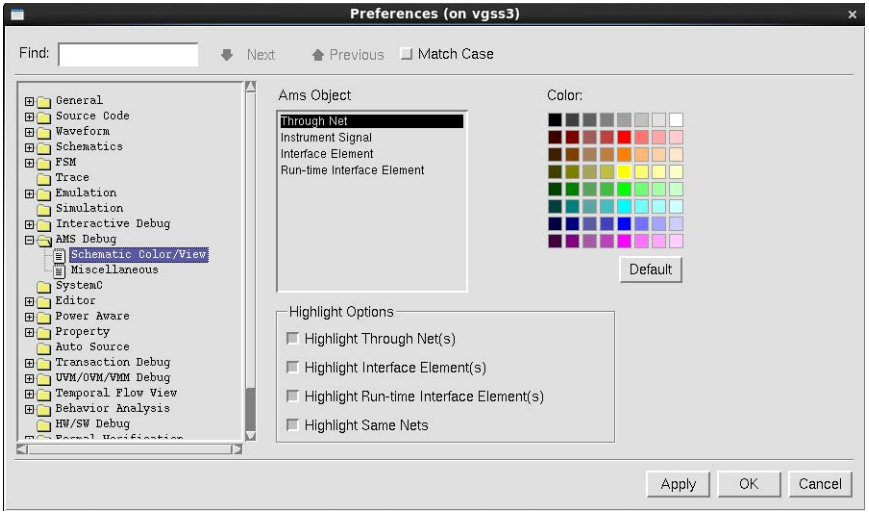


Figure: Schematic Color/View Page

The colors for the following objects in the **AMS Objects** section can be changed (click **Default** to reset these options back to their default settings):

- **Through Net:** The color of through-nets. The default color is ID_YELLOW5.
- **Instrumental Signal:** The color of instrumental signals. The default color is ID_GREEN5.
- **Interface Element:** The color of interface element. The default color is ID_GREEN5.
- **Run-time Interface Element:** The color of run-time interface element. The default color is ID_BLUE5.

The following options are available in the **Highlight Options** section:

- **Highlight Through Net(s):** When this option is turned *on*, the signal which is through net type is colored as ID_YELLOW5. The default value of this option is *on*.
- **Highlight Interface Element(s):** When this option is turned *on*, the signal which is interface element type is colored as ID_GREEN5. The default value of this option is *on*.
- **Highlight Run-time Interface Elements:** When this option is turned *on*, the signal which is run_time interface element type is colored as ID_BLUE5. The default value of this option is *on*.
- **Highlight Same Nets:** When this option is turned *on*, the same nets of through net(s), interface element(s) and run-time interface element(s) are colored the same color as through net(s), interface element(s) and run-time interface element(s). The default value of this option is *on*.

Miscellaneous Page

Use this page to set the miscellaneous preferences.

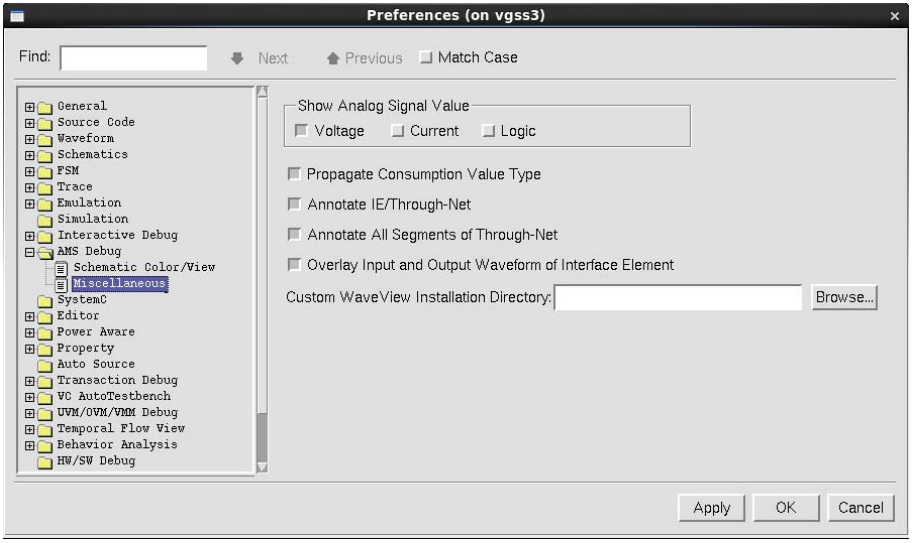


Figure: Miscellaneous Page

The following miscellaneous options are available:

- **Show Analog Signal Value:** You can select **Voltage**, **Current**, or **Logic** to enable the value type in the source code pane. You can select multiple value types.

- **Propagate Consumption Value Type:** When this option is turned *on*, the selected value type is propagated to all its analog scope same net and `a2a` through nets in the digital scope. If not, the value type is not propagated. By default, this option is *on*.
- **Annotate IEs/Thorough Net:** When this option is turned *on*, IEs and through-net type can be annotated in the source code pane. The option is turned *on* as default. Note that when the option is turn *off*, **Annotate All Segments of Through-Net** option is disabled.
- **Annotate All Segments of Through-Net:** When this option is turned *on*, all the same nets of the signals which are available in the through net report file are annotated as *a2a*, *d2d*, or, *n2n* in the source code pane.
- **Overlay Input and Output Waveform of Interface Element:** When this option is turned *on*, the input and output waveforms are automatically over-layed in the following scenarios:
  - The interface elements from the *nTrace* designer browser pane and *nSchema* are selected and added in the waveform.
  - The digital side signal is either single-bit digital signal or analog signal.

  By default, this option is *on.*

  Following is an example of the layout when multiple interface elements from nTrace designer browser pane are dragged and dropped into the *nWave*.

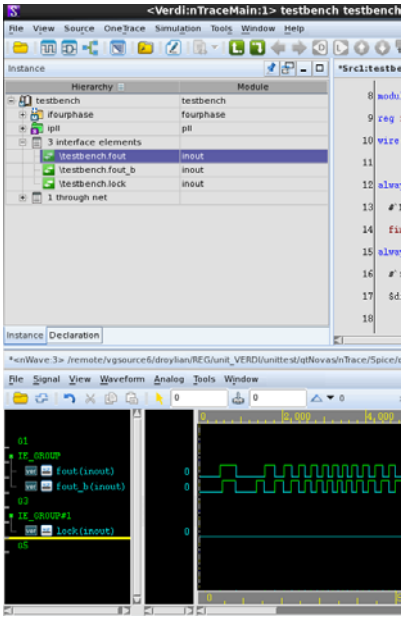


Figure: Signal Layout After Dragging and Dropping the Interface Elements

Spice Debug Folder

The **Spice Debug** folder specifies settings for spice signals.

Miscellaneous Page

Use this page to specify miscellaneous settings.

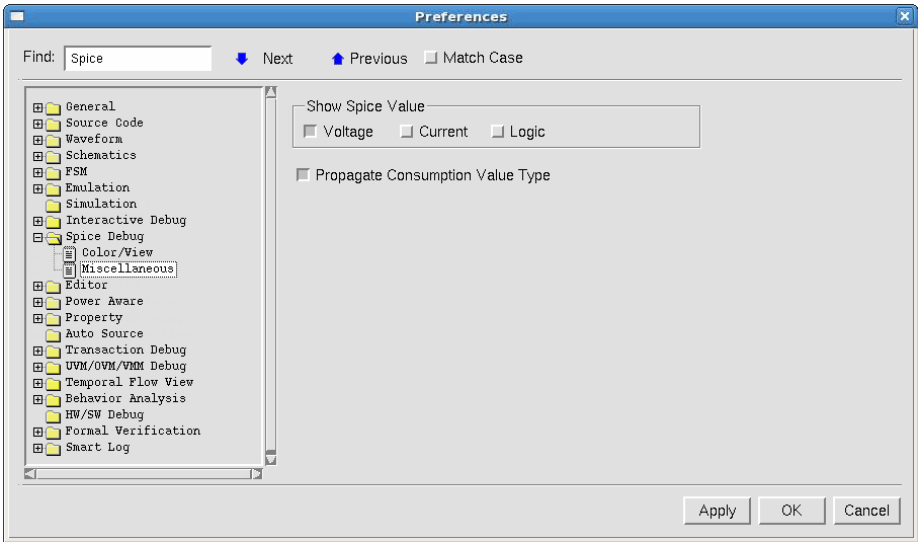


Figure: Spice Debug Folder

The **Miscellaneous** page includes the following options:

- **Show Spice Value:** Select an option from **Voltage**, **Current** and **Logic** to display the annotation type. Multiple selection is supported. The default value of this option is **Voltage**.
- **Propagate Consumption Value Type:** When this option is turned *on*, the value type of the selected signal propagates to all the same nets connected to the selected analog signal and to a2a through nets in digital scopes. When this option is turned *off*, only the value type of the selected signal changes. The default value of this option is *on*.

Editor Folder

The **Editor** folder includes **nEditor**, **VI**, **Emacs**, **Text Editor**, and **Other** pages. Use these pages to specify the editor to be used when invoking the **Source -> Edit Source File** command or clicking the **Edit File** button on the toolbar.

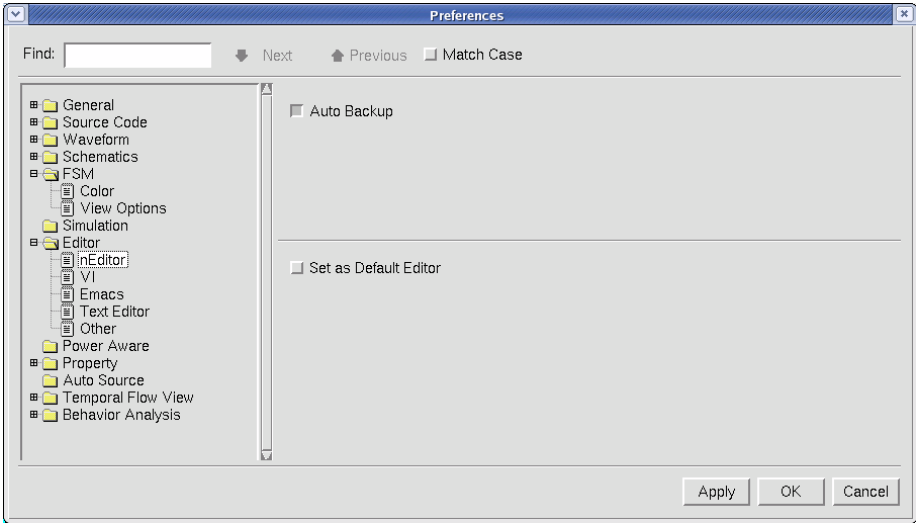


Figure: Editor Page

The Verdi platform provides support for several mainstream editors (**VI**, **Emacs**, **Text Editor**) and its own editor, **nEditor,** (refer to the *nEditor* chapter for details). Each editor is located on a separate page.

To specify the preferred editor, select the appropriate editor page and turn the **Set as Default Editor** option *on* for that editor. Thus, it becomes the default editor.

In the **nEditor** page, use the **Auto Backup** option to automatically create a backup version of the file being edited. The default value of this option is *on*.

Font, **Foreground Color**, and **Background Color** options can also be changed for mainstream editors.

In addition, a different editor and associated options can be specified using the **Other** page.

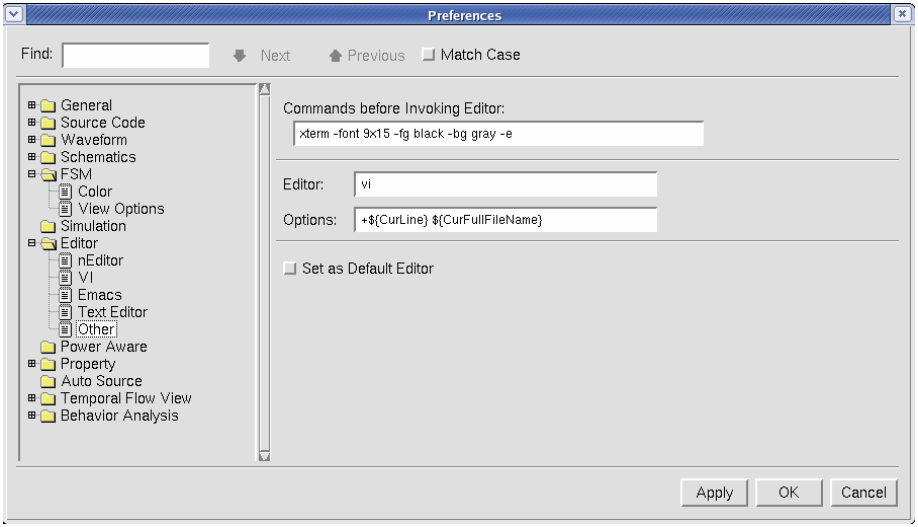


Figure: Other Page

Specify the text font and the color in the text edit file (to be invoked) in the **Commands before Invoking Editor** text field. Then, specify the editor to be used in the **Editor** text field and the parameters to open the specified text editor in the **Options** text field. The **Options** are set as `+${CurLine} ${CurFullFileName}` by default.

To change the specified editor as the default editor, turn the **Set as Default Editor** option *on*.

NOTE: If `gvim` is specified as the default editor on the **Other** page, `-f` must be added in the **Options** text field.

Power Aware Folder

The **Power Aware** folder includes **Color/View** and **Miscellaneous** pages and the **PowerMap** folder.

Color/View Page

Use this page to specify the color settings and viewing options for power aware objects.

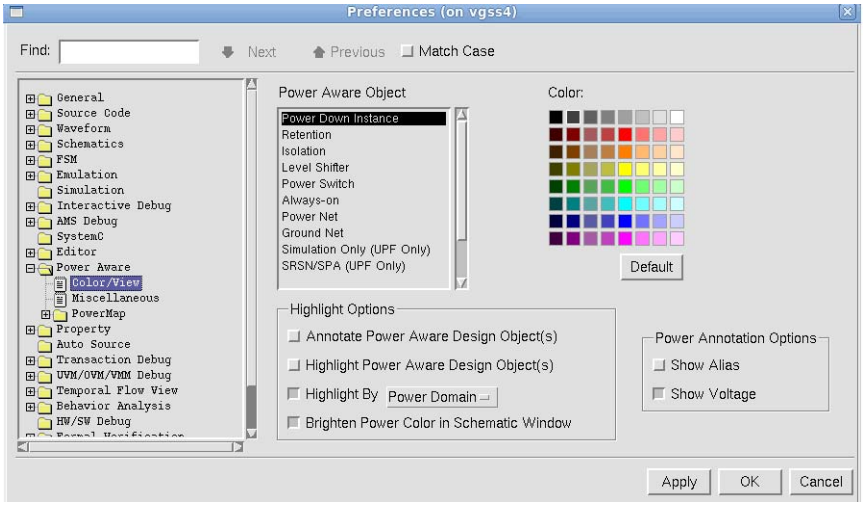


Figure: Color/View Page

The colors for the following objects in the **Power Aware Objects** section can be changed (click the **Default** button to reset these options back to their default settings):

- **Power Down Instance:** The color of powered down instances. The default is ID_GRAY1.
- **Retention:** The color of Retention objects. The default is ID_YELLOW7.
- **Isolation:** The color of Isolation objects. The default is ID_RED7.
- **Level Shifter:** The color of Level-shifted objects. The default is ID_GREEN6.
- **Power Switch:** The color of power switch objects. The default is ID_ORANGE5.
- **Always-on:** The color of always-on objects. The default is ID_GREEN5.

- **Power Net:** The color of power nets. The default is ID_RED2.
- **Ground Net:** The color of ground nets. The default is ID_RED2.
- **Simulation Only (UPF Only):** The color of simulation only objects. The default is ID_CYAN8.
- **SRSN/SPA (UPF Only):** The color of SRSN/SPA signals. The default is ID_CYAN8.
- **CSN (UPF Only):** The color of CSN signals. The default is ID_CYAN8.
- **Acknowledge:** The color of ack signals. The default is ID_CYAN8.
- **Boundary Port (CPF Only):** The color of boundary port objects. The default is ID_CYAN8.

The following options are available in the **Highlight Options** section:

- **Annotate Power Aware Design Object(s):** When this option is turned *on*, the power aware design objects are annotated in the source code pane. The default value of this option is *on*.
- **Highlight Power Aware Design Object(s):** When this option is turned *on*, the power aware design objects are highlighted with the specified colors. The default value of this option is *on*.
- **Highlight By:** When this option is turned on, either the **Power Domains** details or the **Root Supply** option are highlighted with the specified colors. The default value of this option is *Power Domain*. When the **Root Supply** option is selected, the net's color is the same as the connected *instport* in the *nSchema* window.
- **Brighten Power Color in Schematic Window:** When this option is turned on, the power domain colors are brightened. The default value of this option is off.

The following options are available in the **Power Annotation Options** section:

- **Show Alias:** When this option is turned *on*, the nominal condition of the port state name for CPF, or power domains, supply net/port/set signal for UPF are displayed with an annotation in *nTrace, nSchema, nWave, Temporal Flow View,* or *Power Manager*. In addition, after a power domain signal or supply net/port/set signal is added to the *nWave* window, turn *on* this option to show the alias signal indicated with the `#alias` suffix. The default value of this option is *off*.

NOTE: The option setting in the `novas.rc` resource file has higher priority than the setting in the **Preferences** option.

- **Show Voltage:** When this option is turned *on*, the voltage for power domains or supply net/port/set signal is displayed with an annotation in

nTrace, *nSchema*, *nWave*, *Temporal Flow View*, or *Power Manager*. In addition, after a power domain signal or supply net/port/set signal is added to the *nWave* window, turn *on* this option to show the voltage signal indicated with the `#voltage` suffix. The default value of this option is *off*.

NOTE: The option setting in the `novas.rc` resource file has higher priority than the setting in the **Preferences** option.

Miscellaneous Page

Use this page to specify miscellaneous settings for Power Aware Debug.

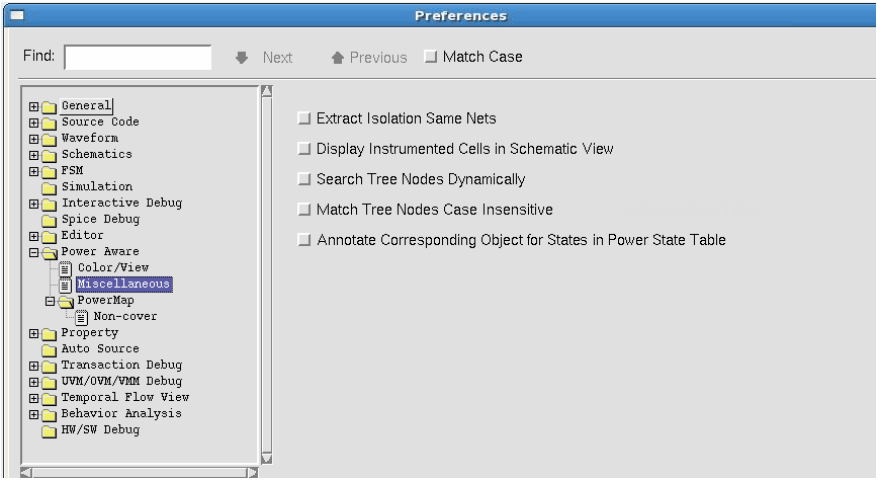


Figure: Miscellaneous Page

The following miscellaneous options are available:

- **Extract Isolation Same Nets:** When this option is turned *on*, the same nets that are derived from an Isolation signal are marked as ISO in *nTrace, nWave, Power Manager, Temporal Flow View,* or *nSchema*. For example, if signals `a`, `b`, `c`, `d`, and `e` are the same nets, `c` is an Isolation signal for the rule `c_iso` and `d` is a Level-shifted signal, `d` and `e` are treated as Isolation signals, but `a` and `b` are not treated as Isolation signals. When this option is turned *off*, the same nets that are derived from an Isolation signal are not marked. The default value of this option is *off*.
- **Display Instrumented Cell in Schematic View:** When this option is turned *on*, Isolation cells are displayed in the flattened schematic windows. The default value of this option is *off*.

NOTE: This option is available when a UPF file has already been loaded.

- **Search Tree Nodes Dynamically:** When this option is turned *on*, tree nodes in the *Power Domain Tree* pane are dynamically searched as the search string is entered. When this option is turned *off*, tree nodes for the specified string in the *Power Domain Tree* frame are searched after pressing the **Enter** key. The default value of this option is *off*.
- **Annotate Corresponding Object for States in Power State Table:** When this option is turned *on*, annotation for states in the `add_pst_state` command is displayed. When this option is turned *off*, annotation for states

in the `add_pst_state` command is not displayed. The default value of this option is *off*.

NOTE: The annotation display is only supported for UPF designs.

- **Display Isolation Annotation Symbol:** When this option is turned *on*, isolation annotation is displayed as symbol. When this option is turned *off*, isolation annotation is displayed as string. The default value of this option is *on*.

PowerMap Folder

The **PowerMap** folder includes the **Cover/Non-cover** page.

Cover/Non-cover Page

Use this page to specify covered/non-covered signal checking options in the *Power Map* window. The *Power Map* window is reset after any of these options are changed.

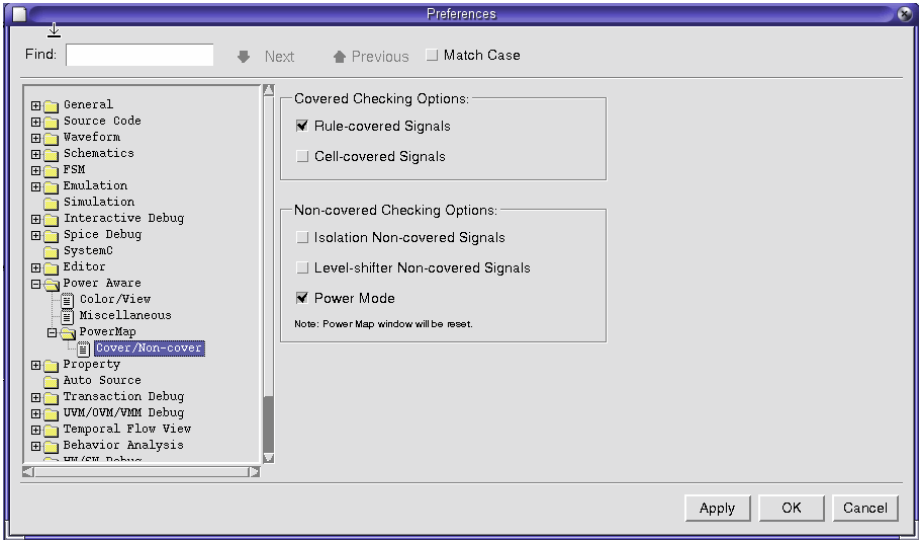


Figure: Cover/Non-cover Page

The following options are available in the **Covered Checking Options** section:

- **Rule-covered Signals:** When this option is turned *on*, isolation and level-shifter rule signals are checked. The default value of this option is *off*.
- **Cell-covered Signals:** When this option is turned *on*, signals connected to the isolation or the level-shifter cell signals are reported. The default value of this option is *off*.

The following options are available in the **Non-covered Checking Options** section:

- **Isolation Non-covered Signals:** When this option is turned *on*, the *Power Map* window performs Isolation non-covered checking. The default value of this option is *off*.

- **Level-shifter Non-covered Signals:** When this option is turned *on*, the *Power Map* window performs Level-shifted non-covered checking. The default value of this option is *off*.
- **Power Mode:** When this option is turned *on*, the domain status and domain voltage defined in the Power mode is checked during Isolation or Level-shifted non-covered checking. The default value of this option is *on*.

Property Folder

The **Property** folder includes the **Property Statistics** page, the **Property Details** folder (with the **Property Details** and **Other** pages), and the **Analyzer** page.

Property Statistics Page

Use this page to specify the filters and the sorting fields for the **Property Statistics** tab of the *Statistics* pane.

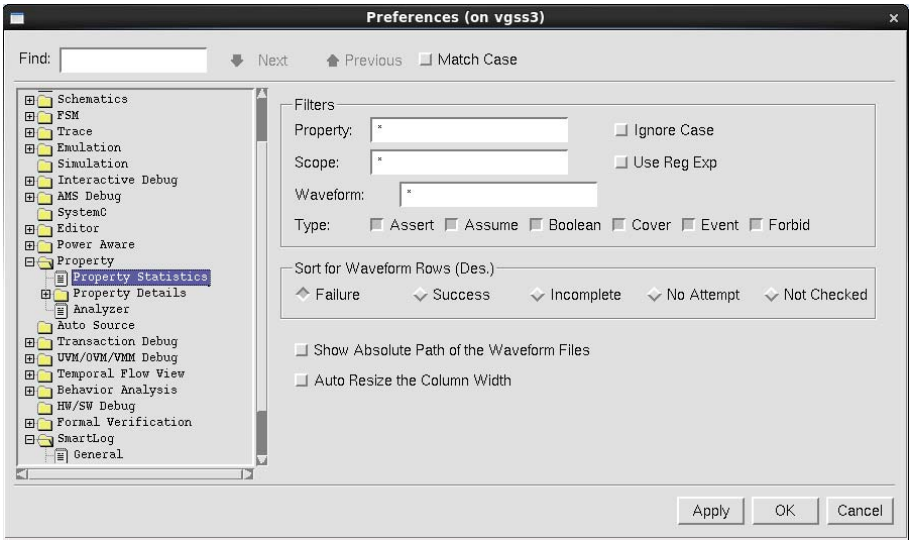


Figure: Property Statistics Page

The fields and the options in the **Filters** section specify the filters for listing the properties. Regular expressions using the wildcard (*) character are accepted in all fields.

- **Property:** Filters the property statistics list by the property name.
- **Scope:** Filters the property statistics list based on the scope. The full hierarchical path is searchable.
- **Waveform:** Filters the property statistics list based on the waveform file name.
- **Ignore Case:** When this option is turned *on*, the case is ignored (such as, ‘a’ finds ‘a’ and ‘A’) in all the filter fields. When this option is turned *off*, case is considered (such as, ‘b’ finds ‘b’ only) in all the filter fields. The default value of this option is *off*.

- **Use Regular Expression:** When this option is turned *on*, additional regular expressions are supported. Refer to the following table for details. The default value of this option is *off*.

Syntax	Expression	Description
? (question mark)	Any character	Matches any one character.
* (asterisk)	Zero or more	Finds zero or more occurrences from the preceding expression.
| (pipe)	OR	Matches the expression before or after the pipe sign "|". Mostly used within a group. For example, "[sponge]|[mud] bath" matches "sponge bath" and "mud bath".
+ (plus)	One or more	Matches at least one occurrence from the preceding expression. For example, "ba+" matches "ba", "baa", and "baaa".
~ (tilde)	Prevent match	Prevents a match on 'X' at the point it appears in the expression. For example, "rea~[ity]" matches the "real" in "really", but not the "real" in "reality".

- **Type:** Filters the property statistics list by the property type. Select the desired properties to be filtered from the following: **Assert, Assume, Boolean, Cover, Event,** and **Forbid**. Multiple selections are supported.

In the **Sort for FSDB Rows (Des.)** section, select one of the following options: **Failure**, **Success**, **Incomplete**, **No Attempt**, or **Not Checked** as the method to sort the results in descending order in the **Property Statistics** table. Only one method can be selected at a time.

Property Details Folder

The **Property Details** folder includes **View** and **Miscellaneous** pages.

View Page

Use this page to specify the filters and the sorting fields for the **Property Details** table of the *Statistics* frame.

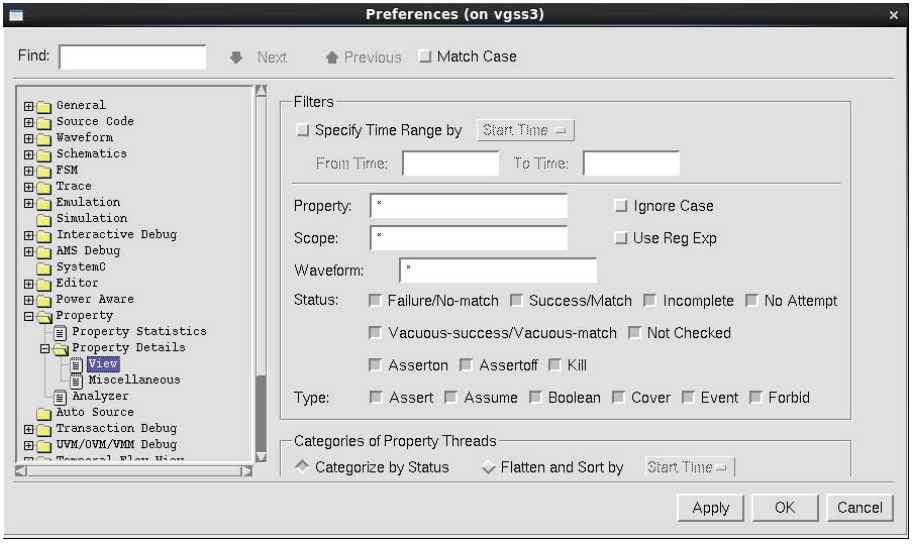


Figure: Property Details Folder - View Page

Filters

- **Specify Time Range by:** When this option is turned *on*, property attempts whose start or end time is within the specified time range are filtered. The default value of this option is *off*.
- **From Time/To Time:** Specify the time range of interest during which the property attempts need to be filtered. These text fields are available if the option above is enabled first.

The following fields and options are available to specify the filters for listing the properties. Regular expressions using the wildcard (*) character are accepted in all fields.

- **Property:** Filters the property details list by the property name.
- **Scope:** Filters the property details list based on the scope. The full hierarchical path is searchable.

- **Waveform:** Filters the property details list based on the waveform file name.
- **Ignore Case:** When this option is turned *on*, the case is ignored (such as, ‘a’ finds ‘a’ and ‘A’) in all the filter fields. When this option is turned *off*, case is considered (such as, ‘b’ finds ‘b’ only) in all the filter fields. The default value of this option is *off*.
- **Use RegExp:** When this option is turned *on*, additional regular expressions are supported (refer to the previous Options description in this section). The default value of this option is *off*.
- **Status:** Filter the property details list by any of the following options. Multiple selections are supported.
  - **Failure/No-Match:** Property attempts with failure results.
  - **Success/Match:** Property attempts with success results.
  - **Incomplete:** Property evaluations that were not completed before the end of the simulation.
  - **No Attempt:** Property that was not activated in any of the simulation runs of the FSDB files.
  - **Vacuous-Success/Vacuous-Match:** Threads with the value string ‘vacuous-success’ or ‘vacuous-match’.
  - **Not Checked:** Property that was not selected for evaluation in the Evaluator tab.

NOTE: Alternatively, when clicking on any row in the *Property Details* table, use **Ctrl+S** to turn the **Success/Match** filter option *on* or *off*, use **Ctrl+A** to select all, and use **Ctrl+D** to disable all **Status** filter options.

- **Type:** Filter the property details list by the property type. Select the desired properties to be filtered from the following: **Assert, Assume, Boolean, Cover, Event,** and **Forbid**. Multiple selections are supported.

The following options are available in the **Categories of Property Threads** section:

- **Categorize by Status:** When this option is selected, the results in the *Property Details* table are sorted by status. This option is set as the default.
- **Flatten and Sort by:** When this option is selected, **Start Time** or **End Time** option can be specified to sort the results in the *Property Details* table.

Miscellaneous Page

Use this page to specify miscellaneous Assertion Debug options.

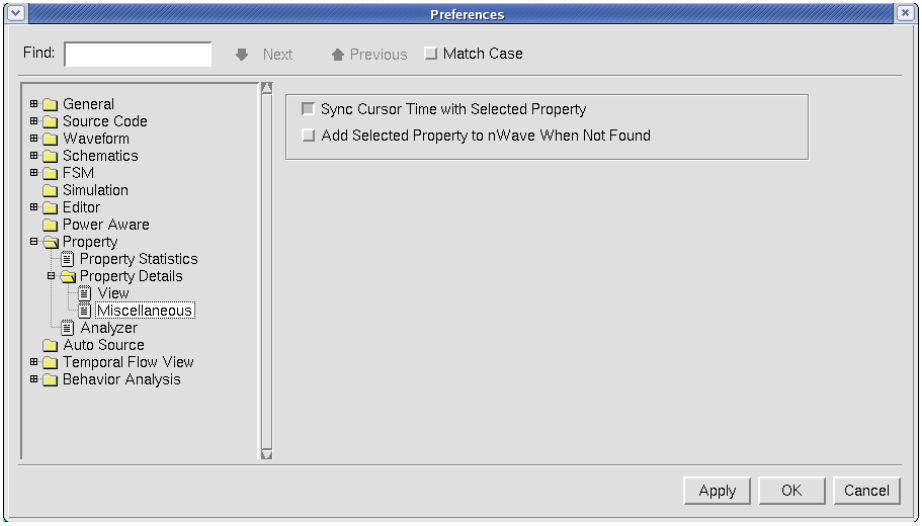


Figure: Property Details Folder - Miscellaneous Page

The following options are available:

- **Sync Cursor Time with Selected Property:** When this option is turned *on*, selecting a row in the *Property Details* table synchronizes the cursor time in *nWave* accordingly. The default value of this option is *on*.
- **Add Selected Property to nWave When Not Found:** When this option is turned *on*, selected property signals are added to the *nWave* window when an identical signal does not already exist in the signal pane. The default value of this option is *off*.

Analyzer Page

Use this page to specify the Assertion Debug Analyzer options.

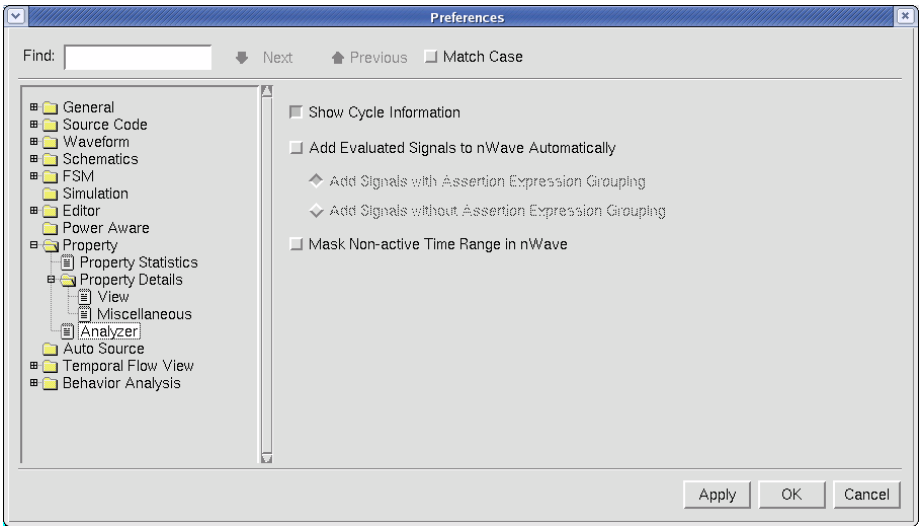


Figure: Analyzer Page

The following options are available:

- **Show Cycle Information:** When this option is turned *on*, cycle or timing information within the delay is shown on the *Analyzer* pane. The default value of this option is *on*.

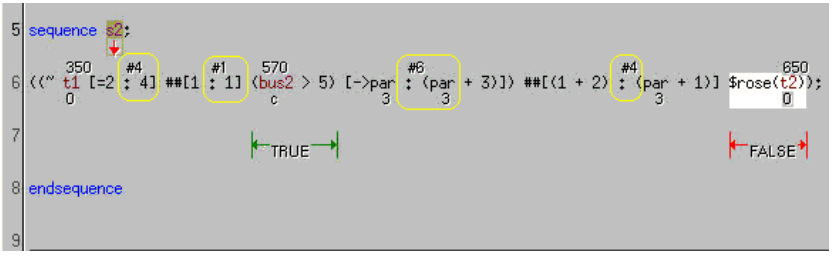


Figure: Show Cycle Information

- **Add Evaluated Signals to nWave Automatically:** When this option is turned *on*, the expression signals being evaluated are added and displayed in *nWave* automatically. By default the **Add Signals with Assertion Expression Grouping** option is also turned *on* and the expression signals are grouped in *nWave*. The expression signals can be displayed as un-grouped by selecting the **Add Signals without Assertion Expression Grouping** option. When the **Add Evaluated Signals to nWave**

Automatically option is turned *off*, the expression signals are not added to the *nWave* frame. The default value of this option is *off*.

- **Mask Non-active Time Range in nWave:** When this option is turned *on*, masks are drawn in the *nWave* waveform pane outside the start and end time of the assertion currently under analysis. The default value of this option is *off*.

NOTE: Use the **View -> Clear Assertion Analysis Mask** command in *nWave* to remove the mask.

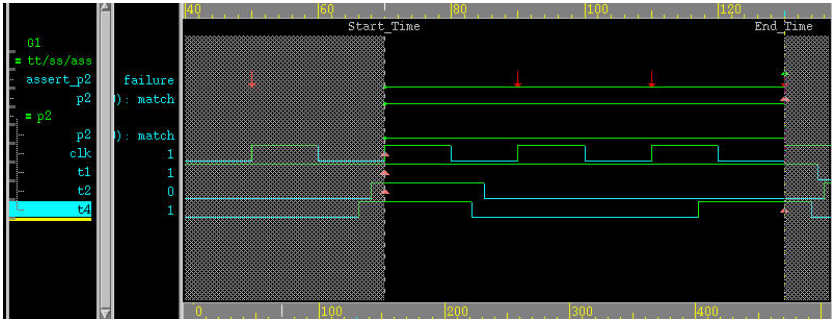


Figure: Mask Non-active Time Range in nWave

Auto Source Folder

Specify the Tcl command file paths or sources to be played when the Verdi platform is initiated. It can be set to the path of the directory or file. If the path of a directory is given, all files under the directory are sourced. Multiple paths can be set and separated by a space with double quotes used to enclose the paths.

Click the **Browse** button to specify the Tcl command file paths or sources to be executed when the Verdi platform is initiated. Click the **Delete** button to delete the selected source path from the **Source Path** list.

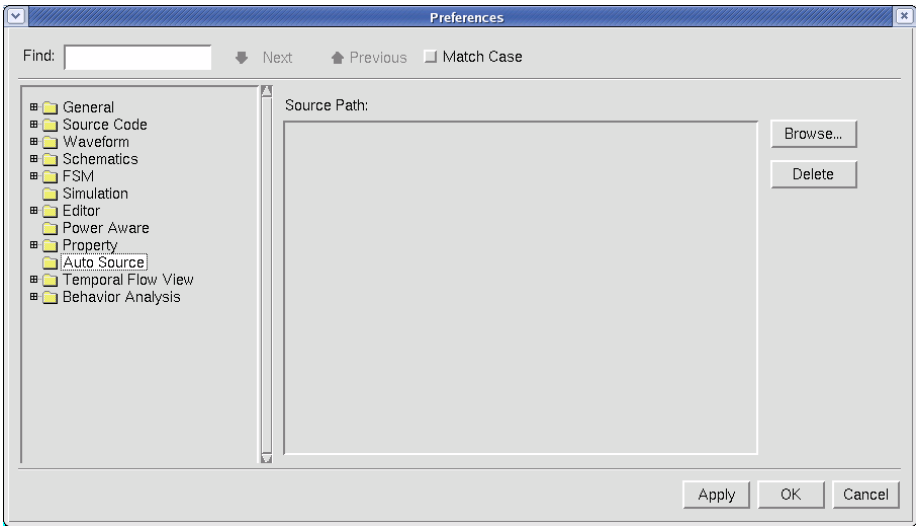


Figure: Auto Source Folder

UVM/OVM/VMM Debug

These preferences are for the *Resource View, Factory View, Phase View, Sequence View, Register View* windows.

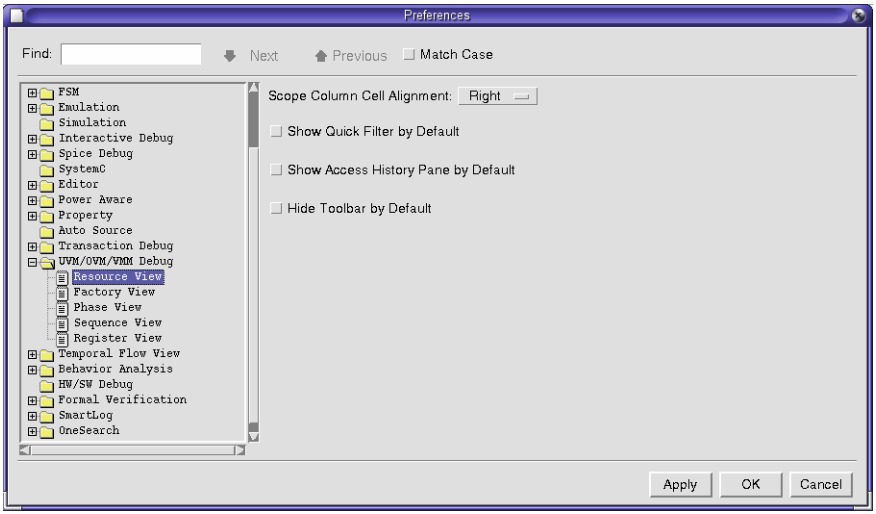


Figure: UVM/OVM/VMM Debug Folder

Resource View Page

The following options are available:

- **Scope Column Cell Alignment:** This option sets the scope column cell alignment rule in the *Resource Access History* pane.

  The available options are: **Right**, **Center**, and **Left**.

- **Show Quick Filter By Default:** This option turns on/off the *Quick Filter* pane automatically when opening the UVM Resource View pane.
- **Show Access History Pane by Default:** This option turns on/off the *Resource Access History* pane automatically when opening the UVM *Resource View* pane.
- **Hide Toolbar by Default:** This option hides the toolbar by default when opening the UVM *Resource View* panes.

Factory View Page

The following options are available:

- **Override Instance Column Cell Alignment:** This option sets the *Override Instance* column cell alignment rule in two panes of the *Factory View* pane.

  The available options are: **Right**, **Center**, and **Left**.
- **Show Quick Filter By Default:** This option turns on/off the *Quick Filter* pane automatically when opening the UVM *Factory View* pane.
- **Show Override Instance Pane by Default:** This option turns on the *Override Instance* pane automatically when opening the UVM *Factory View* pane.
- **Hide Toolbar by Default:** This option hides the toolbar by default when opening the UVM *Factory View* panes.

Phase View Page

The following options are available:

- **Object Column Cell Alignment:** This option sets the scope column cell alignment rule in the *Objection History* pane of the *Phase View*.

  The available options are: **Left**, **Right**, and **Center**.
- **Hide Toolbar by Default:** This option hides the toolbar by default when opening the *UVM Phase View* panes.

Sequence View Page

The following options are available:

- **Sequence Tree Depth:** This option displays the number of generation of descendants in the *Sequence View* pane by default.
- **Quick Filter Time Preference:** There are begin time and end time fields in the *Quick Filter* pane. This option sets this preference for filtering sequence data from begin time to end time range or sequence data exactly matching the begin time and the end time.

  The available options are: **Fixed String** and **Range**.
- **Sequencer Column Cell Alignment:** This option sets the *Sequencer* column cell alignment rule in the *Sequence View* pane.

  The available options are: **Right**, **Center**, and **Left**.

- **Show Quick Filter By Default:** This option turns on/off the *Quick Filter* pane automatically when opening the *UVM Sequence View* pane.
- **Hide Toolbar by Default:** This option hides the toolbar by default when opening the *UVM Sequence View* panes.

Register View Page

The following options are available:

- **Attribute Tree Depth:** This option displays the number of generation of descendants in the *Register Attribute* pane of the *Register View* pane by default.
- **Hide Toolbar by Default:** This option hides the toolbar by default when opening the UVM *Register View* panes.
- **Show Registers and Fields in the Hierarchy Tree:** This option displays the fields of register by default in the *Hierarchy Tree* pane of UVM *Register View* pane.
- **Show Register Declaration Name:** This option displays the declaration name of register by default in the *Hierarchy Tree* pane of UVM *Register View* pane.

Temporal Flow View Folder

These preferences are for the *Temporal Flow View*, *Compact Temporal Flow View,* and *Temporal Register View* windows.

Temporal Flow View Page

Click the **Restore to Default Settings** option in this page to reset the settings to the default settings.

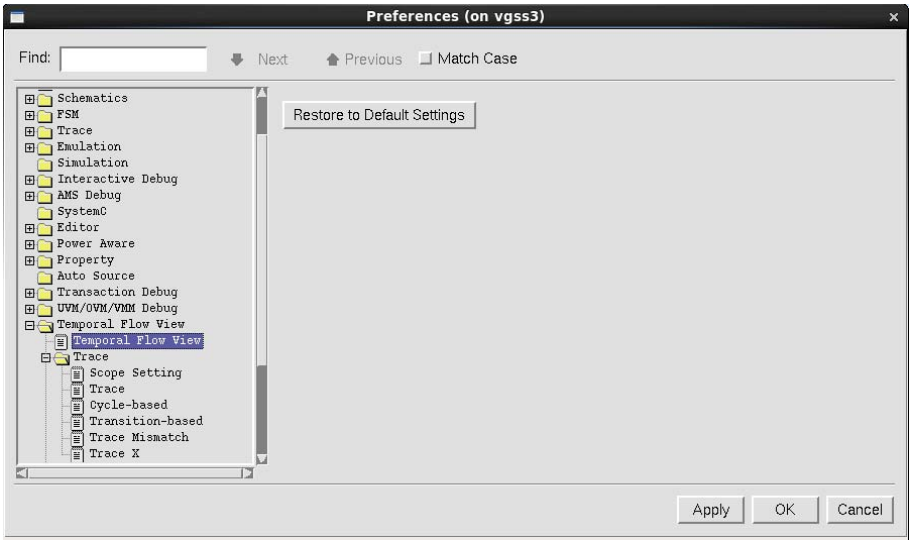


Figure: Temporal Flow View Page

Trace Folder

The **Trace** folder includes the **Scope Setting**, **Trace**, **Cycled Based**, **Transition Based**, and **Trace Mismatch, Trace X** pages.

Scope Setting

Use this page to specify scope setting for trace commands.

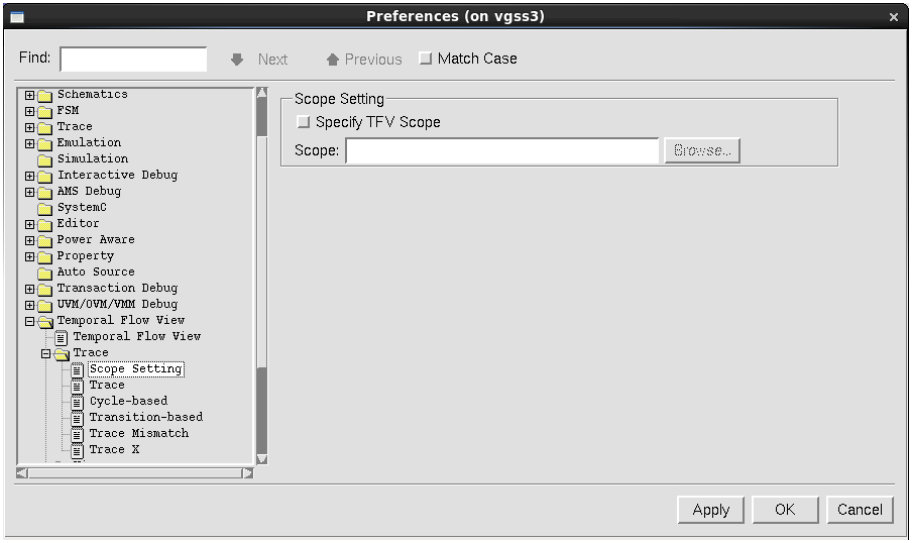


Figure: Scope Setting Page

The following options are available in the **Scope Setting** section:

- **Specify TFV Scope:** When this option is *off*, the scope specified for Behavior Analysis is also used for the *Flow View* windows. When this option is *on*, the **Scope** text field is enabled and a different scope can be specified. The default value of this option is *off*.
- **Scope**: Specify the top level scope to use for the *Flow View* windows. Enter the desired scope in this text field or click the **Browse** button to open the *Instance Manager* form.

  In the *Instance Manager* form, the design hierarchy can be browsed and a scope selected. A white highlighted rectangle indicates the selected scope and the full path is entered on the top text field. Click **OK** to save the scope and close the window when the selection is complete or click **Cancel** to exit the window without saving the scope.

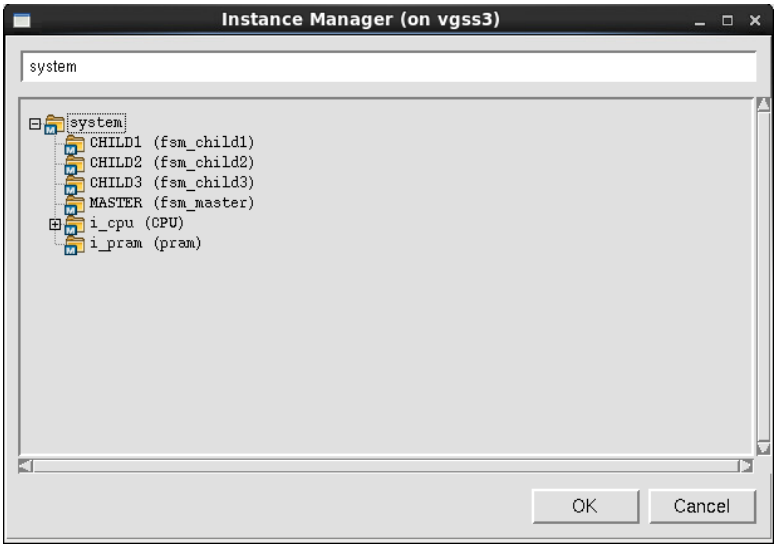


Figure: Instance Manager Form

Trace Page

Use this page to specify general trace options for trace commands.

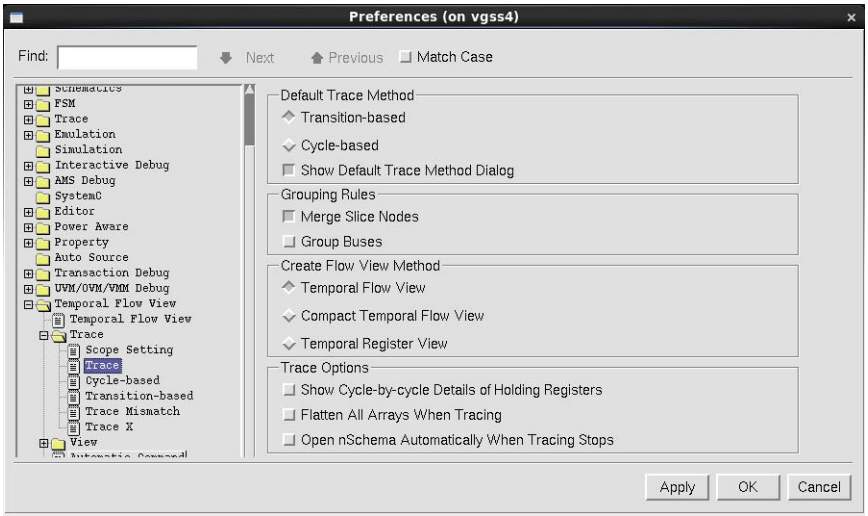


Figure: Trace Page

The following options are available in the **Default Trace Method** section (only one option may be selected at a time):

- **Transition-based:** When this option is selected, the *Temporal Flow View* uses the transition-based engine so input signals that trigger the output transition are automatically traced. This is used to visualize the propagation of signal transitions throughout the design and over time which is very useful for gate level debug. This option cannot be used with a *Temporal Register View* window.
- **Cycle-based:** When this option is selected, the *Temporal Flow View* uses the cycle-based engine. This means design behavior is unrolled based on clock cycles so signals in the *Temporal Flow View* are displayed and traced according to the final stable value within a single clock cycle.
- **Show Default Trace Method Dialog**: When this option is selected, the *Trace Method Setting* form appears again even if the **Do not ask me again** option is selected in the *Trace Method Setting* form.

The following options are available in the **Grouping Rules** section:

- **Merge Slice Nodes:** When this option is turned *on*, sliced signals (such as, `A_reg[0], A_reg[1], ..., A_reg[7]`) in a gate-level netlist are automatically grouped into a bus (such as, `A_reg`). The default value of this option is *on*.
- **Group Buses:** When this option is turned *on*, synthesized nets (such as, `\A_reg[0], \A_reg[1], ..., \A_reg[7]`) in a gate-level netlist are automatically grouped into a bus (such as, `\A_reg`). The default value of this option is *off*.

The following options are available in the **Create Flow View Method** section (only one option may be selected at a time):

- **Temporal Flow View:** Identifies and displays data and control transfers using logic symbols. This is the default view.
- **Compact Temporal Flow View:** Identifies and displays data and control transfers through multi-level combinational logic within one or more register-to-register transfer stages.
- **Temporal Register View:** Identifies and displays data and control transfers between register stages across multiple clock cycles.

The following options are available in the **Trace Options** section:

- **Show Cycle-by-cycle Details of Holding Registers:** When this option is turned *on*, every node on the flow view is expanded cycle-by-cycle when a trace command is invoked. When this option is turned *off*, cycle-by-cycle details of held registers are not displayed when a trace command is invoked. Either one or zero of the ‘held’ register nodes (jumps to the first change) are displayed because each cycle is expanded. This has the same effect as

invoking the **Trace -> Show Cycle-by-Cycle Details of Holding Registers** in a flow view window.

- **Flatten All Arrays When Tracing:** When this option is turned *on*, multiple dimensional arrays are flattened into buses. When this option is turned *off*, multiple dimensional arrays are displayed as memories. The default value of this option is *off*.
- **Open nSchema Automatically When Tracing Stops:** When this option is turned *on*, the *nSchema* window is opened and the unexpandable signal is highlighted when tracing stops at an unexpandable signal. When this option is turned *off*, the *nSchema* window is not opened with the unexpandable signal highlighted when tracing stops at an unexpandable signal. The default value of this option is *off*.

NOTE: Execute the cycle-based **Trace this Value** command or the transition-based **Trace Triggering Path** command to perform this option.

Cycle-based Page

Use this page to specify the options for automatic trace commands in cycle-based flow views.

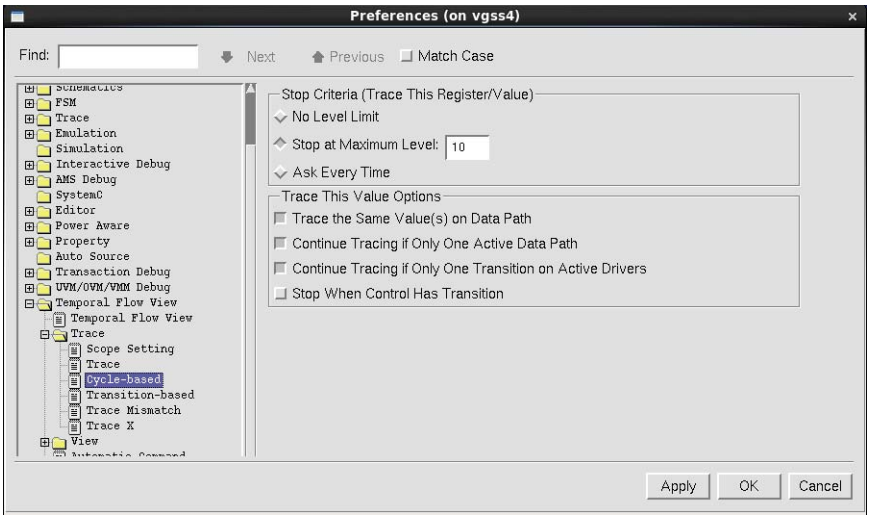


Figure: Cycle-based Page

In the **Stop Criteria (Trace This Register/Value)** section, specify the maximum number of levels that are displayed following the use of **Trace This Register** or **Trace This Value** command by selecting one of the following options:

- **No Level Limit:** No limitation on the number of levels to be traced. The default value of this option is *off*.
- **Stop at Maximum Level = n** (where *n* is an integer value that can be modified): Set the maximum number of levels to be traced. The default value of this option is *on* with a value of *10*.
- **Ask Every Time:** When this option is selected, the *Set Trace Option* form opens each time the **Trace This Value** command is invoked. A time to trace to or the number of cycles can then be specified. The trace options for the **Trace This Value** command can also be changed. The default value of this option is *off*.

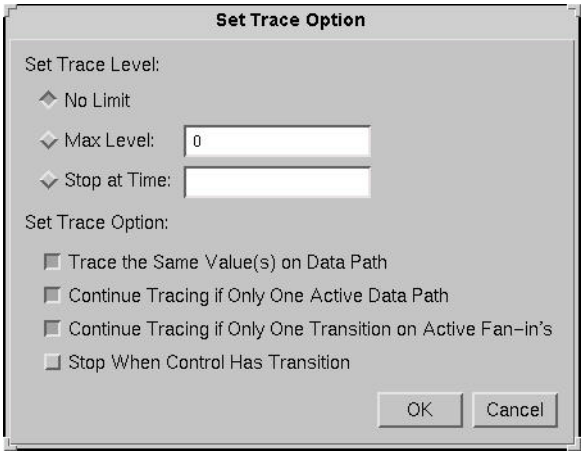


Figure: Set Trace Option Form

In the **Trace This Value Options** section, multiple options may be selected. The results of these options are OR. Whenever one condition is true, tracing continues from the corresponding nodes. The following options are available:

- **Trace the Same Value(s) on Data Path:** Over multiple cycles, trace all active data signals whose value matches the original data value. When matched values on the active data path are found, they are taken as new root nodes and the operation continues. A match means an exact value match (such as, Vin = Vout), unknown (X) and don't care (?) values do not apply. The default value of this option is *on*.
- **Continue Tracing if Only One Active Data Path:** If only one active data fan-in exists, continue tracing on the signal even if the value does not match the original data value. The active data signal is taken as a new root node and the operation continues. The default value of this option is *on*.
- **Continue Tracing if Only One Transition on Active Fan-in's:** If only one active fan-in with transition exists, continue tracing on the signal with a transition regardless of the value. It does not matter if the active fan-in is data or control as long as it is the only one with an active transition. The signal with a transition is taken as a new root node and the operation continues. The default value of this option is *on*.
- **Stop when Control Has Transition:** If a transition on a control signal exists, the trace operation stops. The default value of this option is *off*.

Transition-based Page

Use this page to specify the options for automatic trace commands in transition-based flow views.

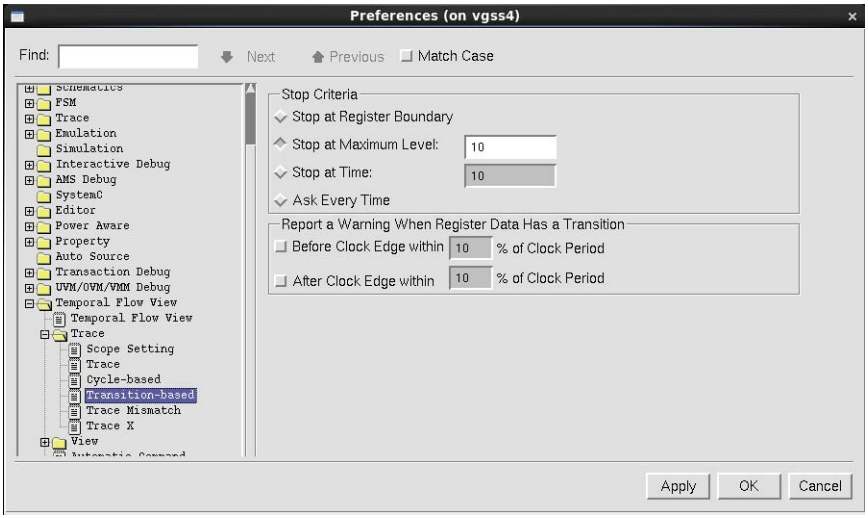


Figure: Transition-based Page

In the **Stop Criteria** section, specify the stop criteria by selecting one of the following options:

- **Stop at Register Boundary:** Trace triggering path stops at a register boundary. The default value of this option is *off*.
- **Stop at Maximum Level:** Trace triggering path stops after the specified level is reached. The default value of this option is *off*.
- **Stop at Time:** Trace triggering path stops after the specified time is reached. The default value of this option is *off*.
- **Ask Every Time:** When this option is selected, the *Set Triggering Option* form opens each time the **Trace Triggering Path** command is invoked. The trace options for the **Trace Triggering Path** command can also be changed. The default value of this option is *on*.

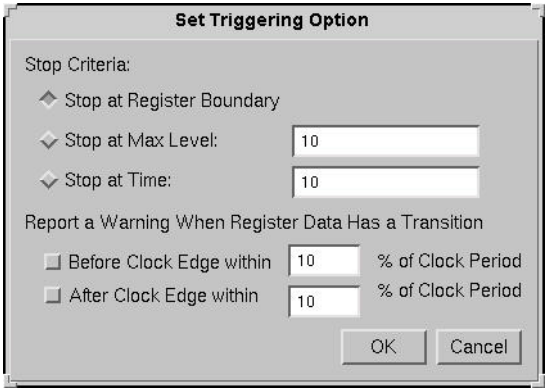


Figure: Set Triggering Option Form

The following options are available in the **Report a Warning When Register Data Has a Transition** section:

- **Before Clock Edge within 'n' % of Clock Period** (where *n* is an integer value that can be modified): When this option is turned *on* and the register data has a transition within the specified time interval before the clock edge, a warning is reported. The default is *off* and *10*.
- **After Clock Edge within 'n' % of Clock Period** (where *n* is an integer value that can be modified): When this option is turned *on* and the register data has a transition within the specified time interval after the clock edge, a warning is reported. The default is *off* and *10*.

Trace Mismatch Page

Use this page to specify the setting for behavior trace for waveform mismatch.

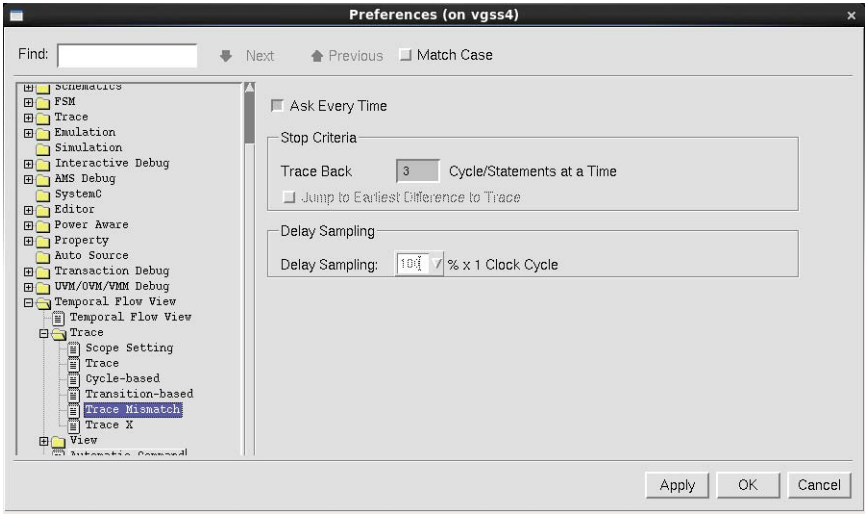


Figure: Trace Mismatch Page

The following options are available (only one option can be selected at a time):

- **Ask Every Time:** When this option is turned *on* and the **Trace -> Behavior Trace for Waveform Mismatch** command in the *Temporal Flow View* window is invoked, the *Trace Mismatch between Two FSDB Files* form opens where the time to trace to or the number of cycles to trace can be specified. The default value of this option is *off*.

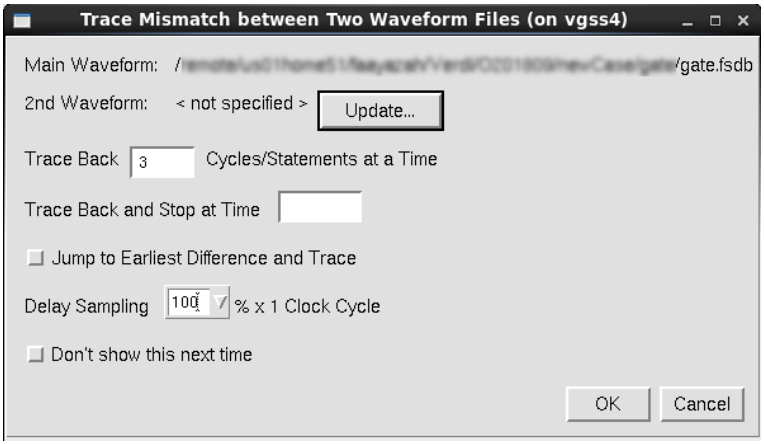


Figure: Trace Mismatch Between Two FSDB Files Form

- **Trace Back 'n' Cycles/Statements at a Time:** Specify the maximum number of levels to be traced in a single execution of the command. The default value is *on* with a value of *3*.
- **Jump to Earliest Difference and Trace:** Jump to the earliest simulation mismatch in the fan-in cone of the selected signal and begin tracing from that time. The default value of this option is *on*.
- **Delay Sampling: % x 1 Clock Cycle**: Set the value (percentage of clock cycle) of delay sampling either as **25**, **50**, or **100** of the clock cycle. The value is saved in the `novas.rc` resource file. The default value of this option is *100*.

Trace X Page

Use this page to specify the setting for tracing the source of unknown X values.

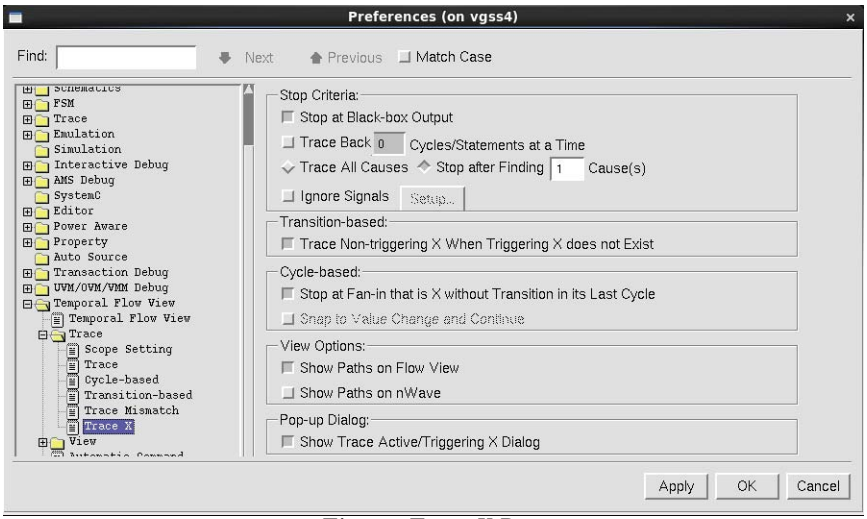


Figure: Trace X Page

The following options are available in the **Stop Criteria** section to control the trace active X function:

- **Stop at Black-box output**: During the trace back process, when this option is enabled, the traces are stopped at signals that are outputs of black boxes (for example, testbench, monitor, system tasks, or behavior model). When this option is disabled, the tracing from black box inputs with unknown value is continued. By default, the option is enabled.
- **Trace Back 'n' Cycles/Statements at a Time:** The value in the **Trace Back 'n' Cycles/Statements at a Time** option determines the number of cycles (Temporal Register View) or statements (Compact Temporal Flow View) to display for each trace. If the number of cycles to display is smaller than the number of cycles to the root cause of the unknown, the *Trace Active X Results* window does not open but the traced cycles are displayed on the flow view or *nWave*. It is only enabled when **Show Paths on nWave** option or **Show Paths on Flow View** are enabled.
- **Trace All Causes** or **Stop after Finding 'n' Cause(s):** The algorithm works in a depth first search manner. The algorithm can be set to **Stop after Finding 'n' Cause(s)** or **Trace All Causes**. Only one option can be selected at a time. The default is to stop tracing after the first cause is found.
- **Ignore Signals:** When this option is enabled, the **Setup** button is enabled. Click **Setup** to open the *Ignore Signal Settings* form to specify the ignore signal settings. By default the option is disabled.

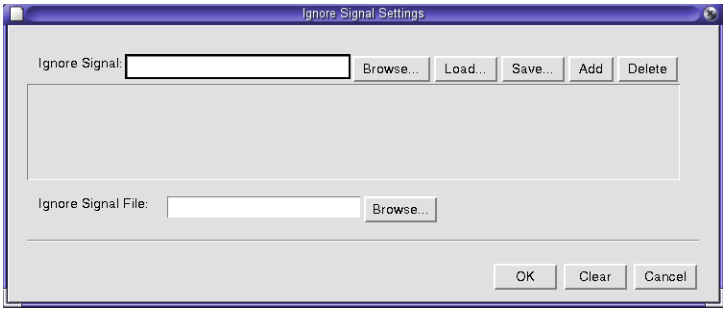


Figure: Ignore Signal Settings Form

The *Ignore Signal Settings* form includes the following fields and buttons:

- **Ignore Signal:** Enter the signal name in this text field to specify a signal.
- **Browse:** Open the *Signal Manager* form where the design hierarchy can be viewed and a signal specified.
- **Load:** Open the *File Manager* form where the directory structure can be viewed and a signal file specified.
- **Save:** Open the *File Manager* form to specify a text file to save signals in the signal list below.
- **Add:** Add specified signals to the signal list. The signal can be specified in the *Signal Manager* form which is opened by clicking the **Browse** button.
- **Delete:** Delete the specified signal from the signal list below.
- **Ignore Signal File:** Enter the signal file name in this text field to specify a signal file.
- **Browse:** Open the *File Manager* form where the design hierarchy can be viewed and a signal file specified.
- **OK:** Complete the ignore signal settings and close the *Ignore Signal Settings* form. The *Trace Active X Results* pane excludes signals according to the settings in the *Ignore Signal Settings* form.
- **Clear:** Clear all signals in the signal list.
- **Cancel:** Cancel changes to the *Ignore Signal Settings* form and close the form.

- **Transition-based**:
  - **Trace Non-triggering X When Triggering X does not Exist:** When this option is enabled, the unknown value on the non-triggering fan-in

signals is traced when the triggering inputs have an unknown value. When this option is disabled, only the triggering input with a transition from a known value to an unknown value is traced. By default, the option is disabled.

- **Cycle-based**:
  - **Stop at Fan-in that is X without Transition in its Last Cycle**: When this option is enabled, the tracing is stopped for the unknown transition when no fan-in signals exist with a transition from a known value to unknown value during the last cycle. When this option is disabled, the tracing is continued for the fan-in signals that are unknown and do not have any transition (or value change) in the last cycle of fan-in signal's clock domain. By default, this option is enabled.
  - **Snap to Value Change and Continue:** When this option is enabled and an unknown transition exists for multiple cycles, then the time snaps to the point where the unknown transition begins and then continues the trace process. When this option is disabled, the time does not snap to where the transition to unknown transition begins. By default, this option is disabled.

The following options are available in the **View Options** section.

- **Show Paths on Flow View:** The results are displayed on the Trace Active X Results Window. When the **Show Paths on Flow View** option is enabled, the results are also displayed on the flow view. Showing the results on the flow view may slow down the search process. By default the option is disabled. You must enable this option only when you want to display the trace X results on the flow view.
- **Show Paths on nWave:** The results are displayed on the Trace Active X Results Window. When the **Show Paths on nWave** option is enabled, the results are also displayed in the *nWave* window as waveforms. The default is *off*. You must enable this option only when you want to display the trace X results on the *nWave* window.
- **Show Details for the Last Fan-in Cone (only enable on Register View):** On Register View, this option provides few details for the Last Fan-in Cone.

View Folder

The **View** folder includes **View, Display,** and **Advanced** pages.

View Page

Use this page to associate different colors with objects in the main display area.

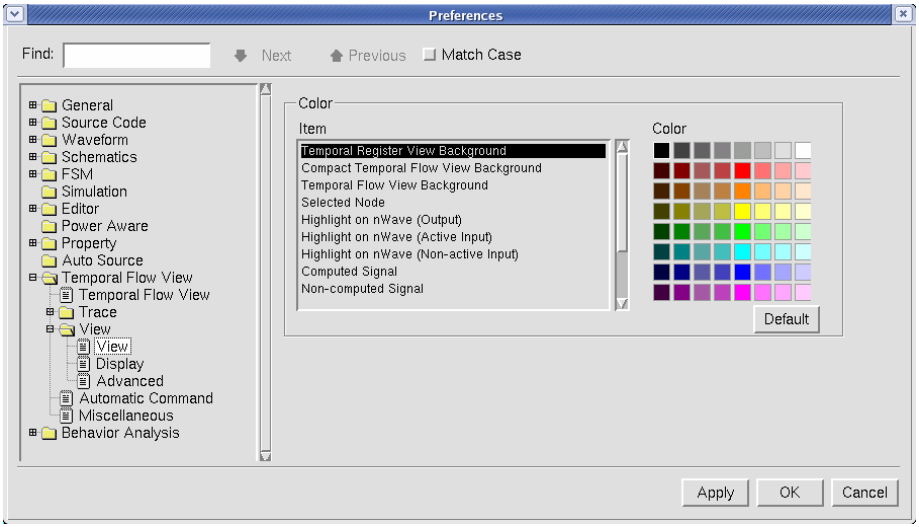


Figure: View Page

Item: Select the type of object to change color attributes for. Options are as follows (only one can be selected at a time):

- **Temporal Register View Background:** Specify the background color of the *Temporal Register View* window. The default is *black*.
- **Compact Temporal Flow View Background:** Specify the background color of the *Compact Temporal Flow View* window. The default is *black*.
- **Temporal Flow View Background:** Specify the background color of the *Temporal Flow View* window. The default is *black*.
- **Selected Node:** Specify the highlight color for a selected node (left-click). The default is *dark red*.
- **Highlight on nWave (Output):** Specify the highlight color for output signals on *nWave*. This is displayed as a bar or an arrow on the signal in *nWave*. The default is *bright yellow*.
- **Highlight on nWave (Active Input):** Specify the highlight color for active input signals on *nWave*. This is displayed as a bar or an arrow on the signal in *nWave*. The default is *bright yellow*.
- **Highlight on nWave (Non-active Input):** Specify the highlight color for non-active input signals on *nWave*. This is displayed as a bar or an arrow on the signal in *nWave*. The default is *gray*.
- **Computed Signal:** Specify the annotation color for computed signals. The default is *green*.

- **Non-computed Signal:** Specify the annotation color for non-computed signals. The default is *dark blue*.
- **Value 0:** Specify the annotation color for signals with value '0'. The default is *yellow*.
- **Value 1:** Specify the annotation color for signals with value '1'. The color is *orange*.
- **Value x:** Specify the annotation color for signals with value 'x'. The color is *purple*.
- **Value z:** Specify the annotation color for signals with value 'z'. The color is *light orange*.

Color: Select (left-click) the target color in the color matrix.

Default: Return the selected **Type** to its default color.

Display Page

Use this page to select the objects to be displayed automatically, instead of enabling them manually each time.

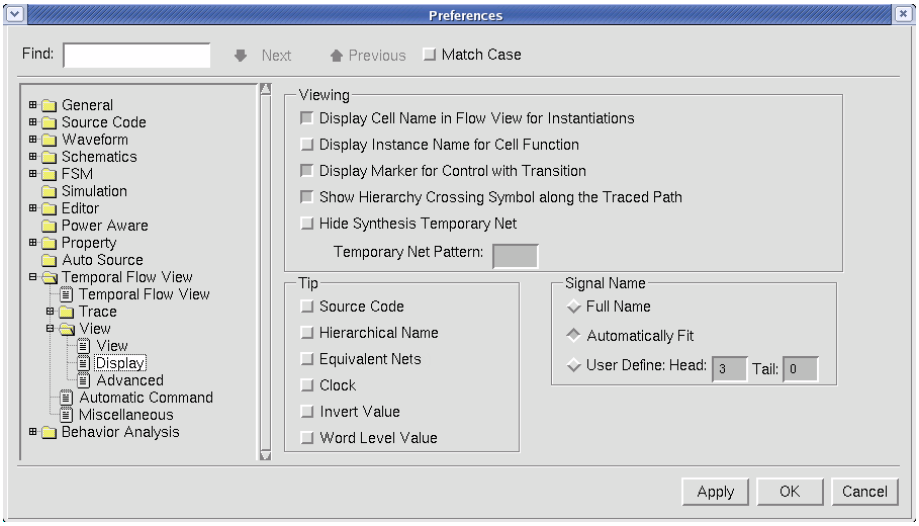


Figure: Display Page

The following options are available in the **Viewing** section:

- **Display Cell Name in Flow View for Instantiations:** When this option is turned *on*, the master cell name appears on the cell instantiation statement. The default value of this option is *on*.
- **Display Instance Name for Cell Function:** When this option is turned *on*, the instance name is displayed above its cell name. The default value of this option is *off*.
- **Display Marker for Control with Transition:** When this option is turned *on*, a yellow exclamation point is placed over output nodes that contain control signals with a transition. The default value of this option is *on*.
- **Show Hierarchy Crossing Symbol along the Traced Path:** When this option is turned *on*, the *nSchema* hierarchy crossing symbol (square box with an 'x') is displayed on the path when a hierarchy is crossed. This option only affects the *Temporal Flow View*. The default value of this option is *on*.
- **Hide Synthesis Temporary Net:** When this option is turned *on*, the *Temporal Flow View* skips temporary nets from the synthesis tool that match the pattern specified in the **Temporary Net Pattern** text field. The Verdi platform continues tracing until it encounters a net that does not match the pattern. Only the local name is matched. The output net of a

flip-flop is not hidden even if its name matches the pattern. When this option is turned *off*, all nets are displayed. This option only affects the *Temporal Flow View*. The default value of this option is *off*.

Temporary Net Pattern: Use regular expressions to specify the pattern to match. Multiple patterns may be specified and separated by a semi-colon (such as, A*;SUM*).

When any of the options in the **Tip** section are turned *on*, the item is displayed in a tip in the main flow view display area when the **View -> Turn on Tip** command is enabled. All options are turned *off* by default:

- **Source Code:** Shows the source code.
- **Hierarchical Name:** Shows the full hierarchical path as opposed to just the local name.
- **Equivalent Nets:** Shows other names used for this net in the design hierarchy.
- **Clock:** Shows the signals that are defined as the clock controlling this net or node.
- **Invert Value:** Shows the inversion of the current value on this net or node.
- **Word Level Value:** Shows the whole bus (multi-bit signal) value for this partial bus node. This tip is only displayed if the node has been created with the **Trace -> Show Active Statement for Partial Bus** command. Aliases that are applied to the whole bus are not displayed in the tip.

The following options are available in the **Signal Name** section (only one option can be selected at a time):

- **Full Name:** Displays the full signal name. The default value of this option is *off*.
- **Automatically Fit:** Displays the portion of the signal that fits in the allowable space. The part that does not fit is denoted with an '*'. The default value of this option is *on*.
- **User Define: Head**: <value>, **Tail**: <value>: Specify the beginning and end of the signal to be displayed in the allowable space. The default value of this option is *off*.

Advanced Page

Use this page to specify options to control the placement of nodes in *Compact Temporal Flow View* and *Temporal Register View*.

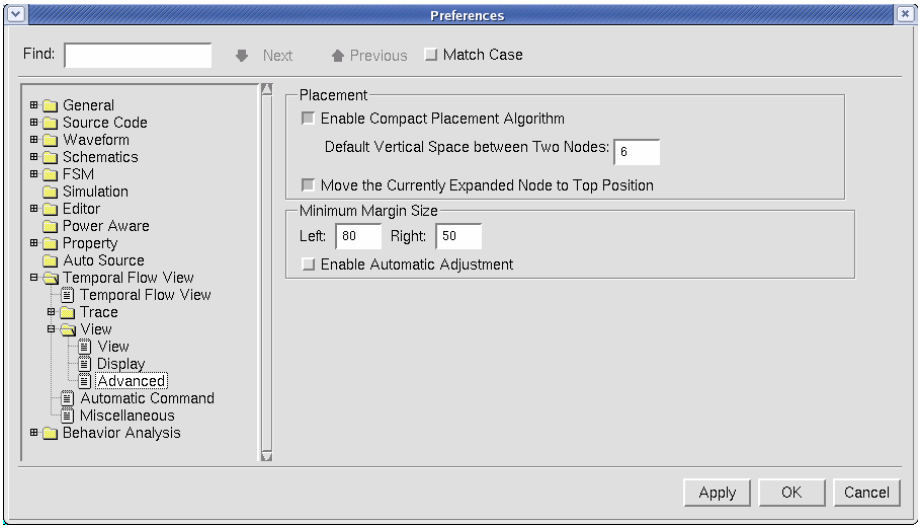


Figure: Advanced Page

The following options are available the **Placement** section:

- **Enable Compact Placement Algorithm:** When this option is turned *on*, the output node is placed at the top of the display area. When this option is turned *off*, the node is expanded horizontally. The default value of this option is *on*.
  - **Default Vertical Space between Two Nodes: n:** The default is *6*. The vertical space is the distance between the last input node of the top trace to the output node of the bottom trace. The spacing can be increased or decreased by changing the value in the field.
- **Move the Currently Expanded Nodes to Top Position:** When this option is turned *on*, the expanded input node is moved to the top-most position after every expansion. When this option is turned *off*, the input node remains in the original position. The default value of this option is *on*.

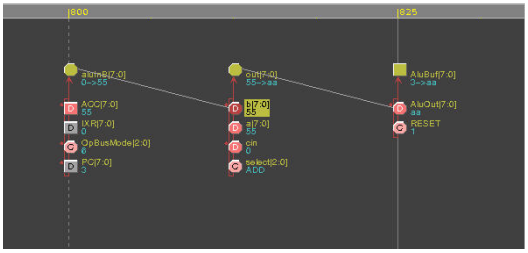


Figure: Enable Compact Placement Algorithm and Move the Currently Expanded Nodes to the Top Position - Both Options On

Figure: Enable Compact Placement Algorithm - Option Off

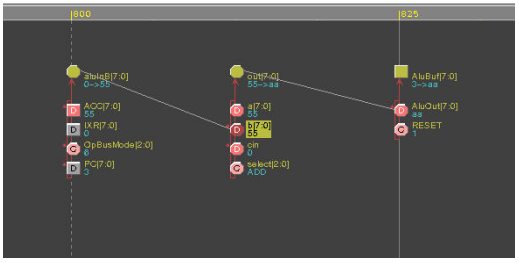


Figure: Move the Currently Expanded Nodes to Top Position - Option Off

The following options are available in the **Minimum Margin Size** section:

- **Left:** Specify the left margin for viewing. The default is *80*.
- **Right:** Specify the right margin for viewing. The default is *50*.
- **Enable Automatic Adjustment:** When this option is turned *on*, the signal name fits into the viewing range. The default value of this option is *off*.

Automatic Command Page

Use this page to specify the types of actions that should occur automatically.

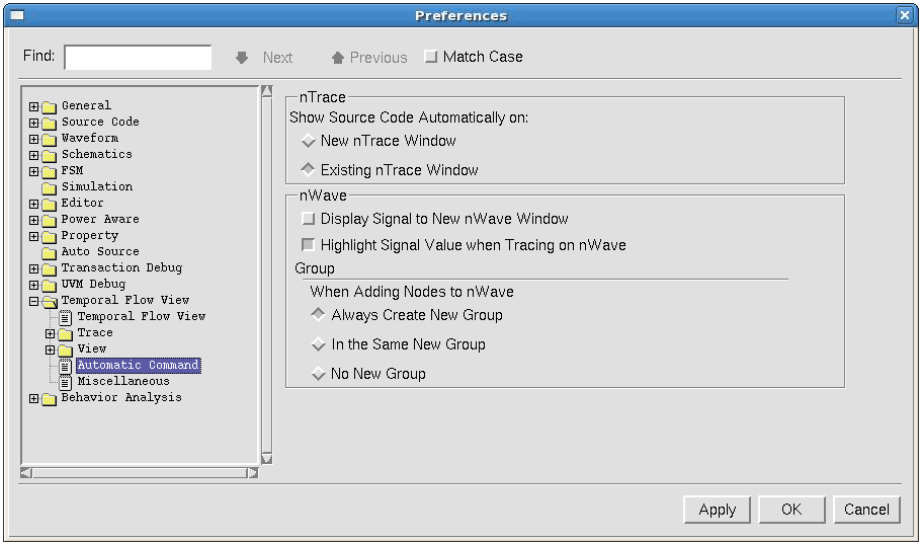


Figure: Automatic Command Page

The following options are available in the **nTrace** section (only one option may be selected at a time):

- **New nTrace Window:** Open a new *nTrace* window to show the source code of a selected node. The default value of this option is *on*.
- **Existing nTrace Window:** Show the source code of a selected node in the current *nTrace* window. The default value of this option is *off*.

The following options are available in the **nWave** section:

- **Display Signals to New nWave Window:** When this option is turned *on*, a new waveform frame opens when **Show Drivers on nWave**, **Show Synchronization Signal**, **Show All Traced Signals on nWave**, or **Trace This Value** command is executed. Additional traces add signals to this new *nWave* pane. The default value of this option is *off*.
- **Highlight Signal Value when Tracing on nWave:** When this option is turned *on*, the traced path is highlighted on the waveform. Internal net sample points are identified with an arrow and registers are identified with a bar. The default value of this option is *on*.

The following options are available in the **When Adding Nodes to nWave** section (only one option can be selected at a time):

- **Always Create New Group:** When this option is selected, the added signals are always added to a new group in the *nWave* window. The traced signals are the results of invoking **Show Drivers on nWave**, **Show Synchronization Signal**, **Show All Traced Signals on nWave**, or **Trace This Value** commands. The default value of this option is *on*.
- **In the Same New Group:** When this option is selected, a new group is created to add signals when **Show Drivers on nWave**, **Show Synchronization Signal**, **Show All Traced Signals on nWave**, or **Trace This Value** command is executed for the first time. After a new group is created, all signals added by the above command are added to the same group. The default value of this option is *off*.
- **No New Group:** When this option is selected, the added signals are added at the current cursor location. A new group is not created. The default value of this option is *off*.

Miscellaneous Page

Use this page to select miscellaneous mouse operation behavior.

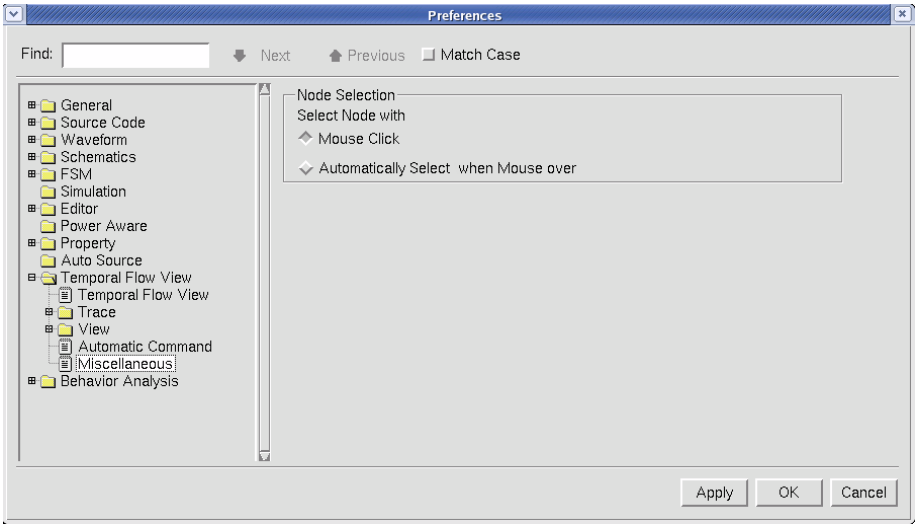


Figure: Miscellaneous Page

The following options are available in the **Node Selection** section (only one option may be selected at a time):

- **Mouse Click:** The node is selected by clicking the left mouse button. The default value of this option is *on*.

- **Automatically Select when Mouse over:** The node is selected when the mouse moves over the node. The default value of this option is *off*.

Behavior Analysis Folder

These preferences are for the *Behavior Analysis* engine.

Behavior Analysis Page

Click the **Restore to Default Settings** option in this page to reset the settings to the default settings.

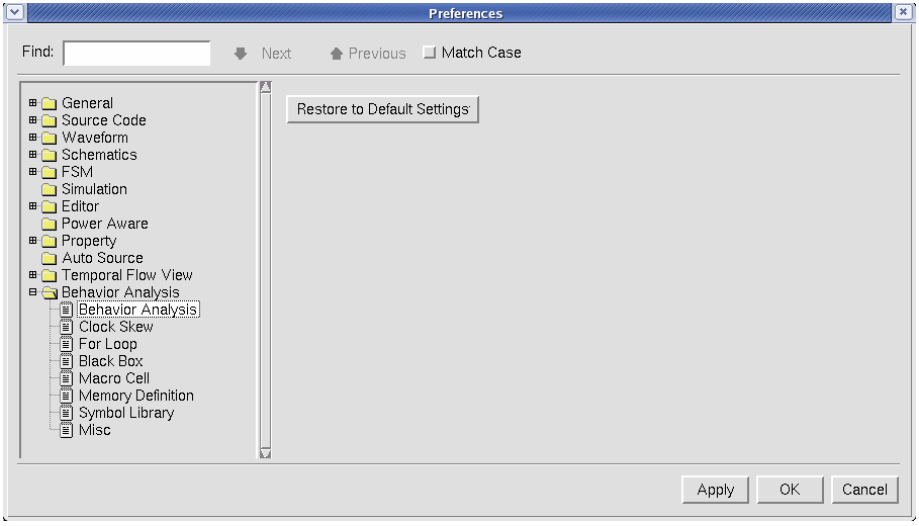


Figure: Behavior Analysis Page

Clock Skew Page

Use this page to specify the clock skew. This option is typically used for gate level designs. Since the behavioral analysis engine uses a cycle-based zero delay model, this option identifies clocks as the same if the skew is within the specified range.

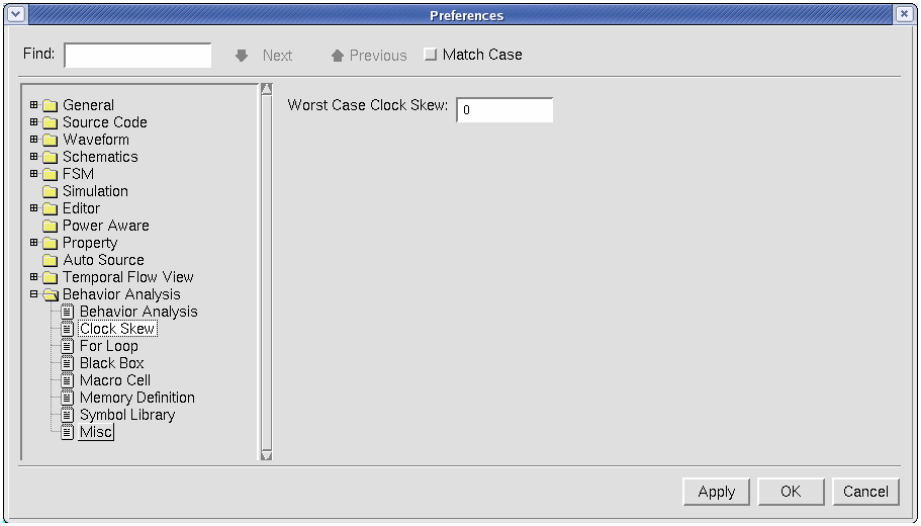


Figure: Clock Skew Page

The following option is available:

- **Worst Case Clock Skew:** Enter the worst case clock skew. If the exact clock skew is not known, the recommendation is to use 1/10th of the smallest clock period. The time units for the clock skew come directly from the loaded FSDB file.

For Loop Page

Use this page to specify what to do with for loops in an RTL design.

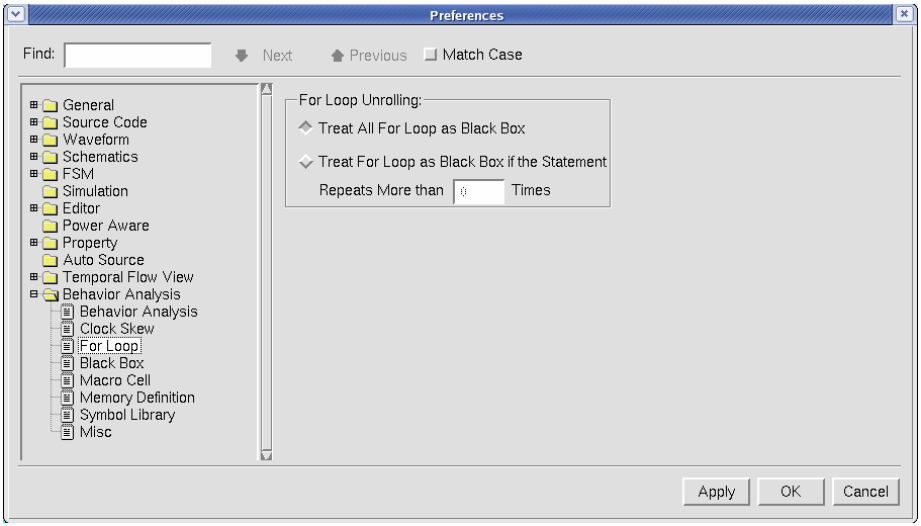


Figure: For Loop Page

The following options are available in the **For Loop Unrolling** section (only one option may be selected at a time):

- **Treat All For Loop as Black Box:** When this option is selected, any logic within a 'for' loop is treated as a black box. This is the default value.
- **Treat For Loop as Black Box if the Statement Repeats More than N Times:** When this option is selected, logic within a 'for' loop is unrolled if the 'for' loop iterator value is less than or equal to 'N'. Any 'for' loops with iterators greater than 'N' are black-boxed. The user can enter the value for 'N' in the text field.

Black Box Page

Use this page to specify the scopes to be treated as a black box. When a scope is set as a black box, the *Behavior Analysis* engine does not perform analysis inside the scope and trace functions stop at the scope boundary.

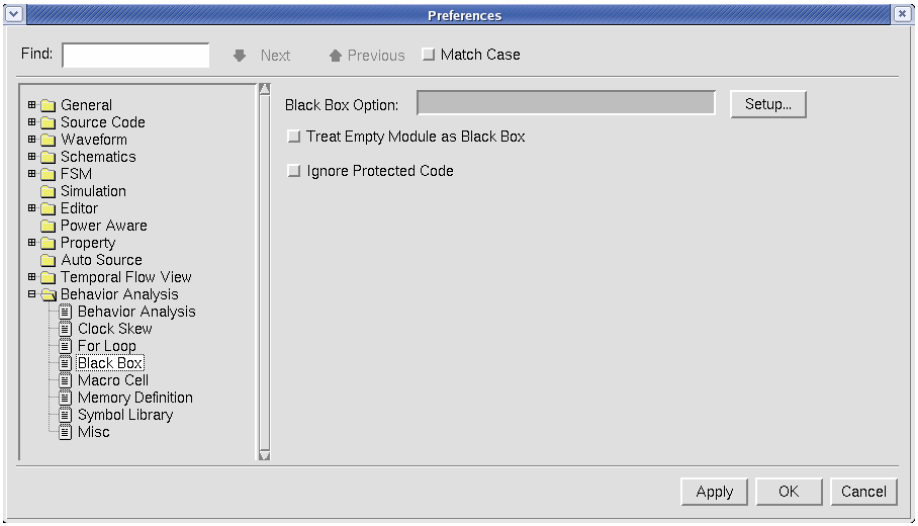


Figure: Black Box Page

The following field and options are available:

- **Black Box Option:** This text field displays the list of scopes (`-BBoxModule`), files (`-BBoxModuleFile`) and system tasks (`-bboxSysTaskFile`) specified as black boxes. This field cannot be edited directly. Click the **Setup** button to open the *Black Box Setup* form to define or change scopes to treat as black boxes. Refer to the *Black Box Setup Form* section for details.
- **Treat Empty Module as Black Box:** When this option is turned *on*, modules that are empty are automatically treated as a black box. When this option is turned *off*, empty modules are not automatically black-boxed. The default value of this option is *off*.
- **Ignore Protected Code:** For general VIP protected code without the `//vcs_vip_protect pragma`, when this option is turned *on*, Behavior Analysis inferences module or scope that contains the encrypted protected code but objects inside the protected code are black boxed. When this option is turned *off*, Behavior Analysis black boxes the whole module or scope that contains the encrypted protected code. The default value of this option is *off*.

Black Box Setup Form

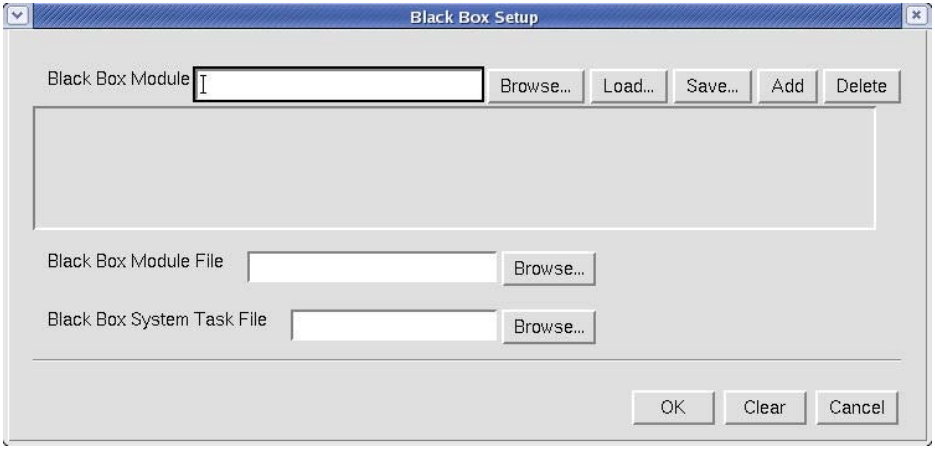


Figure: Black Box Setup Form

The following text fields and buttons are available in the *Black Box Setup* form:

- **Black Box Module:** Specify one or more scopes to be defined as black boxes. Enter the module name or architecture (entity) in the text field directly or click the **Browse** button to open the *Module Manager* form to view the design and select a scope to black box. Click the **OK** button to add the selected scope to the **Black Box Module** text field and close the *Module Manager* form or click the **Cancel** button to exit the form without saving the scope.

  The following regular expressions are supported for module name searching:

Supported Regular Expressions		
Syntax	**Expression**	**Description**
* (asterisk)	Zero or more	Finds zero or more occurrences of the preceding expression.
? (question mark)	Any single character	Matches any single character.
\ (back slash)	A single character	Matches the single character that follows \ exactly. This syntax can avoid interpretation of special characters, such as *?[]\.
[chars] (square bracket)	Any single character	Matches any single character in chars. If chars contains a sequence, such as a-b, any character between a and b inclusive are matched.

After a module name or architecture (entity) is entered in the **Black Box Module** text field, click the **Add** button to add the specified scope to the list in the center pane.

- **Load:** This button opens the *File Manager* form where the directory structure can be viewed and a previously saved black box module list can be loaded. The default file extension is *.mod*. Multiple files can be loaded. Each file's list is appended to the previously loaded file. The module name or architecture (entity) can be listed on the same line with a space separator or on separate lines in the file list.

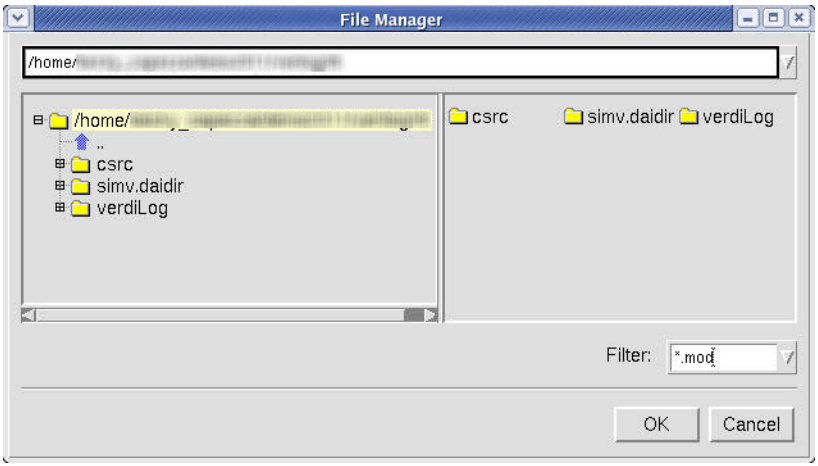


Figure: File Manager Form

- **Save:** This button opens the *File Manager* form where the directory structure can be viewed and the listed scopes can be saved to a file. The default file extension is *.mod*. All module names and architecture (entity) pairs displayed in the center pane are saved to the specified file name.
- **Add:** Click this button to add the current scope listed in the **Black Box Module** text field to the list in the center pane.
- **Delete:** Click this button to delete the selected module name or architecture (entity) from the list. Only one scope can be selected at a time.
- **Black Box Module File:** Specify the file name to be loaded containing a list of scopes to be black-boxed. This file is appended to scopes previously defined although the detailed scope list is not visible in the center pane of the *Black Box Setup* form. Click the **Browse** button to invoke the *File Manager* form to view the directory structure and load a previously saved scope list. The default file extension is *.mod*.

In the scope file, each module name or architecture (entity) pair can be listed on a separate line or on the same line with a space separator. Regular expressions are not supported. For example:

```
CPU
alub(alub)
```

or

```
CPU alub(alub)
```

- **Black Box System Task File:** Specify the file name to be loaded containing a list of system tasks to be black-boxed. This file is appended to scopes previously defined although the detailed system task list is not visible in the center pane of the *Black Box Setup* form. Click the **Browse** button to open the *File Manager* form to view the directory structure and load a previously saved system task list. The default file extension is *.task*.

NOTE: If a scope contains a system task that is not specified as a black box (or recognized by default), the entire scope is black-boxed by the Verdi platform.

- **OK:** Applies the settings and closes the *Black Box Setup* form. Scopes listed in the center pane are added to the **Black Box Option** text field with the `-BBoxModuleFile` option. The file name in the **Black Box Module File** field is added to the **Black Box Option** text field with the `-BBoxModule` option. The file name in the **Black Box System Task File** field is added to the **Black Box Option** text field with the `-BBoxSysTaskFile` option.
- **Clear:** Clears all settings. The *Black Box Setup* form remains open.
- **Cancel:** Closes the *Black Box Setup* form. The previous settings remain in the form.

Macro Cell Page

Use this page to specify the scopes to define as a macro cell. When a scope is set as a macro cell, the behavioral analysis engine still performs analysis inside the scope and trace functions go through the scope. However, the internal signals inside a macro cell are skipped or displayed as an internal state on the flow view.

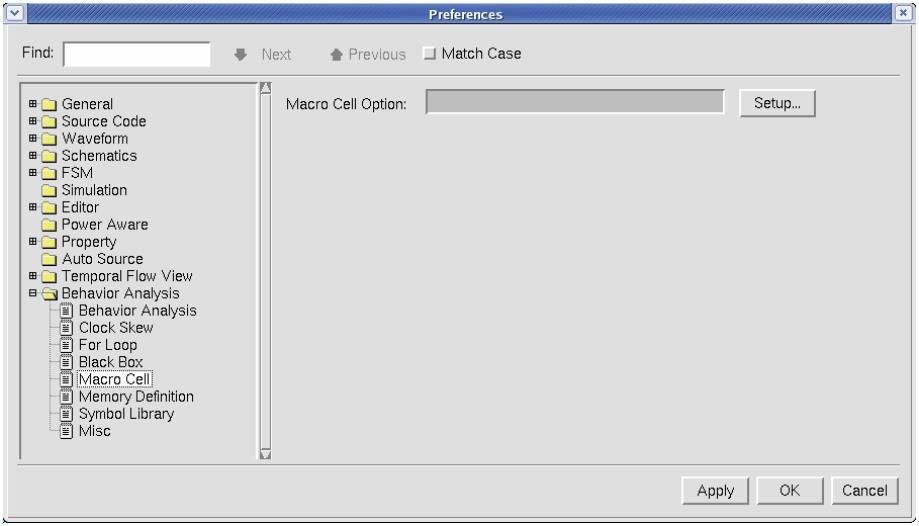


Figure: Macro Cell Page

The following field is available:

- **Macro Cell Option:** This text field displays the list of scopes (-MacroCell) and files (-MacroCellFile) specified as macro cells. This field cannot be edited directly. Click the **Setup** button to open the *Macro Cell Option* form to define or change the scopes to treat as macro cells. Refer to the *Macro Cell Option Form* section for details.

Macro Cell Option Form

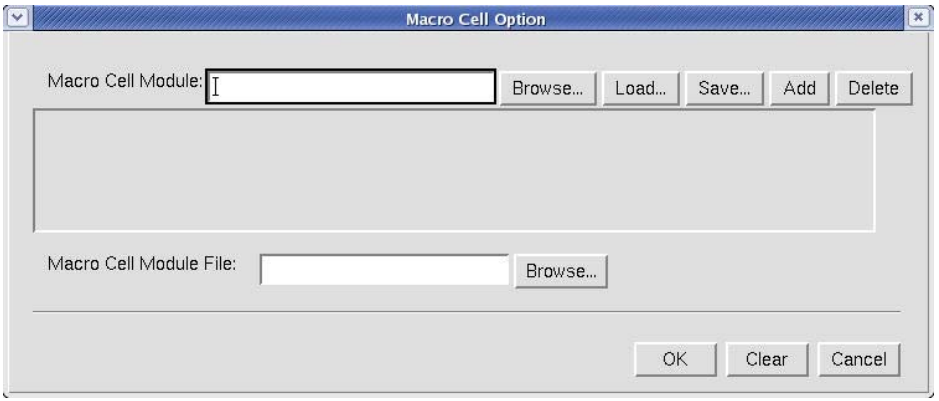


Figure: Macro Cell Option Form

The following fields and buttons are available on the *Macro Cell Option* form:

- **Macro Cell Module:** Specify one or more scopes to be defined as macro cells. Enter the module name or the architecture (entity) in the text field directly or click the **Browse** button to open the *Module Manager* form to view the design and select a scope to define as a macro cell. Click the **OK** button to add the selected scope to the **Macro Cell Module** text field and close the *Module Manager* form or click the **Cancel** button to exit the form without saving the scope.

  Regular expressions are supported for cell name searching. Refer to the *Supported Regular Expressions* table for details.

  After a module name or architecture (entity) is entered in the **Macro Cell Module** text field, click the **Add** button to add the specified scope to the list in the center pane.

- **Load:** This button opens the *File Manager* form where the directory structure can be viewed and a previously saved macro cell module list can be loaded. The default file extension is *.mod*. Multiple files can be loaded. Each file's list is appended to the previously loaded file. The module name or architecture (entity) can be listed on the same line with a space separator or on separate lines in the file list.
- **Save:** This button opens the *File Manager* form where the directory structure can be viewed and the listed scopes can be saved to a file. The default file extension is *.mod*. All module names and architecture (entity) pairs displayed in the center pane are saved to the specified file name.
- **Add:** Click this button to add the current scope listed in the **Macro Cell Module** text field to the list in the center pane.

- **Delete:** Click this button to delete the selected module name or architecture (entity) from the list. Only one scope can be selected at a time.
- **Macro Cell Module File:** Specify the file name to be loaded containing a list of scopes to be defined as macro cells. This file is appended to scopes defined previously although the detailed scope list is not visible in the center pane of the *Macro Cell Option* form. Click the **Browse** button to open the *File Manager* form where the directory structure can be viewed and a previously saved scope list can be loaded. The default file extension is *.mod*.

  In the scope file, each module name or architecture (entity) pair can be listed on a separate line or on the same line with a space separator. Regular expressions are supported. For example:

  ```
  CPU
  alub(alub)
  pram_[1-2]
  ```

  or

  ```
  CPU alub(alub) pram_.*
  ```

- **OK:** Applies the settings and closes the *Macro Cell Option* form. Scopes listed in the center pane are added to the **Macro Cell Option** text field with the *-MacroCell* option. The file name in the **Macro Cell Module File** field is added to the **Macro Cell Option** text field with the *-MacroCellFile* option.
- **Clear:** Clears all settings. The *Macro Cell Option* form remains open.
- **Cancel:** Closes the *Macro Cell Option* form. The previous settings remain in the form.

Memory Definition Page

Use this page to specify any memories that are defined as a PLI. The *Behavior Analysis* engine can trace the memory contents with memory definition.

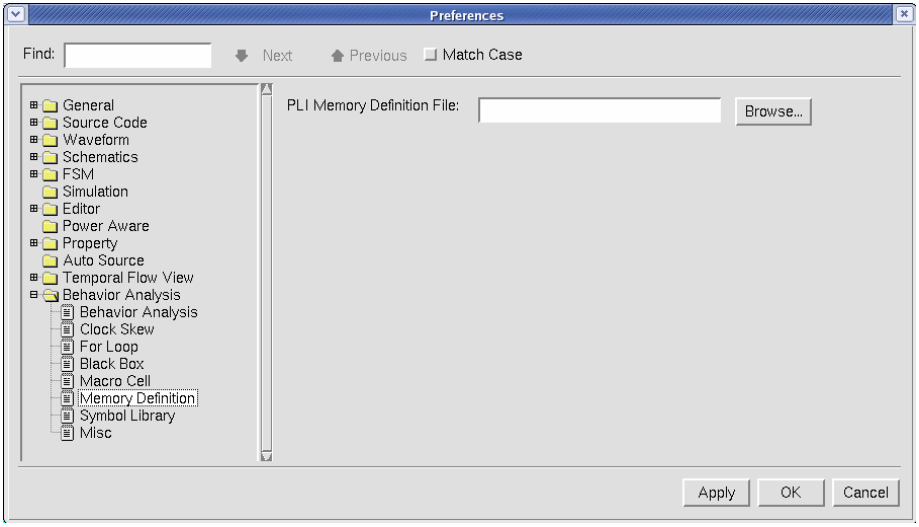


Figure: Memory Definition Page

PLI Memory Definition File: This text field displays the directory path and PLI memory definition file name to be used. Click the **Browse** button to open the *Load* form where the directory can be viewed to load the file. The following is an example of a memory definition (.def) file:

For example, if the following PLI memory call exists in the design:

```
$damem_declare("mem",0,63,0,63);
$damem_write("mem", addr, temp_data);
$damem_read("mem", addr, dout);
```

A PLI memory definition file contains the following two lines:

```
API_MEM_DECL $damem_declare(MNAME, OTHERS, OTHERS, OTHERS,
OTHERS);
API_MEM_WRITE $damem_write(MNAME, ADDR, DATAIN);
API_MEM_READ $damem_read(MNAME, ADDR, DATAOUT);
```

The first line indicates the PLI memory is defined using the *$damem_declare* function. `MNAME` is a keyword which identifies the first parameter of the function as the memory name in the HDL design. `OTHERS` is another keyword which indicates the field is unnecessary for tracing PLI memory content.

The second line indicates data is written into the PLI memory using the `$damem_write` function. `ADDR` is a keyword which means the second parameter of the function represents the address. `DATAIN` is a keyword which means the third parameter of the function represents the data to be written into the PLI memory.

The third line indicates data is read from the PLI memory using the `$damem_read` function. `DATAOUT` is a keyword which means the third parameter of the function represents the data to be read from the PLI memory.

Symbol Library Page

Use this page to specify the symbol source in the *Temporal Flow View* window and the *Behavior Analysis* engine.

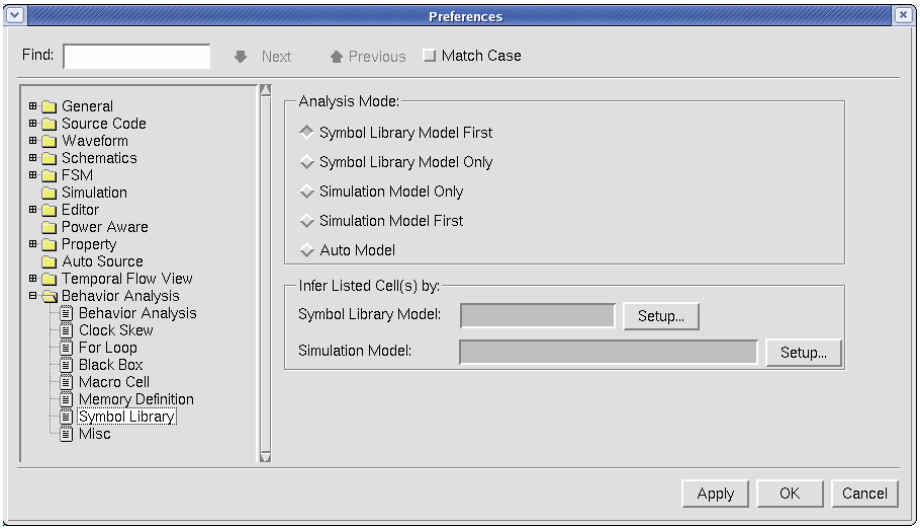


Figure: Symbol Library Page

The following options are available in the **Analysis Mode** section (only one option can be selected at a time):

- **Symbol Library Model First:** Use the cells from the symbol library first. If a cell does not have the correct pin function, use the simulation model for that cell. This is the default.
- **Symbol Library Model Only:** Only use the cells from the symbol library. If a cell does not have the correct pin function, it is black-boxed.

- **Simulation Model Only:** Use the simulation model to extract the cell type and determine the appropriate symbol. If a simulation model for a cell is non-synthesizeable, it is black-boxed.
- **Simulation Model First:** Use the simulation model to extract the cell type and determine the appropriate symbol. If the simulation model for a cell is non-synthesizeable, use the symbol library for the correct pin function.
- **Auto Model:** Automatically use “Simulation Model First” for combinational cells while using “Symbol Library Model First” for sequential cells.

The following fields are available in the **Infer Listed Cell(s) by** section:

- **Symbol Library Model:** This text field displays the list of modules that should reference the symbol library model. This field cannot be edited directly. Click the **Setup** button to open the *Symbol Library Cell Option* form to specify the modules. Refer to the *Symbol Library Option Form* section for details.
- **Simulation Model:** This text field displays the list of modules that should reference the simulation model. This field cannot be edited directly. Click the **Setup** button to open the *Simulation Model Option* form to specify the modules. Refer to the *Simulation Model Option Form* section for details.

Symbol Library Option Form

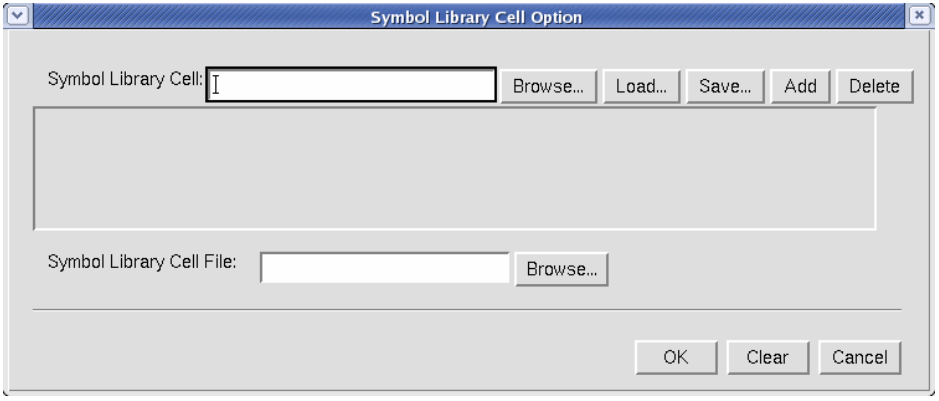


Figure: Symbol Library Option Form

The following fields and buttons are available:

- **Symbol Library Cell:** Specify one or more scopes that should reference the symbol library model. Enter the scope in the text field directly or click the **Browse** button to open the *Module Manager* form to select a scope to add to the list. Click the **OK** button to add the selected scope to the **Symbol**

Library Cell text field and close the *Module Manager* form or click the **Cancel** button to exit the form without saving the scope. Regular expressions are supported for cell name searching. Refer to the *Supported Regular Expressions* table for details.

After a scope is entered in the **Symbol Library Cell** text field, click the **Add** button to add the specified scope to the list in the center pane.

- **Load:** This button opens the *File Manager* form where the directory structure can be viewed and a previously saved scope list can be loaded. The default file extension is *.mod*. Multiple files can be loaded. Each file's list is appended to the previous load. The scopes can be listed on the same line with a space separator or on separate lines in the file list.
- **Save:** This button opens the *File Manager* form where the directory structure can be viewed and the listed scopes can be saved to a file. The default file extension is *.mod*. All scopes displayed in the center pane are saved to the specified file name.
- **Add:** Click this button to add the current scope listed in the **Symbol Library Cell** text field to the list in the center pane.
- **Delete:** Click this button to delete the selected scope from the list. Only one scope can be selected at a time.
- **Symbol Library Cell File:** Specify the file name to be loaded containing a list of scopes to be treated differently. This file is appended to scopes previously defined although the detailed scope list is not visible in the center pane of the form. Click the **Browse** button to invoke the *File Manager* form to view the directory structure and load a previously saved ignore scope list. The default file extension is *.mod*.

Simulation Model Option Form

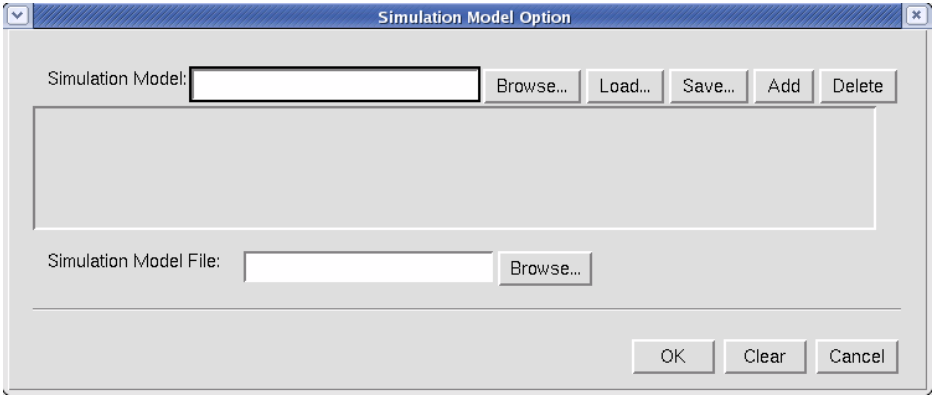


Figure: Simulation Model Option Form

The following fields and buttons are available:

- **Simulation Model:** Specify one or more scopes that should reference the simulation model. Enter the scope in the text field directly or click the **Browse** button to open the *Module Manager* form to select a scope to add to the list. Click the **OK** button to add the selected scope to the **Simulation Model** text field and close the *Module Manager* form or click the **Cancel** button to exit the form without saving a scope. Regular expressions are supported for simulation module name searching. Refer to the *Supported Regular Expressions* table for details.

  After a scope is entered in the **Simulation Model** text field, click the **Add** button to add the specified scope to the list in the center pane.
- **Load:** This button opens the *File Manager* form where the directory structure can be viewed and a previously saved scope list can be saved. The default file extension is *.mod*. Multiple files can be loaded. Each file's list is appended to the previous load. The scopes can be listed on the same line with a space separator or on separate lines in the file list.
- **Save:** This button opens the *File Manager* form where the directory structure can be viewed and the listed scopes can be saved to a file. The default file extension is *.mod*. All scopes displayed in the center pane are saved to the specified file name.
- **Add:** Click this button to add the current scope listed in the **Simulation Model** text field to the list in the center pane.
- **Delete:** Click this button to delete the selected scope from the list. Only one scope can be selected at a time.
- **Simulation Model File:** Specify a file to be loaded containing a list of scopes to be treated differently. This file is appended to scopes previously defined although the detailed scope list is not visible in the center pane of the *Simulation Model Option* form. Click the **Browse** button to invoke the *File Manager* form to view the directory structure and load a previously saved ignore scope list. The default file extension is *.mod*.

Misc Page

Use this page to specify miscellaneous options.

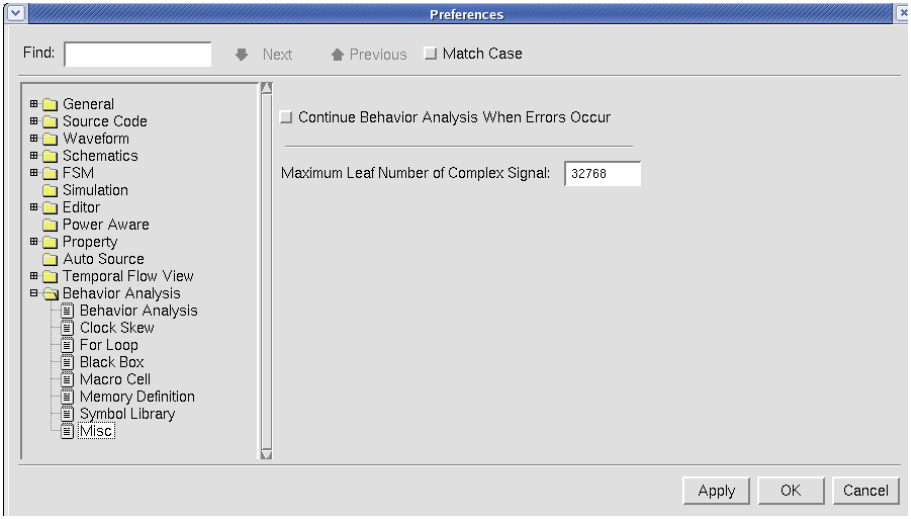


Figure: Misc Page

Continue Behavior Analysis When Errors Occur: This option specifies whether to ignore interactive error messages when errors are found while performing Behavior Analysis.

When this option is turned *off* and an error is found during Behavior Analysis, the message "*Errors occurred during loading the design. It is recommended that you fix these error(s) before continuing Behavior Analysis. Click Abort to cancel the current process and fix these error(s) or click Continue to ignore these error(s).*" is displayed. The program remains inactive until **Continue** or **Abort** is clicked. When this option is turned *on* and an error is found during Behavior Analysis, the message "*Error(s) found and ignored when loading design. It may cause incorrect results to behavior analysis.*" is displayed and Behavior Analysis continues without any interruption. The default value of this option is *off*.

Maximum Leaf Number of Complex Signal: Specify the maximum number of leaves for a complex signal. The default is *32768*.

Formal Verification

These preferences are for the SEQ Debug and VC_static_shell windows.

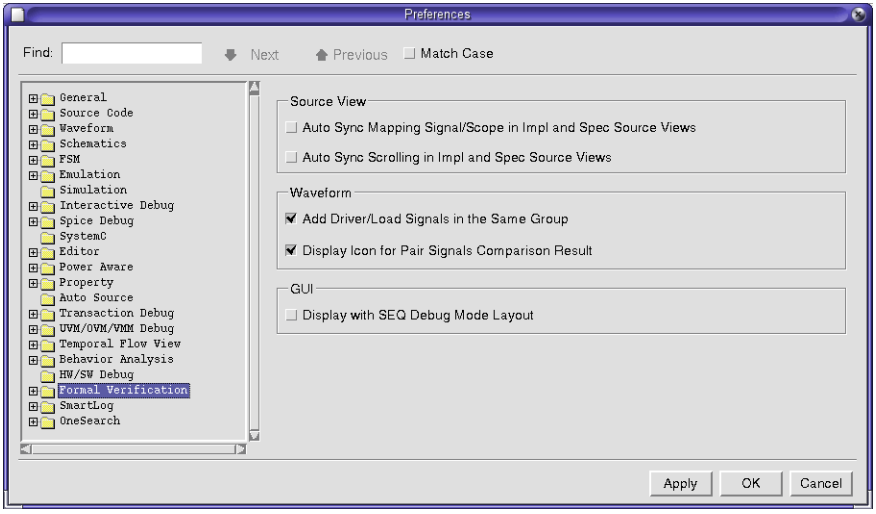


Figure: Formal Verification Page

SEQ Debug Page

The following options are available in the **Source View** section:

- **Auto Sync Mapping Signal/Scope in impl and spec Source Views:** This option turns on/off the mapping signal/scope selection syncing between the two impl/spec source view.
- **Auto Sync Scrolling in impl and spec Source Views:** This option turns on/off the scroll position syncing between the two impl/spec source view.

The following options are available in the **Waveform View** section:

- **Add Driver/Load Signals in the Same Group:** This options turns on/off the add driver/load signals for pair mapping signals in the same signal name group.
- **Display Icon for Pair Signals Comparison Result:** This option displays the pair mapping signals' waveform comparison result in signal pane with match/mismatch icon.

The following options are available in the **GUI View** section:

- **Display with SEQ Debug Mode Layout:** This option enables automatic change to SEQ debug mode layout. The *nWave* pane appears docked in top-right position, and the two *impl/spec* source view panes are docked at the bottom.

VC_static_shell Page

The following options are available in the **Source View** section:

- **Start up Option:** This option allows you to view and modify the vc_static_shell startup options.
- **Source View Active Annotation:** This option turns on/off the source view value annotation.

SmartLog Folder

These preferences are applicable for the *SmartLog* window.

General Page

Use this page to change the general settings associated with the *SmartLog* window.

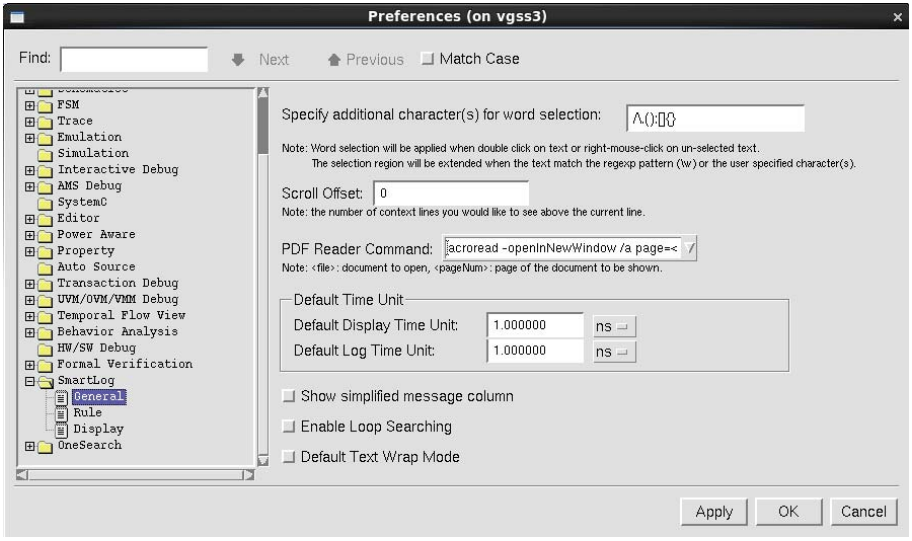
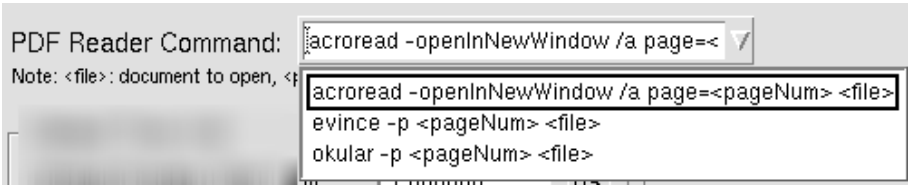


Figure: General Page

The following options are available in the **General** page:

- **Specify additional character(s) for word selection**: Use this option to specify additional character(s) for auto selection. Word selection is applied when you double-click the text or right-click on the selected text. The selection region is extended when the text matches the regexp pattern `(\w)` or the user specified additional character(s).
- **Scroll Offset**: Use this option to specify the number of context lines you want to see above the current line.
- **PDF Reader Command**: Use this option to select the pre-defined PDF Reader command or enter the preferred PDF Reader command for log files.

This setting is saved in the `novas.rc` file. The available pre-defined PDF Reader commands are illustrated in the following figure:



- **Default Time Unit**
  - **Default Display Time Unit**: Use this option to specify the default display time unit.
  - **Default Log Time Unit**: Use this option to specify the default log time unit.

NOTE: **Default Display Time Unit** and **Default Log Time Unit** preference settings do not influence the currently opened log files in *SmartLog*. These settings take effect when a new *SmartLog* window is opened and the settings are saved in the *novas.rc* file.

- **Show simplified message column**: Use this option to specify whether the <Message> column in the structure view shows full message or simplified message. The simplified message discards the message which is shown in the previous columns, like, <Severity>, <Time>, and so on.
  - **Off**: Shows full message in the <Message> column of structure view.
  - **On**: Shows simplified message in the <Message> column of structure view.
- **Enable Loop Searching**: Use this option to specify whether the next search starts over from the first line once the search hits the end of the log.
  - **Off**: Loop Searching is disabled, that is, the search stops after it hits the end of the log.
  - **On**: Loop Searching is enabled, that is, the search starts over from the first line after it hits the end of the log.
- **Default Text Wrap Mode**: Use this option to specify whether the text appearing in the SmartLog window is wrapped by default.
  - **Off**: Text Wrap mode is disabled by default, that is, the text displayed in the SmartLog window is not wrapped in accordance with the window width.
  - **On**: Text Wrap mode is enabled by default, that is, the text displayed in the SmartLog window is wrapped to fit the window width.

Rule Page

Use this page to change the rule settings associated with the *SmartLog* window.

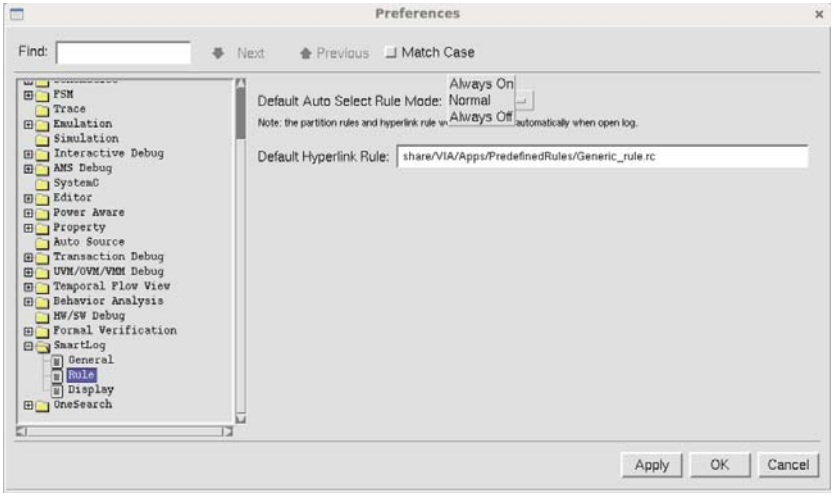


Figure: Rule Page

The following options are available in the **Rule** page:

- **Default Auto Select Rule Mode:** Use this option to specify the auto select rule mode.

  The following options are available in the *Auto Select Rule Mode* field:

  - **Always On**: Enables auto-select rule.
  - **Normal**: Enables auto-select rule.
  - The **Normal** mode changes to the **Always Off** mode, if any partition rule files or hyperlink rule file is added or removed in the *Open Log* form and if the **Open** button is clicked in the *Open Log* form.
  - **Always Off**: Disables auto select rule.
- **Default Hyperlink Rule:** Use this option to specify the hyperlink rule that can be used as a default rule for SmartLog.

Display Page

Use this page to specify the settings associated with severity.

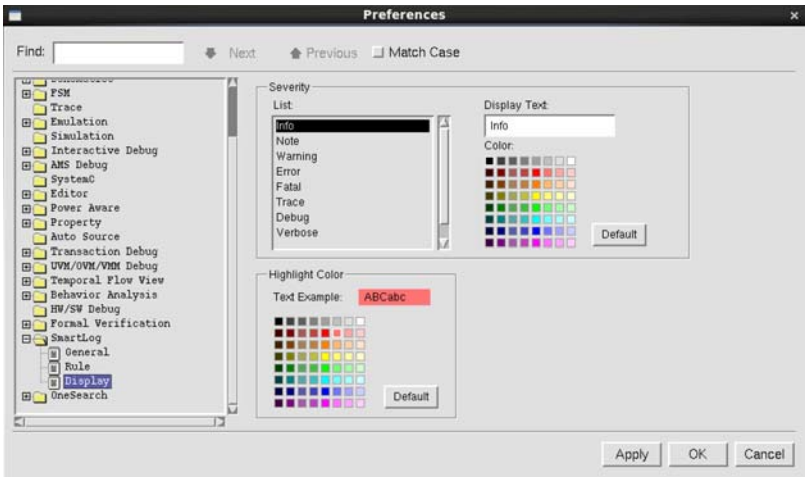


Figure: Severity Page

You can use this page to specify *Display Text* and *Color* for the severity column in the *Structure* view of SmartLog window.

To specify *Display Text* and *Color* for each severity,

1. Select a severity from the *Severity List* field.
2. Change the display text in the *Display Text* field.
3. Change the color in the *Color* field.
4. Click on **Apply** or **OK** to save the settings.

OneSearch Folder

These preferences are applicable for the *OneSearch* framework.

General Page

Use this page to specify the default settings associated with *OneSearch*.

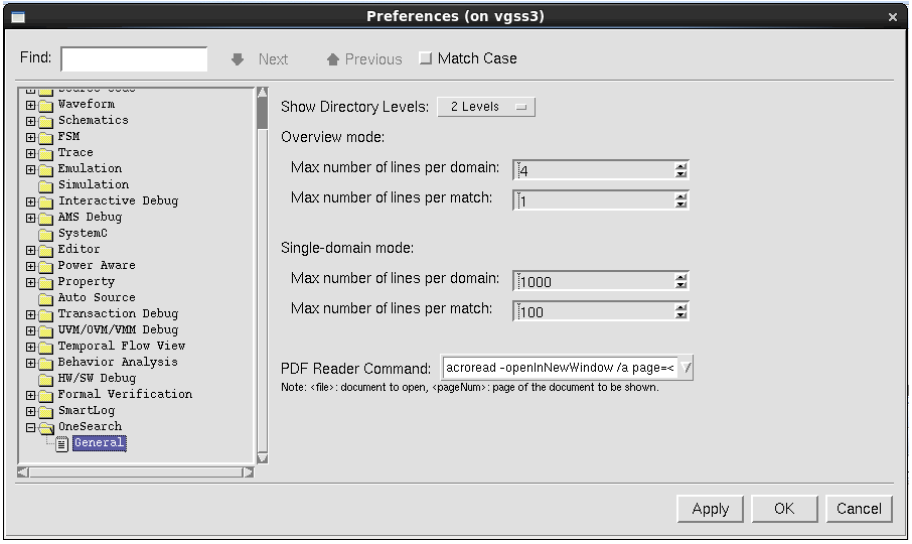
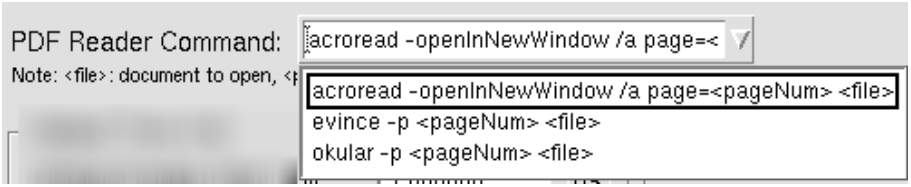


Figure: OneSearch General Page

The following options are available in the **General** page:

- **Show Directory Levels:** This option specifies the path and the directory levels to be displayed for each of the search result. You can select one of the available sub-options: **0 Level**, **1 Level**, **2 Levels**, **3 Levels**, **4 Levels**, **5 Levels**, **Full Paths**. It's default value is **2 Levels**.
- **Overview mode**
  - **Max number of lines per domain -** This option specifies the maximum number of lines to be displayed per domain in the Overview mode. It's default value is 4.
  - **Max number of lines per match -** This option specifies the maximum number of lines to be displayed per match in the Overview mode. It's default value is 1.

- **Single-domain mode**
  - **Max number of lines per domain -** This option specifies the maximum number of lines to be displayed per domain in the Single-domain mode. It's default value is 1000.
  - **Max number of lines per match -** This option specifies the maximum number of lines to be displayed per match in the Single-domain mode. It's default value is 100.
- **PDF Reader Command**: This option specifies the PDF Reader command to view hyperlinked results. You can either select a pre-defined PDF Reader command or enter the preferred PDF Reader command. This setting is saved in the `novas.rc` file. The available pre-defined PDF Reader commands are illustrated in the following figure:



Utilities

Overview

This chapter explains the following utilities:

- `verdi`, `novas`, `siloti`: Commands to invoke the Verdi GUI.
- `nWave`: A command to invoke the *nWave* GUI.
- `setupCer`: A utility to create Certitude setup files.
- Analysis
  - `assertEval`: A batch mode command to evaluate assertions.
  - `esa`: A command to perform Essential Signal Analysis.
  - `nAnalyzer`: A batch mode command for clock analysis and extraction.
  - `nClockTree`: A batch mode command for clock tree extraction.
  - `nCompare`: A batch mode command to invoke the *nCompare* module.
- Compilation
  - `aliasextract`: A utility that creates an alias file for a precompiled mixed language design.
  - `bacom`: A utility to save the results of Behavior Analysis to the Behavior Database (BDB).
  - `crdb`: A command to perform correlation.
  - `elabcom`: A utility that generates the elaboration database.
  - `gencom`: A utility that is used to automatically extract the file list (or command list) from the make file.
  - `libls`: A utility to list the contents of the library.
  - `nc2novasrc`: A utility to convert the library setting of the NC simulator, `cds.lib`, and `hdl.var` to the `novas.rc` resource file.
  - `nfixhier`: A utility to fix the resolving libraries for the hierarchy of the specified cell.
  - `netlistcom`: A utility that performs RTL extraction in the batch mode.
  - `nrun`: A batch mode compiler for mixed design files.
  - `spicom`: A batch mode compiler for SPICE.
  - `vericom`: A batch mode compiler for Verilog (for example, Verilog or SystemVerilog) designs.

- vhdlcom: A batch mode compiler for VHDL.
- vlmap: A function that automatically updates library mapping items to the novas.rc resource file.

- Conversion
  - esdb2txt: A utility to generate a plain text Essential Signal list from the Essential Signal Database (ESDB) for emulation flow.
  - esdbdebug: A utility to dump information in the Essential Signal Database (ESDB) for debugging.
  - fsdb2ns: A utility that translates an EPIC FSDB file into an ASCII format and converts a digital type FSDB file into the Piecewise-Linear (PWL) format.
  - fsdb2saif: A utility that converts an FSDB file into backward SAIF format.
  - fsdb2vcd: A utility that translates an FSDB file into VCD.
  - fsdbdebug: A utility for dumping the FSDB information and generating a log file for debugging.
  - fsdbedit: A utility for modifying scope hierarchies in the FSDB file.
  - fsdbexpand: A utility for processing an existing Essential Signal FSDB file.
  - fsdbextract: A utility for post processing an existing FSDB file.
  - fsdbmangle: A utility that mangles signal names in the FSDB file.
  - fsdbmerge: A utility that merges several FSDB files into a single file.
  - fsdbrecover: A utility that recovers unfinished FSDB files.
  - fsdbreport: A utility that reports time sequence, address values, and instruction values of a simulation.
  - merge_multi_swp_fsdb: A utility to merge multiple sweep analog FSDB files into a single FSDB file.
  - nce2report: A utility to generate a readable waveform comparison report file from the .nce file.
  - txt2fsdb: A utility for transferring transactions from a text file to an FSDB file.
  - txt2uddb: A utility for transferring transactions or messages from a text file to a UDDB file.
  - vcd2fsdb: A conversion utility that is used to convert VCD files to FSDB files.
  - vfast: A conversion utility that is used to convert VCD files to FSDB files.

    - xloc: A utility to translate signals with X value in an FSDB file into an index file.
- General
    - novas_plat: A program to obtain the platform information that reflects the platform used when the Verdi platform is invoked.
- Symbol Libraries
    - edif2SymDB: A utility that translates EDIF symbol library files into the symbol library.
    - map2SymDB: A utility that translates a map file into the symbol library.
    - symDB2map: A utility that translates the symbol library into a map file.
    - syn2SymDB: A utility that translates library files into a symbol library for the Verdi platform.
- Timing
    - AMBIT.pl: A utility to support conversion of Ambit static timing reports to the XML format.
    - astro2Xml.pl: A utility to support conversion of Astro static timing reports to the XML format.
    - crmap.pl: A perl script to partition a gate-level design based on module hierarchies (that is, a subscope under a large design can become a partition).
    - DC.pl: A utility to support conversion of DC static timing reports to the XML format.
    - EinsTimer.pl: A utility to support conversion of EinsTimer static timing reports to the XML format.
    - HAL2Xml.pl: A utility to support conversion of HAL2Xml static timing reports to the XML format.
    - magma.pl: A utility to support conversion of Magma static timing reports to the XML format.
    - pearl.pl: A utility to support conversion of Pearl static timing reports to the XML format.
    - pt2Xml.pl: A perl script to translate PrimeTime timing reports into the XML format.
    - RTLC2Xml.pl: A utility to support conversion of RTL compiler timing reports to the XML format.
    - SE2Xml.pl: A utility to support conversion of SOC Encounter timing reports to the XML format.
    - sdfin: A batch mode compiler for SDF files.

verdi

This command invokes the Verdi platform.

NOTE:
1) The library settings are loaded from the `novas.rc` resource file unless the `manage.rc` resource file is specified. For details on using the `manage.rc` resource file, see *Appendix B: Customizing Verdi*.
2) Loading gzipped (`*.gz`) files for Verilog designs is supported.
3)The `-ssy`, `-ssz`, and `-ssv` options must not be used if the compiled gate-level design is used with the *nECO* module for ECO modification.

Usage:

```
verdi [General Options] [nTrace Options] [Simulator Options]
[Siloti Options] [Environment Options] [Behavior Analysis
Options] [Power Manager Options]
```

General/nTrace Options

Options	Explanation
-2000	Supports VHDL IEEE 1076-2000 standard.
-2001/+v2k	Supports Verilog IEEE 1364-2001 standard.
-2001genblk	Uses the Verilog IEEE 1364-2001 naming style to generate blocks (overrides other language options). IEEE1364-2005 sec12.4.3 unnamed genblk naming is not applied. **NOTE:** This option only works with `-sv` or `-2005`, and it is for VCS users only (VCS does not support IEEE1364-2005 sec12.4.3 unnamed genblk naming through version Y-2006.06-11).
-2005	Supports Verilog IEEE 1364-2005 standard. IEEE1364-2005 sec12.4.3 unnamed genblk naming is applied by default.
-2009	Supports SystemVerilog IEEE 1800-2009 standard.
-2012	Supports SystemVerilog IEEE 1800-2012 standard.
-87	Supports VHDL IEEE 1076-1987 standard.
-93	Supports VHDL IEEE 1076-1993 standard.

Options	Explanation
-aliasIgnoreHier	When specified, apply alias settings to signals with the same signal name regardless of differences in the hierarchy. This option is only used with *-aliasFile*. For example, *verdi -f run.f -aliasFile signal.alias -aliasIgnoreHier* -------signal.alias------------ FSM_snum 1 FSMsignal system.i_cpu.C1 FSM_anum 2 FSMalias READ 1'b0 FSMalias WRITE 1'b1 FSMname The alias rule is applied to all signals with the signal name "C1".
-ams	When this option is specified on the Verdi command line, HSPICE constructs are recognized. This means the content between .subckt and .ends are treated as a comment and a node is added to the design browser to bind the instance.
-assert checker|svaext|svvunit	Specifies the additional syntax to support. checker: Supports the checker construct of SystemVerilog IEEE 1800-2009 standard. svaext: Supports SystemVerilog Assertion features compliant to IEEE 1800-2009 standard. svvunit: Supports the PSL vunit syntax. Files with the **.psl* file extension is treated as PSL files.
-autoalias | -aliasFile *aliasFile*	Specifies -autoalias to automatically extract the aliases, for example `verdi -dbdir simv.daidir -autoalias` or use `-aliasFile` to specify an alias file. The alias file can be automatically created as follows: 1. *vericom -f run.f -lib work* 2. *aliasextract -lib work -output extracted.src_alias* 3. *verdi -top system -lib work -aliasFile extracted.src_alias* **NOTE:** If multiple alias files need to be loaded, concatenate them into one alias file and then load it using the -aliasFile option.
-autoendcelldef	Automatically appends '*endcelldefine* at the end of a file if a matching '*endcelldefine* cannot be found for a declared '*celldefine* in the file.
-batch	Runs in the batch mode. **NOTE**: The *DISPLAY* environment variable is not required. The Verdi platform exits automatically in thirty seconds if no more Tcl commands are executed in this mode.

Options	Explanation
-bdb_load_scope "*pathName*"	Specifies the path to load the top module of the Behavior Database (BDB) file. This option only works when the *-bdb_load* option is specified. **Example**: *> verdi -lib work -top system -bdb_load work.lib++/work.bdb -bdb_load_scope "system.i_cpu"*
-chmod *numeric_mode*	Specifies the access attribute for generated files and directories. For example: *verdi 1.v -chmod 777* Changes the log directory (default is *vericomLog*) and the library directory (default is *work.lib++*) to have full read/write/executable (that is, drwxrwxrwx) access.
-comment_ transoff_regions -suboption | +suboption	Skips the source code between translate_off/on pragmas. The suboptions are vendor names (for example, cadence, ikos, mentor, novas, pragma, quickturn, synopsys, synthesis). To skip a single region, use *-suboption*. To skip multiple regions, use *+suboption1+suboption2...* **Example:** source code: test.sv `#1 module top;` `#2 ......` `#3 //mentor translate_off` `#4 a = i;` `#5 //mentor translate_on` `#6 //quickturn translate_off` `#7 a = i;` `#8 //quickturn translate_on` `#9 ......` `#10 endmodule` 1. Skips a single region *verdi -comment_transoff_regions -mentor test.sv* "a=i" on line #4 is skipped. 2. Skips multiple regions *verdi -comment_transoff_regions+mentor+quickturn test.sv* "a=i"on line #4 and "a=i" on line #7 are skipped.
-cuname *compilation-unit name*	Supports compilation-based compilation-unit mode with the specified name. The global space is modeled as a package named *compilation-unit name*. It is disabled when the name is specified as *-cunit*.
-cunit	Supports compilation-based compilation-unit mode with the default name. The global space is modeled as a package named *novas_cunit_n*.
-disable_ams_ default_disciplines	Disables the defaults provided by the AMS standard discipline definition.

Options	Explanation
+disable_message+<*message_serial_numbers|error|warning*>	Suppresses the specified message(s), all error messages, or all warning messages. Use a plus (+) symbol for different message type combinations. For details about each message number and its description, see *Appendix F: Message Table*. **Examples:** Suppresses the specified message(s) *+disable_message+C00288* *+disable_message+C00288+C00266* Suppresses the specified message and all warning messages *+disable_message+C00288+warning* Suppresses the specified message, all error messages and warning messages *+disable_message+C00288+error+warning*
-doc	Starts PDF Reader to display the bookshelf file for the Verdi documentation.
-dynaconfig *filename*	Specifies a VCS runtime configuration file to load. The supported format of this configuration file includes the full path of the replaced instance and module names of the replaced instance. For example: *top.instance_A.instance_B* *ModuleA* *ModuleB* **NOTE:** The *-dynaconfig* option works only with Verilog designs.
-elab <path>	Specifies the path of Verdi elaborated KDB.
-error=no<Error_ID>, ...	Reports the specified error messages as warning messages. The following error type is supported: MPD: Module <module_name> redefined
-ecoScript *scriptFile*	Executes an ECO script file in the batch mode.
-encode EUCJP	Displays Japanese font in the source code frame.
-errorstop *redefined_module*	Stops parsing the remaining files when there is a redefined module.
-exactcolor	Uses a private colormap.

Options	Explanation
-extinclude	Compiles the included files with the version specified by their extension.When this option is not specified and a source file for one version of Verilog contains the `include compiler directive, *vericom* compiles the included file for the same version of Verilog by default, even if the included file has a different filename extension. **Example:** *<a.v>* *`include "b.sv"* *module a();* *endmodule* *<b.sv>* *module b();* *logic v1;* *endmodule* *> vericom a.v +systemverilogext+.sv -extinclude* "b.sv" is parsed with the SystemVerilog keyword set, syntax, and semantics.
-extractsamenet	Recognizes the existence of the same net at other points in the hierarchy. When this option is specified along with *-ssf*, the Verdi platform extracts the same net after the FSDB file is loaded. When this option is specified without *-ssf*, the default status of the **Extract Same Net** option in *nTrace* or *nWave* is checked.
-F *filename*	Loads an ASCII file containing a specified path to source files and simulator options. A relative path is used to specify the source files. The -F and -path options is used simultaneously. The path assigned with the -path option is ignored. **NOTE:** This option is for Verilog only.
-fixCellHier	Resolves the specified hierarchy in the library when importing the design.
-DEFPARAM <parameter_path>= <value>	Changes the specified parameter to the specified value.
-fastGate	Runs in the fast gate-level debug mode. This mode provides faster searching and tracing in large gate-level designs. For details, see the *Appendix G: Gate-level Debug Mode* chapter.

Options	Explanation
-gvalue+<*parameter_hierarchical_name*>=<*value*>	Specifies a VHDL parameter file to be provided at runtime in order to overwrite the generic value. **Example:** *> -gvalue+l=200 -->all parameter name "l" is updated to 200* *-gvalue+m=200+l=200 -->all parameter name "l" or "m" is updated to 200* *-gvalue+top.a.b.m=200+l=200 -->all parameter name "l" or the parameter "top.a.b.m" is updated to 200*
-G<paramName>=<value>	Overrides the parameter value in the design.
-guiConf *filename*	Specifies the configuration file name to save and restore the layout. If not specified, the *novas.conf* configuration file in the working directory is used. For details, see the *novas.conf* section in *Appendix B*.
-gv|-gvalue generic=value	Overrides the generic value defined in the source code with the value specified in the command line. It is used in both Verdi and elabcom utilities. **NOTE:** The *-gv|-gvalue* option overrides the generic value defined in the source code only if the generic is of type integer or real. **Example:** *> verdi -top work.top -gvalue /TOP/LEN=1*
-h | -help	Prints the help message.
-id	Returns useful information, such as Verdi version and build date, your work station name, platform, and host ID (used in licensing).
-ignore *keyword_argument*	Suppresses error messages associated with the specified keyword argument. The following keyword is supported: *driver_checks* Suppresses error messages about Verilog driver checking.
-ignore_macro_redef	Suppresses warning messages for redefined macros.
-ignorekwd_config	Ignores the "config" keyword of Verilog IEEE 1364-2001.

Options	Explanation
-impConf *confFile*	Specifies a configuration file for the imported design. Partial load and excluded instances is only used with import from library; undefined modules and virtual top are used with both import from library and import from file. **NOTE:** If used with *-vtop*, *-vtop* is ignored. Configuration File Format A configuration file may contain four types of information including *Undefined_modules*, *Virtual_top*, *Partial_load*, and *Exclude*. 1. [*Undefined_modules*]: Suppresses warning messages of undefined modules. **Example:** *[Undefined_modules]* *cpu* *alu* 2. [*Virtual_top*]: Specifies virtual top mapping information. **Example**: *[Virtual_top]* *system = vaTop.sys* *CPU = vbTop.t_cpu* 3. [*Partial_load*]: Specifies tree information to execute partial load. **Example**: If the hierarchy of a design looks like: *- L1c - L2c - L3c - L4c* *Top - L1a - L2a - L3a - L4a* *- L1b - L2b - L3b - L4b* then, to load the design excluding *- L1b - L2b - L3b - L4b* into the Verdi platform, the partial load section in a configuration file is edited as follows: *[Partial_load]* *Top.L1a.L2a.L3a.L4a* *Top.L1a.L1c* **NOTE:** When using partial load, the design is only imported from the library. **NOTE:** All instantiations under L1c is also loaded. **NOTE:** If both *Virtual_top* and *Partial_load* sections are in a configuration file, the *Partial_load* section is ignored and a warning message is displayed. 4. [*Exclude*]: Specifies excluded modules or scopes. **Example:** *[Exclude]* *CCU* Excludes module instances instantiated by CCU. *[Exclude]* *system.i_CCU.i_mprom* Exclude *system.i_CCU.i_mprom* itself and below.

Options	Explanation
	NOTE: 1) When excluding instances, the design is only imported from the library. 2) If a scope is specified in both partial load and exclude sections, the scope is not loaded by exclude (exclude is prior to partial load).
-L/-Lf *libName*	Specifies one or more Verdi libraries compiled with *vericom* or *vhdlcom* that do not contain the top view. The *-L/-Lf* option is specified for each compiled library. This option is used with the -top and -lib option. For example: *verdi -top myTop -lib myLib -L libA -L libB -L libC* where *-top myTop -lib myLib* is used to specify the top view and *-L libA -L libB -L libC* is used to specify the remaining referenced libraries. **NOTE:** 1) As to search sequence, the *-Lf* option has higher priority than the *-L* option. 2) The *.lib++* extension is used when specifying the logical library with the *-L* option on the Verdi command line (that is, *–L libname* and *–L libname.lib++* are equivalent).
-lca	Enables Limited Customer Availability (LCA) features in Verdi.
-lib *libName*	Specifies the Verdi library compiled with the *vericom* or *vhdlcom* utility that contains the top view. The logical library name and physical library path mapping are specified in the *[Library]* section of the *novas.rc* resource file. This option needs to be used with *-top*. For example: *verdi -top myTop -lib libA* **NOTE:** If only one of *-lib* or *-top* are specified, the following default configuration is used: - default library = "work" - default top is automatically determined by the Verilog parser (*vericom*)
-libcellfile *filename*	Specifies the file which contains module/entity as the library cell. The module/entity is taken as the library cell if they are specified in this file. For the file format, each module/entity name must be separated by the space, tab, or enter characters.
-liblist *listNames*	Searches the specified libraries compiled by VCS. The plus sign (+) is used to include multiple libraries in the library list. **NOTE:** The libraries are generated from VC/VCS Native Compile and is defined in the *synopsys_sim.setup* file. **Example:** *> verdi -top top -liblist L1+L2+L3*

Options	Explanation
-liblist_work	Starts searching from the parent library first. The *-liblist_work* option has higher search priority than the *-liblist* option. **Example:** *<top.sv>* *module top;* *M m();* *endmodule* *module M;* *endmodule* *<M.sv>* *module M;* *endmodule* *> vericom -lib top -sv top.sv* The top.sv file is compiled as top.lib++. *> vericom -lib L1 -sv M.sv* The M.sv file is compiled as L1.lib++. *> Verdi -top top -lib top -liblist L1* Searching *top.m* finds *module M* in *L1.lib++*. *> Verdi -top top -lib top -liblist L1 -liblist_work* Searching *top.m* finds *module M* in *top.lib++*. The *-liblist* option is ignored.
-licdebug	Dumps license diagnostic information to the console. It is used for license debugging.
-licverdi	Checks out the Verdi license and enables Verdi features when no other licenses are available.
-load_trace_report	Opens the *SmartLog* and automatically displays the report file provided for `tracediff` and `traceX` utilities. This option should not be used with `-ssf` option. If the option is used then a warning message is generated.
-logdir *logDirectory*	Specifies the location of the log directory.
-logfile *logFile*	Specifies the location of the log directory/file.
-managercFile *path*	Specifies the full path of the *manage.rc* resource file.
-max_mtr_count *value*	Specifies the maximum thread count when running the multiple thread flow for the FSDB file access. When the value is set to 1, the multiple thread flow is disabled. Its valid values are 0 to 16.

Options	Explanation
-mdt *MDTfile*	Loads the Memory Definition Table (MDT) file or an MDT file list. The MDT file list format is one memory definition file per line. For example, the *mem.lst* file contains *mem1.def* and *<path>/mem2.def*.
-mixnc *simulator*	Specifies the NC-Sim simulation executable and enables the interactive mode.
-mti *simulator*	Specifies the MTI simulation executable and enables the interactive mode. **Example:** *verdi -vhdl -f run.f -top -mti "vsim -do stop.do"* The Verdi platform enters the interactive mode directly with the simulation command, *vsim -do stop.do*. **NOTE:** Use double quotes if more than one command line option is added.
-nc *simulator*	Specifies the NC-Verilog simulation executable and enables the interactive mode.
-nclib	Specifies whether to elaborate the precompiled design based on the Cadence NC library *-binding* scheme. For details, see the *-nclib* option in *vericom*.
-netlistcom	Translates design libraries into more specific netlist views to speed up the invocation of schematics in *nSchema*. For details, see *netlistcom*.
-nogui	Runs in the batch mode. **NOTE:** A valid *DISPLAY* environment variable is required. The Verdi platform exits automatically in thirty seconds if no more Tcl commands are executed in this mode.
-noinc	Suppresses incremental loading.
-nologo	Suppresses the *Welcome* page at startup and default to the hardware debug layout. The *Welcome* page is displayed by invoking the **Help -> Welcome** command.
-novasLibPaths	Specifies the file containing the paths of all symbol libraries.

Options	Explanation
-novasLibs *filename(s)*| *libName(s)*	Specifies one or more symbol library names or files containing the names of all symbol libraries. In the following examples, the first example specifies a symbol library name and the second example specifies a file name: *-novasLibs Lsi10k_u* *-novasLibs libsetfileOne* Multiple Symbol Libraries When multiple symbol library names are specified, library names should be separated by a space and enclosed with double quotes as shown in the following example: *-novasLibs "Lsi10k_u Default_u"* Multiple Files When multiple files containing the symbol library names are specified, the file names should be separated by a space and enclosed with double quotes as shown in the following example: *-novasLibs "libsetfileOne libsetfileTwo"* Mix of Symbol Libraries and Files When a mix of symbol library names and file names are specified, the names are separated by a space and enclosed with double quotes as shown in the following example: *-novasLibs "lsi10k_u libfilesetOne"* *===File: libsetfileOne===* *lsi10l_u* *default_u* *===File: libsetfileTwo===* *lsi10k_u* In the file containing the list of symbol library names, each symbol library file is specified on a new line. **Example:** *liba* *libb*

Options	Explanation
-novasLibPaths *filename(s)| pathName(s)*	Specifies one or more symbol library paths or files containing the paths of all symbol libraries. In the following examples, the first example specifies a symbol library path and the second example specifies a file name: *-novasLibPaths /slowfs/sp_dq2/qa/symlib/syn2SymDB/64_bit* *-novasLibPaths libpathOne* Multiple Symbol Library Paths When multiple symbol library paths are specified, path names are separated by a space and enclosed with double quotes as shown in the following example: *-novasLibPaths "./ /slowfs/sp_dq2/qa/symlib/syn2SymDB/ 64_bit"* Multiple Files When multiple file names containing the symbol library paths are specified, the file names are separated by a space and enclosed with double quotes as shown in the following example: *-novasLibPaths "libpathOne libpathTwo"* Mix of Symbol Libraries and Files When a mix of symbol library paths and file names are specified, the paths and file names are separated by a space and enclosed with double quotes as shown in the following example: *-novasLibPaths "libpathsetfileOne /slowfs/sp_dq2/qa/symlib/ syn2SymDB/64_bit"* *===File: libpathOne===* *./* */slowfs/sp_dq2/qa/symlib/syn2SymDB/* *===File: libpathTwo===* */slowfs/sp_dq2/qa/symlib/syn2SymDB/64/* In the file containing the list of symbol library paths, each symbol library path is specified on a new line. **Example:** */home/mylib* */home/proj/lib*

Options	Explanation
+optconfigfile+<*cell_configuration_file*>	Specifies the cell information configuration file that describes which module to potentially replace with which module. The format of the cell configuration file is replace module <*original*> with module <*substitute*>. **Example:** replace module *cpu* with module *cpu_subst* **NOTE:** This option does not work on loading the elaboration database. Specify this option in the *elabcom* utility when loading the elaboration database.
-parameters *filename*	Specifies the file containing a list of parameters to be changed to values. The file format is *assign <value> <parameter_path>*. **Example:** *assign 100 l* *assign 200 top.a.b.m*
-path *pathName*	Specifies the path for import designs.
+pkgdir+<*pkg_name* | *pkg_full_path*>	Loads the specified DesignWare VIP packages as black boxes by referring to the headers located in the release package or in an absolute path. **Examples:** *> verdi +pkgdir+amba <design> ...* Compiles the *amba.f* package located in the default product directory. The default product directory is *<$VERDI_HOME>/etc/kdb/verilog/dwvip/AMBA_PKG*. *> verdi +pkgdir+/home/mydwvip/packages/amba <design> ...* Compiles all **.f* files in */home/mydwvip/packages/amba*.
-play *filename*	Plays the specified command history file. The command history file can be generated by specifying the *-recordhlc* option.
-preTitle *prefixTitleName*	Specifies the user-defined title to add as a prefix for all window banners.
-pvalue+<*parameter_hierarchical_name*>=<*value*>	Change the specified parameter to the specified value.
-rcFile *rcFilename*	Specifies the Verdi resource file. If the resource file exists, settings are loaded and changes are saved to this file. If this option is not specified, the default settings are applied and changes are saved to *novas.rc* under the working directory. For details, see the *novas.rc* section in *Appendix B*.

Options	Explanation
-q	Turns on the quiet mode to suppress the console information of copyright, version, and locations of the log directory, resource file, and GUI configuration file.
-realport	Supports only the "wreal" keyword in Verilog-AMS. This option also supports the "`wrealXState" or "`wrealZState" macros for the real x or z states respectively.
-recordhlc *filename*	Records all high-level commands (command language commands) to a file for playback.
-regf *registerFile*	Loads register restore file (**.register*). Note that an FSDB file must be loaded as well. **Example:** *verdi -regf regFileName -ssf testcase.fsdb*
-RegPort	When specified, a *reg* declaration before the output declaration for the same signal does not generate an error. **Example:** *module test(alpha,reset_,clk);* *input reset_;* *input clk;* *reg [4:0] alpha;* *output [4:0] alpha;* *reg [4:0] beta;* *always@(posedge clk or negedge reset_) begin* *if(!reset_) alpha <= 5'b0000;* *else alpha <= beta;* *end* *endmodule*
-rmkeyword *keyword*	Downgrades the specified keyword to an identifier.
+rmkeyword+ *<keyword(s)>*	Downgrades the specified keywords to identifiers. One or more keywords may be specified.
-sdf delay_type::full hierarchy instance name::sdfFile	Loads an SDF (**.sdf*) and supports the full hierarchy path for an sdf. The options for delay type are as follows: *Min*: Minimum Delay *Typ*: Typical Delay *Max*: Maximum Delay For example, `-sdf Max:system.i_cpu:cpu.sdf` **NOTE:** The design is imported from `simv.dadir` directory.

Options	Explanation
-sdf *type*:instname:*sdfFile*	Loads an SDF (**.sdf*) and uses the specified delay type for SDF annotation. The options for delay type are as follows: *Min*: Minimum Delay *Typ*: Typical Delay *Max*: Maximum Delay For example, `-sdf Max::system.sdf` **NOTE:** Instance name can be empty.
-skip_auto_sdf	Specify this option to skip automatic importing of SDF in the UFE flow. **NOTE:** The option impacts only the automatic loading in the UFE flow. The manual loading of SDF is not impacted. For example, `% verdi -dbdir simv.daidir -skip_auto_sdf` The SDF file (**.sdf)* is not imported. You can later load the FSDB (**.fsdb*) file manually from the GUI. `% verdi -dbdir simv.daidir -skip_auto_sdf -sdf 1.sdf` The `1.sdf` file is loaded. The FSDB (**.fsdb*) file specified in `simv.daidir` is not imported. `% verdi -lib work -skip_auto_sdf -sdf 1.sdf` The `-skip_auto_sdf` option have no impact in the non-UFE flow.
-showtrigger	Shows red trigger line for loop marker.
-simdir *path*	Specifies the path where the VCS executable (vcs) is executed. **NOTE:** If the *synopsys_sim.setup* file specified with the *SYNOPSYS_SIM_SETUP* environment variable exists, this option is ignored. **NOTE:** Use this option with the *-simflow* option.

Options	Explanation
-simflow	Loads the Verdi Knowledge Database (KDB) generated by VCS and uses the library mapping from the *synopsys_sim.setup* file. **NOTE:** Verdi looks for the *synopsys_sim.setup* file in the following directories in the following sequence: 1. The directory specified in the *SYNOPSYS_SIM_SETUP* environment variable 2. The directory and file specified in the *-simDir* command line option 3. The working directory from which Verdi platform is invoked (*./synopsys_sim.setup*) 4. The user home directory (*<HOME>/synopsys_sim.setup*) 5. The directory specified in the *VCS_HOME* environment variable **NOTE:** Before specifying this option, set the *VCS_HOME* environment variable to point to the VCS installation.
-smlog *logFile*	Loads the specified log file with Smart Log.
-smlog_hyper | -smlog_h *ruleFile*	Specifies the hyperlink rule file for the specified log. **NOTE:** Use this option with the *-smlog* option.
-smlog_partition | -smlog_p "ruleFile1 ruleFile2...ruleFileN"	Specifies the partitioning rule file(s) for the specified log. A pair of double quotes (") must be used to enclose the partitioning rule file(s). **NOTE:** Use this option with the *-smlog* option.
-ssc *licenseFile*	Specifies the license file name.
-ssdf *sdfFile*	Loads SDF (**.sdf*) or DDB (**.ddb*) file. The DDB file format is created by the Verdi platform from the SDF file using the *sdfin* utility.

Options	Explanation
-ssf *fastFile(s)*| *dumpFile(s)*| *fastFile list(s)*	Loads FSDB (*.*fsdb*), virtual FSDB (*.*vf*), gzipped FSDB (*.*fsdb.gz*), bzip2 FSDB (*.*fsdb.bz2*), waveform dump (*.*vcd*, *.*vcd.gz*) files, or FSDB file lists (*.*flst*). For individual FSDB files, 1 - 16 files may be specified at a time. **Example:** *verdi -f run.f -ssf file1.fsdb file2.fsdb... file16.fsdb* or *verdi -nWave -ssf demo1.fsdb demo2.fsdb demo3.fsdb* For a file list, each FSDB or virtual file must be listed on a separate line with the full or related path to the file. Unlimited files are supported. A pound (#) sign and a semicolon (;) character are used for comments, but they must be on separate lines. For example: *### Normal FSDB file with full path* */home/proj/mod/file1.fsdb* */proj/simrun/file2.fsdb* *;; Normal FSDB file with related path* *../mod2/file3.fsdb* *./file4.fsdb* *## Virtual FSDB file works as well* *virtual1.vf* */home/proj/mod/virtual2.vf* *####* *./filen.fsdb ## Comments after path are not allowed* **NOTE:** When using *-ssf* to specify an FSDB file and the simulator type is specified in the FSDB, the simulator type is set automatically.
-ssr *sessionFile*	Loads the session file (*.*ses*).
-ssv	It is not recommended to automatically tag library modules in the library file (*-v*) as library cells. **NOTE:** During import, the specification of *-ssv* on the Verdi command line overwrites the specification of *-v* in each design library precompiled with *vericom*.
-ssy	It is not recommended to automatically tag library modules in the library directory (*-y*) as library cells. **NOTE:** During import, the specification of *-ssy* on the Verdi command line overwrites the specification of *-y* in each design library precompiled with *vericom*.
-ssz	Ignores '*celldefine* compiler directives. **NOTE:** During import, the specification of *-ssz* on the Verdi command line ignores '*celldefine* compiler directives in each design library precompiled with *vericom*.

Options	Explanation
-sswr *restoreFile(s)*	Loads waveform restore file (*.rc).
-sv	Supports SystemVerilog IEEE 1800-2005 standard. IEEE1364-2005 sec12.4.3 unnamed genblk naming is applied by default.
-sv_pragma	Compile the SystemVerilog assertions code that follows the *sv_pragma* keyword in a comment.
-syntaxerrormax <*number*>	Specifies the maximum number of syntax errors to stop parsing. If the syntax errors exceed this number, the parser stops parsing the remaining files.
-tkName *name*	Sends commands to the Verdi platform from a Tk application.
-top *topModule* | "*top1 top2 ... topN"*	Specifies the top modules for the imported design. **NOTE:** Verilog modules are case-sensitive. The top module name can be specified using the Cadence NC simulator command format to elaborate the designated design unit, such as *ncelab [library.]cell[:view]* **Example:** *verdi -nclib -top "novas.top(rtl)*" The Verdi platform elaborates *top* in the library *novas* and view *rtl*. When the library is not specified, search *work.lib++* for the top design unit. **Example:** *verdi -nclib -top top* is same as *verdi -nclib -lib work -top top or verdi -nclib -top work.top* Multiple top modules in the same library are supported. **Example:** *verdi -top "top1 top2 top3"* When multiple top modules from different libraries are required, use the following format: *verdi -top "libA.top1 libB.top2 libC.top3"* The Verdi platform invokes three hierarchy trees for *top1*, *top2*, and *top3*.
-translateOn	Ignores the RTL source code between SYNOPSYS compiler directives "synopsys translate_off" and "synopsys translate_on."
-undockWin	When specified, set the default value for the **Undock Newly Created Windows** preference option to *true*, so new windows are undocked.
-useius	Uses IUS style parsing/elaboration. **NOTE:** When more than one of the *-useius*, *-usemti*, and *-usevcs* options are specified, only the first option specified is recognized; subsequent options are ignored.

Options	Explanation
-usemti	Use the MTI style parsing/elaboration. **NOTE:** When more than one of the *-useius*, *-usemti*, and *-usevcs* options are specified, only the first option specified is recognized; subsequent options are ignored.
-usevcs	Use VCS style parsing/elaboration. **NOTE:** When more than one of the *-useius*, *-usemti*, and *-usevcs* options are specified, only the first option specified is recognized; subsequent options are ignored.
-v95	Supports Verilog IEEE 1364-1995 standard.
-vc	Supports the DirectC syntax.
-vcs *simulator*	Specifies the VCS simulation executable and enables the interactive mode.
-veriSimType *XL|VCS*	Specifies the simulator type for the Verilog source code naming convention. If this option is not specified, the simulator type follows the *ohSimType* option in the *novas.rc* resource file. If the *novas.rc* resource file does not exist or the type is not specified, the default is *XL*. **NOTE:** This command must be specified before the load design (*-f* or *-lib*) options. **NOTE:** If *-ssf* is used to specify an FSDB file and the simulator type is specified in the FSDB, the simulator type is set automatically and the *-veriSimType* is ignored.
-vhdl08	Supports the VHDL IEEE 1076-2008 standard.
-vhdl | -verilog	Specifies the language for the imported design source. Its default value is *-verilog*. For VHDL designs, specify *-vhdl*. By default, *-verilog* uses the IEEE 1364-2001 standard and *-vhdl* uses the IEEE 1076-1993 standard.
-vhdlSimType *ModelSim|NC*	Specifies the simulator type for the VHDL source code naming convention. If this option is not specified, the simulator type follows the *vhSimType* option in the *novas.rc* resource file. If the *novas.rc* resource file does not exist or the type is not specified, the default is *ModelSim*. **NOTE:** This command must be specified before the load design (*-f* or *-lib*) options. **NOTE:** If *-ssf* is used to specify an FSDB file and the simulator type is specified in the FSDB, the simulator type is set automatically and the *-vhdlSimType* is ignored.
-vhnc *simulator*	Specifies the NC-VHDL simulation executable and enables the interactive mode.

Options	Explanation
-vtop *file*	Specifies a virtual top file for import designs. Instead of importing the entire design, some subsets of the design are imported. To keep the consistency of the design hierarchy between simulation results and imported design files, specify the correct hierarchical names of the imported top modules. The definitions are kept in a *.map* file whose syntax is as follows: *module_name = hierarchical_instance_name* The Verdi platform can generate a virtual top automatically according to the definitions in the *.map* file and import only the subset of the design. **Example:** Assume a VHDL design whose top module is *system* with two instances, *cpu* and *pram*, invoked from *system*. To load subhierarchies, *cpu* and *pram*, into the Verdi platform instead of loading the whole hierarchy *system*, a virtual top file, *vtop.map* is defined as follows: *cpu = system.i_cpu* *pram = system.i_pram* Specifies all the design files obtained in subhierarchy, *cpu* and *pram*, in *vtop.f*. Then the subset of the design is imported with the virtual top *system* by the following command: *verdi -vtop vtop.map -vhdl -f vtop.f* **NOTE:** The specified top module must be in a logical library, (that is, -lib xxx or default library (for example, *work.lib++*)); otherwise, it fails. For example, *verdi -lib libc -L liba -L libb -vtop vtop.map* In the above example, all the modules specified in *vtop.map* are found in *libc*. The *-vtop* option is used to specify a virtual top from the command line without using a *.map* file: **Examples:** Specify one virtual top *verdi -f run.f -vtop "CPU=system.i_cpu"* which is equivalent to "*verdi -f run.f -vtop vtop.file*" where the content of *vtop.file* is: *CPU = system.i_cpu* Specify multiple virtual tops *verdi -f run.f -vtop "CPU=system.i_cpu;ALUB = system.i_ALUB"* which is equivalent to "*verdi -f run.f -vtop vtop.file*" where the content of *vtop.file* is: *CPU = system.i_cpu* *ALUB = system.i_ALUB*

Options	Explanation
-vtop *vhdl_mapfile* -vtopvhdl	Specifies a virtual top file in VHDL for import designs. **Example:** *verdi -vtop vtopvhdl_src.map -vtopvhdl -vhdl 1.vhd* **NOTE:** Use *-vtopvhdl* with *-vtop*.
-wcFile	Supports the wildcard file list in a *run.f* file. Specify this option before *-f*. Note that the /*...*/ comment syntax is not supported in the *run.f* file.
-workMode *[hardwareDebug | testbenchDebug | powerDebug | assertionDebug | transactionDebug| interactiveDebug | user_defined_mode]*	Specifies the work mode or the user-defined mode. This option is ignored if the *-ssr* option is also specified. For details on the different work modes and instructions for saving a user mode, see the *Window Commands* section of the *nTrace* chapter. **NOTE:** This option can be used with the *-i* option.
-wreal *resolution_function*	Supports only the "wreal" keyword in Verilog-AMS. This option also supports the "`wrealXState" or "`wrealZState" macros for the real x or z states respectively.
+ignorefileext+ <*extensionname*>	Specifies the file extension(s) to ignore during compilation. This option applies to both the run file and *verdi* or *vericom* command line. **Examples:** Specify in the run file Assume a run file contains the following: `//1.f` `+ignorefileext+.vr+vg` `1.v` `1.vr` `1.vg` When it is compiled with *verdi* as follows: *verdi -f 1.f* The information printed to the console shows that files with "*.vr*" (*1.vr*) or "*vg*" (*1.vg*) extensions are ignored. Specify on the command line Specify the option on the command line similar to the following: *verdi -f run.f +ignorefileext+.vr+.vg* *verdi -f run.f +ignorefileext+.vr+.vg*
+systemverilogext+ <*extensionname*>	If a consistent file extension is used for SystemVerilog files, this option is used to specify the file extension. Use it in place of the *-sv* option. **Example:** *verdi -f run.f +systemverilogext+.sv+.SV*
+v95ext+ <*extensionname*> | +verilog1995ext+ <*extensionname*>	Specifies the file extension for Verilog-1995 files.

Options	Explanation
+verilog2001ext+ <*extensionname*>	If a consistent file extension is used for Verilog-2001 files, this option is used to specify the file extension. Use it in place of the *-2001* option. **Example:** *verdi -f run.f +verilog2001ext+.v2k*
+spiceExt+ <*extensionname*>	Specifies the file extensions for HSPICE files. This option is not case-sensitive, but the file extension is case-sensitive (for example, *+spiceExt+.sp+.SP*). **Example:** *verdi -f run.f +spiceExt+.sp+.hsp*
-xl *simulator*	Specifies the Verilog-XL simulation executable and enables the interactive mode.
-xprop= tmerge|xmerge| config_file	Specifies the X-propagation format. Its default value is *tmerge*. *tmerge*: Use the tmerge mode. *xmerge*: Use the xmerge mode. *config_file*: Use the user-defined setup file that contains the X-propagation configurations. **Example:** *verdi -f run.f -ssf myfile.fsdb -xprop=xmerge -xproplog "xprop.log"* **NOTE:** It is recommended to specify the same mode (*tmerge* or *xmerge*) as specified when running VCS. **NOTE:** If the *tmerge* and *xmerge* arguments get specified multiple times for *-xprop* (in any order), only the last specification is recognized and takes effect. If the configuration file argument gets specified with any other argument, an error is issued.
-xproplog "*file_list*"	Specifies the X-propagation log file. Multiple log files are supported by enclosing files with double quotes. Loading the *xprop.log.gz* file with this option is also supported, but only a single file is allowed for this compressed file. **NOTE:** This option is only valid after the *-xprop* option is specified. **Example:** *verdi -f run.f -ssf myfile.fsdb -xprop=xmerge -xproplog "xprop1.log xprop2.log"*

Verilog/SystemVerilog Syntax and Semantic Parsing

1. If one or more of the following options are specified, parse the syntax/ semantic of the entire design with the newest language type resolved from the `-sv`, `-2001`, `+v2k`, and `-v95` options.

   For example:

   ```
   > vericom -v95 +v2k -sv 1.v
   ```

   The `1.v` file is parsed with the SystemVerilog language.

2. If none of the options in the first step are specified, parse the syntax/ semantic of the entire design with Verilog 2001.

Verilog/System Verilog Language Keyword Set Checking

1. If any of the `+verilog1995ext`, `+verilog2001ext`, or `+systemverilogext` options are specified, check the files specified by `+{language}ext` against the keyword set of `{language}`.
2. If none of the options above are specified and any of the `-sv`, `-2005`, `-2001`, `+v2k`, or `-v95` options are specified in the command line, check the entire design against the keyword set of the given language options.
3. If none of the options in 1 and 2 are specified, check the entire design against the Verilog-2001 keyword set.

Environment Options

The following environment options are also supported:

Options	Explanation
-envinfo	Dumps environment variable descriptions.
-envmap	Shows the mapping for all environment variables.
-envset	Specifies the environment variables to be read from the command line. Use a plus (+) symbol as the delimiter. The option can be specified multiple times. **Example:** *-envset ENV1=A+ENV2=B* *-envset "ENV3=1 2 3+ENV4=C"*
-rcinfo	Dumps key descriptions of the resource file.
-sdfDelay [Min|Typ|Max]	Specifies the delay type for use with SDF annotation. Its options are: *Min*: Minimum Delay *Typ*: Typical Delay *Max*: Maximum Delay

Simulator Options

All simulator options are accepted by the Verdi platform. The options it needs to compile the design are used and the rest are ignored. For a complete list and details, refer to the documentation for the simulator. The following simulator command line options are also commonly used by the Verdi platform:

Options	Explanation
+define+<*macro*>	The *+define* option is used to specify macros. If the macro is also defined in the source code, it is overridden by this option.
+incdir+<*directoryname*>	Specifies the search path for files used by the *include* statement.
+libext+<*extensionname*>	Specifies the file extensions for Verilog library files. See also the *-y* option.
+liborder	Searches for the module definitions of unresolved module instances in the *verdi* utility with the following order: 1) Searches the remainder of the library where the unresolved module instances are found. 2) Searches through the rest of the libraries. 3) Searches again from the first library. **Example:** *<top.v>* *module top;* *cellA ca();* *endmodule* *<libA.v>* *module cellA;* *endmodule* *<libB.v>* *module cellA;* *endmodule* *> verdi -sv -v libA.v top.v -v libB.v +librescan* "cellA" is resolved in <libA.v> *> verdi -sv -v libA.v top.v -v libB.v +liborder* "cellA" is resolved in <libB.v>
+librescan	Searches for the module definitions of unresolved module instances by always starting from the first library in the library list specified on the Verdi command line. For examples, see the **+liborder** option in the *verdi* utility description. **NOTE:** This option is not case-sensitive.
+libverbose	Prints a message in the *verdiLog/compiler.log* file to indicate the library file where the instance is resolved when a module is instantiated in the source file or library files. **NOTE:** This option does not take effect when importing the design from the library.

Options	Explanation
-f <*filename*>.f	Loads an ASCII file containing design source files and simulator options.
-i	Runs the simulator specified in the *-simType* option. To load waveform, restore a file in the Interactive Simulation Debug mode. Save waveform signals to a resource file (for example, *sig.rc*) first, and then use the *-sswr* and *-i* options together on the Verdi command line to invoke the Interactive Simulation Debug mode. For example: *> verdi -sv -f run.f -sswr sig.rc -i*
-ntb_opts ovm[-<*version*>]	Loads the OVM library for compilation. The *+define+VCS* option is added automatically on the Verdi command line. **NOTE:** To compile the external OVM library, first set the *VCS_HOME* or *VCS_OVM_HOME* environment variable. The *VCS_OVM_HOME* environment variable has higher priority than *VCS_HOME*. If both *VCS_OVM_HOME* and *VCS_HOME* are set, and *ovm.sv* cannot be found in *$VCS_OVM_HOME*, an error is reported. In the following examples, Example 1 is equivalent to Example 2. **Example 1:** *setenv VCS_HOME /simulator/Synopsys/VCS_MX_vH-2013.06/vcs_mx_vH-2013.06* *> verdi -sv -ntb_opts ovm test.v* **Example 2:** *> setenv VCS_HOME /simulator/Synopsys/VCS_MX_vH-2013.06/vcs_mx_vH-2013.06* *> verdi -sv +define+VCS +incdir+$VCS_HOME/etc/ovm/src \ $VCS_HOME/etc/ovm/src/ovm_pkg.sv \ +incdir+$VCS_HOME/etc/ovm/src/vcs \ $VCS_HOME/etc/ovm/src/vcs/ovm_custom_install_vcs_recorder.sv test.v*

Options	Explanation
-ntb_opts <rvm|vmm>[-<*version*>]	Loads the VMM library for compilation. The *+define+VCS+VMM_IN_PACKAGE* option is added automatically on the Verdi command line. **NOTE:** To compile the external VMM library, first set the *VCS_HOME* environment variable. In the following examples, Example 1 is equivalent to Example 2. **Example 1:** *setenv VCS_HOME /tools/simulator/Synopsys/VCS_MX_vH-2013.06/vcs_mx_vH-2013.06* *> verdi -sv -ntb_opts vmm test.v* **Example 2:** *> setenv VCS_HOME /tools/simulator/Synopsys/* *> verdi -sv +define+VCS+VMM_IN_PACKAGE * *+incdir+$VCS_HOME/etc/rvm * *$VCS_HOME/etc/rvm/vmm.sv test.v* For the usage of mixed UVM and RVM libraries, see the Mixed Usage of UVM and RVM Libraries columns.

Options	Explanation
-ntb_opts uvm[-<*version*>]	Loads the UVM library for compilation. The *+define+VCS* option is added automatically on the Verdi command line. **NOTE:** To compile the external UVM library, first set the *VCS_HOME* or *VCS_UVM_HOME* environment variable. The *VCS_UVM_HOME* environment variable has higher priority than *VCS_HOME*. If both *VCS_UVM_HOME* and *VCS_HOME* are set, and *uvm_pkg.sv* cannot be found in *$VCS_UVM_HOME*, an error is reported. In the following examples, Example 1 is equivalent to Example 2. **Example 1:** *setenv VCS_UVM_HOME /$VERDI_HOME/share/vmlib/uvm/uvm<version>/lib/src* *> verdi -sv -ntb_opts uvm test.v* **Example 2:** *> setenv VCS_UVM_HOME /$VERDI_HOME/share/vmlib/uvm/uvm<version>/lib/src* *> verdi -sv +define+VCS +incdir+$VCS_UVM_HOME \ $VCS_UVM_HOME/uvm.sv \ +incdir+$VCS_UVM_HOME/vcs \ $VCS_UVM_HOME/vcs/ uvm_custom_install_vcs_recorder.sv \ +incdir+$VCS_UVM_HOME/verdi \ $VCS_UVM_HOME/verdi/ uvm_custom_install_verdi_recorder.sv test.v* If the *VCS_UVM_HOME* environment variable is not set, the UVM library path in the *VCS_HOME* environment variable is used. The following paths and files are added automatically: *+incdir+$VCS_HOME/etc/uvm[-<version>] \ $VCS_HOME/etc/uvm[-<version>]/uvm.sv \ $VCS_HOME/etc/uvm[-<version>]/vcs \ $VCS_HOME/etc/uvm[-<version>]/vcs/ uvm_custom_install_vcs_recorder.sv \ +incdir+$VCS_HOME/etc/uvm[-<version>/verdi \ $VCS_HOME/etc/uvm[-<version>/verdi/ uvm_custom_install_verdi_recorder.sv* For the usage of mixed UVM and RVM libraries, see the Mixed Usage of UVM and RVM Libraries columns.

Options	Explanation
Mixed Usage of UVM and RVM Libraries: -ntb_opts uvm[-*$version*]+rvm, -ntb_opts rvm+uvm[-*$version*], -ntb_opts uvm[-*$version*] -ntb_opts rvm	When using mixed UVM and RVM libraries, the options in the leading column are equivalent to the following: *+define+VCS* \ *+define+VCS+VMM_IN_PACKAGE* \ *+incdir+<vmm_lib_path>/etc/rvm* \ *<vmm_lib_path>/etc/rvm/vmm.sv* \ *<uvm_lib_path>/src/uvm_vmm_pkg.sv* \ *+incdir+<uvm_lib_path>/src* \ *<uvm_lib_path>/src/uvm.sv* \ *+incdir+<uvm_lib_path>/src/vcs* \ *<uvm_lib_path>/vcs/uvm_customer_install_vcs_recorder.sv* \ *+define+VMM_UVM_INTEROP* To turn *off* the automatic inclusion of *uvm_vmm_pkg.sv*, use *+define+NO_VMM_UVM_INTEROP* in the two-step flow (that is, compile and import design from libraries) and use *+define+NO_UVM_VMM_INTEROP* in the UUM flow. <*uvm_lib_path*> is determined by $*VCS_UVM_HOME* or $*VCS_HOME*. <*vmm_lib_path*> is determined by $*VCS_HOME*. **NOTE:** Specify the *UVM_ON_TOP* and *VMM_ON_TOP* options when using mixed UVM and RVM libraries.
-ovm[-<*version*>]	Load the default Verdi OVM library. If **-ovm** and **-ovmhome** are specified at the same time, **-ovm** is ignored. -<*version*> is used to specify a non default library. All available library versions are located in the *$VERDI_HOME/share/vmlib/ovm* directory. For example, the available ovm version is: *ovm-2.1.2* When only **-ovm** is specified, the default is *ovm-2.1.2*. The **+define+INCA** option is added automatically on the Verdi command line. In the following examples, Example 1 is equivalent to Example 2. **Example 1:** *> verdi -ovm test.v* **Example 2:** *> verdi -sv +define+INCA* \ *+incdir+$VERDI_HOME/share/vmlib/ovm/ovm-2.1.2/lib/src* \ *$VERDI_HOME/share/vmlib/ovm/ovm-2.1.2/lib/src/ovm_pkg.sv test.v*

Options	Explanation
-ovmhome <*path*>	Specifies the OVM installation directory. The *+define+INCA* option is added automatically on the Verdi command line. In the following examples, Example 1 is equivalent to Example 2. **Example 1:** *> verdi -sv -ovmhome $MyOVM test.v* **Example 2:** *> verdi -sv +define+INCA +incdir+$MyOVM/src \ $MyOVM/src/ovm_pkg.sv test.v*
-simBin “<*path*>”	Specifies the path of the simulation binary file. The path must be double quoted. This option can be used without the **-simType** option. If **-simType** is not specified, the default setting in the *novas.rc* resource file is used. **Example:** *verdi -sv f.sv -simBin “/rcc2/integ/SOL2/deb6.1/nvfy/bin/start.simv”*
-simOpt '*option1 option2 ... optionN*'	Specifies simulation options. A pair of single quotes (') must be used to enclose the options. **Example:** *verdi -sv f.sv -simOpt '+vcs+lic+wait+” TEST\” '*
-simType <VCS|ModelSim|NC>	Specifies the simulator (VCS, ModelSim or NC) to use. The simulator is not case-sensitive.

Options	Explanation
-uvm[-<*version*>]	Loads the default Verdi UVM library. If *-uvm* and *-uvmhome* are specified at the same time, *-uvm* is ignored. -<*version*> is used to specify a non-default library. All available library versions are located in the *$VERDI_HOME/share/vmlib/uvm* directory. For example, the available uvm versions are: *uvm-1.0p1* *uvm-1.1d* When only *-uvm* is specified, the default is *uvm-1.1d*. The *+define+INCA* option is added automatically on the Verdi command line. In the following examples, Example 1 is equivalent to Example 2. **Example 1:** *> verdi -uvm test.v* **Example 2:** *> verdi -sv +define+INCA * *+incdir+$VERDI_HOME/share/vmlib/uvm/uvm-1.1d/lib/src * *$VERDI_HOME/share/vmlib/uvm/uvm-1.1d/lib/src/uvm.sv test.v*
-uvmDebug	When this option is specified and VCS is specified as the simulator type, the following VCS runtime options is automatically appended in the Interactive Simulation Debug mode when the Verdi platform starts the *simv* binary: +UVM_CONFIG_DB_TRACE +UVM_PHASE_TRACE +UVM_RESOURCE_DB_TRACE +UVM_OBJECTION_TRACE +UVM_FACTORY_TRACE +UVM_LOG_RECORD +UVM_VERDI_TRACE=UVM_AWARE, For details about the VCS runtime options, see the **UVM Debug** option in the **Simulation** page of the *Preferences* form.

Options	Explanation
-uvmhome <*path*>	Specifies the UVM installation directory. The *+define+INCA* option is added automatically on the Verdi command line. In the following examples, Example 1 is equivalent to Example 2. **Example 1:** *> verdi -sv -uvmhome $MyUVM test.v* **Example 2:** *> verdi -sv +define+INCA +incdir+$MyUVM/src \ $MyUVM/src/uvm.sv test.v*
-v <*filename*>	Modules in the specified file are treated as library cells. This option is overwritten if the *-ssv* option is specified.
-v_no_elab	Library modules in all library files (*-v*) are elaborated. This option is overwritten if the *-ssv* option is specified.

Options	Explanation
-y <*directoryname*>	Modules in the specified directory are treated as library cells. This option is overwritten if the *-ssy* option is specified. In the directory, the library file name is supposed to be the same as the module name. The *+libext* option is also used to check the library file extension and ensure that the library file is matched. **Example:** Assume the following file hierarchy for the current directory: *top.v* *ydir/cpu.v* *ydir/alu.sv* The file names and module names are as follows: <*top.v*> *module top;* *cpu c1();* *alu a1();* *endmodule* <*cpu.v*> *module cpu;* *endmodule* <*alu.sv*> *module alu;* *endmodule* When the following commands are executed on the current directory, the following are the results: *> verdi -sv top.v -y ydir* When *+libext+.v+.sv* is not specified, the following two messages are issued: **Error* view cpu is not defined for instance c1 "top.v", 2:* **Error* view alu is not defined for instance a1 "top.v", 3:* *> verdi -sv top.v -y ydir +libext+.v* **Error* view alu is not defined for instance a1 "top.v", 3:* *> verdi -sv top.v -y ydir +libext+.v+.sv* Error and warning are not issued.
-y_no_elab	Library modules in all library directories (*-y*) are not elaborated. This option is overwritten if the *-ssy* option is specified.

Options	Explanation
-z_no_elab	Cell modules (`celldefine) are not elaborated. This option is overwritten if the *-ssz* option is specified.

Siloti Options

The following Siloti options are also supported. These are optional.

Options	Explanation
-crdb_load *crdbFile*	Specifies the correlation database (CRDB) to load.
-cr_slave "*options*"	Opens a slave process and loads the gate design. Options specified between the quotation marks are passed to the slave process.
-de	Performs Data Expansion setup (auto time window mode) automatically after the design (specify *-f* or *-lib* on the command line or load from the GUI) and simulation results (specify *-ssf* on the command line or load the FSDB file from the GUI) are loaded. Behavior Analysis is automatically executed.
-DE [“*options”*]	After the design (specify *-f* or *-lib* on the command line or load from the GUI) and simulation results (specify *-ssf* on the command line or load the FSDB file from the GUI) are loaded, automatically perform Data Expansion setup based on the specified options. Behavior Analysis is automatically executed. Available options are as follows: *-gr2_mapping file1 ... filen*: List of mapping files. *-force file*: Force value file. *-start_time value / -end_time value*: Start/end time. *-full_time*: Flexible time window (full time range). *-on_demand*: Automatic time window. *-de_enforce*: Enable expansion for a large time window.
-esAuto	Enables automatic Behavior Analysis and/or Data Expansion according to the Essential Signal tag in the loaded FSDB file.
-licsiloti	Checks out the Siloti license first before Verdi or other available licenses.
-min	Iconifies (minimizes) the *nTrace* window after the Verdi and Siloti platform starts.
-noDE	Disables automatic Data Expansion setup.

Behavior Analysis Options

The table below shows Behavior Analysis options that are supported. These options are found in the *Behavior Analysis* folder of the *Preferences* form (invoked with the **Tools -> Preferences** command). These are optional.

Options	Explanation
-apimem *filename*	Specifies a file to define API memory.
-ba	Performs Behavior Analysis immediately after loading the design. By default, incremental Module-Based Behavior Analysis (MBBA) is performed. If *-top* is also specified, only the first top is used; otherwise, Behavior Analysis uses the settings in the *novas.rc* resource file. **NOTE:** If *-ba* is specified with *-bdb_load*, *-ba* is ignored.
-ba_mode [WSBA|MBBA]	Specifies the mode for incremental Behavior Analysis. Following are its modes and they are not case-sensitive: *WSBA*: Performs non-incremental Behavior Analysis in the working scope mode. *MBBA*: Performs incremental Behavior Analysis in the module-based mode. **NOTE:** The Behavior Analysis Tcl commands (that is, the *-incr* option with *tfgGenerate* or *tfgBehaviorAnalysis* Tcl commands) have priority over both the command line and environment variable for the Behavior Analysis mode. That is, if a command file is replayed and a different Behavior Analysis mode is specified on the command line, the Behavior Analysis mode in the command file is used.
-bas "scope(s)" | "*"	Performs Behavior Analysis (with the given scopes as working scopes) immediately after loading the design. For example, *-bas "{scope1} {scope2}"* takes "scope1" and "scope2" as working scopes and *-bas "{*}"* takes every top module as a working scope.
-bboxEmptyModule 0|1	Specifies whether to define the empty modules as black boxes. 0: Does not define the empty modules as black boxes. 1: Defines the empty modules as black boxes.

Options	Explanation
-bboxIgnoreProtected 0|1	Specifies whether to ignore the protected code. 0: Ignores the protected code. 1: Inferences the protected code. Its default value is *1*.
-bboxModule *{moduleName}*	Specifies one or more modules as black boxes. For the regular expressions that support module name searching, see the equivalent *Black Box Setup Form* section in the *Preferences* chapter.
-bboxModuleFile *filename*	Defines the modules in the specified file as black boxes.
-bboxSysTaskFile *filename*	Defines the system tasks in the specified file as black boxes.
-bdb_incr	Loads the Behavior Database (BDB) incrementally. This option only works when the *-bdb_load* option is specified. For example: *verdi -lib work -top system -bdb_load work.lib++/work.bdb -bdb_incr*
-bdb_load [*filename*]	Loads the Behavior Database (BDB) immediately after loading the design from the library. If *-bdb_load* is specified with *-ba*, *-ba* is ignored. If the file name is specified without a full path, the default path specified in the *[turbo_library]* section of the *novas.rc* resource file is used. For example: *[turbo_library]* *bacom=<the library path>* If the *novas.rc* resource file is not specified, the default value is "*work.lib++*". If the file name is not specified, the default value is "*<lib_path>/work.bdb*" where, *<lib_path>* is the path specified in the *.rc* file if specified; otherwise it is *work.lib++*.
-cellModel 0|1|2|3|4	Specifies the cell usage model. *0*: Uses the cells from the symbol library first. *1*: Only uses the cells from the symbol library. *2*: Only uses the cells from the simulation library. *3*: Uses the cells from the simulation library first. *4*: For combinational cells, the Behavior Analysis engine uses the simulation model first and for sequential cells, the Behavior Analysis engine uses the symbol library model first.
-clockSkew 0 | positive integer	Specifies the worst-case clock skew.
-cont_ba_err	Continues Behavior Analysis even when errors occur.

Options	Explanation
-loopUnroll 0 | positive integer	Defines any *for* loops with a number greater than the specified value as black boxes.
-macroCell {*moduleName}*	Defines the specified module as a macro cell. You can specify multiple modules. For the regular expressions that support cell name searching, see the equivalent *Macro Cell Option Form* section in the *Preferences* chapter.
-macroCellFile *filename*	Defines modules in the specified file as macro cells.
-simModuleExceptionList *{moduleName}*	Uses the specified module in the simulation model. You can specify multiple modules. For the regular expressions that support module name searching, see the equivalent *Simulation Model Option Form* section in the *Preferences* chapter.
-simModuleExceptionListFile *filename*	Uses the modules specified in the file from the simulation library.
-symLibCellExceptionList *cellName*	Uses the specified cell from the symbol library. You can specify multiple modules. For the regular expressions that support cell name searching, see the equivalent *Symbol Library Option Form* section in the *Preferences* chapter.
-symLibCellExceptionListFile *filename*	Uses the cells specified in the file from the symbol library.

Power Manager Options

The following Power Manager options are also supported. These are optional.

Options	Explanation
-aoSignalFile *file*	Specifies the input file where full hierarchy names of always-on signals are listed. **NOTE:** When signals in the *Signal from File* node are double-clicked in the *Power Domain Tree* frame, the power source code frame changes to the related file and line number. The file information is also displayed in the *File* list field on the toolbar.
-cpf1.0 |-cpf1.1 |-cpf2.0 *cpf_filename*	Specifies the CPF file version and the associated file to import. Importing multiple CPF files is also supported. This option can be used with *-lps_cpf*. **Example:** *verdi -sv 1.sv 2.sv ... -cpf1.0 "1.cpf 2.cpf ..."*

Options	Explanation
-domainColor *configFile*	Specifies the configuration file to set the colors of the specified power domains. This option has higher priority than the *novas.rc* resource file and the default. The format of the configuration file is as follows: *@novas rc file Version 1.0* *[PowerDomainColorSetting]* *Domain1 = ID_PURPLE2* *Domain2 = ID_BLUE2* *Domain3 = ID_CYAN2* *Domain4 = ID_GREEN2* *Domain5 = ID_YELLOW2* *Domain6 = ID_ORANGE2* *Domain7 = ID_RED2* *Domain8 = ID_GRAY1* *Domain9 = ID_PURPLE3* If the number of power domains exceeds the color number in the configuration file, colors in the configuration file are used cyclically.
-extractSamePort	When a power design is imported from the command line and this option is specified, the power domain state is evaluated by considering *add_port_state* commands for the same port alias. For example: *verdi -sv 1.v -upf1.0 test.upf -ssf test.fsdb -extractSamePort* **NOTE:** This option is only valid for UPF designs.
-lp_xml *file*	Specifies the name of the low-power XML file to import. The low-power XML file is generated by simulators, such as VCS. **NOTE:** The *-lp_xml* option and the *-debug*, *-debug_all*, or *-debug_pp* options are required during VCS compilation.
-lps_cpf *cpf_filename*	Specifies the name of the CPF file to be used with the design. This option is for compatibility with IUS.
-lps_iso_off	Specifies to turn *off* port Isolation.
-lps_off	Specifies to turn *off* all low-power simulation.
-lps_rtn_off	Specifies to turn *off* state retention.
-power=accurate|auto_complete	Specifies the power compile-time option(s). The plus(+) symbol is used to specify multiple options. The power compile-time options are as follows: *accurate***:** Ignores the impact of isolation-related commands. *auto_complete***:** Supports automatic completion of UPF commands. By default, the automatic completion for UPF commands and command options is disabled. It is recommended to correct the UPF command or command option instead of using *-power=auto_complete*.

Options	Explanation
-power_config *file*	Specifies the name of the power configuration file to import. The power configuration file is also imported from the **Power Configuration File** text field, which is access by clicking the **Import CPF/UPF Files** icon to expand the **CPF/UPF Files** section in the *Library Options Form* or *Import Design Options Form*. The power configuration file supports *, *.*lib*, and *.*db* wildcards.
-powerModel NLP|MVSim|ModelSim|NCSim	Specifies the simulator when parsing the power design. Its default value is NLP. The options and Tcl variables are as follows: *NLP*: This option follows the VCS NLP behavior. *synopsys_program_name: vcs* *upf_tool: MVSIM_NATIVE* *MVSim*: This option follows the VCS MVSim behavior. *synopsys_program_name: upf* *upf_tool: MVSIM* *ModelSim*: This option follows the ModelSim behavior. *synopsys_program_name: Verdi* *upf_tool: ModelSim* *NCSim*: This option follows the NCSim behavior. *synopsys_program_name: Verdi* *upf_tool: MVSIM*
-upf | -upf1.0 | -upf2.0 *upf_filename*	Specifies the UPF file version and the associated file to import. Importing multiple UPF files is also supported by enclosing files with double quotes. The *-upf* option is equal to the *-upf2.0* option. **Example**: *verdi -sv 1.sv 2.sv ... -upf2.0 "1.upf 2.upf ..."*
-upfTop|-power_top *topDesign*	Specifies the top-level design instance for the UPF files. **NOTE:** Instead of using the *-upfTop* option, the *set_design_top* command is specified at the beginning of the loaded UPF file. The Verdi platform sets the UPF power specification's top scope according to either the *-upfTop* option or the *set_design_top* command setting. If the *set_design_top* command is not specified, set the *-upfTop* option. **NOTE:** When the *set_design_top* command and the *-upfTop* option are both used, the specified design top should be the same otherwise an error message is reported.
-upf_ground_logic_value_is_1 *<0|1> | <yes|no>*	Specifies the value to treat ground as "ON is 1". When 0 or no is specified, 0 is considered ON for ground. When 1 or yes is specified, 1 is considered ON for ground. The default value is *1| yes*. This option is valid only for UPF designs.

novas

The *novas* utility is functionally the same as the *verdi* utility. For details, see the *verdi* utility.

siloti

The *siloti* utility is functionally the same as the *verdi* utility. For details, see the *verdi* utility.

nWave

This command invokes the *nWave* GUI.

Usage:

```
nWave [Options]
```

Options

Options	Explanation
-h | -help	Prints the help message.
-nogui	Runs in the batch mode. **NOTE:** A valid *DISPLAY* environment variable is required. The Verdi platform exits automatically in thirty seconds if no more Tcl commands are executed in this mode.
-play *commandHistoryFile*	Plays the command history file. To generate the command history file, specify the *-recordhlc* option.
-quest	Switches the open file GUI to open Quickturn VBF files.
-rcFile *rcFilename*	Specifies the *novas.rc* resource file.
-record *commandHistoryFile*	Records the command history to this file.
-recordhlc *filename*	Records all high-level commands (command-language commands) to a file for playback.
-regf *registerFile*	Loads a register file (*.*register*).
-showtrigger	Shows the red trigger line for the loop marker.

Options	Explanation
-ssc *licenseFile*	Specifies the license file name.
-ssd *hierarchyDelimiter*	Specifies the hierarchy delimiter for EPIC and HSPICE.
-ssf *fastFile(s)*| *dumpFile(s)*| *fastFile list(s)*	Loads FSDB (*.*fsdb*), virtual FSDB (*.*vf*), gzipped FSDB (*.*fsdb.gz*), bzip2 FSDB (*.*fsdb.bz2*), waveform dump (*.*vcd*, *.*vcd.gz*) files, or FSDB file lists (*.*flst*). For individual FSDB files, you can specify 1 - 16 files at a time. **Example**: *nWave -ssf file1.fsdb file2.fsdb... file16.fsdb* or *nWave -ssf demo1.fsdb demo2.fsdb demo3.fsdb* For a file list, each FSDB or virtual file is listed on a separate line with the full or related path to the file. Unlimited files are supported. A pound (#) sign and a semicolon (;) are used for comments; but they are on separate lines. For example: *### Normal FSDB file with full path* */home/proj/mod/file1.fsdb* */proj/simrun/file2.fsdb* *;; Normal FSDB file with related path* *../mod2/file3.fsdb* *./file4.fsdb* *## Virtual FSDB file works as well* *virtual1.vf* */home/proj/mod/virtual2.vf* *####* *./filen.fsdb ## Comments after path are not allowed*
-ssi	Turns on the interactive mode.
-sswr *restoreFile*	Specifies the restore signal file name (*.*rc*).

setupCer

The `setupCer` utility generates the following three Certitude setup files:

- `certitude_config.cer`: This file contains Certitude configuration options settings.
- `certitude_hdl_files.cer`: This file contains the list of source files that make up the design, and the commands that inject faults in some of those listed files.
- `certitude_compile`: This is the compilation script used by Certitude.

All generated files receive the `.generated` extension. You must remove this extension for the file that you want to use.

NOTE: Certitude setup files, `certitude_testcases.cer` and `certitude_execute`, are not generated.

`setupCer` makes interactions between Certitude and the Verdi platform easier and more efficient for Verdi users. You must first adapt the compilation scripts to enable `vericom` (for Verilog designs or mixed with a Verilog top) or `vhdlcom` (for VHDL designs or mixed with a VHDL top). For details, see `vericom` and `vhdlcom` sections.

NOTE: When compiling the design with `vericom/vhdlcom`, specify the `-saveOpts` option to generate the additional information required for `setupCer`.

The `vericom/vhdlcom` step ensures that the appropriate KDB libraries are generated or maintained. `setupCer` then uses these libraries.

Usage:

```
setupCer [Required Options] [Optional Options]
```

Required Options

The following Certitude options are required:

Options	Explanation
-cer_top *topModule*	Specifies a value for the TopName configuration option in Certitude.
-sim_type *nc|ncsim|modelsim|vcs|ifv|jg*	Specifies a value for the simulator configuration option in Certitude.

Options	Explanation
-verdi_top|-novas_top *TopModule*	Specifies the top module to import the design.

Optional Options

The following Certitude options are also supported. These are optional.

Options	Explanation
-G<*parameters*>=<*value*>	Specifies and overrides parameters value of the imported top module.
-help | -h	Prints the list of *setupCer* options and option descriptions.
-InstrumentOnTop	When specified, set the Certitude configuration option *InstrumentOnTop* to *true*.
-L | Lf *libName*	Specifies the search library compiled with the *vericom* utility.
-lib *libName*	Specifies the library name.
-noPostfix	When specified, it is recommended not to add the '*.generated*' postfix to the output file name.
-qualify_cell	When specified, the module compiled by the *-y/-v* library option is instrumented by Certitude.
-rcFile *rcFilename*	Specifies the *novas.rc* resource file.

Example

```
setupCer -cer_top can_top -sim_type vcs -verdi_top can_testbench
-noPostfix
```

Generates the Certitude setup files for a Certitude top name, `can_top` and a Verdi top name, `can_testbench`. The qualification is set to run with VCS. The generated files do have the `.generated` extension.

Analysis

The following utilities are used for batch mode analysis of assertions, Essential Signals, and clocks:

- *assertEval*
- *esa*
- *nAnalyzer*
- *nClockTree*
- *nCompare*

assertEval

`assertEval` is a utility for evaluating assertions in the batch mode. This utility requires the design and FSDB file (where assertion associated signals are dumped) to be specified.

Usage:

```
assertEval [assertEval Options] [Verdi Options] [Simulator
Options] [General Options] [Behavior Analysis Options]
```

Options

```
-out_fsdb fsdbfile [-prop_file filename | -prop_list "{[-type
assert|cover|assume] [-level n] assertion_path} ..."] [other
Options]
```

NOTE: For details on the Verdi, Simulator, General, and Behavior Analysis options, see the Verdi help menu.

The following table summarizes the `assertEval` options:

Options	Explanation
-bt time[unit]	Specifies the start time of evaluation (for example, 2000ns). The time unit is *s*|*ms*|*us*|*ns*|*ps*|*fs*. The default time unit is *ns*.
-enable_sv_display *filename*	Enables *$display* in the SystemVerilog assertion. When "*filename*" is specified, results are saved to this file. Otherwise, results are printed in the console.
-et time[unit]	Specifies the end time of evaluation (for example, 5000ns). The time unit is *s*|*ms*|*us*|*ns*|*ps*|*fs*. The default time unit is *ns*.
-filter_cover_no_match on|off	When this option is set to *on*, no-match coverage results are filtered. When this option is set to *off*, coverage results are not filtered. The default value is *on*.

Options	Explanation
-filter_reset_success on|off	When this option is set to *on*, reset (*disable iff*) success results are filtered. When this option is set to *off*, reset (*disable iff*) success results are not filtered. The default value is *on*.
-filter_vacuous_success on|off	When this option is set to *on*, vacuous-success results are filtered. When this option is set to *off*, vacuous-success results are not filtered. The default value is *on*.
-h | -help	Prints the help message.
-no_hdl_import	Evaluates assertions in the SVA-bound module without importing the design.
-out_fsdb *fsdbfile*	Specifies the path and output file name for the results. If this option is not specified, the results are saved to *evaluate_result.fsdb* in the assertEvalLog directory by default.
-prop_file *filename*	Specifies the file name that contains the list of assertions to be evaluated. The arguments from *-prop_list* are used in the file. **Example:** `-type assert+cover -level 0 top.m1.assert1` `-type cover top.m1.c*` If neither *-prop_file* nor *-prop_list* is specified, all assertions in the design are evaluated.
-prop_list "{[-type assert|cover|assume] [-level n] assertion_path} ... "	Specifies one or more assertion paths to be evaluated. Each assertion path is surrounded by angle brackets and contains the full path to the assertion with the "." delimiter; the wildcard character (*) is supported. The *-type* and *-level* arguments are optional and you can specify either/both with each assertion path. The double quote " must begin and end the *-prop_list* arguments. A space is required to separate two arguments. **Example:** *-prop_list "{-type assume+assert top.m2.checker2.ck*.a_s*}{-type assert+cover -level 0 top.m1.assert1}"* If neither *-prop_file* nor *-prop_list* are specified, all assertions in the design are evaluated.
-level *n*	Evaluates assertions from the specified scope and its *n* descendants. Use this option in conjunction with *-prop_list* (or within the file for *-prop_file*). The default level number is *1*. *-level 0*: Evaluates assertions from the specified scopes and all hierarchies below. *-level 1*: Evaluates assertions of the specified scopes only. *-level 2*: Evaluates assertions of the specified scopes and the first generation scopes.

Options	Explanation
-type assert|cover|assume	Specifies one or more assertions types by selecting from the following: assert, cover, and assume. Multiple types are separated with the plus (+) symbol. The default is *assert+cover+assume*. It must be used in conjunction with *-prop_list*. It may also be specified in the file for *-prop_file*.
-record_success on|off	When this option is set to *on*, success results are recorded. When this option is set to *off*, success results are not recorded. Its default value is *on*.
-statistic_report *textfile*	Specifies the text file to output the statistics report. A statistics report is only created when this option is specified.
-store all | assert_only	Specifies the assertion data types to be stored. Its valid values are as follows: *all*: Stores all assertion data types except SVA local variables. *assert_only*: Stores only assert/cover/assume directives. The default value is *assert_only*.

Examples

1. Evaluate all assertions in the design using default settings and save the assertion results under the log directory, for example:

   `./assertEvalLog/evaluate_result.fsdb`

   ```
   > assertEval -sv dut.sv +define+SVA -ssf dut.fsdb
   ```

2. Evaluate all assertions in the design with different settings (for example, turn off filtering for vacuous success and no-match covers and stores all assertion data types) and generate a text report in the `statistics.log` file using the following command:

   ```
   > assertEval -statistic_report statistics.log -store all \
     -filter_vacuous_success off -filter_cover_no_match off \
     -f run.f -ssf debug.fsdb
   ```

3. Evaluate all assertions of type assume or assert with names matching `a_s*` in scopes matching `top.m2.checker2.ck*` subscopes. Also, generate a text report. For example:

   ```
   > assertEval -statistic_report statistics.log \
   -prop_list "{-type assume+assert top.m2.checker2.ck*.a_s*}" \
   -lib svlib -top top -ssf debug.fsdb
   ```

4. Multiple arguments of the `-prop_list` option are supported. For example:

   ```
   > assertEval -statistic_report statistics.log \
   -prop_list "{-type assume+assert top.m2.checker2.ck*.a_s*} \
   {-type assert -level 2 top.m3.assert*} \
   {-type cover top.m4.cover*.*check*}" \
   -f run.f -ssf debug.fsdb
   ```

5. Evaluate all assertions with assertion type as `cover` in the `top.m1.checker` scope and its next two subscopes.

```
> assertEval -statistic_report statistics.log \
-prop_file propfile -f run.f -ssf debug.fsdb
```

   The propfile includes the following line:

```
-type cover top.m1.checker -level 2
```

6. Translate Example 4 to use the `-prop_file` option. For example:

```
> assertEval -statistic_report statistics.log
-prop_file propfile -f run.f -ssf debug.fsdb
```

   The `propfile` file includes the following lines:

```
-type assume+assert top.m2.checker2.ck*.a_s*
-type assert -level 2 top.m3.assert*
-type cover top.m4.cover*.*check*
```

esa

`esa` is a batch mode executable for Essential Signal Analysis and is used to generate an Essential Signal Database (ESDB). The ESDB provides the ability to set the majority of Essential Signal Analysis (ESA) options at the simulation runtime instead of rerunning ESA whenever the ESA options need to change. The utility progress is printed to the screen and the results are saved to the `esaLog` directory.

Essential Signal Analysis is performed for the specified working scope (using `-bas`). If `-bas` is not specified, `esa` uses the working scope in the `novas.rc` resource file (if one exists); otherwise, the top module of the design (Verilog only) is used.

NOTE:

1) The `esa` utility requires a Siloti-SimVE license.
2) If the `esa` utility is used to perform Essential Signal Analysis, the `-db` option is added and the output file is an ESDB format file. For example: `esa -db es -top top`
3) Set the `NOVAS_BATCH_KILL` environment variable to *1* when using the `esa` utility in LSF environments.

Usage:

```
esa [ESA Options] [General Options] [nTrace Options] [Simulator
Options] [Environment Options] [Behavior Analysis Options] [Power
Manager Options]
```

For details, see *General/nTrace Options*, *Simulator Options*, *Environment Options*, *Behavior Analysis Options*, and *Power Manager Options* sections in the *verdi* utility description.

The following table summarizes the ESA options:

Options	**Explanation**
-db *esdb_filename*	Dumps Essential Signal Database to the specified file.
-bdb_file *filename*	Specifies the file name to save the Behavior Database (BDB).
-bdb_load *filename*	Loads the Behavior Database (BDB) immediately after loading the design from the library. If *-bdb_load* is specified with *-ba*, *-ba* is ignored.
-bdb_load_scope "*pathName*"	Specifies the hierarchical path for the top module to be loaded from the Behavior Database (BDB) file. This option only works when the *-bdb_load* option is specified. **Example:** *> verdi -lib work -top system -bdb_load work.lib++/work.bdb -bdb_load_scope "system.i_cpu"*
-dump_seq_libcell_bound 0|1	When set to *1*, signals connected to sequential library cells is dumped to the Essential Signal list. The default value is *0*. This option is optional.
-exclude *module(s)*	Excludes signals inside the specified modules. You can specify one or more modules. Alternatively, use *-excludefile* for specifying multiple modules. Regular expressions are supported. This option is optional.
-excludefile *filename*	Specifies the file containing a list of modules whose signals need to be excluded from the list. This option is optional. In the file, each module is listed on a separate line. Regular expressions are supported. For example: *CPU* *CCU* *pram_[1-2]* *alu**
-h | -help	Prints the help message.
-ignore_errors_ba 0|1	Set to *0* to stop executing Essential Signal Analysis when there are Behavior Analysis errors. Set to *1* to continue executing Essential Signal Analysis even when Behavior Analysis has errors. Its default value is *0*. This option is optional.

Options	Explanation
-ignore_errors_vcom 0|1	Set to *0* to stop executing Essential Signal Analysis when there are design loading errors. Set to *1* to continue executing Essential Signal Analysis even when there are errors loading the design. Its default value is *0*. This option is optional.
-info *N*	Specifies the information level for the log. Its default value is *N=0*. This option is optional.
-libcell 0|1	Specifies whether to dump library cell inputs and outputs. *0*: Does not dump library cell I/Os. *1*: Dumps library cell I/Os. The default value is *0*. This option is optional.
-libPath *pathName*	Specifies the library path to save the Behavior Database (BDB).
-no_bdb	*esa* does not save the Behavior Database (BDB).
-optimalA *filename*	Specifies the text file containing a list of proposed modules to suppress to obtain optimized Essential Signal Analysis results. The resulting text file is edited and used with the *$fsdbSuppress* dump command during simulation. Alternatively, the listed modules or file itself is used with one of the exclude options (*-exclude modules* or *-excludefile file*) with another run of the *esa* utility. This option is optional. The output list format includes the reason and estimated ES count reduction for each module cell: *#reason(~estimated reduce-ES%)* *module*
-xpinfile *filename*	Specifies the pin(s) in a module/cell to be included. This option is optional. In the module/cell port file, each module/cell name and its corresponding pin name(s) are listed on a separate line. For example: *# Generic example.* *# [module/cell name] [pin name] ([pin name] ...)* *# Specify pin in a cell to be included.* *MXSCANDFFA SIN* *# Specify multiple pins in a cell to be included.* *MXSCANDFFA SIN Q*

Examples

1. Generate an ESDB for the full chip:

   ```
   > esa -f run.f -db es -bas chiptop
   ```

   `es.esdb++` is generated including all Essential Signals in the design.

2. Generate an ESDB for the ALUB module:

   ```
   > esa -f run.f -db es -bas system.i_CPU.i_ALUB
   ```

   `es.esdb++` is generated including all Essential Signals for the ALUB module.

3. Generate an ESDB for the entire chip excluding the PCU module:

   ```
   > esa -f run.f -exclude PCU -db es
   ```

   `es.esdb++` is generated including all Essential Signals except those in the PCU module.

4. List the excluded modules, ranking by percentage the amount the module (for example, `candidates`) reduces the Essential Signal count:

   ```
   > esa -f run.f -db es -bas chiptop -optimalA candidates
   ```

   where, `candidates` contains:

   ```
   #ESA-O: modules suggested to be excluded for saving ES-dump
   # size with limited visibility loss
   #sequential library cell (~21%)
   cell1
   #sequential library cell (~12%)
   cell2
   ```

   As the above example shows, `cell1` and `cell2` are sequential library cells and the ES count is reduced to 21% and 12% respectively when excluded.

nAnalyzer

The `nAnalyzer` command is used to invoke batch mode clock analysis and extraction functions. The command mode helps to reduce memory usage by completing clock analysis, extracting clocks, and checking crossing paths in the batch mode. The clock analysis results are saved to a clock database file whose name is "`nClock.cdb`" by default. After executing the batch command, invoke the GUI and load the results from the **File -> Load Clock Domain Results** command in the *nTrace* window and the *Clock Domain* window gets opened simultaneously. Use the **Tools -> List Crossing Paths** command in the *Clock Domain* window to see the crossing paths, instead of invoking the **Clock Analyzer -> List Crossing Paths** command in the *nTrace* window.

Both "nAnalyzer" and "Verdi" licenses are required and are checked out to execute this command.

Usage:

```
nAnalyzer [Options]
```

Options

Options	Explanation
-f *filename*	Specifies the design file to be imported.
-h | -help	Prints the help message.
-lib *libName*	Specifies the library name.
-load *filename*	Specifies the setting file for clock domain analysis.
-path *pathName*	Specifies the path of the design file.
-rcFile rc*Filename*	Specifies the *novas.rc* resource file.
-simdir *path*	Specifies the path where the VCS executable (vcs) is executed. **NOTE:** Use this option with the *-simflow* option.
-simflow	Loads the Verdi Knowledge Database (KDB) generated by VCS and use the library mapping from the *synopsys_sim.setup* file. **NOTE:** Before specifying this option, set the *VCS_HOME* environment variable to point to the VCS installation.
-top *topModuleName*	Specifies the top module for the imported design.
-translateOn	Ignores the RTL source code between SYNOPSYS compiler directives *synopsys translate_off* and *synopsys translate_on*.
-vhdl | -verilog	Specifies the language type for importing the design from the source.

The setting file for `-load filename` is an `rc` format file. The syntax for the setting file is described below along with an example of its use. The setting file uses the pound (#) sign to designate comment lines and square brackets "[ ]" to define sections. The setting file includes six sections: `[File]`, `[SDC]`, `[CKDomain]`, `[CDC]`, `[Synchronizer]`, and `[Global]`. `[TempSection]` is also available for each user-defined synchronizer template.

NOTE: The '`@Verdi rc file Version 1.0`' line must be the first line in the `novas.rc` resource file for the file to be properly recognized by the Verdi platform.

NOTE: The library settings are loaded from the `novas.rc` resource file unless the `manage.rc` resource file is specified. For details on using the `manage.rc` file, see *Appendix B: Customizing Verdi*.

[File]

This section specifies the name of the clock database. The default value is "`nClock.cdb`".

Syntax

```
ClockDBFile=nClock.cdb
```

Example

```
[File]
ClockDBFile=output.cdb
```

In the above example, the clock database is loaded from `output.cdb`.

[SDC]

This section defines the Synopsys Design Constraint (SDC) files. You can load multiple SDC files.

Syntax

```
Top = module_name
Item_0 = SDC file1
Item_1 = SDC file2
...
Item_n = SDC file3
```

Example

```
Top = system
Item_0 = TOP.sdc
Item_1 = default.sdc
```

In the above example, the top module of the SDC is `system`. The value must be set when loading SDC files. The SDC files are `TOP.sdc` and `default.sdc`.

[CKDomain]

This section specifies the options for Clock Domain Extraction.

Syntax

```
Pass = pass module lists
Bypass = bypass module lists
clkSrcOnly = True|False
mergeCell = GATECLK CombLogic
reglist = True|False
separateRegToMultiClkDmn = True|False
traceMultiResolveSource = True|False
```

In the syntax, the default value of `Pass` and `Bypass` is `Null`. The default value of `clkSrcOnly` and `traceMultiResolveSource` is `True`. The default value for `reglist` and `separateRegToMultiClkDmn` is `False`. The default value of `mergeCell` is `GATECLK CombLogic`.

[CDC]

This section specifies the options for Crossing Paths Checking.

Syntax

```
CheckBetweenSameDomain = True|False
CheckConvergence = True|False
CheckDivergence = True|False
CheckReconvergence = True|False
```

In the syntax, the default value of `CheckBetweenSameDomain`, `CheckConvergence`, `CheckDivergence`, and `CheckReconvergence` is `False`.

[Synchronizer]

This section defines the synchronizer settings of Check Crossing Paths. If no synchronizers are defined, the default value of `Item_0` is `NULL`.

Syntax

```
Item_0 = template1
Item_1 = template2
...
Item_n = templaten
```

Example

```
Item_0 = Template sync_1A
Item_1 = Template sync_configuration BBB
```

In the example, the first synchronizer uses a default template, `sync_1, Type_A`. The second synchronizer uses a user-defined configuration, which has a corresponding *`[TempSection0]`*.

[Global]

This section specifies the number of templates.

Syntax

```
TemplateNum = 0|1
```

In the syntax, the default value of `TemplateNum` is *0* if no template is set as the synchronizer.

[TempSection0]

The example below shows the content of a user-defined synchronizer template. The content varies depending on the synchronizer type.

Example

```
Name = BBB
Type = 2
ConnNum = 2
Conn0 = Latch@2
Conn1 = Latch@2
```

nClockTree

The `nClockTree` utility performs the clock tree extraction in the batch mode and saves the results in a file that can be loaded into the Verdi platform.

Usage:

```
nClockTree [Options]
```

Import Design Options

Options	Explanation
-f *filename*	Specifies the design file to be imported.
-lib *libName*	Specifies the name of the library that contains the source files compiled with the *vericom* or *vhdlcom* utilities.
-path *pathName*	Specifies the path name for the imported design file.
-rcFile *rcFilename*	Specifies the *novas.rc* resource file.
-top *topModule*	Specifies the top module of the imported design.
-vhdl | -verilog	Specifies the language type for the imported design.

SDC/CTS Options

Options	Explanation
-ctsFeFile *ctsFilename*	Specifies *FirstEncounter* as the format of the CTS file for clock tree analysis.
-ctsIccFile *ctsFilename*	Specifies ICC as the format of the CTS file for clock tree analysis.
-sdcFile *sdcFileName*	Specifies an SDC file for clock tree analysis.

Clock Tree Setting Options

Options	Explanation
-checkDesignConstant on|off	Specifies whether to check design constant signals in the clock tree. The default value is *off*.
-ff *clockPin*|*nonClockPin*|*allPins*|*ignore*	Specifies the instance port type for flip-flops set as leaves of the clock tree. The default value is *clockPin*.
-floating *stop*|*ignore*	Specifies whether to show floating gate instances set as leaves of the clock tree. The default value is *ignore*.

Options	Explanation
-latch *controlPin|nonControlPin|allPins|ignore*	Specifies the instance port type for latches set as leaves of the clock tree. The default value is *controlPin*.
-macro *passthrough|clockPin|nonClockPin|allPins|ignore*	Specifies the instance port type for clock macros set as leaves of the clock tree. The default value is *passthrough*.
-outputPort *stop|ignore*	Specifies whether to show output ports set as leaves of the clock tree. The default value is *stop*.
-sdcGLKPassFF *on|off*	Specifies whether to pass through all registers between the clock source and the generated clock during tracing. The default value is *off*.

Report File Options

Options	Explanation
-convergeReport	Enables the converge report mode.
-coverageReport	Enables the coverage report mode.
-GCCPathOutputFile *outputFileName*	Specifies the output file name for the gated clock cell paths.
-output *outputFilename*	Specifies the file name for the clock tree report. The default file name is *default.cld*.
-outputCon *conReportName*	Specifies the file name for the converge report. The default file name is *default.con*.
-outputCov *covReportName*	Specifies the file name for the coverage report. The default file name is *default.cov*.
-outputFormat *Database|Astro*	Specifies the format for the clock tree report. Its default value is *Database*.

Astro Report Options

Option	Explanation
-loop on|off	Specifies whether to add comment "//loop" when the tree node loops the previous path. Its default value is *off*. **NOTE:** The report format takes the loop node as the leaf node.
-outputNets *filename*	Specifies to dump nets to a report file when the nets have crossed module(s) or hierarchical instance(s) during clock tree tracing. Its default value is *netInfo.log*.

Option	Explanation
-sdcGLKPassDataPin on|off	Specifies whether to pass through the data pin of the generated clock instance. Its default value is *on*.
-showIgnoreFlag on|off	Specifies whether to show Astro ignore flags on the Astro report. Its default value is *on* if *-outputFormat* is Astro.
-showIgnoreTree on|off	Specifies whether to show ignore trees on the Astro report. Its default value is *on* if *-outputFormat* is Astro.
-skipRTLPath on|off	Specifies whether to ignore the RTL path. The net that connects to the RTL instance is reported into a side file. Its default value is *off*.

Miscellaneous Option

Option	Explanation
-h | -help	Prints the help message.

nCompare

The `nCompare` utility compares two FSDB files using a defined rule file and reports the errors to a file.

The `nCompare` utility supports three kinds of digital data formats: Verilog and VHDL. The `nCompare` utility also supports analog and other non-digital data; however, the comparison is done using the FSDB file format.

The `nCompare` utility can compare signals of the same value type, and also compare digital signals with analog signals. The related `nce2report` utility transforms comparison results into a readable report, thereby allowing the results to be viewed easily, even in the console mode.

Usage:

```
nCompare [General Options] [GUI Mode Options] [Batch Mode
Options]
```

A default comparison rule is used when no rule file is specified in the batch mode.

Rule:

```
cmpSetSignalPair -level 0
cmpCompare
```

Report File:

If a report file is not specified, the mismatched error report is saved in a default file. For example:

On executing the following command:

```
nCompare -fsdb golden.fsdb -fsdb secondary.fsdb
```

The following message is displayed:

```
The report file is not specified, we saved mismatched error
report to:
…./nCompareLog/Default.nce
For comparison log, refer to:
…/nCompareLog/compare.log
```

General Options

Options	Explanation
-logdir *logDirectory*	Specifies the location of the log directory.
-logfile *logFile*	Specifies the location of the log directory/file.

GUI Options

Options	Explanation
-gui [rule_file | error_file]	Invokes the *nCompare* GUI. If either a rule file (*.*ncr*) or error report file (*.*nce*) is specified, the file is loaded.

Batch Mode Options

Options	Explanation
-fsdb *filename*	Specifies the Golden or Secondary FSDB file.
-log_matched	Specifies the log for the matched compared pair(s). This option is used when you want to view the matched signal pair in the log file.
-precise_name_matching	Specifies the compared pair that have precise matching name. This option is used when golden and second FSDB files have the exact signal names, including the bus range information.
-report *filename*	Specifies the error report file (**nce*). If not specified, it is saved into a default path (current nCompareLog directory).
-rule *filename*	Specifies the rule file (**ncr*).
-signal *signal_name*	Specifies the signal to compare. You can specify multiple signals; however, the *-signal* option needs to be specified for each additional signal.
-silence	Suppresses the comparison progress message.

Compilation

The following utilities are used for design compilation:

- *aliasextract*
- *bacom*
- *elabcom*
- *crdb*
- *gencom*
- *libls*
- *nc2novasrc*
- *nfixhier*
- *netlistcom*
- *nrun*
- *spicom*
- *vericom*
- *vhdlcom*
- *vlmap*

aliasextract

The `aliasextract` utility creates an alias file for a precompiled design (Verilog, VHDL, or mixed). After the alias file is created, it is loaded (using the `-aliasFile` option) with the design on the Verdi command line.

Usage:

```
aliasextract [Options]
```

Options	Explanation
-elab libName	Specifies the elaboration library. For example, `aliasextract -elab simv.daidir/kdb.elab++`
-lib *libName*	Specifies the library. The default value is *work*.
-output *outputFile*	Specifies the output file name. By default, it is saved to *extracted.src_alias* in the working directory.
-rcFile *rcFilename*	Specifies the *novas.rc* resource file.

Options	Explanation
-simflow	Loads the Verdi Knowledge Database (KDB) generated by VCS and uses the library mapping from the *synopsys_sim.setup* file. **NOTE:** Before specifying this option, set the *VCS_HOME* environment variable to point to the VCS installation.
-top *moduleName*	Specifies the top module.

bacom

The `bacom` utility saves the results of Behavior Analysis to the Behavior Database (BDB). The Behavior Database (BDB) is loaded to the Verdi platform by specifying the `-bdb_load` option on the `novas/verdi` command line or through the *Behavior Analysis* form (invoked with the **Tools** -> **Temporal Flow View** -> **Behavior Analysis** command in *nTrace*).

NOTE:

1) The Behavior Database (BDB) must be associated with KDB libraries, so the design needs to be compiled into libraries (using `vericom` or `vhdlcom`) first.
2) The `bacom` utility forces the BA mode in Working Scope Behavior Analysis (WSBA).
3) The options in the `bacom` utility are saved to the `bacom.command` file in the `bacomLog` directory to show how the Behavior Database (BDB) is generated. You can reuse the BDB.

Usage:

```
bacom [Options]
```

Options	Explanation
-apimem *filename*	Specifies a file to define the API memory.
-ba_mode WSBA	Specifies the Behavior Analysis mode. Only Working Scope Behavior Analysis (WSBA) mode is supported.
-bas "scope(s)" | "*"	Performs Behavior Analysis (with the given scopes as working scopes) immediately after loading the design. For example, *-bas "{scope1} {scope2}"* takes "scope1" and "scope2" as working scopes and *-bas "{*}"* takes all loaded top modules as working scopes.

Options	Explanation
-bboxEmptyModule 0|1	Specifies whether to define the empty modules as black boxes. 0: Does not define the empty modules as black boxes. 1: Defines the empty modules as black boxes. Its default value is *0*.
-bboxIgnoreProtected 0|1	Specifies whether to ignore protected code. 0: Ignores the protected code. 1: Inferences the protected code. Its default value is *0*.
-bboxModule {*moduleName*}	Specifies the module to be treated as a black box. You can specify multiple modules. For the regular expressions that support module name searching, see the equivalent *Black Box Setup Form* section in the *Preferences* chapter.
-bboxModuleFile *filename*	Defines the modules in the specified file as black boxes.
-bboxSysTaskFile *filename*	Defines the system tasks in the specified file as black boxes.
-bdb_file *filename*	Specifies the file name to save the Behavior Database (BDB). The default value is *work.bdb*.
-cellModel [0|1|2|3|4]	Specifies the cell usage model. 0: Uses the cells from the symbol library first. 1: Only uses the cells from the symbol library. 2: Only uses the cells from the simulation library. 3: Uses the cells from the simulation library first. 4: For combinational cells, the Behavior Analysis engine uses the simulation model first and for sequential cells, the Behavior Analysis engine uses the symbol library model first.
-clockSkew 0 | positive integer	Specifies the worst-case clock skew.
-confined_flatten 0 | positive integer	Sets the maximum leaf number of complex signals. The default value is *32768*.
-constraints *filename*	Specifies the file containing a list of scopes that is inferenced one by one. The format of the constraint file is as follows: -s scope <newLine>
-cont_ba_err	Continues performing Behavior Analysis even when errors occur from importing the design.
-h | -help	Prints the help message.

Options	Explanation
-impConf *confFile*	Specifies a configuration file for the imported design. Partial load and excluding instances are only used with import from library; undefined modules and virtual top are used with both import from library and import from file. **NOTE:** If used with *-vtop*, *-vtop* is ignored. Configuration File Format: A configuration file contains four types of information including *Virtual_top*, *Undefined_modules*, *Partial_load*, and *Exclude*. 1. *[Undefined_modules]*: Suppresses warning messages of undefined modules. For example: *[Undefined_modules]* *cpu* *alu* 2. *[Virtual_top]*: Specifies virtual top mapping information. For example: *[Virtual_top]* *system = vaTop.sys* *CPU = vbTop.t_cpu* 3. [*Partial_load*]: Specifies tree information to execute partial load. For example: If the hierarchy of a design looks as follows: *- L1c - L2c - L3c - L4c* *Top - L1a - L2a - L3a - L4a* *- L1b - L2b - L3b - L4b* then to load the design excluding *- L1b - L2b - L3b - L4b* into the Verdi platform, the partial load section in a configuration file is edited as: *[Partial_load]* *Top.L1a.L2a.L3a.L4a* *Top.L1a.L1c* **NOTE:** 1) When using partial load, the design is imported from the library. 2) All instantiations under L1c is also loaded. 3) If both *Virtual_top* and *Partial_load* sections are in a configuration file, the *Partial_load* section is ignored and a warning message is displayed. 4) *[Exclude]*: Specifies excluded modules or scopes. **Example:** *[Exclude] CCU* Excludes module instances instantiated by CCU. *[Exclude] system.i_CCU.i_mprom* Exclude *system.i_CCU.i_mprom* itself and below. 5) When excluding instances, the design can only be imported from the library.

Options	Explanation
	If a scope is specified in both the partial load and exclude sections, the scope is not loaded by *exclude* (*exclude* is prior to partial load).
-L/LF *libName*	Specifies the search library compiled with *vericom*.
-lib *libName*	Specifies the library compiled by *vericom* or *vhdlcom*. Its default value is *work*. **NOTE:** The *bacom* utility only supports importing from libraries.
-libcellfile *filename*	Specifies the file that contains module/entity as the library cell. The module/entity is taken as the library cell if they are specified in this file. For the file format, each module/entity name is separated by a space, tab, or enter a character.
-libPath *pathName*	Specifies the library path to save the Behavior Database (BDB). If *-libPath* is not specified, the default path is the path specified in the *[turbo_library]* section of the *novas.rc* resource file. For example: *[turbo_library]* *bacom=<the library path>* If the *novas.rc* resource file is not specified and there is only one *-v* option, the default is the path of the *-lib* option. If multiple *-lib* options are specified, the default is *work*.
-logdir *logDirectory*	Specifies the location of the log directory.
-logfile *logFile*	Specifies the location of the log directory/file.
-loopUnroll 0 | positive integer	Defines any *for* loops with a number greater than the specified value as black boxes.
-macroCell *{moduleName}*	Defines the specified module as a macro cell. You can specify multiple modules. For the regular expressions that support cell name searching, see the equivalent *Macro Cell Option Form* section in the *Preferences* chapter.
-macroCellFile *filename*	Defines the modules in the specified file as macro cells.
-managercFile *path*	Specifies the full path of the *manage.rc* resource file.
-Mentor	Imports from the library with Mentor search rules.
-nclib	Uses the NC binding scheme.
-rcFile *rcFilename*	Specifies the *novas.rc* resource file. If an option in the *novas.rc* resource file conflicts with an option in *bacom*, the *bacom* option is adopted.
-simBin "<*path*>"	Specifies the path of the simulation binary file. The path must be double quoted.

Options	Explanation
-simdir *path*	Specifies the path where the VCS executable (vcs) is executed. **NOTE:** Use this option with the *-simflow* option.
-simflow	Loads the Verdi Knowledge Database (KDB) generated by VCS and uses the library mapping from the *synopsys_sim.setup* file. **NOTE:** Before specifying this option, set the *VCS_HOME* environment variable to point to the VCS installation.
-simModuleExceptionList *{moduleName}*	Uses the specified module in the simulation model. You can specify multiple modules. For the regular expressions that support module name searching, see the equivalent *Simulation Model Option Form* section in the *Preferences* chapter.
-simModuleExceptionList File *filename*	Uses the modules specified in the file from the simulation library.
-ssv	Library modules in the library file (*-v*) are not tagged as library cells.
-ssy	Library modules in the library directory (*-y*) are not tagged as library cells.
-ssz	Ignores '*celldefine* compiler directives. **NOTE:** If you load the symbol library, the cell in symbol library has higher priority than a user-defined module.

Options	Explanation
-top *topModule* | "*top1 top2 ... topN*"	Specifies the top modules for the imported design. **NOTE:** Verilog modules are case-sensitive. The top module name is specified using the Cadence NC simulator command format to elaborate the designated design unit, such as *ncelab [library.]cell[:view]*. **Example:** *verdi -nclib -top "novas.top(rtl)"* The Verdi platform elaborates *top* in the *novas* library and the *rtl* view. When the library is not specified, search *work.lib++* for the top design unit. **Example:** *verdi -nclib -top top* is the same as *verdi -nclib -lib work -top top or verdi -nclib -top work.top* Multiple top modules in the same library are supported. **Example:** *verdi -top "top1 top2 top3"* When multiple top modules from different libraries are required, use the following format: *verdi -top "libA.top1 libB.top2 libC.top3"* The Verdi platform invokes three hierarchy trees for top1, top2, and top3. When multiple arguments are specified and the *-bas* option is not specified, only the first top module is used for Behavior Analysis.
-translateOn	Ignores the modules between Synopsys *translate_off* and Synopsys *translate_on*.
-v_no_elab	Library modules in all library files (*-v*) are not elaborated. This option is overwritten if the *-ssv* option is specified.
-y_no_elab	Library modules in all library directories (*-y*) are not elaborated. This option is overwritten if the *-ssy* option is specified.
-z_no_elab	Cell modules (`celldefine) are not elaborated. This option is overwritten if the *-ssz* option is specified.

Power Manager options are also supported. For details, see the *Power Manager Options* section of the *verdi* utility.

Examples

1. Perform Behavior Analysis on the design with default settings and save the result to the default path, `./work.lib++/work.bdb`:

   ```
   > bacom -lib work -top test
   ```

2. Perform Behavior Analysis on the design with default settings and save the result to the specified path, `./work2.lib++/work2.bdb`:

```
> bacom -lib work -top test -libPath ./work2 -bdb_file work2.bdb
```

3. Perform Behavior Analysis on the design with multiple scopes and save the result to the specified path, `./work2.lib++/work2.bdb`:

```
> bacom -lib work -top test -bas "{test1} {test2}" \
-libPath ./work2 -bdb_file work2.bdb
```

4. Perform Behavior Analysis on the design loaded from multiple libraries and save the result to the default path, `./work.lib++/work.bdb`:

```
> bacom -lib work1 -lib work2 -lib work3 -top test
```

5. Perform Behavior Analysis on the design which takes every top module as a working scope:

```
> bacom -lib work -top "test1 test2" -bas "{*}"
```

6. Perform Behavior Analysis on the design. The working scopes are specified in the constraint file:

```
> bacom -lib work -top test -constraints scopes.f
```

elabcom

`elabcom` is a batch mode utility to elaborate the library created by `vericom`/`vhdlcom`, and generates the elaborated KDB. You can read the elaborated KDB (the default name is `kdb.elab++`) into Verdi with the `-elab` option.

Usage:

```
elabcom [HDL Elaboration Options] [HDL General Options]
```

Elaboration Options

Options	**Explanation**
-dynaconfig <*filename*>	Specifies a VCS runtime configuration file to load. The supported format of this configuration file includes the full path of the replaced instance and module names of the replaced instance. For example: *top.instance_A.instance_B* *ModuleA* *ModuleB* **NOTE**: The *-dynaconfig* option works only with Verilog designs.
-DEFPARAM <parameter_path>= <value>	Changes the specified parameter to the specified value.

Options	Explanation
-elab <path>	Specifies the path of the generated Verdi *elabDB*.
-G<paramName>= <value>	Overrides the parameter value in the design.
-gv | -gvalue generic= <value>	Overrides the value defined in the source code with the value specified in the option.
-ieFile	Specifies the report file of the interface element to be imported.
-impConf *confFile*	Specifies a file to configure how undefined modules, virtual tops, and partial loads are handled during design import. **NOTE:** This option does not load a VHDL configuration file.
-L/-Lf *libName*	Specifies one or more Verdi libraries compiled with *vericom* or *vhdlcom* that do not contain the top view. **NOTE:** As to search sequence, the *-Lf* option has higher priority than the *-L* option.
-lib *libName*	Specifies the library name.
-libcellfile *filename*	Specifies the file which contains the module/entity as the library cell. The module/entity is taken as the library cell if they are specified in this file.
-liblist *listNames*	Searches the specified libraries compiled by VCS. The plus sign (+) is used to include multiple libraries in the library list.
-liblist_work	Starts searching from the parent library first. The *-liblist_work* option has higher search priority than the *-liblist* option.
-msv	Supports the Mixed Signal Verification.
-msvDir *path*	Specifies the Mixed Signal Verification report directory with the full path.
-nclib	Specifies whether to run as the Cadence NC library management scheme.
-noinc	Suppresses incremental loading.
-novasLibPaths *filename(s)*| *libName(s)*	Specifies one or more symbol library paths or files containing the paths to all symbol libraries.
-novasLibs *filename(s)*| *libName(s)*	Specifies one or more symbol library names or files containing the names of all symbol libraries.
-ovm [-<version>]	Loads the default Verdi OVM library. **NOTE:** If *-ovm* and *-ovmhome* are specified at the same time, *-ovm* is ignored.
-ovmhome <path>	Specifies the OVM installation directory.

Options	Explanation
-parameters *filename*	Specifies the file containing a list of parameters to be changed to values. The file format is *assign <value> <parameter_path>*.
-pvalue+<*parameter_hierarchical_name*>=<*value*>	Changes the specified parameter to the specified value.
-simdir <*path*>	Specifies the path where the VCS executable (vcs) is executed. **NOTE**: If the *synopsys_sim.setup* file specified with the *SYNOPSYS_SIM_SETUP* environment variable exists, this option is ignored. NOTE: Use this option with the *-simflow* option.
-simflow	Loads the Verdi Knowledge Database (KDB) generated by VCS and uses the library mapping from the *synopsys_sim.setup* file. **NOTE**: Before specifying this option, set the *VCS_HOME* environment variable to point to the VCS installation.
-top *topModule* | *"top1 top2 ... topN"*	Specifies the top modules for the imported design. Multiple top modules in the same library are supported.
-uvm [-<version>	Loads the default Verdi UVM library. **NOTE:** If *-uvm* and *-uvmhome* are specified at the same time, *-uvm* is ignored.
-uvmhome <path>	Specifies the UVM installation directory.
-v_no_elab	Library modules in all library files (*-v*) are not elaborated. This option is overwritten if the *-ssv* option is specified.
-vtop *file* | *assignment_command*	Specifies a virtual top file with filename or with assignment command for import designs. **NOTE:** Assignment command has higher priority than filename.
-vtop *vhdl_mapfile* -vtopvhdl	Specifies a virtual top file in VHDL for import designs. **NOTE:** Use *-vtopvhdl* with *-vtop*.
-y_no_elab	Library modules in all library directories (*-y*) are not elaborated. This option is overwritten if the *-ssy* option is specified.
-z_no_elab	Cell modules (*`celldefine*) are not elaborated. This option is overwritten if the *-ssz* option is specified.

Examples:

1. Generate the `work.lib++` library:

```
> vericom -f run.f
```

2. Generate the `kdb.elab++` elaborated KDB:
   ```
   > elabcom -top system
   ```

3. Read the elaborated KDB into Verdi:
   ```
   > verdi -elab kdb.elab++
   ```
   Alternatively, execute the following:
   ```
   > verdi -elab kdb
   ```

General Options

Options	Explanation
+disable_message+<*message_serial_numbers*|*error*|*warning*>	Suppresses the specified message(s), all error messages, or all warning messages. Use a plus (+) symbol for different message type combinations.
-ssv	Do not automatically tag library modules in the library file (*-v*) as library cells.
-ssy	Do not automatically tag library modules in the library directory (*-y*) as library cells.
-ssz	Ignores '*celldefine* compiler directives. **NOTE:** If you load the symbol library, the cell in symbol library has higher priority than a user-defined module.
-useius	Uses IUS style parsing/elaboration. **NOTE:** When more than one of the *-useius*, *-usemti*, and *-usevcs* options are specified, only the first option specified is recognized; subsequent options are ignored.
-usemti	Uses MTI style parsing/elaboration. **NOTE:** When more than one of the *-useius*, *-usemti*, and *-usevcs* options are specified, only the first option specified is recognized; subsequent options are ignored.
-usevcs	Use VCS style parsing/elaboration. **NOTE:** When more than one of the *-useius*, *-usemti*, and *-usevcs* options are specified, only the first option specified is recognized; subsequent options are ignored.

crdb

`crdb` is a batch mode executable for correlation database (CRDB) extraction. The results are saved to the `crdbLog` directory. Correlation extraction is performed for the specified working scope (using `-bas`). If `-bas` is not specified, `crdb` uses the working scope in the `novas.rc` resource file (if one exists); otherwise, the top module of the design (Verilog only) is used.

NOTE:
1) The `crdb` utility requires a Siloti-SimVE license.
2) Set the `NOVAS_BATCH_KILL` environment variable to *1* when using the `crdb` utility in LSF environments.

Usage:

```
crdb [CRDB Options] [General Options] [nTrace Options]
[Simulator Options] [Environment Options] [Behavior Analysis
Options] [Power Manager Options]
```

For details, see the *General/nTrace Options*, *Simulator Options*, *Environment Options*, *Behavior Analysis Options*, and *Power Manager Options* sections in the *verdi* utility.
The following table summarizes the CRDB options:

Options	Explanation
-GATE "*Siloti_options*"	Within the double quotes behind *-GATE*, specify the Siloti options needed to load the gate-level design. When this option is used with *-RTL*, the complete CRDB is generated. Its details are logged to the <*logdir*>*/GATELog* directory. The default value for <*logdir*> is *crdbLog*.
-RTL "*Siloti_options*"	Within the double quotes behind *-RTL*, specify the Siloti options needed to load the RTL design. When this option is used with *-GATE*, the complete CRDB is generated. Its details are logged to the <*logdir*>*/RTLLog* directory. The default value for <*logdir*> is *crdbLog*.
-allreg	When specified, all registers are mapped from the gate to the RTL for exporting the mapping file. Use this option with *-expMap* and do not use this option with *-fsdb* or *-txt*.
-append	Appends data into the CRDB database. This is used in an incremental build flow for large designs.
-crdb *filename*	Specifies the working CRDB library name.

Options	Explanation
-excludeModule *filename*	All signals except port signals in excluded modules are excluded from the mapping file. The port signals in the excluded modules are listed in the mapping file. Use this option with the *-expMap* option.
-expGate	When specified, the correlation analyzer exports gate objects from the mapping file. This option is optional and must be used with *-expMap*.
-expGateUncor	When specified, the correlation analyzer exports uncorrelated gate objects. This option is optional and must be used with *-expMap*.
-expMap *filename*	Exports the mapping results to the specified file. This option must be used with only one of the *-fsdb, -txt, or -allreg* options.
-expRtl	When specified, the correlation analyzer exports RTL objects from the mapping file. This option is optional and must be used with the *-expMap* option.
-filterModule *filename*	When specified, all signals in filtered modules are excluded from the uncorrelated signal list. A new mapping rate based on the filtered signals is reported. Use this option with the *-expMap* option.
-filterInstance *filename*	When specified, all signals in filtered cell instances are excluded from the uncorrelated signal list. A new mapping rate based on the filtered signals is reported. Signal names (ports and nets) are also specified directly. Use this option with the *-expMap* option.
-fsdb *filename*	When specified, all signals in the specified FSDB file are mapped from the gate to the RTL for exporting the mapping file. Use this option with the *-expMap* option and do not use with *-txt* or *-allreg*.
-gate	Indicates a gate-level design is loaded and the gate-level portion of the CRDB is updated.
-h | -help	Prints the help message.
-impMap *filename*	Specifies the user-defined mapping file. When specified, the correlation information from the mapping file is applied to the CRDB and the internal correlation algorithms are skipped.
-impSVF *filename1 [filename2...filenameN]*	Imports the specified files of the synthesis tool guidance in the SVF format and the register correlation information from the specified SVF files is applied to the CRDB. Siloti Correlation naming rules are applied to the remaining unmapped signals. Enclose multiple files within square brackets.

Options	Explanation
-impSVFList *filename*	Specifies a file that contains the list of the synthesis tool guidance in the SVF format. The register correlation information from the specified SVF files would be applied to the CRDB. Siloti Correlation naming rules apply to the remaining unmapped signals. If both `-impSVF` and `-impSVFList` are specified, then SVF files from both options are imported.
-novasLibPaths *libpath1 [libpath2 ... libpathN]*	Specifies one or more search paths where the symbol libraries are located. Separate multiple library paths by spaces.
-novasLibs *libname1 [libname2 ... libnameN]*	Specifies one or more symbol library names. Separate multiple library paths by spaces.
-rtl	Indicates a RTL design is loaded and the RTL portion of the CRDB is updated.
-rule *filename*	Specifies the file name containing the name mapping rules for correlation. Use this option with the *-rtl* option.
-target_scope *scopeName*	Specifies the scope to export the mapped file. Use this option with the *-expMap* option.
-top_inport	Includes input ports that are successfully mapped from the gate-level design to the RTL design in the top scope. Use this option with the *-expMap* option.
-txt *filename*	When specified, all signals in the specified text file are mapped from the gate to the RTL for exporting the mapping file. Use this option with the *-expMap* option and do not use with *-fsdb* or *-allreg*.

Examples:

1. Generate a correlation database for the gate design:

```
> crdb -f run_gate.f -gate -crdb extracted.crdb -bas top
```

2. Extract the RTL correlation database and add to the gate extraction, and perform correlation mapping:

```
> crdb -f run_rtl.f -rtl -crdb extracted.crdb \
-rule myrules.lst -bas top
```

3. Generate a correlation database for the gate and RTL designs, and perform correlation mapping:

```
crdb -crdb extracted.crdb -rule myrules.lst \
-GATE "-f run_gate.f -bas top" -RTL "-f run_rtl.f -bas top"
```

4. Load the specified CRDB and export the mapping according to the export related options.

```
> crdb -crdb crdb.crdb -expMap map.gz -allreg
```

gencom

For ModelSim users, `gencom` is developed to reduce the effort when importing the simulator-compiled designs into the Verdi platform. `gencom` uses the existing environment setup, `modelsim.ini`, for ModelSim. It generates a command list or file list that contains all of the compiled source files in the ModelSim libraries and keeps the correct library dependency and compile dependency. Use the generated command list or file list to import the VHDL/Verilog designs into the Verdi platform.

NOTE: `gencom` supports several `vlog` language options (for example, `-2001`, `-sv`, and `-vlog95compat`) as described below:

1) `vlog` uses the 2001 standard as the default language and `vericom` uses 95; therefore, `gencom` automatically generates commands for `vericom` with *-2001* as the default. For example, when `vlog a.v` is used, `gencom` generates the following:
`vericom -2001 +systemverilogext+.sv a.v`.

2) `vlog` enables the SystemVerilog support if either of the following exists:
(a) With the `-sv` option
(b) The file has an `.sv` file extension

In (a), if `vlog` is used with `-sv`, `gencom` generates commands for `vericom` with `-sv`. For example, when `vlog -sv a.v` is used, `gencom` generates '`vericom -2001 -sv a.v`' (`+systemverilog+.sv` is redundant).

In (b), `vericom` has an option similar to `+systemverilogext+.sv` to support SystemVerilog parsing for `.sv` files. `gencom` automatically generates commands for `vericom` with `+systemverilogext+.sv` as the default. For example, when '`vlog a.sv`' is used, `gencom` generates '`vericom -2001 +systemverilogext+.sv a.sv`'.

3) `vlog` has the `-vlog95compat` option to disable the 2001 standard and SystemVerilog support (both situations, 2(a) and 2(b) are overridden). If `vlog` is used with this option and `-sv`, `gencom` generates commands for `vericom` without the `-2001`, `-sv`, and `+systemverilogext+.sv` options. For example, when '`vlog -vlog95compat -sv a.sv`' is used, `gencom` generates '`vericom a.sv`'.

Usage:

```
gencom [Options]
```

Options	Explanation
-filelist	Generates output as a file list. By default, *gencom* generates a command list. To see the format for the file list and command list, refer the NOTE below.
-fullsrcpath	Outputs the full path for each design file name.
-h | -help	Prints the help message.
-h -lf	Shows the usage for the *Leapfrog* environment.
-lf	Generates from the *Leapfrog* environment. The default is generated from MTI.
-lib *libName*	Specifies a dedicated VHDL library to extract. *gencom* extracts the source files that are compiled in this specified ModelSim library.
-map *mapFile*	Uses a map file to specify the source file searching paths for libraries. For the format of *map_file*, see the NOTE below.
-nrc	Do not auto update the *novas.rc* resource file. The default is auto update the library mapping in the *novas.rc* resource file.
-o file	Specifies the output file name. Its default value is *run.com*.
-skipothers	Skips the other references in the *modelsim.ini* file.
-tf *filename*	Specifies a dedicated VHDL design file to extract. Without this option, *gencom* extracts all of the source files for those libraries specified in the *modelsim.ini* file.

***NOTE:**

1) The format of `map_file` is as follows:

```
Modelsim_initial_file = <full_path/modelsim_initial_file>
<library_name = "search_path";...;"search_path">
<library_name = "search_path";...;"search_path">
                          .
                          .
OTHERS=<"search_path";...; "search_path">
```

For example, `gencom.map` is the map file and contains the following:

```
modelsim_initial_file = "./modelsim.ini"
work = ".";"../src_dir"
vital = "."
memory = "../src_dir"
OTHERS = "."
```

If the paths of the source files are updated after using `vcom` to compile ModelSim's libraries or the source files are not specified with their full path to use `vcom` to compile source files into ModelSim libraries,

gencom is unable to find those source files from the specified ModelSim's initial file in the map file. With the <library_name= Paths of the source files> in the map file, gencom can extract the exhausted source files of the design.

2) The format of the command list is as follows:

```
vhdlcom -work libraryName <path>/source_file (for VHDL
source code)
vericom -work libraryName <path>/source_file (for Verilog
source code)
```

The example of the command list file is as follows:

```
run.com
  vhdlcom -work work ../src/pred_fns.vhd
  vhdlcom -work work ../src/packageCPU.vhd
  vhdlcom -work work ../src/maprom.vhd
  vhdlcom -work work ../src/microrom.vhd
  vhdlcom -work work ../src/pram.vhd
  vhdlcom -work alub_lib ./arithlogic.vhd
  vhdlcom -work alub_lib ./ALU.vhd
  vhdlcom -work ccu_lib ./CCU.vhd
  vhdlcom -work pcu_lib ./PCU.vhd
  vhdlcom -work cpu_lib ./CPU.vhd
  vhdlcom -work work ../testbench/system.vhd
```

The format of the file list is as follows:

```
<path/source_file>
```

The example of the file list is as follows:

```
run.f
  ../src/pred_fns.vhd
  ../src/packageCPU.vhd
  ../src/maprom.vhd
  ../src/microrom.vhd
  ../src/pram.vhd
  ./arithlogic.vhd
  ./ALU.vhd
  ./CCU.vhd
  ./PCU.vhd
  ./CPU.vhd
  ../testbench/system.vhd
```

Use the -filelist option to generate a file list that is imported easily into the Verdi platform. gencom generates a command list output if the design includes different VHDL options, even if the -filelist option is already applied. This includes source files that use VHDL-87 and some use VHDL-93, mixed-language designs, or designs that are

compiled into multiple libraries to run the simulation. Under these circumstances, you cannot use a file list to compile and import the design into the Verdi platform.

libls

`libls` is a utility that lists the contents of a library.

NOTE: The library settings are loaded from the `novas.rc` resource file unless the `manage.rc` resource file is specified. For details on using the `manage.rc` resource file, see *Appendix B: Customizing Verdi*.

Usage:

```
libls [-help] [-rcFile rcFileName] [-lib libName]
[-dependency] [designUnit] [-all]
```

Options	Explanation
-all	Dumps all the design units of all libraries specified in the resource file.
-checksum	Checks the correctness of all dependencies.
-dependency	Lists the dependencies of the library unit.
-h | -help	Prints the help message.
-lib *libName*	Specifies the logical library name. When this option is not specified, all libraries listed in the specified resource file (*-rcFile* option) are dumped.
-rcFile *rcFilename*	Specifies the *novas.rc* resource file. The resource file is either a file in the local directory, or it can include the path. **Example:** *-rcFile novas.rc* or *-rcFile /home/usr1/a.rc*

Example:

```
libls -rcFile novas.rc -lib cpu ALUB
```

Dumps the information of the library unit ALUB in the `cpu` library.

nc2novasrc

`nc2novasrc` is a utility that converts the library setting of the NC simulator, `cds.lib` and `hdl.var`, to the `novas.rc` resource file. The library mappings from `cds.lib`, including the nested `cds.lib` files, are extracted and the corresponding `[Library]` section is built into the `novas.rc` resource file. Some predefined libraries, such as `ieee`, `std`, `synopsys`, `vital2000`, and `novas`, are skipped. `LIB_MAP` and `VIEW_MAP` from `hdl.var`, including nested `hdl.var` files, are also extracted and the corresponding `[NC_Sim]` section built into the `novas.rc` resource file.

Usage:

```
nc2novasrc [Options]
```

Options	Explanation
-a	Automatically finds *hdl.var* and *cds.lib* in the current directory.
-cdslib *filename*	Specifies the input *cds.lib* file. If neither the *-cdslib* nor *-hdlvar* options are specified, *nc2novasrc* tries to find *cds.lib* and *hdl.var* in the current directory.
-hdlvar *filename*	Specifies the input *hdl.var* file. If neither the *-cdslib* nor *-hdlvar* options are specified, *nc2novasrc* tries to find *cds.lib* and *hdl.var* in the current directory.
-h | -help	Prints the help message.
-o *rcFile*	Specifies the output resource file name. Its default value is *novas.rc*. If the output resource file does not exist, a new one is created. If the output resource file exists, *nc2novasrc* appends/updates the new library mapping to the existing library mapping in the *[Library]* section or appends/updates the *[NC_Sim]* section.

Example:

`cds.lib` contains the following:

```
define worklib worklib
define vlog_lib ../vlog_lib
define vhdl_lib /da100/integ/vlog_lib
include /rd42b/LDV2.2/tools.sun4v/inca/files/cds.lib
```

`hdl.var` contains the following:

```
include hdl1.var
define VIEW_MAP (${VIEW_MAP}, .v => behav,\
                 .rtl => rtl,\
                 .gate => gate)
```

```
define LIB_MAP ( ../designlib =>designlib,\
                 ../src => src,\
                 + => worklib )
```

`hdl1.var` contains the following:

```
#hdl1.var: Defines view_map
include hdl0.var
define VIEW_MAP ($view_map ,.vrtl => rtl)
```

`hdl0.var` contains the following:

```
#hdl0.var: Defines view_map
define VIEW_MAP (.vgate => .gate)
```

When `nc2novasrc -a -o project.rc` is executed, the `project.rc` file is created and the following `[Library]` and `[NC-Sim]` sections are added:

```
[Library]
vlog_lib = /lx115a/RCC/GCC2_96/LINUX/bt4.1/regression/kdbreg/
Misc/nc2novasrc/vlog_lib
vhdl_lib = /da100/integ/vlog_lib
worklib = /lx115a/RCC/GCC2_96/LINUX/bt4.1/regression/kdbreg/
Misc/nc2novasrc/SPSqa57421/worklib

[NC_Sim]
viewMap = (.vgate => .gate ,.vrtl => rtl, .v => behav,
.rtl => rtl,.gate => gate)
libMap = ( ../designlib =>designlib,../src => src,
+ => worklib )
```

nfixhier

`nfixhier` is a utility to fix the resolving libraries for the hierarchy of the specified cell and to save the cell information in libraries. Afterwards, the design is loaded from the library using the `-fixCellHier` option on the Verdi command line. The search priority is higher than the priority of the `-L` option.

Usage:

```
nfixhier [Options]
```

Options	Explanation
-cell [*lib*:]cell[:*view*]	Specifies the cell name. The library name and view name are optional. The library name, cell name, and view name are separated with a colon (:) character.
-getCellHierInfo	Displays the cell hierarchy information.
-h | -help	Prints the help message.
-L *libname*	Specifies the library name to search.
-removeCellHierInfo	Removes the cell hierarchy information.

Example:

```
top.sv
-------
module top;
modA i_modA();
endmodule

1.sv
-------
module modA;
endmodule

2.sv
------
module modA;
endmodule

% vericom -sv top.sv
% vericom -sv 1.sv -lib L1
% vericom -sv 2.sv -lib L2
% nfixhier -cell top -L L2

% verdi -top top -fixCellHier -lib work -L L1 -L L2
```

t`op.i_modA()` is from `2.sv`, not `1.sv`, even though the `-L L1` option is specified.

netlistcom

`netlistcom` is a utility that performs RTL extraction in the batch mode. After HDL designs are compiled into libraries with `vericom` or `vhdlcom`, `netlistcom` can further compile the libraries into more specific netlist views enabling the schematics in `nSchema` to be displayed faster.

NOTE: The library settings are loaded from the `novas.rc` resource file unless the `manage.rc` resource file is specified. For details, see *Appendix B: Customizing Verdi*.

Usage:

```
netlistcom [-completeRTL] [-detailRTL levelNumber] [-f
constraintFile]
[-lib libName] [-libPath pathName] [-rcFile filename] [-s
scopePath] [-top moduleName] [-translateOn] [-pr]
```

Options	Explanation
-completeRTL	Generates both non-detailed and detailed RTL schematic DB libraries.
-detailRTL *levelNumber*	Specifies the detailed RTL extraction mode and the level number.
-elab <path>	Specifies the path of Verdi elaborated KDB.
-f *constraintFile*	Specifies a list of scopes in the constraint file, which is compiled one by one. **Example:** *netlistcom -lib work -f scopes.f* The format of the constraint file: *-s scope <newLine>*
-h | -help	Prints the help message.
-L/LF *libName*	Specifies the search library compiled by *vericom*.
-lib *libName*	Specifies the library created with *vericom* or *vhdlcom*. Its default value is *work*.
-libPath *pathName*	Specifies the library path to save the extracted schematic database. By default, the database (*VerdiLib.lib++*) is stored in *work.lib++*. For options, specify the path to use in the *[turbo_library]* section of the *novas.rc* resource file. For example: *[turbo_library]* *netlistcom=<the library path>*
-rcFile *rcFilename*	Specifies the resource file where the Verdi platform can find the library mapping information. The default resource file is *novas.rc*.

Options	Explanation
-s *scopePath*	Specifies the sub design to be compiled. **Example:** *netlistcom -lib work -s system.i_cpu.i_ALUB*
-simdir *path*	Specifies the path where the VCS executable (vcs) is executed. **NOTE:** Use this option with the *-simflow* option.
-simflow	Loads the Verdi Knowledge Database (KDB) generated by VCS and uses the library mapping from the *synopsys_sim.setup* file. **NOTE:** Before specifying this option, set the *VCS_HOME* environment variable to point to the VCS installation.
-top *moduleName*	Specifies the top module of the design to be compiled.
-translateOn	Ignores the modules between Synopsys *translate_off* and Synopsys *translate_on*.

nrun

`nrun` is a batch mode compiler for mixed-languages. It compiles mixed design files (Verilog and VHDL) to libraries by classifying the design files to SystemVerilog/Verilog and VHDL types. It then invokes `vericom` and `vhdlcom` utilities respectively to compile the design files of the related language type to libraries.

Usage:

```
nrun [Options] inputFiles
```

Options

Language Extension Options:

Options	Explanation
-ams	When this option is specified on the *nrun* command line, HSPICE constructs are recognized. This means the content between *.subckt* and *.ends* is treated as a comment and a node is added to the design browser to bind the instance. **NOTE:** This option is not case-sensitive.
-default_ext *fileType*	Set the default language type. Its default value is *systemverilog*. All files with unrecognized extensions are treated as the given file type. The available values of file types are: *verilog95*, *verilog*, *systemverilog*, and *vhdl*. **Example:** *> nrun -default_ext systemverilog cpu.v alu.sss* is equivalent to *> vericom +verilog2001ext+.v +systemverilogext+.sss cpu.v alu.sss* As the default language type is *systemverilog*, the unrecognized file extension *.sss* of the file *alu.sss* is treated as a SystemVerilog type. **NOTE:** This option is not case-sensitive.
-define	Defines a macro. **NOTE:** This option is not case-sensitive.
+disable_message+<*message_serial_numbers*>	Ignores the specified error/warning message(s). For details, see *Appendix F: Message Table*.
-incdir *directory*	Specifies an include directory. **NOTE:** This option is not case-sensitive.
+libext+ <extension_name>	Specifies the file extensions for the Verilog library files. **NOTE:** This option is not case-sensitive.

Options	Explanation
+librescan	Searches the library list again from the first element of the library list when an unresolved module is discovered. **NOTE:** This option is not case-sensitive.
+libverbose	Displays a message when the module definition is found in the Verilog library files. It resolves the module instantiation in the source files or library files and displays the location where the generated *lib++* is saved. **NOTE:** This option is not case-sensitive.
-libmap <*filename*>	Specifies the library mapping file. **NOTE:** This option is not case-sensitive.
-makelib *libName*	Specifies the library to save the design. Its default value is *work*. **NOTE:** This option is not case-sensitive. **Example:** *nrun -sv 1.v -makelib work_mine*
-reflib *libName*	Searches for the package in the specified library when converting the *vericom*/*vhdlcom* compilation script. Its default value is *work*. **Example:** *nrun -sv 1.v -reflib work_mine*
-smartorder	Compiles in order of the independent mode. **NOTE:** This option is not case-sensitive. **Limitations**: 1) You cannot use the *-smartorder* option with the *-skip/just* and *-comment_transoff_regions* options. Because the *-skip/just* and *-comment_transoff_regions* options may hide design content. Also, it may ignore some design units or use clauses that may change dependencies and affect the results of using the *-smartorder* option. Therefore, such usage is not supported. If the *-smartorder* option is used with these two options, they are ignored automatically. 2) Library dependence is not considered with the *-smartorder* option. Dependencies are handled separately. **Example:** The *main_des (a.vhd, b.vhd)* library depends on the pkg_pram (p.vhd, q.vhd) library. Its correct usage is as follows: 1) *vhdlcom -smartorder p.vhd q.vhd -lib pkg_pram* 2) *vhdlcom -smartorder a.vhd b.vhd -lib main_des* An alternative usage is to set the library section in the *novas.rc* resource file as follows: *Pkg_pram = wk* *Main_des = wk* and then compile with: *vhdlcom -smartorder a.vhd b.vhd p.vhd q.vhd -lib wk*

Options	Explanation
-sv	Supports the selected SystemVerilog syntax. **NOTE:** This option is not case-sensitive.
-sysv_ext +*fileext,..* | *fileext,...*	Extends or replaces the default SystemVerilog file extensions. **NOTE:** This option is not case-sensitive. **Example:** *nrun -sysv_ext +.vv top.vv* *= vericom +systemverilogext+.vv top.vv*
-timescale=<time_scale>	Sets the default timescale for the modules without the timescale definition. **NOTE:** This option is not case-sensitive.
-v200x	Enables the VHDL -2008 syntax. **NOTE:** This option is not case-sensitive.
-vhcfg_ext *fileext,..* | *fileext,...* -vhdlext *fileext,..* | *fileext,...*	Extends or replaces the default VHDL file extensions. **NOTE:** This option is not case-sensitive. **Example:** *nrun -vhcfg_ext .aa,.bb,.cc top.aa cpu.bb alu.cc* *= vhdlcom top.aa cpu.bb alu.cc*
-vhdl_ext +*fileext,..* | *fileext,...*	Extends or replaces the default VHDL file extensions. Using a plus (+) symbol adds the new extensions to the current extension file. Not using a plus (+) symbol replaces the original default mapping file list with a new list. The plus (+) symbol is optional. **NOTE:** This option is not case-sensitive. **Example:** *nrun -vhdl_ext .aa,.bb,.cc top.aa cpu.bb alu.cc* *= vhdlcom top.aa cpu.bb alu.cc*
-vlog95_ext +*fileext,..* | *fileext,...*	Extends or replaces the default Verilog 95 file extensions. **NOTE:** This option is not case-sensitive. **Example:** *nrun -vlog95_ext .aa 2.aa* *= vericom +verilog1995ext+.aa 2.aa*
-vlogext *fileext,..* | *fileext,...* -vcfgext *fileext,..* | *fileext,...*	Extends or replaces the default Verilog file extensions. Using a plus (+) symbol adds the new extensions to the current extension file. Not using a plus (+) symbol replaces the original default mapping file list with a new list. The plus (+) symbol is optional. **NOTE:** This option is not case-sensitive. **Example:** *nrun -vlogext .aa+,.bb -vlogext .cc top.aa cpu.bb alu.cc* *= vericom +verilog2001ext+.aa+.bb+.cc top.aa cpu.bb alu.cc*

Options	Explanation
-vlog_ext +*fileext,..* | *fileext,...*	Extends or replaces the default Verilog file extensions. Use +*fileext,...* to add new extensions to the current extension list. For example, *> nrun a.vg b.vg c.vg -vlog_ext +.vg* Use *fileext,...* to replace the default Verilog file extensions with a new list. For example, *> nrun a.vg b.vg c.vg -vlog_ext .v,.vlog,.vp* **NOTE:** This option is not case-sensitive. **Example:** *> nrun -vlog_ext .vv -vlog_ext +.pp,.qq 1.sv 2.vv 3.pp 4.vhd 5.qq* is equivalent to: *> vericom +verilog2001ext+.vv+.pp+.qq 1.sv 2.vv 3.pp 5.qq* *> vhdlcom 4.vhd*
-work <*libName*>	Uses the specified library name for the design units of the input Verilog source files. **NOTE:** This option is case-insensitive.

Library Options:

Options	Explanation
-makelib *libraryName sourceFiles* [-endlib|-end]	Compiles the files that follow the option into the specified library. The files without a specified library are saved to the default library named as "*work*". **NOTE:** This option is not case-sensitive. **Example:** *> nrun -makelib libA 1.sv 2.vhd -makelib libB 3.vhd -endlib 4.v* is equivalent to: *> vericom +systemverilogext+.sv 1.sv -lib libA* *> vhdlcom 2.vhd -lib libA* *> vhdlcom 3.vhd -lib libB* *> vhdlcom 4.v -lib work*
-reflib *libraryName sourceFile*	Adds the specified library to the searching library when compiling *vericom*/*vhdlcom* source files. Multiple libraries are allowed. **Example:** *nrun –reflib another –reflib another2 sourcefile.[v,vhd]* is equivalent to: *> vlmap another another* *> vlmap another2 another2* *> vhdlcom sourcefile.vhd (for VHDL files)* *> vericom -L another –L another2 sourcefile.v (for Verilog files)*

Miscellaneous Options:

Options	Explanation
-f *runFile*	Specifies the run file.
-h | -help	Prints the help message. **NOTE:** This option is not case-sensitive.

Input Files

The `inputFiles` argument is used to specify design files. Each specified input file must have a full path or a local path.

The `nrun` utility determines the language type of a file by its file extension as shown below.

Language Type	File Extensions	vericom/vhdlcom options
Verilog 95	.v95, .V95, .v95p, .V95P	+verilog1995ext+<extension>
Verilog (2001)	.v, .V, .vp, .VP, .vs, .VS, vcfg	+verilog2001ext+<extension>
System Verilog	.sv, .SV, .svp, .SVP, .svi, .svh, .vlib, .VLIB	+systemverilogext+<extension>
VHDL (93)	.vhd, .VHD, .vhdl, .VHDL, .vhdp, .VHDP, .vhdlp, .VHDLP, .vhcfg	

Example:

```
> nrun a.v95 b.v c.sv d.vhd
```

is equivalent to:

```
> vericom
+verilog1995ext+.v95
+systemverilogext+.sv
a.v95 b.v c.sv
> vhdlcom d.vhd
```

nrun Log Files

`nrun` produces the following log files in the `nrunLog` directory:

- The `dispatch.log` file shows the dispatch information.
- The `nrun.log file` shows unrecognizable options and files.
- The `nrun_vericom_libName_lib.run_f` file shows the `vericom` file list for `libName`.

- The `nrun_vhdlcom_libName_lib.run_f` file shows the `vhdlcom` file list for `libName`.

The relevant `vericom/vhdlcom` log files are found in `vericomLog/compiler.log` and `vhdlcomLog/compiler.log`.

Example:

```
> nrun -sysv_ext +.v cpu.v top.v -93 cpu.vhd -veriopts
"-2001genblk" -vhdlopts "-sup_sem_error"
> cat nrunLog/dispatch.log
vericom -2001genblk -sv cpu top.v
vhdlcom -sup_sem_error -93 cpu.vhd

> nrun -nologo top.v -lib libA cpu.v
> cat nrunLog/nrun.log
Unrecognized option : -nologo
Unrecognized option : -lib

> nrun -sv -makelib lib1 top.v -endlib
> cat nrunLog/nrun_vericom_lib1_lib.run_f
top.v
```

The `-nologo` and `-lib` options are not recognized by `nrun`.

Supported vericom and vhdlcom Options

The options for the `vericom` and `vhdlcom` utilities are also specified on the `nrun` command line. `nrun` dispatches the options to `vericom` or `vhdlcom`.

Utility	Options
vericom	-ams, -cunit, +define+<macro>, -define <A>=<B>, +incdir+<directory_name>, -incdir <directory>, -L <library>, +libext+<extension_name>, +librescan, +libverbose, -libmap <filename>, -nclib, -ny <directory>, -nv <filename>, -P <filename>, -RegPort, -rep[lace], -ssv, -ssy, -ssz, -sv_pragma, -timescale=<time_scale>, -u, -v <filename>, -vc, -view <viewName>, -work <libName>, -y <directory_name>, -2001, -2005, -sv /sverilog
vhdlcom	-ignoreSameEntity, -libcell, -fv <packfile>, -just eapbc, -nochecking, -onerrorstop, -onfatalerrorcontinue, -prelib <yes|no>, -skip eapbc, -smartscript <filename>,-2000, -87, -93, -v93, -v200x
Both for vericom and vhdlcom	-rcFile

For `vericom/vhdlcom` options that are not supported in the table above, the following options are used to pass them to `vericom/vhdlcom`.

-veriopts option_list

Explicitly specifies a list of `vericom` options for `nrun` to pass to the `vericom` utility.

-vhdlopts option_list

Explicitly specifies a list of `vhdlcom` options for `nrun` to pass to the `vhdlcom` utility.

Example:

```
nrun -vc -nochecking -veriopts "-stdout" \
-vdhlopts "-rcFile vhd.rc" 1.sv 1.vhd
```

is equivalent to:

```
vericom +systemverilogext+.sv -vc -stdout 1.sv
vhdlcom -nochecking -rcFile vhd.rc 1.vhd
```

spicom

The `spicom` utility supports to compile the SPICE and Spectre formats to the Verdi Database (KDB) library.

Usage:

```
spicom [Options] files
```

Options	Explanation
-ad *filename*	Loads the simulation initial file.
-eldo	Supports the Eldo format.
-f *filename*	Loads an ASCII file containing design source files.
-lib *libName*	Specifies the library to save your design. Its default value is "*work*".
-runfile *filename*	Loads the simulation run file.
-simdir *path*	Specifies the path where the VCS executable (vcs) is executed. **NOTE:** Use this option with the *-simflow* option.
-simflow	Generates the Verdi Knowledge Database (KDB) using the library mapping from the *synopsys_sim.setup* file. **NOTE:** Before specifying this option, set the *VCS_HOME* environment variable to point to the VCS installation.
-topname *cellName*	Specifies the top cell name. Its default value is "*topcell*".

vericom

`vericom` is a batch mode compiler for Verilog languages (for example, Verilog, Verilog-2001, and SystemVerilog). `vericom` checks and saves the Verilog design into a library. By default, `vericom` updates existing modules in the library.

NOTE:
1) The default Verilog compile mode is Verilog-2001.
2) Loading gzipped (`*.gz`) files for the Verilog design is supported.
3) The library settings are loaded from the `novas.rc` resource file unless the `manage.rc` resource file is specified. For details on using the `manage.rc` resource file, see *Appendix B: Customizing Verdi*.
4)The `-ssy`, `-ssz`, and `-ssv` options must not be used if the compiled gate-level design is used with the `nECO` module for ECO modification.

Usage:

```
vericom [Options...] [Verilog Options]
```

Options	Explanation
-2001	Supports Verilog IEEE 1364-2001 standard.
-2001genblk	Uses Verilog IEEE 1364-2001 naming style for generate blocks (overrides other language options). IEEE1364-2005 sec12.4.3 unnamed genblk naming is not applied. **NOTE:** This option only works with *-sv* or *-2005*, and it is for VCS users only (VCS does not support IEEE1364-2005 sec12.4.3 unnamed genblk naming through version Y-2006.06-11).
-2005	Supports Verilog IEEE 1364-2005 standard. IEEE1364-2005 sec12.4.3 unnamed genblk naming is applied by default.
-2009	Supports SystemVerilog 2009 constructs.
-2012	Supports the SystemVerilog IEEE 1800-2012 standard.
-ams	When this option is specified on the *vericom* command line, HSPICE constructs are recognized. This means the content between *.subckt* and *.ends* is treated as a comment and a node is added to the design browser to bind the instance.
-applog	Appends the compile messages to a log file. The default mode is *overwriting*.

Options	Explanation
-assert checker|svaext|svvunit	Specifies the additional syntax to support. *checker*: Supports the checker construct of the SystemVerilog IEEE 1800-2009 standard. *svaext*: Supports SystemVerilog Assertion features compliant to the IEEE 1800-2009 standard. *svvunit*: Supports PSL vunit syntax. Files with the **.psl* file extension are treated as PSL files.
-autoendcelldef	Automatically appends '*endcelldefine* at the end of a file if a matching '*endcelldefine* is not found for a declared '*celldefine* in the file.
-chmod *numeric_mode*	Specifies the access attribute for generated files and directories. For example: *vericom 1.v -chmod 777* Changes the log directory (default is *vericomLog*) and the library directory (default is *work.lib++*) to have full read/write/executable (for example, *drwxrwxrwx*) access.
-comment_transoff_regions -suboption | +suboption	Skips the source code between *translate_off* and *translate_on* pragmas. The suboptions are vendor names (for example, *cadence, ikos, mentor, novas, pragma, quickturn, synopsys,* and *synthesis*). To skip a single region, use the *-suboption*. To skip multiple regions, use *+suboption1+suboption2...* For examples, see the **-comment_transoff_regions** option in the *verdi* utility.
-cuname *cuName*	Supports compilation-based compilation-unit mode with a specific name. The global space is modeled as a package named "*cuName*". It is disabled if specified with *-cunit*. **Example:** *vericom -sv 1.v -cuname myCU -lib work* (*cunit* with the name *myCU(work)*)
-cunit	Supports compilation-based compilation-unit mode with a default name. The global space is modeled as a package named "*novas_cunit_n*", where *n* is an increased serial number to distinguish compilation units in the same library. **Example:** *vericom -sv 1.v -cunit -lib work* (*cunit* with the name *_$novas_unit__1 (work)*) *vericom -sv 1.v 2.v -cunit -lib work* (*cunit* with the name *_$ novas_unit__2 (work)*) **NOTE:** *-cunit* has higher priority than *-cuname*.
-define	Defines a macro.

Options	Explanation
-disable_ams_default_ disciplines	Disables the defaults provided by the AMS standard discipline definition.
+disable_message+<*message_serial_numbers|error|warning*>	Suppresses the specified message(s), all error messages, or all warning messages. Use a plus (+) symbol for different message type combinations. For details about each message number and its description, see *Appendix F: Message Table*. **Examples:** Suppress the specified message(s) *+disable_message+C00288* *+disable_message+C00288+C00266* Suppress the specified message and all warning messages *+disable_message+C00288+warning* Suppress the specified message, all error messages, and warning messages *+disable_message+C00288+error+warning*
-disableVYDbg	Disables the knowledge database creation for *-v -y* files.
-error=no<Error_ID>, ...	Reports the specified error messages as warning messages. The following error type is supported: *MPD*: Module <*module_name*> redefined
-errormax *number*	Specifies the maximum number of errors to report. If the errors exceed this number, the parser does not report the remaining errors.
-errorstop *redefined_module*	Stops parsing the remaining files when there is a redefined module.

Options	Explanation
-extinclude	Compiles the included files with the version specified by their extension.When this option is not specified and a source file for one version of Verilog contains the '*include* compiler directive, *vericom* by default compiles the included file for the same version of Verilog, even if the included file has a different filename extension. **Example:** *<a.v>* *`include "b.sv"* *module a();* *endmodule* *<b.sv>* *module b();* *logic v1;* *endmodule* *> vericom a.v +systemverilogext+.sv -extinclude* "*b.sv*" is parsed with the SystemVerilog keyword set, syntax, and semantics.
-extractRTL	Automatically recognizes the RTL storage elements. The extracted storage elements are saved into the library.
-f | -file *filename*	Loads an ASCII file containing design source files and additional simulator options.
-F *filename*	Loads an ASCII file containing a specified path to source files and simulator options. Relative paths are used to specify the source files. You cannot use the *-F* and *-path* options simultaneously. The path assigned with the *-path* option is ignored. **NOTE:** This option is for Verilog only.
-forceIncsaveVY	Forces the knowledge database creation for *-v -y* files in each library module resolution to decrease the compilation memory. This option is valid only when the *-incsave* option is specified.
-h | -help	Prints the help message.
-ignore *keyword_argument*	Suppresses error messages associated with the specified keyword argument. The following keyword is supported: *driver_checks* Suppresses error messages about Verilog driver checking.

Options	Explanation
+ignorefileext+ <*extensionname*>	Specifies the file extensions to ignore during compilation. This option applies to both the run file and *verdi* or *vericom* command line. **Example 1: Specify in Run File** Assume a run file contains the following: *//1.f* *+ignorefileext+.vr+vg* *1.v* *1.vr* *1.vg* when it is compiled with *vericom* as follows: *> vericom -f 1.f* the information printed to the console shows that files with the "*.vr*" (*1.vr*) or "*vg*" (*1.vg*) extension are ignored. **Example 2: Specify on the Command Line** Specify the option on the command line similar to the following: *> vericom -f run.f +ignorefileext+.vr+.vg* *> vericom -f run.f +ignorefileext+.vr+.vg*
-ignorekwd_config	Ignores the *config* keyword of Verilog IEEE 1364-2001.
-ignore_macro_redef	Suppresses warning messages for redefined macros.
-incdir *directory*	Specifies an include directory.
-incsave	Specifies to decrease the compilation memory usage.
-L	Specifies the library to search for packages. **Example:** *vericom -sv branch.v -L testlib*
-lib *libName*	Specifies the library to save the design to (the default value is "*work*").
-libmap *filename*	Specifies the library mapping file.
+liborder	Searches for the module definitions of unresolved module instances in the *vericom* command line with the following order: 1) Search the remainder of the library where the unresolved module instances are found. 2) Search through the rest of the libraries. 3) Search again from the first library. For examples, see the **+liborder** option in the *verdi* utility.

Options	Explanation
+librescan	Searches for the module definitions of unresolved module instances by always starting from the first library in the library list specified in the *vericom* command line. For examples, see the **+liborder** option in the *verdi* utility. **NOTE:** This option is not case-sensitive.
+libverbose	Prints a message in the *compiler.log* file to indicate the library file where the instance is resolved when a module is instantiated in the source file or library files. The path of the generated **.lib++/* directory is also saved into the log file.
-logdir *directory*	Specifies the location of the log directory.
-nclib	Specifies to elaborate the precompiled design based on the Cadence NC library *-binding* scheme. With this option, *vericom* looks for *libMap*, *workLibrary*, *viewMap*, and *defaultView* in the *[NC_Sim]* section of the *novas.rc* resource file for compiling Verilog source files into the Verdi Knowledge Database (KDB).
+nlog *filename*	Appends the compile messages to a specified file.
-nlog *filename*	Outputs the compile messages to a specified file.
-ntb_opts ovm[-<*version*>]	Loads the OVM library for compilation. **NOTE:** To compile the external OVM library, first set the *VCS_HOME* or *VCS_OVM_HOME* environment. The *VCS_OVM_HOME* environment variable has higher priority than *VCS_HOME*. If both *VCS_OVM_HOME* and *VCS_HOME* are set, and *ovm.sv* cannot be found in *$VCS_OVM_HOME*, an error is reported. For examples, see the **-ntb_opts ovm** option in the *verdi* utility.
-ntb_opts <rvm|vmm>[-<*version*>]	Loads the VMM library for compilation. **NOTE:** To compile the external VMM library, first set the *VCS_HOME* environment variable. For examples, see the **-ntb_opts <rvm|vmm>** option in the *verdi* utility. For the usage of mixed UVM and RVM libraries, see the Mixed Usage of UVM and RVM Libraries columns in the *verdi* utility.

Options	Explanation
-ntb_opts uvm[-<*version*>]	Loads the UVM library for compilation. The *+define+VCS* option is added automatically in the *vericom* command line. **NOTE:** To compile the external UVM library, first set the *VCS_HOME* or *VCS_UVM_HOME* environment variable. The *VCS_UVM_HOME* environment variable has higher priority than *VCS_HOME*. If both *VCS_UVM_HOME* and *VCS_HOME* are set, and *uvm_pkg.sv* cannot be found in *$VCS_UVM_HOME*, an error is reported. For examples, see the **-ntb_opts uvm** option in the *verdi* utility. For the usage of mixed UVM and RVM libraries, see the Mixed Usage of UVM and RVM Libraries columns in the *verdi* utility.
-nv *filename*	Specifies a library file to use. Modules in this library file are not treated as library cells.
-ny *directory*	Specifies a library directory to use. Modules in this library directory are not treated as library cells.
-onlylog	Dumps the *compiler.log* only without parsing the design.
-ovm[-<*version*>]	Loads the default Verdi OVM library. If *-ovm* and *-ovmhome* are specified at the same time, *-ovm* is ignored. For examples, see the **-ovm** option in the *verdi* utility.
-ovmhome <*path*>	Specifies the OVM installation directory. For examples, see the **-ovmhome** option in the *verdi* utility.
-P *file1 file2 ...*	Specifies PLI table files. **Example:** *vericom -sv -P pli1.a pli2.a test.v*
+pkgdir+<*pkg_name* | *pkg_full_path*>	Loads the specified DesignWare VIP packages as black boxes by referring to the headers located in the release package or in an absolute path. **Examples**: *> vericom +pkgdir+amba <design> ...* Compiles the *amba.f* package located in the default product directory. The default product directory is *<$VERDI_HOME>/etc/kdb/verilog/dwvip/AMBA_PKG*. *> vericom +pkgdir+/home/mydwvip/packages/amba <design> ...* Compiles all **.f* files in */home/mydwvip/packages/amba*.

Options	Explanation
-q	Turns on the quiet mode to suppress the console information of copyright, version, locations of the log directory, resource file, and GUI configuration file.
-quiet	Disables minor (compile and load) messages. Only the total number of errors and warnings are reported.
-realport	Supports only the "wreal" keyword in Verilog-AMS. This option also supports the "`wrealXState" or "`wrealZState" macros for the real x or z states respectively.
-RegPort	When specified, a *reg* declaration before the output declaration for the same signal does not generate an error. **Example:** *module test(alpha,reset_,clk);* *input reset_;* *input clk;* *reg [4:0] alpha;* *output [4:0] alpha;* *reg [4:0] beta;* *always@(posedge clk or negedge reset_) begin* *if(!reset_) alpha <= 5'b0000;* *else alpha <= beta;* *end* *endmodule*
-rcFile *rcFilename*	Specifies the *novas.rc* resource file.
-rep[lace]	With this option, *vericom* removes previously saved Verilog modules in the library before compiling. Without this option, *vericom* updates existing modules in the library by default.
-rmkeyword *keyword*	Downgrades the specified keyword to an identifier.
+rmkeyword+ <*keyword(s)*>	Downgrades the specified keywords to identifiers. You can specify one or more keywords.
-saveOpts	Saves the compilation options into the Verdi Knowledge Database (KDB) library.
-silent	Prevents bells (*Ctrl-G*) from being issued.
-sort_net_instance	Sorts signals and instances alphabetically.
-stdout	Prints compiler messages to *stdout*.

Options	Explanation
-ssv	Library modules in the library file (*-v*) are not tagged as library cells. The state (specified or not specified) of this option in the compiled library is overwritten each time *vericom* runs. The final state (specified or not specified) is saved in the precompiled library.
-ssy	Library modules in the library directory (*-y*) are not tagged as library cells. The state (specified or not specified) of this option in the compiled library is overwritten each time *vericom* runs. The final state (specified or not specified) is saved in the precompiled library.
-ssz	Ignores '*celldefine* compiler directives. The state (specified or not specified) of this option in the compiled library is overwritten each time *vericom* runs. The final state (specified or not specified) is saved in the precompiled library. **NOTE:** If you load the symbol library, the cell in symbol library has higher priority than a user-defined module.
-sv	Supports the selected SystemVerilog syntax.
-sverilog	Supports the selected SystemVerilog syntax.
-sv_pragma	Compiles the SystemVerilog assertion code that follows the *sv_pragma* keyword in a comment.
-syntaxerrormax *number*	Specifies the maximum number of syntax errors to stop parsing. If the syntax errors exceed this number, the parser stops parsing the remaining files. **NOTE:** When the *NOVAS_VERILOG_SYNTAX_MAX_ERROR* environment variable is set to a negative number, *vericom* does not stop even if there are syntax errors.
-u[ppercase]	Changes all identifiers to uppercase.
-useius	Uses IUS style parsing/elaboration. **NOTE:** When more than one of the *-useius*, *-usemti*, and *-usevcs* options are specified, only the first option specified is recognized; subsequent options are ignored.
-usemti	Use MTI style parsing/elaboration. **NOTE:** When more than one of the *-useius*, *-usemti*, and *-usevcs* options are specified, only the first option specified is recognized; subsequent options are ignored.
-usevcs	Use sVCS style parsing/elaboration. **NOTE:** When more than one of the *-useius*, *-usemti*, and *-usevcs* options are specified, only the first option specified is recognized; subsequent options are ignored.

Options	Explanation
-uvm[-<*version*>]	Loads the default Verdi UVM library. If *-uvm* and *-uvmhome* are specified at the same time, *-uvm* is ignored. For examples, see the **-uvm** option in the *verdi* utility.
-uvmhome <*path*>	Specifies the UVM installation directory. For examples, see the **-uvmhome** option in the *verdi* utility.
-v95	Supports the Verilog IEEE 1364-1995 standard.
+v2k	Supports the Verilog IEEE 1364-2001 standard.
-vc	Supports the DirectC syntax.
-view *viewName*	Specifies NC viewName (valid if *-nclib* is specified). This provides the same function as *ncvlog -view view_name* to use the specified view name for the design units of the input Verilog source files. **Example:** *vericom -nclib -view novas ../src/novas.v* All design units in *../src/novas.v* have the view name *novas*. **NOTE:** Using the command line option, *-view*, overrides *defaultView* and *viewMap* variables in the *novas.rc* resource file.
-wcFile	Supports the wildcard file list in a *run.f* file. Specify this option before *-f*. **NOTE:** 1) The /*...*/ comment syntax is not supported in the *run.f* file. 2) When this option is specified and the *NRUN_CSTYLE_COMMENT* environment variable is set to 1, the "*" is recognized as part of the comment. When the *NRUN_CSTYLE_COMMENT* environment variable is set to 0, the asterisk (*) character is recognized as a wildcard.
-work *libName*	Specifies NC libName (valid if *-nclib* specified). This provides the same function as *ncvlog -work library_name* to use the specified library name for the design units of the input Verilog source files. **Example:** *vericom -nclib -work novas -view novas -f ../src/novas.f* All design units in the *../src/novas.f* file are compiled into the *novas* library and have a view name *novas*. **NOTE:** Using the command line option, *-work*, overrides the *workLibrary* and *libMap* variables in the *novas.rc* resource file.

All simulator options are accepted by `vericom`. The options it needs to compile the design are used and the rest options are ignored. For a complete list and details, see the documentation for the simulator. The following simulator command-line options are also commonly used by `vericom`.

Options	**Explanation**
-f <*filename*>.f	Loads an ASCII file containing design source files and simulator options.
-v <*filename*>	Modules in the specified file are treated as library cells. This option is overwritten if the *-ssv* option is specified on the *vericom* command line during compile or if the *-ssv* option is specified on the Verdi command line during import.
-y <*directoryname*>	Modules in the specified directory are treated as library cells. For details, see the *-y* option in the *verdi* utility. This option is overwritten if the *-ssy* option is specified on the *vericom* command line during compile or if the *-ssy* option is specified on the Verdi command line during import.
+define+<*macro*>	The *+define* option is used to specify macros. If the macro is also defined in the source code, it is overridden by this option.
+incdir+<*directoryname*>	Specifies the search path for files used by the *‘include* statement.
+libext+<*extensionname*>	Specifies the file extensions for Verilog library files. Also, see the *-y* option in the *verdi* utility.
+libverbose	Prints a message on the screen and in the *vericomLog/compiler.log* file to indicate the library file where the instance is resolved when a module is instantiated in the source file or library files. The path of the generated *lib++* is also saved into the log.

Options	Explanation
+systemverilogext+<*extensionname*>	If a consistent file extension is used for SystemVerilog files, this option is used to specify the file extension. Use it in place of the *-sv* option. **Example:** *verdi -f run.f +systemverilogext+.sv+.SV* **NOTE:** When none of the language options (*+v2k*, *-v95*, *-2001*, *-sv*, and *-sverilog*) are specified, the files specified by *+systemverilogext* is parsed with the SystemVerilog parser. However, when any of the language options (*+v2k*, *-v95*, *-2001*, *-sv*, and *-sverilog*) are specified, all files are parsed with the related language parser (for example, Verilog 2001 parser for *-2001*). The files specified by *+systemverilogext* may have keywords compatible with the SystemVerilog standard.
+v95ext+<*extensionname*> | +verilog1995ext+<*extensionname*>	Specifies the file extension for Verilog 1995 files. **NOTE:** When none of the language options (*+v2k*, *-v95*, *-2001*, *-sv*, and *-sverilog*) are specified, the files specified by *+v95ext* are parsed with the Verilog 1995 parser. However, when any of the above language options are specified, all files are parsed with the related language parser (for example, System Verilog parser for *-sv* or *-sverilog*). The files specified by *+verilog1995ext* or *+v95ext* are allowed to have keywords compatible with the Verilog 1995 standard.

Options	Explanation
+verilog2001ext+<*extension name*>	If a consistent file extension is used for Verilog 2001 files, this option is used to specify the file extension. Use it in place of the *-2001* option. **Example:** *verdi -f run.f +verilog2001ext+.v2k* **NOTE:** When none of the language options (*+v2k*, *-v95*, *-2001*,*-sv*, and *-sverilog*) are specified, the files specified by *+verilog2001ext* are parsed with the Verilog 2001 parser. However, when any of the above language options are specified, all files are parsed with the related language parser (for example, System Verilog parser for *-sv* or *-sverilog*). The files specified by *+verilog2001ext* are allowed to have keywords compatible with the Verilog 2001 standard. **Example:** *> vericom -sverilog +verilog2001ext+.v 1.v 2.sv* where, *1.v* and *2.sv* are as follows: *//1.v* *module test;* *reg logic; // logic is a SystemVerilog keyword* *...* *endmodule* *//2.sv* *module test;* *logic test; // logic is a SystemVerilog keyword* *...* *endmodule* Both *1.v* and *2.sv* are parsed using the SystemVerilog parser (affected by the *-sverilog* option). However, the parser validates the *1.v* keyword set against the Verilog 2001 keyword set (affected by the *+verilog2001ext* option) instead of the SystemVerilog keyword set. In this example, the compilation is successful as 'logic' is not a keyword in Verilog 2001.
+spiceExt+<*extensionname*>	Specifies the file extensions for HSPICE files. This option is not case-sensitive, but the file extension is case-sensitive (for example, *+spiceExt+.sp+.SP*). **Example:** *verdi -f run.f +spiceExt+.sp+.hsp*

Variables for NC Binding Scheme in novas.rc

For details on the NC library mapping variables, see the *NC-Like Library Mapping Scheme for Verilog Designs* section of *Appendix B: Customizing Verdi* .

Verilog-2001 Configuration

Configuration is a Verilog-2001 construct. Configuration constructs consist of two parts: library mapping and configurations. The library mapping is used at compilation to determine which library to compile the source. The configuration is a set of rules to apply when searching for a design unit to bind to an instance.

The Verdi platform treats library mapping and configuration as two independent constructs. Library mapping is not part of a Verilog design. It is treated as a library setting for `vericom` only. A configuration is treated as part of the Verilog design, like VHDL configurations.

Library Map

Library mapping is only effective when using the `vericom` utility to compile the design. The library map file is specified using the `-libmap` command-line option.

`-libmap <filename>`: Specifies the library mapping file.

`vericom` is used to compile the design into the libraries specified in the library maps. You must explicitly specify the library map file. For example:

```
[test.map]
library rtlLib *.v
library gateLib *.vg
include lib2.map

> vericom -libmap test.map *.v *.vg
```

Configuration

A configuration is treated as a new kind of Verilog design unit. If the Verilog configuration is used, the configuration must be compiled into the library and the design imported by explicitly specifying the configuration as the top module. For example:

```
[gate.cfg]
config mux_cfg_gate;
design work.mux_tb;
default liblist rtllib
instance mux_tb.dut.i1 liblist gatelib
endconfig
```

```
> vericom -2001 gate.cfg
> verdi -top mux_cfg_gate
```

NOTE:

1. Library mapping and configurations can exist in the same or different files.
2. Multiple library map files are used but each file must be preceded by the `-libmap` option.
3. The library declaration cannot be overridden. For example, only the first library declaration is effective (same as ModelSim):
   ```
   library rtllib ./mux81.v;
   library rtllib ./mux41_rtl.v;
   library rtllib ./mux21.v;
   ```
4. Module resolution priority:
   a. Instance-clause has higher priority than cell-clause.
   b. The `-Lf` option has higher priority than default `liblist` (following the convention of ModelSim).
   c. The default `liblist` has higher priority than `NC `uselib lib=`, (following the convention of NC).

Limitation:

1. Library mappings are only used in the `vericom` utility.
2. Configurations must be compiled into a library and be loaded (used) from a library. You cannot use a configuration when importing the design into the Verdi platform from the source files directly.
3. Configurations cannot be applied to instance arrays and generated instances.
4. Elaboration for the following coding style is not supported:
   ```
   cell libA.cpu use top.myCPU
   ```

vhdlcom

`vhdlcom` is a batch mode compiler for VHDL. `vhdlcom` checks and saves the VHDL design into a library.

NOTE:

1) Before VHDL compilation, you must copy the `novas.rc` resource file to the local working directory and then manually modify according to the library mapping.
2) The library settings are loaded from the `novas.rc` resource file unless the `manage.rc` resource file is specified. For details on using the `manage.rc` resource file, see *Appendix B: Customizing Verdi*.

Usage:

```
vhdlcom [Options...] [VHDL files]
```

Options	Explanation
-2000	Supports the VHDL IEEE 1076-2000 standard.
-87	Supports the VHDL IEEE 1076-1987 standard.
-93	Supports the VHDL IEEE 1076-1993 standard.
-applog	Appends compile messages to the log file. The default is to overwrite previous compilation messages.
-comment_transoff_regions	Skips the source code between *translate_off* and *translate_on* pragmas. For examples, see the **-comment_transoff_regions** option in the *verdi* utility description.
+disable_message+<*message_serial_numbers|error|warning*>	Suppresses the specified message(s), all error messages, or all warning messages. Use a plus (+) symbol for different message type combinations. For details about each message number and its description, see *Appendix F: Message Table*. **Examples:** Suppress the specified message(s) *+disable_message+C00288* *+disable_message+C00288+C00266* Suppress the specified message and all warning messages *+disable_message+C00288+warning* Suppress the specified message, all error messages and warning messages *+disable_message+C00288+error+warning*
-disable_protect	Analyzes the source code between *--begin_protected* and *--end_protected* pragmas.

Options	Explanation
-f | -file | -fv *filename*	Specifies the run file. If there are several VHDL files to compile, pack them sequentially into a file, then pass this file to *vhdlcom*. **Example:** If three source files called *v1.vhd*, *v2.vhd*, and *v3.vhd* exist, create a file called *run.f* (or another desired name) and create its contents as follows (note that each file name must occupy its own line): *v1.vhd* *v2.vhd* *v3.vhd* **NOTE:** To add comments to a *vhdl -f* or *-fv filename*, add one of the following notations before the comment text: -- *;* -- *//* -- */* ... **/*
-h | -help	Prints the help message.
-ignoreSameEntity	Ignores code for entities that are previously parsed. **Example:** *vhdlcom -ignoreSameEntity blk.vhd blk_new.vhd* In this example, the entity described in *blk_new.vhd* is ignored as it was initially parsed in *blk.vhd*.
-just *eapbc*	Compiles only the specified type of design units (that is, *e*: entity, *a*: architecture, *p*: package, *b*: package body, *c*: configuration). For example, *-just e* only compiles entities and *-just ea* only compiles entities and architectures.
-lib *libName*	Specifies the library to save the design. The default library is *work*.
-libcell	Compiles all entities (architectures) the same way as the symbol library.
+libverbose	Prints a message on the screen and in the *compiler.log* file to indicate the path of the generated *lib++* file.
-logdir *logDirectory*	Specifies the location of the log directory.
-nlog *filename*	Outputs the compile messages to the specified file.
+nlog *filename*	Appends the compile messages to the specified file.
-nochecking	Skips semantic error checking.
-onerrorstop	Stops compiling the design if an error occurs.
-onfatalerrorcontinue	Continues to compile the design if a fatal error occurs.

Options	Explanation
-prelib yes | no	When specifying yes, *vhdlcom* loads two standard packages (*standard* and *textio*) before parsing the design. Its default value is *yes*.
-psl_inline *filename*	Extracts inline-PSL code and stores it in the specified file.
-psl_inline_append *filename*	Extracts inline-PSL code and appends it in the specified file.
-q	Turns on the quiet mode to suppress the console information of copyright, version, locations of the log directory, resource file, and GUI configuration file.
-quiet	Disables minor (compile and load) messages. Only the total number of errors and warnings are reported.
-rcFile *rcFilename*	Specifies the resource file. Its default value is *novas.rc*.
-saveOpts	Saves the compilation options into the Verdi Knowledge Database (KDB) library.
-silent	Prevents bells (**Ctrl-G**) from being issued.
-skip *eapbc*	Directs the compiler to skip the specified type of design units (that is, *e*: entity, *a*: architecture, *p*: package, *b*: package body, *c*: configuration). For example, *-skip e* does not compile entities and *-skip ea* does not compile entities and architectures.

Options	Explanation
-smartorder	Compiles in order independent mode. **Limitations**: 1) You cannot use the *-smartorder* option with the *-skip/just* and *-comment_transoff_regions* options. Because the *-skip/just* and *-comment_transoff_regions* options may hide design content. Also, these may ignore some design units or use clauses that would change dependencies and affect the results of using *-smartorder*. Therefore, such usage is not supported. If *-smartorder* is used with these two options, they are ignored automatically. 2) The library dependence is not considered by the *-smartorder* option. The dependencies are handled by users. **Example:** The *main_des (a.vhd, b.vhd)* library depends on the *pkg_pram (p.vhd, q.vhd)* library. One correct usage is: 1) *vhdlcom -smartorder p.vhd q.vhd -lib pkg_pram* *2) vhdlcom -smartorder a.vhd b.vhd -lib main_des* An alternative usage is to set the library section in the *rc* file as: *Pkg_pram = wk* *Main_des = wk* Then, compile with the following: *vhdlcom -smartorder a.vhd b.vhd p.vhd q.vhd -lib wk*
-smartscript *filename*	Generates a compilation script file with the specified name.
-stdout	Shows the compile messages to *stdout*.
-sup_sem_error	Suppresses semantic errors.
-topdown -src *filename*	Enables top-down compilation and specifies the top-level file.
-vhdl08	Supports the VHDL IEEE 1076-2008 standard.
-work *libname*	Specifies the library to save the design. The default library is *work*.

vlmap

`vlmap` is a function which automatically updates the library mapping to the `novas.rc` resource file.

NOTE: The library settings are loaded from the `novas.rc` resource file unless the `manage.rc` resource file is specified. For details, see *Appendix B: Customizing Verdi*.

Usage:

```
vlmap [Options]
```

Options	Explanation
-del	Deletes the mapping specified by <*logical_name*>.
-field *sectionName string*	Specifies the section name in the *novas.rc* resource file to write the specified string. The section name and string are separated by spaces. The specified section name must match the section name in the *novas.rc* resource file exactly; otherwise, *vlmap* creates a new section. The *-field* option is specified only once for each execution of *vlmap*. **Example:** *vlmap -field MDT SPS_ARRAY DP_M01* *vlmap -field MDT MY_ARRAY DP_M02* After *vlmap* is executed twice, the specified field strings are written into the *novas.rc* resource file as follows: *[MDT]* *SPS_ARRAY = DP_M01* *MY_ARRAY = DP_M02*
-h | -help	Prints the help message.
<logical_ name>	Specifies the logical name of the library to be mapped.
<path>	Specifies the directory pathname of the library to be mapped.
-rcFile *rcFilename*	Specifies the *novas.rc* resource file.

Example:

To add a library mapping, use the following:

```
vlmap vital work
vlmap work work.lib++
```

To delete a library mapping, use the following:

```
vlmap -del vital
```

To query a current library mapping, use the following:

```
vlmap
```

NOTE: This function is based on the current working directory. If the current directory does not have the `novas.rc` resource file, it creates one in the current directory.

Conversion

The following utilities are used for FSDB file manipulation:

- *esdb2txt*
- *esdbdebug*
- *fsdb2ns*
- *fsdb2saif*
- *fsdb2vcd*
- *fsdbdebug*
- *fsdbedit*
- *fsdbexpand*
- *fsdbextract*
- *fsdbjoin*
- *fsdbmangle*
- *fsdbmerge*
- *fsdbrecover*
- *fsdbreport*
- *merge_multi_swp_fsdb*
- *nce2report*
- *txt2fsdb*
- *txt2uddb*
- *vcd2fsdb*
- *vfast*
- *xloc*
- *xml2fsdb*

esdb2txt

`esdb2txt` is a utility to generate a plain text Essential Signal list from the Essential Signal Database (ESDB) for the emulation flow.

Usage:

```
esdb2txt esdb_filename [Options]
```

Options	Explanation
-dumpclk none(n) | all(a)	Specifies whether to dump all (a) or none (n) of the clock and reset signals. Its default value is *all*. This option is optional.
esdb_filename	Specifies the ESDB directory name. For example, if *ABC.esdb++* is the ESDB directory name, *ABC* is the ESDB name. This option must be specified.
-ff_only 0|1	Dumps register signals only. Its default value is *0*. This option is optional.
-h | -help	Prints the help message.
-mda 0|1	Dumps Essential Signals that are array signals. Its default value is *1*. This option is optional.
-plain_eslist *filename*	Dumps Essential Signals in a plain list format to the file specified. This option must be specified.
-scope N *scopeName*	Specifies a depth-scope pair to be made visible.You can specify only one pair. All Essential Signals in the associated scopes are included except boundary ports. This option is optional. **NOTE:** The *-scope* option may only be specified once. If it is used multiple times (for example, -scope N1 *name1* -scope N2 *name2*), only the last pair (that is, N2 *name2*) is used.
-struct 0|1	Dumps Essential Signals that are struct or record signals. Its default value is *1*. This option is optional.

Example

1. Generate the `esdb.esdb++` directory:

   ```
   > esa -f run.f -path /qa/usddts_case/cpu/src -db esdb
   ```

   Then, generate the `plist` file that contains a plain ES list.

   ```
   > esdb2txt esdb -plain_eslist plist -dumpclk all
   ```

   `esdb` is the input that indicates the ESDB directory.

esdbdebug

`esdbdebug` is a utility to dump information in the Essential Signal Database (ESDB) for debugging.

Usage:

```
esdbdebug -db esdb_filename [Options]
```

Options	Explanation
-h | -help	Prints the help message.
-info	Dumps header information from the ESDB, including the ESDB version, working scope, ESDB options, cellModel, undefined modules, sequential symbol libraries, and combinational symbol libraries.
-scope [*full_hier_scope_name1... namen*]	Dumps Essential Signals in the ESDB. If one or more scopes are specified, only dump for the specified scopes; otherwise, dump for all scopes with a scope-based view.
-signal *name1 ... namen*	Dumps the attribute for the specified signals. Specify at least one signal.
-stats [*full_hier_scopename1... namen*]	Dumps the ESDB statistics for all scopes or the specified scopes. If one or more scopes are specified, only dump for the specified scopes; otherwise, dump for all scopes with a scope-based view.
-top_n_var *Number*	Dumps the specified number of signals with the most value changes.
-with_computed_signal	Includes computed signals. This option should work with the *-top_n_var* option.
esdb_filename	Specifies the ESDB directory name. For example, if *ABC.esdb*++ is the ESDB directory name, *ABC* is the ESDB name. This option must be specified.

Example

```
esdbdebug -db es -scope
```

Dumps Essential Signals in the ESDB for all scopes with a scope-based view.

fsdb2ns

`fsdb2ns` is a utility to translate an EPIC FSDB file into the ASCII format and convert a digital type FSDB file into the Piecewise Linear (PWL) format. The output format (`-fmt`) is PWL depending on the specified format.

In the PWL format, some or all signals with delays are extracted to a new ASCII file. Note that only two types of the signals are handled:

`v` - Analog voltage (PWL, voltage resolution)

`i` - Instantaneous current (Piecewise Constant (PWC), current resolution)

Other signals, even though they are specified with `-s`, are ignored. The value changes of instantaneous current signals, which are PWC signals, are converted to the PWL format by only keeping the first value change in the same time step.

`fsdb2ns` only keeps the upper-most signal of all equivalent signals when converting all signals (not a signal list), and converts all signals when a signal list is specified.

Example:

Time	Value	Signal Value
0	0	keep
0	1	ignore
1	1	keep
1	2	ignore
2	2	keep

Note that all time values, including time steps and delays, are displayed in pico seconds.

Usage:

```
fsdb2ns fsdb_filename [-o filename] [-fmt PWL|TBL|out]
 [-s signal] [-delay delay_in_sec_signal] [-vih voltage_value
"signal"| "all_signals"] [-vil voltage_value
"signal"|"all_signals"] [-eval] [-repeat]
[-tr] [-tf] [-f config_file] [-info] [-v][-ignore_2G]
```

Options	Explanation
-delay *delay_in_sec_signal*	Specifies signal delay in pico seconds. This applies to the PWL format only. You can specify delay for a signal, no matter whether the signal is previously specified by *-s* or not. In case of conflicts, the latter overrides the former. See the following examples: *fsdb2ns epic.fsdb -o epic.pwl -s "x1.i(g)" -delay 0.01 "x1.i(g)"* extracts signal *x1.i(g)* with a specified delay of *.01ps*. *fsdb2ns epic.fsdb -o epic.pwl -delay 0.01 "x1.i(g)"* extracts all signals and specifies a *.01ps* delay for the *x1.i(g)*. If *x1.i(g)* signal does not exist in the FSDB file, the delay option is ignored.
-eval	When specified, transfer an expression (for example, "1000ps-tr") in PWL to a time (for example, 998ps, tr = 2ps) based on the specified value of rising time (*-tr*) or falling time (*-tf*).
-f *config_file*	Specifies the configuration file. The file can contain *-s, -delay, -vih* and *-vil*. Other options are not recognized. A configuration file, *sig_list.f*, may look like: *-s x1.v(s20)* *-delay 3 x1.v(s20)* *-s v(s22)* *-s x2.v(s23)* *-delay 4 v(s22)* If the *-delay* option in the command line conflicts with the *-delay* option in the configuration file, the rule of the latter overriding the former still applies. For example, The delay of *x1.v(s20)* is specified both in the command line and the configuration file as follows: *fsdb2ns epic.fsdb -o epic.pwl -f sig_list.f -delay 100 "x1.v(s20)"* The delay value is *100* as *-delay* is specified after the configuration file.
-fmt PWL|TBL|out	The format can be (*PWL*|*TBL*). *PWL* indicates the piecewise linear (PWL) format. If this option is not specified, the default value is the *PWL* format.
-info	Prints a summary and supports information about the FSDB file.
-o *filename*	Specifies the output file name of the extracted FDSB file. If not specified, the result is printed to *stdout*.

Options	Explanation
-repeat	Specifies this option to calculate the PWL repeat parameter (R=time). Use this option with *-eval*. **Example:** *Vsystem.En_BC system.En_BC 0 PWL DELAY=0ps* *+ 0ps 0* *+ 799998ps 0* *+ 800000ps 3* *+ 899997ps 3* *+ 900000ps 0* *+ 1399998ps 0* *+ 1400000ps 3* *+ 1499997ps 3* *+ 1500000ps 0* *+ 1999998ps 0* ... After *-repeat* is specified, it is shown as follows: *Vsystem.En_BC system.En_BC 0 PWL R=800000ps DELAY=0ps* *+ 0ps 0* *+ 799998ps 0* *+ 800000ps 3* *+ 899997ps 3* *+ 900000ps 0* *+ 1399998ps 0* *+ 1400000ps 3* The value repeats as follows: *3->3->0->0->3->3->0->0 ...* The time range repeats as follows: *2->99997->3->499998->2->99997->3->499998 ...*
-s *signal*	Specifies a full hierarchical signal name whose value changes are extracted. This applies to the PWL format only. The scope delimiter is '.'. For the signal name, see *nWave*'s displayed name and replace '/' with '.'. The wildcard character (*) is supported. For example, if "a.b.*" is specified, all the signals under the scope *b* and subscope of *b* are processed. If *-s* is not specified, all signals are extracted. For example, to extract all signals in the FSDB file in the PWL format, specify a signal list with multiple *-s*. For example, to extract two signals in PWL format, use the following: *fsdb2ns epic.fsdb -o epic.pwl -s "x1.i(g)" -s "v(out)"* **NOTE:** For system tasks or system functions, when a scope or signal name begins with '$' (for example, *$root*), it denotes a variable in the Unix shell, and gives a warning that you cannot obtain the variable root value. To avoid this error usage, use a single quote (') to replace double quotes ("). For example, *fsdb2ns rtl.fsdb -s '$root.En_a' -o test.pwl*
-tr	Specifies the rising time (transitioning from 0 to 1) to calculate the expression in PWL.

Options	Explanation
-tf	Specifies the falling time (transitioning from 1 to 0) to calculate the expression in PWL.
-v	Turns on the verbose mode.
-vih *voltage_value* “*signal*”| “*all_signals”*	Specifies the high voltage value in voltage which is mapped to the digital value, *FSDB_BT_VCD_1*. If “*all_signals*” is given for the signal name, *voltage_high_value* is applied to all the signals.
-vil *voltage_value* “*signal*”| “*all_signals”*	Specifies the low voltage value in voltage which is mapped to the digital value, *FSDB_BT_VCD_0*. If “*all_signals*” is given for the signal name, *voltage_low_value* is applied to all the signals.
-ignore_2G	Transforms FSDB if signal count of input FSDB is over 2G. Transformation fails if the count of user interested signal is over 2G.

Example 1:

```
fsdb2ns epic.fsdb -s "x1.i(g)" -delay 0.01 "x1.i(g)"
```

Extracts the `x1.i(g)` signal whose type is PWL and specifies the delay value of `x1.i(g) 0.01 ps` to `stdout`.

Example 2:

```
fsdb2ns verilog.fsdb -vil 0 "system.i_cpu.i_PCU.IDB[7:0]" \
-vih 3 "system.i_cpu.i_PCU.IDB[7:0]" -eval -tr 2ps -tf 3ps \
-s "system.i_cpu.i_PCU.IDB[7:0]" -o out.pwl
```

Extracts the `system.i_cpu.i_PCU.IDB[7:0]` signal and specifies its high and low voltage value and the rising and falling time to calculate the expression of the PWL output file.

Example 3:

```
fsdb2ns verilog.fsdb -vil 0 "all_signals" -vih 3 "all_signals" \
-eval -tr 2ps -tf 3ps -s "system.i_cpu.i_PCU.*" -o out.pwl
```

Extracts the `system.i_cpu.i_PCU.*` signal, specifies the high and low voltage value where the high and low voltage value is applied to all signals, and specifies the rising and falling time to calculate the expression to the PWL output file.

Example 4:

Extracts signals and specifies delay values according to the configuration file.

Suppose that `sig_list.f` looks like as follows

```
-s x1.v(s20)
-delay 3 x1.v(s20)
-s v(s22)
-s x2.v(s23)
-delay 4 v(s22)
```

```
fsdb2ns epic.fsdb -o epic.pwl -f sig_list.f
```

Executing the above command extracts signals `x1.v(s20)`, `v(s22)`, and `s x2.v(s23)` in the `sig_list.f` file. Specify the delay value of `x1.v(s20) 3ps` and `v(s22) 4 ps`.

fsdb2saif

`fsdb2saif` is a utility to convert an FSDB file into a backward SAIF format.

Usage:

```
fsdb2saif fsdb_filename [-o saif_filename] [-vhdl]
[-keep_task] [-bt time[uint]] [-et time[unit]
[-s]][-merge_same_scope][-to_stdout][-ignore_2G]
```

Options	Explanation
-bt time[unit]	Specifies the start time of the translation. If omitted, the start time of the FSDB is used. The time unit is *Ms*|*Ks*|*s*|*ms*|*us*|*ns*|*ps*|*fs*. The default time unit is *ns*.
-et time[unit]	Specifies the end time of the translation. If omitted, the end time of the FSDB is used. The time unit is *Ms*|*Ks*|*s*|*ms*|*us*|*ns*|*ps*|*fs*. The default time unit is *ns*.
-flatten_genhier	Eliminates blocks created by *generate* constructs and appends the block name into the signal name as a prefix.
-h | -help	Prints the help message.
-ignore_2G	Allows you to read the FSDB signals for which signal count is over 2G. Note: The utility stops when reading the FSDB that is more than 2G signals. By using this option you can force the utility to read the FSDB. But you might notice that there is a huge memory usage.
-keep_task	Leaves the *vcd_task/vcd_function/vhdl_procedure/vhdl_function* scopes and their sub-scopes and signals.
-merge_same_scope	Merges the duplicated scopes and signals.
-o *saif_filename*	Specifies the output SAIF file name.
-s	Specifies the scope and its sub-scopes to convert to the output SAIF file. This option is not used to specify a signal name. Wildcard characters are not supported.
-to_stdout	Redirects the content of the output SAIF file to `stdout` and the generation of the output SAIF file is disabled even if option `-o` is specified.
-vhdl	Convert 9 MVL to 4 states value with the following table instead of skipping VHDL signals: *U X 0 1 Z W L H - (9 MVL)* *X X 0 1 Z X 0 1 X (4 states)*

Example 1:

```
fsdb2saif verilog.fsdb -o verilog.fsdb.saif
```

Converts an FSDB file into a backward SAIF format.

Example 2:

```
fsdb2saif verilog_and_vhdl.fsdb -vhdl
```

Converts an FSDB file into a backward SAIF format with 9 MVL to 4 states conversion for VHDL signals.

Example 3:

```
fsdb2saif verilog.fsdb -bt 10ps -et 1000ps -o verilog.fsdb.saif
```

Converts an FSDB file into a backward SAIF format with a specified time range.

fsdb2vcd

`fsdb2vcd` is a utility to translate an FSDB file into the VCD format. The data types of EVCD, MDA, and structure are supported.

NOTE: EVCD conversion is only possible if the data is dumped into the FSDB file. The only way this occurs is if the FSDB file is converted from an original EVCD file or if it is created with the FSDB writer API and a custom application performs the encoding. The object files for FSDB dumping do not dump EVCD signals directly.

Usage:

```
fsdb2vcd fsdb_fname [-o vcd_file_name] [Options] or
fsdb2vcd fsdb_fname -summary
```

Options

Options	Explanation
-bt time[unit]	Specifies the start time of the translation. If omitted, the start time of the FSDB is used. The time unit is *Ms*|*Ks*|*s*|*ms*|*us*|*ns*|*ps*|*fs*. The default time unit is *ns*.
-compacttree	Combines different signals in the same level to the same scope.
-consolidate_bus	When specified, single bus elements are consolidated to bus signals in the generated VCD file. **NOTE:** You cannot use this option with the *-s* option.
-et time[unit]	Specifies the end time of the translation. If omitted, the end time of the FSDB is used. The time unit is *Ms*|*Ks*|*s*|*ms*|*us*|*ns*|*ps*|*fs*. The default time unit is *ns*.

Options	Explanation
-f *config_file*	The *config_file* is a Tcl file, which is used to specify the *-s*, *-bt*, and *-et* options. If the options are used both in the command line and *config_file*, the command-line options override those in *config_file*. The Tcl commands are as follows: *fuSetSignal*: For specifying the signal list (*-s*) *fuSetTimeRange*: For specifying the time range (*-bt, -et*) **Example of a configuration file**: *fuSetSignal {system/clock}* *fuSetSignal {system/reset}* *fuSetSignal {system/addr}* *fuSetSignal {system/r_w}*
-flat_bus	When specified, bus signals are flattened to scalar signals in the VCD file. For example: *fsdb2vcd verilog.fsdb -flat_bus -o test.vcd* The bus[7:0] is written as *bus[7], bus[6], bus[5], bus[4], bus[3], bus[2], bus[1], bus[0]*.
-keep_enum	Dumps enum-type signals as integer-type signals. This option is only for VHDL designs.
-keep_escaped_ symbol	Do not skip the escaped back slash (\) before a variable name or a scope name.
-keep_last_time	Dumps the end time tag of the simulation. **NOTE:** To keep the full time in the VCD file, this option is specified because the VCD time stops at the last transition by default.
-level n	Translates signals from the specified scope and its *n* descendants. Use this option in conjunction with *-s*.
-memory_will_ alloc_in_mega *number*	Specifies the memory size, in Megabytes, to use to execute the *fsdb2vcd* utility. **NOTE:** When the memory size specified is not sufficient, the *fsdb2vcd* utility calculates the minimum size required to execute and then use this value.
-nocase	When included, the mapping of signal names is not case-sensitive. Use this option in conjunction with *-s*.
-o	Specifies the output file name for the translated VCD file.
-s	Specifies the signals or scopes whose value changes are extracted. **NOTE:** You cannot use this option with the *-consolidate_bus* option. **NOTE:** For system tasks or system functions, when a scope or signal name begins with '$' (for example, *$root*), it denotes a variable in the Unix shell, and gives a warning that the variable root value cannot be obtained. To avoid this error usage, use a single quote (') to replace double quotes ("). For example: *fsdb2vcd rtl.fsdb -s '/$root/En_a' -o test.vcd*

Options	Explanation
-seq	When specified, the value change in the VCD file is written based on the sequence number.
-summary	Generates a summary for the FSDB without performing the translation.
-sv	Supports SystemVerilog formats which means structs are converted to VCD's fork/join blocks. Other new SystemVerilog data types and constructs are mapped by the object files for FSDB dumping.
-v	Turns on the verbose mode.
-vcs	Converts a variable name to the VCS simulator output format (with an extra space inserted between the variable name). **Example:** *tx[1] -> tx[1]* *tx[1][2][3] -> tx[1][2] [3]* *tx[1][1:7] -> tx[1] [1:7]*
-verilog | -vhdl	The specified scope is in the Verilog or VHDL format. Use this option in conjunction with the *-s* option.
-ignore_2G	Transforms FSDB if signal count of input FSDB is over 2G. Transformation fails if the count of user interested signal is over 2G.

Example:

1. Translate all signals.
   ```
   > fsdb2vcd verilog.fsdb -o output.vcd
   ```
2. Translate signals within a specified scope and time range.
   ```
   > fsdb2vcd verilog.fsdb -s /system/i_cpu -level 1 -o scope.vcd
   ```
3. Generate a summary for the FSDB without performing the translation.
   ```
   > fsdb2vcd verilog.fsdb -summary
   ```

fsdbdebug

`fsdbdebug` is a utility to dump the FSDB information and generate a log file for debugging.

Usage:

```
fsdbdebug [Options] fsdb_fname
```

Options	Explanation
-allvc	Dumps all value changes.
-analysis	Analyzes the FSDB file for version 5.0 or higher.
-data_type	Dumps the data type definitions.
-dump_mda	Dumps the signals in the FSDB file if the signal is a multiple dimension array and displays 2D-array bus signals available in the FSDB file.
-h | -help	Prints the help message.
-hier_tree	Dumps design trees in a full hierarchical fashion.
-info	Dumps the general information of the FSDB file.
-scope	Dumps scopes only.
-show_prop_stat	Dumps the property statistics table.
-top_n_var *Number*	Dumps the specified number of signals with the most value changes. When set to 0, all the value change counts and hierarchy names are dumped. If the reported signal count is less than 10485760 (10 x 1024 x 1024), the output is sorted by value change count in the descending order. If the reported signal count is larger than 10485760, the output is not sorted.
-tree	Dumps design trees.
-vc	Dumps value changes of the variable specified by *vidcode*.
-vc_count	Dumps the variable change count of a variable.
-vidcode	Specifies the variable *idcode*.
-with_computed_signal	Includes computed signals. This option should work with the *-top_n_var* option.
-show_glitch	Displays the list of nodes with glitches and the time when those glitches occurred. This utility can capture the glitches if it is enabled during the simulation.
-glitch_start_xtag	Specifies the start time for *-show_glitch*

-glitch_end_xtag	Specifies the end time for *-show_glitch*
fsdb_fname	Specifies the FSDB file name.

Example:

1. Dump the design hierarchy.
   `fsdbdebug -tree fsdb_fname`
2. Dump the design hierarchy in a full hierarchical fashion.
   `fsdbdebug -hier_tree fsdb_fname`
3. Dump scopes (excluding variables) in the design hierarchy only.
   `fsdbdebug -scope fsdb_fname`
4. Dump the data type definitions, if any.
   `fsdbdebug -data_type fsdb_fname`
5. Dump the value changes of a variable (`idcode = 3`).
   `fsdbdebug -vc -vidcode 3 fsdb_fname`
6. Dump the value change count of a variable (`idcode = 3`).
   `fsdbdebug -vc_count -vidcode 3 fsdb_fname`
7. Dump the value change count of all variables.
   `fsdbdebug -allvc fsdb_fname`
8. Dump general information about the FSDB file.
   `fsdbdebug -info fsdb_fname`
9. Dump the signals in the FSDB file if the signal is a multiple dimension array.
   `fsdbdebug -dump_mda fsdb_fname`

Example for -tree and -hier_tree

If the `v1` variable exists below the `Top` scope, then `fsdbdebug -hier_tree` returns its full hierarchical name of `Top.v1` (where, '.' is the delimiter), and `fsdbdebug -tree` only returns the variable name of `v1`.

fsdbedit

`fsdbedit` is a utility to modify scope hierarchies in the FSDB file.

Usage:

```
fsdbedit fsdb_file_name [Options] -o filename
```

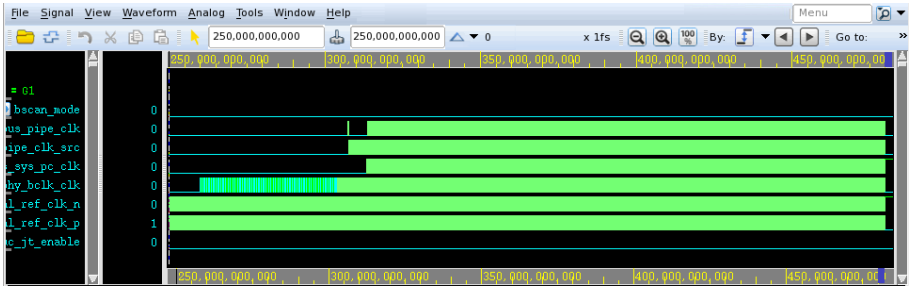
Options

```
[-insert configFile] [-insert_scope scope_string [-attribute
attr_string]]
[-delete configFile] [-delete_scope scope_string]
[-add_top_analog_scope scope_name]
[-rename configFile]
[-rename_scope target_scope rename_scope_name]
[-time_shift time_with_unit]
[-o fsdb_file_name][-ignore_2G]
```

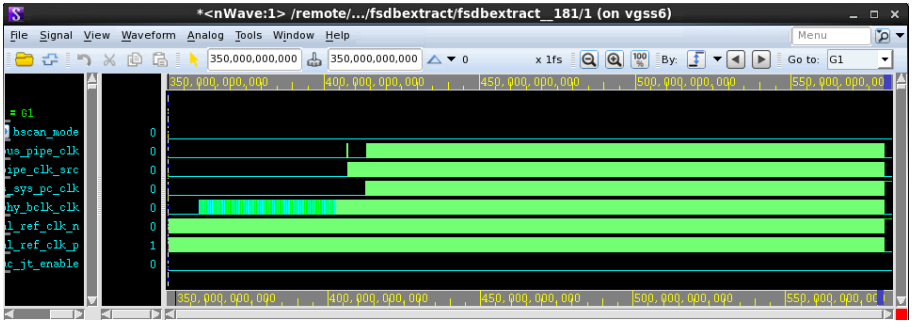
Options	Explanation
-add_top_analog_ scope *scope_name*	Specifies the scopes to be added as top scopes. This option supports Spice FSDB files only. **NOTE:** This option only applies to analog signals, including all Spice FSDB signals. For EPIC FSDB files, all digital signals remain unchanged.
-attribute *string*	Specifies the attributes of the scopes to be inserted. The syntax of *attr_string* is: *module_name=string;is_case_insensitive=1* or *0* **Example:** *-attribute module_name=aaa;is_case_insensitive=0* **NOTE:** The attribute string value should not contain any blank spaces.
-delete *configFile*	Specifies the configuration file name containing one or more scopes to be deleted.
-delete_scope *scope_string*	Specifies the scopes to be deleted.
-insert *configFile*	Specifies the configuration file name containing one or more scopes to be inserted. **Example:** *> fsdbedit gate.fsdb -insert -insert.cfg -o modified.fsdb*
-insert_scope *scope_string*	Specifies the scopes to be inserted.
-o *fsdb_file_name*	Specifies the output file name. Its default value is *default.fsdb*.
-rename *configFile*	Specifies the configuration file name containing the scopes to be renamed.

-rename_scope *target_scope rename_scope_name*	Specifies the scopes to be renamed with a new name. **Example**: *fsdbedit novas.fsdb -rename_scope "TOP/A" "B" -o rename.fsdb* **NOTE**: This option does not work with other options in *fsdbedit*, such as, *-insert_scope*, *-insert*, *-delete_scope*. If the *target_scope* is not found in the source FSDB, a warning message is reported as follows: **WARN* The input scope string <target_scope> can not be found.*
-time_shift *time_with_unit*	Specifies the shifted time (with respect to the original time range) for the output FSDB file. **Example**: *fsdbedit novas.fsdb -time_shift -1ns -o shift.fsdb* *fsdbedit novas.fsdb -time_shift 10ps -o shift.fsdb* **NOTE**: This option supports negative values. If the final time range exceeds the valid time window, it issues the following warning message: **WARN* Time shift out of time range.*
-ignore_2G	Transforms FSDB if the signal count of input FSDB is over 2G. Transformation fails if the count of user interested signal is over 2G.

The following figure illustrates the waveform before time shift:



On entering the `fsdbedit out2.fsdb -time_shift 100000000000fs -o 1` command, the waveform is time shifted, as illustrated in the following figure:



Examples:

Suppose that the original hierarchy is `/system/i_cpu/i_PCU`:

1. Insert before the list.

   ```
   %fsdbedit -insert_scope '/$scope(AA)/system/i_cpu/i_PCU'
   ```

   Final path: `/AA/system/i_cpu/i_PCU`

2. Insert a leaf scope.

   ```
   % fsdbedit verilog.fsdb \
   -insert_scope '/system/i_cpu/i_PCU/$scope(BB)' -o BB.fsdb
   ```

   Final path: `/system/i_cpu/i_PCU/BB`

3. Insert a scope before another scope.

   ```
   % fsdbedit verilog.fsdb -insert_scope '/system/i_cpu/i_PCU' \
   -o sig.fsdb
   ```

4. Delete the first scope.

   ```
   % fsdbedit verilog.fsdb \
   -delete_scope '/$scope(system)/i_cpu/i_PCU'
   ```

   Final path: `/i_cpu/i_PCU`

5. Delete the last scope.

   ```
   % fsdbedit verilog.fsdb \
   -delete_scope '/system/i_cpu/$scope(i_PCU)'
   ```

   Final path: `/system/i_cpu`

6. Delete one scope.

   ```
   % fsdbedit verilog.fsdb -delete_scope '/system/$scope(i_cpu)'
   \-o sig.fsdb
   ```

7. Delete one scope and all the subscopes and signals.

   ```
   % fsdbedit verilog.fsdb \
   -delete_scope '/system/$scope(i_cpu)/*' -o sig.fsdb
   ```

8. Insert the scopes in the configuration file.

   ```
   %fsdbedit verilog.fsdb -insert insert.cfg -o sig.fsdb
   ```

   where, the `insert.cfg` *file* contains the following:

   ```
   -insert_scope /$scope(Top)/system
   ```

```
-insert_scope /system/i_cpu/$scope(Middle)/i_ALUB
```

9. Delete the scope in the configuration file

   ```
   %fsdbedit verilog.fsdb -delete delete.cfg -o sig.fsdb
   ```

   where, the `delete.cfg` file contains the following:

   ```
   -delete_scope /system/$scope(i_cpu)
   ```

To specify the top scopes for Spice created analog FSDB files:

1. If the original FSDB file has scopes, it creates scopes specified by `scope_name` above the original scopes.

   Consider the original hierarchy as follows and `scope_name` as `s1.s2`:

   ```
   begin tree
     scope A;
        var va;
     upscope;
   end tree
   ```

   It creates two scopes before `scope A` and the generated FSDB hierarchy is as follows:

   ```
   begin tree
     Scope s1
       Scope s2
         Scope A
           var va;
         upscope
       upscope
     upscope
   end tree
   ```

2. If the signal name is in format `I(xxx)`, or `V(xxx)`, or other start code followed with the opening parenthesis '(' character and the closing parenthesis ')' character, `scope_name` is put in after the closing parenthesis '(' character. For example, if the original signal name is `I(a.b)`, `scope_name` is `x.y`, the generated FSDB has a signal named as `I(x.y.a.b)`.
3. If the original scope does not exist, it adds `scope_name` before the original signal name. For example, if the original signal name is `a.b`, `scope_name` is `x.y`, the generated FSDB has a signal named as `x.y.a.b`.
4. If the original signal name starts with the backslash (\) character, it adds `scope_name` before the original signal name though the addition is after the first backslash (\) character. For example, if the original signal name is `\\a.b`, the generated FSDB file has a signal named as `\\x.y.a.b`.

5. For mixed EPIC FSDB files, the `-add_top_analog_scope` option only applies to analog signals. Digital signals remain unchanged.

   The original hierarchy is as follows:

   ```
   begin tree
     Scope A
       var: digital_var
     upscope
     Scope B
       var: I(analog_var)
     upscope
   end tree
   ```

   After executing this following command:

   ```
   fsdbedit source.fsdb -add_top_analog_scope "aa" -o dest.fsdb
   ```

   The `dest.fsdb` hierarchy becomes as follows:

   ```
   begin tree
     Scope A
       var: digital_var
     upscope
     Scope B
       var: I(aa.analog_var)
     upscope
   end tree
   ```

fsdbexpand

`fsdbexpand` is a utility to process an existing Essential Signal FSDB file. With this utility, you can expand signals, scopes, and time periods of interest to a new FSDB file without rerunning time-consuming simulations. The `fsdbexpand` utility requires the design (using `-f` or `-lib`) and the Essential Signal FSDB file to be specified; the signal/scope list and time range options are optional.

The options are passed to the Siloti system in a non-GUI mode where Behavior Analysis and Data Expansion are performed and the new FSDB file is saved. The new FSDB file is used with other utilities (for example, `fsdbreport`, `fsdb2vcd`) or sent to another engineer for further analysis.

NOTE:

1) The `fsdbexpand` utility requires a Siloti-SimVE license.
2) For reasonable performance and memory consumption, the recommendation is to use `fsdbexpand` for a limited number of (`<100k`) signals and (`<100k`) cycles at a time.
3) If the `SKIP_CELL_INSTANCE` environment variable is set to *1*, library cells are skipped (not expanded) by the `fsdbexpand` utility.
4) Set the `NOVAS_BATCH_KILL` environment variable to *1* when using the `fsdbexpand` utility in LSF environments.

Usage:

```
fsdbexpand [FSDB Expansion Options] [General Options] [nTrace
Options] [Simulator Options] [Environment Options] [Behavior
Analysis Options] [Power Manager Options]
```

For details, see the *General/nTrace Options*, *Simulator Options*, *Environment Options*, *Behavior Analysis Options*, and *Power Manager Options* sections in the *verdi* utility.

FSDB Expansion Options

```
es_filename [-s {signal_name [-level n]}] [-sfile filename]
[-bt begin_time[unit]] [-et end_time[time_unit]]
[-ignore_errors_ba]
[-ignore_errors_vcom] [-crdbfile filename] [-extForceFSDB
filename] [-force filename] [-g2r_mapping filename] [-o
expanded_filename]
```

Option	Explanation
-bt *time [unit]*	Specifies the start time of the expansion. If omitted, the start time of the FSDB is used. The time unit is *Ms*|*Ks*|*s*|*ms*|*us*|*ps*|*fs*. The default time unit is the unit of the input FSDB.
-crdbfile *filename*	When specified, use the specified correlation database for gate/RTL mapping.
-et *time [unit]*	Specifies the end time of the expansion. If omitted, the end time of the FSDB is used. The time unit is *Ms*|*Ks*|*s*|*ms*|*us*|*ps*|*fs*. The default time unit is the unit of the input FSDB.
-extForceFSDB *filename*	Specifies the external force FSDB file.
-force *filename*	Specifies the force file name.
-g2r_mapping *filename*	When specified, use the specified mapping files for gate/RTL mapping.
-h | -help	Prints the help message.
-ignore_errors_ba 0|1	When specified, continue running *fsdbexpand* even when Behavior Analysis has errors; otherwise, stop executing *fsdbexpand*. Its default value is *0* (optional).
-ignore_errors_vcom 0|1	When specified, continue executing Behavior Analysis and *fsdbexpand* even when there are errors loading the design; otherwise stop executing *fsdbexpand*. Its default value is *0* (optional).
-level *n*	Expands signals from the specified scope and its *n* descendants. Use this option in conjunction with each scope in the -*s* option. The default level number is *1*. -*level 0*: Expands signals from the specified scopes and all hierarchies below. -*level 1*: Expands signals of the specified scopes only. -*level 2*: Expands signals of the specified scopes and the first generation scopes.
-nc_sig_file *filename*	Specifies the file containing the list of Not Computed (NC) signals (which can happen when the Essential Signal data is incomplete, resulting in less than 100% coverage). The default file name is *nc_sig.txt*.
-noNC	When this option is specified, replace any 'NC' values in the *fsdbexpand* output FSDB with '*x*'.
-o *filename*	Specifies the expanded output file name. If not specified, the results are saved to *out.fsdb* by default. *fsdbexpand* automatically creates a virtual file if multiple FSDB files are created.

Option	Explanation
-s *name1 ... namen*	Specifies the signals or scopes whose value changes are expanded. Multiple signals, scopes, or a combination are supported and are separated by spaces. The design hierarchical delimiter is the forward slash (/) character. If the *-s* and *-sfile* options are omitted, all signals are expanded. **NOTE:** As the exact signal name should be given, a back slash character (\) needs to be added to a signal name with any of the escaped characters (that is, *"\", "[", "]"*). **Examples**: *fsdbexpand ess.fsdb -f run.f -s 'top.u1.\carry[1] '* *fsdbexpand ess.fsdb -f run.f -s "top.u1.\carry[1] "* *fsdbexpand ess.fsdb -f run.f -s top.u1.\\carry\[1\]*
-sfile *filename*	Specifies the file containing the list of signals or scopes whose value changes are expanded. One signal or scope may be listed per line in the file. For example: */system/clock* */system/i_cpu/i_ALUB -level 1* If the *–s* and *–sfile* options are omitted, all signals are expanded.
es_filename	Specifies the Essential Signal FSDB file (*.*fsdb*) to be expanded. If multiple FSDB file names are specified, only the first file is expanded.

Examples:

1. Expand signals.

```
> fsdbexpand -ssf ess.fsdb -f run.f -ba –o sig.fsdb \
–s /system/clock
> fsdbexpand -ssf ess.fsdb -f run.f -ba –o sig.fsdb \
–s “system/clock”
> fsdbexpand -ssf ess.fsdb -f run.f -ba –o sig.fsdb \
–s /system/clock /system/i_cpu/C0
```

NOTE: When `-ba` is specified, the Behavior Analysis settings from the *novas.rc* resource file are used. This option is optional.

2. Expand all signals in the specified scopes and their subscopes.

```
> fsdbexpand -ssf ess.fsdb -lib work -ba -o scope.fsdb \
-s /system/i_cpu
> fsdbexpand -ssf ess.fsdb -lib work -ba –o scope.fsdb \
–s “/system/i_cpu” –level 0
> fsdbexpand -ssf ess.fsdb -lib work -ba –o scope.fsdb \
–s /system/i_pram -level 0 “/system/i_cpu/i_ALUB” –level 0
> fsdbexpand -ssf ess.fsdb -f run.f -ba –o sig.fsdb \
–s “/SYSTEM/CLOCK”
```

```
> fsdbexpand -ssf ess.fsdb -lib work -ba –o scope.fsdb \
–s /SYSTEM/i_cpu/ –level 0
```

NOTE: If double quotes are used with the `-s` option, the quotes can only surround one signal or scope name (for example, `/system/clock`). If the quotes surround multiple `-s` arguments (for example, `/system/clock /system/i_cpu -level 0`), `fsdbexpand` does not recognize it as either a scope or signal.

3. Expand signals from the specified scopes and multiple levels.

```
> fsdbexpand -ssf esd.fsdb -s top.scope1 -level 2 top.sig1 \
top.scope2 top.scope3 -level 0 -f run.f -o out.fsdb
```

NOTE: In the example above, `scope1` and 2 levels of subscopes are expanded. Only `scope2` is expanded and all levels below `scope3` are expanded.

4. Expand multiple signals and scopes.

```
> fsdbexpand -ssf ess.fsdb -lib work -bas {system} \
-loopUnroll 0 -clockSkew 0 -bboxEmptyModule 1 -cellModel 2 \
-simMode 0 -s /system/VMA /system/i_cpu/i_ALUB -level 0
> fsdbexpand -ssf ess.fsdb -f run.f -bas {system} \
-loopUnroll 0 -clockSkew 0 -bboxEmptyModule 1 -cellModel 2 \
-simMode 0 –s /system/i_cpu/i_ALUB –level 0 /system/VMA
```

5. Expand signals and scopes within a specified time range.

```
> fsdbexpand -ssf ess.fsdb -f run.f -o time.fsdb \
-s /system/VMA /system/i_cpu -bt 100 -et 1000
> fsdbexpand -ssf ess.fsdb -f run.f –o time.fsdb \
–s /system/VMA /system/i_cpu –bt 100ps –et 1000ps
```

NOTE: If `-ba` is not specified, the settings from the *novas.rc* resource file automatically perform Behavior Analysis.

6. Expand signals and scopes from the specified file.

```
> fsdbexpand -ssf ess.fsdb -f run.f -sfile sigfile.txt
```

The `sigfile.txt` file contains the following:

```
/system/i_cpu/C0
“system/clock”
“/system/i_cpu” –level 1
/system/i_cpu/i_ALUB -level 0
```

7. Expand all signals in the Essential Signal FSDB file.

```
> fsdbexpand -ssf ess.fsdb -f run.f -o all.fsdb
```

fsdbextract

`fsdbextract` is a utility to post-process an existing FSDB file. This utility extracts signals, scopes, and time periods of interest to a new FSDB file without rerunning time-consuming simulations. Depending on the dumped signal, the full or partial range of a bus signal are extracted. Also, the FSDB file can be split into multiple files by size constraint.

NOTE:

1) `fsdbextract` supports extracting FSDB files with transaction-type signals and local variables.

If the FSDB file contains glitches and the `FSDB_ENV_MAX_GLITCH_NUM` environment variable is:
(a) Set to 0, all glitches are extracted.
(b) Set to *1* or not set (the default value), no glitches are extracted to the new FSDB files.
(c) Set to *2*, the first and the last values of glitches are extracted. For example:
At time 100ns, if the values are 0, 1, x, z, then 0 and z are recorded.
At time 100ns, if the values are 0, 1, x, 0, then only 0 is recorded.
(d) Set to 3, the first two and the last values are extracted.
If the value of the `FSDB_ENV_MAX_GLITCH_NUM` environment variable is changed, `fsdbextract` needs to be rerun for the new value to take effect.

2) For the `-s` or `-f` options, if the specified signals are not found in the FSDB file, the setting of this environment variable impacts the results as follows:
(a) If the `NOVAS_FSDBEXTRACT_DISABLE_EXIT_ON_ERROR` environment variable is set (null, 0, 1, included), warning messages stating that the specified signals do not exist in the `<rtl.fsdb>` FSDB file are displayed (for example, `WARN: Command line: signal </a/b> does not exist in the fsdb <rtl.fsdb>`.) and an FSDB file containing only the found signals are generated.
(b) If the `NOVAS_FSDBEXTRACT_DISABLE_EXIT_ON_ERROR` environment variable is not set, error messages stating that the specified signals do not exist in the `<rtl.fsdb>` FSDB file are displayed (for example, `*ERROR* Command line: signal </a/b> does not exist in the fsdb <rtl.fsdb>`.) and the program exits.

3) If the FSDB file includes transaction data, the transaction data is ignored and only RTL signals in the FSDB file are extracted.

Usage:

```
fsdbextract fsdb_file_name [Options]
```

Options

Options	Explanation
-bt time[unit]	Specifies the start time of the extraction. If omitted, the start time of the FSDB is used. The time unit is *Ms*|*Ks*|*s*|*ms*|*us*|*ns*|*ps*|*fs*. The default time unit is *ns*.
-compact	Creates value changes in a compact format. This option may reduce file size but takes longer to create the FSDB file.
-et time[unit]	Specifies the end time of the extraction. If omitted, the end time of the FSDB is used. The time unit is *Ms*|*Ks*|*s*|*ms*|*us*|*ns*|*ps*|*fs*. The default time unit is *ns*.
-exclude_scope | *"scope1 scope2 ... scopeN"*	Signals inside the specified scope and its subscopes are not extracted. You may specify multiple scopes.
-f *config_file*	The *config_file* file is a Tcl file, which is used to specify the *-s*, *-bt*, *-et*, and *-fs* options. If the options are used both in the command line and *config_file*, the command-line options override those in *config_file*. The Tcl commands are as follows: *fuSetSignal:* For specifying the signal list (*-s*) *fuSetTimeRange:* For specifying the time range (*-bt, -et*) *fuSetFileSize:* For specifying the file size (*-fs*)
-fs *file_size*	Splits the extracted FSDB file into multiple files with the size of *file_size*1048576 bytes*. The file size must be >= *2*. If the original FSDB file size is less than 2, no split is performed. The split files are named with a 5-bit serial number before *.fsdb* (for example, *XXX00001.fsdb, XXX00002.fsdb, ... XXX0000n.fsdb*).
-h | -help	Prints the help message.
-level *n*	Extracts signals from the specified scope and its *n* descendants. Use this option in conjunction with *-s*. The default level number is *1*. *-level 0*: Extracts signals from the specified scopes and all hierarchies below. *-level 1*: Extracts signals of the specified scopes only. *-level 2*: Extracts signals of the specified scopes and the first generation scopes.
-log *filename*	Specifies the output log file name. The file name default is *err.log*.
-memory_will_alloc_in_mega *number*	Specifies the memory size, in Megabytes, to use to execute the *fsdbextract* utility. **NOTE:** When the memory size specified is not sufficient, the *fsdbextract* utility calculates the minimum size required to execute and then uses this value.

Options	Explanation
-nocase	When included, the mapping of signal names would be case insensitive. This option must be used in conjunction with *-s, -sigOnly or fuSetSignal*. The option affects all the specified scopes and signals in command line or configuration file.
-nolog	Disables generation of the *zwdutilsLog* log directory.
-o *extracted_fsdb_filename*	Specifies the extracted output file name.
-relation	This option extracts the relationships between transactions to the output FSDB file.
-s *name1...namen*	Specifies the signals or scopes whose value changes are extracted. Multiple signals, scopes, or a combination are supported and separated by spaces. If the *-s* option is omitted, all signals are extracted. The design hierarchical delimiter is '/'. The '*', '?', '\' and '%' characters are supported and signals/scopes containing them are double quoted on UNIX platforms. '*' cannot be used across scopes. For example, “*/data” matches */top_module/data* but cannot match */top_module/sub_scope/data*. The first '%'s is recognized as the escape character. For example, “%%”, “%*”, “%?” and “\a%?b%%%*” represent “%”, “*”, “?” and “\a?b%*” respectively. **NOTE:** 1) For details about the usage of backslash "\" escaped characters, see the Examples section below. 2) For system tasks or system functions, when a scope or signal name begins with '$' (for example, *$root*) it denotes a variable in the Unix shell, and gives a warning that the variable root value cannot be obtained. To avoid this error usage, use a single quote (') to replace double quotes ("). For example: *fsdbextract rtl.fsdb -s '/$root/En_a' -o test.fsdb*
-seq	Turns on sequence number dumping for the extracted FSDB file when the source file contains sequence number information. Sequence number dumping records the order of value changes that occur on different signals at the same time.
-time_shift time[unit]	Specifies the time offset for the extraction. If omitted, no time shift is applied. The time unit is optional and is *Ms*|*Ks*|*s*|*ms*|*us*|*ns*|*ps*|*fs*. The default time unit is *ns*. The time is positive or negative. For example, assuming the original FSDB has a time range of 10-100ns, when time shift is 20ns, the extracted FSDB file has a time range from 30ns to 120ns. When time shift is -5ns, the extracted FSDB file has a time range from 5ns to 95ns.

Options	Explanation
-tscale time[unit]	Specifies the time scale unit for the extracted FSDB file. The specified time must be positive. The time unit is *Ms*|*Ks*|*s*|*ms*|*us*|*ns*|*ps*|*fs*. The default time unit is *ns*.
-top_signals	Extracts signals that do not belong to any scope.
-verilog | -vhdl	The specified scope is in Verilog or VHDL format. Use it in conjunction with *-s*. The default format is *Verilog*. If multiple *-s* options are specified, only the last one is used.
-ignore_2G	Extracts FSDB if the signal count of input FSDB is over 2G. Extraction fails if the count of user interested signal is over 2G.

Examples:

1. Extract signals.

```
> fsdbextract rtl.fsdb –s /system/clock –o sig.fsdb
> fsdbextract rtl.fsdb –s “system/clock” –o sig.fsdb
> fsdbextract rtl.fsdb –s /system/clock /system/i_cpu/C0 \
–o sig.fsdb
> fsdbextract rtl.fsdb –s –nocase /system/clock \
/system/i_cpu/c0 –o sig.fsdb
> fsdbextract rtl.fsdb –s –nocase /SYSTEM/CLOCK –o sig.fsdb
> fsdbextract rtl.fsdb –s “/system/i_cpu/C*” –o sig.fsdb
> fsdbextract rtl.fsdb –s “/system/i_cpu/i_CCU/C?” –o sig.fsdb
> fsdbextract rtl.fsdb –s “/system/I_cpu/i_CCU/C??” –o sig.fsdb
```

NOTE: Surrounding double quotes (") are necessary if any signal contains one of the following characters: '*', '%' or ‘?".

2. Extract all signals in the specified scopes and their subscopes.

```
> fsdbextract rtl.fsdb -s /system/i_cpu -level 0 -o scope.fsdb
> fsdbextract rtl.fsdb –s “/system/i_cpu” –level 0 \
–o scope.fsdb
> fsdbextract rtl.fsdb –s “/system/i*” –level 0 –o scope.fsdb
> fsdbextract rtl.fsdb –s “/system/i_cpu/i_?C?” -level 0 \
–o scope.fsdb
> fsdbextract rtl.fsdb –s /system/i_pram “/system/i_cpu/i_ALUB”
–level 0 –o scope.fsdb
> fsdbextract rtl.fsdb –s –nocase /SYSTEM/I_cpu –level 0 \
–o scope.fsdb
```

3. Extract multiple signals and scopes.

```
> fsdbextract verilog.fsdb -s /system/VMA /system/i_cpu/i_ALUB
-level 0 -o ss.fsdb
> fsdbextract rtl.fsdb –s /system/i_cpu/i_ALUB –level 0 \
/system/VMA –o ss.fsdb
```

4. Extract signals and scopes within a specified time range.

```
> fsdbextract verilog.fsdb -s /system/VMA /system/i_cpu \
-bt 100 -et 1000 -o ss.fsdb
> fsdbextract rtl.fsdb –s /system/VMA /system/i_cpu \
–bt 100ps –et 1000ns –o ss.fsdb
```

5. Use a configuration file.

   Assume a configuration file, `ex.cfg`, with the following commands:

```
fuSetSignal {/system/I_???/c*} –nocase
fuSetSignal {/system/I_cpu/I_*} –level 0 –nocase
fuSetTimeRange –bt 100 –et 12345
fuSetFileSize 2
```

   The following command is executed:

```
> fsdbextract rtl.fsdb –f ex.cfg –o cfg.fsdb
```

   The result is the same as executing:

```
> fsdbextract –s –nocase “/system/I_???/c*” “/system/I_cpu/I_*”
–level 0 -bt 100 –et 12345 –fs 2 –o cfg.fsdb
```

6. Extract signals and scopes when backslash "\" escaped characters are present in the signal or scope name:

```
> fsdbextract verilog.fsdb -o ss.fsdb \
-s "/system/i_cpu/i_ALUB/\ACC_reg[0] "
```

NOTE: Signals/scopes containing backslashes must be double quoted. Also, an extra space must be added immediately after the part of the signal/scope with the backslash (in this case, `\ACC_reg[0]`).

fsdbjoin

`fsdbjoin` is a utility to concatenate the virtual file and real file into the joined file.

Usage:

```
fsdbjoin [General Option] joined_fsdbFile [Function Options]
```

General Options:

Options	Explanation
-f *joined_fsdbFile*	Displays the information of the target joined FSDB file.
-c *joined_fsdbFile*	Converts the virtual FSDB file and real FSDB file to a joined file.
-a *joined_fsdbFile*	Appends all input real FSDB files to the joined FSDB file and updates the contained virtual FSDB file.
-e *joined_fsdbFile*	Extracts the target file from the joined FSDB file.
-r *joined_fsdbFile*	Replaces the contained virtual FSDB file.
-h	Displays the usage of the *fsdbjoin* utility.

Function Options:

Options	Explanation
-i *input_fsdbFile*	Inputs the virtual or real FSDB file.
-o *input_fsdbFile*	Outputs the virtual or real FSDB file.
-s *scope*	Specifies the scope to add to the joined FSDB file.
-n *fileIndex*	Specifies the index of the joined FSDB file.

Example 1:

```
fsdbjoin -f 001.jf
```

Displays the information of a joined FSDB file.

Example 2:

```
fsdbjoin -c 001.jf -i 001.vf
```

Creates a joined FSDB file from a virtual FSDB file.

Example 3:

```
fsdbjoin -a test.jf -i 001.fsbb 002.FSDB
```

Appends FSDB file(s) to a joined FSDB file.

Example 4:

```
fsdbjoin -a test.jf -i 001.fsbb 002.FSDB -s "/top" "/top"s
```

Appends FSDB file(s) to a joined FSDB file with pre-scope settings.

Example 5:

```
fsdbjoin -e 001.jf -n 1 -o 001.fsdb
fsdbjoin -e 001.jf -n 0 -o 000.vf
```

Extracts the specified file from a joined FSDB file.

Example 6:

```
fsdbjoin -r 001.jf -i 001.vf
```

Replaces the virtual FSDB in a joined FSDB file.

Example 7:

```
fsdbjoin -h
```

Displays the usage of the `fsdbjoin` utility.

fsdbmangle

`fsdbmangle` is a utility to mangle signal names in the FSDB file. It is useful when an FSDB file is being sent to a third party without revealing the original signal names to the third party.

Usage:

```
fsdbmangle source_fsdb_fname [-o output_file_name] [Options]
```

Options	Explanation
-o	Specifies the output file name. Its default value is `source_fsdb_fname` with the postfix `".mangle.fsdb"`.
-seed	Specifies the random seed for generating random names.

Example:

```
fsdbmangle source.fsdb
fsdbmangle source.fsdb -seed 100
```

NOTE: Using a different seed number leads to different mangled names.

fsdbmerge

`fsdbmerge` is a utility to merge several FSDB files into a single file. It only works for FSDB files that can be converted into VCD files by the `fsdb2vcd` utility; it does not work for analog or rawfile dumping. `fsdbmerge` does not work for combining different types (Verilog, VHDL, or mixed) of FSDB files.

NOTE:

1) The `fsdbmerge` utility supports merging FSDB files with transaction-type signals, coverage-type signals, and local variables.
2) The `fsdbmerge` utility is not able to merge a packed FSDB with an un-packed FSDB.
3)The `fsdbmerge` utility is designed to merge small FSDB files. For large FSDB files, a virtual file is recommended.
4) The `fsdbmerge` utility automatically compares signals and hierarchies when the `-sw` or `-split` options are not specified, although the comparison consumes time and memory. When you know the FSDB files have either the same or different signals, these two options are recommended, so that comparison time and memory are reduced.
5) If two or more FSDB files have trigger points, all trigger points are merged into a single one.

The maximum (minimum) time point of the resulting FSDB file is the maximum (minimum) time of the merged FSDB files. Hierarchy trees of the merged FSDB files are merged into one tree based on hierarchical scope names and variable names. Therefore, the scopes and variables names are merged as long as the full path name is the same.

Merging files with different time scales is supported. The smallest time scale is selected for the output FSDB file.

Usage:

```
fsdbmerge fsdbFileName [Options] -o mergedFileName
```

Options	Explanation
-compact	Creates value changes in a compact format. This option reduces file size but takes longer to create the FSDB file.
-h | -help	Prints the help message.
-keep_clashed	Do not discard value changes of clashed signals (those with the same hierarchical name, but different var types). The original signal names are renamed.
-keep_clashed_toscope	Do not discard value changes of clashed signals (those with the same hierarchical name, but different var types). Put them into different subscopes.

Options	Explanation
-keep_overlapped	Do not discard value changes from overlapped signals. Overlapped signals are those from different source files that have both the same hierarchical name and overlapped time ranges.
-no_x	Skips adding value X to the end of the signals from FSDB whose end time is smaller than that of the merged FSDB.
-o *mergedFileName*	The file name of the final merged FSDB file.
-overwrite	For clashed signals (those with the same hierarchical name but different var types), write the signal in the last FSDB file.
-seq	Turns on the sequence number.
-show_disjoint_warning	Displays a warning message on the screen if the FSDB files have disconnected time ranges. For example, if a *bus1* signal exists in two FSDB files with different time ranges (for example, *bus1* in *1.fsdb* has a time range of 0~100 ns and *bus1* in *2.fsdb* has a time range of 110~200 ns), then *bus1* has a disconnected time range of 10 ns.
-smallrange	Updates begin time and end time of the merged FSDB to that of the FSDB with the smallest time range.
-split	Merges FSDB files with completely different signals and hierarchy to a single FSDB file. Signals in different FSDB files cannot be overlapped (that is, the same signal cannot exist in both files). FSDB files with user-defined data types and transactions are supported.
-stop_if_disjoint	Stops *fsdbmerge* if the FSDB files have disconnected time ranges.
-sw	Merges FSDB files with the same signals and hierarchy to a single FSDB file. This option is recommended when the FSDB files are dumped using the *$fsdbAutoSwitchDumpfile* dumping task.

Example 1:

```
fsdbmerge test.01.fsdb test.02.fsdb test.03.fsdb -o test.fsdb
```

The command merges `test.01.fsdb`, `test.02.fsdb`, and `test.03.fsdb` to `test.fsdb`.

NOTE: If time ranges are overlapped, the signal's value change of the latter FSDB file in the command line is used. See the example illustrated below.

Suppose `file_1.fsdb` has signals `top.a`, `top.system.b`, and `top.c` with simulation time from 0 to 1200ns. `file_2.fsdb` has signals `top.c`, `top.system.d`, and `top.e` with simulation time from 1000ns to 1600ns. During 1000ns to 1200ns, *top.c* in `file_1.fsdb` undergoes the following value changes:

```
(1010,x) (1070,1) (1150,z)
```

In `file_2.fsdb`, it undergoes the following value changes:

```
(1010,0) (1050,z) (1120,0) (1170,1)
```

Then, the merged `top.c` has the following value changes:

```
(1010,0) (1050,z) (1120,0) (1170,1)
```

Example 2:

```
fsdbmerge addr4.fsdb verdi.fsdb -keep_clashed
```

After the `-keep_clashed` option is applied, `fsdbmerge` changes the names of the clashed signals by appending `//clashed@[var_type_string]`.

For example, both `addr4.fsdb` and `nonosim.fsdb` have a signal named `/top/dup/n1`. After merging with the `-keep_clashed` option, the names of the clashed signals are changed to `/top/dup/n1//clashed@verdi` and `/top/dup/n1//clashed@verilog` respectively.

Example 3:

```
fsdbmerge addr4.fsdb verdi.fsdb -keep_clashed_toscope
```

After the `-keep_clashed_toscope` option is applied, `fsdbmerge` creates subscopes named after clashed signals' variable types, and moves the clashed signal to the newly created subscope respectively.

For example, both `addr4.fsdb` and `nonosim.fsdb` have a signal named `/top/dup/n1`. After merging, the names of the clashed signals are changed to `/top/dup/_@verilog/n1` and `/top/dup/_@verdi/n1` respectively.

fsdbrecover

`fsdbrecover` is a utility to recover unfinished FSDB files. An FSDB file is only recoverable when all the necessary temporary files are available and are located in the same directory as the source FSDB file. The FSDB file is recoverable even when it is locked after being abnormally closed or when a different user account is used.

Usage:

```
fsdbrecover source_fsdb_fname [-o output_file_name] [Options]
```

Options	Explanation
-del_src	Deletes source FSDB file and its temporary files after recovery.
-o	Specifies the output file name. Its default value is *source_fsdb_fname* with postfix ".recover.fsdb".

NOTE: `fsdbrecover` does not fix any corruptions in the original FSDB file. `fsdbrecover` only merges the temporary files to the main FSDB file. First, you need to analyze the corrupted FSDB file and then, the FSDB file is dumped again. Dumping FSDB files under more stable NFS environments may also avoid crashes.

fsdbreport

`fsdbreport` is a utility to generate an ASCII text file of value change lists for specified signals from an FSDB dump file.

NOTE: `fsdbreport` supports FSDB files with transaction-type signals with the following exceptions:
1) Transaction type signals cannot be used with `-strobe` or `-levelstrobe` options. Otherwise, error messages similar to the following are printed to the console:

```
*WARN* Transaction type signal </BusTop/MyAHB_1/
AhbTransaction> cannot be used with -strobe!
*WARN* Report stop!
*WARN* Transaction type signal </BusTop/MyAHB_1/
AhbTransaction> cannot be used with -levelstrobe!
*WARN* Report stop!
```

2) To accommodate the long and variable length of transaction type value changes, `fsdbreport` automatically uses a non-table format report (that is, `-cn 0`) whenever a transaction type signal exists in the signal list. The output format looks like the following:

```
Time:0
SystemC/HRESP00
SystemC/HADDR00000000000000000000000000000000
SystemC/HWRITE0
Time:60000
/BusTop/MyAHB_1/AhbTransaction
    *** single read (0 60000)-(0 140000) ***
    Command=single read
    Master='d 2
    Slave='d 2
    StartAddress='d 14526
    Data="00000000000000000000000010101010"
    Response=split
    BurstType=single
    BeatCount='d 1
    SizePerBeat=1 byte
```

3) The `-of` and `-af` options do not change the output format of a transaction's attribute value.

Usage:

```
fsdbreport fsdb_filename [-f config_file] [-bt time[unit]]
[-et time[unit]]
[-nocase] [-w column_width] [-of output_format] [-verilog |
-vhdl] -s {signal [-level level_depth] [-a name] [-w
column_width] [-af alias_file]
```

```
[-of [b|o|d|u|h]] [-verilog|-vhdl] [-precision
precision_value]}[-strobe [signal=="value"] [-a name] [-w
column_width] [-verilog|-vhdl]]
[-levelstrobe "expression"] [-a name] [-w column_width]
[-verilog|-vhdl]]
[-shift shift_time | -shiftneg shiftneg_time] [-period
period_time]
[-cn column_number] [-log filename] [-o reported_filename]
[-pt time_precision] [-csv] [-find_forces  [-no_value]
[-no_fdr_glitch]] [-ignore_2G][-first_force]
```

Options	**Explanation**
-a *alias_name*	Defines the alias for the output signal.
-af *alias_file*	Specifies an *nWave* waveform alias file. **NOTE:** Specifies the *-af* option must be specified for each signal. For example: *-s* */tb/chip/lbu/core/dmac/haddr -a haddr -af novas.alias* */tb/chip/lbu/core/dmac/hwdata -a hwdata -af novas.alias*
-bt time[unit]	Specifies the begin time of the report. If omitted, the begin time of the FSDB file is used. The time unit is *Ms, Ks, s, ms, us, ns, ps*, or *fs*. The default time unit is *ns*.
-cn *column_number*	Defines the number of columns for the report, including the time column. Column number is set to be 0 or an integer larger than 1. When setting to *0*, the signal name and its value are not printed in the format of a table, but line by line. **NOTE:** This option is ignored if the *-csv* option is also specified.
-csv	Saves the output report file in the CSV format. If this option is specified with *-cn* and *-w*, these options are ignored. **NOTE:** If the selected signals contain stream, coverage, or SVA type signals, the *-csv* option is ignored and non-csv format is output.
-et time[unit]	Specifies the end time of the report. If omitted, the end time of the FSDB is used. The time unit is *Ms, Ks, s, ms, us, ns, ps,* or *fs*. The default time unit is *ns*.
-exclude_scope "*scope1*" ["*scope2*" … "*scopeN*"]	Excludes signals under the specified scopes. Each scope is enclosed with double quotes. To exclude subscopes of the specified scopes, the wildcard character "*" is appended at the end of the scopes. **Example:** *-find_forces -exclude_scope "/system/cpu/*" "/system/s1"* **NOTE:** 1) Use this option with the *-find_forces* option. 2) The *-exclude_scope* option has higher priority than the *-s* option.

Options	Explanation
-exp *expression*	Reports values when the expression changes to *true* (==1). The expression syntax is the same as the **Logical Operation** command under the **Signal** menu in *nWave*. **Example:** *fsdbreport verilog.fsdb -s system/data -exp "/system/addr=='h30 & /system/clock==1"*
-f *config_file*	Specifies a text file which defines all the options except *-h* and *-f*. The pound (#) sign isadded to the beginning of a line as a comment line. **Example:** *fsdbreport example.fsdb -f config_file -o output.txt*
-find_forces	Shows signals with force, release, or deposit events and the signal values.
-h | -help	Prints the help message.
-level *level_depth*	Specifies the number of levels to be dumped under the specified scope. Use this option with the *-s* option. When set to *0*, all signals below the specified scope is dumped.
-levelstrobe "expression"	Dumps values when the expression holds true (level sampling). If the expression is not true, the time point and value are not dumped. The expression can consist of one strobe signal (for example, "*a==1*") or multiple strobe signals (for example, "*a==1 && b==1*"). The *-strobe* and *-levelstrobe* options cannot be used together. **Example:** *fsdbreport verilog.fsdb -levelstrobe "/system/clock==1 && /system/sig==1" -s /system/data /system/addr*
-log *filename*	Specifies the output log file name. The default file name is *err.log*.
-nolog	Disables generation of the *fsdbreportLog* log directory.
-nocase	When included, the mapping of signal names is not case-sensitive. The default is case-sensitive.
-no_fdr_glitch	Shows the stable value for force, release, and deposit events. **NOTE:** This option is optional and is used with the *-find_forces* option; otherwise, it is ignored.
-no_value	Disables value display of force, release, and deposit events. **NOTE:** This option is optional and is used with the *-find_forces* option; otherwise, it is ignored.
-o *reported_file_name*	Specifies the output report file name. If the *-o* option is not specified, the default is *report.txt*.

<table>
<tr><th>Options</th><th>Explanation</th></tr>
<tr><td>-of [b|o|d|u|h]</td><td>Defines the output display format as binary, octal, decimal, unsigned decimal, or hexadecimal. Its default value is binary. When this option is specified before the -s option, it is applied globally. When this option is specified after a signal, it is applied locally.</td></tr>
<tr><td>-period period_time</td><td>Dumps values at each specified time.</td></tr>
<tr><td>-precision
precision_value</td><td>Defines the precision (the number of decimal places to include) of output values for analog signal types. This option is ignored for digital signal types.
Example:
- Original output without the -precision option:
> fsdbreport test.tr0.fsdb -s "I(vcc"

<pre>Time(1.0)   I(vcc
==========   ==========
0.000000e+00   -10.0pA
5.000000e-11   23.4nA
1.000000e-10   25.0nA
3.000000e-10   30.4nA
1.100000e-09   41.0nA
2.100000e-09   43.3nA
3.100000e-09   43.4nA</pre>
- Report with the -precision 7 option:
> fsdbreport test.tr0.fsdb -s "I(vcc" -precision 7 -w 20 -o report_with_precision_5.txt

<pre>Time(1.0)   I(vcc
==========   ====================
0.000000e+00   -1.0000000e-11A
5.000000e-11   2.3453000e-08A
1.000000e-10   2.5039000e-08A
3.000000e-10   3.0451002e-08A
1.100000e-09   4.1056001e-08A
2.100000e-09   4.3309001e-08A
3.100000e-09   4.3463999e-08A
4.100000e-09   4.3423999e-08A
5.100000e-09   4.3402000e-08A
6.100000e-09   4.3387999e-08A
7.100000e-09   4.3374001e-08A</pre></td></tr>
<tr><td>-pt time_precision</td><td>Defines the time precision (the number of decimal places to include) of the output value for analog signal types. This option is ignored for digital signal types.
Example:
fsdbreport hspice.fsdb -s "/v(_be0" -pt 5 -o 1.txt</td></tr>
</table>

Options	Explanation
-s {*signal_name* [options]}	Specifies the signals or scopes to be reported. When specifying a scope name, the wildcard character "*" is appended to the end with double quotes. At least one *-s* must be included. **NOTE:** For system tasks or system functions, when a scope or signal name begins with '$' (for example, *$root*), it denotes a variable in the Unix shell, and gives a warning that the variable root value is not obtained. To avoid this error usage, use single quotes (') to replace double quotes ("). For example: *fsdbreport test.fsdb -s '/$root/En_a' -o test_report.txt* **NOTE:** If the output format needs to be specified for multiple signals, the output specification must immediately follow each of the desired signals. For example: *fsdbreport test_fsdb.fsdb -o multi_scope.txt -s "/U_core_top*" -precision 5 -w 17 "/U_pad_ring*" -precision 6 -w 20 -bt 555 -et 555* *-precision 5 -w 17* belongs to the *"/U_core_top*" signal* *-precision 6 -w 20* belongs to the *"/U_pad_ring*"* signal If the following are specified, the *-precision 6 -w 20* only belongs to */U_pad_ring:* *fsdbreport test_fsdb.fsdb -o multi_scope.txt -s "/U_core_top*" "/U_pad_ring*" -precision 6 -w 20 -bt 555 -et 555 -cn 20*
-shift | -shiftneg	Specifies to *shift (plus)* or *shiftneg (minus)* the report time when the strobe signal matches the specified value. Use this option with the *-strobe* option. **Example:** *fsdbreport verilog.fsdb -shift 10 -s "system/VMA" -strobe "system/clock==1" -o 1* Samples the value for VMA 10 time units after the clock becomes 1. *fsdbreport verilog.fsdb -shiftneg 300ps -s "system/VMA" -strobe "system/clock==1" -o 1* Samples the value for VMA 300ps before clock becomes 1.
-strobe "expression"	Reports values when the value of the strobe signal changes to the specified value (edge sampling). The strobe is in the format of "*signal==value*". For multi-bit signals, the value is in binary without leading zeros.
-verilog | -vhdl	Specifies the output format as Verilog or VHDL format. When this option is specified before the *-s* option, it is applied globally. When this option is specified after a signal, it is applied locally.

Options	Explanation
-w column_width	Defines the width of the signal column. When this option is specified before the *-s* option, it is applied globally. When this option is specified after a signal, it is applied locally. The default column width is *10*. If the width of the signal or value is greater than the specified column, then '*' is used in the signal or value. For example, if */system/addr* is an 8-bit bus: > *fsdbreport verilog.fsdb -w 5 -s /system/addr* *addr is reported with values, such as, *0000 in *report.txt*. For the strobe, *level_strobe* or expression signal, if the width is less than the maximum time width, the width is automatically expanded to the maximum time width. **NOTE:** This option is ignored if the *-csv* option is also specified.
-ignore_2G	Reports FSDB if the signal count of input FSDB is over 2G. Reporting fails if the count of user interested signal is over 2G.
-first_force	Only shows the first force event for signal. This option works only with the `-find_forces` option.

Examples:

1. Assign the begin time (1000ps) and end time (2000ps) for the report. The value of the end time must be greater than the value of the begin time; otherwise, no output is generated.

   ```
   > fsdbreport verilog.fsdb -s /system/addr -bt 1000ps -et 2000ps
   ```

2. Report multiple signals. Only one `-s` option takes effect. If two or more are specified, only the last `-s` takes effect. For example, the following commands lead to the same result with `addr` being reported only:

   ```
   > fsdbreport verilog.fsdb -s /system/clock -s /system/addr
   > fsdbreport verilog.fsdb -s /system/addr
   ```

   However, both `clock` and `addr` are reported with the following command:

   ```
   > fsdbreport verilog.fsdb -s /system/clock /system/addr
   ```

3. Report a slice of a bus signal. The MSB and LSB of the partial bus must match the original bus. For example, if there is the `addr[7:0]` bus, bits 7, 6, 5, and 4 are extracted with the following command:

   ```
   > fsdbreport verilog.fsdb -s "/system/addr[7:4]"
   ```

   If the following command is executed, nothing is extracted:

```
> fsdbreport verilog.fsdb -s "/system/addr[4:7]"
```

4. Report signals in the signal list using different formats.

   ```
   > fsdbreport fsdb/vhdl_typecase.fsdb -nocase -s top/
   SIMPLE_REC.FIELD3
   -a simple.field3 -w 15 TOP/COMPLEX_REC.F1.FIELD3 -a
   complex.f1.field3
   -w 20 top/a_std_logic_vector -af sean2.alias -of a -o
   output.txt
   -bt 1000 -et 2000
   ```

   `simple.field3` is reported as the alias for the `top/SIMPLE_REC.FIELD3` signal, `complex.fl.field3` is reported as the alias for the `TOP/COMPLEX_REC.F1.FIELD3` signal, and the alias values in `sean2.alias` are reported for the `top/a_std_logic_vector` signal.

5. Report signals using different output formats. For example, the formats are applied globally with the following command:

   ```
   > fsdbreport verilog.fsdb -w 15 -of d \
   -s /system/addr /system/clock
   ```

   The column width is 15 and the radix is decimal for both `/system/addr` and `/system/clock`.

   The following command applies column width of 12 and radix of 16 (hexadecimal) locally to `/system/addr`.

   ```
   > fsdbreport verilog.fsdb -w 15 -of d -s /system/addr -w 12 \
   -of h /system/clock
   ```

   However, column width of 15 and radix of 10 (decimal) are applied globally to `/system/clock`.

6. Report a scope and its descendants. Multiple scopes may be specified.

   ```
   > fsdbreport rtl.dump.fsdb -bt 10 -et 100 \
   -s "/system/i_cpu/*" -level 3 /system/i_pram/clock -cn 0
   ```

   Up to 3 levels of scopes below */system/i_cpu* and the */system/i_pram/clock* signal between 10-100ns are reported. The results are printed line by line.

7. Report the results for the specified strobe point using `-strobe`.

   ```
   > fsdbreport verilog.fsdb -strobe "/system/clock==1" -s /
   system/data /system/addr
   ```

   Only when the value of the `/system/clock` signal is 1, the values of the `/system/data` and `/system/addr` signals are reported.

8. Specify a configuration file. In the configuration file, all the command options (except `-h` or `-f`) are specified in a single line or multiple lines. The `fsdbreport` utility reads them in order. For example, the following three files generate the same extracted results:

   File1:

   ```
   -w 12 -s /system/addr /system/clock
   -w 15 -et 1200
   ```

   File2:

   ```
   -w 12
   -s /system/addr /system/clock -w 15 -et 12000
   ```

   File3:

   ```
   -w 12
   -s /system/addr
   /system/clock
   -w 15 -et 12000
   ```

   ```
   > fsdbreport verilog.fsdb -f File1
   > fsdbreport verilog.fsdb -f File2
   > fsdbreport verilog.fsdb -f File3
   ```

9. Report the results when the expression value changes to `true`.

   ```
   > fsdbreport verilog.fsdb -exp "/system/addr=='h30 & /system/
   clock==1"
   -s /system/data
   ```

   When the value of the `/system/addr` signal is `'h30`, and the value of the `/system/clock` signal is 1, the expression is `TRUE`. The value of the `/system/data` signal is reported when the expression is `TRUE`.

10. Report the force, release or deposit information of the specified signals using `-find_forces`.

    ```
    > fsdbreport rtl.fsdb -find_forces -s "/system/i_cpu/*" -level
    2 -o report.txt
    ```

    When the specified signals meet the following conditions, the signal information is reported:

    - Signals with force, release, or deposit events
    - Signals under the `/system/i_cpu` scope
    - Signals located two levels under the specified scope. If the level is more than 2, the signals are ignored.

11. Report the force of the specified signals using `-find_forces` and `-exclude_scope`.

```
> fsdbreport rtl.fsdb -find_forces -s "/system/i_cpu/*"
-exclude_scope "/system/i_cpu/s1/*" "/system/i_cpu/s2" -o
report.txt
```

12. Exclude a specific scope without excluding its subscopes.

```
> fsdbreport test.fsdb –exclude_scope "/top/system" –o out.txt
```

13. Specify a scope and exclude the specified scope and its subscopes.

```
> fsdbreport test.fsdb –exclude_scope "/top/system/*" –o
out.txt
```

14. Exclude multiple scopes.

```
> fsdbreport test.fsdb –exclude_scope "/top/system/*" "/top/S1"
"/top/S2/" –o out.txt
```

merge_multi_swp_fsdb

`merge_multi_swp_fsdb` is a utility to merge multiple sweep analog FSDB files into a single FSDB file.

Usage:

```
merge_multi_swp_fsdb {-in file_name -key condition_string_key
-value condition_string_value} [-o output_fsdb_file]
```

Options	Explanation
-help | -h	Prints the help message.
-in	Specifies the input file name of a sweep. This option is used with each new FSDB file name.
-key	Specifies the simulation condition key string of a sweep. Skipping the key after the first input file's key condition is specified automatically follows the key sequence of the first input file.
-o	Specifies the output file name. The default output file name is *merged.fsdb*.
-value	Specifies the simulation condition value string of the current sweep.

Example:

```
merge_multi_swp_fsdb -in test.tr0.fsdb -key Temp -value 10 \
-key VOLT -value 20 -in test.tr0_2.fsdb -value 20 -value 30 \
-in test.tr0_3.fsdb -value 25.5 -value 20 -o my_merge.fsdb
```

nce2report

`nce2report` is a utility to generate a readable waveform comparison report file from the `.nce` file.

Usage:

```
nce2report [Options]
```

Options	Explanation
-column *value*	When *-format tabular* is specified, set the number of columns. Its default value is *8*.
-duration *value*	Specifies the duration column width in the report. Its default value is *7*.
-f *filename*	Specifies the command file name.
-format [*plain* | *tabular*]	Sets the display mode for the output format. Its default value is *plain*. When *tabular* is specified, the *-column* and *-scope* options are also specified.
-h | -help	Prints the help message.
-hidden	When specified, the "*error width*" in mismatch and unstable results table is hidden.
-html	When specified, an HTML document is generated.
-i *filename*	Specifies the input file (*.*nce*) name.
-name *value*	Specifies the signal name column width in the report. Its default value is *20*.
-o *filename*	Specifies the output file name.
-scope	When *-format tabular* is specified and this option is specified, signals are grouped under the same scopes. The default is to put all signals together.
-sort *time* | *hier* [+*cmp_option*]	Sort comparison errors by time or hierarchy. Its default value is *time*. When *+cmp_option* is specified, the errors with the same compare options are grouped together.
-time *value*	Specifies the time column width in the report. Its default value is *8*.
-total *value*	Specifies the total page width. Its default value is *80*.
-value *value*	Specifies the signal value column width in the report. Its default value is *10*.

Report File Format

The following paragraphs describe the file format of a comparison report generated by `nce2report`.

When the [-sort time] and [-format plain] Options are Applied

When the `-sort time` and `-format plain` options are applied, the error items are sorted by the ascending order of their starting times. If there are two signals in an item, it is the starting time of the secondary signal. The file is printed in plain format as follows.

```
*********************************************************************
*********************************************************************
**                                                                 **
**                  A Readable Report of nCompare                  **
**                                                                 **
*********************************************************************
*********************************************************************


^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
                            The 1st Comparison
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Golden fsdb :                                              Golden.fsdb
Secondary fsdb :                                         Secondary.fsdb
The number of compared signal pairs :                                 8
The number of signals pairs with errors :                             4
Total errors :                                                        6
Golden time unit :                                                  1ns
Secondary time unit :                                               1ns


=====================================================================
                            Mismatch results table
---------------------------------------------------------------------
TIMEWIDTHSECONDARY   GOLDEN      V_ACTUALV_EXPECTED
3505 /system/R_W      10
3505 /system/i_cpu/C5  /system/i_cpu/S5  0  1
40010/system/VMA      x 1
105011/system/i_cpu/i_ALUB/IDB[7:0]      0000101000001011

=====================================================================
                            Unstable results table
---------------------------------------------------------------------
TIMEWIDTHSIGNAL   G/S     REASON
50010/system/i_cpu/S5  G     setup
50010/system/i_cpu/C5  S     setup
```

The following conditions apply:

- Each `cmpOpenFsdb` command in the rules file represents a comparison.
- The meaning of each error is as follows:

  `[time of error] [error width] [S_signal] [G_signal] [actual value] [expected value] [time of error] [error width] [signal] [golden or secondary] [unstable]`

- The value of `unstable` is either `"setup","hold", "cycle", "expr"` or `"none"` (`"none"` indicates that the signal has fewer trigger points than other signals).
- The width of each column is specified (defaults are provided). If the width of a value exceeds the specified width, it is truncated. If the width of a column name exceeds the specified width, it is also truncated. If the name of the secondary signal is the same as that of the golden signal, the name of the golden signal is omitted and its space is occupied by the name of the secondary signal.
- The page width is specified.

  Using the `-total` option can set the width of the report page. If the specified page width is less than 80, the page width is automatically set to 80. If the sum of the widths of all columns is less than the page width, the remaining room is allocated as the following rules:

  a. If there are columns with maximum widths greater than the specified width, assign the remaining room to them.
  b. If there is still space left, assign the remaining space equally amongst the columns.

  If the sum of the widths of all columns is greater than the page width, the widths of the columns are adjusted according to the following rules:

  a. Decrease the widths of columns with widths greater than their actual widths.
  b. If the page width is still insufficient, decrease the widths of all columns equally.
- The error width is hidden by users.

In the previous example, the comparison options are ignored. If the options (invoked by specifying the `+cmp_option` option) are required, the report is grouped by each comparison options setting. The file is printed in plain format as follows.

```
****************************************************************
****************************************************************
**                                                            **
**                A Readable Report of nCompare               **
**                                                            **
****************************************************************
****************************************************************


^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
                          The 1st Comparison
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Golden fsdb :                                          Golden.fsdb
Secondary fsdb :                                     Secondary.fsdb
The number of compared signal pairs :                            8
The number of signals pairs with errors :                        4
Total errors :                                                   6
Golden time unit :                                             1ns
Secondary time unit :                                          1ns


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Time range :                          0ns 12472ns
X switch :                  on    Z switch :           on
Glitch switch :             off   Strength switch :    off
Maximum of errors :         1000  Shift :              0ns
Maximum of errors of each pair :100   Tolerance window :   pos=0ns
neg=0ns

Golden clock :                                                 disable
Secondary clock :                                              disable
Golden cycle :                                                 disable
Secondary cycle :                                              disable
Golden exp :                                                   disable
Secondary exp :                                                disable

================================================================================
                          Mismatch results table
--------------------------------------------------------------------------------
TIMEWIDTHSECONDARYGOLDEN V_ACTUALV_EXPECTED
3505 /system/R_W 10
40010/system/VMA x1
105011/system/i_cpu/i_ALUB/IDB[7:0]  0000101000001011
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Time range :                          0ns 12472ns
X switch :                  on    Z switch :           on
Glitch switch :             off   Strength switch :    off
Maximum of errors :         1000  Shift :              0ns
Maximum of errors of each pair :100   Tolerance window :   pos=0ns
neg=0ns

Golden clock :    /system/clock setup=10ns hold=10ns sample=0
trigger=BOTH
Secondary clock : /system/clock setup=10ns hold=10ns sample=0
trigger=BOTH
Golden cycle :    50ns delay=0 width=5ns
Secondary cycle : 50ns delay=0 width=5ns
Golden exp   :    /system/reset
Secondary exp   : /system/reset

================================================================================
                          Mismatch results table
--------------------------------------------------------------------------------
TIMEWIDTHSECONDARY GOLDEN    V_ACTUALV_EXPECTED
3505 /system/i_cpu/C5 /system/i_cpu/S5 01
```

```
================================================================================
                          Unstable results table
--------------------------------------------------------------------------------
TIMEWIDTHSIGNAL     G/SREASON
50010/system/i_cpu/S5    Gsetup
50010/system/i_cpu/C5    Ssetup
```

When the [-sort hier] and [-format plain] Options are Applied

When the `-sort hier` and `-format plain` options are applied, the error items are sorted by their signal names. If there are two signals in an item, it is the name of the secondary signal compared to the first. If equal, then compare the name of the golden signal. The file is printed in plain format as follows.

```
**************************************************************************
**************************************************************************
**                                                                      **
**                  A Readable Report of nCompare                       **
**                                                                      **
**************************************************************************
**************************************************************************

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
                          The 1st Comparison
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Golden fsdb :                                                        rtl.fsdb
Secondary fsdb :                                                    gate.fsdb
The number of compared signal pairs :                                     35
The number of signals pairs with errors :                                  6
Total errors :                                                            15
Golden time unit :                                                       1ns
Secondary time unit :                                                    1ns

================================================================================
                         Mismatch results table
--------------------------------------------------------------------------------
SECONDARY                     GOLDEN             TIME WI+ V_ACTUAL
V_EXPEC+
/system/addr[7:0]            450   15 00000000 00000001
/system/addr[7:0]            550   9  00000001 00000000
/system/i_cpu/\IXR[7:0]     /system/i_cpu/IXR[7+ 2300 6  01010101
00100000
/system/i_cpu/\IXR[7:0]     /system/i_cpu/IXR[7+ 3300 6  01010101
00100000
/system/i_cpu/\IXR[7:0]     /system/i_cpu/IXR[7+ 4500 6  01010101
00100000
/system/i_cpu/\IXR[7:0]     /system/i_cpu/IXR[7+ 7200 6  01010101
00100000
/system/i_cpu/i_ALUB/\X0[0] /system/i_cpu/i_ALU+ 450  15 0        1
/system/i_cpu/i_PCU/C5                           950  5  0        1
/system/i_cpu/i_PCU/C5                           1050 5  0        1
/system/i_cpu/i_PCU/C5                           1950 5  0        1
/system/i_cpu/i_PCU/IDB[7:0]                      450  15 00000000
00000001
/system/i_cpu/i_PCU/IDB[7:0]                      550  14 00000000
00000001
/system/i_cpu/i_PCU/IDB[7:0]                      650  7  00000000
00000001

================================================================================
                          Unstable results table
--------------------------------------------------------------------------------
SIGNAL                          G/S  TIME           WIDTH
REASON
```

```
/system/i_cpu/i_CCU/TDB[7:0]    S    400           10             hold
/system/i_cpu/i_CCU/TDB[7:0]    G    400           10             hold
```

The following conditions apply:

- The errors are sorted in ascending order by the number of hierarchies of the signal names first. Then, if the numbers of hierarchies are the same, they are sorted by the lexicographical order of the signal names.
- The conditions listed with sorting by time.

When the [-sort time] and [-format tabular] Options are Applied

When the `-sort time` and `-format tabular` options are applied, the error items are printed in a tabular format. The X-axis of the table represents the times of the error items. This is sorted in ascending order from left to right. The Y-axis represents the signals (or signal pairs) of the error items. This is sorted in ascending order of from top to bottom. The report is printed in plain format as follows.

```
*********************************************************************
*********************************************************************
**                                                                 **
**                  A Readable Report of nCompare                  **
**                                                                 **
*********************************************************************
*********************************************************************


^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
                          The 1st Comparison
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Golden fsdb :                                                    rtl.fsdb
Secondary fsdb :                                                gate.fsdb
The number of compared signal pairs :                                  35
The number of signals pairs with errors :                               6
Total errors :                                                         16
Golden time unit :                                                    1ns
Secondary time unit :                                                 1ns

=====================================================================
                         Mismatch results table
---------------------------------------------------------------------
/system/i_cpu/ <= (Note: hierarchy prefix for secondary signals)
     <= (Note: hierarchy prefix for golden signals. It's omitted for it's
the same as the secondary)

---------------------------------------------------------------------
               350 351      450      451      500      550      551
---------------|---|--------|--------|--------|--------|--------|------
i_ALUB/\X0[0]  <= (Note: local names of secondary signals)            1
<= (Note: actual values)
i_ALUB/X0[0]    <= (Note: local names of golden signals)             0 <=
(Note: expected values)
               |   |        |        |                 10 <= (Note:
error width)
---------------|---|--------|--------|--------|--------|--------|------
i_PCU/ALU[7:0]  |   00000000 |        00000001 |        |        00000010
               |   00000001 |        00000010 |        |        00000000
               |   11       |        25       |        |        11
```

```
----------------|------------|-----------------|--------|---------------
i_PCU/S1        |            1                 |        0
                |            0                 |        1
                |            6                 |        6
----------------|------------------------------|--------|---------------
i_PCU/TDB[7:0]  |                              00000100 |
                |                              00110100 |
                |                              5        |
----------------|---------------------------------------|---------------
i_PCU/mux_sel[+ 000                                     001
                001                                     000
                5                                       5
=============================================================================


=============================================================================
                         Unstable results table
-----------------------------------------------------------------------------
/system/i_cpu/i_CCU/
-----------------------------------------------------------------------------
          400     500     800     1100    1200    1700    1800     2100
------------|-------|-------|-------|-------|-------|-------|--------|--
TDB[7:0](S) hold    hold    hold    hold    hold    hold    hold     hold
            10      10      10      10      10      10      10       10
=============================================================================
```

The following conditions apply:

- The maximum time points in a table is specified using the `-column` option.
- In addition to the `-column` option, the number of time points in a table is also limited by the `-total` option. When the width of a page is not enough, `nce2report` dynamically determines the maximum time points in a table, so that the width of the table does not exceed the page width.
- A results table can contain only signals in the same scope using the `-scope` option. If it does, the mismatch results table is split into two tables, one for `system/i_cpu/i_PCU` and the other for `/system/i_cpu/i_ALUB`, see the example below.
- The conditions listed previously.

```
=============================================================================
                          Mismatch results table
-----------------------------------------------------------------------------
/system/i_cpu/i_ALUB/

-----------------------------------------------------------------------------
                         550
--------------------------------|--------------------------------------------
\X0[0]       1
X0[0]        0


=============================================================================
                          Mismatch results table
-----------------------------------------------------------------------------
/system/i_cpu/i_PCU/

-----------------------------------------------------------------------------
                350 351      450      451      500      550      551
----------------|---|--------|--------|--------|--------|--------|-----
ALU[7:0]    |   00000000 |        00000001 |        |        00000010
```

```
                |   00000001 |        00000010 |        |
00000000
                |   11       |        25       |        |        11
----------------|------------|-----------------|--------|-------------
S1              |            1                 |        0
                |            0                 |        1
                |            6                 |        6
----------------|------------------------------|--------|-------------
TDB[7:0]        |                              00000100 |
                |                              00110100 |
                |                              5        |
----------------|---------------------------------------|-------------
mux_sel[2:0]    000                                     001
                001                                     000
                5                                       5
=================================================================
```

When the [-sort hier] and [-format tabular] Options are Applied

When the `-sort hier` and `-format tabular` options are applied, the error items are printed in a tabular format. The X-axis of the table represents the signals (or signal pairs) of the error items, this is sorted by the ascending order from left to right. The Y-axis represents the time points of the error items, this is sorted by the ascending order from top to bottom. The report is printed in plain format as follows.

```
**************************************************************************
**************************************************************************
**                                                                      **
**                   A Readable Report of nCompare                      **
**                                                                      **
**************************************************************************
**************************************************************************
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
                          The 1st Comparison
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Golden fsdb :                                              Golden.fsdb
Secondary fsdb :                                         Secondary.fsdb
The number of compared signal pairs :                                86
The number of signals pairs with errors :                             8
Total errors :                                                       12
Golden time unit :                                                  1ns
Secondary time unit :                                               1ns

=========================================================================
                          Mismatch results table
-------------------------------------------------------------------------
/top/                                                          /system/
(Note: hierarchy prefix for secondary signals)   (Note: hierarchy prefix
for golden signals)
-------------------------------------------------------------------------
R_ <=(Note: local names of secondary signals)      (Note: local names of
golden signals)=> R_W
------------------|------------------------------------------------------
R_W               |         *
------------------|---------|--------------------------------------------
clk1              |         |         *                              clk
------------------|---------|---------|----------------------------------
VMA               |         |         |         *
------------------|---------|---------|---------|------------------------
addr[7:0]         |         |         |         |          *
------------------|---------|---------|---------|----------|-------------
350               |         0         |         0          |
```

```
                 |         1         |         1         |
                 |         5         |         5         |
-----------------|-------------------|---------|---------|-----------
4350             0                   0         |         |
                 1                   1         |         |
                 5                   5         |         |
-------------------------------------------------|---------|-----------
4650                                             1         00100000
                                                 0         01010101
                                                 5         14
=============================================================================


=============================================================================
                          Unstable results table
-----------------------------------------------------------------------------
/system/i_cpu/i_CCU/
-----------------------------------------------------------------------------
TB[7:0](S)          *
-----------|-----------------------------------------------------------------
TDB[7:0](S)         |              *
--------------------|--------------|--------------------------------------
TDB[7:0](G)         |              |                *
--------------------|--------------|--------------|--------------------
TD[7:0](S)          |              |              |              *
--------------------|--------------|--------------|--------------|-----
410                 |              cycle          cycle          |
                    |              10             10             |
--------------------|-----------------------------|--------------|----
420                 |                             setup          setup
                    |                             10             10
--------------------|-----------------------------|-------------------
430                 none                          none
                    10                            10
=============================================================================
```

The following conditions apply:

- The maximum signals (signal pairs) in a table is specified using the `-column` option.
- The conditions listed previously.

txt2fsdb

`txt2fsdb` is a utility to transfer transactions or messages from a text file to an FSDB file and set the relationship between transactions.

Usage:

```
txt2fsdb txt_file [Options] or
txt2fsdb [Options] txt_file
```

Options	Explanation
-h | -help	Prints the help message.
-msg	Records messages to a log file. When the *-msg* option is specified, logged messages are transferred from a text file to an FSDB file, and the hierarchical tree of the specified scope, stream variable, and attribute definitions are listed under the created "*msg_root*" scope in the FSDB file. Otherwise, *txt2fsdb* transfers transactions to an FSDB file.
-o *fsdb_file*	The name of the output file. Its default value is *txt_file.fsdb*.

Text File Format:

```
// Comments are preceded by "//" in this document to describe the
// different fields but "//" are not allowed in the .txt file
// that is being used for conversion.
//
// The format is similar to the VCD file. The header information
// comes first. Then the hierarchical tree of the scope, stream
// var, and attribute definition are declared. Next is the vc
// part.
//
// The keywords start with the "$" sign.
// Keyword "$end" finishes the definition.
// The available keywords are as follows:
// $data, $version, $timescale
// $scope, $stream, $attr which refers to the attribute,
// $transaction, $upscope, $enddefinitions, $dumpvars
//
// Scope Definition: the type of scope is preceded by the
// keyword "$scope" and then the scope name.
// Stream Variable Definition: define the identification code
// (idcode) and variable name.
// Attribute Definition: the type, attribute idcode, attribute
// name, MSB and LSB are described after the keyword "attr"
//
// At the value change part:
// The time tag is recorded as #tag.
// Steps for value change creation:
// 1. Reference the stream variable and indicate its idcode.
// 2. Create transaction - record the number of attributes after
```

```
//    the transaction's label; then specify the start and end
//    time.
// 3. Describe the number of attributes - the number of vc
//    elements is equivalent to the absolute value of (MSB - LSB)
in
//    the attribute definition.

//
// Date
//
$date
       Tue Dec 10 13:46:49 2002
$end
//
// version
//
$version
        Model Technology ModelSim EE/Plus vsim 5.4 Beta 4
Simulator 200
$end
//
// timescale
//
$timescale
         1fs
$end
//
// hierarchical tree
//
$scope module Top1 $end
$stream 1 stream1 $end
$stream 2 stream2 $end
//
// attribute:
// $attr ATTRIBUTE_TYPE IDCODE NAME MSB LSB $end
// $attr enum           IDCODE NAME MSB LSB LITERAL_COUNT VAL_LEN
//                      VAL_TYPE LITERAL_ARR_0 LITERAL_ARR_1...
//                      LITERAL_ARR_n
//                      VAL_ARR_0 VAL_ARR_1... VAL_ARR_n $end
//
$scope module Level1 $end
$stream 3 stream3 $end
$attr string 1 command 0 0 $end
$attr sv_logic 2 address 31 0 $end
$attr bool 3 write 0 0 $end
$attr logic 4 data 0 0 $end
$attr uint32 5 length 0 0 $end
$attr string 6 response 2 0 $end
$attr char   7 val 3 0 $end
$attr enum 8 day 3 0 7 4 integer
"Sunday" "Monday" "Tuesday" "Wednesday" "Thursday" "Friday"
"Saturday"
0 1 2 3 4 5 6
$end
```

```
$upscope $end
$enddefinitions $end
//
// value change
//
$dumpvars
#500
//
// $stream STREAM_IDCODE
// $transaction LABEL ATTR_COUNT TIME
// $attr IDCODE VAL_1 VAL_2... VAL_n ( n = | MSB - LSB + 1 |)
$stream 1
$transaction t1 8 50 100
$attr 1 "write"
$attr 2 10110000110101001000100100001110
$attr 3 1
$attr 4 0
$attr 5 60
$attr 6 "s1" "s2" "s3"
$attr 8 0 2 3 5
$attr 7 aqre
$end
#600
$stream 3
$transaction t2 4 50 300
$attr 1 "read"
$attr 8 0 3 5 2
$attr 2 01010101010101011100010101001111
$attr 7 abcd
$end
```

Figure: Transaction Results From the Example Text File

txt2uddb

`txt2uddb` is a utility to transfer transactions or messages from a text file to a UDDB file. The utility also creates the correct relationship between transactions.

NOTE: Loading gzipped (*.*gz*) files for the text file is also supported.

Usage:

```
txt2uddb txt_file [Options] or
txt2uddb [Options] txt_file
```

Options	Explanation
-h | -help	Prints the help message.
-msg	When the *-msg* option is specified, logged messages are transferred from a text file to a UDDB file. Otherwise, *txt2uddb* transfers transactions to a UDDB file.
-o *uddb_file*	Specifies the output file name. Its default value is *txt_file.uddb*.

Text File Format:

```
$scope module NOVAS_USER_ANNOT $end
$struct 1 SCH_ANNOT $end
$attr 1 User_Annot $end
$endstruct $end
$upscope $end
$scope module top $end
$struct 4 signal_N $end
$attr 2 color $end
$attr 3 fulltext $end
$attr 4 hor $end // or ($attr 4 ver $end )
$attr 5 sameNet $end //This signal should have driver or load
$endstruct $end
$scope module inst $end
$struct 2 port1_P  $end
$attr 6 color $end
$attr 7 fulltext $end
$endstruct $end
$upscope $end
$struct 3 inst2_I  $end
$attr 8 color $end
$attr 9 shape $end
$attr 10 pos $end
$endstruct $end
$scope module inst3 $end
$struct 2 pin_I $end
$attr 11 color $end
$attr 12 text $end
$endstruct $end
```

```
$upscope $end
$struct 3 inst3_I $end
$attr 13 color $end
$attr 14 shapeFile $end //full path name and need have
refreshShape option
$attr 15 refreshShape $end
$endstruct $end
$enddefinitions $end
#0
$attr 1 Yes
$attr 2 ID_RED4
$attr 3 "This is signal"
$attr 4 hor //ver(value == hor or ver respectively)
$attr 5 sameNet//(value == sameNet)
$attr 6 ID_RED5
$attr 7 "This is port"
$attr 8 ID_RED6
$attr 9 star
$attr 10 out
$attr 11 ID_RED7
$attr 12 "This is pin"
$attr 13 ID_GREEN5
$attr 14 /qa/mfg/prod/novas2013/novasv201301_1019/etc/share/icon/
xpm/iso_open.xpm
$attr 15 refreshShape(value == refreshShape)
```

vcd2fsdb

`vcd2fsdb` is a utility to convert VCD files to FSDB files. For details, see the *vfast* utility.

Usage:

```
vcd2fsdb [dump_file] [Options]
```

or

```
vcd2fsdb [Options] [dump_file]
```

When the dump file is not specified, `vcd2fsdb` obtains data available from `stdin`.

Example:

```
> cat a.vcd | vcd2fsdb
```

In the example, a `stdin.fsdb` file is created which is converted from "`a.vcd`" from `stdin`.

vfast

`vfast` is a utility to convert Verilog, Spice, xp, rawFile, wfm, and VCD files to FSDB files.

File Types	Extensions
Verilog	.vcd
gzipped VCD	.vcd.gz
Spice	.tr0
xp, rawfile, or wfm	.raw

NOTE:

1) For a multi-plot `.raw` file (for example, `Spectre`), `vfast` is used to convert it to an FSDB file. The usage is:

```
vfast <input_rawFile> -ft rawfile -o <fsdb_outFile>
```

2) The `vhdl2vcd` value change mapping is as follows:

```
u or U ---> x
x or X ---> x
0 ---> 0
1 ---> 1
z or Z ---> z
w or W ---> x
l or L ---> 0
h or H ---> 1
dash ---> x
```

Usage:

```
vfast dump_file [Options]
```

or

```
vfast [Options] dump_file
```

Options	Explanation
-allenv	Brief explanation of all environment variables.
-compact	Creates value changes in a compact format. This option reduces file size but takes longer to create the FSDB file.
-del <delimiter>	The delimiter for parsing and separating the full path signal names of analog type dump files. This option is only valid for the following types of dump files: spice, xp, rawfile, or wfm.
-direction	Enable EVCD variable direction. This option is for VCD files generated by ModelSim with the *-direction* option.
Dump_file	The name of the dump file.
-dumpoff	Enable the VCD converter to recognize the *$dumpoff* and *$dumpon* keywords.
-fileType | -ft	Specifies the type of dump file that is going to be converted to an FSDB file. The type is Verilog, spice, xp, rawfile, or wfm. Its default value is *Verilog*.
-h | -help	Prints the help message.
-hier_scopename	Parses the VCD file whose *$scope* is in hierarchical format and creates the corresponding scopes.
-hier_varname	Parses the VCD file whose *$var* is in hierarchical format and creates the corresponding scopes.
-inc	Parses the file by the incremental mode. That is, enable incremental parsing when the file is still growing.
-inst_array	The VCD file contains instance arrays in the design.
-lcase	Converts all signals' definitions to lowercase.
-not_merge_multiple_sweep	Converts each signal sweep simulated by HSPICE into a separate FSDB file instead of converting all sweeps into the same FSDB file.
-o *fsdb_file*	Specifies the name of the output FSDB file. Its default value is *dump_file.fsdb*.
-orig_esc_varname | -esc	Keeps the original escape var name in the VCD file. That is, *vfast* does not perform any conversion on var name.
-orig_scopename	Keeps the original scope name defined in the VCD file. That is, *vfast* does not perform any conversion on scope name.

Options	Explanation
-orig_varname	Keeps the original var name defined in the VCD file. That is, *vfast* does not perform any conversion on var name.
-outDir | -outdir	Specifies the directory to hold the output file. If the argument for *-o* uses an absolute path format, then *-outdir* is ignored.
-seq_num	Enables the sequence number dumping.
-sv	Supports SystemVerilog formats which means VCD fork/join blocks are converted to a struct. Without this option, the fork type scope is treated as a normal scope.
-ucase	Converts all signals' definitions to uppercase.
-vcdIDstring | -idstr	Parses the VCD ID string by handle scheme. Its default value is *idcode* scheme.
-vhdlnaming	Uses VHDL naming rules.

NOTE: `vfast` cannot be used without specifying the input file, but `stdin` can be used as the input. For example:
`cat a.vcd | vfast`

Examples:

```
vfast a.vcd -o a.fsdb
vfast a.tro -ft spice -o a.fsdb
```

xloc

The `xloc` utility is used to translate signals with X value in an FSDB file into an index file.

Usage:

```
xloc fsdb_file [Options]
```

Options	Explanation
-endTime *endTime*	Specifies the end time of the extraction. If omitted, the end time of the FSDB file is used. The default time unit is *ns*, unless another unit is specified.
-h | -help	Prints the help message.
-o *xloc_file*	Specifies the output file name. Its default value is *fsdb_file.xloc*.
-scopeFile *scopeFile*	Specifies the file containing the scopes to be extracted. In the file, each scope is listed on a separate line with a period (.) character used as the scope separator (for example, system.i_cpu). If "/*" or "/" exist, the scope and its subscopes are extracted. This option cannot be used with the *-signalFile* option.
-signalFile *signalFile*	Specifies the file containing the signals to be extracted. In the file, each signal is listed on a separate line with a period (.) character used as the scope separator (for example, system.i_cpu.clock). The wildcard (*) character is used to specify all signals in a scope. This option cannot be used with the *-scopeFile* option.
-startTime *startTime*	Specifies the start time of the extraction. If omitted, the start time of the FSDB file is used. The default time unit is *ns*, unless another unit is specified.
fsdb_file *filename*	Specifies the input FSDB file name.
-ignore_2G	Xloc FSDB if the signal count of input FSDB is over 2G. Xloc fails if the count of user interested signal is over 2G.

Examples:

1. Extract all signals with X values between 10000 and 20000ns and save to `mytime.xloc`.

   ```
   xloc myfile.fsdb -startTime 10000 -endTime 20000 -o mytime.xloc
   ```

2. Extract all signals with X values in the specified scopes and save to `myscopes.xloc`.

   ```
   xloc myfile.fsdb -scopeFile myscopes -o myscopes.xloc
   ```

where, the *myscopes* file contains:

```
system.i_cpu.i_ALUB.*
system.i_cpu.i_PLU
```

3. Extract the X values for the specified signals and save to `myfile.xloc`.

   ```
   xloc myfile.fsdb -signalFile mysigs
   ```

   where, the `mysigs` file contains the following:

   ```
   system.i_cpu.data
   system.i_cpu.addr
   system.i_cpu.i_ALUB.*
   ```

4. Extract all signals with X values for the entire FSDB file and save to `allxvalues.xloc`.

   ```
   xloc myfile.fsdb -o allxvalues.xloc
   ```

xml2fsdb

The `xml2fsdb` utility is used convert XML files to FSDB files.

Usage:

```
xml2fsdb -vip xml_file [-extension ext_file] [Options]
```

Options	Explanation
-unit *timeUnit*	Specifies the time unit. Valid time units include: *[1|10|100]<s|ms|us|ns|ps>*. The default time unit is *ns*.
-fsdb *filename*	Specifies the name of the output FSDB file. The default file name is *vip.fsdb*.
-h | -help	Prints the help message.
-vip	Indicates that XML files are VIP generated XML files.
xml_file *filename*	Specifies the XML input file name. Multiple files can be specified.
-extension *ext_file*	Specifies the path to the extension XML file. This option must be used after the XML file is specified.

Examples:

```
xml2fsdb –vip dev_agent.svt_usb_vip.xml
xml2fsdb –vip dev_agent.svt_usb_vip.xml
host_agent.svt_usb_vip.xml
xml2fsdb –vip dev_agent.svt_usb_vip.xml –extension extension.xml
xml2fsdb –vip dev_agent.svt_usb_vip.xml
host_agent.svt_usb_vip.xml –fsdb svt_usb_vip.fsdb
```

After the FSDB file is generated and loaded into the Verdi platform, the Verdi platform shows the transactions and relations.

General

This section consists of the following utilities:

- *novas_plat*

novas_plat

Use the `novas_plat` program to obtain the platform information that reflects the platform used when the system is invoked.

Example:

When invoking the system on the Solaris2 platform,

```
> novas_plat
```

returns SOLARIS2.

Symbol Libraries

The following utilities are used for symbol library creation:

- *edif2SymDB*
- *map2SymDB*
- *symDB2map*
- *syn2SymDB*

edif2SymDB

Use the `edif2SymDB` program to translate the EDIF symbol library files into a symbol library for the Verdi platform.

NOTE:
1) `-syn synFile` is needed to obtain logic information before using Trace commands and Extract FSM in *nSchema*.
2) `-scale value` (default is `-scale 1`): The value is used to translate all location values in the EDIF symbol file to correct the symbol size in the schematic tool.

Usage:

```
edif2SymDB [Option] edifFile [edifFile2 ...]
```

Options	Explanation
-h | -help	Prints the help message.
-L	Converts all cell names to lowercase.
-U	Converts all cell names to uppercase.
-l	Converts all pin names to lowercase.
-u	Converts all pin names to uppercase.
-vb	Turns on the verbose mode.
-o *libName*	Specifies the output library name.
-syn *synFile*	Specifies the SYNOPSYS library file. Multiple *-syn synFile* statements are allowed.
-syndir *synDir*	Specifies the directory of the referenced SYNOPSYS file. Multiple *-syndir synDir* statements are allowed.
-synext *synExt*	Specifies the file extension name for *-syndir*. The default of *synExt* is *lib*.
-map *mapFile*	Specifies the referenced symbol map file.

Options	Explanation
-scale *value*	Specifies the scale value for graphic location.
-scalef *scaleFile*	Specifies the file which contains the scale value.
-keep_all	Keep all shapes in the statement *portImplementation*.
-pu	Partial update; update existing symbols with the symbol shape, cell type, and port attributes only; must use with the *-map* option. The option gets the change information from the *.map* file and updates the existing symbol library specified with *-o*.
-list *listFile*	Specifies the text file containing the EDIF file list.
edifFile	Specifies the input EDIF symbol file.

Example:

```
edif2SymDB -syn a.lib -syn b.lib -syn c.lib sample.edf
edif2SymDB -syndir /home/user1 -syndir /home/user2 -synext lib \
sample.edf
edif2SymDB -syndir /home/user1 -syndir /home/user2 -synext lib \
-syn a.lib -syn b.lib sample.edf
```

List File Format

```
[-map mapFile] [-syn synFile] -edif edifFile
```

Example:

```
-map 1.map -edif 1.edf
-map 2.map -edif 2.edf
-map 3.map -edif 3.edf
....
....
....
```

map2SymDB

Use the `map2SymDB` program to translate a map file into a Verdi symbol library.

NOTE: Behavior Analysis does not support symbol libraries created with `map2SymDB`. Use `syn2SymDB` instead.

Usage:

```
map2SymDB: [Options] mapFile
```

Options	Explanation
-h | -help	Prints the help message.
-L	Converts all cell names to lowercase.
-U	Converts all cell names to uppercase.
-l	Converts all pin names to lowercase.
-u	Converts all pin names to uppercase.
-v	Generates symbol library in Verilog HDL format with the default file <*Library Name*>.*v*.
-vb	Turns on the verbose mode.
-b *genericSymfile*	Specifies the generic symbol file name instead of the default file *generic.sym*.
-o *symbolLibraryName*	Specifies the output library name.
-kp	Keep the pin order the same as the pin definition in the map file.
-pu	Partial update; update existing symbols with the symbol shape, cell type, and port attributes only. The option gets the change information from the *.map* file and updates the existing symbol library specified with *-o*.
mapFile	The mapping file or symbol file.

symDB2map

`symDB2map` is a program that is used to translate the Verdi symbol library into a map file.

Usage:

```
symDB2map libName [-out outFile] [-p]
```

Options	Explanation
-h | -help	Prints the help message.
libName	Specifies the input library name.
-out *outFile*	Specifies the output map file.
-p	Dumps all ports even if the cell is an equation cell.
-g	Dumps the library generator information.

Examples:

```
symDB2map work (or symDB2map work.lib++)
```

The output file is `workSym.map`.

```
symDB2map work -out work (or symDB2map work.lib++ -out work.map)
```

The output file is `work.map`.

syn2SymDB

Use the `syn2SymDB` program to translate Synopsys liberty and database files into a symbol library for the Verdi platform. Loading gzipped (`*.gz`) files for the Synopsys liberty (`.lib`) file is supported.

NOTE:

1) The Verdi platform only supports `.lib` files, not `.slib` files.
2) Behavior Analysis only supports symbol libraries created with `syn2SymDB`.
3) The `syn2SymDB` utility takes `nwell` and `deepnwell` as `primary_power`, and `pwell` and `deeppwell` as `primary_ground`.
4) The power information is processed automatically.

Usage:

```
syn2SymDB [Option] dbFile|libFile [dbFile2|libFile2 ...]
```

Options	Explanation
-allcase	Generates libraries with different combinations of case conversion for cell names and pin names. They are libraries with no case conversion. Libraries with cell/pin names converted to uppercases/uppercases, uppercases/lowercases, lowercases/uppercases, and lowercases/lowercases respectively.
dbFile	Specifies the input database file (*.*db*). This option cannot be used with the *libFile* option.
-E	Create the pin type 'E' for latch enable pin. The default type is 'C'.
-format db | lib	Specifies the input file format as either database (*.*db*) or liberty (*.*lib*). If this option is not specified, the default input file format is the liberty file (*.*lib*).
-h | -help	Prints the help message.
-ignorepin "pinName(s)"	Ignores the specified pin names. Pin names for state tables/flip-flops/latches are not ignored. Multiple names can be specified by using double quotes to enclose the pin names. **Examples:** > *syn2SymDB -ignorepin VDD test.lib* > *syn2SymDB -ignorepin "VDD VSS" test.lib*
libFile	Specifies the input liberty file (*.*lib*). This option cannot be used with the *dbFile* option.
-l	Converts all pin names to lowercase.
-L	Converts all cell names to lowercase.

Options	Explanation
-m	Generates the mapping file.
-nowarn <ALL | POLARITY | STATETABLE>	Suppresses warning messages. *ALL*: Suppresses all warning messages. *POLARITY*: Suppresses warning message for polarity. *STATETABLE*: Suppresses warning message for statetable.
-o *libName*	Specifies the output name for the symbol library.
-pu	Updates the cell equation for existing symbols. The option gets the new information from the *.lib* file and updates the existing symbol library specified with the *-o* option. This option cannot be used with the *-format db* option.
-skfl	Skips cells with logic expressions in *ff()*/*latch()*.
-skst	Skips cells generated by the state table.
-stop	Stops the program when the warning about information loss is issued. This option cannot be used with the *-format_db* option.
-u	Converts all pin names to uppercase.
-U	Converts all cell names to uppercase.
-vb	Turns on the verbose mode.

Timing

The following utilities (Perl scripts) are used to support standard timing report formats and SDF files.

AMBIT.pl

The `AMBIT.pl` utility supports conversion of the static timing report generated by Cadence AMBIT to XML format.

Usage:

```
AMBIT.pl [Options] report_filename
```

Options	Explanation
-h | -help	Prints the help message.
-o *xml_filename*	Specifies the output xml file name.

astro2Xml.pl

The `astrol2Xml.pl` utility supports conversion of the static timing report generated by Synopsys Astro to XML format.

Usage:

```
astro2Xml.pl [Options] report_filename
```

Options	Explanation
-h | -help	Prints the help message.
-o *xml_filename*	Specifies the output xml file name.

crmap.pl

The `cramp.pl` utility is used to partition a gate-level design based on module hierarchies (that is, a subscope under a large design can become a partition). Each gate-level partition is correlated to the original RTL design and will generate a sub mapping file. Then, all the sub mapping files are merged into a larger file. The final mapping file can be used together with the original gate-level FSDB for debugging the RTL design.

Usage:

```
crmap.pl -c filename -o filename [-fsdb filename|-gendb]
[-merge] [-dbgcmd] [-sort] [-s] [-d] [-h]
```

Options	Explanation
-c *filename*	Specifies the configuration file.
-d	Specifies the debug mode.
-dbgcmd	Generates Tcl command scripts for interactive debugging.
-fsdb *filename*	Specifies the input FSDB file.
-gendb	Generates mapped CRDBs to files.
-h | -help	Prints the help message.
-merge	Merges all the partitions into a single CRDB.
-o *filename*	Specifies the output mapping file.
-s	Specifies the silent mode.
-sort	Sorts the generated mapping file alphabetically.

Examples:

```
crmap.pl -c info-lev3 -fsdb lev3.fsdb -o lev3.map
crmap.pl -c info-lev3 -fsdb lev3.fsdb -o lev3_merge.map -merge
```

DC.pl

The `DC.pl` utility supports the conversion of the static timing report generated by Synopsys DesignCompiler to XML format.

Usage:

```
DC.pl [Options] report_filename
```

Options	Explanation
-h | -help	Prints the help message.
-o *xml_filename*	Specifies the output XML file name.

EinsTimer.pl

The `EinsTimer.pl` utility supports the conversion of the static timing report generated by IBM EinsTimer to XML format.

Usage:

```
EinsTimer.pl [Options] report_filename
```

Options	Explanation
-h | -help	Prints the help message.
-o *xml_filename*	Specifies the output XML file name.

HAL2Xml.pl

The `HAL2Xml.pl` utility supports the conversion of the static timing report generated by Cadence Incisive HDL analysis (HAL) to XML format.

Usage:

```
HAL2Xml.pl [Options] report_filename
```

Options	Explanation
-h | -help	Prints the help message.
-o *xml_filename*	Specifies the output XML file name.

magma.pl

The `magma.pl` utility supports the conversion of the static timing report generated by Synopsys (MAGMA) Tekton to XML format.

Usage:

```
magma.pl [Options] report_filename
```

Options	Explanation
-h | -help	Prints the help message.
-o *xml_filename*	Specifies the output XML file name.

pearl.pl

The `pearl.pl` utility supports the conversion of the static timing report generated by Cadence Pearl Timing Analysis to XML format.

Usage:

```
pearl.pl [Options] report_filename
```

Options	Explanation
-h | -help	Prints the help message.
-o *xml_filename*	Specifies the output XML file name.

pt2Xml.pl

The `pt2Xml.pl` utility supports conversion of the timing report generated by Synopsys PrimeTime to XML format. This utility only accepts one input report file name but this file can contain the results from multiple timing runs.

With XML format, PrimeTime timing reports can be loaded into the Verdi platform.

Usage:

```
pt2Xml.pl [Options] report_filename
```

Options	Explanation
-flat	Treats the last escape character ("/") as the delimiter.
-h | -help	Prints the help message.
-o *xml_filename*	Specifies the output XML file name. *-o* is optional. When *-o* is not specified, the default for the XML file output name is *output.xml*. When *-o* is specified, the output XML file name is the specified file name (for example, *myfile.xml*). For a timing report file with multiple report sessions, an underscore number is automatically appended to the output file name for each additional report. For example, with the *-o* option, the output file names are *myfile.xml*, *myfile_2.xml*, *myfile_3.xml*, etc. Without the *-o* option, the output file names are *output.xml*, *output_2.xml*, *output_3.xml*, etc.

NOTE: Usage of `pt2xml.pl` Perl script file in WinNT platform:
1) Before using the `pt2xml.pl` Perl script to translate the PrimeTime timing report to the XML file, the Perl program must be installed on its

platform. Visit the *http://www.perl.com* website to obtain the latest version of the Perl interpreter. If the version of the Perl interpreter is prior to version 5.0, extra Perl commands must be put before the `pt2xml.pl` script file.
Syntax:

```
c:\ perl pt2xml.pl pt_timing_path.rpt
-o output.xml
```

2) To make this utility more user-friendly, open Windows Explorer, click the **Tools** menu, select **Options**, then the **File Types** folder. Select *pl* as the extension name to enter the following:

```
C:\Perl\bin\Perl.exe "%1" %*
```

The first part is the full path name of the Perl executable.

RTLC2Xml.pl

The `RTLC2Xml.pl` utility supports the conversion of the timing report generated by Cadence RTL Compiler to XML format.

Usage:

```
RTLC2Xml.pl [Options] report_filename
```

Options	Explanation
-h | -help	Prints the help message.
-o *xml_filename*	Specifies the output XML file name.

SE2Xml.pl

The `SE2Xml.pl` utility supports the conversion of the timing report generated by Cadence SOC Encounter to XML format.

Usage:

```
SE2Xml.pl [Options] report_filename
```

Options	Explanation
-h | -help	Prints the help message.
-o *xml_filename*	Specifies the output XML file name.

Example:

```
SE2Xml.pl -o output.xml test.rpt
```

sdfin

`sdfin` is a batch mode compiler for SDF files. `sdfin` parses and saves the SDF file into a `*.ddb` file.

NOTE:

1) The data in the old `.ddb` file is replaced by the new `.ddb` data if the names are the same.
2) If the `-incr` option is specified, the `-out` option does not need to be specified. The `-incr` option implies the output file.
3) If the length of the delay value list is more than 12, a warning message is displayed:

```
*WARN* the length of the delay value list is over 12, only
12 values are kept.
```

Usage:

```
sdfin [-vhdl] [-mem N] [-hier divider hierarchy] [-out name]
[-incr ddbFile] sdfFile [sdfFile ... ]
```

Options	Explanation
-h | -help	Prints the help message.
-vhdl	Uses VHDL rules to translate special characters. The default uses Verilog rules.
-mem N	Specifies how much memory can be used in Megabytes unit. Its default value is 512MBytes (N = 512) and the minimum is 64MBytes.
-hier *hierarchyName*	Only parse items under the specified hierarchy (not including the design name). Only one `-hier` option is allowed. A forward slash (/) character or period (.) character is used as the hierarchy divider character. The hierarchy must be in SDF naming format.
-out *ddbFile*	Specifies the output file name. If the same file name is used, it is overwritten with new results.
-incr *ddbFile*	Merges the compiled data into the existing *ddbFile*. If the output file is not specified, the usage information for this option is displayed.
-vb	Verbose mode. The details are printed to the screen. The output message can be redirected such as *sdfin -vb CPU.sdf > out*. The output format for the *-vb* option is: *<cellType> <instance>* *[Skipped --- %s(line %d)]*

Options	Explanation
-design *designName*	Changes the design name.
sdfFile	Inputs file in the SDF format. If more than one SDF file exists, either *-out* or *-incr* is specified.

Example 1

```
sdfin -vhdl alu.sdf
```

- Use VHDL rules to translate special characters `A\[0\] --> \A[0]\ (Verilog: "\A[0] ")`.
- Output file: `alu.ddb`

Example 2

```
sdfin -mem 32 test1.sdf
```

- 64MBytes memory is used (although 32 is specified, the minimum is 64MB).
- Output file: `test1.ddb`

Example 3

```
sdfin -mem 256 -hier / top/CPU/ffp CPU.sdf
```

- 256MBytes memory is used.
- Parses items under `top/CPU/ffp`; does not parse with `top/CPU/ffu`.
- Output file: `CPU.ddb`

Example 4

```
sdfin -out CPU1.ddb CPU.sdf
```

- Input file: `CPU.sdf`
- Output file: `CPU1.ddb`

Example 5

```
sdfin -incr CPU.ddb CPU_1.sdf
```

- Compile the `CPU_1.sdf` file and merge its data into `CPU.ddb`.
- Output file: `CPU.ddb`
- If it exists, the original `CPU.ddb` file is renamed as `CPU.ddb.old`.

Example 6

```
sdfin CPU_1.sdf CPU_2.sdf
```

- Syntax error. Either `-out` or `-incr` should be specified.

Example 7

```
sdfin -out CPU2.ddb -incr CPU1.ddb CPU_2.sdf
```

- Compile `CPU_2.sdf` into `CPU1.ddb` with incremental mode.
- The `-out` option is overridden by the `-incr` option.
- Output file: `CPU1.ddb`

Example 8

```
sdfin -vb CPU.sdf
```

- Verbose results similar to:

```
del_spec device
BUF 1587
net_delay delay_1
Skipped --- NETDELAY(line 301)
```

The instance may be empty, so the output may be the following:

```
del_spec
BUF
net_delay delay_1
Skipped --- NETDELAY(line 301)
```

- Output file: `CPU.ddb`

Example 9

```
[SDF File: test.sdf]
(DELAYFILE
(SDFVERSION "3.0")
(DESIGN "TOP")
....
....
)

// The design name is "TOP".
sdfin -design NEWTOP test.sdf
```

- Change the design name from `TOP` to `NEWTOP`.

Appendix A: Environment Variables

This chapter consists of the environment variables related to the following:

- *FSDB - General*
- *FSDB - Corresponding Dump Options*
- *Compiler*
- *Verdi*
- *Siloti - Essential Signal Analysis*
- *Siloti - Data Expansion*
- *Siloti - Correlation*
- *VC Apps*

NOTE: To obtain the mapping information between old and new environment variables, execute the Verdi command-line option, `-envmap`.

FSDB - General

This section consists of the details of the following environment variables:

- *FSDB_CHAIN_FLUSH_THRESHOLD*
- *FSDB_DUMP_INFPORT*
- *FSDB_DUMP_LOG*
- *FSDB_GATE*
- *FSDB_NO_PARALLEL*
- *FSDB_NO_SIM_VERSION_INFO*
- *NOVAS_FSDB_BUSY_BAR_MAX_WAIT_SECONDS*
- *NOVAS_FSDB_DECOMPRESS_FILE_LOC*
- *NOVAS_FSDB_DEFAULT_FILE_NAME*
- *NOVAS_FSDB_ENV_CALC_MIN_TIME_STEP_ INCREMENT*
- *NOVAS_FSDB_ENV_DISABLE_OLD_FSDB_VER_MSG*

FSDB_CHAIN_FLUSH_THRESHOLD

Description

Set this environment variable to a value to specify the number of sessions to accumulate before flushing. Its default value is *1*.

During simulation, the FSDB writer accumulates the received value changes in a chain-like data structure in memory before actually writing them to the target FSDB file. When the accumulated data reaches a certain amount of units (named a session) in the chain, the data is flushed out of the chain and appended to the FSDB file. There is an overhead on the disk I/Os when the data is flushed out of the chain to the FSDB file. However, without flushing, you cannot open the FSDB file during simulation because the data is kept in memory until the simulation is complete.

Currently, the chain flushing mechanism is enabled by default. It can be turned off for better dumping performance if there is no need to open the FSDB file during simulation. Alternatively, if the FSDB file needs to be opened during simulation, the value can be set larger so that flushing occurs less frequently to gain more performance.

Syntax

```
FSDB_CHAIN_FLUSH_THRESHOLD = value
```

Example

`setenv FSDB_CHAIN_FLUSH_THRESHOLD 3`
Flushes the chain data structure after three sessions are accumulated in the chain.

`setenv FSDB_CHAIN_FLUSH_THRESHOLD 0`
Disables chain flushing; however, this means the FSDB file cannot be opened during simulation.

FSDB_DUMP_INFPORT

Description
Enables dumping the interface port and modport port.

Syntax
```
FSDB_DUMP_INFPORT
```

Example
```
setenv FSDB_DUMPER_INFPORT
```

FSDB_DUMP_LOG

Description
Turns off the `novas_dump.log` creation. Its default value is *on*.

Syntax
```
FSDB_DUMP_LOG = on|off
```

Example
```
setenv FSDB_DUMP_LOG off
```

FSDB_GATE

Description
Set to *1* to enable FSDB gate features. Set to *0* to disable FSDB gate features. The default is *1*.

Syntax
```
FSDB_GATE = 0|1
```

Example
```
setenv FSDB_GATE 0
```

FSDB_NO_PARALLEL

Description
Disables the parallel dumping scheme.

Syntax
```
FSDB_NO_PARALLEL
```

Example
```
setenv FSDB_NO_PARALLEL
```

FSDB_NO_SIM_VERSION_INFO

Description
Disables the warning message when the simulator version and the Verdi version mismatch, or when the version for the supported simulator is in the beta stage.

Syntax
```
FSDB_NO_SIM_VERSION_INFO
```

Example
```
setenv FSDB_NO_SIM_VERSION_INFO
```

NOVAS_FSDB_BUSY_BAR_MAX_WAIT_SECONDS

Description
Sets the maximum waiting time (in seconds) before the busy bar is killed.

Syntax
```
NOVAS_FSDB_BUSY_BAR_MAX_WAIT_SECONDS = number
```

Example
```
setenv NOVAS_FSDB_BUSY_BAR_MAX_WAIT_SECONDS 10
```

NOVAS_FSDB_DECOMPRESS_FILE_LOC

Description
Specifies a file location for the generated decompressed FSDB file.

Syntax
```
NOVAS_FSDB_DECOMPRESS_FILE_LOC = /a/b
```

Example
```
setenv NOVAS_FSDB_DECOMPRESS_FILE_LOC /a/b
```

NOVAS_FSDB_DEFAULT_FILE_NAME

Description
Changes the file name of the FSDB default file name. The default FSDB file name is `novas.fsdb`.

Syntax
```
NOVAS_FSDB_DEFAULT_FILE_NAME = filename
```

Example
```
setenv NOVAS_FSDB_DEFAULT_FILE_NAME filename
```

NOVAS_FSDB_ENV_CALC_MIN_TIME_STEP_INCREMENT

Description
Turns on the minimum time step increment calculation in the FSDB writer. Its default value is *off*.

Syntax
```
NOVAS_FSDB_ENV_CALC_MIN_TIME_STEP_INCREMENT = on|off
```

Example
```
setenv NOVAS_FSDB_ENV_CALC_MIN_TIME_STEP_INCREMENT on
```

NOVAS_FSDB_ENV_DISABLE_OLD_FSDB_VER_MSG

Description

Set this environment variable to a non-null value to hide the older versions of FSDB file-related messages when an older version of the FSDB file is opened.

Syntax

```
NOVAS_FSDB_ENV_DISABLE_OLD_FSDB_VER_MSG = on
```

Example

```
setenv NOVAS_FSDB_ENV_DISABLE_OLD_FSDB_VER_MSG on
```

NOVAS_FSDB_ENV_NOVAS_LOCK

Description

Set the environment variable to a non-null value before starting the simulation. This enables the additional locking mechanism that combats the existence of bugs in the locking system call of some UNIX clones, that is, Linux.

Syntax

```
NOVAS_FSDB_ENV_NOVAS_LOCK = on
```

Example

```
setenv NOVAS_FSDB_ENV_NOVAS_LOCK on
```

NOVAS_FSDB_ENV_SYNC_CONTROL

Description

Set this environment variable to *off* to disable synchronization and block viewing of waveforms while dumping. The Verdi platform uses this environment variable to disable the lock scheme, as some operating systems cannot handle it.

Syntax

```
NOVAS_FSDB_ENV_SYNC_CONTROL = off
```

Example

```
setenv NOVAS_FSDB_ENV_SYNC_CONTROL off
```

NOVAS_FSDB_IGNORE_VHDL_COMPLEX

Description

Set this environment variable to a non-zero value to disable dumping of complex signal types, such as records, multi-dimensional arrays, and array of records. For designs with many complex signals, this can save simulation time and reduce the FSDB file size. This system environment variable is only effective in FSDB dumpers that support VHDL designs.

Syntax

```
NOVAS_FSDB_IGNORE_VHDL_COMPLEX = non-zero value
```

Example

```
setenv NOVAS_FSDB_IGNORE_VHDL_COMPLEX 1
```

NOVAS_FSDB_NO_UNFINISHED_ATTEMPT

Set this environment variable to counteract the existence of incorrect property results when VCS's runtime option `-assert filter` is used.

Syntax

```
NOVAS_FSDB_NO_UNFINISHED_ATTEMPT = 1
```

Example

```
setenv NOVAS_FSDB_NO_UNFINISHED_ATTEMPT 1
```

NOVAS_FSDB_NOT_USE_VHDL_VDA

Description

Do not use the IUS VDA mechanism. This environment variable affects IUS users only.

Syntax

```
NOVAS_FSDB_NOT_USE_VHDL_VDA = 1
```

Example

```
setenv NOVAS_FSDB_NOT_USE_VHDL_VDA 1
```

NOVAS_FSDBLOG_DUMP_ATTR_TWOROWS

Description
Writes attributes to two rows when using `$fsdbLog`. This environment variable affects `$fsdbLog` only.

Syntax
```
NOVAS_FSDBLOG_DUMP_ATTR_TWOROWS = 1
```

Example
```
setenv NOVAS_FSDBLOG_DUMP_ATTR_TWOROWS 1
```

NOVAS_FSDBLOG_NO_CALLSTACK

Description
Disables the dumping of call stacks when using the `$fsdbLog` dumping command.

Syntax
```
NOVAS_FSDBLOG_NO_CALLSTACK
```

Example
```
setenv NOVAS_FSDBLOG_NO_CALLSTACK
```

NOVAS_FSDBLOG_NO_VARIABLE_IN_LOG

Description
Disables variable logging when using `$fsdbLog`. This environment variable affects `$fsdbLog` only, and prints the "`$fsdbLog Dump String Var Only`" message to the console if triggered.

Syntax
```
NOVAS_FSDBLOG_NO_VARIABLE_IN_LOG = 1
```

Example
```
setenv NOVAS_FSDBLOG_NO_VARIABLE_IN_LOG 1
```

FSDB - Corresponding Dump Options

The environment variables in this section have corresponding dump options on the simulator command line and with FSDB dumping commands. There are three methods to specify the dump options: 1) On the simulator command line, 2) With an environment variable, 3) In an FSDB dumping command. If the option is set with more than one method, the priority is Method 1 > Method 2 > Method 3. For more details, see the *Command Line, Environment Variable, and Dumping Command Option Mapping* table in the *Linking Novas Files With Simulators and Enabling FSDB Dumping* document.

This section explains the following variables:

- FSDB_DUMP_PSL_PROP
-

FSDB_DUMP_PSL_PROP

Description

Set this environment variable to dump the PSL property.

Syntax

```
FSDB_DUMP_PSL_PROP = 1
```

Example

```
setenv FSDB_DUMP_PSL_PROP 1
```

FSDB_ESDB

Description

Specifies the Essential Signal Database (ESDB) file name. The FSDB dumper uses essential signals in this file to determine which signals to dump.

NOTE: The ESDB file name is the file generated by the `esa` utility with the `-db` option.

Syntax

```
FSDB_ESDB = filename
```

Example

```
setenv FSDB_ESDB "es"
```

FSDB_FORCE

Description

This environment variable is for Verilog and VHDL. Set this environment variable to enable dumping the force status of the Verilog and VHDL signals.

NOTE: This environment variable is to be used with VCS only.

Syntax

```
FSDB_FORCE
```

Example

```
setenv FSDB_FORCE 1
```

NOTE: In addition to the environment variable, an alternative way to dump force status is to use the `+fsdb+force` runtime option during simulation.

Example

```
simv +fsdb+force
```

Limitations

Following are the limitations inherited from VCS for debugging the forced signals:

- Language force for VHDL variable is not supported as the VCS simulation does not support it.
- UCLI force for VHDL variable only supports **Deposit** option as VCS simulation does not support **Force** and **Release** variables for VHDL in UCLI. Also, initial force state is not available for the VHDL variable.
- The file name and the line number information for external force in pure VHDL is not available due to limitation in VCS for the force list support.

FSDB_FUNCTIONS

Description

Set this environment variable to enable dumping signals in the function and task.

NOTE: The `-debug_access+dmptf` option must be specified for VCS compile before setting the `FSDB_FUNCTIONS` environment variable.

Syntax

```
FSDB_FUNCTIONS
```

Example

```
setenv FSDB_FUNCTIONS
```

FSDB_IO_ONLY

Description

Dumps only input/output port signals.

Syntax

```
FSDB_IO_ONLY
```

Example

```
setenv FSDB_IO_ONLY
```

FSDB_LOG_FILE

Description

Specifies the FSDB file name for the `$fsdbLog` dumping command.

Syntax

```
FSDB_LOG_FILE = filename
```

Example

```
setenv FSDB_LOG_FILE mylog.fsdb
```

FSDB_NO_PSL_SCOPE

Description

Set this environment variable to skip dumping the PSL scope named `cds_assertion_models` that is generated automatically in the simulator.

Syntax

```
FSDB_NO_PSL_SCOPE
```

Example

```
setenv FSDB_NO_PSL_SCOPE
```

FSDB_NOESD

Description

Set this environment variable to skip performing Essential Signal Dumping (ESD) for the `fsdbDumpvarsByFile` dumping command.

Syntax

```
FSDB_NOESD
```

Example

```
setenv FSDB_NOESD
```

FSDB_REG_ONLY

Description

Set this environment variable to dump only register-type signals.

Syntax

```
FSDB_REG_ONLY
```

Example

```
setenv FSDB_REG_ONLY
```

FSDB_REGION

Description

Set this environment variable to enable region mode dumping.

NOTE: Region mode dumping only works for VCS 2013.06 and later version.

Syntax

```
FSDB_REGION
```

Example

```
setenv FSDB_REGION
```

NOVAS_FSDB_ALL

Description

Dumps all signals including the memory, MDA, packed array, structure, union and packed structure signals in all scopes specified in the `$fsdbDumpvars` dumping command or the entire design when no scope is specified in the `$fsdbDumpvars` dumping command.

Syntax

```
NOVAS_FSDB_ALL = 1
```

Example

```
setenv NOVAS_FSDB_ALL 1
```

NOVAS_FSDB_ENV_DUMP_SEQ_NUM

Description

Set this environment variable to a non-null value to enable sequence dumping. Sequence dumping records the order of value changes that occur on different signals simultaneously. Enabling sequence dumping may increase the size of the FSDB file. Unset this environment variable to disable sequence dumping.

Syntax

```
NOVAS_FSDB_ENV_DUMP_SEQ_NUM = non-null value
```

Example

```
setenv NOVAS_FSDB_ENV_DUMP_SEQ_NUM on
```

NOTE: For Verilog and mixed Verilog/VHDL simulations, the plus options can also be used to add this environment variable.

Option	Example
`+fsdb+sequential`	`verilog -f run.f +fsdb+sequential`

NOVAS_FSDB_ENV_MAX_GLITCH_NUM

Description

Specifies how to store glitches in the FSDB file. This environment variable can be set to a value from 0 to 254. After loading an FSDB file with glitches, the **View** -> **Display Glitch** command in *nWave* must be enabled to see the glitch locations in the waveform.

Syntax

```
NOVAS_FSDB_ENV_MAX_GLITCH_NUM = number
```

Argument

The `number` argument can have any of the following values:

0: Indicates that all glitches are stored.

1: Indicates that the last glitch is stored.

2: Indicates that the first and the last glitch are stored.

n: Indicates that the first n-1 and last glitch are stored.

Example

```
setenv NOVAS_FSDB_ENV_MAX_GLITCH_NUM 0
```

NOTE: For Verilog and mixed Verilog/VHDL simulations, the plus options can also be used to add this environment variable.

Option	Example
`+fsdb+glitch=num`	`verilog -f run.f +fsdb+glitch=0`

NOVAS_FSDB_ENV_WRITER_MEM_LIMIT

Description

Sets the value of the environment variable to change the memory limit. When a value change is created, the FSDB writer stores it to memory. If a certain amount of value changes is accumulated, the FSDB writer flushes all of the value changes to a file and then recycles the memory. The amount of value changes is referred to as the *value change memory limit*.

Syntax

```
NOVAS_FSDB_ENV_WRITER_MEM_LIMIT = number
```

Argument

The `number` argument can have any of the following values:

The default is *64* when the FSDB file has less than 5M of signals.
The default is *128* when the FSDB file has 5M - 10M of signals.
The default is *256* when the FSDB file has more than 10M of signals.
The valid range is between 4 to 2048. The units are in MB.

Example

```
setenv NOVAS_FSDB_ENV_WRITER_MEM_LIMIT 20
```

NOVAS_FSDB_FILE

Description

Specifies the FSDB file name. If not specified, the default FSDB file name is *novas.fsdb*. This environment variable takes precedence over filename options (`+fsdbfile+...`) specified in the design.

Syntax

```
NOVAS_FSDB_FILE = filename
```

Example

```
setenv NOVAS_FSDB_FILE filename
```

NOVAS_FSDB_MDA

Description

Dumps all memory and MDA signals in all scopes specified in `$fsdbDumpvars` or the entire design when no scope is specified. This environment is covered by the `FSDB_ALL` environment variable and is effective for SystemVerilog MDAs only.

Syntax

```
NOVAS_FSDB_MDA = 1
```

Example

```
setenv NOVAS_FSDB_MFA 1
```

NOVAS_FSDB_PACKEDMDA

Description

Dumps only the packed signals (without array and memory types) in all scopes specified in `$fsdbDumpvars` or the entire design when no scope is specified. This environment variable is effective for SystemVerilog MDAs only.

NOTE: This environment variable replaces the `NOVAS_FSDB_MDA_PACKONLY` environment variable.

Syntax

```
NOVAS_FSDB_PACKEDMDA = 1
```

Example

```
setenv NOVAS_FSDB_PACKEDMDA 1
```

NOVAS_FSDB_NO_FUNCTIONS

Description

Disable dumping functions in the design. This option is effective for SystemVerilog function syntax only.

Syntax

```
NOVAS_FSDB_NO_FUNCTIONS = 1
```

Example

```
setenv NOVAS_FSDB_NO_FUNCTIONS 1
```

NOVAS_FSDB_PARAMETER

Description

Enables parameter dumping in all scopes specified in `$fsdbDumpvars` or the entire design when no scope is specified. This option is covered by the `+all` option and is effective for SystemVerilog parameter syntax only.

Syntax

```
NOVAS_FSDB_PARAMETER = 1
```

Example

```
setenv NOVAS_FSDB_PARAMETER 1
```

NOVAS_FSDB_RESERVE

Description

It does not overwrite the existing FSDB file in the current directory. If the upcoming FSDB file shares the same name as the reserved file, the upcoming file is renamed as `xxx_r0.fsdb`.

Syntax

```
NOVAS_FSDB_RESERVE = 1
```

Example

```
setenv NOVAS_FSDB_RESERVE 1
```

NOVAS_FSDB_SKIP_CELL_INSTANCE

Enables dumping of cell instances (that is, `` `celldefine ...`endcelldefine``) where the meaning of mode varies depending on different dumping commands, as shown in the following table:

mode	$fsdbDumpvars $fsdbDumpvarsByFile	$fsdbDumpMDA	$fsdbDumpSVA	$fsdbDumpvars with ESDB
0	Disables functionality.			

1	Skips all cell information.	Skips all cell information.	Skips all cell information.	Dumps out/inout ports of cell instances.
2	Dumps all ports of cell instances.	Skips all cell information.	Skips all cell information.	Dumps out/inout ports of cell instances.
3	Keeps dumping –v/-y library cells.	Skips library cell information.	Skips library cell information.	Keeps dumping –v/-y library cells.
4	Dumps out/inout ports of –v/-y library cells.	Skips library cell information.	Skips library cell information.	Dumps out/inout ports of –v/-y library cells.
Others	Ignore			

NOTE: The primitive cells compiled through the simulator compile option `-v` or `-y` are treated as library cells.

NOTE: To skip cell instances, use the `NOVAS_FSDB_SKIP_CELL_INSTANCE` environment variable or the `+fsdb+skip_cell_instance=mode` runtime option. For example:
```
simv +fsdb+skip_cell_instance=1
```

NOTE: If `NOVAS_FSDB_SKIP_CELL_INSTANCE` is *on* and an essential signal is inside a cell instance, the output and inout ports of the cell instance are dumped and the essential signal inside the cell instance is not dumped.

Syntax
```
NOVAS_FSDB_SKIP_CELL_INSTANCE = number
```

Example
```
setenv NOVAS_FSDB_SKIP_CELL_INSTANCE 1
```

NOVAS_FSDB_STRENGTH

Description
Enables strength dumping in all scopes specified in `$fsdbDumpvars` or the entire design when no scope is specified. This option is covered by the `+all` option and is effective for SystemVerilog syntax only.

Syntax
```
NOVAS_FSDB_STRENGTH = 1
```

Example
```
setenv NOVAS_FSDB_STRENGTH 1
```

NOVAS_FSDB_STRUCT

Description
Dumps structure, MDA structure (that is, `st2[0:1]`), MDA in structure (that is, `st3.r[1:0][2:0]`), and packed MDA in all scopes specified in `$fsdbDumpvars` or the entire design when no scope is specified. Unpacked array/ MDA is not dumped. This option is covered by the `+all` option and is effective for SystemVerilog structure syntax only.

Syntax
```
NOVAS_FSDB_STRUCT = 1
```

Example
```
setenv NOVAS_FSDB_STRUCT 1
```

NOVAS_FSDB_TRACE_PROCESS

Description
Enables dumping processes for VHDL designs. This option is effective for VHDL or mixed languages with VHDL processes.

Syntax
```
NOVAS_FSDB_TRACE_PROCESS = 1
```

Example
```
setenv NOVAS_FSDB_TRACE_PROCESS 1
```

NOVAS_FSDB_TRANS_BEGIN_CALLSTACK

Description
Records call stacks to the `BeginTime_Call_Stack` attribute of the transaction. When the call stacks are recorded, the transaction is dragged from the *nWave*

window and dropped to the *FSDB Message Pane* (`FSDB_Msg`) to jump to the calling line.

Syntax

```
NOVAS_FSDB_TRANS_BEGIN_CALLSTACK = 1
```

Example

```
setenv NOVAS_FSDB_TRANS_BEGIN_CALLSTACK 1
```

NOVAS_FSDB_TRANS_END_CALLSTACK

Description

Records call stacks to the `EndTime_Call_Stack` attribute of the transaction. When the call stacks are recorded, the transaction is dragged from the *nWave* window and dropped to the *FSDB Message Pane* (`FSDB_Msg`) to jump to the calling line.

Syntax

```
NOVAS_FSDB_TRANS_END_CALLSTACK = 1
```

Example

```
setenv NOVAS_FSDB_TRANS_END_CALLSTACK 1
```

NOVAS_FSDB_TRANS_ERROR_CALLSTACK

Description

Dumps the full call stack for each transaction error message from the Novas object files for FSDB dumping.

Syntax

```
NOVAS_FSDB_TRANS_ERROR_CALLSTACK = 1
```

Example

```
setenv NOVAS_FSDB_TRANS_ERROR_CALLSTACK 1
```

NOVAS_FSDB_TRANS_HIDE_ERROR

Description

Hides all transaction error messages from the Novas object files for FSDB dumping.

Syntax

```
NOVAS_FSDB_TRANS_HIDE_ERROR = 1
```

Example

```
setenv NOVAS_FSDB_TRANS_HIDE_ERROR 1
```

NOVAS_FSDB_VC_BLK_SIZE

Description

Specifies the size of the value change block. This environment variable is used to dump a smaller FSDB file.

Syntax

```
NOVAS_FSDB_VC_BLK_SIZE = number
```

Example

```
setenv NOVAS_FSDB_VC_BLK_SIZE 256
```

NOTE:

1. If `number` is bigger than 32, the size benefit may be bigger than the expectation. Not all numbers result in the size benefit.
2. There is a performance penalty (dumping speed is slower). The bigger the `number` value, the more obvious the performance penalty becomes.

Compiler

This section consists of the details of the following environment variables:

NOVAS_AUTOENDIF_AT_EOF

Description

Sets this environment variable to append `endif at the end of a file automatically if a matched `endif cannot be found for a `ifdef, `ifndef, `else, or `elsif in the file.

Syntax

```
NOVAS_AUTOENDIF_AT_EOF = 1
```

Example

```
setenv NOVAS_AUTOENDIF_AT_EOF 1
```

INC_THREAD_MODE

Description

Set this environment variable to decrease the compilation time from the *vericom* utility.

NOTE: This environment variable should be used with the `-incsave` option on the *vericom* command line.

Syntax

```
INC_THREAD_MODE = 1
```

Example

```
setenv INC_THREAD_MODE 1
vericom top.v -incsave
```

MODELSIM

Description

Specifies the path where the `modelsim.ini` file is located for the *gencom* utility.

NOVAS_PROTECTED_DOWNGRADE

Description

Downgrades an error message to a warning message when reporting an encrypted signal with definition protected is used.

Syntax

```
NOVAS_PROTECTED_DOWNGRADE = 1
```

Example

```
setenv NOVAS_PROTECTED_DOWNGRADE 1
```

NRUN_CSTYLE_COMMENT

Description

Set this environment variable to enable the *vericom* utility to recognize the "*" as part of the comment. Its default value is *0*.

NOTE: When the `-wcFile` option is specified on the *vericom* command line and this environment variable is set to *1*, the "*" is recognized as part of the comment. When the `-wcFile` option is specified and this environment variable is set to *0*, the "*" is recognized as a wildcard character.

Syntax

```
NRUN_CSTYLE_COMMENT = 0|1
```

Example

```
setenv NRUN_CSTYLE_COMMENT 1
```

SNPS_VCS_UFE_KDB

Description

Set this environment variable to automatically add the `-kdb` option to all `vlogan`, `vhdlan` and `vcs` commands in a flow. It is useful for big designs or third-party IPs where there are many scripts and it is not easy to locate the makefiles and individual commands.

Syntax

```
setenv SNPS_VCS_UFE_KDB 1
```

VCS_HOME

Description

Specifies the VCS installation path to access VCS libraries.

Syntax

```
VCS_HOME = vcs_installation_path
```

Example

```
setenv VCS_HOME /simulator/Synopsys/VCS_MX_vH-2013.06/
vcs_mx_vH-2013.06
```

VCS_OVM_HOME

Description

Specifies the OVM library directory.

Syntax

```
VCS_OVM_HOME = path_to_ovm_directory
```

Example

```
setenv VCS_OVM_HOME /<path_to_ovm_library>/myOVM-2.1.2
```

VCS_UVM_HOME

Description

Specifies the UVM library directory.

Syntax:

```
VCS_UVM_HOME = path_to_uvm_directory
```

Example:

```
setenv VCS_UVM_HOME /<path_to_uvm_library>/myUVM-1.1
```

VCS_UVM_HOME

Description

Specifies the UVM library directory.

Syntax:

```
VCS_UVM_HOME = path_to_uvm_directory
```

Example:

```
setenv VCS_UVM_HOME /<path_to_uvm_library>/myUVM-1.1
```

VERDI_MAX_SRCDOC_LENGTH

Description

Set this environment variable to a value to avoid compilation errors by configuring the maximum length of file names. Its default value is *180* and the default works for a regular compilation flow. When the length of the file name exceeds 180, a special flow is applied to avoid compilation errors.

In cases when the compilation fails because of long file names that result from an OS limitation, the specified value is reduced to force running with the special compile flow, but the minimum value is *64*.

NOTE: The special flow causes performance overhead. It is not recommended to set this environment variable unless an OS limitation occurs.

Syntax

```
VERDI_MAX_SRCDOC_LENGTH = number
```

Example

```
setenv VERDI_MAX_SRCDOC_LENGTH 150
```

Verdi

This section consists of the details of the following environment variables:

- *VERDI_HOME*
- *NOVAS_HT_TRACE*
- *NOVAS_ICC_CONN_FLOAT_PORT*
- *NOVAS_ICC_FORMAT*
- *NOVAS_ICC_FREEZE_MODE*
- *NOVAS_ICC_FULL_PATH*
- *NOVAS_ICC_LIB_PATH*
- *NOVAS_IDLE_LICENSE_CHECKBACK*
- *NOVAS_IDLE_LICENSE_CHECKBACK_SILENCE*
- *NOVAS_INTERACTIVE_MAP_DIR*
- *NOVAS_LANG_VER*
- *NOVAS_LIBPATHS*
- *NOVAS_LIBS*
- *NOVAS_LIC_EXP_CTRL*
- *NOVAS_LICENSE_FILE*
- *NOVAS_LICENSE_QUEUE*
- *NOVAS_MAGMA_FLAT_DB*
- *NOVAS_MANAGE_RC_DIR*
- *NOVAS_MAX_ITEMS_IN_FINDSIGNAL_FORM*
- *NOVAS_MAX_LOGFILE_SIZE*
- *NOVAS_MDT_LIBPATHS*
- *NOVAS_MDT_LIBS*
- *NOVAS_NO_AUTO_ICONIZE*
- *NOVAS_NO_BUSY_DIALOG*
- *NOVAS_NO_ENTER_FOCUS*
- *NOVAS_NOT_FOUND_SAVE*
- *NOVAS_PCELL_EXTRACTION*
- *NOVAS_PURIFY_POOL_ARENA*
- *NOVAS_QUESTA_66_GENFOR*
- *NOVAS_RC*
- *NOVAS_SCHEMA_LOOP_LEVEL*
- *NOVAS_SDC_LIMIT*
- *NOVAS_SESSION_KEEP_NTRACE_POS*
- *NOVAS_SET_DUMP_NET_REAL_NAME*

- *NOVAS_SET_PROTECTED_MODULE_TOBE_LIBCELL*
- *NOVAS_SHOW_FULL_RANGE_OF_SIG*
- *NOVAS_SIM_NO_MSG_INOUT*
- *NOVAS_SKIP_FORM_BACKSLASH_BUS*
- *NOVAS_SPECIFIC_LIBRARY_IO_HANDLE*
- *NOVAS_SWAP_FILE_INIT_NUM*
- *NOVAS_SWAP_FILE_SIZE*
- *NOVAS_SWAP_MEM_SIZE*
- *NOVAS_SWAP_ROOT_DIR*
- *NOVAS_SWITCH_LOCATION_DEFAULT_WITH_VCS*
- *NOVAS_SYS_VM_PAGE_SIZE*
- *NOVAS_UNSET_FASTEN_MODE*
- *NOVAS_USE_VERDI_LICENSE*
- *NOVAS_VCS_HOME*
- *NOVAS_VERICOM_RETURN_SUCCESS*
- *NOVAS_VERILOG_SYNTAX_MAX_ERROR*
- *NOVAS_VHDLPATH*
- *NOVAS_WAVE_CASE_INSENSITIVE*
- *NOVAS_WAVE_LIST_ALL_UNRECOG_RESTORED_SIG*
- *NOVAS_XDND_WITHOUT_SHIFT*
- *NOVAS_XFONT_PATH*
- *SPS_LICENSE_FILE*
- *SNPSLMD_LICENSE_FILE*
- *VC_DEBUG_IDLE_LICENSE_CHECKBACK*
- *WORKTIME_INTERVAL*

LM_LICENSE_FILE

Description

Check the FLEXlm documentation and see the *NOVAS_LICENSE_QUEUE* description for details.

NON_SPLIT_BUS

Description

Set this environment variable to turn the **Hide Extraneous Bus** toggle menu command in a flattened schematic window or an ECO window *off* by default. Otherwise, the **Hide Extraneous Bus** toggle menu command in a flattened schematic window or an ECO window is turned *on* by default.

NOTE: The **Hide Extraneous Bus** toggle menu command is available under the **Tools -> Options** menu in *nSchema*.

Syntax

```
NON_SPLIT_BUS
```

Example

```
setenv NON_SPLIT_BUS
```

NOVAS_ASSERT_WITHOUT_NOT_CHECKED

Description

It does not show the **Not Checked** column in the *Statistics Pane*.

Syntax

```
NOVAS_ASSERT_WITHOUT_NOT_CHECKED
```

Example

```
setenv NOVAS_ASSERT_WITHOUT_NOT_CHECKED
```

NOVAS_AUTO_SOURCE

Description

Set this environment variable to source Tcl command file(s) when the Verdi platform is initiated. This environment variable is set to the directory path for the Tcl or a file. If the directory path is given, all files in the directory are sourced. You can set multiple paths and the paths are separated by a space. You can use double quotes to enclose the paths.

Syntax

```
NOVAS_AUTO_SOURCE = path [path]
```

Examples

```
setenv NOVAS_AUTO_SOURCE *.tcl
setenv NOVAS_AUTO_SOURCE mytcl.tcl file.tcl
```

NOTE: The `TclAutoSource` key is in the `novas.rc` resource file in the **General** section.
For example:
`TclAutoSource =../Command/test.command`

NOVAS_CHECK_ESA_BA_SCOPE

Description

Specifies whether to check for the scope mismatch of Essential Signal Analysis (ESA) and Working Scope Behavior Analysis (WSBA). This environment variable is used by Siloti and by `fsdbexpand` to decide whether to abort or continue in the event of a mismatch. When set to *0*, the scope mismatch check is not performed. When set to *1*, the check is performed and a question form is displayed asking whether to abort or continue, except for `fsdbexpand` which continues in the event of a mismatch. When set to *2*, `fsdbexpand` aborts in the event of a mismatch. The default value of this variable is *1*.

Syntax

```
NOVAS_CHECK_ESA_BA_SCOPE = 0|1|2
```

Example

```
setenv NOVAS_CHECK_ESA_BA_SCOPE 1
```

NOVAS_CHECK_MDA_INDEX

Description

Set this environment variable to correct the MDA array index to display the active annotation correctly in *nTrace*.

Syntax

```
NOVAS_CHECK_MDA_INDEX = 1
```

Example

```
setenv NOVAS_CHECK_MDA_INDEX 1
```

NOVAS_CMD_ERROR_EXIT

Description

Set the value to >0 to force the process to run when a Tcl error occurs.

Syntax

```
NOVAS_CMD_ERROR_EXIT = 0|1
```

Example

```
setenv NOVAS_CMD_ERROR_EXIT 1
```

NOVAS_CMD_TEST

Description

Set this environment variable to display a form that allows you to enter Tcl commands and that returns the result.

Syntax

```
NOVAS_CMD_TEST = 0|1
```

Example

```
setenv NOVAS_CMD_TEST 1
```

NOVAS_CMP_BATCH_MODE_DEFAULT_RULE_FILE

Description

Specifies the default compared rule file in the batch mode.

Syntax

```
NOVAS_CMP_BATCH_MODE_DEFAULT_RULE_FILE = fileName
```

Example

```
setenv NOVAS_CMP_BATCH_MODE_DEFAULT_RULE_FILE/tmp/rule.ncr
```

NOVAS_COMB_UDP_AS_SIM

Description

Specify *1* to enable Behavior Analysis to use the HDL behavior to evaluate the cells that are defined with UDP. You can also specify *1* to enable the cell to instantiate UDP when the **Symbol Library Model First** option on the **Behavior Analysis -> Symbol Library** page of the *Preferences* form (invoked with the **Tools -> Preferences** command). The default value of this variable is *0*.

Syntax

```
NOVAS_COMB_UDP_AS_SIM = 0|1
```

Example

```
setenv NOVAS_COMB_UDP_AS_SIM 1
```

NOVAS_DBIN

Description

Specifies the VHDL precompiled libraries path for using the `-topdown` option.

Syntax

```
NOVAS_DBIN = directory
```

Example

```
setenv NOVAS_DBIN /home/user/vhdl/lib
```

NOVAS_DEB_AUTO_SESSION

Description

Set this environment variable to *1* or *on* to automatically save the session file before the Verdi platform exits.

Syntax

```
NOVAS_DEB_AUTO_SESSION = 0|1
```

Example

```
setenv NOVAS_DEB_AUTO_SESSION 1
```

NOVAS_DENALI

Description

Set this environment variable to enable Denali Pureview and invoke the **New Pureview** function from the Verdi platform.

Syntax

```
NOVAS_DENALI = denali_path
```

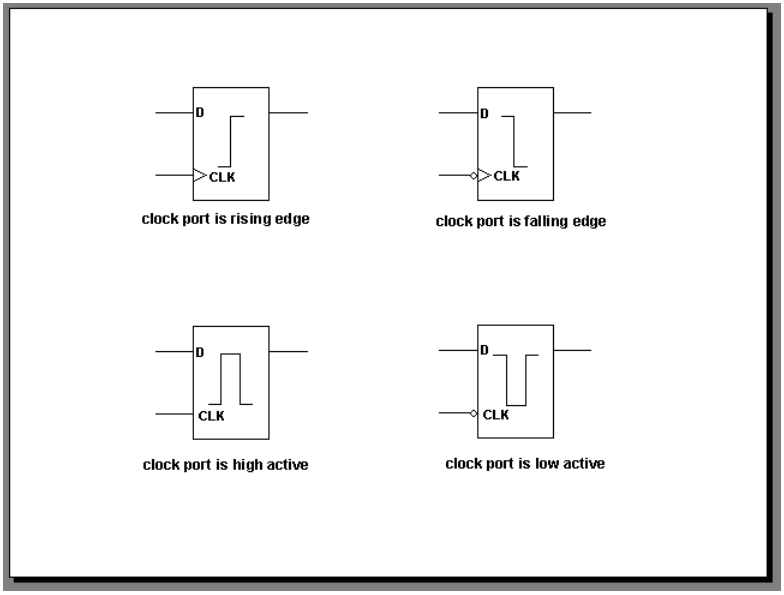
Example

```
setenv NOVAS_DENALI /usr/local/denali
```

NOVAS_DETAILED_REGISTER_SYMBOL

Description

Set this environment variable to show the clock signal shape on flip-flops and latch symbols based on the clock port state as shown in the following figure:



Syntax

```
NOVAS_DETAILED_REGISTER_SYMBOL = 1
```

Example

```
setenv NOVAS_DETAILED_REGISTER_SYMBOL 1
```

NOVAS_DISABLE_SWAP

Description

Disables the swap system. It is enabled by default.

Syntax

```
NOVAS_DISABLE_SWAP = 1
```

Example

```
setenv NOVAS_DISABLE_SWAP 1
```

NOVAS_DUMP_DOC_DETAIL

Description

Dumps contents of the specified file.

Syntax

```
NOVAS_DUMP_DOC_DETAIL = <fileName>
```

Example

```
setenv NOVAS_DUMP_DOC_DETAIL system.v
```

NOVAS_DUMP_FSM_TOPOLOGY_PNR

Description

Dumps the bounding box information after rearranging FSM Place and Route.

Syntax

```
NOVAS_DUMP_FSM_TOPOLOGY_PNR = 1
```

Example

```
setenv NOVAS_DUMP_FSM_TOPOLOGY_PNR
```

NOVAS_ECO_ASTRO_SCRIPT

Description

Enables **Astro Format** in the *Save ECO Script* form opened with the **File** -> **Save ECO Script** command and in the **Preference** -> **ECO** -> **Script File** page.

Syntax

```
NOVAS_ECO_ASTRO_SCRIPT = 0|1
```

Example

```
setenv NOVAS_ECO_ASTRO_SCRIPT 1
```

NOVAS_ENABLE_SIGNED_KEYWORD

Description

Set this environment variable to *off* or *0* to label `signed` as an identifier, otherwise it remains a keyword. The `signed` keyword is supported by default.

Syntax

```
NOVAS_ENABLE_SIGNED_KEYWORD = 0|1
```

Example

```
setenv NOVAS_ENABLE_SIGNED_KEYWORD 1
```

NOVAS_ENV_YELLOW_LIGHT_DURATION

Description

It is used by the FSDB writer test program.

Syntax

```
NOVAS_ENV_YELLOW_LIGHT_DURATION = number
```

Example

```
setenv NOVAS_ENV_YELLOW_LIGHT_DURATION 20
```

NOVAS_ETC_DIR

Description
Specifies the location of the *etc* directory.

Syntax
```
NOVAS_ETC_DIR = path
```

Example
```
setenv NOVAS_ETC_DIR /home/novas/etc
```

NOVAS_EXECUTABLE_ARCH

Description
Set this environment variable to force `.wrapper` to use the 32-bit or 64-bit executable on all platforms. For example, when the environment variable is set to 32, only the 32-bit executable is invoked. When the 32-bit executable is not installed, a warning stating that the 32-bit executable cannot be found is reported.

Syntax
```
NOVAS_EXECUTABLE_ARCH = 32|64
```

Examples
```
setenv NOVAS_EXECUTABLE_ARCH 32
setenv NOVAS_EXECUTABLE_ARCH 64
```

NOTE: The environment variable is applied to both the Verdi platform and other executables.

NOVAS_EXIT_STATUS_CODE

Description
Specifies the status code passed to the exit function call in `sysExit`.

Syntax
```
NOVAS_EXIT_STATUS_CODE = number
```

Example

```
setenv NOVAS_EXIT_STATUS_CODE 1
```

NOVAS_FLAT_ALL_ARRAYS

Description

Set this environment variable to *1* to use Behavior Analysis to flatten MDA signals into buses. The default value of this variable is *0*.

For example, when a signal is declared as follows and `FLAT_ALL_ARRAYS` is set to *1*:

```
reg [1:0] sig [2:0]
```

Behavior Analysis models the `sig` signal as the following three nets:

```
sig[2][1:0]
sig[1][1:0]
sig[0][1:0]
```

However, when `FLAT_ALL_ARRAYS` is set to *0*, Behavior Analysis models the `sig` signal as one net.

NOTE: For the *Temporal Flow View* behavior, it is recommended to set this environment variable to *0*. For the *Siloti* behavior, it is recommended to set this environment variable to *1*.

Temporal Flow View Behaviors

When FLAT_ALL_ARRAYS is set to *1*, memory tracing cannot be used.

1. Memory signals cannot be recognized and is flattened in the *Flow View* behavior.
2. In the *nTrace* window, the following right-click commands do not work:
   - **Debug Memory -> Dump Memory Waveform to FSDB**
   - **Debug Memory -> Trace Memory Write**
   - **Debug Memory -> Memory Definition Table**
3. In the *nMemory* window, the **Get Memory Variable -> Calculated by Verdi** command cannot be used.

Siloti Behaviors

When FLAT_ALL_ARRAYS is set to *0*, Siloti cannot compute an MDA signal. This variable allows Siloti to compute an MDA signal if this signal is not dumped or flattened.

MDA Signals Dumped	FLAT_ALL_ARRAYS	Computable by Siloti
YES	N/A	YES
NO	1	YES
NO	0	NO

Syntax

```
FLAT_ALL_ARRAYS = 0|1
```

Example

```
setenv FLAT_ALL_ARRAYS 1
```

NOVAS_FND_HIER_MORE_NUM

Description

Specifies a value to change the default number of instances to be displayed in the *Find Instances* frame opened in the *Gate Level Debug* mode. If this environment variable is not specified, the default number of instances is *5000*.

Syntax

```
NOVAS_FND_HIER_MORE_NUM = number
```

Example

```
setenv NOVAS_FND_HIER_MORE_NUM 1000
```

In the example, the *Find Instances* frame displays 1000 instances for the initial view. To view additional instances after the first 1000 results, double-click the **More** line for more instance results.

NOVAS_FONT_PATH

Description

Specifies the true type font path. This string cannot contain multiple paths.

Syntax

```
NOVAS_FONT_PATH = path
```

Example

```
setenv NOVAS_FONT_PATH /dar/tarfile/third/TTFont
```

NOVAS_FSM_DUMP_GROUP_CV

Description

Dumps the cell view information for a group.

Syntax

```
NOVAS_FSM_DUMP_GROUP_CV
```

Example

```
setenv NOVAS_FSM_DUMP_GROUP_CV
```

NOVAS_FSM_DUMP_TOPOLOGY_MODE_FLAG

Description

Dumps the FSM topology place and route data.

Syntax

```
NOVAS_FSM_DUMP_TOPOLOGY_MODE_FLAG
```

Example

```
setenv NOVAS_FSM_DUMP_TOPOLOGY_MODE_FLAG
```

NOVAS_GUICONF

Description

Specifies the path and file name of the `novas.conf` configuration file to use at startup.

For more details about the configuration file, see the novas.conf description in the *Appendix B* chapter.

Syntax

```
NOVAS_GUICONF = <file.conf>
```

Example

```
setenv NOVAS_GUICONF /da100/integ/my.conf
```

NOVAS_HDBSM_DETAILRTL

Description

Set this environment variable to *on* to enable detailed RTL settings.

Syntax

```
NOVAS_HDBSM_DETAILRTL = on
```

Example

```
setenv NOVAS_HDBSM_DETAILRTL on
```

NOVAS_HLPPATH

Description

Specifies the Help file directory and supports multiple paths.

Syntax

```
NOVAS_HLPPATH = path
```

Examples

UNIX

```
setenv NOVAS_HLPPATH /dar/vital/doc1:/dar/vital/doc2
```

Win32 (`env.ini`)

```
setenv NOVAS_HLPPATH=..\doc1;e:\win32\doc2
```

VERDI_HOME

Description

Specifies the path to the Verdi installation. This environment variable is used by the FSDB dumper.

Syntax

```
VERDI_HOME = directory
```

Example

```
setenv VERDI_HOME /tools/Verdi/201207
```

NOTE: Prior to `VERDI_HOME`, `NOVAS_HOME` was used to indicate the Verdi installation path. From the Verdi 1606 release onwards, `VERDI_HOME` is used to replace `NOVAS_HOME`. However, `NOVAS_HOME` still works unless there is a content conflict with `VERDI_HOME`.

NOVAS_HT_TRACE

Description

Set this environment variable to *on* to trace all drivers.

Syntax

```
NOVAS_HT_TRACE = on
```

Example

```
setenv NOVAS_HT_TRACE on
```

NOVAS_ICC_CONN_FLOAT_PORT

Description

Logs ICC Tcl commands, namely `create_net` and `connect_net`, when connecting the floating port to the instance port in the ICC place and route stage.

Syntax

```
NOVAS_ICC_CONN_FLOAT_PORT = 1
```

Example

```
setenv NOVAS_ICC_CONN_FLOAT_PORT 1
```

NOVAS_ICC_FORMAT

Description

Logs the full path for the ICC Tcl commands without the top scope.

Syntax

```
NOVAS_ICC_FORMAT
```

Example

```
setenv NOVAS_ICC_FORMAT
```

NOVAS_ICC_FREEZE_MODE

Description

Logs the extra ICC Tcl commands for the *Freeze Silicon* mode when generating the ICC ECO script (via **File -> ECO -> Save ECO Script**).

Syntax

```
NOVAS_ICC_FREEZE_MODE
```

Example

```
setenv NOVAS_ICC_FREEZE_MODE
```

NOVAS_ICC_FULL_PATH

Description

Logs ICC Tcl commands with their full path. If this environment variable is not set, the full path is not logged.

Syntax

```
NOVAS_ICC_FULL_PATH
```

Example

```
setenv NOVAS_ICC_FULL_PATH
```

NOVAS_ICC_LIB_PATH

Description

Logs the symbol library name for the ICC Tcl command, "`create_cell`".

NOTE: The format of the symbol library name is "`create_cell instName symbol_lib_name/cell_name`". For example: `create_cell i_ALUB/eco_i6 lsi10k_u/AN3P`.

Syntax

```
NOVAS_ICC_LIB_PATH
```

Example

```
setenv NOVAS_ICC_LIB_PATH
```

NOVAS_IDLE_LICENSE_CHECKBACK

Description

Specifies the idle time (minutes) after which the license is checked back. When set to a negative value, the warning message associated with setting this environment variable is suppressed. If the environment variable is not set, the license is not released. The minimum idle time is 5 minutes and the maximum idle time is 255 minutes. If no licenses are available when an attempt is made to

check out an idle license from the pool again, there are five grace commands available so that signals or sessions can be saved.

NOTE: A shared license key (that is, `VC-DEBUG-RUNTIME` used in the *Verification Compiler Platform* mode) is used when a Verdi debug session involves Hardware/Software debug. The *license idle-check-back* feature is not supported for Hardware/Software debug.

NOTE: If the license is released after the specified idle time, GUI operations are delayed because of network communication between license nodes.

Syntax

```
NOVAS_IDLE_LICENSE_CHECKBACK = [-]minutes
```

Example

```
setenv NOVAS_IDLE_LICENSE_CHECKBACK -10
```

(is same as the following settings:

```
setenv NOVAS_IDLE_LICENSE_CHECKBACK 10
```

and

```
setenv NOVAS_IDLE_LICENSE_CHECKBACK_SILENCE <any value>
```

)

NOTE: Do not use `NOVAS_IDLE_LICENSE_CHECKBACK` with `NOVAS_LICENSE_QUEUE` as the `NOVAS_LICENSE_QUEUE` function does not work correctly.

NOVAS_IDLE_LICENSE_CHECKBACK_SILENCE

Description

Disables the warning message from `NOVAS_IDLE_LICENSE_CHECKBACK`. It suppresses the license expiration warning message.

Syntax

```
NOVAS_IDLE_LICENSE_CHECKBACK_SILENCE = 1
```

Example

```
setenv NOVAS_IDLE_LICENSE_CHECKBACK_SILENCE 1
```

NOTE: Do not use `NOVAS_IDLE_LICENSE_CHECKBACK_SILENCE` with `NOVAS_LICENSE_QUEUE` as the `NOVAS_LICENSE_QUEUE` function does not work correctly.

NOVAS_INTERACTIVE_MAP_DIR

Description

Specifies the mapping directories with the following rules:

- Separate the directory and its mapping directory by a colon (:).
- Enclose multiple directories with double quotes.

NOTE: If the specified source file cannot be found in the *Interactive Simulation Debug* mode, Verdi finds the source file in the mapping directory.

Syntax

```
NOVAS_INTERACTIVE_MAP_DIR "[dir_1]:[dir_map_1]
[dir_2]:[dir_map_2] … [dir_N]:[dir_map_N]"
```

Example

```
setenv NOVAS_INTERACTIVE_MAP_DIR "/top/a1:/top/b1 /top/a2:/top/
b2"
```

NOVAS_LANG_VER

Description

Specifies the language version for the compilers.

Syntax

```
NOVAS_LANG_VER = sv|2001|2000|87|93
where, sv   = SystemVerilog
       2001 = Verilog 2001
       2000 = VHDL 2000
       87   = VHDL 87
       93   = VHDL 93
```

Example

```
setenv NOVAS_LANG_VER 2001
```

NOVAS_LIBPATHS

Description

Specifies where the symbol libraries are located. The value string may contain multiple path strings separated by either a space or a colon (:).

Syntax

```
NOVAS_LIBPATHS = libraryPath [libraryPath...]
```

Example

```
setenv NOVAS_LIBPATHS $VERDI_HOME/share/symlib/32
```

NOVAS_LIBS

Description

Specifies the symbol library list. The value string may contain multiple path strings separated by either a space or a colon (:). The file extension is added to specify the file type of the symbol library list. For example, the *Synopsys Library* (`*.db`) *Liberty File* (`*.lib`), and *gzipped Liberty* (*.lib.gz) file types.

Syntax

```
NOVAS_LIBS = libraryName [libraryName...]
```

Example

```
setenv NOVAS_LIBS “lsi10k_u default_u.lib”
setenv NOVAS_LIBS xx.lib.gz
```

NOVAS_LIC_EXP_CTRL

Description

Set this environment variable to control the license expiration warning message. When set, the value represents the number of days for which the license expiration warning message is displayed. The default value of this variable is *30*.

Setting the value to 0 disables the warning message. The maximum value is 127 (any values greater than 127 is set to 127).

Syntax

```
NOVAS_LIC_EXP_CTRL = number
```

Example

```
setenv NOVAS_LIC_EXP_CTRL 10
```

NOVAS_LICENSE_FILE

Starting in the Verdi-2014.03 release, this environment variable is replaced by *SNPSLMD_LICENSE_FILE*.

NOVAS_LICENSE_QUEUE

Description

Set this environment variable to wait for a license to be released when a license is not available.

Syntax

```
NOVAS_LICENSE_QUEUE = 1
```

Example

```
setenv NOVAS_LICENSE_QUEUE 1
```

NOTE: Do not use `NOVAS_IDLE_LICENSE_CHECKBACK` with `NOVAS_LICENSE_QUEUE` as the `NOVAS_LICENSE_QUEUE` function does not work correctly.

NOVAS_MAGMA_FLAT_DB

Description

Supports the new Magma format for the **Save ECO Script** command of *nECO*.

When the environment variable is set, the saved Magma script is as follows:

```
data detach $m/net:i_cpu/i_ALUB/n744 net_pin $m/i_cpu/i_ALUB/
U254/A
data attach $m/net:i_cpu/i_ALUB/aaa net_pin $m/i_cpu/i_ALUB/U254/
A
```

When the environment variable is not set, the saved Magma script is as follows:

```
data detach $m/i_cpu/i_ALUB/net:n744 net_pin $m/i_cpu/i_ALUB/
U254/A
data attach $m/i_cpu/i_ALUB/net:aaa net_pin $m/i_cpu/i_ALUB/U254/
A
```

Syntax

```
NOVAS_MAGMA_FLAT_DB = 0|1
```

Example

```
setenv NOVAS_MAGMA_FLAT_DB 1
```

NOVAS_MANAGE_RC_DIR

Description

Specifies the path to the `manage.rc` resource file. The Verdi platform only recognizes and loads the file named `manage.rc`. For details on the `manage.rc` file, see *Appendix B: Customizing Verdi*.

Syntax

```
NOVAS_MANAGE_RC_DIR = path
```

Example

```
setenv NOVAS_MANAGE_RC_DIR /tools/proj/pref
```

NOVAS_MAX_ITEMS_IN_FINDSIGNAL_FORM

Description

Specifies the maximum number of signals listed in the *Find Signals* form. This number takes effect when it is larger than 5000.

Syntax

```
NOVAS_MAX_ITEMS_IN_FINDSIGNAL_FORM = number
```

Example

```
setenv NOVAS_MAX_ITEMS_IN_FINDSIGNAL_FORM 10000
```

NOVAS_MAX_LOGFILE_SIZE

Description

Limits the file size of the Verdi log file, `turbo.log`, within the specified number of megabytes (MB).

Syntax

```
NOVAS_MAX_LOGFILE_SIZE = number
```

Examples

```
setenv NOVAS_MAX_LOGFILE_SIZE 1
```

The example limits the `turbo.log` file size to 1 megabyte. It also stops logging when the size is exceeded.

NOVAS_MDT_LIBPATHS

Description

Specifies where the predefined memory definition files are located. The current working directory is automatically included in the search path. The memory definition file(s) are specified by `NOVAS_MDT_LIBS`.

The value string may contain multiple path strings separated by either a space or a colon (`:`).

Syntax

```
NOVAS_MDT_LIBPATHS = path:[path...]
```

Examples

```
setenv NOVAS_MDT_LIBPATHS $VERDI_HOME/share/mdtlib
setenv NOVAS_MDT_LIBPATHS $VERDI_HOME/share/mdtlib:<path>/
project/memory
```

NOVAS_MDT_LIBS

Description

Specifies one or more memory definition files for automatic loading in the Memory Definition Table (MDT) list. The memory definition files must have an `.mdt` extension. This extension is assumed by the environment variable and does not need to be explicitly included. The path(s) to the file(s) are specified by `NOVAS_MDT_LIBPATHS`.

The value string may contain multiple path strings separated by either a space or a colon (`:`).

Syntax

```
NOVAS_MDT_LIBS = name:[name...]
```

Examples

```
setenv NOVAS_MDT_LIBS ART_RF_SP
```

```
setenv NOVAS_MDT_LIBS ART_RF_SP:mymemory:VIR_SRAMPWBV
```

NOVAS_NO_AUTO_ICONIZE

Description

Set this environment variable to turn *off* auto-iconizing of additional dialog windows when the parent window is iconized.

Syntax

```
NOVAS_NO_AUTO_ICONIZE = 1
```

Example

```
setenv NOVAS_NO_AUTO_ICONIZE 1
```

NOVAS_NO_BUSY_DIALOG

Description

Set this environment variable to replace the busy dialog with a busy cursor. To restore the busy dialog, unset this variable.

Syntax

```
NOVAS_NO_BUSY_DIALOG = 1
```

Examples

```
setenv NOVAS_NO_BUSY_DIALOG 1
unsetenv NOVAS_NO_BUSY_DIALOG
```

NOVAS_NO_ENTER_FOCUS

Description

Set this environment variable to enable clicking the mouse button to change the focus of the selected window to bind the bind keys with this window to work. If the mouse button is not clicked, the bind keys only work if the cursor is moved on the bound window.

Syntax

```
NOVAS_NO_ENTER_FOCUS = 1
```

Examples

```
setenv NOVAS_NO_ENTER_FOCUS
unsetenv NOVAS_NO_ENTER_FOCUS
```

NOVAS_NOT_FOUND_SAVE

Description

Saves the signals, which cannot be found in the FSDB file, from the loaded signal list to the file specified by this variable. When this environment variable is set, the form listing the missing signals is not opened and *nWave* automatically saves the contents to the file specified by `NOVAS_NOT_FOUND_SAVE`. When the missing signals are found, the file is also updated.

Syntax

```
NOVAS_NOT_FOUND_SAVE = string
```

where, `string` is the specified file name including the path.

Example

```
setenv NOVAS_NOT_FOUND_SAVE /usr/not_found
```

NOVAS_PCELL_EXTRACTION

Description

Set this environment variable to extract the Isolation commands from the RTL code.

Syntax

```
NOVAS_PCELL_EXTRACTION
```

Examples

```
setenv NOVAS_PCELL_EXTRACTION
```

NOVAS_PURIFY_POOL_ARENA

Description

Switches the memory allocation scheme to use the default group of functions, such as `libc malloc()` and `free()`.

Syntax

```
NOVAS_PURIFY_POOL_ARENA = 1
```

Examples

```
setenv NOVAS_PURIFY_POOL_ARENA 1
```

NOVAS_QUESTA_66_GENFOR

Description

Conforms the unnamed generate block naming rules for VHDL with versions earlier than ModelSim 6.6.

Syntax

```
NOVAS_QUESTA_66_GENFOR = 1
```

Example

```
setenv NOVAS_QUESTA_66_GENFOR 1
```

NOVAS_RC

Description

Specifies the path and file name of the `novas.rc` resource file to use at startup. Do not set this path to the `manage.rc` resource file.

For more details about the resource file, see the novas.rc description in the *Appendix B* chapter.

Syntax

```
NOVAS_RC = file.rc
```

Example

```
setenv NOVAS_RC /da100/integ/my.rc
```

NOVAS_SCHEMA_LOOP_LEVEL

Description

Specifies the maximum number of the loop expansion level in *nSchema*. The default value is *128*.

Syntax

```
NOVAS_SCHEMA_LOOP_LEVEL = number
```

Example

```
setenv NOVAS_SCHEMA_LOOP_LEVEL 200
```

NOVAS_SDC_LIMIT

Description

Sets the boundary size that is needed to force the Synopsys Design Constraint (SDC) false path file to use a swap file.

Example

```
setenv NOVAS_SDC_LIMIT 1
```

NOVAS_SESSION_KEEP_NTRACE_POS

Description

Set this environment variable when the save session *nTrace* window is outside the current visible desktop region. If set, *nTrace* attempts to keep its location same as the save session when the session is restored.

Syntax

```
NOVAS_SESSION_KEEP_NTRACE_POS = 1
```

Example

```
setenv NOVAS_SESSION_KEEP_NTRACE_POS 1
```

NOVAS_SET_DUMP_NET_REAL_NAME

Description
Dumps the real net name into the log file.

Syntax
```
NOVAS_SET_DUMP_NET_REAL_NAME
```

Example
```
setenv NOVAS_SET_DUMP_NET_REAL_NAME
```

NOVAS_SET_PROTECTED_MODULE_TOBE_LIBCELL

Description
Set this environment variable to ignore entire source code within a module containing protected code. The module contents are treated as a library cell (only the module header and I/O ports are kept).

Syntax
```
NOVAS_SET_PROTECTED_MODULE_TOBE_LIBCELL = 1
```

Example
```
setenv NOVAS_SET_PROTECTED_MODULE_TOBE_LIBCELL 1
```

NOVAS_SHOW_FULL_RANGE_OF_SIG

Description
Shows "OOR" as active annotation in *nTrace* for the MDA signal inside a `for` loop.

Syntax
```
NOVAS_SHOW_FULL_RANGE_OF_SIG = 1
```

Example
```
setenv NOVAS_SHOW_FULL_RANGE_OF_SIG 1
```

NOVAS_SIM_NO_MSG_INOUT

Description

Disables the simulator message display and command input in the *Message* frame.

Syntax

```
NOVAS_SIM_NO_MSG_INOUT = 1
```

Examples

```
setenv NOVAS_SIM_NO_MSG_INOUT 1
unsetenv NOVAS_SIM_NO_MSG_INOUT
```

NOVAS_SKIP_FORM_BACKSLASH_BUS

Description

Set this environment variable to prevent *nWave* from forming a bus for signals that contain a backslash (\) in their name.

Syntax

```
NOVAS_SKIP_FORM_BACKSLASH_BUS
```

Example

```
setenv NOVAS_SKIP_FORM_BACKSLASH_BUS
```

```
  /top/\a[0][0]
  /top/\a[0][1]
```

The `/top/\a[0][1:0]` bus is not formed.

NOVAS_SPECIFIC_LIBRARY_IO_HANDLE

Description

Specifies whether the specific `novas.rc` resource file handling a scheme needs to be used. The scheme includes ignoring the `rc` Header and Library section string when the `novas.rc` resource file only contains the Library section, writing `DEFINE` or '=' key in the Library section as `DEFINE`, and ignoring the "`--`" comment line.

Syntax

```
NOVAS_SPECIFIC_LIBRARY_IO_HANDLE = 1
```

Examples

```
setenv NOVAS_SPECIFIC_LIBRARY_IO_HANDLE 1
```

NOVAS_SWAP_FILE_INIT_NUM

Description

Specifies the maximum number of swapping files to be created. The default value of the variable is *256*.

Syntax

```
NOVAS_SWAP_FILE_INIT_NUM = number
```

Example

```
setenv NOVAS_SWAP_FILE_INIT_NUM 8
```

NOVAS_SWAP_FILE_SIZE

Description

Specifies the single swap file size in megabytes (MB). The default value of the variable is *200*.

Syntax

```
NOVAS_SWAP_FILE_SIZE = number
```

Example

```
setenv NOVAS_SWAP_FILE_SIZE 2000
```

NOVAS_SWAP_MEM_SIZE

Description

Specifies the available memory size for the swapping scheme. When the specified memory size is reached, the data is swapped to the swapping file. The default unit for the specified size is megabytes (MB). If the K or k is added as the post-fix, then the unit is bytes. The minimum size is 32MB. If the specified size

is smaller than this, then 32MB is used. The default value of this variable is *64MB*.

Syntax

```
NOVAS_SWAP_MEM_SIZE = <number>|<number>k|<number>K
```

Example

```
setenv NOVAS_SWAP_MEM_SIZE 256
```

NOVAS_SWAP_ROOT_DIR

Description

Specifies the directory in which you want to create the swap file (must exist and be writable). The default directory is the default log directory (for example, *verdiLog*, *vericomLog*, and *novasLog*) for the executable in progress.

Syntax

```
NOVAS_SWAP_ROOT_DIR = directory
```

Example

```
setenv NOVAS_SWAP_ROOT_DIR /tmp/Log
```

NOVAS_SWITCH_LOCATION_DEFAULT_WITH_VCS

Description

Specifies *1* to set the location of an inserted signal/cell to be same as the VCS location. In UPF 2.0 files, the `-location <keyword>` option specifies the location of the inserted signal/cell. When this environment variable is specified as *0*, the inserted signals/cells may not be in the same location as in the VCS location. When the `-location` option is not specified or the `-location automatic` option is specified, the inserted location is determined by default, regardless of different default locations for different tools.

Syntax

```
NOVAS_SWITCH_LOCATION_DEFAULT_WITH_VCS = 0|1
```

Examples

```
setenv NOVAS_SWITCH_LOCATION_DEFAULT_WITH_VCS 1
setenv NOVAS_SWITCH_LOCATION_DEFAULT_WITH_VCS 0
```

NOVAS_SYS_VM_PAGE_SIZE

Description

Specifies the default VM region size. Its default value is *8K*, and its default value for Linux is *64K*. A larger page size can improve performance, but it also consumes more memory. Page size can only be set in increments of 8K (if not, it is rounded off).

Syntax

```
NOVAS_SYS_VM_PAGE_SIZE = <page_size>
```

Example

```
setenv NOVAS_SWAP_FILE_INIT_NUM 8
```

NOVAS_UNSET_FASTEN_MODE

Description

Set this environment variable when one of the following situation occurs:

- Searching is slow. Set this environment variable to *1* to speed up the searching.
- Displaying all signals in a single page of the *nTrace*'s *Signal List* frame is preferred. The setting for the **Maximum Signals to Display in One Page** option is not available.
- In the **Source Code -> Design Tree** page of the *Preferences* form, the default of the **Display Library Cells** option is changed from *off* to *on*.

NOTE: When this environment variable is set, performance for source code display is downgraded.

Syntax

```
NOVAS_UNSET_FASTEN_MODE = 0|1
```

Example

```
setenv NOVAS_UNSET_FASTEN_MODE 1
```

NOVAS_USE_VERDI_LICENSE

Description
Set this environment variable to use the Verdi license file.

Syntax
```
NOVAS_USE_VERDI_LICNESE
```

Example
```
setenv NOVAS_USE_VERDI_LICNESE
```

NOVAS_VCS_HOME

Description
Specifies the path to the VCS installation for the Vera parser.

Syntax
```
NOVAS_VCS_HOME = directory
```

Example
```
setenv NOVAS_VCS_HOME /simulator/Synopsys/VCS7.2FCS/vcs7.2/linux
```

NOVAS_VERICOM_RETURN_SUCCESS

Description
Set this environment variable to always return success when the *vericom* utility is executed.

Syntax
```
NOVAS_VERICOM_RETURN_SUCCESS
```

Example
```
setenv NOVAS_VERICOM_RETURN_SUCCESS
```

NOVAS_VERILOG_SYNTAX_MAX_ERROR

Description

Specifies the maximum number of syntax errors before parsing stops. When the number is less than 0, the *vericom* utility continues parsing when syntax errors are found.

Syntax

```
NOVAS_VERILOG_SYNTAX_MAX_ERROR number
```

Example

```
setenv NOVAS_VERILOG_SYNTAX_MAX_ERROR 20
```

NOVAS_VHDLPATH

Description

Specifies the VHDL source file path for using the `-topdown` option.

Syntax

```
NOVAS_VHDLPATH = <directory>
```

Example

```
setenv NOVAS_VHDLPATH /home/user/vhdl/src
```

NOVAS_WAVE_CASE_INSENSITIVE

Description

Provides the matched signal name and it is not case-sensitive.

Syntax

```
NOVAS_WAVE_CASE_INSENSITIVE
```

Example

```
setenv NOVAS_WAVE_CASE_INSENSITIVE
```

NOVAS_WAVE_LIST_ALL_UNRECOG_RESTORED_SIG

Description

Lists all unrecognized signals in the *List of Unrecognized Signal(s)* form when restoring the `signal.rc` file in the *nWave* window.

Syntax

```
NOVAS_WAVE_LIST_ALL_UNRECOG_RESTORED_SIG
```

Example

```
setenv NOVAS_WAVE_LIST_ALL_UNRECOG_RESTORED_SIG
```

NOVAS_XDND_WITHOUT_SHIFT

Description

Set this environment variable to use XDND as the default protocol for drag and drop.

Syntax

```
NOVAS_XDND_WITHOUT_SHIFT
```

Example

```
setenv NOVAS_XDND_WITHOUT_SHIFT
```

NOVAS_XFONT_PATH

Description

Set this environment variable for the X font path. You cannot specify multiple paths.

Syntax

```
NOVAS_XFONT_PATH = path
```

Example

```
setenv NOVAS_XFONT_PATH /rcc/tarfile/third/XFont
```

SPS_LICENSE_FILE

Starting in the Verdi-2014.03 release, this environment variable is replaced by *SNPSLMD_LICENSE_FILE*.

SNPSLMD_LICENSE_FILE

Description

Specifies the location of the license file. Starting in the Verdi-2014.03 release, the old SpringSoft license daemon is no longer available. Licensing is now supported with the Synopsys Common Licensing software. For details, see the *Setting Up the License Server* section of the *Installation and System Administration Guide*.

NOTE: If both `LM_LICENSE_FILE` and `SNPSLMD_LICENSE_FILE` are used, any Verdi keys that exist in `LM_LICENSE_FILE` is ignored and only keys that exist in `SNPSLMD_LICENSE_FILE` is recognized.

Syntax

```
SNPSLMD_LICENSE_FILE = license.dat
```

Example

```
setenv SNPSLMD_LICENSE_FILE /da100/integ/license.dat
```

VC_DEBUG_IDLE_LICENSE_CHECKBACK

Description

Specifies the idle time (in minutes) after which the license(s) are released. When set to a negative value, the warning message associated with setting this environment variable is suppressed. If the environment variable is not set, the license is not released. The minimum idle time is 5 (minutes) and the maximum idle time is 255 (minutes). If no licenses are available when an attempt is made

to check out an idle license from the pool again, there are five grace commands available so that signals or sessions are saved.

NOTE: If the license is released after the specified idle time, GUI operations may be delayed because of network communication between license nodes.

NOTE: `VC_DEBUG_IDLE_LICENSE_CHECKBACK` is valid only in the Verification Compiler Platform mode. `VC_HOME` and the `$VC_HOME/etc/.sanity` file are used to detect the Verification Compiler Platform mode.

Syntax

```
VC_DEBUG_IDLE_LICENSE_CHECKBACK = [-]minutes
```

Examples

```
setenv VC_DEBUG_IDLE_LICENSE_CHECKBACK -10
```

NOTE: It is not recommended to use `VC_DEBUG_IDLE_LICENSE_CHECKBACK` with `NOVAS_LICENSE_QUEUE` as `NOVAS_LICENSE_QUEUE` does not work correctly.

NOTE: If `VC_DEBUG_IDLE_LICENSE_CHECKBACK` and `NOVAS_IDLE_LICENSE_CHECKBACK` are specified simultaneously, the `VC_DEBUG_IDLE_LICENSE_CHECKBACK` setting has a higher priority.

NOTE: When a Verdi debug session involves Protocol Analyzer, HW/SW, or VC-Static tools, the tool and the Verdi platform share the same `VC_DEBUG_RUNTIME` license key. In addition, the idle license check back feature is not supported. Therefore, even when the Protocol Analyzer, HW/SW, or VC-Static tools are idle, the `VC_DEBUG_RUNTIME` license is not released (even if the Verdi platform is also idle).

WORKTIME_INTERVAL

Description

Set this environment variable to a value to reduce the CPU usage when Verdi is idle. This variable controls the internal timer interval.

NOTE: The default value is *60*. To future reduce the CPU usage, you can set it from 100 to 500.

Syntax

```
WORKTIME_INTERVAL = number
```

Example

```
setenv WORKTIME_INTERVAL 100
```

Siloti - Essential Signal Analysis

This section consists of the details of the following environment variables:

NOVAS_ESA_DE_INIT_MISMATCH

Description

Set this environment variable to include signals with an initial value mismatch to the Essential Signal Database (ESDB). Specify this environment variable during both the Essential Signal Analysis phase and the simulation phase.

The types of initial value mismatch signals include the following:

- Signals with an initial value set multiple times, for example:

```
initial begin
  a = 1'b1;
#5
  a = 1'b0;
end
```

- Wire type signals with a function call on the right-hand side, for example:

```
wire a;
a = foo(b, c); //a is dumped
```

- Signals with different initial values, for example:

```
reg a;
  wire b;
  always@(b)
    a = b; // initial value of a is x, but b is z;
           //a is dumped
```

Syntax

```
NOVAS_ESA_DE_INIT_MISMATCH = 1
```

Example

```
setenv NOVAS_ESA_DE_INIT_MISMATCH 1
```

NOVAS_ESA_DUMP_INITIAL

Description

Set this environment variable to include all signals that have an initial attribute in the Essential Signal Database (ESDB). When set to *2*, only signals with an initial value set multiple times are included in ESDB. Specify this environment variable during both the Essential Signal Analysis phase and the simulation phase.

Syntax

```
NOVAS_ESA_DUMP_INITIAL = 1|2
```

Example

```
setenv NOVAS_ESA_DUMP_INITIAL 1|2
```

NOVAS_ESA_DUMP_QN

Description

Set this environment variable before running a simulation with the Essential Signal Database (ESDB) to include the QN pin of registers in the FSDB file. When set to *0*, the QN pin of registers is not included in the FSDB file. The default value of this variable is *1*.

Syntax

```
NOVAS_ESA_DUMP_QN = 1
```

Example

```
setenv NOVAS_ESA_DUMP_QN 1
```

NOVAS_ESA_DUMP_REAL

Description

Set this environment variable before running a simulation with the Essential Signal Database (ESDB) to include real signals in the FSDB file.

Syntax

```
NOVAS_ESA_DUMP_REAL = 1
```

Example

```
setenv NOVAS_ESA_DUMP_REAL 1
```

NOVAS_FSDB_DUMP_ES_ABORT

Description

Set this environment variable to make `$fsdbDumpvars` exit the simulation when the Essential Signal Database (ESDB) passed to the dumping command is not found or is not readable. The default value for this variable is *0*.

Syntax

```
NOVAS_FSDB_DUMP_ES_ABORT = 1
```

Example

```
setenv NOVAS_FSDB_DUMP_ES_ABORT 1
```

NOVAS_FSDB_DUMP_ES_VTOP

Description

Changes the top instance name in the Essential Signal Database (ESDB) without modifying the file itself. This environment variable is valid only with `modelsim_fli dumpers` and only impacts the `$fsdbDumpvars` dumping command.

Syntax

```
NOVAS_FSDB_DUMP_ES_VTOP = my_vtop
```

Example

Consider the following ESDB file:

```
#ESD
...
$workscope = tb.dut.u0
<tb
<top
<u0
...
```

Set the environment variable as follows:

```
setenv NOVAS_FSDB_DUMP_ES_VTOP my_top
```

After setting the environment variable, the name of the `$workscope` top instance is changed from `tb.dut.u0` to `my_vtop.dut.u0`. Its result looks like as follows:

```
$workscope = my_vtop.dut.u0

my_vtop
  +-dut
    +u0
```

NOVAS_FSDB_ES_HIER_REPLACE

Description

Specifies `source_scope` to change to `tar_scope` during simulation. All sub-scopes and signals immediately under source_scope are changed but intermediate signals and sub-scopes of the source scope are ignored.

NOTE: *source_scope* is the scope that exists in the design to generate the Essential Signal Database (ESDB). *tar_scope* is the scope that exists in the design for simulation.

Syntax

```
NOVAS_FSDB_ES_HIER_REPLACE source_scope=tar_scope
```

Example

```
Original hierarchy in ESDB:
-------------------------
<a    // replaced
<aa   // *ignored
sigAA // *ignored
>     // *ignored
sigA  // *ignored
<b    // replaced
<bb
sigBB
>
```

```
sigB
>
>
------------------------
After (setenv NOVAS_FSDB_ES_HIER_REPLACE a.b=c.d.e)
------------------------
<c
<d
<e
<bb
sigBB
>
sigB
>
>
>
------------------------
```

Siloti - Data Expansion

This section consists of the details of the following environment variables:

NOVAS_DE_INCOMPLETE_CASE

Description

Used by the Data Expansion engine to evaluate the latches that are generated by an incomplete assigned case statement. Its default value is *0*.

Syntax

```
NOVAS_DE_INCOMPLETE_CASE = 1
```

Example

```
setenv NOVAS_DE_INCOMPLETE_CASE 1
```

NOVAS_FSDBEXP_VHDL_CASE

Description

Used by the *fsdbexpand* utility to specify whether to retain the case in the original VHDL design or convert everything to lowercase or uppercase while writing the output FSDB. Setting the value to 0 retains the original case. Setting the value to 1 converts everything to lowercase. Setting the value to 2 converts everything to uppercase. The default value of this variable is *1*.

Syntax

```
NOVAS_FSDBEXP_VHDL_CASE = 0|1|2
```

Example

```
setenv NOVAS_FSDBEXP_VHDL_CASE 1
```

Siloti - Correlation

This section consists of the details of the following environment variables:

CR_SKIP_HIER_FILE

Description

Specifies a file that lists a set of hierarchy names to skip in the gate design to correlate to RTL. If the * character is specified at the end of the hierarchy name, all modules with instance names that begin with the part before the * character of the hierarchy name is skipped.

Syntax

```
CR_SKIP_HIER_FILE = <filename>
```

Example

```
setenv CR_SKIP_HIER_FILE hier_name.txt
```

NOVAS_CR_VHDL_ORIG_CASE

Description

Set the value of this environment variable to 1 to save the name of VHDL design objects to CRDB with the original case in the design. Set the value of this environment variable to 0 to convert the name of VHDL design objects to lower case before saving it to CRDB. Its default value is *0*.

Syntax

```
CR_VHDL_ORIG_CASE = 0|1
```

Example

```
setenv CR_VHDL_ORIG_CASE 1
```

NOVAS_DISPLAY_INVERSE_CORRELATION

Description

Set this environment variable to show inversion with correlation annotation. Its default value is *0*.

Syntax

```
NOVAS_DISPLAY_INVERSE_CORRELATION = 0|1
```

Example

```
setenv NOVAS_DISPLAY_INVERSE_CORRELATION 1
```

VC Apps

This section consists of the details of the following environment variables:

VC_APPS_DM_NO_SESSION

Description

Set this environment variable to stop saving and restoring the session. The amount of the used memory is reduced. However, the HierMan cannot restore the session once the operation fails. The failed operation also affects output files.

Syntax

```
VC_APPS_DM_NO_SESSION = 1
```

Example

```
setenv VC_APPS_DM_NO_SESSION 1
```

VC_APPS_DM_NO_WRITE_NONEXISTENT_FILE

Description

Set this environment variable to stop writing files that do not exist. This environment variable only affects the import-from-library flow and the original file that is lost. If this environment variable is not set, the file that is the same as the original one is written to show differences between the original and new file.

Syntax

```
VC_APPS_DM_NO_WRITE_NONEXISTENT_FILE = 1
```

Example

```
setenv VC_APPS_DM_NO_WRITE_NONEXISTENT_FILE 1
```

VC_APPS_DM_REMOVE_UNUSED_MODULE

Description

Set this environment variable to remove modules that is no longer instantiated after the design manipulation. If this environment variable is not set, the unused modules are kept.

Syntax

```
VC_APPS_DM_REMOVE_UNUSED_MODULE = 1
```

Example

```
setenv VC_APPS_DM_REMOVE_UNUSED_MODULE 1
```

VC_APPS_DM_SEPARATE_PORT

Description

Set this environment variable to separate the port declaration and the type declaration. If this environment variable is not set, the port and type are declared in the same line.

Syntax

```
VC_APPS_DM_SEPARATE_PORT = 1
```

Example

```
setenv VC_APPS_DM_SEPARATE_PORT 1
```

Appendix B: Customizing Verdi

This appendix describes various ways to customize the Verdi platform. The appendix also describes the associated resource and configuration files.

This chapter consists of the following sections:

Resource and Configuration Files

The resource and configuration files can be customized. This section consists of the following resource/configuration files:

novas.rc

The `novas.rc` resource file stores the information to set window size, window position, signal height, spacing, default signal colors, menus, buttons, toolbars, and library information. The `novas.rc` resource file is saved in the local working directory after the Verdi platform is invoked for the first time. The next time the Verdi platform is started, the program looks for the `novas.rc` resource file in the following directories in the following sequence:

1. The directory and file specified by the `-rcFile` option on the Verdi command line.
2. The directory and file specified by the `NOVAS_RC` environment variable.
3. The working directory from which the Verdi platform is invoked (`./novas.rc`).
4. The home directory (`<HOME>/novas.rc`).
5. The installation directory (`$VERDI_HOME/etc/novas.rc`).

The `novas.rc` resource file only allows one set of preferences to be loaded at a time. To have a combination of global/local settings, use the `manage.rc` resource file instead.

NOTE: Any `novas.rc` resource file loaded with `-rcFile` option takes precedence over the `manage.rc` resource file located in the `/etc` directory or specified with the `NOVAS_MANAGE_RC_DIR` environment variable. It is not recommended to use `-rcFile` to save/load preferences to different files.

A sample `novas.rc` resource file with comments is available for reference in the `$VERDI_HOME/etc/novas.rc` directory.

IMPORTANT!

Unless the library information is updated, it is strongly recommended that the `novas.rc` resource file is only modified via the *Preferences* form by selecting the **Tools -> Preferences** command from a frame.

The syntax of the `novas.rc` resource file is described below along with an example of its use. The `novas.rc` resource file uses the pound # sign to designate comment lines and square brackets `[ ]` to define sections.

Syntax:

```
[Section]
```

```
Key = Value(s)
```

NOTE: Use the `-rcinfo` option on the Verdi command line to obtain a listing with descriptions of all sections, keys, and values available in the `novas.rc` resource file.

Example:

```
[Editor]
editorName = Vi
```

manage.rc

The `manage.rc` resource file specifies which resource files (*.*rc*) to use for the specified preferences and enables you to choose a file/location to save the preferences. If the `manage.rc` resource file exists, the file is searched from this location in the following sequence:

1. The installation directory (`$VERDI_HOME/etc/manage.rc`) if the `NOVAS_MANAGE_RC_DIR` environment variable is not set; or
2. The directory and file specified by the `NOVAS_MANAGE_RC_DIR` environment variable (`<specified_path>/manage.rc`).

NOTE: For more details and an example file, see the *Specifying Preference Load/Save Locations* section.

In the `manage.rc` resource file, multiple resource files are supported. Each resource file can contain partial or full sections of the environment setting. For overlapped information, the latter overwrites the previous. The search is stopped after the `manage.rc` resource file is found. An existing `novas.rc` resource file is split into multiple `*.rc` resource files and then the load order of the resource files is specified. The file sections that must be saved to the specified resource file is also specified.

The `manage.rc` resource file uses the pound `#` sign to designate comment lines and square brackets `[ ]` to define sections. The `manage.rc` resource file includes three sections: `[File]`, `[Load]`, and `[Save]`. Before defining these sections, specify the first line in the `manage.rc` resource file as `@Manage rc file Version 1.0` so that the Verdi platform can correctly recognize the file.

The three sections of the `manage.rc` resource file are described below.

[File]

This section defines the macros that represent the file names for use in other sections. It provides an association between a symbolic name and a file on the disk. The name of the macro is not important but it must be unique.

You can set a system environment variable to use as a substitution of all or part of the file path.

If the specified file does not exist, a warning message is printed and only the default preferences from the Verdi installation are loaded. However, upon exiting the Verdi platform, the file is created as specified.

Syntax:

```
name = path/rc_file.rc
name2 = $ENV/rc_file2.rc
...
```

Example:

```
[File]
library_file = ./novas_lib.rc
form_file = $PROJ/novas_file.rc
default_file = ~/novas.rc
```

In the above example:

`library_file` is used to specify preferences from `./novas_lib.rc`.
`form_file` is used to specify preferences from `$PROJ/novas_file.rc`.
where, `PROJ` is set to `/home/proj/pref` (for example,
`setenv PROJ /home/proj/pref`)
`default_file` is used to specify preferences from `~/novas.rc`.

[Load]

This section defines the load order of the resource files (`*.rc`). Duplicate keys (preferences) are replaced by later definitions if the key is repeated in files later in the load order.

Syntax:

```
rc_file
...
```

Example:

```
[Load]
default_file
form_file
```

```
library_file
```

In the above example:

The load order is `default_file`, followed by `form_file`, followed by `library_file`.

[Save]

Specifies which sections are saved to the specified `rc` file.

Syntax:

```
rc_file = SECTION_ALL | SECTION_REMAIN | MODIFIED_KEY |
section [,section ...]
```

where,

`SECTION_ALL`: All sections get saved to the specified *rc* file.

`SECTION_REMAIN`: Only sections not assigned to an `rc` file are saved to the specified *rc* file.

`MODIFIED_KEY`: All keys whose value is modified. Before writing the modified key to the destination file, it is merged with the destination file first if the destination file exists. You cannot use `MODIFIED_KEY` with other keywords in the same file. Keys are modified by the user behavior. They can also be modified by the tool.

`section`: Save the specified *novas.rc* "section" to the specified *rc* file. Multiple sections can be specified; they are separated by commas.

Example:

```
[Save]
library_file = Library
form_file = Form,General,XXXX
default_file = SECTION_REMAIN
```

In the above example:

[`Library`] is saved to `./novas_lib.rc`
[`Form`], [`General`] and [`V`] are saved to `./novas_file.rc`
All other sections are saved to `~/novas.rc`.

novas.conf

The `novas.conf` configuration file stores the information about frame layout information (dock/undock, maximize/restore, and display/hide).

When the Verdi platform is started, the program looks for the `novas.conf` configuration file to read in the following directories in the following sequence:

1. The file specified by the `-guiConf` option on the Verdi command line.
2. The file specified by the `NOVAS_GUICONF` environment variable.
3. The working directory from where the Verdi platform is invoked (`./novas.conf`).
4. The home directory (`<HOME>/novas.conf`).

The write sequence for the `novas.conf` configuration file is the same sequence as the read sequence.

For example:

1. If there is a readable/writable `/a/b/c/d/novas.conf` file, and the `/a/b/c/d/` working directory is readable/writable. The `/a/b/e/f` directory is writable, but `/a/b/e/f/novas.conf` does not exist then the `NOVAS_GUICONF` environment variable does not exist.

   If you execute the `novas -guiConf /a/b/e/f/novas.conf` command, then Verdi reads the `novas.conf` file from the `/a/b/c/d/novas.conf` file, and writes frame layout information into the */a/b/e/f/novas.conf* file. The console message appears as follows:

   ```
   guiConf (read) = /a/b/c/d/novas.conf (working directory)
   guiConf (write) = /a/b/e/f/novas.conf (-guiConf)
   ```

2. If there is a readable/writable `/a/b/c/d/novas.conf` file, and the `/a/b/c/d/`working directory is readable/writable. The `/a/b/e/f` directory is writable, but `/a/b/e/f/novas.conf` file does not exist then `NOVAS_GUICONF` environment variable points to the `/a/b/e/f/novas.conf` file.

   If you execute the `novas` command, then Verdi reads the `novas.conf` file from the `/a/b/c/d/novas.conf` file, and writes frame layout information into `/a/b/e/f/novas.conf` file. The console message appears as follows:

   ```
   guiConf (read) = /a/b/c/d/novas.conf (working directory)
   guiConf (write) = /a/b/e/f/novas.conf (NOVAS_GUICONF)
   ```

3. If there is no readable *novas.conf* file in both the `/a/b/c/d/` working directory and the `$HOME` directory. The `/a/b/c/d` directory is writable, but `/a/b/c/d/novas.conf` does not exist.

If you execute the novas command in the /a/b/c/d/ working directory, then Verdi does not read the novas.conf file from the /a/b/c/d/ working directory, and writes frame layout information into a new file, /a/b/c/d/novas.conf. The console message appears as follows:

```
guiConf (read) = N/A (default)
guiConf (write)= /a/b/c/d/novas.conf (working directory)
```

signal.rc

The signal.rc configuration file stores the current *nWave* information to restore window layout, opened files, view range, cursor/marker position, loaded signals, and signal attributes. The syntax of the signal.rc file is described below along with an example of its use.

For all $$top/signal_name_with_full_scope_name lines in the signal.rc file, $$top is translated as a one-level top path of the current active file if the current active file has only one top path. If the current active file has more than one top path, $$top is translated as each top path until matching the first $$top/signal_name_with_full_scope_name.

For example, when the signal.rc file contains $$top/dut/reset:

- If the current active file has only one top path testbench, $$top/dut/reset becomes /testbench/dut/reset.
- If the current active file has three top paths and they are a, b, and c, $$top initially is replaced with a. If /a/dut/reset exists, no further action is taken. If /a/dut/reset does not exist, b and then c are tried until a match is found.

The signal.rc file uses the ";" character to designate comment lines.

NOTE: For details on using the same signal.rc file at different hierarchical levels, see the *Using the Same Signal File at Different Scopes* section.

Syntax:

```
Key Value(s)
```

The following table lists the key values in the *nWave* signal file (signal.rc) with descriptions and examples:

Item	Description	Example
viewPort	Specifies *nWave* frame position, size, and layout.	ViewPort 56 169 960 332 102 67

Item	Description	Example
openDirFile	Specifies files (that is, FSDB or VCD files) to open in *nWave* including information about delimiter, time offset, and file path. Each file is enclosed with double quotes (“ ”). -*d* defines the delimiter for hierarchical signal names. Default delimiter is “/ ”. -*s* defines the time offset.	openDirFile -d / -s 11.0K 01ns "" "/qa/ddts_case/ SPS0127333/rtl.evcd.fsdb"
fileTimeScale	The time scale of the active file specified through the **Waveform -> Waveform Time -> Set File Time Scale** command. The specified time scale affects the whole waveform pane - signals are shown based on the same ruler with the same time scale.	fileTimeScale "/dq2/demo/ verilog/extract.vf" 20ns
signalSpacing	Specifies the spacing between signals with the height measured in pixels.	signalSpacing 3
zoom	Specifies the waveform viewport range.	zoom 0.0 1245.0
cursor	Specifies the current cursor position.	cursor 0.0
marker	Specifies the current marker position.	marker 0.0
top	Specifies the first visible signal in the *nWave* signal pane. It is used when the window is not large enough to show all signals at once.	top 2
markerPos	Specifies the position of the signal cursor in the *nWave* signal pane.	markerPos 25
curSTATUS	Specifies the current Toolbar search type. Its options include *ByChange*, *ByEvent*, *ByRising*, *ByFalling*, *ByValue*, and *ByCmpError*.	curSTATUS ByChange
addGroup	Adds group(s) with the specified name. A default group name (G1) is generated by *nWave*. If signals are in different groups, the group names are specified.	addGroup “G1”

Item	Description	Example
addSignal	Adds signals after the desired group name.	addSignal -h 15 /system/Clock
	-h is for height and is always specified. The default value *15* is used unless specified otherwise.	addSignal -h 35 /system/Clock
	-c is for color and is optional.	addSignal -c ID_CYAN3 -h 15 /system/Clock
	-holdScope keeps the same scope as the signal above.	addSignal -h15 -holdScope addr[7:0]
	-ls is for line style and is optional.	addSignal -h15 -ls l_dot
	-lw is for line width and is optional.	addSignal -h15 -lw 2
	Specifying radix and notation is optional. Radix options include *-HEX*, *-OCT*, *-UDEC*, *-BIN*, *-ASC*, and *-IEEE754*. Notation options include *-UNSIGNED*, *-2COMP*, *-1COMP*, and *-MAGN*.	addSignal -h 15 -MAGN -BIN -holdScope Mux3_Sel[1:0]
addDelaySig	Specifies the delay type and value of the delayed signal by using the suboptions *-Delay*, *-Rising_Delay*, or *-Falling_Delay* with the delay time and unit.	addDelaySig "addr[7:0] ->>300" -Delay 300.000000 1n "/system/addr[7:0]"
addExprSig	Adds a new signal which is a logical expression of existing signals and includes the full hierarchical name.	addExprSig -b 1 Test1 "/system/VMA" & "/system/clock->>300"
	-b is for bit size of expression signals.	
addRenameSig	Adds the renamed signal. Specifies the renamed signal first and original signal second.	addRenameSig "/system/R_VMA" "/system/VMA"

Item	Description	Example
addEventSignal	Adds the event signal with the specified name. -*b* and -*e* specify the begin and end time for the event signal. -*m* indicates whether there are multiple events. It is the opposite of the status of the *One-shot* option in the *Complex Event* tab of the *Event Window* form ("1" indicates multiple, that is, *One-shot* is toggled *off*). -*h* is for height and is always specified. Uses default value unless specified otherwise.	addEventSignal -b 0.000000 -e 0.000000 -m 1 -h 15 /event5
activeDirFile	Sets active file(s). If multiple files are opened and *activeDirFile* is commented out, *nWave* does not find the correct signals.	activeDirFile "/da2/steven" "../../verify1/rtl/fsdb/v1.6/verilog.dump.fsdb"
getSignalForm	Specifies the *Get Signals* form status.	getSignalForm Close
userBusMem	Specifies one signal to be included in the user-defined bus. Repeat for each bus member. Must be followed by *saveRunSig*.	userBusMem /system/VMA userBusMem /system/reset saveRunSig "mybus[1:0]"
saveRunSig	Creates runtime bus with the specified name. Must be preceded by one or more *userBusMem*.	
EVENT	Specifies the event name and condition.	EVENT event3 "/system/clock" === 'b0
aliasmapname	Creates an alias map with the specified name. Must be followed by one or more aliases.	aliasmapname ALU_en nalias OK 0 NULL nalias NO 1 NULL
alias	Adds an alias member. Must be preceded by *aliasmapname*.	
slicemapname	Creates a slice map with the specified name. Must be followed by *nslice*.	slicemapname slice_ALU nslice 7:4 Hexadecimal nslice 3:2 Binary nslice 1:0 number2string
nslice	Adds a member to the slice alias. Must be preceded by *slicemapname*.	
aliasname	Applies alias to the next signal.	aliasname ALIAS_test addSignal -h 15 /system/data[7:0]
windowTimeUnit	Defines window time unit.	windowTimeUnit 1ns

Item	Description	Example
include	Includes multiple resource files (*.rc). **NOTE:** Restoring the top resource file restores all sub-resource files that it includes.	rc_file_1: waveDefine A system //A is defined as system; include rc_file_2 include rc_file_3 rc_file_2: waveDefine A top //A is defined as top; rc_file_3: addSignal ${A}/a //addSignal top/a because A is defined as top

Example:

```
; Window Layout <x> <y> <width> <height> <signalwidth>
<valuewidth>
viewPort 302 116 960 394 286 65

; File list:
; openDirFile [-d delimiter] [-s time_offset] [-rf
auto_bus_rule_file] path_name file_name
openDirFile -d / "" "/dq2/qa/demo/53/verilog/rtl/rtl.fsdb"

; file time scale:
; fileTimeScale ### s|ms|us|ns|ps

; signal spacing:
signalSpacing 3

; windowTimeUnit is used for zoom, cursor, and marker
; waveform viewport range
zoom 5859.375000 6640.625000
cursor 6251.000000
marker 0.000000

; user define markers
; userMarker time_pos marker_name
; visible top row signal index
top 1
; marker line index
markerPos 11


; Run Time Signal and Member
; userBusMem member ...
; saveRunSig name
userBusMem /system/i_cpu/i_ALUB/PC[3]
```

```
userBusMem /system/i_cpu/i_ALUB/PC[2]
userBusMem /system/i_cpu/i_ALUB/PC[1]
saveRunSig "PC_addrSelector[3:1]"

; Run Time Signal and Member
; userBusMem member ...
; saveRunSig name
userBusMem /system/i_cpu/i_ALUB/PC[7]
userBusMem /system/i_cpu/i_ALUB/PC[6]
userBusMem /system/i_cpu/i_ALUB/PC[5]
userBusMem /system/i_cpu/i_ALUB/PC[4]
saveRunSig "PC_Data[7:4]"

; rename signal list
; addRenameSig new_name org_name

addRenameSig "/system/i_cpu/i_ALUB/ALU_en" "/system/i_cpu/i_ALUB/
ALU[0]"

; logical expression list
; addExprSig expr_name expression_string

addExprSig -b 1 NotClock !"/system/i_cpu/i_ALUB/clock"

; event list
; addEvent event_name event_expression
; curEvent event_name

COMPLEX_EVENT_BEGIN

EVENT event0 "/NotClock" === 'b1  && "/system/i_cpu/i_ALUB/
ALU_en" === 'h1
EVENT event1 "/system/i_cpu/ALU[7:0]" === 'h23

COMPLEX-EVENT  seq1  {

   LEVEL 0  {
         timer1: STOP  timer2: STOP
        IF   ( event0 OCCURS  1)  THEN goto level 1
        ELSE    goto level 0
   }

   LEVEL 1  {
         timer1: STOP  timer2: STOP
        IF   ( event1 OCCURS  1)  THEN trigger
        ELSE    goto level 1
   }
}
       curEvent event0

COMPLEX_EVENT_END

; toolbar current search type
; curSTATUS search_type
```

```
curSTATUS ByValue

aliasmapname number2string
nalias Zero 0 NULL
nalias One  1 NULL

slicemapname slice_ALU
nslice 7:4 Hexadecimal
nslice 3:2 Binary
nslice 1:0 number2string

slicemapname slice_Bus
nslice 7:4 Hexadecimal
nslice 3:2 Binary
nslice 1:0 number2string

aliasmapname ALU_en
nalias OK 0 NULL
nalias NO 1 NULL

addGroup "G1"
aliasname slice_ALU
activeDirFile "" "/dq2/qa/demo/53/verilog/rtl/rtl.fsdb"
addSignal -h 15 -UNSIGNED -HEX /system/i_cpu/ALU[7:0]
addSignal -h 15 -UNSIGNED -UDEC /system/i_cpu/i_ALUB/ALU[7:0]
aliasname ALU_en
addSignal -h 15 -UNSIGNED -HEX -holdScope ALU_en
aliasname slice_Bus
addSignal -h 15 -UNSIGNED -HEX /system/i_cpu/PC[7:0]
addSignal -h 15 -UNSIGNED -HEX /system/i_cpu/i_ALUB/PC[7:0]
addSignal -h 15 -UNSIGNED -HEX /PC_addrSelector[3:1]
addSignal -h 15 -UNSIGNED -HEX /PC_Data[7:4]
addSignal -h 15 /system/i_cpu/i_ALUB/clock
addSignal -h 15 /NotClock
addEventSignal -b 0.000000 -e 0.000000 -m 1 -h 15  /event0
addEventSignal -b 0.000000 -e 12500.000000 -m 1 -h 30  /seq1
addGroup "G2"
aliasname slice_Bus
addSignal -h 15 -UNSIGNED -HEX /system/i_cpu/i_ALUB/X0[7:0]
addSignal -h 15 -UNSIGNED -HEX -holdScope Y0[7:0]
addGroup "G3"
aliasname slice_Bus
addSignal -c ID_RED5 -ls solid -lw 1 -h 15 -UNSIGNED -HEX
/system/i_cpu/i_PCU/C1
aliasname slice_Bus
addSignal -c ID_RED5 -ls solid -lw 1 -h 15 -UNSIGNED -HEX
-holdScope C5
aliasname slice_Bus
addSignal -c ID_RED5 -ls solid -lw 1 -h 15 -UNSIGNED -HEX
-holdScope C6
addGroup "G4"
addSignal -h 15 -UNSIGNED -HEX /system/i_pram/dataout[7:0]
addGroup "G5"
addGroup "G6"
```

```
; getSignalForm Scope Hierarchy Status
; active file of getSignalForm
activeDirFile "" "/dq2/qa/demo/53/verilog/rtl/rtl.fsdb"

GETSIGNALFORM_SCOPE_HIERARCHY_BEGIN
getSignalForm close

"/system"
"/system/i_cpu"

SCOPE_LIST_BEGIN
"/system"
"/system/i_cpu"
"/system/i_cpu/i_ALUB"
"/system/i_cpu/i_PCU"
"/system/i_pram"
"/system/i_cpu/i_CCU"
SCOPE_LIST_END

GETSIGNALFORM_SCOPE_HIERARCHY_END
```

NOTE: For users that look at an existing signal (`*.rc`) file and see *Magic 271485* and *Revision 5.3v16* in the first line, *Magic 271485* and *Revision 5.3v16* are optional.

Session (.ses) File

The `session.ses` resource file stores the current Verdi information to restore window layout, current scopes, current design, viewing options, and bookmarks. The syntax of the session file is described below along with an example of its usage.

The session file uses square brackets `[ ]` to define sections.

Syntax:

```
[section_name]
section_key = section_value1 section_value2 section_value3...
```

Example:

```
[hb]
viewport = 156 253 804 500 182 299 148 804
activeNode = "system”
interactiveMode = False
viewType = Source
simulatorMode = False
```

Modifying Application Windows/Frames

This section provides details on how to customize the main window and individual frames of the Verdi platform. You can change the appearance of the following items:

- Menus, Bind Keys, and Toolbars
- Frame Layout
- Position and Dimensions
- Colors and Fonts

The Window Settings for Qt Applications section provides details on settings in the window environment that impact the display of Qt applications.

Menus, Bind Keys, and Toolbars

You can customize menus, bind keys, and toolbars through the **Tools** -> **Customize Menu/Toolbar** command. For details, see the Customize Menu/Toolbar command description in the *nTrace* chapter. To view an example, see the *User Interface Tutorial* chapter in the *Verdi User Guide and Tutorial* document.

You can modify menu commands using Tcl. For the customization of the Tcl commands, see the *User Interface Commands* section in the *Verdi and Siloti Tcl Reference*.

When the Verdi platform is exited, changes to the menus, bind keys, and toolbars are saved to the `novas.rc` resource file.

Frame Layout

The Verdi platform consists of a central framework where all components are docked. Each component (window or form) has a default frame location. The default frame locations may be changed and some components may also become fully-functioning separate windows. For details on docking/undocking components, see the *User Interface* chapter of the *Verdi User Guide and Tutorial* manual. For commands related to frame layout, see the Save/Restore User Layout and Window Manager commands in the *nTrace* chapter.

When the Verdi platform is exited, the previous and current layouts are saved to the `novas.conf` configuration file unless the `-guiConf` option is specified on the Verdi command line. For details about the `-guiConf` option, see the *verdi* utility description.

Position and Dimensions

The simplest way to adjust the application window's position and dimensions is to drag the window directly. The Verdi platform saves the last position and dimensions of the window when the program is exited.

Alternatively, you can change the `novas.rc` resource file, which specifies the information for the *nTrace* main window, the *Text Viewer* frame, the *nWave* window, and the *nSchema* window. The `novas.rc` resource file syntaxes are listed below with examples.

Syntax:

```
Key = Value(s)
```

Example:

```
# Definition for nTrace window
# windowLayout = x y width height tree_window_width
#tree_window_height
# message_window_width message_window_height
[hb]
...
windowLayout = 0 0 755 567 238 350 755 148
...
# Definition for schematics window
#viewport = x y width height
[schematics]
...
viewport = 396 -3 500 600
...
# Definition for File Viewer window
# viewport = x y width height
[Text]
...
viewport= -3 179 475 480
...
[Wave]
...
viewPort= 0 29 1019 161 119 45
```

Colors and Fonts

The definitions of a window colors and fonts are defined in the `novas.rc` resource file.

NOTE: It is strongly recommended that colors and fonts are only changed via the *Preferences* form opened by the **Tools -> Preferences** command for all windows/frames.

The syntax of the `novas.rc` resource file is listed below with examples.

Syntax:

```
Key = Value(s)
```

Example:

```
textLineNo = black
```

For the values of the color and font variables, see the following list:

Variables	Possible Values
Color	ID_BLACK, ID_GRAY1, ID_GRAY2, ID_GRAY3, ID_GRAY4, ID_GRAY5, ID_GRAY6, ID_WHITE, ID_RED1, ID_RED2, ID_RED3, ID_RED4, ID_RED5, ID_RED6, ID_RED7, ID_RED8, ID_ORANGE1, ID_ORANGE2, ID_ORANGE3, ID_ORANGE4, ID_ORANGE5, ID_ORANGE6, ID_ORANGE7, ID_ORANGE8, ID_YELLOW1, ID_YELLOW2, ID_YELLOW3, ID_YELLOW4, ID_YELLOW5, ID_YELLOW6, ID_YELLOW7, ID_YELLOW8, ID_GREEN1, ID_GREEN2, ID_GREEN3, ID_GREEN4, ID_GREEN5, ID_GREEN6, ID_GREEN7, ID_GREEN8, ID_CYAN1, ID_CYAN2, ID_CYAN3, ID_CYAN4, ID_CYAN5, ID_CYAN6, ID_CYAN7, ID_CYAN8, ID_BLUE1, ID_BLUE2, ID_BLUE3, ID_BLUE4, ID_BLUE5, ID_BLUE6, ID_BLUE7, ID_BLUE8, ID_PURPLE1, ID_PURPLE2, ID_PURPLE3, ID_PURPLE4, ID_PURPLE5, ID_PURPLE6, ID_PURPLE7, ID_PURPLE8

Variables	Possible Values
Font	"Helvetica 8," "Helvetica 10," "Helvetica 12," "Helvetica 14," "Helvetica 18," "Helvetica 24," "Helvetica bold 8," "Helvetica bold 10," "Helvetica bold 12," "Helvetica bold 14," "Helvetica bold 18," "Helvetica bold 24," "Times 8," "Times 10," "Times 12," "Times 14," "Times 18," "Times 24," "Times bold 8," "Times bold 10," "Times bold 12," "Times bold 14," "Times bold 18," "Times bold 24," "Clean 8," "Clean 10," "Clean 12," "Clean 14," "Clean 16," "Clean bold 8," "Clean bold 10," "Clean bold 12," "Clean bold 14," "Clean bold 16," "Courier 8," "Courier 10," "Courier 12," "Courier 14," "Courier 18," "Courier 24," "Courier oblique 8," "Courier oblique 10," "Courier oblique 12," "Courier oblique 14," "Courier oblique 18," "Courier oblique 24," "Courier bold 8," "Courier bold 10," "Courier bold 12," "Courier bold 14," "Courier bold 18," "Courier bold 24," "Fixed 8," "Fixed 10," "Fixed 12," "Fixed 14"

Changing Fonts for Verdi Menus

The fonts used in the Verdi menus are configurable through the `novas.rc` resource file. To change the default font, the following section must be manually added in the `novas.rc` resource file:

```
[Menu]
font = Helvetica 12
```

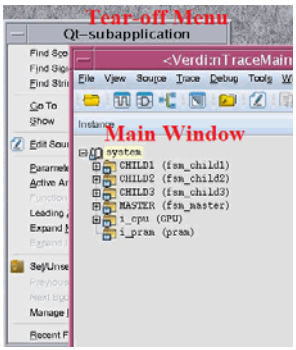
The GUI cannot be used to configure the font. The default font is *Helvetica 12*. As summarized above, other possible font values are available.

Changing Toolbar Icon Sizes

To change toolbar icon sizes, see the **Toolbar Icon Size** option on the **General -> General Page** of the *Preferences* form (invoked with the **Tools -> Preferences** command).

Window Settings for Qt Applications

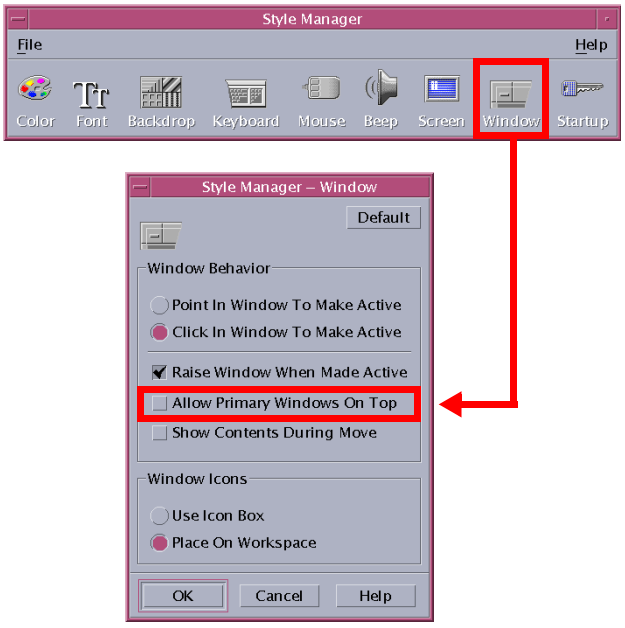
The window environment settings can impact the way Verdi GUI windows are displayed. For example, the following figure shows a situation where the main window causes the tear-off menu to be hidden:



This section describes recommended window environment settings.

CDE Environment

To prevent dialog forms, undocked widgets, or tear-off menus from being hidden by the active main window, turn *off* the **Allow Primary Windows On Top** option in the *Style Manager - Window* form opened by clicking the **Window** icon on the *Style Manager* toolbar.



Defining Default Compiler Language Mode

The Verdi platform provides a language mode compiler option in the `novas.rc` resource file. [HB] is related to VHDL/Verilog importing and is applied to Verdi and Siloti systems.

Session Definitions and Keys

```
[HB]
import_default_language = Verilog | Verilog-2001 | SystemVerilog
| SystemVerilog-2009 | SystemVerilog-2012 | VHDL-87 |VHDL-93 |
VHDL-2000 | VHDL-2008
```

The default import language is *Verilog*.

NOTE: The import_default_language field is only supported and used in GUI mode.

The Verdi platform looks for the language mode setting according to the following search sequence:

1. Command-line option.

2. Environment variable (NOVAS_LANG_VER).
3. The novas.rc resource file setting (described in this section).

Defining Libraries

Design libraries and Memory Definition Table (MDT) libraries used in the Verdi platform are configurable via the novas.rc resource file. To change the default libraries, the following sections must be manually modified in the novas.rc resource file. The GUI is not available to configure the libraries.

This section provides details on how to customize the following libraries:

- *Design Libraries*
- *Memory Definition Table Libraries*

Design Libraries

The Verdi platform provides library-mapping schemes for better design library usage and sharing. Modules are compiled into different libraries and then the Verdi platform is instructed to search the libraries through the library-mapping scheme. Library mapping is used when performing design compilation (invoking vericom or vhdlcom) and elaboration (invoking the Verdi platform).

NOTE: Before VHDL compilation, you must copy the novas.rc resource file to the local working directory and then manually modify according to the desired library mapping.

This section consists of the following topics:

- *Basic Library Mapping*
- *NC-Like Library Mapping Scheme for Verilog Designs*

Basic Library Mapping

The basic library mapping scheme is defined in the `[Library]` section of the `novas.rc` resource file. The physical location of a logical library is defined here. The following is the library-mapping format in the `novas.rc` resource file:

Syntax:

```
[Library]
<logical_name> = <physical_location>
others = <physical location of another resource file>
```

Example:

```
[Library]
work = ./work
mylib = /home/module/mylib.lib++
others = /project/chip/special_lib.rc
```

The following are the basic library-mapping rules:

1. The `<logical_name>` is case-insensitive and is converted to lowercase.
2. The `<physical_location>` is case-sensitive.
3. The `<logical_name>` defined under the `[Library]` section is searched prior to the `<logical_name>` defined under the `[Library]` section of the resource file specified by the `others` keyword.
4. The `[Library]` section of the resource file specified by the `others` keyword can have another resource file.
5. If the same `<logical_name>` is applied, the former one is used.
6. The `<physical_location>` combines the relative or absolute path and physical library name. For example,

   ```
   novaslib1 = /design/novasLib
   novaslib2 = ../novasLib
   ```

7. The Verdi platform extends the physical location name of a logical library with `.lib++` automatically if `.lib++` is not specified in the library mapping. The following two mappings are the same:

   ```
   novaslib = ./novasLib.lib++
   novaslib = ./novasLib
   ```

Multiple Verilog and VHDL libraries are supported in the `[Library]` section.

NOTE: The recommendation is to compile Verilog and VHDL design sources into separate libraries.

NC-Like Library Mapping Scheme for Verilog Designs

The Verdi platform supports the Cadence NC's library binding scheme using some variables in the `novas.rc` resource file and some options for `vericom`. If the provided NC binding scheme is applied to the Verdi-compiled library, you can see the libraries and the cells (the design units) that are built and the view names.

The NC-like library-mapping scheme is defined in the `[NC_Sim]` section the Verdi resource file, `novas.rc` using the following four variables:

- *The libMap Variable*
- *The viewMap Variable*
- *The workLibrary Variable*
- *The defaultView Variable*

The libMap Variable

This variable has the same function as `LIB_MAP` in `hdl.var` of NC to map files and directories to library names, but the syntax is different. The syntax is as follows:

```
libMap=(directory=>library, directory/..=>library,
file=>library, +=>library)
```

Use the plus sign (+) to specify a default library for files or directories that are not explicitly specified. For example:

```
libMap=(../src_1=>src_1, ../src_2/..=>src_2,../src/
top.v=>src_1, +=>worklib)
```

This causes:

- Any files in the `../src_1` directory to be compiled into `src_1.lib++`.
- All files in and below the `../src_2` directory to be compiled into `src_2.lib++`.
- `../src/top.v` to be compiled into `src_1.lib++`.
- Any other files to be compiled into `worklib.lib++`.

If the plus sign (+) is not specified for a default library, `work` is used as the default library.

The viewMap Variable

This variable has the same function as `VIEW_MAP` in `hdl.var` of NC to map file extensions to view names, but the syntax is different. The syntax is as follows:

```
viewMap=(file_extension=>view_name, ..., +=>view_name)
```

Use the plus sign (+) to specify a default view. For example:

```
viewMap=(.v => behav, .rtl => rtl, .gate => gate, + => vlog)
```

This causes:

- Any files with the `.v` extension create the view name of `behav`.
- Any files with the `.rtl` extension create the view name of `rtl`.
- Any files with the `.gate` extension create the view name of `gate`.
- Any other file extensions map to a view called `vlog`.

If the plus sign (+) is not specified for a default view, then the following rules are used:

- Use the view name of `module` for modules and macro modules.
- Use the view name of `udp` for UDPs.

The workLibrary Variable

This variable has the same function as `WORK` in `hdl.var` of NC to define the work library. All design units are compiled into the specified library. The syntax is as follows:

```
workLibrary=library_name
```

For example:

```
workLibrary=novas
```

All design units are compiled into `novas.lib++`.

If both `libMap` and `workLibrary` are defined, the definition of `workLibrary` overrides `libMap`.

The defaultView Variable

This variable has the same function as `VIEW` in `hdl.var` of NC to define the view name. All design units receive the same view name. The syntax is as follows:

```
defaultView=view_name
```

For example,

```
defaultView=novas
```

All design units are assigned a view name of `novas`.

If both `viewMap` and `defaultView` are defined, the definition of `defaultView` overrides `viewMap`.

Memory Definition Table Libraries

The Verdi platform provides a simple mechanism for mapping memory models generated by a vendor memory compiler. In the `novas.rc` resource file, the `[MDT]` section specifies the memory name mapping. These mappings are modified if a different naming convention is used. For a summary of supported vendor memory models, see the *MDT Library Installation* section of the *Installation and System Administration Guide* f.

Syntax

```
[MDT]
Verdi_supported_vendor_memory_name = user_memory_module_name
```

Example:

```
[MDT]
ART_RF_SP   = spr[0-9]*bx[0-9]*
ART_RF_2P   = dpr[0-9]*bx[0-9]*
ART_SRAM_SP = spm[0-9]*bx[0-9]*
ART_SRAM_DP = dpm[0-9]*bx[0-9]*
VIR_SRAM_SP=   hdsd1_[0-9]*x[0-9]*cm4sw1
VIR_SRAM_DP=   hdsd2_[0-9]*x[0-9]*cm4sw1
VIR_RF_SP=   rfsd1_[0-9]*x[0-9]*cm2sw0
VIR_RF_DP=   rfsd2_[0-9]*x[0-9]*cm2sw1
VIR_STAR_SRAM_SP=   shsd1_[0-9]*x[0-9]*cm4sw0
```

The Verdi platform supports wildcard name matching to accommodate word/width parameter settings (for example, `spr256bx16` and `spr1024bx32`) in the memory compiler. The basic rules for module-name pattern match (regular expressions) are as follows:

*	A sequence of 0 or more matches of the atom.
+	A sequence of 1 or more matches of the atom.
?	A sequence of 0 or 1 match of the atom.
{m}	A sequence of exactly m matches of the atom.
{m,}	A sequence of m or more matches of the atom.
{m,n}	A sequence of m through n (inclusive) matches of the atom; m must not exceed n.

An atom can consist of the following:

.	Matches any single character.
[chars]	A bracket expression, matching any one of the chars.
(re)	Matches for *re*, where, *re* is any regular expression.

Specifying Preference Load/Save Locations

The `manage.rc` resource file specifies loading and saving rules for Verdi preferences. There are the following three methods to include this resource file:

1. Place the `manage.rc` resource file in the `$VERDI_HOME/etc` directory. If the `manage.rc` resource file does not exist, the `novas.rc` resource file is used for loading preferences.
2. Set the `NOVAS_MANAGE_RC_DIR` environment variable to specify the location of the `manage.rc` resource file.
3. Use the `-managercFile` option on the Verdi command line to specify the full path of the `manage.rc` resource file. A `NOVAS_MANAGE_RC_FILE` environment variable is provided for the same purpose.

   When multiple `manage.rc` or `novas.rc` resource files are loaded, the conflicts are resolved in the following order:

   a. Command-line option: `-managercFile`
   b. Command-line option: `-rcFile`
   c. Environment variable: `NOVAS_MANAGE_RC_FILE`
   d. Environment variable: `NOVAS_MANAGE_RC_DIR`
   e. Environment variable: `NOVAS_RC`

Examples:

```
verdi -rcFile /dir1/novas.rc -managercFile /dir2/managercFile.rc
```

The resolved `rcFile` is "`/dir2/managercFile.rc`".

```
setenv NOVAS_MANAGE_RC_FILE /dir1/managercFile.rc
setenv NOVAS_MANAGE_RC_DIR /dir2
```

The resolved `rcFile` is "`/dir1/managercFile.rc`".

NOTE: 1) The `manage.rc` resource file indicated by the `NOVAS_MANAGE_RC_DIR` environment variable takes precedence over the `manage.rc` resource file that is located in the `$VERDI_HOME/etc` directory.

2) It is not recommended to use the `-rcFile` option to load the `manage.rc` resource file. The `manage.rc` resource file has a different file format and it is overwritten by the `novas.rc` format on exiting the Verdi platform.

3) Any `novas.rc` resource file loaded with the `-rcFile` option takes precedence over the `manage.rc` resource file located in the `/etc` directory or specified with `NOVAS_MANAGE_RC_DIR`. It is not recommended to use the `-rcFile` option if you need to save/load preferences to different files.

Example manage.rc File

The '`@Manage rc file Version 1.0`' line must be the first line in the `manage.rc` resource file so that the file is properly recognized by the Verdi platform.

```
@Manage rc file Version 1.0
# [File] section:
# <install-dir> will be substituted for the installation path.
# $ENV for environment variable substitution.
#
# [Load] Section:
# Duplicate keys in later files will override previous keys
# of the same name.
#
# [Save] section:
# SECTION_REMAIN: All sections not already included in other
# files.
# SECTION_ALL: All sections.
# MODIFIED_KEY: All keys whose values are modified.
#
[File]
file1 = ~/my_sch.rc
file2 = ./my_lib.rc
file3 = $TEST/other.rc
file4 = ./modified.rc
[Load]
file1
file2
file3
file4
[Save]
file2 = Library
file1 = schematics
```

```
file3 = SECTION_REMAIN
file4 = MODIFIED_KEY
```

NOTE: VERDI_HOME takes precedence over HOME; therefore, the tilde (~) used in the manage.rc resource file is replaced with the value of VERDI_HOME (instead of HOME) if both HOME and VERDI_HOME exist at the same time.

Specifying Signal Grouping Rules

The specified grouping rule file is used to perform automatic bus grouping with the rules applied in the *nWave* window.

The syntax of a signal grouping rule file is described below along with an example of its use.

Syntax

```
DELIMITER delimiter
LANGUAGE [Verilog | VHDL]

BUS name
   [LANGUAGE Verilog | VHDL]
   SCOPE scope_name
   MODULE module_name(s)
   SIGNAL signal_name(s) | RULE rule
ENDBUS
```

Keywords

Keywords are not case-sensitive. When used, the keywords are the leading word of each line in the rule file.

`DELIMITER`

Specifies the delimiter for the scope or module path. The `DELIMITER` keyword is used globally in the rule file. Its valid options are `.` or `/` .

`LANGUAGE`

Specifies the language as either Verilog or VHDL for the FSDB file. You can specify this keyword globally or within a local `BUS`/`ENDBUS` specification.

`BUS name`

Specifies the bus name for the rule (name cannot be NULL). Each BUS must have a matching ENDBUS.

`SCOPE scope_name`

Specifies one or more scope names to create buses.

`MODULE module_name(s)`

Specifies one or more module names to create buses. You can use `MODULE` with `SCOPE`.

`RULE rule`

Specifies the rules for signal name matching. The rule is a combination of a legal character set, `%s` and `%d`; where `%s` is any legal character and `%d` is the expected index of the created bus. You cannot use `RULE` with `SIGNAL` in the same `BUS`/`ENDBUS`.

`SIGNAL signal_name`

Specifies one or more existing signals to create a bus. The `SIGNAL` keyword is optional. You cannot use `SIGNAL` with `RULE` in the same `BUS`/`ENDBUS`.

`ENDBUS`

Specifies the end of a bus rule.

The following is a list of behaviors that apply when a signal grouping rule file is used:

1. If signals matching a bus rule have the same prefix and their bit index is continued from `a` to `b`, *nWave* forms a bus with range `[b:a]`. Otherwise, *nWave* forms a bus with `range[total_member_count-1 : 0]`.
2. If one or more white spaces exist in a scope name, module name, or signal name, enclose the name with double quotes (for example, `"myspace "`).
3. If a scope is not specified within a `BUS`/`ENDBUS` rule, *nWave* searches for signals in all scopes to match the rule.
4. If a collision exists between a specified bus name and an existing signal name, *nWave* skips this rule definition.
5. Single bit signals cannot be matched to a user rule.
6. Consider the `ab%s_%d_` rule, where a regular expression is created for it as follows:

   ```
   ab[a-zA-Z0-9_ ]*_[0-9]+_
   ```

7. A line beginning with a pound (#) sign in the rule file is recognized as a comment line and ignored.

Example Rules

Consider the following signals: /aaa1, /aaa2, /aaa4, /aab0, /aab1, /aab2, /aacc[0], /aacc[1], /dd_1_d, /dd_2_d, /Scope1/aa1, /Scope1/aa2, /Scope2/aa3, /Scope2/aa4. Rules can be created as shown in the following examples:

Rule 1: Specify the existing signals to form a bus.

```
BUS aaa
    aaa1 aaa2
ENDBUS
```

NOTE: The SIGNAL keyword is optional if no other keywords are used between a BUS/ENDBUS.

Results in bus aaa[2:1] consisting of signals /aaa2 and /aaa1.

Rule 2: Specify the bit index that is not continued.

```
BUS a3
    aaa%d
ENDBUS
```

NOTE: The RULE keyword is optional if no other keywords are used between a BUS/ENDBUS.

Results in bus a3[2:0] consisting of signals /aaa4, /aaa2, and /aaa1.

Rule 3: Specify the bus delimiter.

```
BUS aacc
    aa%s[%d]
ENDBUS
```

Results in bus aacc[1:0] consisting of signals /aacc[1] and /aacc[0].

Rule 4: Search all scopes.

```
bus aa
    aa%s%d
endbus
```

Results in bus /aa[5:0] consisting of signals /aaa4, /aaa2, /aaa1, /aab2, /aab1, and /aab0; results in bus /Scope1/aa[2:1] consisting of signals /Scope1/aa1 and /Scope1/aa2; results in bus /Scope2/aa[4:3] consisting of signals /Scope2/aa4 and /Scope2/aa3.

Rule 5: Search the specified scope.

```
BUS aa
    SCOPE /Scope1
    RULE aa%d
ENDBUS
```

Results in bus `/Scope1/aa[2:1]` consisting of signals `/Scope1/aa1` and `/Scope1/aa2`.

Rule 6: Ignore the specified scope or module name if it does not exist.

```
BUS aa
    SCOPE /Scope1
    SCOPE /Scope2
    SCOPE /Scope3
    MODULE Module1
    RULE aa%d
ENDBUS
```

Results in bus `/Scope1/aa[2:1]` consisting of signals `/Scope1/aa1` and `/Scope1/aa2`; results in bus `/Scope2/aa[4:3]` consisting of signals `/Scope2/aa4` and `/Scope2/aa3`. `/Scope3` and `Module1` do not exist and are ignored.

Rule 7: Specify some characters after %d.

```
BUS dd
    RULE dd_%d%s
ENDBUS
```

Results in bus `/dd[2:1]` consisting of signals `/dd_1_d` and `/dd_2_d`.

Example Rule File

```
### delimiter

DELIMITER /
LANGUAGE Verilog
### Verilog ### : [verilog.dump.fsdb]

# The following rule can take a long time to run
BUS BUS_forAllScope
 RULE %s%d
ENDBUS

BUS  net_n_at_SCOPE_system_icpu_iALUB
 SCOPE /system/i_cpu/i_ALUB
 RULE net_%d
ENDBUS
```

```
BUS  net_n_at_root
 SCOPE /
 RULE net_%d
ENDBUS
```

Using the Same Signal File at Different Scopes

The Verdi platform supports the ability to create macros and use them in the save signal (`signal.rc`) file to easily re-define the scope for a signal. For example, a signal list is created for a module. Then, the module is included in the testbench or chip integration so the scope of the module changes and the signal list is no longer valid.

nWave does not generate a signal file with macro definitions directly. The signal file must be modified after the initial creation and after any saves of the same `signal.rc` file.

The macro is defined in the `signal.rc` file or as an environment variable. The syntax to define the macro is as follows:

- In the `.rc` file:

  ```
  waveDefine macro_name token_string
  ```

- As an environment variable:

  ```
  setenv macro_name "token_string"
  ```

where, `token_string` is equivalent to one scope level (for example, `TOP`) or multiple scope levels (for example, `TB/TOP`).

After the macro is defined, it is then used in signal definitions. The syntax to use the macro in the `signal.rc` file is as follows:

```
${macro_name}
```

The following is a list of behaviors that apply when a macro is used:

1. Instances of `macro_name` is replaced with `token_string`.
2. A macro defined in a `signal.rc` file takes priority over the same macro defined as an environment variable.
3. The macro name replacement is case-sensitive.

   For example, a `signal.rc` file contains the following:

```
waveDefine TOP path
```

This macro used as '`addSignal –h 15 ${TOP}/signal`' is identified but '`addSignal –h 15 ${top}/signal`' is not identified.

This command results in a list of unrecognized signals when the signal file is loaded.

4. If a macro is used that does not match a defined macro definition or an environment variable, *nWave* replaces the macro itself without the macro enclosure (`${...}`). For example:

```
addSignal -h 15 -UNSIGNED ${SHOULD_NOT_BE_FOUND}/i_maprom/
addr[7:0]
```

becomes

```
addSignal -h 15 -UNSIGNED SHOULD_NOT_BE_FOUND/i_maprom/
addr[7:0]
```

This command results in a list of unrecognized signals when the signal file is loaded.

5. You can use Macros anywhere in the signal definition. You can also use multiple macros together.

For example, if a signal file contains the following macros:

```
waveDefine TOP tb/top
waveDefine ALUB i_cpu/i_ALUB
```

The following uses are all acceptable:

```
addSignal –h 15 /${TOP}/signal
addSignal –h 15 /${TOP}/${ALUB}/signal1
addSignal –h 15 /${TOP}/${ALUB}/i_alu/signal2
addSignal –h 15 /tb/top/${ALUB}/signal3
```

6. For duplicated macro definitions, the latter definitions replace the original definitions.
7. *nWave* does not generate signal files that automatically contains macro definitions. You must modify the file after it is created by *nWave*. Even if a signal file with macro definitions is loaded and saved to another file, the macro definition is not saved.

For example, if an environment variable is set as follows:

```
setenv SEQ "seq_wrapper/seq"
```

If the original file (*signal.rc*) contains the following:

```
waveDefine TOP tb_test1
```

```
…
addSignal -h 15 /${TOP}/xyz/signal1
addSignal -h 15 /${SEQ}/abc/signal2
```

A new save signal still results in the following:

```
…
addSignal -h 15 /tb_test1/xyz/signal1
addSignal -h 15 /seq_wrapper/seq/abc/signal2
```

Appendix C: nRegister File Format

Overview

nRegister file format consists of a number of sections. Each section contains a number of key/value pairs to describe object properties. Each section typically describes one *nRegister* object. Sections also exist for describing overall *nRegister* options.

NOTE: Section and key names are case-sensitive.

[nRegister.X] Sections

Visible *nRegister* objects are contained in [`nRegister.X`] sections. The numeric sequence number *X* is used to separate object sections. Its value is not important but it must be unique.

Example

```
[nRegister.1]
type = Line
...
[nRegister.2]
type = Rectangle
```

Object Types

Each [`nRegister.X`] section describes one object. The keys for each object depend on the object type. 6 different object types are available:

1. Rectangle
2. Ellipse
3. Paragraph
4. Line
5. Arrow
6. SignalValuePair

Every [`nRegister.X`] section must contain a type key:

```
[nRegister.1]
type = Line
```

The type key determines which keys are interpreted by the object.

The following sections describe each object type and the keys that are interpreted by that object. Default values (when no key exists) are also listed.

Rectangle/Ellipse

Describes a rectangular or elliptical object, that can be solid (filled) or an outline.

Key	Description	Default
type	Identify this object as a Rectangle or an Ellipse.	None
pt1	Lower left corner of object. Represented as an x y pair. Example: 142 -18	None
pt2	Upper right corner of object. Represented as an x y pair. Example: 286 63	None
background	Color of object. Values: See Colors and Fonts in *Appendix B* for color values.	*ID_ORANGE5*
foreground	Unused	*None*
selected	Select state: Values: True, False.	*False*
fill	Solid or unfilled. Values: True, False.	*False (unfilled)*

Example

```
[nRegister.12]
type = Rectangle
pt1 = 367 611
pt2 = 504 659
background = ID_WHITE
foreground = ID_WHITE
selected = True
fill = False
```

Example

```
[nRegister.9]
type = Ellipse
pt1 = 216 686
pt2 = 347 722
background = ID_WHITE
foreground = ID_WHITE
selected = False
```

Paragraph

Describes a region of text.

Key	Description	Default
type	Identify this object as a Paragraph.	None
pt1	Lower left corner of object. Represented as an x y pair. Example: 142 -18	None
pt2	Upper right corner of object. Represented as an x y pair. Example: 286 63	None
background	Background color of edit box. Only used during editing. Values: See Colors and Fonts in *Appendix B* for color values.	*ID_BLACK*
foreground	Foreground color. Color of text. Values: See Colors and Fonts in *Appendix B* for color values.	*ID_ORANGE5*
text	The text to display in this object.	*Empty string*
angle	Text rotation. Values: 0, 90, 180, 270.	*0* *(no orientation)*
cursorPos	Drop position in characters from beginning of string. Used to remember last insertion point. Values: For register file editing set to 0.	*0*
selected	Select state: Values: True, False.	*False*
font	Text font. Values: See Colors and Fonts in *Appendix B* for font values.	*"Helvetica 14"*

Example

```
[nRegister.1]
type = Paragraph
pt1 = 28 823
pt2 = 299 888
background = ID_BLACK
foreground = ID_ORANGE5
selected = False
font = -misc-fixed-medium-r-*-*-14-*-*-*-*-*-iso8859-*
text = TOP LEVEL MODULE\\
\\
 \\
====================================
```

```
cursorPos = 21
angle = 0
```

Line

Describes a line connecting two points. This line does not contain arrowheads.

Key	Description	Default
type	Identify this object as a Line.	None
pt1	Lower left corner of object. Represented as an x y pair. Example: 142 1	None
pt2	Upper right corner of object. Represented as an x y pair. Example: 286 63	None
background	Color of object. Values: See Colors and Fonts in *Appendix B* for color values.	*ID_BLACK*
foreground	Unused	None
selected	Select state: Values: True, False.	*False*
width	Line width in pixels.	*1*

Example

```
[nRegister.11]
type = Line
pt1 = 71 630
pt2 = 258 630
background = ID_WHITE
foreground = ID_WHITE
selected = False
width = 2
```

Arrow

Describes a line with arrowheads connecting two points.

Key	Description	Default
type	Identify this object as an Arrow.	None
pt1	Lower left corner of object. Represented as an x y pair. Example: 142 1	None
pt2	Upper right corner of object. Represented as an x y pair. Example: 286 63	None
background	Color of object. Values: See Colors and Fonts in *Appendix B* for color values.	*ID_BLACK*
foreground	Unused	None
selected	Select state: Values: True, False.	*False*
width	Arrow width in pixels.	*1*
doubleHead	Arrowhead on each end of line. Values: True, False.	*False*

Example

```
[nRegister.10]
type = Arrow
pt1 = 338 817
pt2 = 434 817
background = ID_WHITE
foreground = ID_WHITE
selected = True
width = 2
doubleHead = True
```

SignalValuePair

Describes a signal and its associated value.

Key	Description	Default
type	Identify this object as a SignalValuePair.	None
pt1	Lower left corner of object. Represented as an x y pair. Example: 142 -18	None

pt2	Upper right corner of object. Represented as an x y pair. Example: 286 63	None
background	Unused - reserved for future use.	None
foreground	Unused - reserved for future use.	None
selected	Select state: Values: True, False.	*False*
grow	Auto adjustment of signal value box width. Values: 0 (None), 1 (Fit if width is smaller than signal value text), 2 (Always fit width to signal value text)	*1*
signal_name	Signal name with full hierarchy. Example: system.reset_cpu	None
value_format	Signal value format. Values: Bin, Oct, Hex, Dec, Ascii, Enum	None
divide_xpos	The x coordinate of the separation bar between the signal name and value. Example: 304	None
signal_color	Signal name text color. Values: See Colors and Fonts in *Appendix B* for color values.	*ID_YELLOW5*
value_color	Signal value text color. Values: See Colors and Fonts in *Appendix B* for color values.	*ID_YELLOW5*
border_color	Signal border color. Values: See Colors and Fonts in *Appendix B* for color values.	*ID_YELLOW5*
font	Signal name and value text font. Values: See Colors and Fonts in *Appendix B* for color values.	*"Helvetica 14"*
signal_string	Signal name without full hierarchy. Example: reset_cpu	None
value_string	Signal value. Example: x->1	None
border	Show signal box border. Values: True, False.	*True*
value_shadow	Shadow signal value box. Values: True, False.	*True*
signal_display	Display signal name. Values: True, False.	*True*

angle	Rotation. Values: 0, 90, 180, 270.	*0*
signal_width	Signal name box width. Example: 63	None
value_width	Signal value box width. Example: 7	None
signal_alignment	Signal name text alignment. Values: Left, Right, Center	*Center*
value_alignment	Signal value text alignment. Values: Left, Right, Center	None
time_offset = 0.000000	Value of time offset. Example: 500.000000	*0.000000*

Example

```
[nRegister.2]
type = SignalValuePair
pt1 = 28 750
pt2 = 150 768
background = ID_ORANGE5
foreground = ID_ORANGE5
selected = False
grow = 1
signal_name = system.data[7:0]
value_format = Hex
divide_xpos = 96
signal_color = ID_GREEN4
value_color = ID_RED4
border_color = ID_ORANGE5
font = -misc-fixed-medium-r-*-*-14-*-*-*-*-*-iso8859-*
signal_string = data[7:0]
value_string = XX
border = True
value_shadow = True
signal_display = True
angle = 0
signal_width = 63
value_width = 14
signal_alignment = Left
value_alignment = Left
time_offset = 0.000000
```

[nregDisplayOpt] Sections

This section contains information about *nRegister's* display attribute settings. Some of these attributes are for the entire window display; while some are for signal display.

nRegister Window Display Attributes

The following table describes the display options of the *nRegister* window.

Key	Description	Default
AutoExpand	Specify whether the signal value field width is automatically adjusted or not. Values: True, False.	*True*
HierName	Specify whether the hierarchical name is displayed or not. Values: True, False.	*False*
LeadZero	Specify leading zero flag for signal value display. Values: True, False.	*False*
SearchBy	Specify the search by option. Values: ByChange (Search by any change), ByRising (Search by rising edge), ByFalling (Search by falling edge).	*ByChange*
ShowTransition	Specify whether to show the signal value transition or not. Values: True, False.	*True*
ValueSpace	Insert a space between digits when a signal value is being displayed in hexadecimal format. Values: True, False	*False*
ValueSpaceChar	Replace the value space above with another single character. Example: ","	“_”
ViewBBox	Specify viewing area of this window. For example, ViewBBox = x1 y1 x2 y2 x1 - X coordinate of lower left corner y1 - Y coordinate of lower left corner x2 - X coordinate of upper right corner y2 - Y coordinate of upper right corner	None

WindowSize	Specify window position and size. For example, WindowSize = x y w h x - X coordinate of upper left corner y - Y coordinate of upper left corner w - Width of this window h - Height of this window	None

[nregMacro.X] Sections

Signal variables in text widgets are contained in [`nregMacro.X`] sections. The numeric sequence number *X* is used to separate variable macro sections. Its value is not important but it must be unique.

Example

```
[nregMacro.0]
name = $1
def = $Val(system.VMA,Hex,Unsigned)
[nregMacro.1]
name = $2
def = $Val(system.addr[7:0],Hex,Unsigned)
...
```

Signal Variable Attributes in Text Widgets

The following table describes macro attributes.

Key	Description	Default
def	Specify macro definition including signal name, display format, and notation.	None
name	Specify macro name as a variable.	None

Appendix D: nCompare File and Comparison Formats

Rule File

This section describes the syntax and semantics of the *nCompare* rule file.

nCompare reads all commands in the rule file (`*.ncr`) in advance and then completes the comparison sequentially.

The *nCompare* module uses Tcl language for rule file formatting. The rule file is used to describe how a comparison is performed.

Syntax Conventions

The following conventions are used in the command syntax:

- `[]`: indicates the object within is optional
- `(a | b)`: indicates that either `a` or `b` is allowable
- `(a)+`: `+` indicates that `a` can appear at least once
- `(a)*`: `*` indicates that `a` can either not appear or appears many times
- `%f`: indicates a floating number
- `%b`: indicates a boolean value, `T` or `F`
- `%d`: indicates a decimal integer number
- `%s`: indicates a string

NOTE: All commands are case-sensitive. All options are case-insensitive.

Commands

cmpChkStability

Description

Start time-based comparison to check if compared signal pairs are stable while the trigger condition is occurring. This command only checks stability based on

current comparing triggers and does not perform waveform comparison. The meanings of these options are the same as those in `cmpSetSignalPair`. The check is not performed until the `cmpCompare` command is issued. If all triggers are disabled at this point, a warning is reported indicating that the *nCompare* module ignores the `cmpChkStability` command.

Syntax

```
cmpChkStability [GOLDEN | SECONDARY] SignalOrModuleName
[-level %d] [-type SignalType]
```

Arguments

`GOLDEN | SECONDARY`

Specify whether the signal or module is in the GOLDEN or SECONDARY FSDB file. If these argument values are omitted, the setting is applied to both waveforms.

`SignalOrModuleName`

Specify the signal or module name for the stability check.

`-level %d`

Specify the hierarchy level limit for the module to be compared. Refer to the option [`-level %d`] of `cmpSetSignalPair` for details.

`-type SignalType`

Specify the type of signals in the modules that is compared. Refer to the option [`-type SignalType`] of `cmpSetSignalPair` for details.

Value Returned

TRUE if successful; otherwise returns FALSE.

Example

```
cmpOpenFsdb rtl.fsdb gate.fsdb
cmpSetSignalPair top -level 0
cmpSetCmpTrigger GOLDEN -Clock system.clock
cmpChkStability
cmpCompare
```

Check the stability of all signals of `rtl.fsdb` and `gate.fsdb`.

cmpCompare

Description

Start time-based comparison to compare all signal pairs specified by `cmpSetSignalPair`.

Multiple `cmpCompare` commands can be specified in a rule file (*.ncr). This can improve memory performance. Only commands immediately preceding a `cmpCompare` command from the beginning of a file or from another `cmpCompare` command are used by the `cmpCompare` command.

Syntax

```
cmpCompare [-diff]
```

Argument

`-diff`

When the `-diff` option is turned on, the comparison is performed in the style of UNIX diff and the mismatches are generated as an output in the format of diff style too. The format is as follows:

s_time1 to s_time2 s_signal has more changes than g_signal from g_time1 to g_time2

s_time1 to s_time2 s_signal has less changes than g_signal from g_time1 to g_time2

s_time1 to s_time2 s_signal differs from g_signal from g_time1 to g_time2

NOTE: Comparing in diff style causes the tolerance window to be ignored. If the time range is not specified, the respective time range of golden and secondary FSDB files are used for `-diff` mode comparison.

Value Returned

The number of mismatch errors if successful; otherwise returns -1.

Example

```
cmpOpenFsdb rtl.fsdb gate.fsdb
cmpSetSignalPair top -level 0
cmpCompare
```

Compare all signals of `rtl.fsdb` and `gate.fsdb`.

cmpGenShift

Description

Automatically generate the shift value for the secondary waveform based on the options defined by `cmpSetCmpTrigger` and `cmpSetCmpOption`. The returned value can be used in `cmpSetSignalPair` as the shift value and

minimizes the mismatch error between the golden and the secondary signal based on the current options.

Syntax

```
cmpGenShift GoldenSignal [SecondarySignal] [-max %f[Unit]]
[(-pos | -neg)]
```

Arguments

`GoldenSignal [SecondarySignal]`

Specify the signal name in the golden FSDB file. Specify the signal name in the secondary FSDB file if the signal name is different.

`-max %f[Unit]`

`-max` is the maximum of the absolute value of the time offset. If the `-Max` option is not defined, its default is *100* and the unit is the time unit defined by `cmpSetTimeUnit`. The default time unit is the time unit defined by `cmpSetTimeUnit`.

`-pos | -neg`

`-pos` and `-neg` indicates the direction in which to search for the shift value. `-pos` indicates that the shift value should be positive. `-neg` indicates that the shift value should be negative. If the direction is not specified, the shift value is searched for in both directions.

Value Returned

The shift value and its time unit if successful; otherwise an empty string.

Example

```
set MyShift [cmpGenShift top.clk -max 50 –pos]
cmpSetSignalPair top -level 0 -shift $MyShift
cmpCompare
```

cmpGetCmpOption

Description

Set options for comparison.

Syntax

```
cmpGetCmpOption [-glitch TclVariableName]
[-strength TclVariableName] [-x TclVariableName]
[-z TclVariableName] [-maxerror TclVariableName]
[-maxerrorpersignal TclVariableName] [-tolerance TclVariableName]
```

```
[-valuetolerance TclVariableName] [-time TclVariableName]
[-ignoreWidth TclVariableName]
```

Arguments

`-glitch TclVariableName`

Store the glitch flag into a `TclVariableName`. The value is T or F.

`-ignoreWidth TclVariableName`

Store the option value of `-IgnoreWidth` into `TclVariableName`. The value is a non-negative integer number.

`-maxerror TclVariableName`

Store the maximum number of mismatches in a comparison into a `TclVariableName`. The value is a non-negative integer number.

`-maxerrorpersignal TclVariableName`

Store the maximum number of mismatches of each pair of signals into a `TclVariableName`. The value is a non-negative integer number.

`-strength TclVariableName`

Store the strength flag into a `TclVariableName`. The value is T or F.

`-time TclVariableName`

Store the time range of comparison into a `TclVariableName`. The value is a list. Each element of the list is still a list, the first element of which is the start time of a section of time range and the second of which is the end time of a section of time range. Each element has no unit and takes the time unit defined by `cmpSetTimeUnit`.

`-tolerance TclVariableName`

Store the tolerance window into a `TclVariableName`. The value is a list. The first element of the list indicates the negative tolerance and its second element indicates the positive tolerance. Both of these elements are non-negative. They have no unit and take the time unit defined by `cmpSetTimeUnit`.

`-valuetolerance TclVariableName`

Store the value tolerance into a `TclVariableName` variable. The value is a non-negative number. It has no unit and takes the time unit defined by `cmpSetTimeUnit`.

`-x TclVariableName`

Store the X flag into a `TclVariableName`. The value is T or F.

`-z TclVariableName`

Store the Z flag into a `TclVariableName`. The value is T or F.

Value Returned

TRUE if successful; otherwise returns FALSE.

Example

```
cmpGetCmpOption -glitch GlitchFlag -strength StrengthFlag \
   -x X_Flag -z Z_Flag -time TimeRange -maxError TotalError \
   -maxerrorpersignal ErrorPerSignal -tolerance Tol  \
   -valueTolerance ValTol
```

Displays `-Glitch is $GlitchFlag\n`
Displays `-Strength is $StrengthFlag\n`
Displays `-X is $X_Flag\n`
Displays `-Z is $Z_Flag\n`
Displays `-MaxError is $TotalError\n`
Displays `-MaxErrorPerSignal is $ErrorPerSignal\n`
Displays `-Tolerance is [lindex $Tol 0] [lindex $Tol 1]\n`
Displays `-ValueTolerance is $ValTol\n`
Displays The time range of the comparison `is:\n`

```
set Len [llength $TimeRange]
for {set i 0} {$i < $Len} {incr i 1} {
set Sect [lindex $TimeRange $i]
```

Displays Section `$i: from [lindex $Sect 0]` to `[lindex $Sect 1]\n}`

cmpGetDelimiter

Description

Get the default scope delimiter.

Syntax

```
cmpGetDelimiter
```

Value Returned

TRUE if successful; otherwise returns FALSE.

Example

```
set CurDelimiter [cmpGetDelimiter]
```

Displays `Current default scope delimiter is $CurDelimiter`.

cmpGetFsdb

Description
Get the file name of the opened golden or secondary waveform.

Syntax
```
cmpGetFsdb (GOLDEN | SECONDARY)
```

Arguments
`GOLDEN | SECONDARY`

`GOLDEN` returns the opened golden waveform file name and `SECONDARY` returns the secondary waveform file name.

Value Returned
The file name of the golden or secondary waveform.

Examples
```
set SecondaryFsdb [cmpGetFsdb SECONDARY]
```
Displays `Secondary waveform file name is $SecondaryFsdb`.

cmpGetReport

Description
Get the defined file name of the error report file.

Syntax
```
cmpGetReport
```

Value Returned
The defined file name of the error report file.

Examples
```
Set CurReportFileName [cmpGetReport]
```
Displays `Current error report file name is $CurReportFileName`.

cmpGetScopePath

Description
Get the defined default scope paths for the golden or secondary waveforms.

Syntax
```
cmpGetScopePath (GOLDEN | SECONDARY)
```

Argument
```
GOLDEN | SECONDARY
```
GOLDEN represents the golden waveform that the scope path is applied to; and SECONDARY represents the secondary waveform.

Value Returned
The scope path for golden or secondary waveforms.

Example
```
set GoldenScopePath [cmpGetScopePath GOLDEN]
```
Displays `Golden scope path is $GoldenScopePath`.

cmpGetTimeScale

Description
Get the default time scale from the *GOLDEN* or *SECONDARY* waveform file. This is similar to *cmpGetTimeUnit* except the source must be specified.

Syntax
```
cmpGetTimeScale [GOLDEN | SECONDARY]
```

Argument
```
GOLDEN | SECONDARY
```
GOLDEN represents the golden waveform; and SECONDARY represents the secondary waveform.

Value Returned
A list with time and unit. The unit can be Gs | Ms | Ks | s | ms | us | ns | ps | fs.

Examples
```
cmpGetTimeScale SECONDARY
```

Gets the time scale from the *SECONDARY* waveform file.

```
set CurGoldenTimeScale [cmpGetTimeScale GOLDEN]
```

Puts the current golden time scale value in the `$CurGoldenTimeScale` variable .

cmpGetTimeUnit

Description

Get the default time unit.

Syntax

```
cmpGetTimeUnit
```

Value Returned

A list. The first element of the list is a floating number. The second is the unit string. The unit can be `Gs` | `Ms` | `Ks` | `s` | `ms` | `us` | `ns` | `ps` | `fs`.

Example

```
set CurTimeUnit [cmpGetTimeUnit]
```

Displays `Current time unit is [lindex $CurTimeUnit 0] [lindex $CurTimeUnit 1]`.

cmpNewSignal

Description

Generate a new vector signal formed by several signals.

NOTE: Escaped double quotes (\"\") need to be added for the escaped signal and two-dimensional signal names. For example,

```
set s1 [cmpNewSignal Golden -exp \"top/
\\c\[1\]\" >>1]
set g1 [cmpNewSignal Golden -exp \"/top/
a\[1\]\[1\]\" >>1 ]
```

Syntax

```
cmpNewSignal [GOLDEN | SECONDARY] SignalName, ... [,SignalName]
| -exp LogicExp]
```

Arguments

`GOLDEN | SECONDARY`

Specify whether the signal is in the GOLDEN or SECONDARY FSDB file.

`SignalName`

Specify one or more signal names to group for comparison. Multiple signal names can be separated by a comma (,) character.

Value Returned

The name of the new signal if successful; otherwise returns an empty string. A new vector signal is created from its highest bit to its lowest bit by the specified `SignalName` order.

Example

```
set MyNewBus [cmpNewSignal GOLDEN a, b, c]
cmpSetSignalPair $MyNewBus {d[2:0]}
cmpCompare
```

Compare `{a, b, c}` with `d[2:0]`.

cmpOpenFsdb

Description

Open the golden and secondary waveform files and close the current opened waveform files. If the default time unit is not previously set in `cmpSetTimeUnit`, this command sets the default time unit the same as the opened golden FSDB.

NOTE: The **cmpSetTimeUnit**, **cmpSetTimeScale**,and **cmpToFsdb** commands must be set before the **cmpOpenFsdb** command.

Syntax

```
cmpOpenFsdb GoldenFSDB [SecondaryFSDB]
```

Arguments

`GoldenFSDB [SecondaryFSDB]`

If the secondary waveform file name is not specified, the default secondary waveform file name is the golden waveform file name.

After this command is executed, the current comparison options that are associated with waveforms are set to default. These options include:

a. Options that are set by the `cmpSetCmpTrigger` command.

b. The tolerance window options and comparing time range options that are set by the `cmpSetCmpOption` command.

All other options are reserved.

Value Returned

TRUE if successful; otherwise returns FALSE.

Example

```
cmpOpenFsdb rtl.fsdb gate.fsdb
cmpSetSignalPair top -level 0
cmpCompare
```

Compare all signals of `rtl.fsdb` and `gate.fsdb`.

cmpSetADThreshold

Description

Specify the threshold value for translating the analog value to digital for future comparison.

Syntax

```
cmpSetADThreshold [-datatype (VERILOG|VHDL|EVerilog)]
[( '( ' [[=] %f], [[=]%f], %v ')' )+]
```

Arguments

```
-datatype (VERILOG|VHDL|EVerilog)
[( '( ' [[=] %f], [[=]%f], %v ')' )] ...
```

Indicates one signal type from Verilog, VHDL, or EVerilog that this command is applied to. The default is *Verilog*. If the `cmpSetADThreshold` command is specified twice, the thresholds defined by the latter overwrites those defined by the former.

For Verilog, `%v` is a logic value of Verilog with strength attributes. The format is `[strength0][strength1][0|1|x|z]`, where `strength0` and `strength1` can have one of the following 8 values:

Value	Meaning
`Su`	vclSupply
`St`	vclStrong

Pu	vclPull
La	vclLarge
We	vclWeak
Me	vclMedium
Sm	vclSmall
HiZ	vclHighZ

For VHDL, %v can be one of the following logic values:

Value	Meaning
U	Uninitialized
X	Forcing unknown
0	Forcing 0
1	Forcing 1
Z	High impedance
W	Weak unknown
L	Weak 0
H	Weak 1
-	Don’t care
^	Special format
V	Special format
i	Special format

For EVerilog, %v format is [strength0] [strength1] [L|l|H|h|X|T|D|d|U|u|N|Z|?|0|1|A|a|B|b|C|c|F|f], where strength0 and strength1 can have one of the following 8 values:

Value	Meaning
Su	vclSupply
St	vclStrong
Pu	vclPull
La	vclLarge
We	vclWeak
Me	vclMedium
Sm	vclSmall

HiZ	vclHighZ

`L|l|H|h|X|T|D|d|U|u|N|Z|?|0|1|A|a|B|b|C|c|F|f` refers to the following port values:

Value	Meaning
L	Low
l	Low (two or more drivers active)
H	High
h	High (two or more drivers active)
X	Unknown (Don’t care)
T	Three-state
D	Low
d	Low (two or more drivers active)
U	High
u	High
N	Unknown
Z	Three-state
?	Unknown
0	Low (both input and output are active with 0 value)
1	High (both input and output are active with 1 value)
A	Unknown (input 0 and output 1)
a	Unknown (input 0 and output X)
B	Unknown (input 1 and output 0)
b	Unknown (input 1 and output X)
C	Unknown (input X and output 0)
c	Unknown (input X and output 1)
F	Three-state (input and output unconnected)
f	Unknown (input and output three-stated)

Two floating numbers define a threshold. The first specifies the lower bound and the second specifies the upper bound. If = is added before the bound, this indicates the threshold includes the bound; otherwise, the threshold excludes the bound. If the first number is omitted, it indicates that the lower bound is a negative maximum. If the second number is omitted, it indicates that the upper bound is a positive maximum. For each floating

value within the threshold, it maps to the corresponding digital value. If a floating value does not belong to any defined threshold, it maps to SuSux for Verilog, - for VHDL, X for SuSu? for EVerilog respectively.

If no arguments exist, all thresholds defined before are cleared up. While comparing analog signals, if no threshold has been specified, their analog values are compared directly.

For Verilog, the meaning of strength0 and strength1 is dependent upon the logic value as shown in the following table:

Logic Value	Strength0	Strength1
0	Min	Max
1	Min	Max
x	Strength of logic 1	Strength of logic 0
z	HiZ	HiZ

If both strength0 and strength1 are omitted, the default strengths are listed as shown in the following table:

Logic Value	Strength0	Strength1
0	Su	Su
1	Su	Su
x	Su	Su
z	HiZ	HiZ

If only strength0 is omitted, strength0 is set to the same value as strength1. If only strength1 is omitted, strength1 is set to the same value as strength0.

If a threshold overlaps with another threshold, an error is reported.If the specified strength values do not satisfy the following rules, errors are reported.

1. For logic value z, both strength0 and strength1 can only be HiZ;
2. For logic value 0 or 1, strength0 cannot be greater than strength1.

For EVerilog, if both strength0 and strength1 are omitted, the default strengths are listed as shown in the following table:

Input Ports	Output Ports	Inout Ports

D (St Hi)	L (St Hi)	1 (Hi St)
U (Hi St)	H (Hi St)	0 (St Hi)
N (St St)	X (St St)	? (St St)
Z (Hi Hi)	T (Hi Hi)	F (Hi Hi)
d (St Hi)	l (St Hi)	A (St St)
u (Hi St)	h (Hi St)	a (St St)
		B (St St)
		b (St St)
		C (St St)
		c (St St)
		f (Hi Hi)

If only `strength0` is omitted, `strength0` is set to the same value as `strength1`. If only `strength1` is omitted, `strength1` is set to the same value as `strength0`.

Value Returned

TRUE if successful; otherwise returns FALSE.

Examples

```
cmpSetADThreshold (=2.0,3.5,0) (=3.5,5.0,1)
```

>=2.0 && <3.5 are SuSu0; >=3.5 && <5.0 are SuSu1; others are SuSux.

```
cmpSetADThreshold (=2.0,3.5,HiZSt0) (=3.5,5.0,Su0)
(=5.0,=8.0,HiZSu1) (,2.0,z)
```

>=2.0 && <3.5 are HiZSt0; >=3.5 && <5.0 are SuSu0;
>=5.0 && <=8.0 are HiZSu1; <2.0 are HiZHiZz; others are SuSux.

```
cmpSetADThreshold -datatype VHDL (=2.0,3.5,0) (=3.5,5.0,H)
(=5.0,=8.0,1)
```

>=2.0 && <3.5 are forcing 0; >=3.5 && < 5.0 are weak 1;
>=5.0 && <=8.0 are forcing 1; others don't care.

cmpSetCmpOption

Description

Set options for comparison.

NOTE: When the `cmpSetCmpOption` command is called without any argument or the `cmpCompare` command is invoked, the value of `cmpSetCmpOption` is inherited in the compilation.

Syntax

```
cmpSetCmpOption [-nocase] [-glitch T | F] [-strength T | F]
[-x T | F][-z T | F] [-maxerror %d] [-maxerrorpersignal %d]
[-tolerance %f[Unit] [%f[Unit]]] [-valuetolerance %f[Unit]]
[-time ([%f[Unit]], [%f[Unit]])] [-IgnoreWidth %f[Unit]]
[-datatype (VERILOG|VHDL|EVerilog) (VERILOG|VHDL|EVerilog)]
[-outputradix bin|hex|oct|dec|udec|ascii] [-IgnoreTriggerNotFound
T | F] [-IgnoreSizeCheck T | F] [-nc T | F]
```

Arguments

`-datatype (VERILOG|VHDL|EVerilog)(VERILOG|VHDL|EVerilog)`

The first value indicates the data type of the golden signals from Verilog, VHDL, or EVerilog; the second value indicates the data type of the secondary signals. If the second value is omitted, the first one is applied to both golden and secondary signals. The default is *Verilog*.

NOTE: If the data type in the golden or the secondary file is unknown or mixed, then it is highly recommended not to specify `-datatype` in the `cmpSetCmpOption` rule command as the *nCompare* module automatically selects the data type for comparison.

`-nocase`

When included, the mapping of signal names is not case-sensitive. Use this option in conjunction with *-s*.

`-glitch T | F`

Determine whether the *nCompare* module compares glitch events or not. The default is *F*.

`-IgnoreSizeCheck T | F`

default is F (off).

Determine whether the *nCompare* module ignores size checking for struct or array, and then compare common members between the golden and second signal. The default is *F*.

Example:

```
Golden fsdb array_begin A L:0, R:5 (s=2) A[2], A[5] array_end
{ V2, V5 }
Second fsdb array_begin A L:0, R:5 (s=3) A[2], A[3], A[5]
array_end { V2, V3, V5 }
=> With the option turned off, signal A comparison is
skipped. With the option turned on, comparison is performed
for both A[2] and A[5].
```

`-IgnoreTriggerNotFound T | F`

Determine whether the *nCompare* module stops when a trigger signal is not found. When the option is set as *T*, *nCompare* module ignores the related compared pairs if a trigger signal is not found. When this option is set as *F*, *nCompare* module stops when a trigger signal is not found. The default is *F*.

`-IgnoreWidth %f[Unit]`

Define the minimum width of the reported mismatch. All mismatches with width smaller than `IgnoreWidth` are ignored.

`-maxerror %d`

Specify the maximum number of mismatches in a comparison. Zero indicates 200000. This option stops reporting the mismatches when the total reported error reaches the maximum. The default is *1000*.

NOTE: The *nCompare* module does not always follow the value specified for the maximum mismatch. More mismatches than the specified mismatches may be reported.

`-maxerrorpersignal %d`

Specify the maximum number of mismatches for each pair of signals. Zero indicates 200000. When the maximum is reached, the remaining pairs of signals are not compared. The default is *100*.

`-nc T | F`

Determine whether the *nCompare* module compares NC (Not Computed) value or not. The default is *T*.

`-outputradix bin|hex|oct|dec|udec|ascii`

Specify the radix for the output signal value. The available radix includes bin (binary), hex (hexadecimal), oct (octal), dec (signed decimal), udec (unsigned decimal), and ascii (ASCII text string). The default is *binary*.

`-strength T | F`

Determine whether the *nCompare* module compares strength or not. The default is *F*.

`-time ([%f[Unit]], [%f[Unit]])`

Specify one or more time ranges for comparison. A pair of numbers (`%f[Unit], %f[Unit]`) specify a section of the time range. The first number is the start time of the section. If it is omitted, it is regarded as 0. The second number is the end time of the section. If it is omitted, it is regarded as the end time of the golden waveform. This range is for the golden waveform. For the secondary waveform, the range should be adjusted by the offset N, where N is the shift value defined in `cmpSetSignalPair -shift`. The default is the full range of the golden waveform. The default time unit of `-tolerance` and `-time` is the time unit defined by `cmpSetTimeUnit`.

`-tolerance %f[Unit] [%f[Unit]]`

Define the tolerance window (a time section) around each event of the golden signal. A match is when the secondary signal has the same value as the golden signal plus the tolerance window. This option is effective only when all trigger conditions are disabled. The default is *0*.

The first argument specifies the negative tolerance (the offset to a golden event in the negative direction) and the second argument specifies the positive tolerance (the offset to a golden event in the positive direction). If the second argument is omitted, it indicates that the positive tolerance is the same as the negative tolerance. Each argument is non-negative.

The principles of the tolerance window of the *nCompare* module can be summarized as follows:

a. An edge of the golden signal may only match the first to be compared edge of the secondary signal in its tolerance window.

b. If the tolerance window is defined, when a mismatch takes place at the edge of a signal (golden or secondary), the *nCompare* module first finds the golden edge to be compared (which may be the same edge causing the mismatch). Then check if the first edge to be compared of the secondary signal is within its tolerance window. If it is and the values at the two edges are identical, then no mismatch error is reported. Otherwise, a mismatch error is reported.

`-valuetolerance %f[Unit]`

Define the value tolerance of the comparing pair. A match is made when the secondary signal has the same value as the golden signal plus (or minus) the value tolerance. The default is *0.0* (no value tolerance).

The value tolerance setting applies to all the analog signal pairs following the `cmpSetCmpOption` command. To compare analog signal pairs with different value tolerances, specify a different value tolerance for different

signal pairs by repeatedly setting the `cmpSetCmpOption` and `cmpSetSignalPair` commands.

For Analog FSDB file (such as EPIC format) comparison, specify the data type with the `-datatype` option in the `cmpSetCmpOption` command. Otherwise, the compared results are not correct.

`-x T | F`

Determine whether the *nCompare* module compares 'X' or not. The default is *F*.

`-z T | F`

Determine whether the *nCompare* module compares 'Z' or not. The default is *F*.

Value Returned

TRUE if successful; otherwise returns FALSE.

Examples

```
cmpSetCmpOption -maxerror 500 –maxerrorpersignal 20 \
   -glitch T -strength T -x T -z F -tolerance 10, 20 \
   -time (, 100) (300, 1000) (1500,)
```

```
cmpSetSignalPair system.DATA
```

Even if only system.data exists in the FSDB, `cmpSetSignalPair` still finds it.

```
cmpSetNameMap %s.DATA %1s.DATA, %s.ADDR %1s.ADDR
```

The first mapping rule only maps system.data to system.data. The second maps `system.addr` to `system.ADDR`.

```
cmpSetCmpOption -valuetolerance 0.5 -datatype VERILOG VERILOG
cmpSetCmpOption -datatype VERILOG VERILOG -NC F
```

cmpSetCmpTrigger

Description

Specify a comparison trigger object for comparison.

The specified triggers are used with the following `cmpSetSignalPair` commands (can be multiple) until a new `cmpSetCmpTrigger` command is specified.

Syntax

```
cmpSetCmpTrigger [GOLDEN | SECONDARY]
[-clock [[G: | S:] ClkName [-setup %f[Unit]] [-hold %f[Unit]]
[-trigger (F | R | B)] [-sample %f[Unit]]]] [-cycle [%f[Unit]
[-delay %f[Unit]] [-width %f[Unit]]]]
[-when [[G: | S:] {LogicExp}]]
```

Arguments

`GOLDEN | SECONDARY`

GOLDEN represents the golden waveform that the setting is applied to and SECONDARY represents the secondary waveform. If these argument values are omitted, the setting is applied to both waveforms.

```
[-clock [[G: | S:] ClkName [-setup %f[Unit]] [-hold %f[Unit]]
[-trigger (F | R | B)] [-sample %f[Unit]]]] [-cycle [%f[Unit]
[-delay %f[Unit]] [-width %f[Unit]]]] [-when [[G: | S:]
{LogicExp}]]
```

For the `-trigger` option, `R` indicates a rising edge, `F` indicates a falling edge and `B` indicates both edges. When `ClkName` is defined, `-setup` and `-hold` check (stability check) is performed; `-sample` specifies the time offset from the clock, where the actual sample, `-setup` and `-hold` checks (stability check) should be performed.

If additional options are not specified with the -clock option, the `-clock` options are turned off. `-cycle` and `-when` are similar. If `-clock`, `-cycle,` and `-when` are omitted, all the options defined before are turned off.

The types of triggers for golden signals may be different from those for secondary signals. However, golden (secondary) signals cannot have some triggers enabled if secondary (golden) signals have no triggers enabled.

When an FSDB selector is specified after the `-clock` option, `ClkName` belongs to the specified FSDB instead of the FSDB that the trigger is used for. If no FSDB selector is specified after the `-clock` option, `ClkName` still belongs to the FSDB that the trigger is used for. It is similar for the `-When` option. Whichever FSDB the trigger signals belong to, they use the predefined scope path for the FSDB if the leading symbol of their names is not the at (@) character.

Besides the clock trigger, each trigger defines one strobe window to check stability of the signals to be compared. The clock trigger defines two strobe windows: setup and hold. A `cmpSetCmpTrigger` can define multiple triggers, namely multiple strobe windows, and a stability check must be performed in all the windows.

By default, the `-clock`, `-cycle`, and `-when` options are turned *off* so they have no effect. If `ClkName` is defined, the default of `-setup` is *20*, `-hold` is *20*, trigger is *R*, and `-sample` is *0*. If `-cycle` is defined, the default of delay is *0* and width is *50*.

NOTE: Escaped double quotes (\"\") need to be added for the escaped signal and two-dimensional signal names. For example,

```
cmpSetCmpTrigger -When \"top/\\c\[1\]\"
cmpSetCmpTrigger -When \"/top/a\[1\]\[1\]\"
```

The `-cycle` and `-clock` command options cannot be specified simultaneously with `cmpSetCmpTrigger`.

The `-when` command option does not support stability checking for `cmpSetCmpTrigger`.

The `LogicExp` must be a Verilog logic expression whose value is TRUE or FALSE. Signal names must be enclosed by double quotes ("") or precede an escaped character \ of special characters, including \ ' " $ { } [and]. The legal operands of `LogicExp` include integer constants and signals (or slices) in golden or secondary waveforms. The legal opcodes of `LogicExp` contain:

~ : bit-wise negation
& : bit-wise and
| : bit-wise or
^ : bit-wise xor
~^ : bit-wise xnor
+ : addition
- : subtraction
<< : left shift
>> : right shift
! : logical negation
&& : logical and
|| : logical or
== : logical equality
!= : logical inequality
< : less
> : greater
<= : less or equal
>= : greater or equal
: time offset

Value Returned

TRUE if successful; otherwise returns FALSE.

Examples

```
cmpSetCmpTrigger -when #30 (system.reset && #-40 system.clock)
```

A comparison is triggered at time T if *system.reset* is 1 at time (T - 30) and *system.clock* is 1 at time (T + 10).

```
cmpSetCmpTrigger -when #30ps system.reset
```

A comparison is triggered at time T if *system.reset* is 1 at time (T - 30ps).

```
cmpSetCmpTrigger GOLDEN -clock system.clock
cmpSetCmpTrigger SECONDARY -cycle 100
```

Golden and secondary can have different trigger types.

```
cmpSetScopePath system system.i_cpu
cmpSetCmpTrigger GOLDEN -clock S:clock
```

The clock trigger for the golden FSDB is using the *system.i_cpu.clock* signal in the secondary FSDB

```
cmpSetScopePath system system.i_cpu
cmpSetCmpTrigger GOLDEN -clock clock
```

The clock trigger for golden FSDB is using the *system.clock* signal in the golden FSDB

```
cmpSetCmpTrigger SECONDARY -when G: \
   {"reset" & "@system.i_pram.R_W"}
```

The expression trigger for the secondary FSDB contains signal *system.reset* in the secondary FSDB and *system.i_pram.R_W* which belongs to the golden FSDB.

```
cmpSetCmpTrigger -when {"reset"}
```

The expression trigger for the golden FSDB is the *system.reset* signal in the golden FSDB, and the expression trigger for the secondary FSDB is the *system.i_cpu.reset* signal in the secondary FSDB.

```
cmpSetCmpTrigger -clock "system.clock" -sample 10 -trigger F
```

This indicates the compare trigger for both the Golden and Secondary FSDB file references the *system.clock* in its respective FSDB file and samples 10ns after the falling edge. Specifically, the Golden compare strobe time is 10ns after the falling edge of *system.clock* in the Golden FSDB file.

The Secondary compare strobe time is 10ns after the falling edge of *system.clock* in the Secondary FSDB file.

NOTE: The rule needs to be modified as follows if the strobe time needs to reference the same clock signal (such as *system.clock* in the Golden FSDB file).

```
cmpSetCmpTrigger -clock G: "system.clock" -sample 10 -trigger F
```

NOTE: EVCD signals can be used as the clock.

cmpSetDelimiter

Description

Set the default scope delimiter.

Syntax

```
cmpSetDelimiter . | /
```

Argument

`. | /`

The default scope delimiter is the period (.) character.

Value Returned

TRUE if successful; otherwise returns FALSE.

Example

```
cmpSetDelimiter /
cmpSetSignalPair system/i_cpu/data # compare system.i_cpu.data
```

cmpSetNameMap

Description

Specify the mapping rules between different signal names. Multiple signal name pairs may be specified.

NOTE: For MDA signals, this command can only expand the top-most dimension instead of all dimensions.

Syntax

```
cmpSetNameMap [GoldenNamePattern SecondaryNamePattern [-nocase],]
... [GoldenNamePattern SecondaryNamePattern [-nocase],]
```

Arguments

`GoldenNamePattern SecondaryNamePattern`

`GoldenNamePattern` specifies which signals (modules) of the golden waveform the rule match; `SecondaryNamePattern` specifies which signals (modules) of the secondary waveform the rule match. They have the following features:

They can contain the following variable types: (these variable names are case-sensitive)

d: indicates an integer
c: indicates a character
s: indicates a string
h: indicates a hierarchy

When these variable types are referenced in the pattern, they should have a leading symbol, %, such as `/system/%s`.

In `SecondaryNamePattern`, `%nt` is allowed, where n is a positive integer and t is a variable type defined as above. `%nt` represents the n^{th} `%t` field of a signal (module) name, which matches `GoldenNamePattern`. Suppose `GoldenNamePattern` is `m%spp%s`, and `SecondaryNamePattern` is `h%1stt%2s`. Then the string "mapping" matches `GoldenNamePattern` and its first `%s` field is "a" and its second `%s` field is "ing", so the match for `SecondaryNamePattern2` is "hatting".

`%(expression)` is also allowed in `SecondaryNamePattern`. Operands of expressions can be constant or `%d`. Opcodes of expressions can be "+", "-", "*", "/", "%", "(" and ")".

Suppose an expression is `%1d*3+%2d`. If `%1d` represents 2 and `%2d` represents 1, then `%(%1d*3+%2d)` represents 7.

% is also used as an escaped character in both `GoldenNamePattern` and `SecondaryNamePattern` except in `%(expression)`. So, `%%` indicates one % in both `GoldenNamePattern` and `SecondaryNamePattern` except in `%(expression)`. For a scope delimiter defined by `cmpSetDelimiter`, if it is referenced as a common character in `GoldenNamePattern` or `SecondaryNamePattern`, it must follow a % immediately.

Suppose, the period (.) character is the path delimiter. To represent a common period, %. must be used.

When a signal (module) in the golden waveform cannot map to a signal (module) in the secondary waveform based on the defined mapping rules (the checking order takes the definition order of those rules in `cmpSetNameMap`), it maps to the signal (module) of the same full name or local name in secondary waveform.

The rules defined by a `cmpSetNameMap` is effective until they come across another `cmpSetNameMap`. If a `cmpSetNameMap` has no argument, it clears all name-mapping rules.

`-nocase`

Match the signal name without considering the state of the `cmpSetCmpOption -nocase [T|F]` command.

This option impacts all signals in the `cmpSetSignalPair`, `cmpGenShift`, `cmpSetCmpTrigger`, `cmpSetScopePath` and `cmpSetNameMap` commands. The case matching type of the `cmpSetNameMap` command is also controlled by its `-nocase` option.

Value Returned

TRUE if successful; otherwise returns FALSE.

Examples

```
cmpSetNameMap {tb/\%s.%s /%s} {tb/_%1s_%2s_/%3s} -NoCase , {top/
\%s.%s /%s} {top/_%1s_%2s_/%3s} -NoCase
```

cmpSetNameMap can set multiple pairs of name maps.

```
cmpSetNameMap {top.%s[%d]} {top.\%1s[%1d] }
```

"top.addr[7]" maps to "top.\addr[7] "

```
cmpSetNameMap addr[%d] r_addr[%(7-%1d)]
```

"addr[0]" maps to "r_addr[7]"
"addr[1]" maps to "r_addr[6]"
...
"addr[7]" maps to "r_addr[0]"

```
cmpSetNameMap %s.%s.%s.a %1s.%2s_%3s.a
```

"top.cpu.alu.a" maps to "top.cpu_alu.a"

```
cmpSetNameMap %h.clk %1h.clk2
```

"*.clk" maps to "*.clk2"

cmpSetPref

Description
Set preference options in *nCompare*.

Syntax
```
cmpSetPref [-showByBottomUp on|off]
```

Argument
`-showByBottomUp on|off`

Specify whether to show mismatches in bottom-up order when viewing by time. The default is *off*.

Value Returned
TRUE if successful; otherwise returns FALSE.

Example
```
cmpSetPref -showByBottomUp on
```

All mismatches are shown in bottom-up order.

cmpSetReport

Description
Specify the file to store the comparison error messages to.

Syntax
```
cmpSetReport ErrorReportFileName
```

Argument
`ErrorReportFileName`

The default extension is *.nce*. The error report file can only be defined once in a rule file. For more information on the error report file format, refer to the *Error File* section.

Value Returned
TRUE if successful; otherwise returns FALSE.

Example
```
cmpSetReport cpu_case1.nce
```

All mismatch error messages are generated as output to `cpu_case1.nce`.

cmpSetScopePath

Description

Set the default scope paths for signals or scopes.

Syntax

```
cmpSetScopePath [GoldenPath [SecondaryPath]]
```

Argument

`GoldenPath [SecondaryPath]`

If no arguments exist, the default scope paths are cleared. If only one argument is specified, it is used for both the golden and secondary waveforms.

For signals in the rules file that have the at (@) sign as their leading symbol, the default scope paths have no impact on them.

The default scope path is an empty string ("").

Value Returned

TRUE if successful; otherwise returns FALSE.

Examples

```
cmpSetScopePath system.i_cpu
cmpSetSignalPair data         # compare system.i_cpu.data
cmpSetSignalPair @system.data # compare system.data
```

cmpSetSignalPair

Description

Specify the signal pair to be compared. The comparison is based on the options specified in `cmpSetCmpOption` and the triggers specified in `cmpSetCmpTrigger`. The default value is used when the options and triggers are not specified in the `cmpSetCmpOption` and `cmpSetCmpTrigger` commands.

NOTE: For MDA signals, this command can only expand the top-most dimension instead of all dimensions.

Syntax

```
cmpSetSignalPair [GoldenSignalOrModule [SecondarySignalOrModule]]
[-shift %f[Unit]] [-type GoldenSignalType [SecondarySignalType]]
[-Level %d [%d]]
```

Arguments

`GoldenSignalOrModule [SecondarySignalOrModule]`

`GoldenSignalOrModule` or `SecondarySignalOrModule` can be a signal name, module name, or NamePattern. The meaning of NamePattern is the same as those in the `cmpSetNameMap` command. However, NamePattern in `cmpSetSignalPair` can only match signal names. It cannot match module names. The signal name can also be a memory signal whose values are dumped into the FSDB files with the `$FSDBDumpMem` dumping command.

If `GoldenSignalOrModule` is omitted, the default is the top module. If `SecondarySignalOrModule` is omitted, the secondary signals are generated from `GoldenSignalOrModule` based on the defined mapping rules. If no mapping rules are defined, the name of the secondary signals is the same as `GoldenSignalOrModule`.

If `GoldenSignalOrModule` or `SecondarySignalOrModule` represents a field of a VHDL composite signal, the scope delimiter must be set to '/'.

`-shift %f[Unit]`

Indicate the offset time to be shifted for secondary signals, including the signals to be compared and the signals defined by `cmpSetCmpTrigger`. The default is *0*. For example:

```
cmpSetSignalPair system.i_cpu.data -shift 100
```

After the secondary FSDB file is shifted by 100 time unit, perform further comparison based on current options.

`-type GoldenSignalType [SecondarySignalType]`

Specify the type of signals to be compared in the modules. If only one argument is specified, it is used for both golden and secondary waveforms. `GoldenSignalType` and `SecondSignalType` can be one of `{IN, OUT, INOUT, BOUNDARY, LOCAL, ALL, Buffer, Linkage, Net, Register}` or the union of some of them, such as `IN+OUT`. The default is *ALL*.

The signal types are further explained in the following table:

SignalType	Explanation
IN	Input direction signal
OUT	Output direction signal
INOUT	Inout direction signal
BOUNDARY	IN | OUT | INOUT
LOCAL	Signals that cannot have a specified direction
ALL	All types + directions
Buffer	Buffer direction in VHDL
Linkage	Linkage direction in VHDL
Net	Net data type
Register	Register data type

`-level %d [%d]`

Specify the hierarchy level limit for the module to be compared. If only one argument is specified, it is used for both golden and secondary waveforms. If `GoldenSignalOrModule` is not a module name, this option have no impact. Nor the `SecondarySignalOrModule` have any impact. The level n indicates that all signals in the module specified in `cmpSetSignalPair` must be compared, and also the submodules whose depths, starting from the specified module, are less than n. The default is 1.

When n is 0, all signals in the hierarchy starting from the specified module is compared. Consider this example:

```
CmpSetSignalPair system.i_cpu -Level n
```

If n is 1, only the signals in `system.i_cpu` can be compared; if n is 2, besides the signals in `system.i_cpu`, the signals in submodules of `system.i_cpu` can also be compared.

The two signals to be compared have to satisfy the constraints below.

1. When neither signal is a composite signal, digital values can only be compared with digital values if the sizes of the two values are the same. An analog signal is also considered a digital signal if the corresponding AD threshold has been defined. For non-digital values, they have to be the same value type.
2. When only one signal is a composite signal, all elements of the composite signal and the non-composite signal must be digital and their total sizes must be the same.

3. When both signals are composite signals, their number of elements are the same and each corresponding element pair should satisfy the constraints above.

Value Returned
TRUE if successful; otherwise returns FALSE.

Examples
```
cmpSetSignalPair {%h.%s[%d:%d]}{%1h.%1s[%1d:%2d]}
```
Compare all bus signals. The mapping rule is full name match.

```
cmpSetSignalPair system.%s
```
Compare all signals under system; `system.i_cpu.data` is not compared. The mapping rule is full name match.

```
cmpOpenFsdb rtl.fsdb gate.fsdb
cmpSetSignalPair top -level 0
cmpCompare
```
Compare all signals of `rtl.fsdb` and *g*`ate.fsdb`.

cmpSetSkipSignal

Description
Specify signals and skip them when comparing. This can apply to iterators. `cmpSetSkipSignal` needs to be called before `cmpSetNameMap` and `cmpSetSignalPair` and the previous rules needs to be cleared. If `cmpSetSkipSignal` has no argument, all rules are cleared.

Syntax
```
cmpSetSkipSignal [str[%s|%d]*]*
```

Argument
`str`

String of signal name. Full hierarchy name is NOT supported.

Value Returned
TRUE if successful; otherwise returns FALSE.

Examples
```
cmpSetSkipSignal addr%d C%s i j k
```
Iterators are i, j, k, addr1, addr11, C_l, and C_sel.

`cmpSetSkipSignal %s`

Skip all signals.

`cmpSetSkipSignal "C%d" "addr%s"`

Skip signals with the matched name under ANY scope.

cmpSetStateMap

Description

Set arbitrary state mapping. Multiple state maps may be specified.

Syntax

```
cmpSetStateMap [-datatype %s [%s]] [-asym]
[( '(' %v, %v, [T | F | -] ')')] ...
```

Arguments

`-datatype %s [%s]`

`%s` can be Verilog, VHDL, or EVerilog. The default is *Verilog*.

`[-asym] [( '(' %v, %v, [T | F | -] ')')]`

Indicate that the specified mappings are asymmetrical; for example, `cmpSetStateMap –asym (0,1,F) (1,0,T)` indicates the *nCompare* module flags a mismatch error if the golden signal is 0 and the secondary signal is 1. However, the *nCompare* module does not flag an error if the golden signal is 1 and the secondary signal is 0.

If the second `%s` is omitted, this command defines the mapping among states that belong to the same data type. If the `-DataType` option is omitted, the data type is in Verilog format. For Verilog, `%v` can be 0, 1, x, or z. For VHDL, `%v` can be U, X, 0, 1, Z, W, L, or H. For convenience, `%v` can be * for Verilog, and VHDL, which represents all data types except the value that indicates "don't care". `T` indicates the two values are equal. `F` indicates they are unequal. A dash (`-`) indicates that the two values do not need to be compared.

When two values are equal, if their strengths are different and the strength switch is turned on, they are regarded as different values. If the X switch in the `cmpSetCmpOption` command is turned *off*, x (or X if VHDL is considered) does not need to be compared with any other value. The `cmpSetStateMap` command can more accurately specify which values do not need to be compared with x(X). On the other hand, if the X switch in the `cmpSetCmpOption` command is turned *on*, x(X) has to be compared with all values. However, if the Z switch in the `cmpSetCmpOption`

command is turned off, x(X) is not compared with z(Z). The Z switch is similar.

If no state mappings are specified in the command, the state mappings are be restored to the defaults. Defaults of state mapping between values are defined as the following:

1. Two identical values are equal.
2. If the X switch is off, the X value does not need to be compared with any values.
3. If the Z switch is off, the Z value does not need to be compared with any values.
4. The value "don't care" does not need to be compared with any values.
5. Two values that do not satisfy the conditions above are different.

Value Returned

TRUE if successful; otherwise returns FALSE.

Examples

```
cmpSetStateMap (1,x,F) (0,z,T)
```

Indicates: 1!=x, 0==z.

```
cmpSetCmpOption -X F -Z T
```

Indicates that x is not compared with any value, and z has to be compared with all values and 0==z but 1!=z.

```
cmpSetStateMap (0,x,-) (1,x,T) (z,z,-)
```

Indicatesthat 0 is not compared with x; 1==x; z is not compared with z.

```
cmpSetStateMap
```

Restore to defaults.

```
cmpSetStateMap -datatype VHDL (U,*,F)
```

U is different than any value except '-'.

cmpSetTimeScale

Description

Set the time scale for the *GOLDEN* or *SECONDARY* waveform file. This is similar to *cmpGetTimeUnit* except the destination must be specified.

NOTE: This command must be set before the **cmpOpenFSDB** command.

Syntax
```
cmpSetTimeScale GOLDEN|SECONDARY %f[Unit]
```

Argument
`Unit`

The unit can be Gs | Ms | Ks | s | ms | us | ns | ps | fs.

Value Returned
TRUE if successful; otherwise returns FALSE.

Examples
```
cmpSetTimeScale GOLDEN 10ps
```
Sets the time scale in the *GOLDEN* file to *10ps*.

```
cmpSetTimeScale SECONDARY 1.0ns
```
Sets the time scale in the *SECONDARY* file to *1ns*.

cmpSetTimeUnit

Description
Set the default time unit for the time value in the rule file. If the time unit is not set in `cmpSetTimeUnit`, the default time unit is set the same as the golden FSDB in `cmpOpenFsdb`.

NOTE: This command must be set before the **cmpOpenFSDB** command.

Syntax
```
cmpSetTimeUnit %f Unit
```

Argument
`Unit`

The unit can be (`Gs` | `Ms` | `Ks` | `s` | `ms` | `us` | `ns` | `ps` | `fs`). If it is omitted, the unit is `ns`.

Value Returned
TRUE if successful; otherwise returns FALSE.

Example
```
cmpSetTimeUnit 10ps
```

cmpSetTypeVar

Description

Define type variables for name mapping.

Syntax

```
cmpSetTypeVar varname ([%d]STRING | [%d]INTEGER | [%d]HIERARCHY |
CHAR) [-'%c'] [-INTEGER] [+EMPTY]
```

Arguments

`varname`

The `varname` can only be a letter and is case-sensitive. If `varname` is one of the predefined type variables, including h, s, d, and c, a syntax error is reported.

`[%d]STRING`

`[%d]STRING` indicates that `varname` is a string with the length of `%d` and it cannot exceed a hierarchy scope.

`[%d]INTEGER`

`[%d]INTEGER` indicates that `varname` is an integer with the length of `%d`, including the length of leading zeros (for example, the length of 001 is 3).

`[%d]HIERARCHY`

`[%d]HIERARCHY` indicates that `varname` is a string of `%d` hierarchies, where these hierarchies are separated by the scope delimiter.

`CHAR`

`CHAR` indicates that `varname` is a character and cannot be the scope delimiter.

`-'%c'`

`-'%c'` indicates that `varname` does not contain `%c`.

`-INTEGER`

`-INTEGER` indicates that `varname` cannot contain integer numbers.

`+EMPTY`

`+EMPTY` indicates that `varname` can be empty.

Value Returned

TRUE if successful; otherwise returns FALSE.

Examples

```
cmpSetTypeVar a 3STRING -'_' + EMPTY
```

'a' indicates a string with a length of 3 and cannot contain "_", 'abc' matches 'a', "" matches 'a' also, but 'a/b' does not match 'a', where / is the delimiter of hierarchy, and 'ab_' does not match 'a' either.

```
cmpSetTypeVar b INTEGER +EMPTY
```

'b' indicates an integer '321' matches 'b', as does "", but 'm2' does not.

```
cmpSetTypeVar H 2HIERARCHY
```

'H' indicates a string of two hierarchies for /system/i_cpu/data, system/i_cpu matches H, but /system/i_cpu does not.

cmpToFsdb

Description

Convert the results of the simulation file into FSDB format. This command converts `dump_file_name` into `fsdb_file_name`. If the signal name of `dump_file_name` is specified by its full path, `cmpToFsdb` uses the default delimiter that is set by `cmpSetDelimiter` command as the scope delimiter for separating the full path signal name.

NOTE: This command must be set before the **cmpOpenFSDB** command.

Syntax

```
cmpToFsdb dump_file [fsdb_file]
```

Arguments

`dump_file`

Specify the input dump file name. The `dump_file` can be in Verilog, HSPICE, xp, rawfile, or wfm format.

`fsdb_file`

Specify the desired file name of the generated FSDB file.

Value Returned

TRUE if successful; otherwise returns FALSE. If `fsdb_file` is omitted, the generated FSDB file is `dump_file_name.fsdb`.

Examples

```
cmpToFsdb rtl.dump rtl.fsdb
```

```
cmpOpenFsdb rtl.fsdb
cmpCompare
```

Error File

The *nCompare* comparison results are written to an error report file. The default extension name of the error report file is *.nce*. It can be read by any general text editor or viewer, such as vi, textedit, or notepad.

Each warning message or mismatch result is described on an individual line in the file. Each line contains a number of fields and a comment is identified with a semicolon (;) character. The number of fields in the line depends on the type of error it has.

File Format

Each mismatch result, error message, or warning message is represented by the following format:

```
[error type code] [error field 1] . . . [error field N]
```

If the first character of a line is a semicolon (;) character, then the semicolon and all characters following it to the next new line are treated as a comment.

The error type code is an integer number and each number describes a different error. Each error field has a different format that is described below.

Value Type	Field Description
S	STRING format. The STRING format field must be enclosed by the double quote (“) character. For example, "*Cannot find specified signal in golden FSDB*".
E	HDL signal or expression format. This format field should be enclosed by single back quotes. For example,"*top.CPU.clock*".
N	INT format
T	TIME format. For example, 137x100ps.
B	BOOL(T/F) format

100: Rule File Information

Field 1:	S:	The full path of the rule file.
Field 2:	N:	Total rule errors.
Field 3:	N:	Total rule warnings.

101: Rule File Error Message

Field 1:	N:	Line number.
Field 2:	S:	Error message.

201: Rule File Warning Message

Field 1:	N:	Line number.
Field 2:	S:	Warning message.

301: Rule File Information Message

Field 1:	N:	Line number.
Field 2:	S:	Information message.

010: Common Comparison Option

Field 1:	N:	Index.
Field 2:	B:	Compare glitch?
Field 3:	B:	Compare strength?
Field 4:	N:	Maximum errors.
Field 5:	N:	Maximum errors per signal.
Field 6:	B:	Compare 'X?’
Field 7:	B:	Compare 'Z?’
Field 8:	T:	Positive value of tolerance window.
Field 9:	T:	Negative value of tolerance window.
Field 10:	T:	Shift time.
Field 11:	N:	The number of the time range.
Filed 10+(n*2):	T:	Begin time of the nth time range.
Filed 11+(n*2):	T:	End time of the nth time range.
Filed 12+(n*2):	N:	Signal format of golden FSDB is Verilog(2) style or VHDL(3) style.
Filed 13+(n*2):	N:	Signal format of secondary FSDB is Verilog(2) style or VHDL(3) style.

011: Signal Comparison Option

Field 1:	N:	Index.
Field 2:	B:	Enable clock trigger option?
Field 3:	E:	Clock signal name. If the prefix of the clock signal name is “G:”, it indicates that this signal came from Golden FSDB file. If the prefix of the clock signal name is “S:”, it indicates that this signal came from Secondary FSDB file.
Field 4:	S:	Clock edge, falling ("F"), rising ("R"), or both edges ("B").
Field 5:	T:	Setup time.
Field 6:	T:	Hold time.
Field 7:	T:	Sampling time.
Field 8:	B:	Enable cycle option?
Field 9:	T:	Cycle period time.
Field 10:	T:	Delay time.
Field 11:	T:	Time width of cycle window.
Field 12:	B:	Enable expression trigger option?
Field 13:	E:	HDL logical expression. If the prefix of the logical expression name is “G:”, it indicates that this signal came from Golden FSDB file. If the prefix of the logical expression name is “S:”, it indicates that this signal came from Secondary FSDB file.

500: FSDB Information

Field 1:	S:	The full path of the golden FSDB file.
Field 2:	S:	The full path of the secondary FSDB file.
Field 3:	N:	The number of total compared signals.
Field 4:	N:	The number of mismatched signals.
Field 5:	N:	The number of total mismatched errors.

510: Comparison Mismatch Error

Field 1:	T:	Mismatch time for the golden signal.
Field 2:	T:	Mismatch time for the secondary signal.
Field 3:	T:	Mismatch time width for the golden signal.
Field 4:	T:	Mismatch time width for the secondary signal.
Field 5:	E:	Golden signal name.

Field 6:	E:	Secondary signal name.
Field 7:	N:	The index of the common comparison option.
Field 8:	N:	The index of the golden signal comparison option.
Field 9:	N:	The index of the secondary signal comparison option.
Field 10:	N:	Golden signal value.
Field 11:	N:	Secondary signal value.

511 - 513: Comparison Diff Error

511: Secondary signal has more changes than the golden signal.

512: Secondary signal has less changes than the golden signal.

513: Secondary signal differs from the golden signal.

Field 1:	T:	Mismatch time for the golden signal.
Field 2:	T:	Mismatch time for the secondary signal.
Field 3:	T:	Mismatch time width for the golden signal.
Field 4:	T:	Mismatch time width for the secondary signal.
Field 5:	E:	Golden signal name.
Field 6:	E:	Secondary signal name.
Field 7:	N:	Index of the common comparison option.
Field 8:	N:	Index of the golden signal comparison option.
Field 9:	N:	Index of the secondary signal comparison option.

530: Unstable Error in Setup Time

Field 1:	T:	Unstable time.
Field 2:	T:	Unstable time width.
Field 3:	B:	Is [Field 4] a golden signal or secondary signal?
Field 4:	E:	Unstable signal name.
Field 5:	N:	Index of the common comparison option.
Field 6:	N:	Index of the golden signal comparison option.
Field 7:	N:	Index of the secondary signal comparison option.

531: Unstable Error in Hold Time

Same as 530.

532: Unstable Error in Cycle Time

Same as 530.

533: No More Matched Value Errors

Field 1:	T:	The last time point at which the trigger conditions were satisfied.
Field 2:	T:	Plus field1 is the total simulation time of the design.
Field 3:	B:	Is [Field 4] a golden signal or secondary signal?
Field 4:	E:	The signal name has no more trigger points from field1; however, the other signal (being compared with it) does.
Field 5:	N:	Index of the common comparison option.
Field 6:	N:	Index of the golden signal comparison option.
Field 7:	N:	Index of the secondary signal comparison option.

540: Comparison Warning Message

Report this if the number of found errors of this signal pair reaches *MaxErrorPerSignal*.

Field 1:	T:	Occurred time for the golden signal.
Field 2:	T:	Occurred time for the secondary signal.
Field 3:	E:	Golden signal name.
Field 4:	E:	Secondary signal name.
Field 5:	N:	Index of the common comparison option.
Field 6:	N:	Index of the golden signal comparison option.
Field 7:	N:	Index of the secondary signal comparison option.

Mixed Type FSDB Comparison

The *nCompare* module is able to compare the mixed type of FSDB files between Verilog, VHDL, and EVerilog. The comparison rules are listed in the following sections: **Verilog vs. VHDL**, *** Blank cell indicates mismatch.**, *** Blank cell indicates mismatch.**, **Verilog vs. EVerilog**, and **VHDL vs. EVerilog.**

Legends

Verilog

Value Filled	Definition	Strength
0	Logic Zero	Strong
1	Logic One	Strong
X	Logic unknown	Strong
Z	High impedance	

VHDL

Value Filled	Definition	Strength
0	Logic Zero	Strong
1	Logic One	Strong
X	Logic unknown	Strong
Z	High impedance	
W	Logic unknown	Weak
L	Logic Zero	Weak
H	Logic One	Weak
U	Unset/Not initialized	
-	Don’t care	

EVerilog

Value Filled	Definition	Strength
L	Low	St Hi

l	Low (two or more drivers active)	St Hi
H	High	Hi St
h	High (two or more drivers active)	Hi St
X	Unknown (Don't care)	St St
T	Three-state	Hi Hi
D	Low	St Hi
d	Low (two or more drivers active)	St Hi
U	High	Hi St
u	High	Hi St
N	Unknown	St St
Z	Three-state	Hi Hi
?	Unknown	St St
0	Low (both input and output are active with 0 value)	St Hi
1	High(both input and output are active with 1 value)	Hi St
A	Unknown (input 0 and output 1)	St St
a	Unknown (input 0 and output X)	St St
B	Unknown (input 1 and output 0)	St St
b	Unknown (input 1 and output X)	St St
C	Unknown (input X and output 0)	St St
c	Unknown (input X and output 1)	St St
F	Three-state (input and output unconnected)	Hi Hi

f	Unknown (input and output three-stated)	Hi Hi

Verilog vs. VHDL

Column: Verilog; **Row**: VHDL

	0	**1**	**X**	**Z**
0	Match			
1		Match		
X			Match	
Z				Match

	W	**L**	**H**	**U**	**-**
0		Match			Match
1			Match		Match
X	Match			Match	Match
Z					Match

* Blank cell indicates mismatch.

Verilog vs. EVerilog

Column: Verilog; **Row**: EVerilog

	L	**l**	**H**	**h**	**X**	**T**
0	Match	Match				
1			Match	Match		
X					Match	
Z						Match

	D	**d**	**U**	**u**	**N**	**Z**
0	Match	Match				
1			Match	Match		
X					Match	
Z						Match

	?	0	1	A	a
0		Match			
1			Match		
X	Match			Match	Match
Z					

	B	b	C	c	F	f
0						
1						
X	Match	Match	Match	Match		
Z					Match	Match

* Blank cell indicates mismatch.

VHDL vs. EVerilog

Column: VHDL; **Row**: EVerilog

	L	l	H	h	X	T
0	Match	Match				
1			Match	Match		
X					Match	
Z						Match
U					Match	
W					Match	
L	Match	Match				
H			Match	Match		
-	Match	Match	Match	Match	Match	Match

	D	d	U	u	N	Z
0	Match	Match				
1			Match	Match		
X					Match	
Z						Match
U					Match	

W					Match	
L	Match	Match				
H			Match	Match		
-	Match	Match	Match	Match	Match	Match

	?	**0**	**1**	**A**	**a**
0		Match			
1			Match		
X	Match			Match	Match
Z					
U	Match			Match	Match
W	Match			Match	Match
L		Match			
H			Match		
-	Match	Match	Match	Match	Match

	B	**b**	**C**	**c**	**F**	**f**
0						
1						
X	Match	Match	Match	Match		
Z					Match	Match
U	Match	Match	Match	Match		
W	Match	Match	Match	Match		
L						
H						
-	Match	Match	Match	Match	Match	Match

* Blank cell indicates mismatch.

Appendix E: Power Aware Formats

Check Power Sequence Report Format

The results of checking the power sequence can be saved to a file and then restored at a later time. The syntax of the check power sequence report format is described in the following sections along with an example of its use. The semi-colon (;) character is used to designate comment lines and to define sections. There are four main pieces of information in the report file: design information, rule information, Isolation rule list, and error report list.

Design Information

This section defines the RTL design, power design, and simulation results information.

RTL Design
Specifies the RTL design files.

Syntax:

```
RTL_Design [File_List]
```

Example:

```
RTL_Design test.v
```

Power Design
Specifies the CPF/UPF power files.

Syntax:

```
Power_Design [File_List]
```

Example:

```
Power_Design test.cpf
```

Simulation Results

Specifies the FSDB file.

Syntax:

```
Simulation_Result [File_Name]
```

Example:

```
Simulation_Result /verdi/home/cases/220715/test.fsdb
```

Constraint Information

This section lists all the checking constraints.

Syntax:

```
FromTo   fromTime toTime
```

The default uses full time range.

```
MaxErrors   maxErrorCnt
```

The default is *1000*.

```
PowerDomain   powerDomainList
```

The default uses all power domains.

Example:

```
FromTo 0ps 3000000ps
MaxErrors 10000
PowerDomain "/testbench/top/\$VDMA_domain " "/testbench/top/
\$VMEM_domain " "/  testbench/\$VDD_domain "
```

Rule Information

This section lists all of the rules and indicates whether they were enabled for checking or not.

Syntax:

```
ToggleOn|ToggleOff ruleId ruleName
```

Refer to the **Power Rules Table** section in *Tcl Command Reference Manual* for details about the `ruleId` and `ruleName` mapping.

Example:

```
ToggleOn 100 "Isolation Control Signal has X"
ToggleOff 101 "Isolation Control Signal not On when Power Domain
Status Off"
    ...
```

Isolation Rule List

This section summarizes the Isolation rules.

Syntax:

```
Isolation isolationName commandKind
```

NOTE: The `commandKind` is for internal use.

Example:

```
Isolation "/TOP/system/PD_tx_iso1" 264
```

Error Report List

This section summarizes the reported errors.

Syntax:

```
ruleId Severity [PDName time|time_range] [IsoName IsoCtrlName
time|time_range] [ImpactedSigName time|time_range]
```

where `Severity` can be **E** for error or **W** for warning.

Example:

```
101 E "/TOP/system/@{\$PD_tx }" 800ns "/TOP/system/PD_tx_iso1" "/
@{\tx_iso&(tx_iso|tx_iso)&test[1]/b[1] }" 800ns
102 E "/TOP/system/PD_tx_iso1" "/
@{\tx_iso&(tx_iso|tx_iso)&test[1]/b[1] }" 750ns 760ns "/TOP/
system/receiver/in1[2]" 750ns
```

Example

```
; RTL_Design File_List
RTL_Design test.v

; Power_Design File_List
Power_Design test.cpf

; Simulation_Result File_Name
```

```
Simulation_Result /verdi/home/cases/220715/test.fsdb

; ToggleOn ruleId ruleName
ToggleOn 100 "Isolation Control Signal has X"
ToggleOn 101 "Isolation Control Signal not On when Power Domain
Status Off"
ToggleOn 102 "Isolation On but Signal has no Clamp Value"
ToggleOff 103 "Isolation Control never Enabled"
ToggleOff 200 "Power Domain Status Off but Signal Value Change
not Corrupt"
ToggleOn 201 "Power Domain Status has X"
ToggleOn 202 "Power Domain Status On but no Signal has Value
Change"
ToggleOn 203 "Power Domain Status always On"
ToggleOff 204 "Power Domain Status always Off"
ToggleOff 300 "Power Mode Table has invalid mode"
ToggleOff 301 "Power Mode Table has uncovered mode"

; Isolation isolationName commandKind
Isolation "/TOP/system/PD_tx_iso1" 264
…

; 101 Type PDName time IsoName IsoCtrlName time
101 E "/TOP/system/@{\$PD_tx }" 800ns "/TOP/system/PD_tx_iso1" "/
@{\tx_iso&(tx_iso|tx_iso)&test[1]/b[1] }" 800ns
101 E "/TOP/system/@{\$PD_tx }" 900ns "/TOP/system/PD_tx_iso1" "/
@{\tx_iso&(tx_iso|tx_iso)&test[1]/b[1] }" 800ns
101 E "/TOP/system/@{\$PD_tx }" 100ns "/TOP/system/PD_tx_iso1" "/
@{\tx_iso&(tx_iso|tx_iso)&test[1]/b[1] }" 0ns

; 102 Type IsoName IsoCtrlName time_range ImpactedSig time
102 E "/TOP/system/PD_tx_iso1" "/
@{\tx_iso&(tx_iso|tx_iso)&test[1]/b[1] }" 750ns 760ns "/TOP/
system/receiver/in1[2]" 750ns
…
```

Appendix F: Message Table

Verilog

The following table lists each available message number and its mapping usage when specifying the `+disable_message+<message_serial_numbers>` option in the `novas/verdi`, `vericom`, and `nrun` utility commands.

Number	Message Content
C00008	Non-constant instance parameter.
C00010	Identifier %s(type=%d) is not a task or function.
C00013	No source file.
C00018	Expression uses reals.
C00019	Failed to find identifier %s.
C00024	Specify filename after **-f**.
C00025	Specify filename after **-v**.
C00026	Specify directory name after **-y**.
C00027	Specify directory name after **-b**.
C00036	Invalid UDP table entry.
C00037	Invalid output.
C00038	Redundant UDP entry.
C00039	`include file \"%s\" cannot be read.
C00041	No register for UDP initial statement.
C00045	Failed to open source file \"%s\" with read access: %s.
C00047	Too many digits in binary.
C00048	Too many digits in octal.
C00049	Too many digits in decimal.
C00050	Too many digits in hexadecimal.
C00055	Number %d overflow.
C00059	Too much delay information for gate.
C00073	Invalid object type.
C00077	Unknown range in %s.

Number	Message Content
C00078	Only one register is allowed in UDP.
C00079	Invalid register in UDP.
C00080	Only one output is allowed in UDP.
C00081	First UDP port is not output.
C00083	Unknown memory in %s.
C00085	Mismatched arguments in function %s.
C00086	Identifier %s (type=%d) is not a function.
C00089	Missing event in event trigger statement.
C00091	Invalid path delay.
C00092	Invalid path specification.
C00097	More than one default.
C00098	Not a case item.
C00099	Conflict expansion %s.
C00100	Conflict expansion.
C00101	Too few operands for unary operator.
C00102	Too many operands for unary operator.
C00103	Too few operands for binary operator.
C00104	Too many operands for binary operator.
C00105	Too few operands for ternary operator.
C00106	Too many operands for ternary operator.
C00107	Identifier %s (type=%d) not a task or block.
C00134	Task or function (%s) duplicated.
C00144	Failed to find symbol %s.
C00147	Symbol %s duplicated.
C00148	Incompatible %s.
C00149	Invalid delay value.
C00150	Invalid delay.
C00152	Illegal strength.
C00153	Illegal delay.
C00154	Define memory %s as IO.
C00157	Illegal declaration.
C00169	Duplicate port reference %s.

Number	Message Content
C00182	Illegal identifier %s.
C00186	Incorrect UDP table.
C00188	Illegal specify statement.
C00194	The expression must have a constant value.
C00195	Macro %s redefined from (%s) to (%s).
C00196	Macro %s not defined.
C00197	Macro %s recursion.
C00198	Invalid UDP initial value.
C00199	UDP should have at least one input.
C00200	Illegal memory %s.
C00201	Only slice, index, concatenate are allowed for ports.
C00202	Port %s not input or inout port.
C00203	Port %s not input or inout port in task or function.
C00204	Port %s not input, output or inout port.
C00205	Invalid strength value.
C00206	Unknown range declaration in %s.
C00207	Illegal posedge/negedge.
C00208	Wrong task %s usage.
C00210	Missing statement in always statement.
C00211	Missing statement in initial statement.
C00212	Invalid task enable in function.
C00213	Missing task in enable statement.
C00215	Mismatched task %s arguments.
C00216	Illegal output/inout task parameter (%ld).
C00217	Missing system task in enable statement.
C00218	No task or block in disable statement.
C00219	Invalid strength in continuous assign.
C00220	Missing strength, delay, LHS or RHS in continuous assignment.
C00221	Missing delay, LHS or RHS in continuous assignment.
C00222	Missing condition in if statement.
C00223	Missing then statement in if statement.
C00224	Missing end in for statement.

Number	Message Content
C00225	Missing initial in for statement.
C00226	Invalid LHS in initial part of statement.
C00227	Missing step assignment in for assignment.
C00228	Missing repeat in for statement.
C00229	Missing statement in forever statement.
C00230	Missing repeat in repeat statement.
C00231	Missing statement in repeat statement.
C00232	Missing end condition in while statement.
C00233	Missing statement in while statement.
C00234	Missing select in case, casex, casez statement.
C00235	Missing LHS in event assignment.
C00236	Missing event and RHS in event assignment.
C00237	Missing RHS in event assignment.
C00238	Statement not allowed in function.
C00239	Missing LHS in delay assignment.
C00240	Illegal LHS in procedural assignment.
C00241	Missing delay and RHS in delay assignment.
C00242	Missing RHS in delay assignment.
C00243	Missing LHS in repeat assignment.
C00244	Missing repeat, event and RHS in repeat assignment.
C00245	Missing event and RHS in repeat assignment.
C00247	Missing LHS in deassign statement.
C00248	Illegal data in deassign statement.
C00249	Missing LHS in release statement.
C00250	Illegal data in release statement.
C00251	Missing event in event statement.
C00252	Missing statement in event statement.
C00253	Missing statement in delay statement.
C00254	Missing wait in wait statement.
C00255	Missing statement in wait statement.
C00256	Multi-input vector path with should be *> (not ==>).
C00257	Multi-input path not allowed.

Number	Message Content
C00258	Multi-output vector path should be *> (not ==>).
C00259	Diffsize vector should not have parallel path => must be *>.
C00260	Illegal port conditional expression.
C00261	Signal %s not module output or inout.
C00262	Signal %s not module input or inout.
C00263	Signal %s not module port.
C00264	Illegal parameter %s.
C00265	Incomplete slice.
C00266	Nonconstant index.
C00267	Unknown in part-select index.
C00268	Part-select to scalar.
C00269	Part-select on real type register or function.
C00270	Part-select to scalar Register or Function.
C00271	Invalid object type for part-select.
C00272	Invalid part-select index.
C00273	Most significant in part-select is out of range.
C00274	Most significant in part-select is out of range.
C00275	Least significant in part-select is out of range.
C00276	Least significant in part-select is out of range.
C00277	Incomplete index.
C00278	Bit select in scalar.
C00279	Invalid bit-select object.
C00280	Bit-select on real register.
C00281	Bit-select on scalar register.
C00283	Bit-select out of range.
C00284	Bit-select out of range.
C00302	No selection in case item %d.
C00303	Missing LHS in assignment %d.
C00304	Illegal LHS in continuous assignment.
C00305	Illegal LHS in quasi continuous assignment.
C00306	Illegal LHS in force statement.
C00307	Illegal LHS in step assignment in for statement.

Number	Message Content
C00308	Missing RHS in assignment.
C00315	Too many delays for gate: usage is not portable across Verilog simulators.
C00333	Concatenate must have numbers of known length.
C00337	Macro %s redefined.
C00341	Mismatched compiler directive `else.
C00342	Mismatched compiler directive `endif.
C00345	Expected filename after `include.
C00347	Number should be in [%d %d].
C00348	Only 0, 1, x, and - are allowed in UDP table.
C00349	Edge not allowed in combinational UDP.
C00350	Only one edge is allowed in UDP.
C00351	Invalid state in UDP.
C00355	Failed to open (**-f**) file \"%s\".
C00360	Illegal symbol %s.
C00360	Illegal symbol.
C00371	View %s is not defined for instance %s.
C00372	Too many instance parameters in instance.
C00373	Failed to find port %s for instance %s.
C00374	Too many instance parameter in instance %s.
C00375	Too many delays for UDP in instance %s.
C00379	Too many port connections on instance %s port %d.
C00380	Illegal output port on instance %s port %s.
C00381	Illegal output port on instance %s port %d
C00382	Illegal inout port on instance %s port %s.
C00383	Illegal inout port on instance %s port %d.
C00384	Illegal mixed port on instance %s port %s.
C00385	Illegal mixed port on instance %s port %d.
C00386	Expression for null port on instance %s port %s.
C00387	Expression for null port on instance %s port %d.
C00388	Port sizes differ (%d vs %d) in port connection (port %s).
C00389	Port sizes differ (%d vs %d) in port connection (port %d).
C00390	Port %s for instance %s: usage is not portable across Verilog simulators.

Number	Message Content
C00391	Too few portInstance connections on instance.
C00392	Too few portInstance connections on instance %s.
C00393	Illegal defparam %s.
C00394	Task %s must not exist in expression.
C00395	Illegal type in deassign statement.
C00396	Illegal LHS in force assignment.
C00397	Illegal type in release statement.
C00398	Illegal output/inout parameter in task %s (position %d).
C00399	Incompatible parameters on task/function %s.
C00407	Primitive instance %s has incorrect ports.
C00412	Port %s duplicated.
C00413	Too many delays.
C00414	%s conflict expansion.
C00502	Module %s of instance %s is from the symbol library.
C00506	Unmatched `celldefine `endcelldefine.
C00507	Incompatible range declaration of %s.
C00508	Maximum error count reached (%d).
C00509	Usage of %s is only supported if the **-2001** option is specified.
C00510	Mismatched compiler directive `elsif.
C00513	Wrong indexed part-select.
C00514	Duplicated named instance parameter assignment for %s in instance %s.
C00515	Named instance parameter %s is a localparam.
C00516	Net %s must be explicitly declared when `default_nettype is set to none.
C00517	`default_nettype can be used only outside of module definitions.
C00518	Incomplete array address.
C00519	Usage of %s is supported only when the **-sv** option is specified
C00526	Net %s is not declared for implicit port connection.
C00527	Failed to bind implicit port %s of module %s.
C00528	Failed to find member %s in structure %s.
C00529	Block label and name cannot exist at the same time.
C00530	Block name is inconsistent.
C00531	Task/Function Null portlist using parentheses.

Number	Message Content
C00533	Unbound user defined type %s.
C00534	Function %s has no argument.
C00541	Illegal usage of unpacked dimension.
C00543	%s is a protected module, some related errors will be reported as warnings.
C00545	Specify filename after %s.
C00546	Usage of %s is supported only when the **-vc** option is specified.
C00547	Library mapping confliction. %s matches rule:\"%s %s\" and rule:\"%s %s\".\n
C00548	UDP table entry confliction: \nentry %s at line %d conflicts with \nentry %s at line %d.
C00549	Too many indexes: %s.
C00551	Inline PSL: %s.
C00552	File %s is skipped since the file format is binary or encrypted.
C00555	Invalid initialization. Mismatched number of elements.
C00556	Port %s duplicated for instance %s.
C00557	-%s ignored when **-nclib** is specified.
C00558	Subcircuit has no name.
C00559	Failed to find the package %s.
C00561	Failed to find %s in package %s.
C00562	Identifier '%s' has multiple wildcard definitions in '%s' and '%s'.
C00563	Net %s is not connected by dot-star.
C00564	Range ignored for integer or time variables.
C00565	'%s' is not a parameter type.
C00566	Packed array is not allowed for this data type.
C00560	Failed to find the class %s.
C00568	Failed to open library file \"%s\" with read access: %s.
C00569	Failed to find class %s.
C00570	Concurrent assertion inside looping statement is not allowed.
C00572	An inout port (%s) must be a net type.
C00573	Cyclical type (%s) dependency.
C00574	Failed to find the corresponding comment mark for \"/*\" in \"%s\".
C00576	%s is referenced before declaration.

Number	Message Content
C00577	Conflict in UDP table.
C00579	%s %s redefined.
C00583	Class %s duplicated.
C00584	The **`-cunit** option has higher priority than **`-cuname**, and **`-cuname** will be ignored.
C00586	Multiple protected/local qualifiers specified.
C00587	Preprocessing syntax error of token '%s'.
C00590	%d extra actual argument(s) of function macro %s.
C00591	Missing %d actual argument(s) of function macro %s.
C00592	'%s' is an illegal command line macro definition.
C00593	'%s' is an illegal delimiter command line macro definition.
C00594	Invalid time unit : %s.
C00595	Invalid time precision : %s.
C00596	Invalid argument '%s' for `default_nettype.
C00597	Invalid keyword specifier %s.
C00598	Mismatched compiler directive `end_keywords
C00599	Invalid unconnected driver '%s'.
C00601	`ifdef or `ifndef or `else or `elsif with no matching `endif at source file %s(%d).
C00602	Illegal enumeration base type.
C00603	No identifier after %s. The first character of an identifier should not be a digit or $; it can\ be a letter or an underscore. Identifiers should be case-sensitive.
C00604	Unmatched '%s' by the protected code at source file %s(%d).
C00605	Failed to bind a duplicate instance %s in %s.
C00606	Evaluation of a generate loop has exceeded the iteration limit of %d.
C00607	Mismatched simulation type. The library is compiled as %s type but import design as %s type.
C00608	Expect input signal %s to be net type.
C00609	Failed to find class %s or class %s is not inherited from class ovm_test/uvm_test of package ovm_pkg/uvm_pkg.
C00610	Failed to find class %s (+OVM_TESTNAME/UVM_TESTNAME=%s) or class %s is not inherited from class ovm_test/uvm_test of package ovm_pkg/uvm_pkg.
C00611	Interface port connected illegally at line %d.

Number	Message Content
C00612	Package importing/exporting itself %s.
C00613	Illegal parameter type '%s'.
C00614	Failed to load module %s because SVA/SVTB exists. To load this module and its leaf/child instances, add **-licverdi** if a \"Verdi\" license is available.
C00615	A value for the **-cuname** option should be specified.
C00616	Variable '%s' has an automatic lifetime in module '%s'. All variables in a module scope must have a static lifetime.
C00617	Variable '%s' has an automatic lifetime in package '%s'. All variables in a package scope must have a static lifetime.
C00618	The **-logdir** option should be used in the command line. It cannot be used in the <filename>.f.
C00619	An import declaration should not be used in a class declaration.
C00622	Only one global clocking declaration is allowed.
C00623	Failed to find the specified instance parameter %s in the master %s.
C00624	No debugging information for '%s'.
C00625	Failed to bind instance '%s'. Only modules or instances can be bound.
C00626	Reusing the genvar %s in the nested generate loops is not allowed.
C00627	Using different genvars for the genvar assignments in the same generate loop is not allowed.
C00628	Range ignored for real variables.
C00629	Only one default clocking can be specified in a module, interface, program, or checker.
C00630	Clocking block %s duplicated.
C00631	The specified default clocking %s does not exist.
C00632	The enum variable %s is not assigned to the correct enum type. The enum variable should be assigned to its corresponding enum type.
C00633	An x or z value cannot be assigned to the enum variable with 2-state declaration.
C00634	A value should be assigned to the enum variable that follows the enum variable with an x or z assignment.
C00635	The library %s is compiled as type %s which does not match the import design type %s.
C00636	Genvar %s cannot be used in this context. It should be referenced in the generate loop it indexes.
C00637	The **-incsave** option is disabled because **-smartinc** and **-incsave** are specified at the same time.

Number	Message Content
C00638	Illegal part-select or slice exists in the array %s.
C00639	Variable %s cannot be driven by multiple continuous assignments.
C00640	File '%s' is included recursively.
C00641	The view '%s' in file '%s' cannot be found for dynamic configuration.
C00660	Failed to read the XML file because the file cannot be recognized.
C00661	The XML file generated by VCS-%s cannot be read by this Verdi version. Use the latest Verdi version or contact local support.
C00662	The XML file is generated by VCS-%s. Verdi only supports versions after %s. Correct results cannot be guaranteed when earlier versions of the XML file are used.
C00678	The -wreal option is an illegal option in compilation time. The **-realport** option should be used instead.

VHDL

The following table lists each available message number and its mapping usage when specifying the `+disable_message+<message_serial_numbers>` option in the `novas/verdi`, `vhdlcom`, and `nrun` utility commands.

Number	Message Content
C02000	Illegal symbol.
C02001	Illegal integer.
C02002	Number must end with a digit.
C02003	Integers must have a positive exponent.
C02004	Underscore must be between two digits.
C02005	Illegal identifier.
C02006	Failed to find the alias name.
C02007	Alias must refer to an object.
C02008	subtype indication is required when aliasing a slice or index expression.
C02009	Aliased object cannot contain a signature.
C02010	Aliased object name must be a static name.
C02011	The %s is not a component instance label.
C02012	Failed to find %s.
C02013	Component %s: %s is already bound in an architecture configuration specification.
C02014	Failed to find component instance %s: %s.
C02015	Failed to find component instances.
C02016	Failed to find component %s.
C02017	Resolution functions do not apply to access types.
C02018	Invalid resolution function.
C02019	Base type is not an array.
C02020	Array index must be a discrete type.
C02021	Too many indexes for this array type.
C02022	Index constraint cannot be applied to the constrained type.
C02023	Too few indexes for this array type.
C02024	Target of signal assignment cannot be in parenthesis.
C02025	Target of %s assignment is not %s.
C02026	Object with %s type must be a variable.

Number	Message Content
C02027	Generic %s must be a constant.
C02028	Generic must have a mode IN.
C02029	Port %s is not a signal.
C02030	Parameters must be 'in', 'out', or 'inout'.
C02031	Formal parameters of functions can only be of mode IN.
C02032	Constant parameter must be mode IN.
C02033	Variable parameters are not allowed for functions.
C02034	Value %s is out of type range.
C02035	Value %s is out of type range.
C02036	Value %d is out of type range.
C02037	Type error in range expression.
C02038	Type error found when resolving infix expression.
C02039	Slice direction does not match subtype direction.
C02040	Labels do not match (%s and %s).
C02041	Symbol %s was already declared in this region.
C02042	Label %s was declared but not used.
C02043	Array type for %s is not constrained.
C02044	File type is not allowed for the object.
C02045	Failed to find the name in the use clause.
C02046	A procedure designator must always be an identifier.
C02047	Failed to find %s name: %s.
C02048	Failed to find name: %s.
C02049	Failed to find target type.
C02050	Process sensitivity list cannot be a qualified expression.
C02051	%s is not a signal.
C02052	Signal name in sensitivity list is not static.
C02053	Duplicated signals in sensitivity list.
C02054	%s label is needed.
C02055	No guarded signal or expression is visible.
C02056	Record element name cannot be used by itself.
C02057	Failed to find object in expression: %s.
C02058	%s is not an attribute of %s.

Number	Message Content
C02059	Failed to read %s: %s.
C02060	Driving %s signal: %s is not allowed.
C02061	Making two objects' named %s visible.
C02062	Illegal use of deferred constant %s.
C02063	The ALL suffix must be used only with the access type prefix.
C02064	Illegal ALL suffix.
C02065	Attribute parameter must be locally-static.
C02066	Ambiguous type conversion.
C02067	Prefix of range attribute has an error.
C02068	The range of slice name is illegal.
C02069	Illegal range of slice name.
C02070	Failed to find declaration: %s.
C02071	Unknown identifier: %s.
C02072	Failed to find object type of selected name's prefix: %s.
C02073	Failed to find declaration type of selected name's prefix: %s.
C02074	Failed to find array type object of selected name's prefix: %s.
C02075	Illegal type conversion: %s.
C02076	Failed to find the binding entity %s.
C02077	Failed to find object type of index name's prefix: %s.
C02078	Failed to find indexed name's prefix: %s.
C02079	Prefix of slice name must be a one-dimensional array.
C02080	Failed to find object type of slice name's prefix: %s.
C02081	Failed to find slice name's prefix: %s.
C02082	Ambiguous map.
C02083	Failed to find actual part of map.
C02084	Formal %s is of mode IN. Type conversions and conversion functions of mode IN are not allowed on formals.
C02085	Actual expression for generic %s cannot reference a signal.
C02086	Only associating IN port %s with IN, INOUT, or BUFFER is allowed.
C02087	Only associating OUT port %s with OUT or INOUT is allowed.
C02088	Only associating INOUT port %s with INOUT is allowed.
C02089	Only associating BUFFER port %s with BUFFER is allowed.
C02090	Assigning to object with mode IN: %s is not allowed.

Number	Message Content
C02091	Assigning to object with mode LINKAGE: %s is not allowed.
C02092	Incompatible modes for port %s.
C02093	Actual for formal %s is not a signal.
C02094	Invalid conversion function of formal %s.
C02095	Entity aspect is not consistent with component.
C02096	Actual of port is not a signal.
C02097	Array length is %d. Expected length is %d.
C02098	No actual specified for %s.
C02099	Invalid configuration name.
C02100	Invalid generic map.
C02101	Invalid port map.
C02102	Invalid expression.
C02103	Ambiguous types in assignment.
C02104	Type conflicts in expression.
C02105	Illegal assignment in right hand side.
C02106	Failed to resolve expression type.
C02107	Failed to find aggregate.
C02108	Expression needs a return type.
C02109	Aggregate length is %d. Expected length is %d.
C02110	String length is %d. Expected length is %d.
C02111	%s was specified previously.
C02112	Ambiguous expression.
C02113	Two indistinct aggregates may result in an ambiguous expression.
C02114	Syntax error at %s.
C02115	Missing token.
C02116	%s is VHDL-93's keyword. Try the \"-93\" option instead.
C02117	%s is VHDL-93's keyword.
C02118	Illegal range.
C02119	Ambiguous range.
C02120	OTHERS must be specified as the last aggregate index.
C02121	Element associations -with element declarations in array aggregate is not allowed.
C02122	Aggregate with multiple choices has a non-static choice.

Number	Message Content
C02123	Attribute %s requires a static signal prefix.
C02124	Failed to find predefined attribute's prefix: %s.
C02125	Invalid type of attribute prefix.
C02126	Static divided by zero.
C02127	Negative exponents are not allowed with integers.
C02128	Type error found when resolving resolution function.
C02129	Failed to find subprogram call.
C02130	Ambiguous subprogram call.
C02131	Subprogram \"%s\" is ambiguous.
C02132	Types do not conform for deferred constant %s.
C02133	Deferred constants are only allowed in a package declaration.
C02134	Guarded signals must be resolved.
C02135	Driving implicit signal: guard is not allowed.
C02136	Driving signal %s from this subprogram is not allowed.
C02137	Non-resolved signal %s may have multiple sources.
C02138	Non-resolved signal %s has multiple sources.
C02139	The open mode (%s) has an error.
C02140	Type error found when resolving alias object.
C02141	Failed to alias multi-dimensional array type.
C02142	%s is not a logical predefined operator.
C02143	Function %s may complete without a return.
C02144	Failed to open library %s.
C02145	Failed to label this statement.
C02146	Attribute %s may not be accessed from parameter %s.
C02147	Attribute may not have file type or access type.
C02148	Failed to find group template name.
C02149	No feasible entries for infix operation %s.
C02150	Failed to find physical name.
C02151	Failed to find aggregate target.
C02152	Illegal qualified expression.
C02153	Aggregate target has a non-static name.
C02154	%s is illegal in target aggregate.

Number	Message Content
C02155	Subtype indication of allocator should not include resolution function.
C02156	Allocating an unconstrained array is not allowed.
C02157	%s is not an attribute declaration.
C02158	The ALL or OTHERS attribute was already specified for attribute %s.
C02159	Attribute %s was already specified for %s.
C02160	%s is not in the same declarative part.
C02161	%s is not in the specified attribute entity class.
C02162	The ALL or OTHERS configuration specification for component %s must be specified last.
C02163	Multiple configuration specifications apply to instances of component %s.
C02164	Failed to find char literal.
C02165	Incompatible range expression.
C02166	Range constraint in type definition must be locally static.
C02167	Range from %d up to %d is null.
C02168	Range from %d down to %d is null.
C02169	Failed to find bit string literal.
C02170	Using null waveform in a concurrent signal assignment is not allowed.
C02171	Assigning null waveform to a non-guarded signal is not allowed.
C02172	Choice must be a locally-static expression.
C02173	Component instances must be bound to the same entity.
C02174	Component all: %s was already configured in this block configuration.
C02175	Component configuration for component %s does not apply to component instantiations.
C02176	Block specification must include an architecture.
C02177	Block configuration name must match bound architecture.
C02178	%s must be a label on a block or generate statement.
C02179	Possible infinite loop: process does not have any wait statements.
C02180	Default binding does not exist in component.
C02181	Types do not match for %s %s.
C02182	%s %s is on entity used for default binding, but it is not on the component declaration.
C02183	The default binding usage for this component instantiation will result in an elaboration error.

Number	Message Content
C02184	Calling a procedure with a wait statement from a function is not allowed.
C02185	Signal assignments are not allowed in an entity statement part.
C02186	Guarded and unguarded signals cannot be used together.
C02187	Signal assignment must be guarded because the target is a guarded signal.
C02188	Delay in signal assignment must be ascending.
C02189	Ending a conditional signal assignment with a condition is not allowed.
C02190	Wait statements are not allowed in functions.
C02191	Wait statements are not allowed in a process with a sensitivity list.
C02192	Array case expression must have a static subtype.
C02193	Expression must return a discrete value.
C02194	OTHERS must be the final specified case choice.
C02195	Case statement only covers %d out of %d cases.
C02196	Loop parameters must be a discrete type.
C02197	Not a loop label: %s.
C02198	Exit or Next statement is not within loop %s.
C02199	Exit and Next statements must be inside a loop.
C02200	Return statement must be inside a subprogram.
C02201	Return statement must return a value.
C02202	Failed to return value from procedure.
C02203	The physical unit (%s) is not defined.
C02204	Range in physical type definition must be an integer.
C02205	Position number of %s exceeds range of physical type.
C02206	Element of array type is unconstrained.
C02207	Illegal type for array element.
C02208	Array indexes must be consistent.
C02209	Failed to access file type.
C02210	Declaring a file of multi-dimensional array type is not allowed.
C02211	Declaring a file of access type is not allowed.
C02212	Declaring a file of file type is not allowed.
C02213	Failed to find signature.
C02214	Missing full type definition for %s.
C02215	Deferred constant %s was not initialized.

Number	Message Content
C02216	Subprogram %s does not have a body.
C02217	Conversion functions are not allowed on subprogram signal parameters.
C02218	Formal %s is of mode OUT, so conversion functions are not allowed on actuals.
C02219	Actual for formal %s is not a variable.
C02220	The actual for parameter %s must denote a static signal name.
C02221	Calling a procedure with a wait statement from a process with a sensitivity list is not allowed.
C02222	%s
C02223	Actual for formal %s is not a globally-static expression.
C02224	Case statement only covers %d out of case range.
C02225	OTHERS is in unconstrained array aggregate.
C02226	Numeric value exceeds the INTEGER range.
C02227	%s is associated multiple times.
C02228	Component configuration does not have a primary binding.
C02229	Wrong attribute type was used to get value.
C02230	Generic \"%s\" cannot include range constraint within an association.
C02231	Type of generic \"%s\" is wrong.
C02232	Component instance \"%s\" has no mapping architecture.
C02233	No default binding for component '%s' (port '%s' is not on the entity).
C02234	Invalid black box identifier from %s.
C02235	Unmatched black box.
C02236	Unnamed black box.
C02237	Line-based black box specification cannot be placed inside block-based black box.
C02238	Protected type %s does not have full type definition.
C02239	Formal port '%s' declared in entity '%s' is not in the instantiated component declaration.
C02240	An expression of type %s is expected.
C02241	Failed to find the declaration type of %s.
C02242	Failed to check out VHDL license.
C02243	----- error(s) exist. Elaboration terminated.
C02244	Usage: [options ...] [files].
C02245	Options: .

Number	Message Content
C02246	-prelib <yes_or_no> preloads basic library (default: yes).
C02247	-lib <libname> specifies the name of the working library (default: work).
C02248	-fv <pack_file> specifies the file containing the list of design files.
C02249	-2000/93/87 enables vhdl-2000/1993/1987 (default: vhdl-1987).
C02250	Files: VHDL files to parse.
C02251	Illegal symbol (%s).
C02252	Illegal integer (%s).
C02253	Illegal string (%s).
C02254	Illegal name (%s).
C02255	Compiling entity %s.
C02256	Compiling architecture %s of %s.
C02257	Compiling configuration %s.
C02258	Compiling package %s.
C02259	Compiling package body %s.
C02260	Failed to find the library (%s).
C02261	Failed to find the entity %s.
C02262	Failed to find the package %s.
C02263	Failed to find the binding entry %s.
C02264	Failed to find the default binding entry %s.
C02265	Failed to find the declaration (%s).
C02266	Failed to find the jump label (%s).
C02267	Failed to find the physical unit (%s).
C02268	Ambiguous declarations (%s).
C02269	A signal declaration is expected.
C02270	A constant declaration is expected.
C02271	A constraint is expected.
C02272	A target is expected.
C02273	A label is expected.
C02274	Specifying a jump label for %s statement is expected.
C02275	An index constraint for array type definition is expected.
C02276	At least one element declaration for record type definition is expected.
C02277	A generic/port aspect in %s statement is not allowed.

Number	Message Content
C02278	A constraint in this type (%s) is not allowed.
C02279	A signature in an object alias declaration (%s) is not allowed.
C02280	A type definition in a none-object alias declaration (%s) is not allowed.
C02281	Failed to find subtype indication.
C02282	Index constraints may only be applied to array types.
C02283	A buffer or LINKAGE mode in parameters is not allowed.
C02284	Only signal interface declarations can be defined as guarded.
C02285	Failed to open the file (%s).
C02286	Failed to save the data %s.
C02287	Syntax error at %s.
C02288	Failed to find the package %s.
C02289	Failed to find the configuration %s.
C02290	Failed to find the package body %s.%s.
C02291	Opening %s from %s library.
C02292	Failed to open %s from %s library.
C02293	The wrong topmost entry (%s) of design was specified.
C02294	The name (%s) specified at the end is not consistent with (%s).
C02295	Failed to get the constraint in an object alias declaration (%s).
C02296	Failed to find the entity %s. %s is in instantiated statement (%s).
C02297	Failed to find the binding architecture %s. %s is in instantiated statement (%s).
C02298	Failed to find the default binding entry in instantiated statement (%s).
C02299	Failed to find the binding configuration %s. %s is in instantiated statement (%s).
C02300	Failed to find the entity %s.%s.
C02301	Failed to find the architecture %s.%s.
C02302	Failed to find the mapping architecture in configuration %s.%s.
C02303	Failed to find the subprogram declaration (%s).
C02304	Unknown object (%s) type in interface declaration.
C02305	Need a condition.
C02292	Failed to open %s from %s library. The library should be recompiled.
C02307	Subversion (%s in %s library) is too old. The library should be recompiled.

Number	Message Content
C02308	The %s in %s library is corrupted (%s). The library should be recompiled.
C02309	Error mapping format (%s) in mapping file.
C02310	-libcell compiles the same entities (architectures) as the symbol library.
C02311	-extractrtl extracts the RTL file after a entity (architecture) is compiled.
C02312	-verbosemsg prints verbose messages.
C02313	-onerrorstop stops compiling the design when an error occurs.
C02314	This file (%s) is not a VHDL file.
C02315	-rcFile specifies the resource file.
C02316	%s has a generic map error.
C02317	%s needs a generic map.
C02318	%s cannot specify any function call or type conversion in the formal part of the %dth generic association element.
C02319	%s cannot be unconnected or un-associated with the %d generic declaration.
C02320	%s's %dth generic declaration has a mapping error.
C02321	%s has a port map error.
C02322	%s needs port mapping.
C02323	%s cannot specify any function call or type conversion in the formal part of the %dth port association element.
C02324	%s cannot specify any function call or type conversion in the actual part of the %dth port association element.
C02325	%s cannot be unconnected or un-associated with the %d port declaration.
C02326	%s's %dth port declaration has a mapping error.
C02327	Labels (%s and %s) are not consistent.
C02328	-skip eapbc directs the compiler to skip (do not save) all.
C02329	-just eapbc directs the compiler to just (save) all.
C02330	Failed to read %s.
C02331	Failed to update %s.
C02332	-showWarn show the warning message.
C02333	Recursive calling path (Instance name: %s): \"%s\".
C02334	The number of scopes exceeds the maximum scope number.
C02335	Operator '%s' is ambiguous. Suitable definitions exist in packages '%s' and '%s'.
C02336	Configuration \"%s\" is the same as the entity name.

Number	Message Content
C02337	Failed to open the library %s because this library does not exist.
C02338	Failed to open the library %s because this library cannot be identified.
C02339	Failed to open the library %s because this library does not have access permission.
C02340	Resolved design unit '%s' at the instance '%s' to '%s' (using the Verilog configuration).
C02341	Library %s is saved.
C02342	"--synthesis translate_on" is not specified. "--synthesis translate_on" should be added.
C02343	Loading will be stopped because the loaded libraries use different VHDL language versions. Recompile the libraries with the -vhdl08 option.
C02346	Port map and generic map are not allowed if the entity aspect is OPEN.
C02347	Parameter value (%d) for attribute "%s" is greater than the array dimension (%d).
C02348	The protected code is not supported.
C02349	An error occurred in the encrypted source file. The detailed information about the error cannot be shown.
C02350	A warning occurred in the encrypted source file. The detailed information about the warning cannot be shown.

Appendix G: Gate-level Debug Mode

Overview and Usage

The Verdi platform provides full functionality for you to debug large gate-level designs. Before you can debug the design, the Verdi platform needs to spend some time importing the design source code for huge designs and then displaying the complex flattened schematic. When this is complete, you can then use different Verdi functions to debug the imported design.

However, in some cases, you are dealing with huge and flat gate-level designs and the design may contain more than a hundred million instances in a flatten scope. In this case, it is time consuming to open the huge schematic and it is also very difficult for you to do any debugging from this complex schematic. In this kind of scenario, instead of advance debug capabilities, you are more concerned about the performance – you just need basic debug capability, such as searching instance ports and tracing fan-in/fan-out – performance is the key concern.

To resolve the above issues, the Verdi platform provides a Gate-level Debug Mode that skips displaying the source code and full schematics, but provides the basic debug capability, such as searching instance ports and tracing fan-in/fan-out connections in a fast way. Instead of browsing the complex source code and huge schematics, this mode also provides a fast search mechanism for instance and ports so you can see the partial schematic and waveforms from the found instance ports quickly.

To use the Gate-level Debug Mode, add the `-fastGate` command-line option when invoking the Verdi platform. You can apply this mode only to a gate-level netlist, so that lots of syntax errors can be encountered if the `-fastGate` option is used with an RTL design as follows:.

```
% Verdi –fastGate –f run.f &
```

After the Verdi platform is invoked, the *Welcome* page is displayed. Because the `-fastGate` option is added and it can only be applied to a gate-level netlist, the

Work Modes icon is disabled, which means the work mode may not be changed. Click the **Hide** button in the upper right corner to close this *Welcome* page.

NOTE: Select the **Do not show this page again** option before hiding the *Welcome* page so that it does not appear the next time the Verdi platform is invoked.

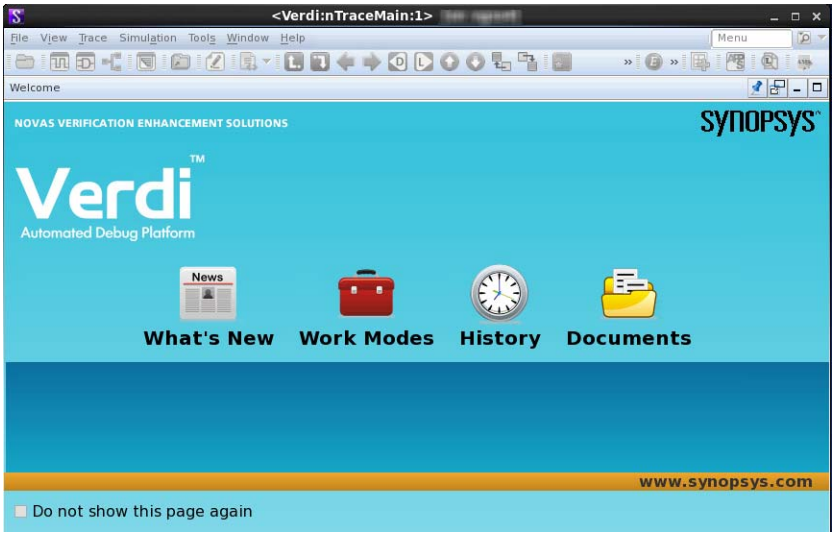


Figure: Welcome Page

After hiding the **Welcome** page, the top window of the Gate-level Debug Mode appears. The window contains the following four pane areas:

- The *Find Instances* pane in the left: This pane lists the matched instances.
- The *Port List* pane in the middle: This pane lists instance ports of the selected instance in the *Find Instances* pane.
- The *Trace Results* pane in the right: This pane shows the trace fan-in and trace fan-out results as a text grid.
- The *nWave* window in the bottom: This pane shows the related waveform from the FSDB file.
- The *Message* pane in the bottom: This pane shows the general messages (for example, the import messages) in the **General** tab and shows the compilation error and warning messages in the **Compile** tab.

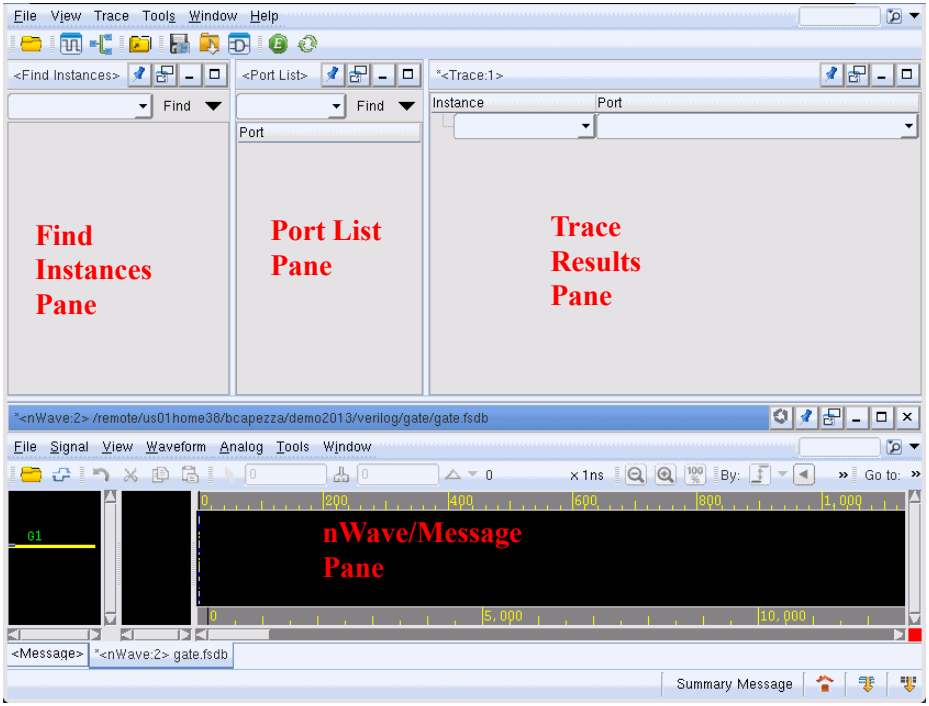


Figure: Gate-level Debug Mode Overview

The following sections explain how to perform gate-level debugging with these panes.

Find Instances Pane

The *Find Instances* pane is opened in the left side of the main window after the Verdi platform is invoked, as shown in *Figure: Gate-level Debug Mode Overview*.

To perform the string search, in the **Find** field enter the string to locate and press the **Enter** key on the keyboard or click the **Find** button. Wildcard characters are supported in the **Find** field. For example, use the asterisk character (*****) to specify zero or more alphanumeric characters, or use the question mark character (**?**) to represent a single alphanumeric character, or use the back slash character (\) to treat special characters as normal characters. For example, `\*abc` finds `*abc`, but not `abc`. Similarly, find `\a` is equal to find `a`.

Click the drop-down arrow beside the **Find** button. Two options are displayed: **Match Case** and **Match Substring**.

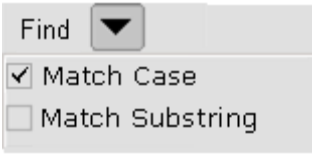


Figure: Find Options

- **Match Case:** When this option is turned on, a case sensitive search is performed. The default value is on.
- **Match Substring:** When this option is turned on, the search is applied to all substrings. The default value is off.

For example, if the **Match Substring** option is turned off and the `system.i_cpu` is entered and a search is performed, only the instance `system.i_cpu` is found as shown in the following figure.

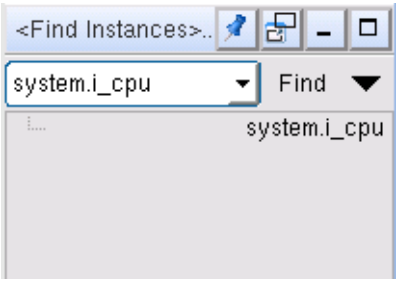


Figure: Match Substring - Off

If `system.i_cpu.*` is entered with the same setting (the **Match Substring** option is turned off) and then a search is performed, all instances under `system.i_cpu` is found as shown in the following figure. The instance under `system.i_cpu.i_ALUB` is not found unless the string is specified as `system.i_cpu.*.*`.

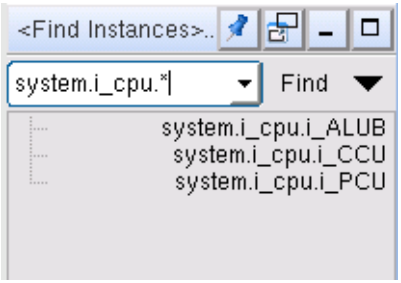


Figure: Results Using Wildcard Character

If the **Match Substring** option is turned *on* and then the `i_cpu` string is entered to perform a search, all instances that contain the `i_cpu` string are found. This

option can perform the search on all substrings without regard to the hierarchy delimiter. The following figure shows an example.

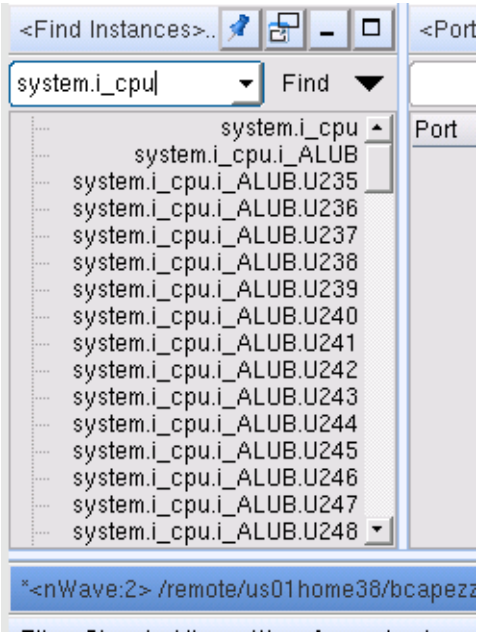


Figure: Match Substring - On

While searching the instances, a progress bar appears if there are too many instances. Click the **Cancel** button to stop searching anytime. If the **Cancel** button is clicked, searching is stopped and the instances that have already been found are listed.

By default, 5000 results are listed. The default number can be changed via the *NOVAS_FND_HIER_MORE_NUM* environment variable. If there are more results, the **More…** line is listed at the bottom. Click **More…** to view more results.

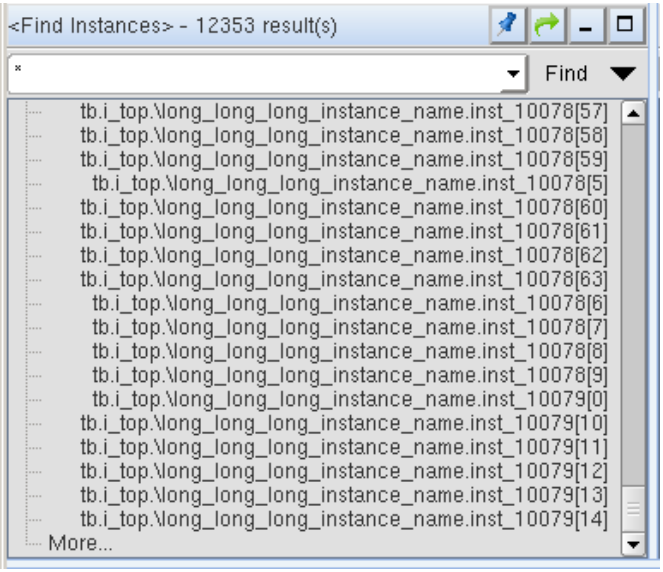


Figure: More Results

Find Instances Pane Right-Click Options

Some options are provided after the instance is found. Select the instance of interest and right click to display the options:

- **Show Selected in Port List:** Shows all ports of the selected instance in the *Port List* pane.
- **Schematic Flatten Window:** Opens a new flatten schematic pane to display the selected instance.
- **Delete Selected:** Removes the selected instance from the *Instance List* pane.
- **Save:** Saves all instances listed in the *Instance List* pane to a text file.
- **Load:** Loads the instance list from a saved file. A warning message is displayed if the loaded instance cannot be found in the design.

Port List Pane

The *Port List* pane appears in the middle of the main window after the Verdi platform is invoked as shown in *Figure: Gate-level Debug Mode Overview*. This pane shows the port list of the selected instance in the *Find Instances* pane. To view the port list, double-click the instance and select the **Show Selected in Port List** command right-click option, or drag and drop the instance from other panes.

A search mechanism is also provided in the *Port List* pane. Enter the string in the **Find** field and then click the **Find** button to perform the search. A type filter is provided in the pane to filter the port list by a specific type. Click the drop-down arrow beside the **Find** button to display four types of filters: **input**, **output**, **inout**, and **all**. By default, the **all** filter option is selected.

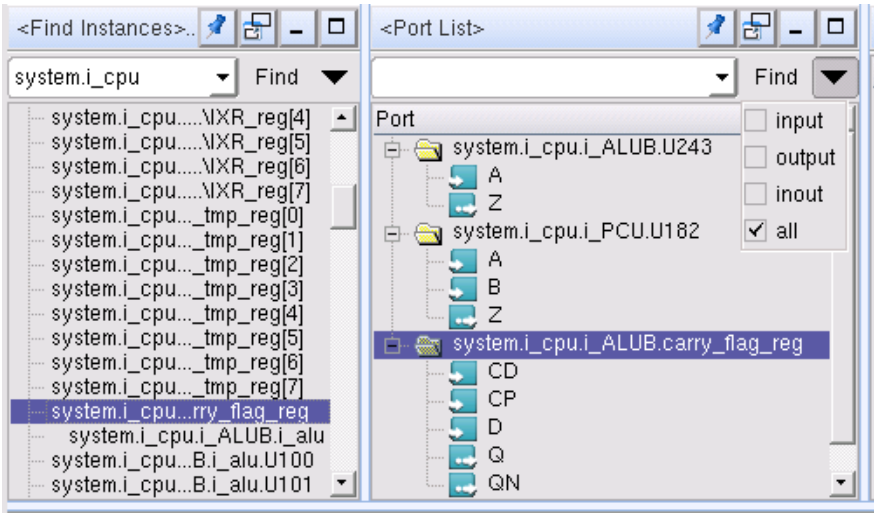


Figure: Find Options

If the FSDB file is loaded, the FSDB value can be annotated to the *Port List* pane. Turn *on* the **View -> Active Annotation** option in the main window and all values are annotated to the right of the port list. The value is synchronized with the cursor time in *nWave* window.

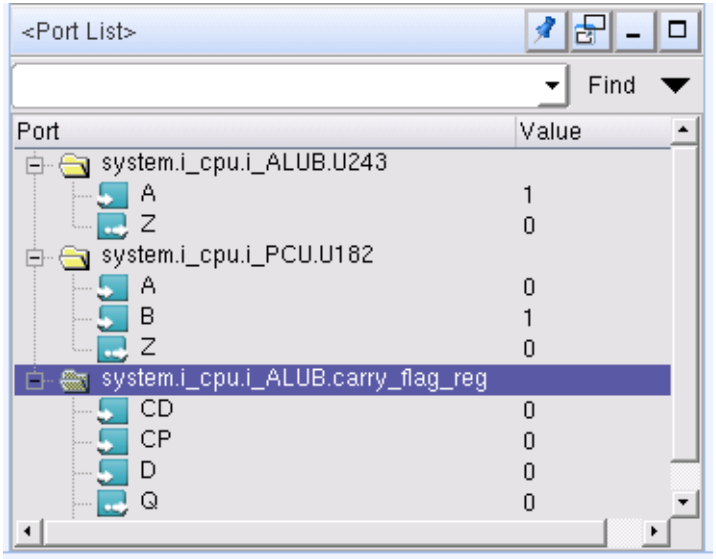


Figure: Active Annotation

Port List Pane Right-click Options

Some options are provided after the port is found. Select the port of interest and right click to display the options:

- **Report Fan-in Result:** Reports the fan-in cone leaf results in the *Trace Results* pane. The option is available if one and only one signal is selected. You can also double-click a port in the *Port List* pane to execute this option by default.
- **Report Fan-out Result:** Reports the fan-out cone leaf results in the *Trace Results* pane. The option is available if one and only one signal is selected.
- **Create Temporal Flow View:** Creates the *Temporal Flow View* pane from the selected port at the current time. Refer to the **Tools -> Temporal Flow View** option description for details.
- **Add Signal to Waveform:** Adds the selected signal to the *nWave* window. This option is enabled when the FSDB file is loaded. Refer to the **Signal** command description for details.
- **Delete All:** Clears all ports in the *Port List* pane.
- **Delete Selected:** Removes the root instance node and all its child nodes of the selected port or instance.

Trace Results Pane

The *Trace Results* pane appears in the right pane of the main window after the Verdi platform is invoked as shown in *Figure: Gate-level Debug Mode Overview*. This pane shows the traced results of the selected port. To view the traced results, invoke the **Report Fan-in Result** or the **Report Fan-out Result** options in the *Port List* pane. If the **View -> Active Annotation** option is on, the value in the loaded FSDB file is also be shown in this pane.

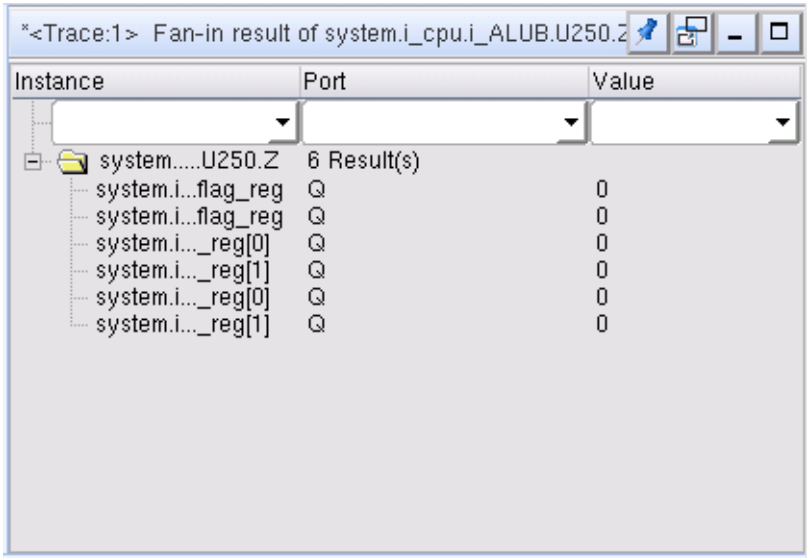


Figure: Trace Results Pane

There are three columns in the *Trace Result* pane: **Instance**, **Port**, and **Value** (only appears if the **Active Annotation** option is on and the FSDB file is loaded). Each column provides a search field so that the string or value can be entered to find the instances or ports of interest. The wildcard character is supported in these search fields.

Trace Results Pane Right-click Options

Some options are provided after the instance or port is found. Select the instance or port of interest and right click to display the commands:

- **New Schematic -> Partial Trace Path:** Shows the partial fan-in/fan-out cone from the traced instance/signal to the selected instance/signal in a new schematic pane. The new schematic pane is opened as a new tab in the same location as the *Trace Results* pane.

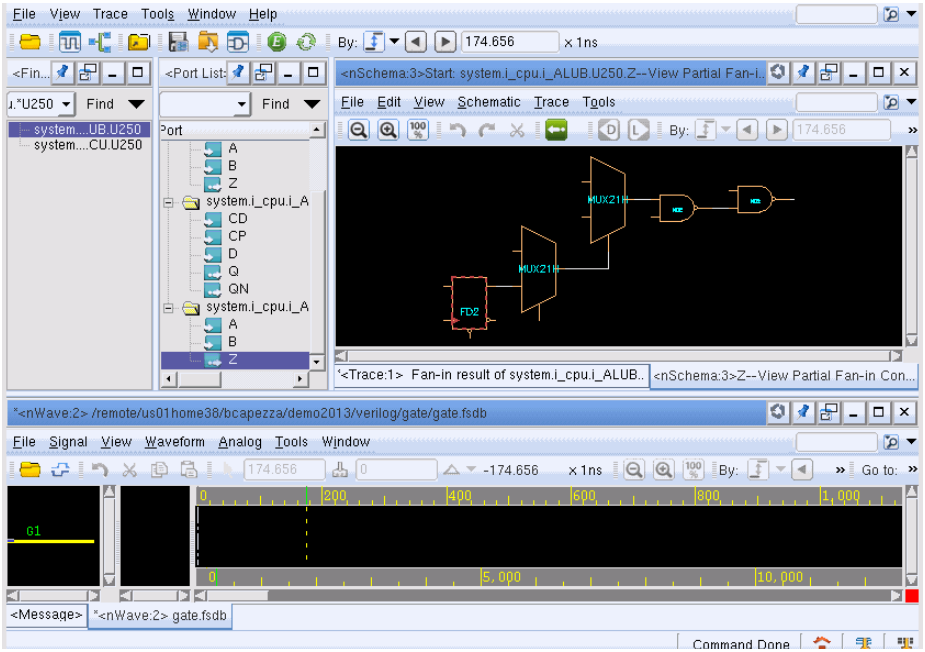


Figure: Partial Trace Path

- **New Schematic -> Flatten Window**: Shows the flattened window for the selected instance in a new flatten schematic pane. This option is only available when you select an instance. The new schematic pane is opened as a new tab in the same location as the *Trace Results* pane.
- **Create Temporal Flow View:** Creates the *Temporal Flow View* pane from the selected port at the current time. Refer to the **Tools -> Temporal Flow View** option description for details.
- **Add Signal(s) to Waveform:** Adds the selected signal to the *nWave* window. This option is enabled only when a port is selected and the FSDB file is loaded. Refer to the **Signal** command description for details.
- **Select All -> Ports:** Selects all ports in the *Trace Results* pane.
- **Select All -> Instances:** Selects all instances in the *Trace Results* pane.
- **Delete Selected:** Removes the selected row from the *Trace Results* pane.
- **Save Traced Result:** Saves the traced results to a text file.