

Using SimVision with AMS Simulator

Product Version 22.09

September 2022

© 2022 Cadence Design Systems, Inc. All rights reserved.

Cadence Design Systems, Inc. (Cadence), 2655 Seely Ave., San Jose, CA 95134, USA.

The AMS Designer Simulator product contains technology licensed from, and copyrighted by: Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, and other parties and is © 1989-1994 Regents of the University of California, 1984, the Australian National University, 1990-1999 Scriptics Corporation, and other parties. All rights reserved.

MMSIM contains technology licensed from, and copyrighted by: C. L. Lawson, R. J. Hanson, D. Kincaid, and F. T. Krogh © 1979, J. J. Dongarra, J. Du Croz, S. Hammarling, and R. J. Hanson © 1988, J. J. Dongarra, J. Du Croz, I. S. Duff, and S. Hammarling © 1990; University of Tennessee, Knoxville, TN and Oak Ridge National Laboratory, Oak Ridge, TN © 1992-1996; Brian Paul © 1999-2003; M. G. Johnson, Brisbane, Queensland, Australia © 1994; Kenneth S. Kundert and the University of California, 1111 Franklin St., Oakland, CA 94607-5200 © 1985-1988; Hewlett-Packard Company, 3000 Hanover Street, Palo Alto, CA 94304-1185 USA © 1994, Silicon Graphics Computer Systems, Inc., 1140 E. Arques Ave., Sunnyvale, CA 94085 © 1996-1997, Moscow Center for SPARC Technology, Moscow, Russia © 1997; Regents of the University of California, 1111 Franklin St., Oakland, CA 94607-5200 © 1990-1994, Sun Microsystems, Inc., 4150 Network Circle Santa Clara, CA 95054 USA © 1994-2000, Scriptics Corporation, and other parties © 1998-1999; Aladdin Enterprises, 35 Eyal St., Kiryat Arye, Petach Tikva, Israel 49511 © 1999 and Jean-loup Gailly and Mark Adler © 1995-2005; RSA Security, Inc., 174 Middlesex Turnpike Bedford, MA 01730 © 2005.

Associated third party license terms may be found at `install_dir/doc/OpenSource/*`

Trademarks: Trademarks and service marks of Cadence Design Systems, Inc. contained in this document are attributed to Cadence with the appropriate symbol. For queries regarding Cadence's trademarks, contact the corporate legal department at the address shown above or call 800.862.4522. All other trademarks are the property of their respective holders. Trademarks and service marks of Cadence Design Systems, Inc. contained in this document are attributed to Cadence with the appropriate symbol. For queries regarding Cadence's trademarks, contact the corporate legal department at the address shown above or call 800.862.4522. Open SystemC, Open SystemC Initiative, OSCI, SystemC, and Accellera Systems Initiative Inc. are trademarks or registered trademarks of Open SystemC Initiative, Inc. in the United States and other countries and are used with permission. All other trademarks are the property of their respective owners.

Restricted Permission: This publication is protected by copyright law and international treaties and contains trade secrets and proprietary information owned by Cadence. Unauthorized reproduction or distribution of this publication, or any portion of it, may result in civil and criminal penalties. Except as specified in this permission statement, this publication may not be copied, reproduced, modified, published, uploaded, posted, transmitted, or distributed in any way, without prior written permission from Cadence. Unless otherwise agreed to by Cadence in writing, this

statement permits Cadence customers permission to print one (1) hard copy of this publication subject to the following conditions:

1. The publication may be used only in accordance with a written agreement between Cadence and its customer.
2. The publication may not be modified in any way.
3. Any authorized copy of the publication or portion thereof must include all original copyright, trademark, and other proprietary notices and this permission statement.
4. The information contained in this document cannot be used in the development of like products or software, whether for internal or external use, and shall not be used for the benefit of any other party, whether or not for consideration.

Disclaimer: Information in this publication is subject to change without notice and does not represent a commitment on the part of Cadence. Except as may be explicitly outlined in such agreement, Cadence does not make, and expressly disclaims any representations or warranties as to the completeness, accuracy, or usefulness of the information contained in this document. Cadence does not warrant that use of such information will not infringe any third-party rights, nor does Cadence assume any liability for damages or costs of any kind that may result from the use of such information.

Restricted Rights: Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor.

Contents

About This Manual	7
Related Documents	7
Typographic and Syntax Conventions	7
1	10
Introduction to Cadence® SimVision™ Mixed-Signal Debug Option	10
Overview of SimVision Mixed-Signal Debug Option	10
Terminology	11
Components of SimVision MS	12
Supported Spectre Modes	14
Supported Languages	14
Benefits of Debugging Mixed-Signal Designs Using SimVision MS	15
Licensing	15
Managing Licenses	15
Invoking SimVision MS	16
From Command Line	16
From GUI	16
Managing Custom Buttons	17
Connecting with a Live Simulator	18
2	19
Using the Console Window	19
Controlling the Simulation from the Console Window	20
3	22
Working with Design Browser	22
Related Topics	24
Viewing the Mixed-Signal Design Hierarchy	24
Opening Multiple Databases	27
Using the Menus and Forms for AMS Designs	28
Loading an Analog or Mixed-Signal Database	29
Loading Unified Database	30
Loading an Analog Database	31
Setting Display and Formatting Preferences for Verilog-AMS Objects	32

Setting Formatting Options for Verilog-AMS Branches	33
Selecting Objects	35
Finding Edges	35
Searching Instances	35
Searching Modules or Subcircuits	36
Disabling Grouping of Library Cells	37
Sending Instances to Other Windows	38
4	40
Working with Mixed Net Browser	40
Related Topics	41
Defining Scope for Mixed-Signal Modules in Design Hierarchy	42
Viewing the Mixed-Signal Nets in the Defined Scope	43
Connect Module Instances Table	44
Customizing the Connect Modules Instances Table	45
Generating the Direct Waveform Plot	47
Plotting Waveform for a Connect Module	47
Plotting Waveform for a Mixed Net	48
Viewing Mixed Nets on Schematic Tracer	48
Highlighting the Conversion Point in Schematic Tracer	49
5	51
Working with Schematic Tracer	51
Related Topics	52
Setting Up Electrical Symbol Mapping	52
Built-In Symbols and Ports	55
Tracing the Critical Signal Path	58
Checking the Current Distribution	58
Sorting Currents	60
Understanding the Current Flow Direction	60
Checking the Current Distribution at Different Levels in the Hierarchy	61
Exporting Terminal and Current Data to a CSV File	64
Cross-Selecting Instances and Nets	66
Cross-Selecting Instances	66
Cross-Selecting Nets	69
Viewing Instances in Other Windows	71
6	73

Working with Source Browser	73
Debugging the Connect Module Source	74
Auto Scope Focus	75
Defining Breakpoints	76
Editing a Source File Using Your Own Editor	77
Searching for a Line Number in the Source Code	78
Searching for a Text String in the Source Code	79
Viewing the Analog Source	79
7	81
Working with the Waveform Window	81
Displaying Waveforms in the Waveform Window	82
Opening the SimVision Waveform Window	82
Using Analog Mnemonic Maps in SimVision	84
Creating Analog Mnemonic Map	84
Applying Analog Mnemonic Map	85
Limitations of Analog Mnemonic Maps	85
Example of Creating a Mnemonic Map	85
Searching Values for RNM or Electrical Signal	86
Sending Signals to ViVA	87
Enabling Real Value Probe Filtering	88

About This Manual

This user manual describes the features of Cadence® SimVision™ Mixed-Signal Debug Option and book contains the following chapters:

- [Introduction to Cadence® SimVision™ Mixed-Signal Debug Option](#)
- [Using the Console Window](#)
- [Working with Design Browser](#)
- [Working with Mixed Net Browser](#)
- [Working with Schematic Tracer](#)
- [Working with Source Browser](#)
- [Working with the Waveform Window](#)

Related Documents

For more information about SimVision and related products, consult the sources listed below.

- SimVision Introduction
- SimVision Analysis Environment
- SimVision: Using the Design Browser
- SimVision: Using The Schematic Tracer
- SimVision: Using the Source Browser
- SimVision: Using the Waveform Window
- Spectre AMS Designer and Xcelium Simulator Mixed-Signal User Guide
- Spectre Circuit Simulator Reference
- Spectre Classic Simulator, Spectre APS, Spectre X, and Spectre XPS User Guide

Typographic and Syntax Conventions

Special typographical conventions are used to distinguish certain kinds of text in this document. The formal syntax used in this reference uses the definition operator, `: :=`, to define the more complex elements of the Verilog-AMS language in terms of less complex elements.

- Lowercase words represent syntactic categories. For example,

`module_declaration`

Some names begin with a part that indicates how the name is used. For example,

`node_identifier`

represents an identifier that is used to declare or reference a node.

- Boldface words represent elements of the syntax that must be used exactly as presented (except as noted below). Such items include keywords, operators, and punctuation marks. For example,

endmodule

- The shortest permitted abbreviation is shown by capital letters but you can use either upper or lowercase letters in your code. For example, the syntax

-CHecktasks

means that you can type the option as `-checktasks`, `-CHECKTASKS`, `-ch`, `-CH`, `-cH`, and so on.

- Vertical bars indicate alternatives. You can choose to use any one of the items separated by the bars. For example,

```
attribute ::=  
    abstol  
    | access  
    | ddt_nature  
    | idt_nature  
    | units  
    | huge  
    | blowup  
    | identifier
```

- Square brackets enclose optional items. For example,
input declaration ::=
 input [range] list_of_port_identifiers ;

- Braces enclose an item that can be repeated zero or more times. For example,
list_of_ports ::=
 (port { , port })
- Code examples are displayed in constant-width font.

```
/* This is an example of the font used for code.*/
```
- Variables are in italic font, like this: *allowed_errors*.
- Keywords, filenames, names of natures, and names of disciplines are set in constant-width font, like this: keyword, file_name, name_of_nature, name_of_discipline.
- If a statement is too long to fit on one line, the remainder of the statement is indented on the next line, like this:

```
qgf = width*length*cfbb*(vgfs - wkf - qb/(2*cbb) -  
    (vgbs - vfbb + qb/(2*cob))) + qgf_par ;
```

Introduction to Cadence® SimVision™ Mixed-Signal Debug Option

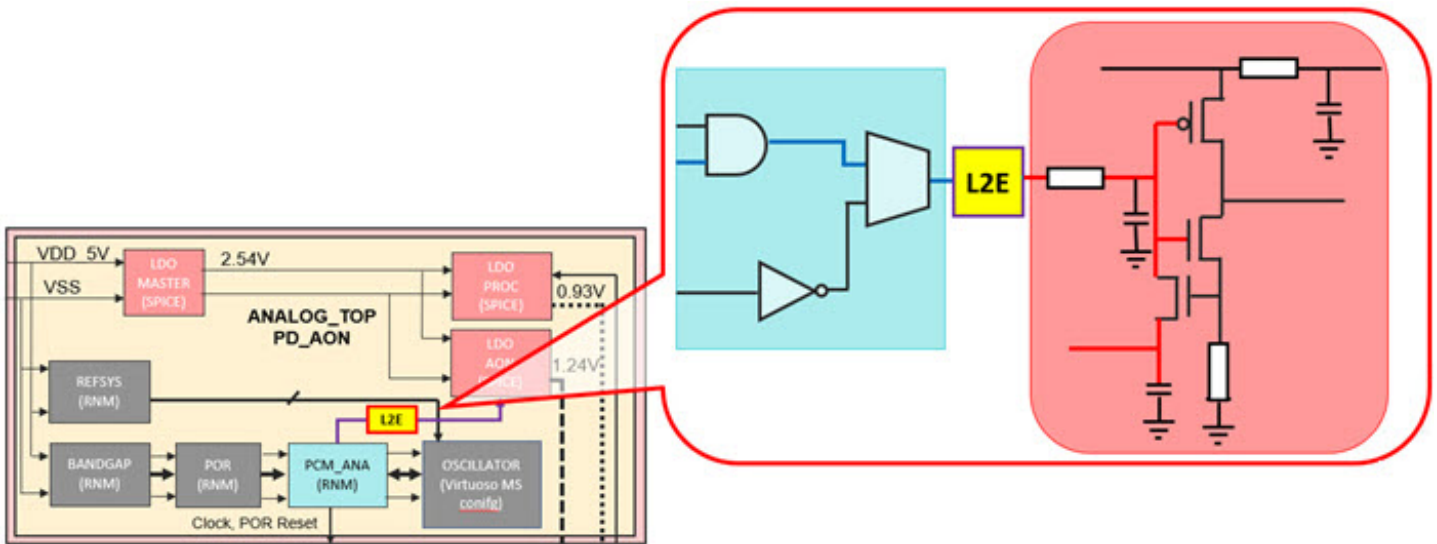
This chapter contains the following topics:

- [Overview of SimVision Mixed-Signal Debug Option](#)
- [Benefits of Debugging Mixed-Signal Designs Using SimVision MS](#)
- [Licensing](#)
- [Invoking SimVision MS](#)
- [Managing Custom Buttons](#)
- [Connecting with a Live Simulator](#)

Overview of SimVision Mixed-Signal Debug Option

SimVision, the unified graphical debugging environment for Cadence simulators, is now fully extended to the analog and mixed-signal simulations in the form of SimVision Mixed-Signal Debug Option (SimVision MS). With a comprehensive set of new features, SimVision MS lets you view and descend into the Spectre/SPICE netlists in the schematic view, explore the mixed-signal boundary, and disclose the port/terminal current flow at any time during a mixed-signal simulation.

Consider the following illustration where the mixed-signal boundary is connected with a connect module `L2E`. Using SimVision MS, you can go in to the source code of the connect module and then descend into the analog schematic to debug the analog circuit.



When you use the SimVision environment with the Spectre®AMS Designer simulator, you have access to additional controls to support Verilog®-AMS and other AMS-related design units.

See also "[Viewing Analog Data in the Waveform Viewer](#)" in the *SimVision: Using the Waveform Viewer* book.

⚠ For information on analyzing and debugging purely digital designs, see the [Running the SimVision Analysis Environment](#) book.

Terminology

The descriptions in this manual use terminology that might be new to you. The terms have to do with the way that the compiler simulates mixed analog and digital designs.

Term	Definition
analog solver	The part of the AMS Designer simulator that simulates the analog portions of a design. Some SimVision capabilities appear only when the analog solver is active.
digital solver	The part of the AMS Designer simulator that simulates the digital portions of a design. Some capabilities are enabled only when the digital solver is active.
analog context	The context of statements that appear in the body of an <code>analog</code> block.
digital context	The context of statements that appear in a location other than an <code>analog</code> block.

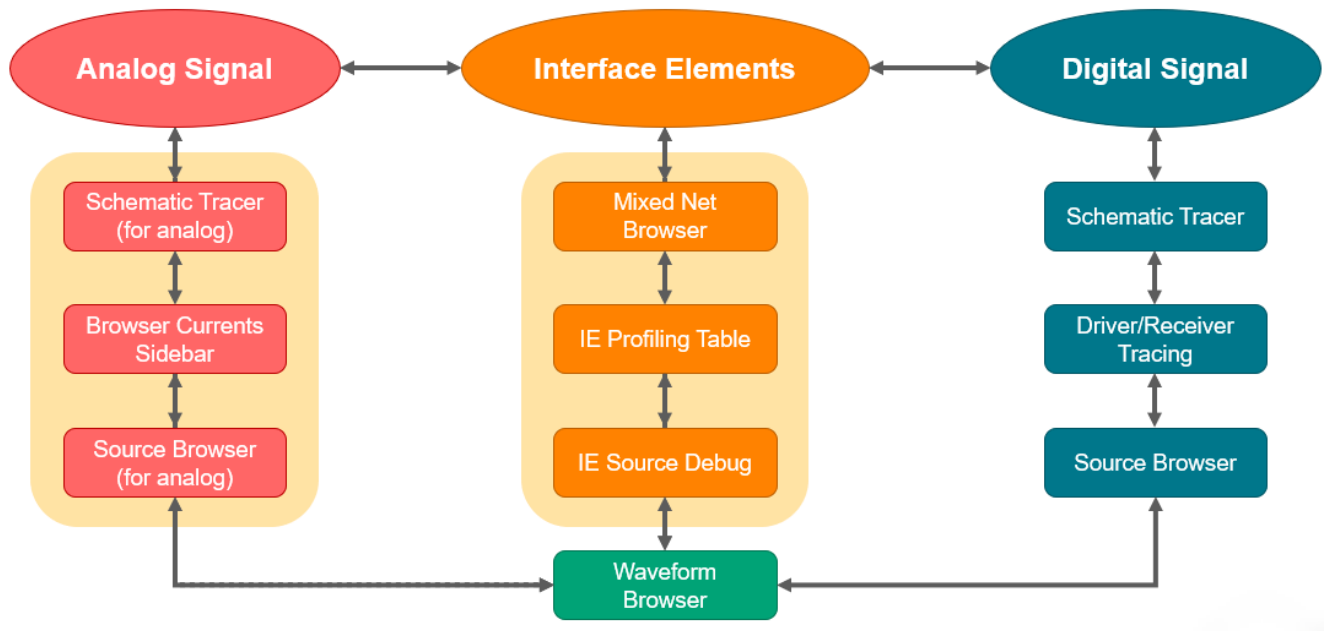
Components of SimVision MS

The following table lists the various windows/tools within SimVision MS that provide access to the analog and digital components of a mixed-signal design.

Window/Tool	Icon	Description
Design Browser		Shows the true design hierarchy tree with different icons and colors for mixed and electrical instances.
Schematic Tracer		Generates the Verilog-AMS/Spectre/SPICE context electrical schematic displaying the critical path.
Browse Currents Sidebar		Allows interactive current probing in analog context (Verilog-AMS/Spectre/SPICE) without requiring to probe the static current.

Mixed Net Browser		Displays detailed identification of the connect module instances. The connect module instances table is similar to profiling, it lets you sort and compare all the connect modules in the design.
Source Browser		Shows the source code of connect module instances and also displays the Spectre and SPICE nestlists.
Waveform Window		Plots simulation data along the X=time and Y axis. Data is usually shown as signal values versus time, but it can be any recorded data. It also has the capability to send a vector that is already plotted in SimVision to ViVA.

The following illustration displays how these tools interact with each other to give you a seamless debugging experience.



Supported Spectre Modes

SimVision MS is supported in the following Spectre Modes:

- Spectre X
- Spectre Fast APS
- Spectre APS
- Spectre Classic
- Spectre MS

Supported Languages

SimVision MS is supported with the following languages:

- SPICE
- Spectre
- Verilog
- VHDL
- SystemVerilog

- Verilog-AMS
- SystemC
- SystemVerilog-AMS

Benefits of Debugging Mixed-Signal Designs Using SimVision MS

With SimVision MS, you can start debugging your design right from where the issue is found. Following are the key benefits of SimVision MS:

- The elaboration-based database that provides a view of what is actually running in the session.
- Ability to interact with the simulator and control the simulation in real time.
- Intuitive environment and use model.
- Cross-referencing between various windows to perform extensive debugging.

Licensing

To run SimVision MS, add the following license with the existing license requirements:

Product number: 29010

License feature: SimVision™ Mixed-Signal Debug Option

Managing Licenses

The SimVision MS licenses works in User Host Display (UHD) mode. It uses a single license for any number of windows as long as it is the same UHD when launching each window. A check out is considered when SimVision connects to a MS debug enabled `xrun` session (with `-ams_edb` option). When SimVision disconnects from an MS debug enabled session, the licenses are checked back in.

For example:

- If you run SimVision MS with the following command:

```
% xrun ... -ams_edb -gui
```

`xrun` will run in the MS mode (`-ams_edb`), boot, and connect to SimVision (`-gui`). You may

disconnect at any time in the usual way.

- If you run SimVision MS with the following command:

```
% xrun ... -ams_edb
```

No licenses will be checked out until it is connected with SimVision MS at a later time because this command does not invoke GUI.

Invoking SimVision MS

The SimVision environment appears when you use the `-gui` option with the `xrun` or `xmsim` command or when you run the `simvision` command (either from the command line or from a Tcl file that you specify using the `-input` option to the `xrun xmsim`).

SimVision MS can be invoked from the command-line interface and GUI.

From Command Line

Run the following command to invoke SimVision MS from the command-line interface:

```
% xrun -f xrunArgs ... -ams_edb -gui
```



- SimVision MS only works in the flex mode.
- `-ams_edb` can be used without `-gui` so that no GUI will be invoked at the beginning. Later a standalone SimVision session can connect to a live Xcelium session with enabled mixed-signal debug features.

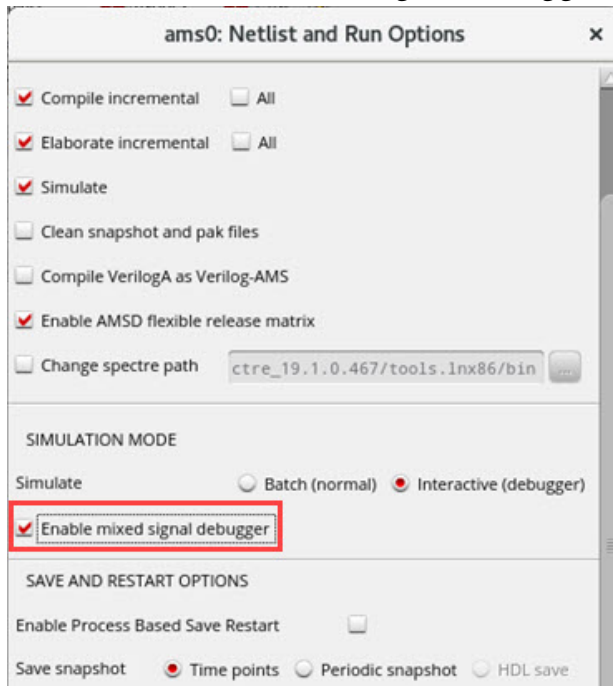
From GUI

Virtuoso Schematic Editor (VSE) and SimVision MS can interact with each other to display cross selection of the instances on the schematic. This feature is helpful when you have an analog schematic open in VSE and want to understand where exactly are its components placed on the mixed-signal net.

Perform the following steps to invoke SimVision MS from ADE Explorer and enable VSE and Simvision MS cross selection:

1. In ADE Explorer, select *Simulation - Netlist and Run Options*.
The Netlist and Run Options form appears.

2. In the *SIMULATION MODE* group box
 - a. Select the *Interactive (debugger)* option.
 - b. Select the *Enable mixed signal debugger* option.



⚠ The *Enable mixed signal debugger* option is available in IC6.1.8/ICADVM20.1 ISR15 and later releases.

3. Click *OK* to close the form.
SimVision MS will be launched in the background.

Managing Custom Buttons

In SimVision, you can create custom buttons for commands and add them to the toolbar. You can assign one or multiple Tcl commands to a custom button. Customized buttons can automate your debugging tasks by letting you execute a series of Tcl commands with one click.

You can

- Create buttons that perform a specified function
- Edit the buttons to change the functions they perform
- Reorder the buttons on the toolbar

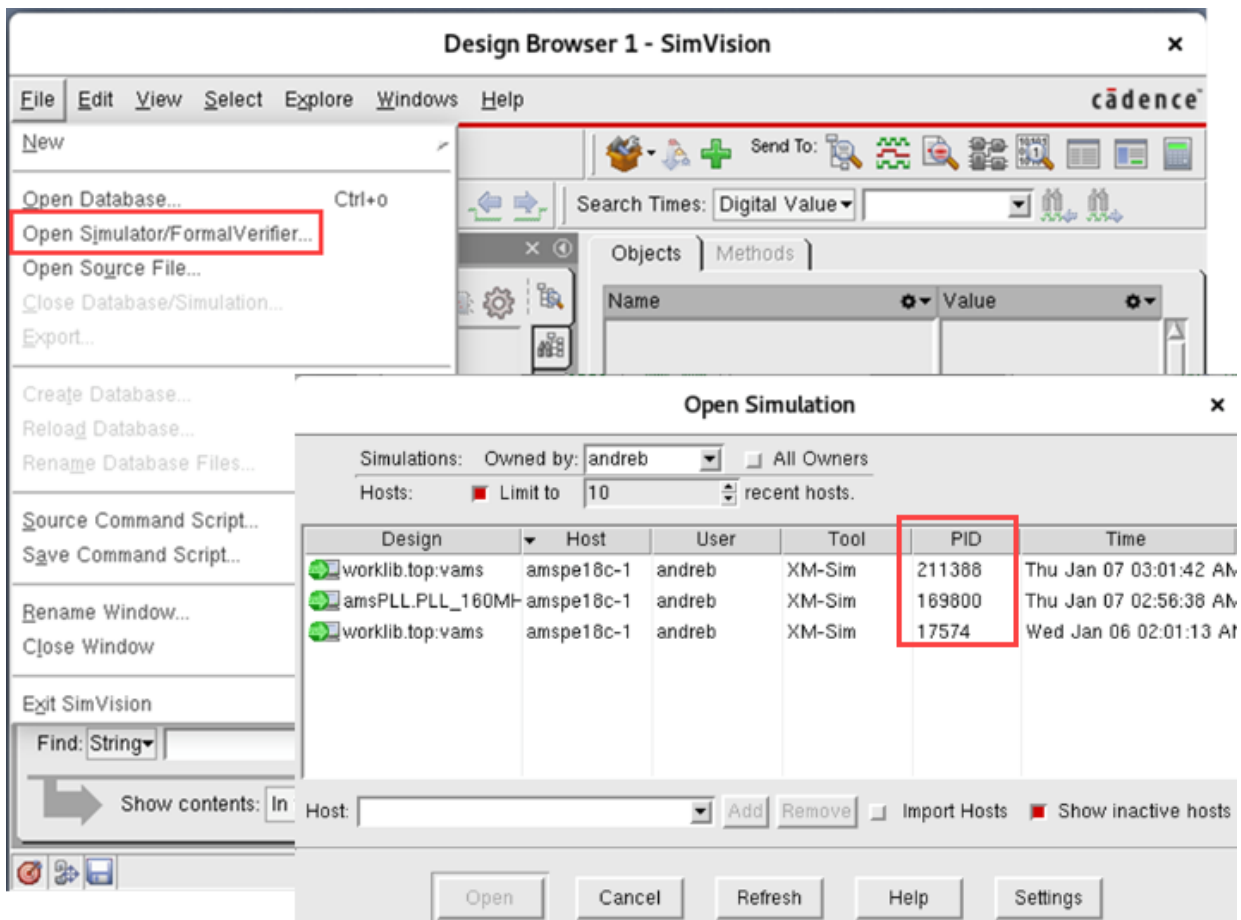
- Export the buttons to save them in a file
- Import a file of user-defined buttons

See "[Customizing Toolbars](#)" in the *Introduction to SimVision* book for details and examples.

Connecting with a Live Simulator

SimVision MS provides the ability to connect with the live simulators using a process ID. To do so:

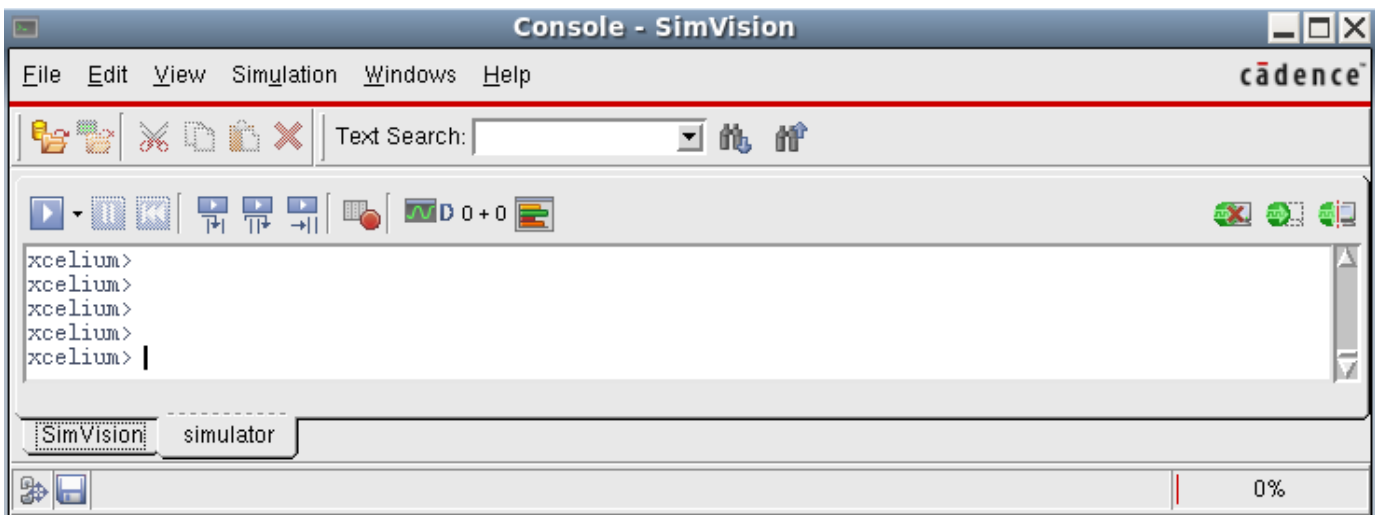
1. Select *File - Open Simulator/FormalVerifier*.
The Open Simulation form is displayed.
2. Select the required process ID from the *PID* column and click *Open*.
The selected process will be opened.



Using the Console Window

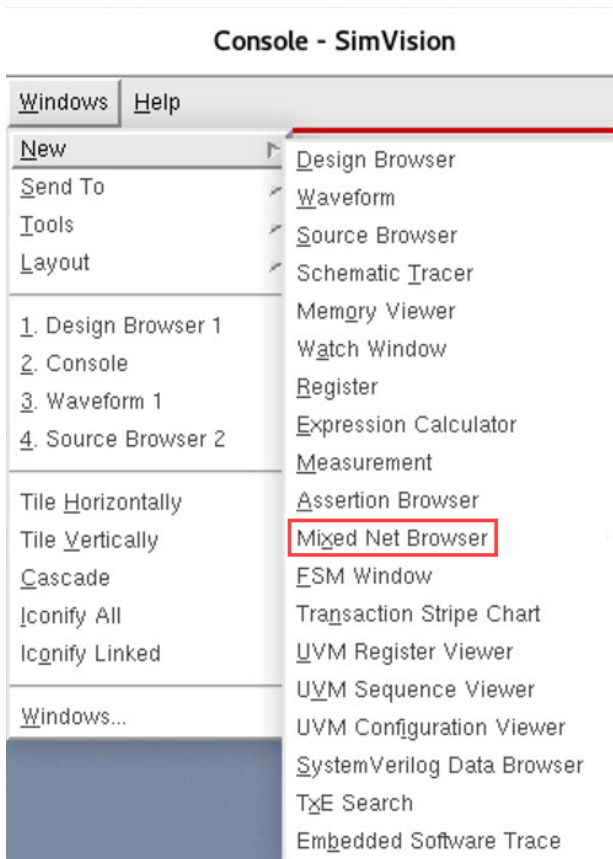
The Console window gives you access to the command languages for SimVisionMS and for the simulator you are running.

Each tool is represented by a tab in the Console window. For example, if you run SimVisionMS with the simulator, the Console window contains a *SimVision* tab and a *simulator* tab, as shown in figure below.



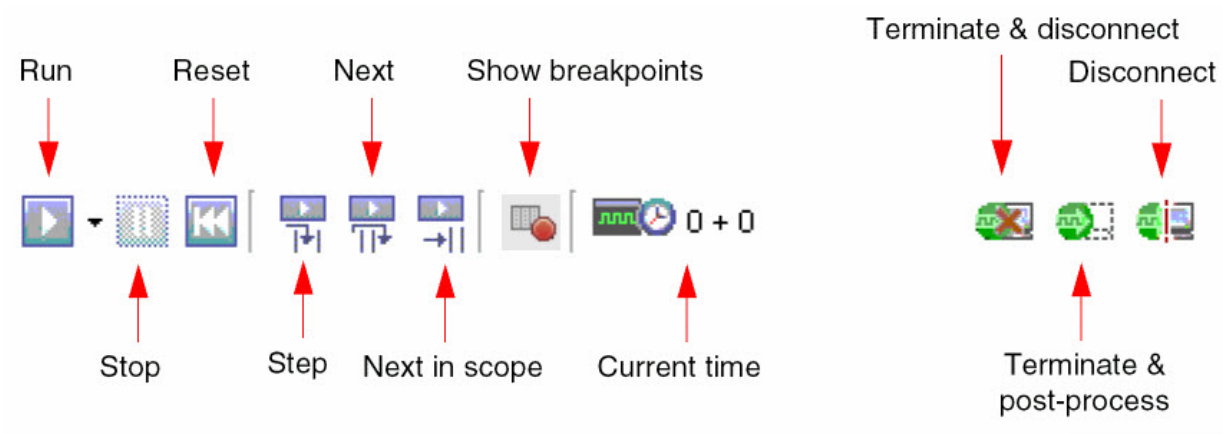
You can type simulator, SimVision, or Tcl commands on the *SimVision* tab, and monitor the messages returned by the tool.

In SimVision MS, the *Windows>New* menu of the Console window has an additional option, *Mixed Net Browser*.



Controlling the Simulation from the Console Window

The Console window contains a simulation toolbar, which lets you control the simulation, as shown in the figure below. It is the only unconfigurable toolbar in the window. That is, you can add and remove other toolbars, or remove individual buttons from other toolbars, but not in the simulation toolbar.



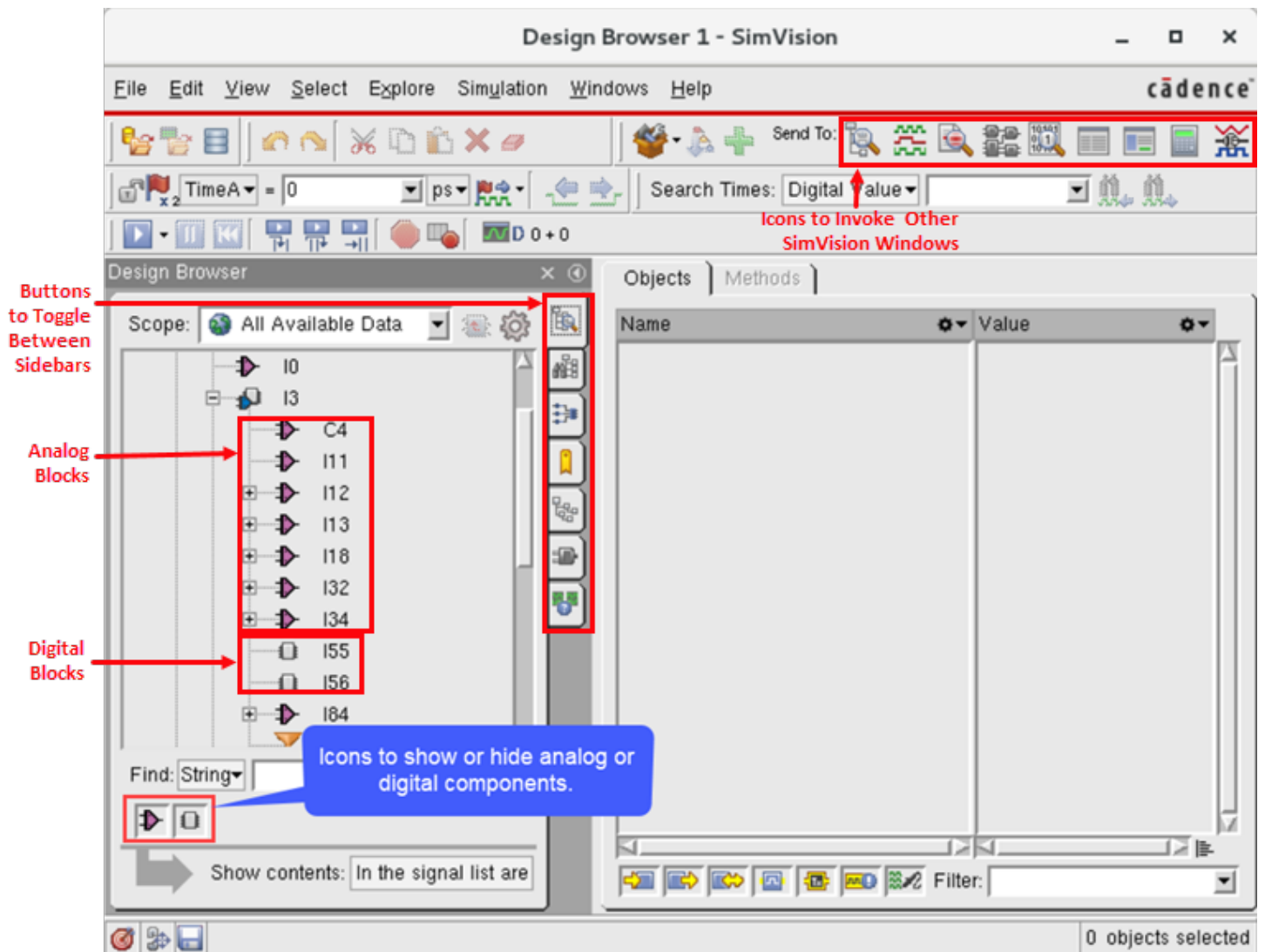
Refer to the [Introduction to SimVision](#) manual for detailed information about the Console window.

Working with Design Browser

The Design Browser window presents a graphical display of your design and provides access to the other SimVision MS windows.

The Design Browser window lets you access the objects in your design. You can use it to locate the signals and variables that you want to debug. You can also select objects in Design Browser and send to the Schematic Tracer or the Waveform window for further analysis.

Design Browser in SimVision MS is as shown in the following figure.



Design Browser contains two primary panes. On the left is the Scope Tree pane, which displays your current design hierarchy in a graphical tree representation. On the right is the Signal List pane, which displays a list of signals with their current simulation values.

Related Topics

- [Viewing the Mixed-Signal Design Hierarchy](#)
- [Using the Menus and Forms for AMS Designs](#)
- [Setting Display and Formatting Preferences for Verilog-AMS Objects](#)
- [Selecting Objects](#)
- [Searching Instances](#)
- [Searching Modules or Subcircuits](#)
- [Disabling Grouping of Library Cells](#)
- [Sending Instances to Other Windows](#)

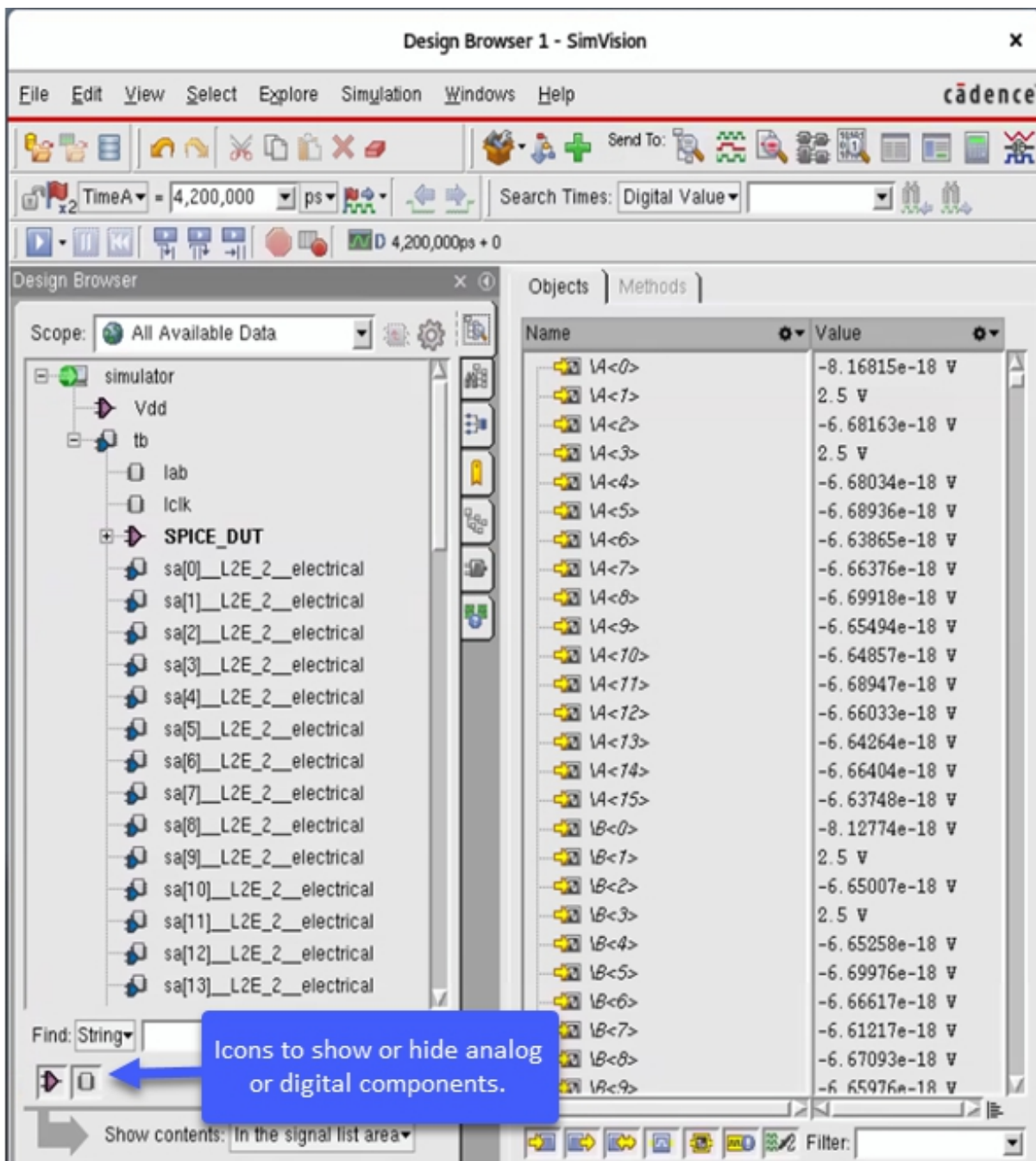
Viewing the Mixed-Signal Design Hierarchy

When you load a database, the design hierarchy is listed in the Design Browser sidebar. A design can contain pure analog, pure digital, or mixed-signal designs. Therefore, the tool uses different icons to identify the type of each design component listed in the Design Browser sidebar.

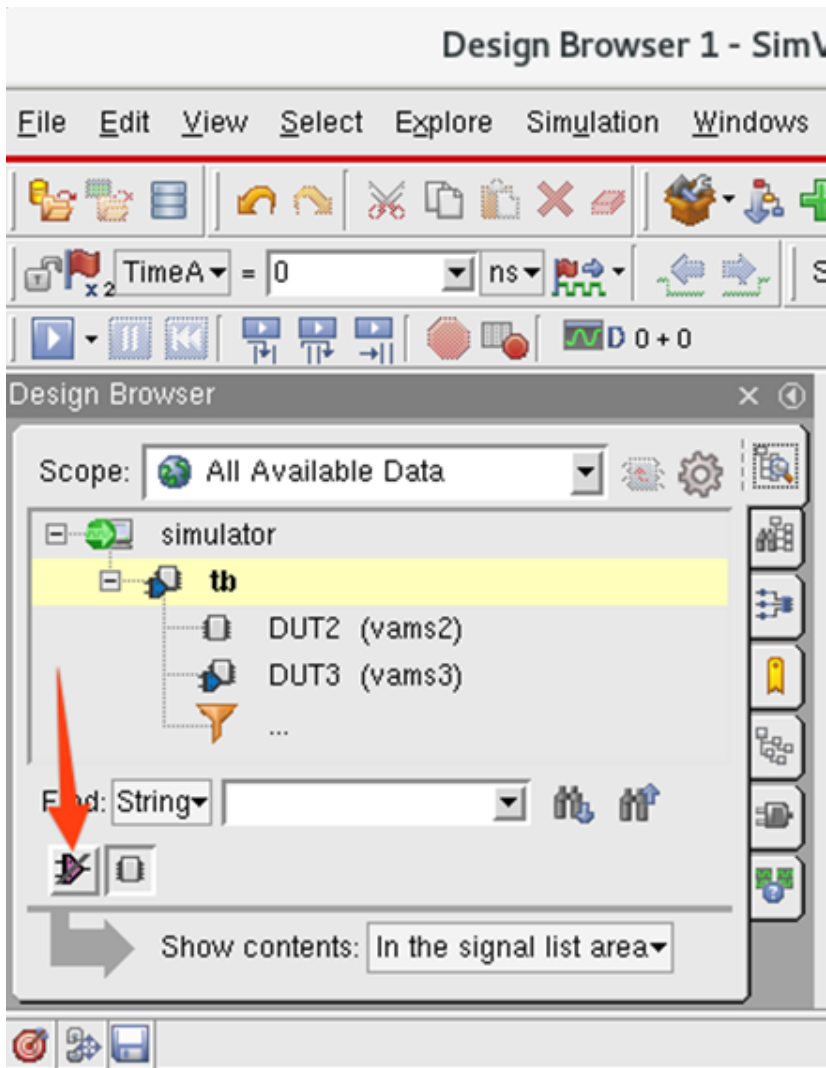
The following table describes the icons:

	Indicates the top of a design hierarchy and is named <i>simulator</i> .
	Represents mixed-signal instances in the design.
	Represents the digital instances in the design.
	Represents the analog instances in the design.

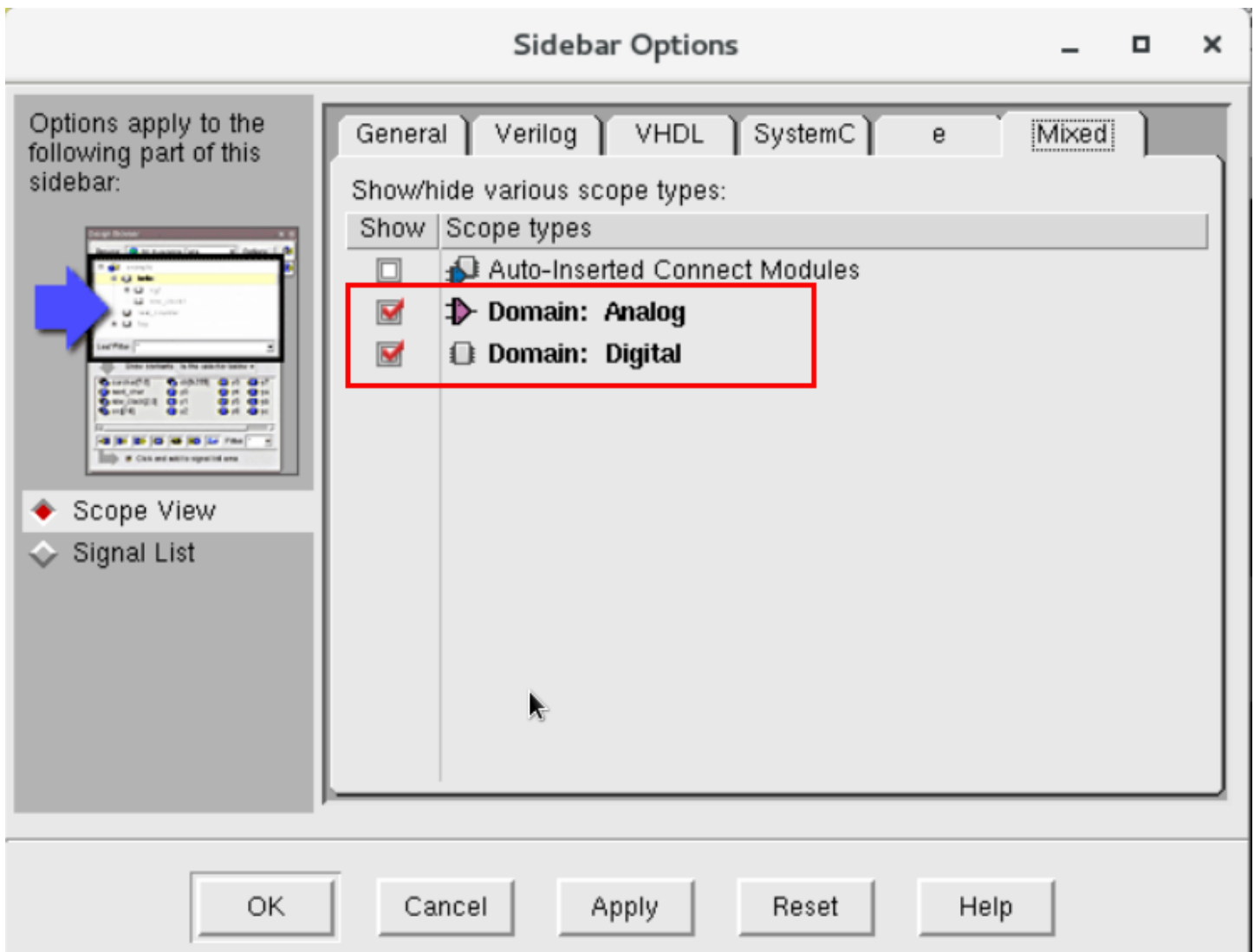
By default, the Design Browser sidebar displays all types of design component. You can use the *Show/Hide Analog* or *Show/Hide Digital* icons in the sidebar to selectively view only analog or only digital components.



The following example shows digital components filtered out.



Alternatively, you can change the default visibility setting for analog and digital components by selecting or deselecting the check boxes on the *Mixed* tab in the *Sidebar Options* form.



Opening Multiple Databases

You can connect to only one simulation at a time. Therefore, by default, the Design Browser sidebar displays a single hierarchy.

When running a simulation, you might probe objects to one or more databases. These databases store all the information about the objects in the simulator design hierarchy.

You might also open any number of databases that contain the results of previous simulations. These databases are not connected to the simulator.

Related Topics

- [Loading an Analog or Mixed-Signal Database](#)

Using the Menus and Forms for AMS Designs

To support Verilog-AMS/Spectre/SPICE, the menu choices in Design Browser (and in the other SimVision windows) for the AMS Designer simulator differ from the choices available for purely digital simulators. See the comments and cross-references in the following table. (For information about other SimVision menu selections, see the [Introduction to SimVision](#) book.)

SimVisionMS Window Menu Choices

Menu item	Comments and cross-references
<i>Edit - Preferences</i>	See Setting Display and Formatting Preferences for Verilog-AMS Objects .
<i>Select - Branches</i>	See Selecting Objects .
<i>Simulation - Advance - To Synchronization Point</i>	Run the simulator until the digital solver gains control from the analog solver. You can use this command to leave the analog solver and return to the digital solver. (Some commands, such as Simulation - Create Force, are available only when the digital solver is active.) Note: The simulator stops at any breakpoint it encounters before it reaches the synchronization point.
<i>Simulation - Advance - Timepoint</i>	Simulate up to the beginning of the next analog timepoint or to the next time at which a digital event is scheduled. Note: The simulator stops at any breakpoint it encounters before it reaches the specified timepoint.
<i>Simulation - Reset to Start</i>	Supported for RNM designs. For more details of that feature, please refer to <i>SimVision User Guide</i> .

<i>Simulation - Set Breakpoint - Time</i>	For the AMS Designer simulator, you can specify a time breakpoint at any time: You do not have to use an integer to specify the number of time units. For more information on setting time points, see "Setting and Managing Breakpoints," in Chapter 8, "Setting and Managing Breakpoints," in the <i>SimVision User Guide</i>.
<i>Simulation - Deposit Value</i>	You cannot deposit values to analog quantities, including analog nets, ports, variables, or branches. Also, you cannot deposit values to digital quantities while the analog solver is active. For more information about using Deposit Value for digital quantities, see Chapter 9, "Changing and Monitoring the Value of an Object During Simulation," in the <i>Cadence SimVision User Guide</i>.
<i>Simulation - Create Probe</i>	Probe values of digital and most analog objects to a database. Note: To probe the values of currents and Spectre primitives, you must use Tcl probe commands. For more information about using Create Probe for digital quantities, see Chapter 7, "Creating and Managing Probes," in the <i>Cadence SimVision User Guide</i>.
<i>Simulation - Create Force</i>	Only available when the digital solver is active. You cannot force a value for an analog quantity, including analog nets, branches, and analog variables. Also, you cannot force values for digital variables and signals while the analog solver is active. For more information about using Create Force for digital quantities, see Chapter 9, "Changing and Monitoring the Value of an Object During Simulation," in the <i>Cadence SimVision User Guide</i>.

Loading an Analog or Mixed-Signal Database

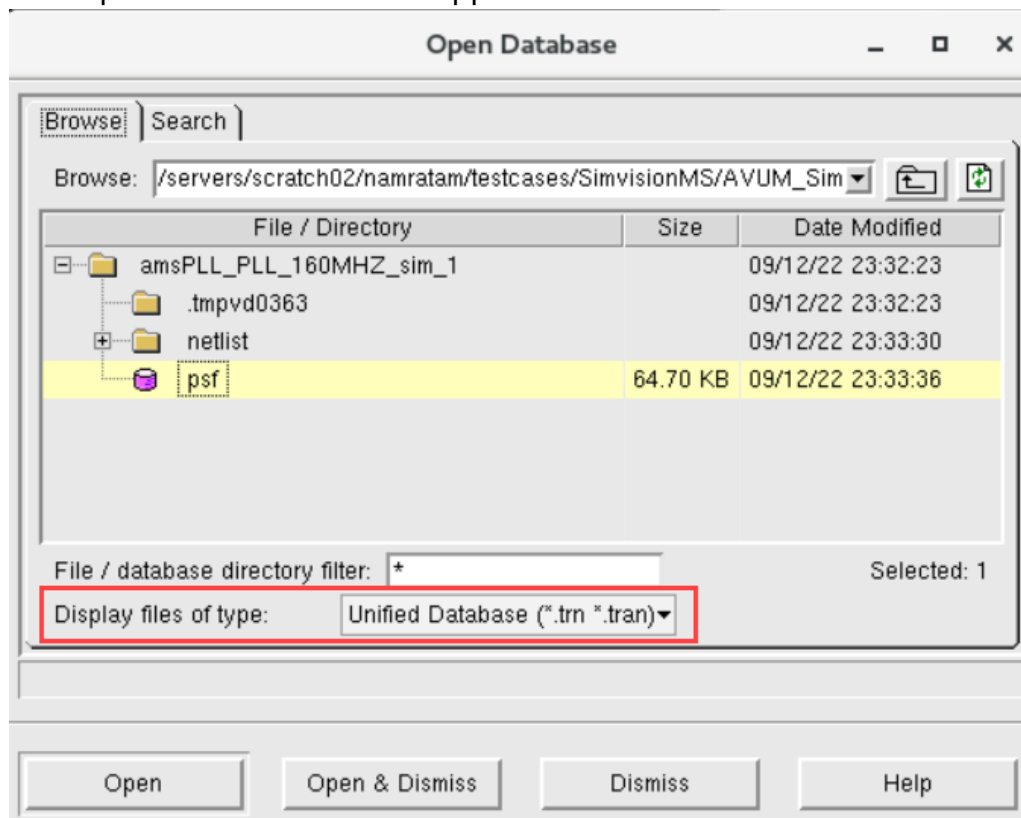
The SimVision environment supports a unified PSFXL/SST2 database that contains both SST2 digital data and PSFXL analog data so that you do not need to manually segregate analog probes in a separate database when probing or opening databases to view the results. However, if required, you can also load data for analog and digital components separately.

Loading Unified Database

The unified database stores the results in the psf directory that contains two containers. The digital signal results are stored in a SST2 container (*.trn) and the analog signal results are stored in an analog container (*.tran). The logFile, which is created during the simulation, links these two containers in a unified database.

To open both PSFXL and SST2 database for your design:

1. Select the *File - Open Database* menu option or click the *Open an SST2 database* icon. The Open Database window appears.



The default file type selected in the *Display files of type* drop-down list is *Unified Database*. This file type displays the psf directory from which both *.tran and *.trn types of files are opened.

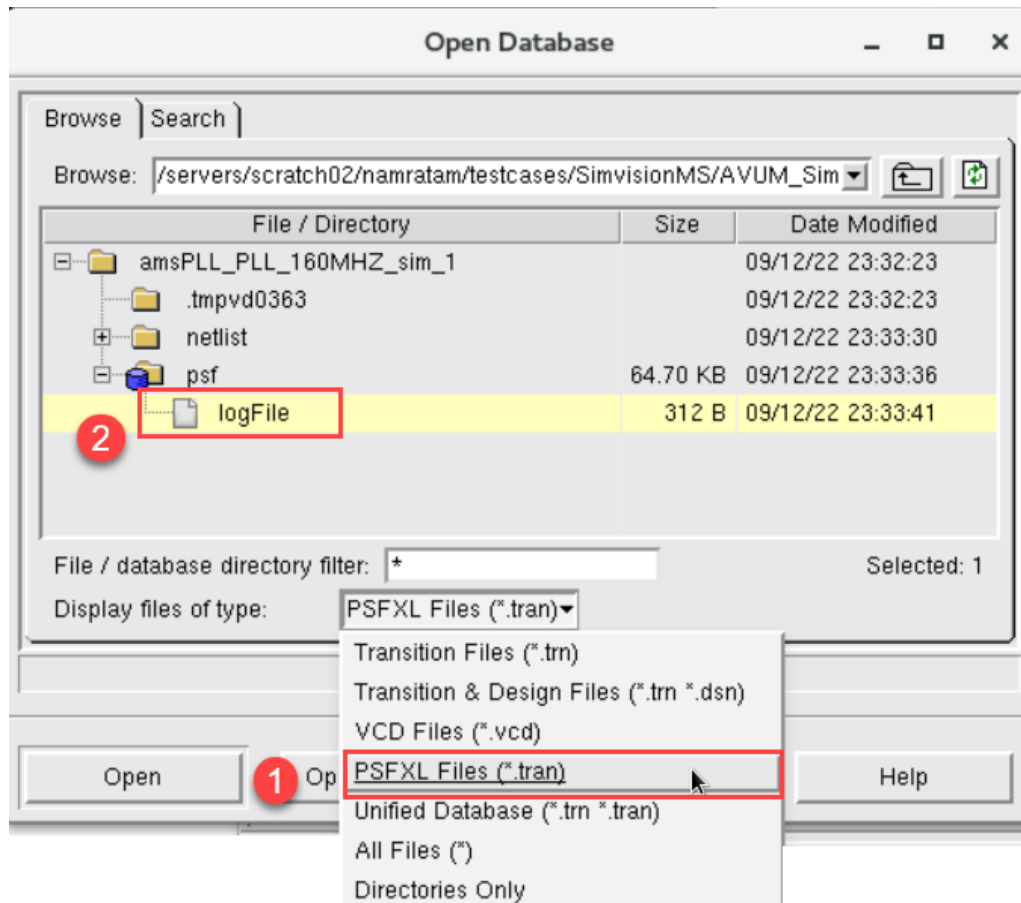
2. Select the directory containing the results.
3. Click *Open & Dismiss*.

The unified database containing both the containers is displayed in SimVisionMS.

Loading an Analog Database

To open the database containing only the analog container in SimVisionMS, expand the results directory in the Open Database form and do the following.

1. Choose *PSFXL Files (*.tran)* from the *Display files of type* drop-down list.
2. Select `logFile` that is created during simulation.

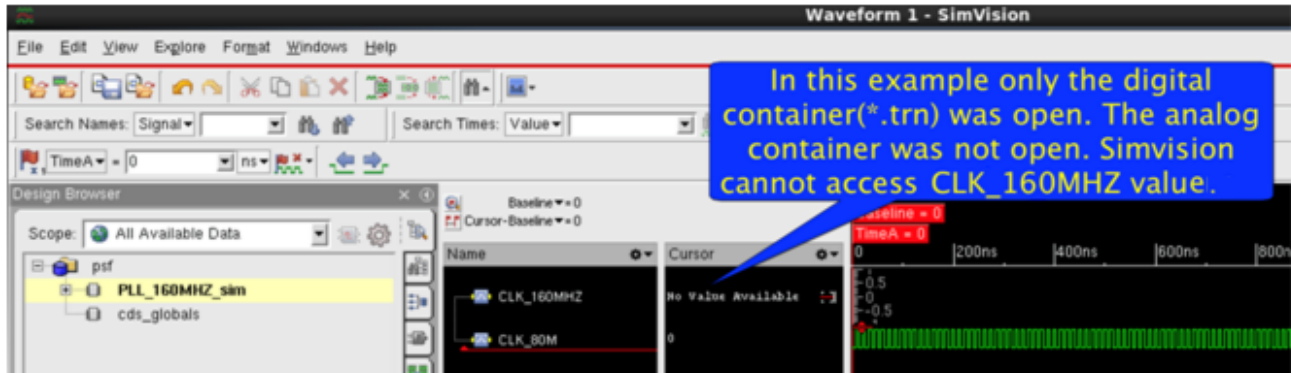


3. Click *Open & Dismiss*.

The database containing only the analog container is displayed in SimVisionMS.

If you have selected PSFXL as the file type, but the Spectre solver option `rawfmt` is set to `sst2` in the `.scs` file, the tool reports a DBOPEN error indicating that the file format is not supported. Ensure that the `rawfmt` setting is in sync with the type of data saved by the simulation.

⚠ Though it is possible to open only the simulation digital (*.trn) container results during post processing by selecting *Transition Files (*.trn)* from the *Display files of type* drop-down list, it is not recommended, because in this case, SimVisionMS is not able to plot the analog vector values that are stored in the analog container. For such cases, SimVisionMS displays *No Value Available*, as shown below.



You need to specify the `-amsformat <psfxl | psfxl_all> xmsim` or `xrun` option to enable PSFXL/SST2 storage for analog probes.

Related Topics

- [Working with Design Browser](#)

Setting Display and Formatting Preferences for Verilog-AMS Objects

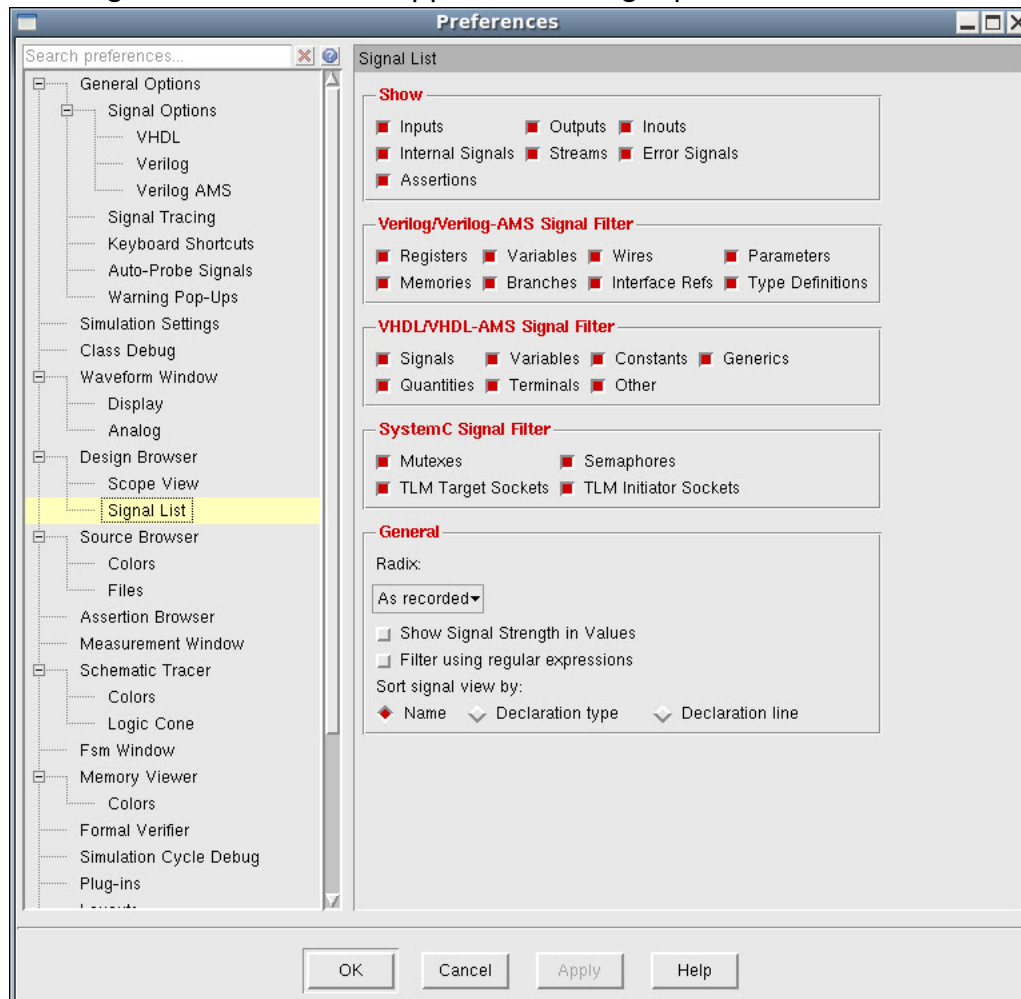
Items available on the Signal List pane for the AMS Designer simulator include analog branch objects.

To specify the signal list preferences, do the following:

1. From the Design Browser, choose *Edit - Preferences*.
The Preferences form appears.

2. Select *Design Browser - Signal List* in the left pane.

The Signal List selections appear on the right pane of the form.



3. On the right pane of the form, select your preferences.

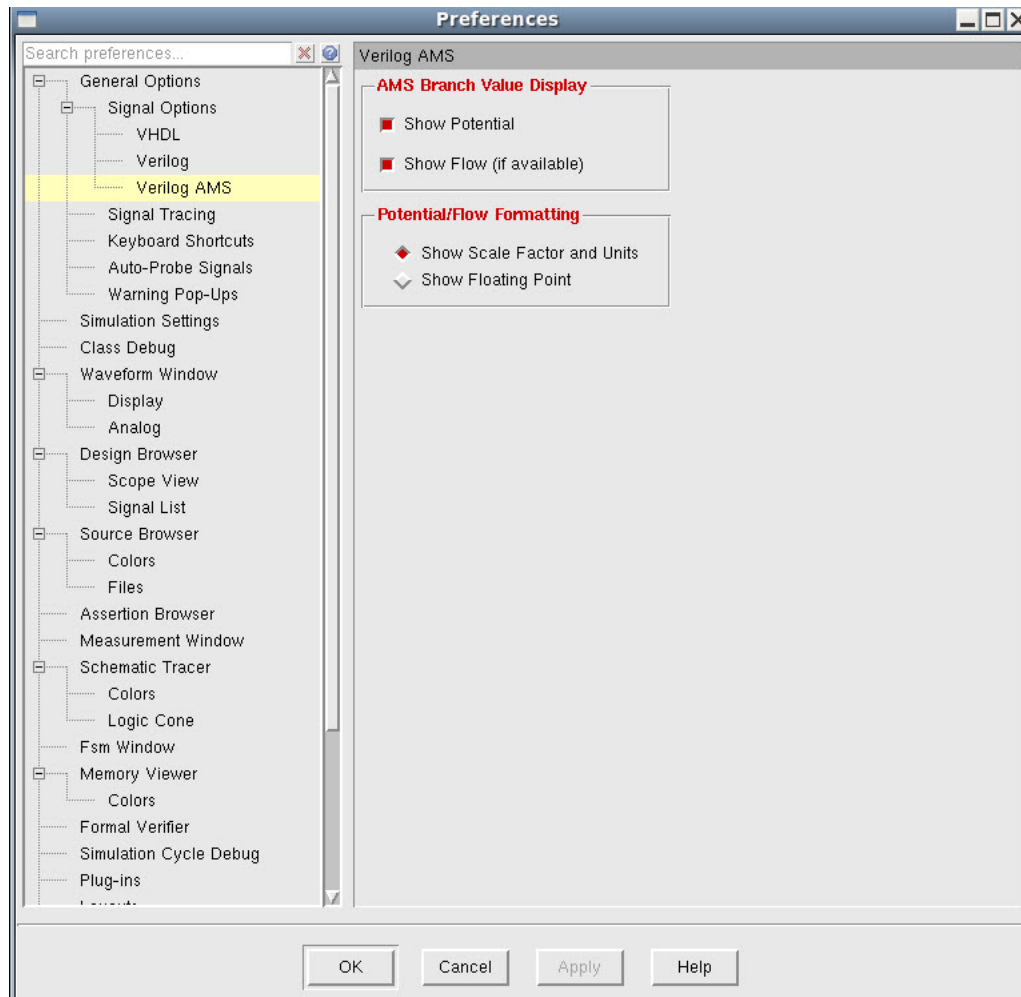
4. Click *Apply*.

Design Browser modifies the Signal List pane to show the object types you select.

Setting Formatting Options for Verilog-AMS Branches

To specify formatting options for Verilog-AMS branches, do the following:

1. From the Design Browser, choose *Edit - Preferences*.
The Preferences form appears.
2. Under General Options (on the left side of the form), select *Signal Options - Verilog AMS*.
The Verilog AMS preferences appear on the right side of the form.



3. In the AMS Branch Value Display group box, select the options you want:

<i>Show Potential</i>	Show potential quantities, such as 5.44V
<i>Show Flow</i>	Show flow quantities, such as 3.45mA

4. In the Potential/Flow Formatting group box, select one of the following format options:

<i>Show Scale Factor and Units</i>	Use scalar multipliers and nature units; for example: <code>result=-30.646mV</code>
<i>Show Floating Point</i>	Use a floating-point format; for example: <code>result=-0.0306461</code>

5. Click *Apply*.

Selecting Objects

To select branches in addition to the objects available for a Verilog module, do the following:

In the Design Browser, choose *Select - Branches*.

In the Source Browser, the left parenthesis enclosing each branch appears highlighted, as illustrated here:

```
// A digital event to which analog is made sensitive to
@(posedge(trigger))
    value = V(inSig);

V(holdSig) <+ transition( value, 1n, 2.5n);
```

Finding Edges

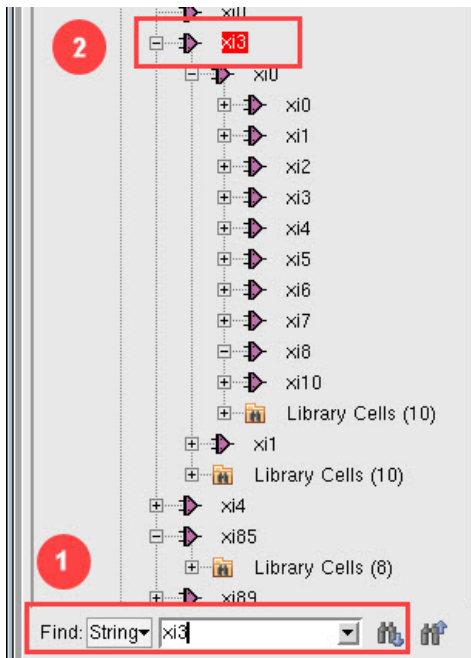
The *Previous Edge* and *Next Edge* buttons apply to only digital signals, not to analog ones.



Searching Instances

To search for instances in Design Browser, do the following:

In the Design Browser sidebar, enter the name of the instance you want to search in the *Find* field.

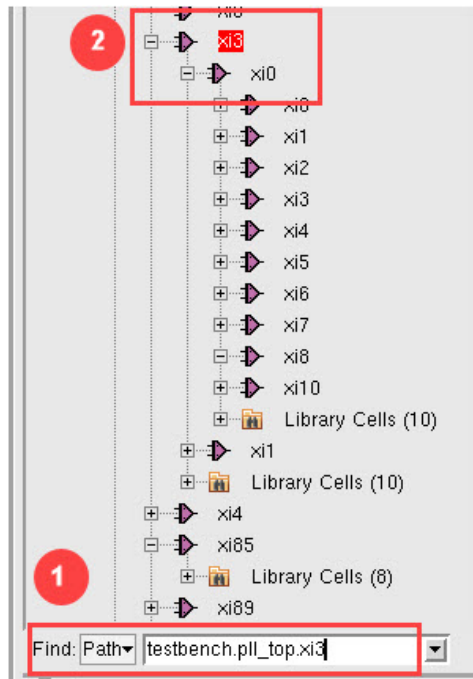


The instance is highlighted in the design hierarchy.

Searching Modules or Subcircuits

To search for modules or subcircuits in Design Browser, do the following:

1. In the Design Browser sidebar, select *Path* from the *Find* dropdown list.
2. Enter the path of the module or subcircuit that you want to search and press Enter.



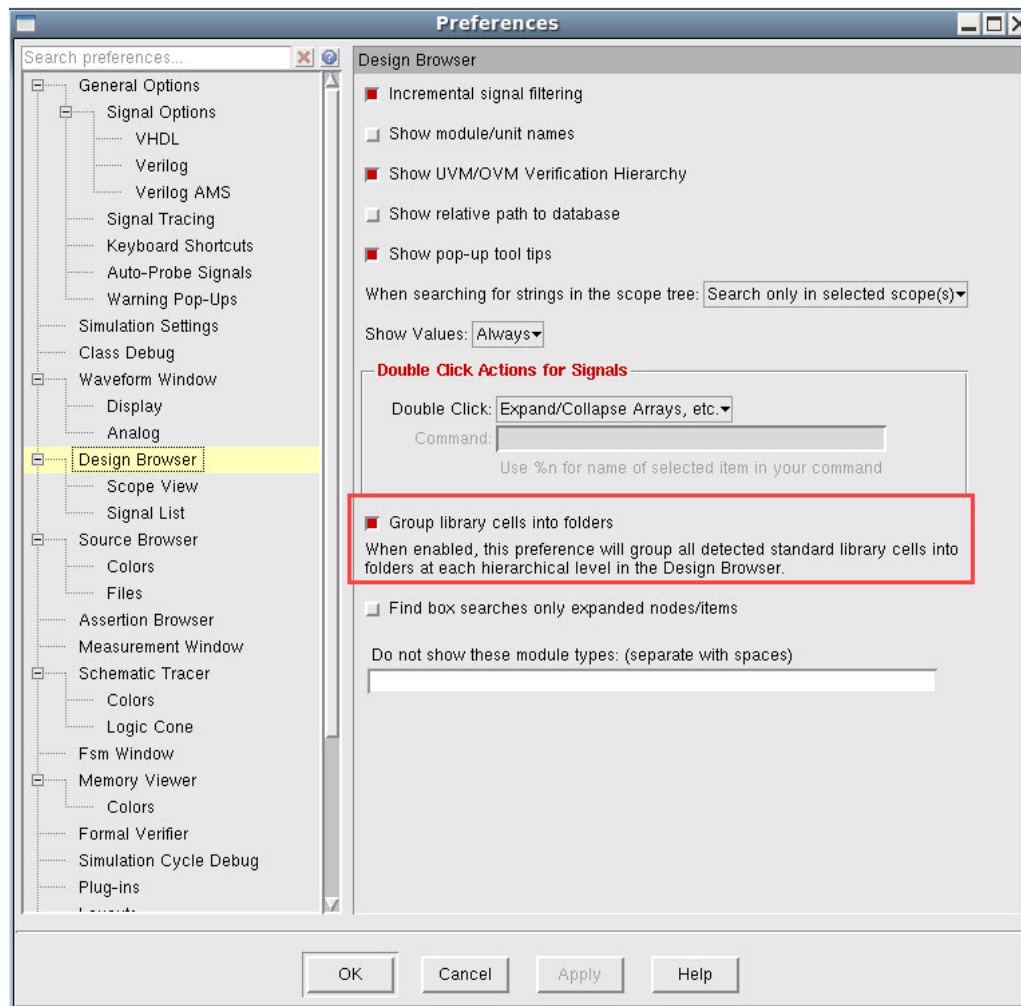
The search result is highlighted in the design hierarchy.

Disabling Grouping of Library Cells

By default, the analog instances are grouped into folders named *Library Cells*. You can set your preference to stop such grouping and display an expanded list of analog instances in the scope tree.

1. Click *Edit - Preferences*.

The Preferences form appears.



2. Deselect *Group library cells into folders*.

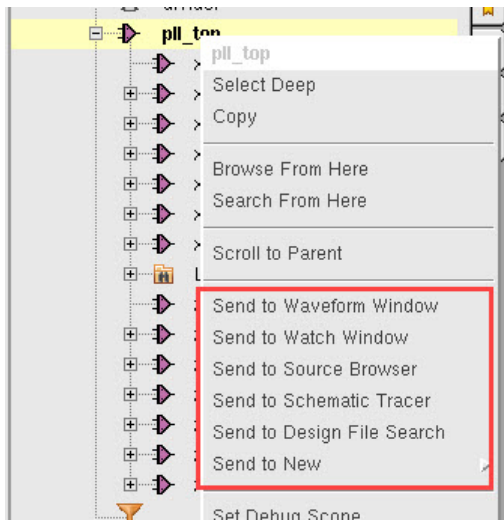
3. Click *Apply*.

You will notice an expanded list of analog instances in the scope tree.

Sending Instances to Other Windows

To cross-select and view the instances listed in Design Browser in the other SimVision MS windows, do the following:

1. Select and right-click the required instance in Design Browser.
2. Select one of the following options:



The instance will be opened in the selected window.

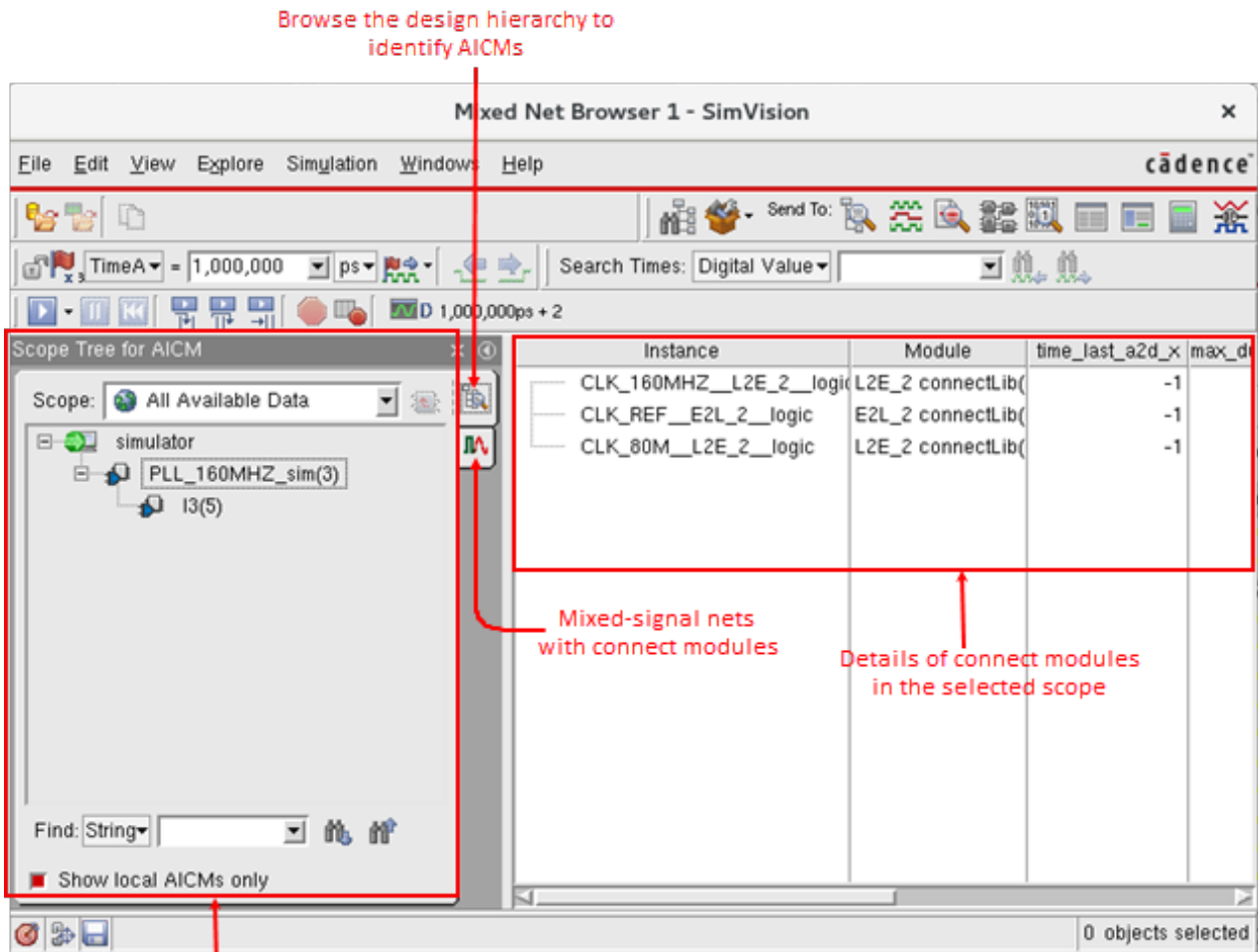
Working with Mixed Net Browser

The Mixed Net Browser window lists all the mixed-signal nets in the Mixed Nets sidebar. Each net in the Mixed Nets sidebar lists the connect modules in a mixed-signal design and gives you a picture of whether the connect modules are working as expected. By analyzing the values of the connect module variables, you can decide if the design needs optimization.

To invoke the Mixed Net Browser, do one of the following in other SimVision MS windows:

- Click the *Send to Mixed Net Browser* () icon on the toolbar
- Right-click a net and choose *Send to Mixed Net Browser*

The following figure describes the various elements of the Mixed Net Browser user interface:



Defines the scope to display the instances with connect modules

Related Topics

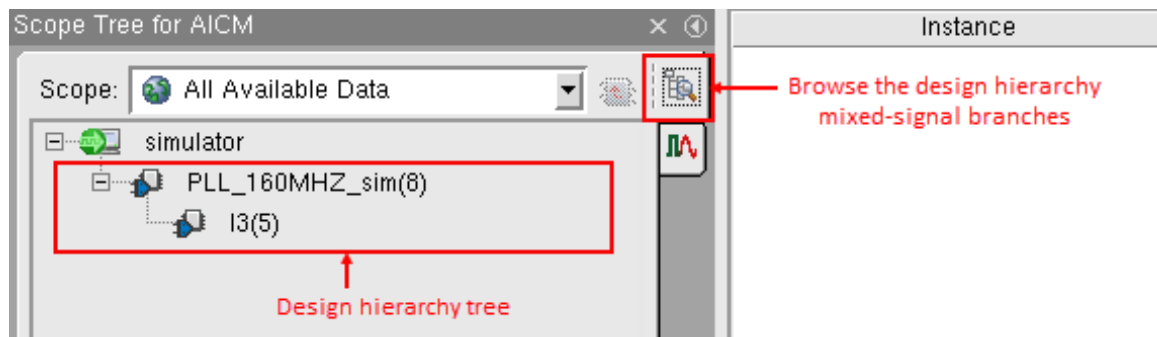
- [Defining Scope for Mixed-Signal Modules in Design Hierarchy](#)
- [Viewing the Mixed-Signal Nets in the Defined Scope](#)
- [Connect Module Instances Table](#)
- [Customizing the Connect Modules Instances Table](#)
- [Generating the Direct Waveform Plot](#)
- [Viewing Mixed Nets on Schematic Tracer](#)
- [Highlighting the Conversion Point in Schematic Tracer](#)

Defining Scope for Mixed-Signal Modules in Design Hierarchy

By default, the Mixed Net Browser shows the top-most module in the design hierarchy. You can use the *Scope Tree for AICM* sidebar to change the scope to display the modules with mixed nets.

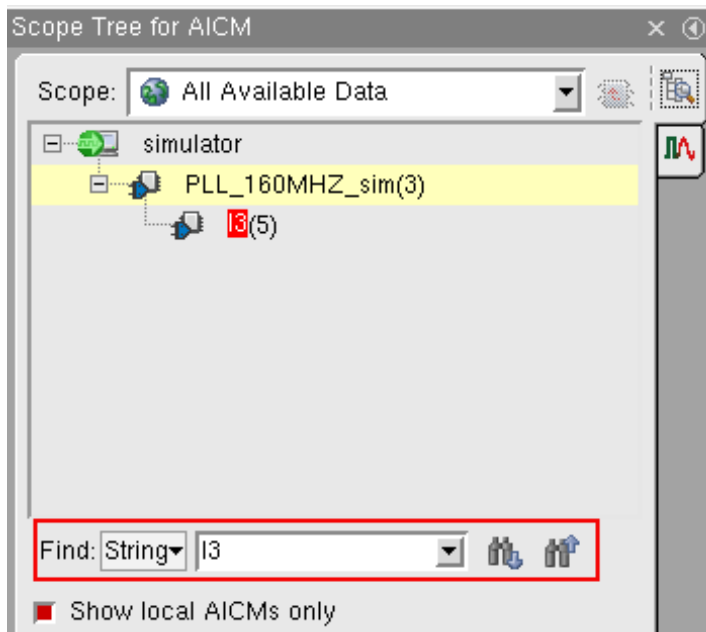
To define the scope for mixed-signal net display, do the following in the Mixed Net Browser:

1. On the *Browse the design hierarchy mixed-signal branches* tab, expand the design hierarchy tree to reach the level that contains the module for which you want to view mixed-signal nets.



 The total count of connect modules at each instance in the design hierarchy is shown next to the instance name.

2. Click the instance for which you want to display the auto-inserted connect modules in the table to the right.
The Mixed Net Browser displays the connect module instances at the selected level and all levels below it.
3. (Optional) To view the connect module instances at the selected level only, select the *Show local AICMs only* check box on the *Browse the design hierarchy mixed-signal branches* tab. The count of connect modules is changed to reflect the connect modules at an instance level only.
4. (Optional) In case of a large design, use the Find field to specify the substring to be searched in the design hierarchy.
All instance names that contain the searched substring shown the text highlighted in red.



Related Topics

[Working with Mixed Net Browser](#)

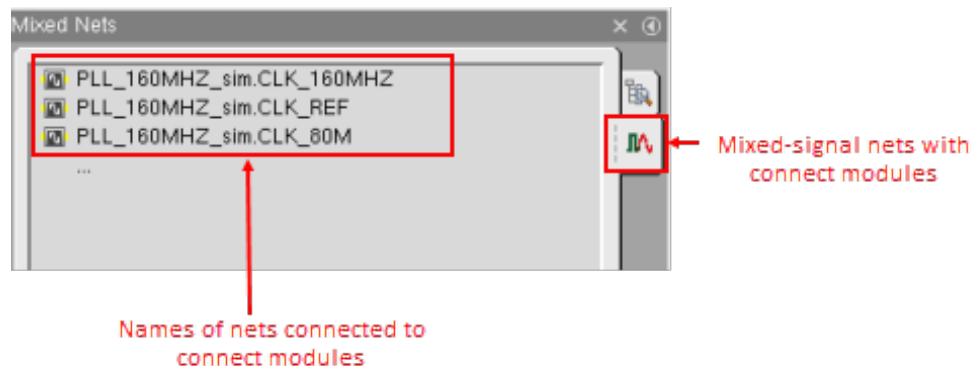
[Viewing the Mixed-Signal Nets in the Defined Scope](#)

Viewing the Mixed-Signal Nets in the Defined Scope

To view the names of the mixed-signal nets connected to with connect modules:

1. Open the *Mixed-signal nets with connect modules* tab in the left pane of the Mixed Net Browser window.

The names of the nets in the design hierarchy selected in the *Browse the design hierarchy mixes-signal branches* tab are displayed.



2. (Optional) If the number of the displayed nets is large, use the filter at the bottom of this tab to view a selected set of nets that have the specified substring in their name.

Related Topics

[Working with Mixed Net Browser](#)

[Defining Scope for Mixed-Signal Modules in Design Hierarchy](#)

Connect Module Instances Table

The connect module instances table is similar to profiling and it helps you to sort and compare all the connect modules in the design. Each mixed net listed in the sidebar displays the logical and electrical connect modules associated with it on the right pane. The following table lists the most commonly referred variables in the connect module instance table on the right pane.

Feature	Variable	Description
X State	time_last_a2d_x	The last time when the digital port signal went into the X state.
	max_duration_a2d_x	The maximum duration for which the digital port signal stayed in the X state.
Event Number	num_d2a_events	Number of d2a events from the digital port of IE, covering all possible toggling among 0, 1, X, Z.

	num_a2d_events	Number of a2d events to the digital port of IE, covering all possible toggling among 0, 1, X, Z.
Exceed Value	vin_exceeded_limit	Flags 1 when the analog input signal voltage has exceeded the inputlimit, which is vsup*4/3 by default.
	vdd_exceeded_limit	Flags 1 when the vdd supply net voltage level has exceeded the vddlimit, which is vsup*4/3 by default.

Consider the following example screenshot. When the table is sorted by the *num_d2a_events* column, you can see that the most busy connect module is the one connected with the *sclk_L2E_2_electrical* instance.

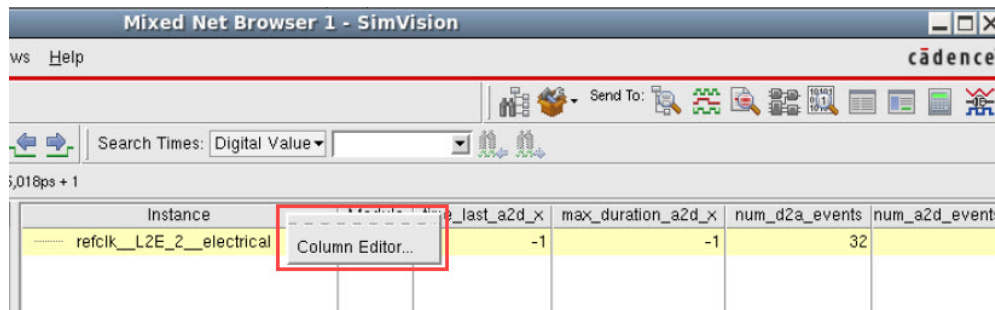
Instance	Module	time_last_a2d_x	max_duration_a2d_x	num_d2a_events	num_a2d_events
sclk_L2E_2_electrical	L2E_2	-1	-1	34	0
sa[1]_L2E_2_electrical	L2E_2	-1	-1	13	0
sb[1]_L2E_2_electrical	L2E_2	-1	-1	13	0
sa[0]_L2E_2_electrical	L2E_2	-1	-1	12	0
sb[0]_L2E_2_electrical	L2E_2	-1	-1	12	0
sa[3]_L2E_2_electrical	L2E_2	-1	-1	8	0
sb[3]_L2E_2_electrical	L2E_2	-1	-1	8	0
sa[2]_L2E_2_electrical	L2E_2	-1	-1	7	0
sb[2]_L2E_2_electrical	L2E_2	-1	-1	7	0
sa[4]_L2E_2_electrical	L2E_2	-1	-1	5	0
sb[4]_L2E_2_electrical	L2E_2	-1	-1	5	0
sa[6]_L2E_2_electrical	L2E_2	-1	-1	4	0
sa[5]_L2E_2_electrical	L2E_2	-1	-1	4	0
sb[6]_L2E_2_electrical	L2E_2	-1	-1	4	0
sb[5]_L2E_2_electrical	L2E_2	-1	-1	4	0
sa[7]_L2E_2_electrical	L2E_2	-1	-1	3	0

This kind of profiling and sorting lets you analyze variables values of the connect modules in the table and evaluate if any of them is not working as expected or is slowing down the test.

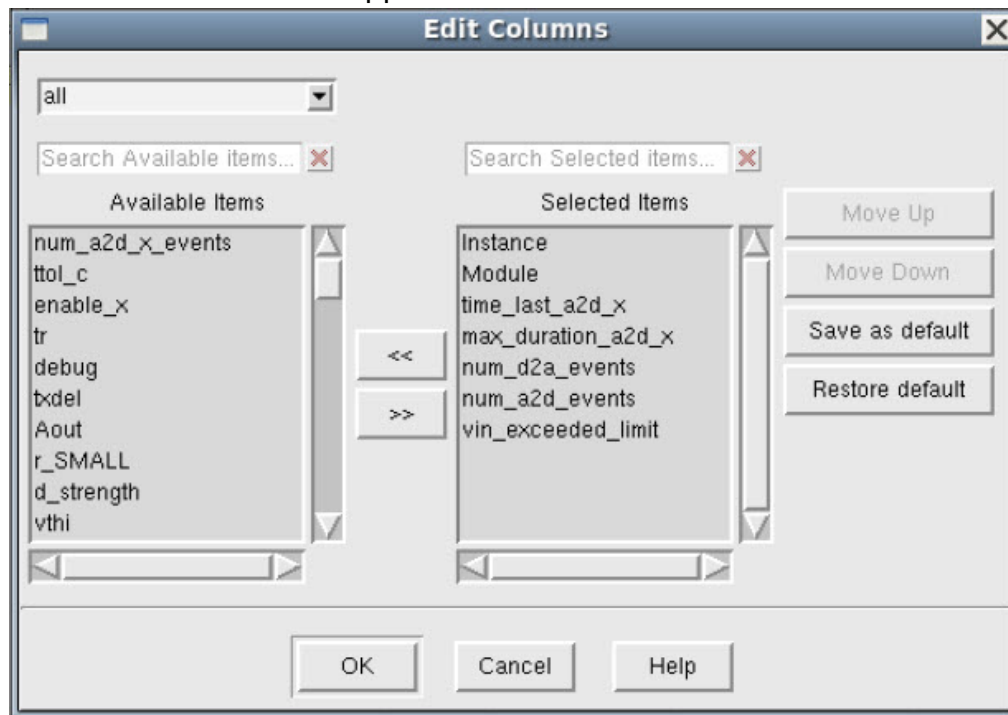
Customizing the Connect Modules Instances Table

There is an option to customize the connect module table to add or delete the variables to be displayed as per your requirement. Perform the following steps to customize the table:

1. Click the *View* menu and select *Column Editor*. Alternatively, right-click the connect module instance table header and select *Column Editor*.



The Edit Columns form appears.



2. Update the variables, as required:

- To add a column to the table, select the required items from the *Available Items* list and click **>>**.
- To remove a column from the table, select the required items from the *Selected Items* list and click **<<**.
- Reorder the items as required in the *Selected Items* list using the *Move Up* and *Move Down* buttons.

3. (Optional) To save the updated list of *Selected Items* as your default selection of columns, click *Save as default*.

You can update `preferences` set in the `~/.simvisionrc` file in your home directory. For

example, add the following in the file:

```
preferences set iebrowser-default-columns {treeView moduleName vsup num_a2d_events  
num_d2a_events}
```

This will ensure that your customization is saved and restored every time you launch SimVision.

4. Click *OK* to save the changes and close the form.

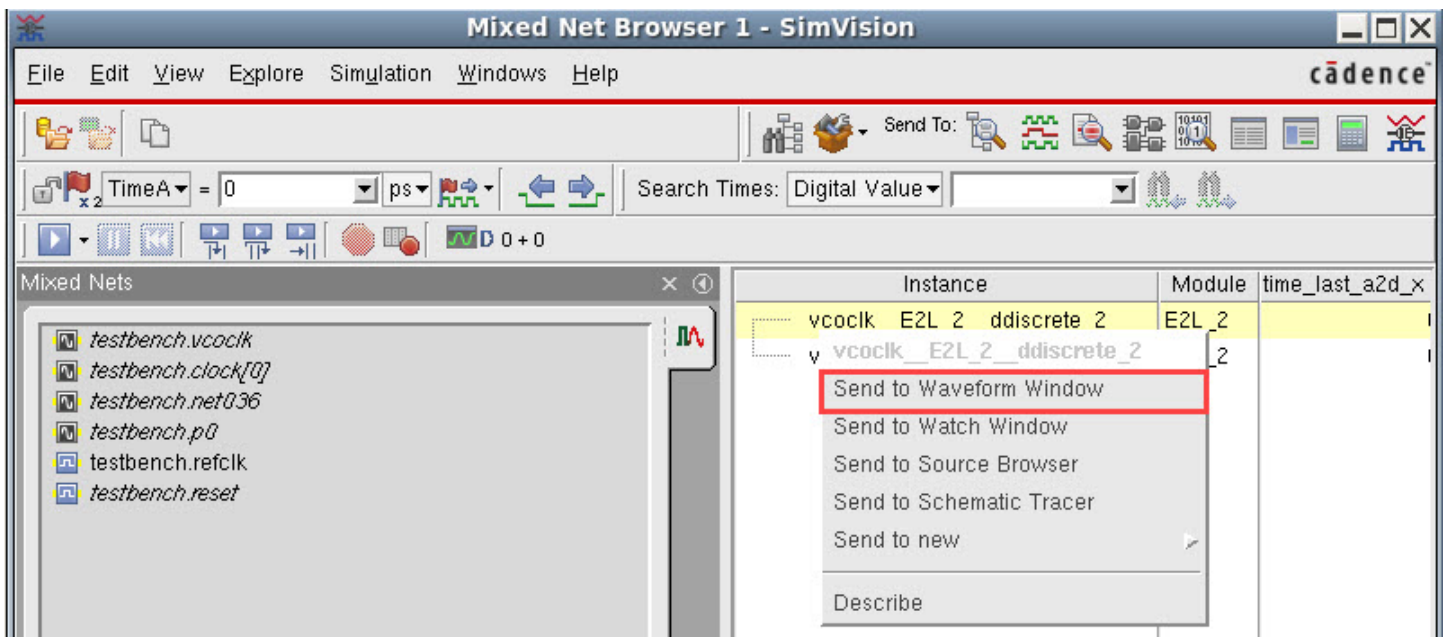
Generating the Direct Waveform Plot

Waveforms can be plotted for viewing performance of a connect module or for the entire mixed net. See the following topics for more details:

- [Plotting Waveform for a Connect Module](#)
- [Plotting Waveform for a Mixed Net](#)

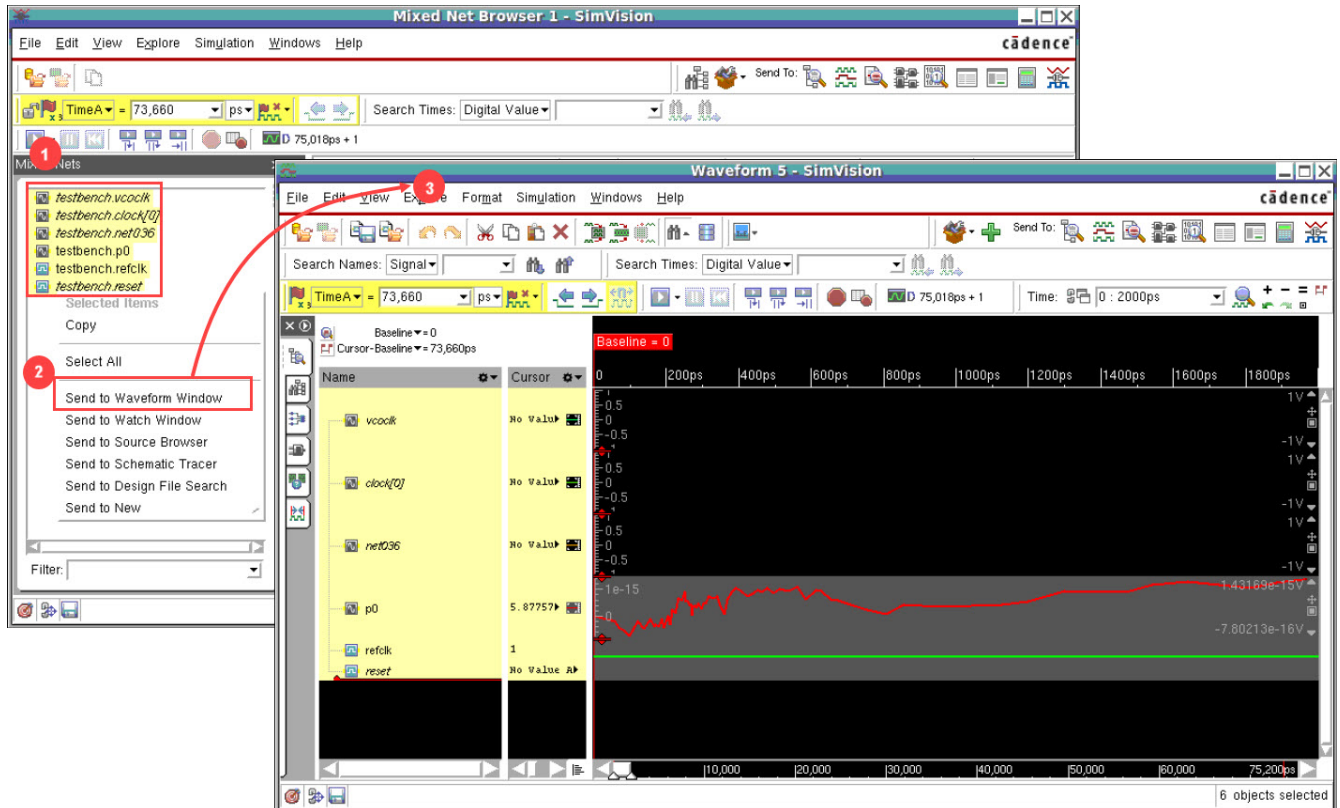
Plotting Waveform for a Connect Module

To view the performance of the connect modules in a waveform, right-click a connect module instance in the right pane of Mixed Net Browser and select *Send to Waveform Window*. Alternatively, select the connect module and click  in the toolbar.



Plotting Waveform for a Mixed Net

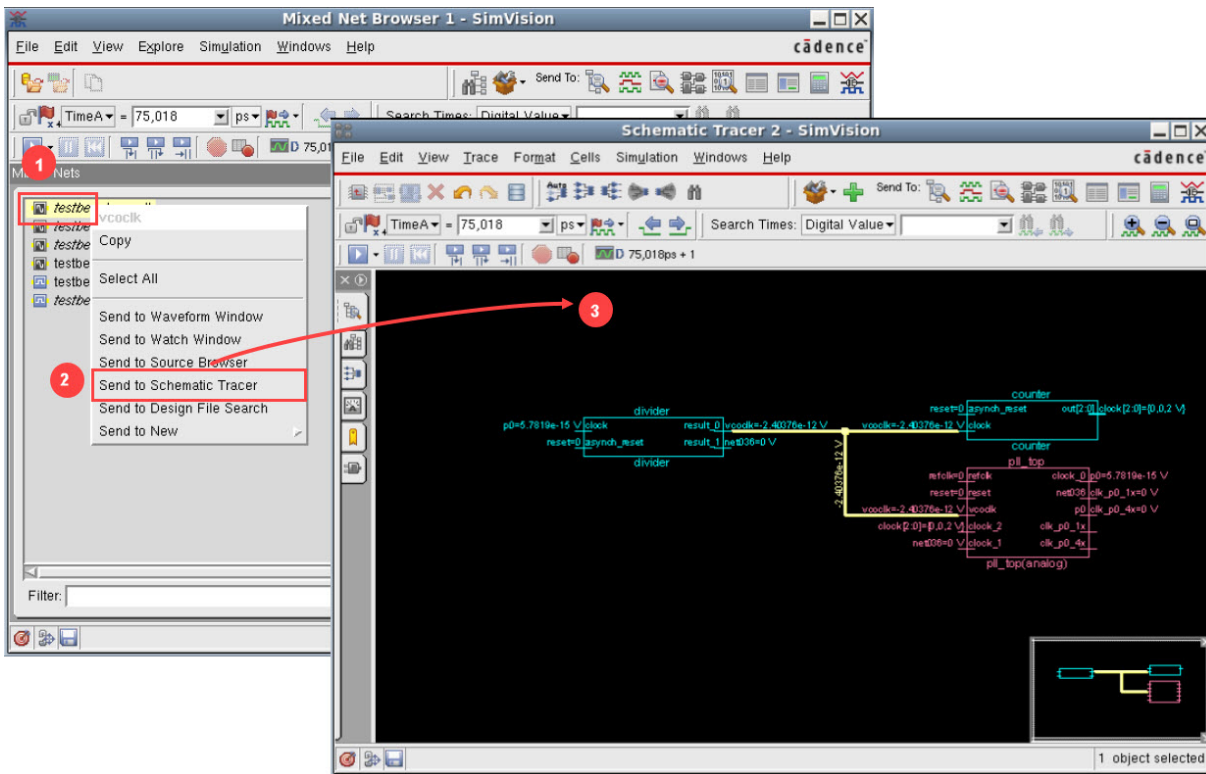
To generate a waveform plot, select a net in the Mixed Nets sidebar and select *Send to Waveform Window*. Alternatively, select the net and click  in the toolbar.



Viewing Mixed Nets on Schematic Tracer

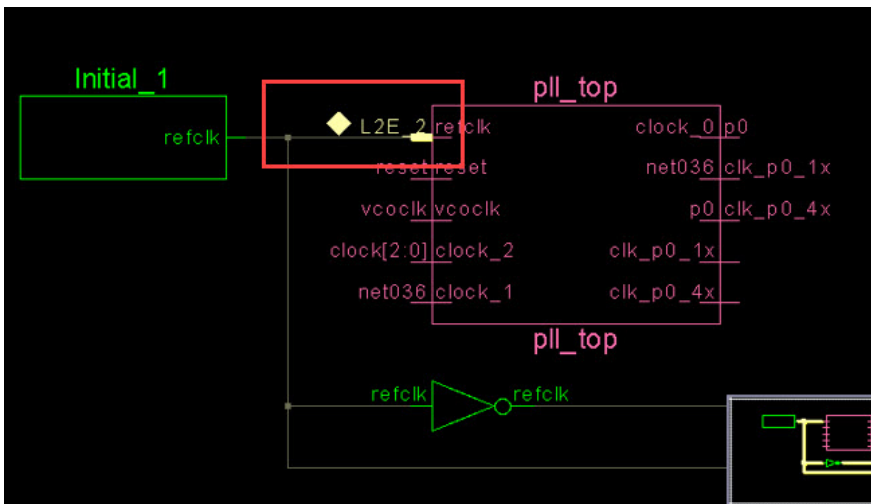
After analyzing the values of the connect modules you can view them in the Schematic Tracer for debugging.

To view connect modules in Schematic Tracer, right-click a net in the Mixed Nets sidebar and select *Send to Schematic Tracer*. Alternatively, select a net and click  in the toolbar.



Highlighting the Conversion Point in Schematic Tracer

In Schematic Tracer, the logic to electrical conversion point is highlighted with the  symbol and the connect module is highlighted with the  symbol.

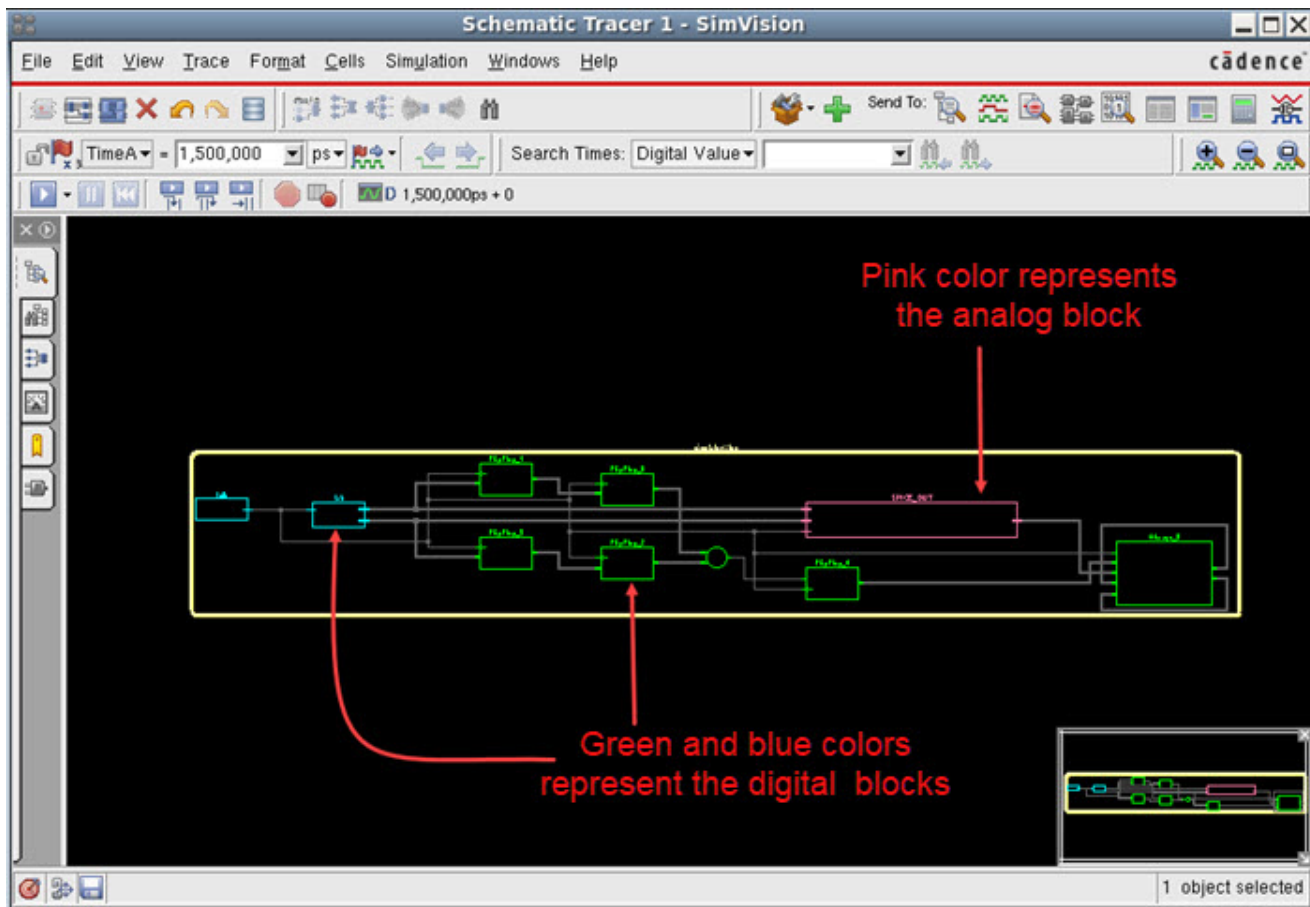


The marking of the logic conversion points and insertion of the connect modules in Schematic Tracer makes it very convenient to understand and debug the connect modules.

Working with Schematic Tracer

Schematic Tracer in SimVision MS extends support to display the analog circuit with the digital devices of a mixed-signal simulation and overcomes the challenge to debug a complex mixed-signal design.

Schematic Tracer can be invoked using the  icon on the toolbar. It can also be invoked using the right-click option, *Send to Schematic Tracer*, from the other Simvision MS windows. When launched, Schematic Tracer looks like the following:



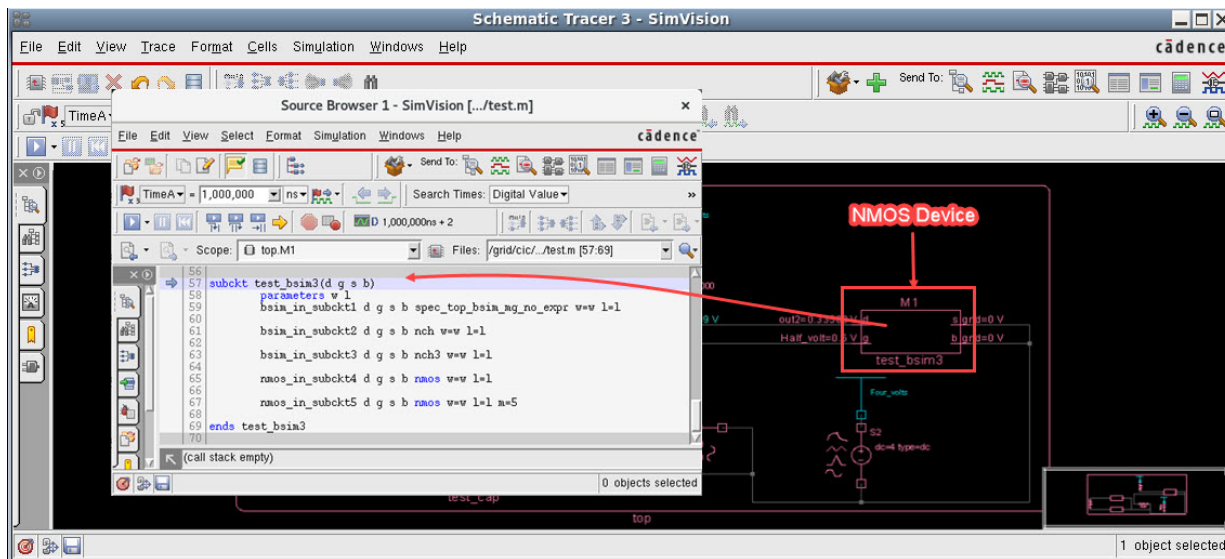
Related Topics

- [Setting up Electrical Symbol Mapping](#)
- [Tracing the Critical Signal Path](#)
- [Checking the Current Distribution](#)
- [Cross-Selecting Instances and Nets](#)
- [Viewing Instances in Other Windows](#)

Setting Up Electrical Symbol Mapping

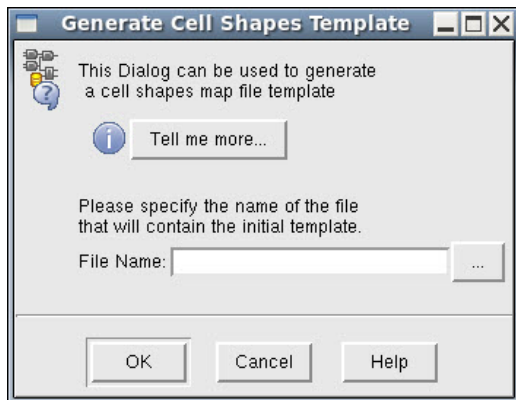
Devices that are not analog primitives but are part of the analog subcircuit are displayed in rectangular boxes in the schematic. You will not be able to descend into these subcircuits or inline subcircuit symbols in the Schematic Tracer.

For example, the highlighted symbol in the image below is an NMOS device. It is represented in a rectangular box in Schematic Tracer. When this symbol is sent to Source Browser, the name of the subcircuit and the other details of the symbol can be found.



Perform the following steps to map these symbols into meaningful electrical symbols in SimVision MS:

1. In Schematic Tracer, click the *Cells* menu and select *Generate Cell Template*. The Generate Cell Shapes Template form appears.



2. In the *File Name* field, enter a name for the template file with `.map` extension. For example, `analog.map`.

3. Click OK.

A `.map` file will be generated in the current working directory. Following is the sample of the contents of a `.map` file.

```
cell buf shape <<unknown>>
cell L2E_2 shape <<unknown>> ports {{Din input normal} {Aout output normal}}
cell E2L_2 shape <<unknown>> ports {{Ain input normal} {Dout output normal}}

section analog

cell test_bsim3 symbol <<unknown>> ports {{d:<<unknown>> inout normal} {g:<<unknown>> inout
normal} {s:<<unknown>> inout normal} {b:<<unknown>> inout normal}}

cell test_cap symbol <<unknown>> ports {{p:<<unknown>> inout normal} {n:<<unknown>> inout
normal}}

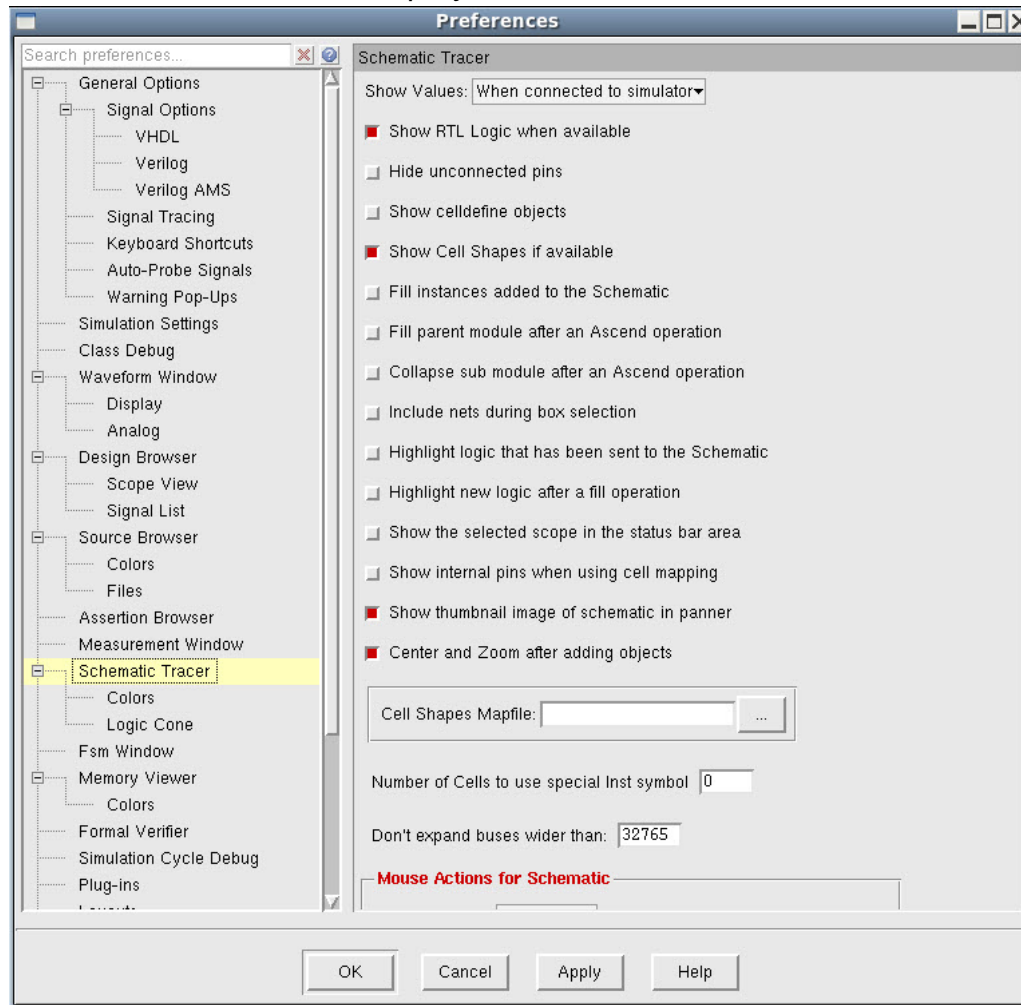
cell vsrc symbol <<unknown>> ports {{src:<<unknown>> inout normal} {gnd:<<unknown>> inout
normal}}

.....
endsection
```

The upper section of the file represents the digital part. The analog section has a line beginning with `cell` for each subcircuit and inline subcircuit.

4. Delete the digital contents from file.
5. In the analog section, replace all the `<<unknown>>` symbols and ports with the [built-in](#) keywords.
6. Save and close the file.
7. In Schematic Tracer, click *Edit - Preferences*.

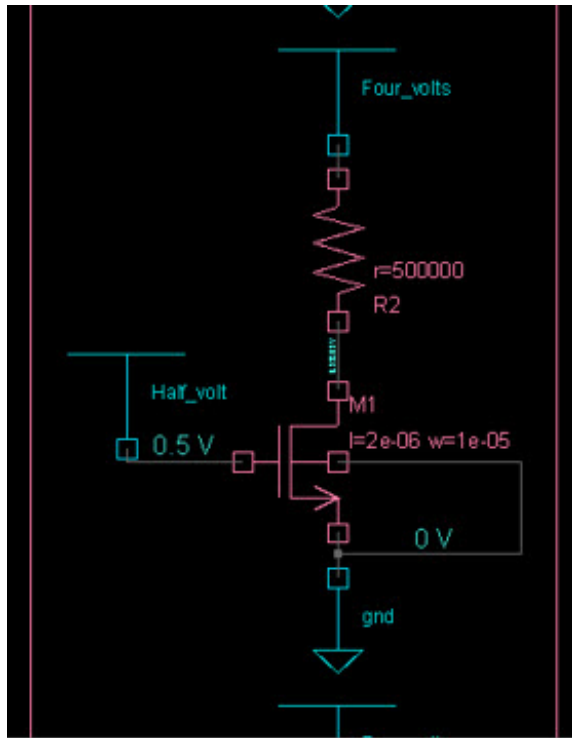
The Preferences form is displayed.



8. In the *Cell Shapes Mapfile* field, enter the path to the `.map` file that you saved. Alternatively, you can click ... to browse and select the file.

9. Click *Apply*.

When you re-invoke Schematic Tracer, the rectangular boxes will be modified as meaningful electrical symbols.



Built-In Symbols and Ports

The following table lists all the built-in symbols and ports. Note that these are case-sensitive.

Symbol	Port List	Symbol	Port List	Symbol	Port List
res	port: PLUS inout port: MINUS inout	psource	port: PLUS inout port: MINUS inout	njfet	port: D inout port: S inout port: G inout
cap	port: PLUS inout port: MINUS inout	vccs	port: PLUS inout port: MINUS inout port: NC+ inout port: NC- inout	pjfet	port: D inout port: S inout port: G inout

ind	port: PLUS inout port: MINUS inout	vcvs	port: PLUS inout port: MINUS inout port: NC+ inout port: NC- inout	nmes4	port: D inout port: S inout port: G inout
phyres	port: t1 inout port: t2 inout port: t0 inout	cccs	port: PLUS inout port: MINUS inout	npn	port: C inout port: E inout port: B inout
xfmr	port: P+ inout port: P- inout port: S+ inout port: S- inout	ccvs	port: PLUS inout port: MINUS inout	pnp	port: C inout port: E inout port: B inout
winding	port: t inout port: b inout	n1port	port: t1 inout port: b1 inout	nmos	port: S inout port: D inout port: G inout
tline	port: IN+ inout port: IN- inout port: OUT+ inout port: OUT- inout	n2port	port: t1 inout port: b1 inout port: t2 inout port: b2 inout	nmos4	port: S inout port: D inout port: G inout port: B inout

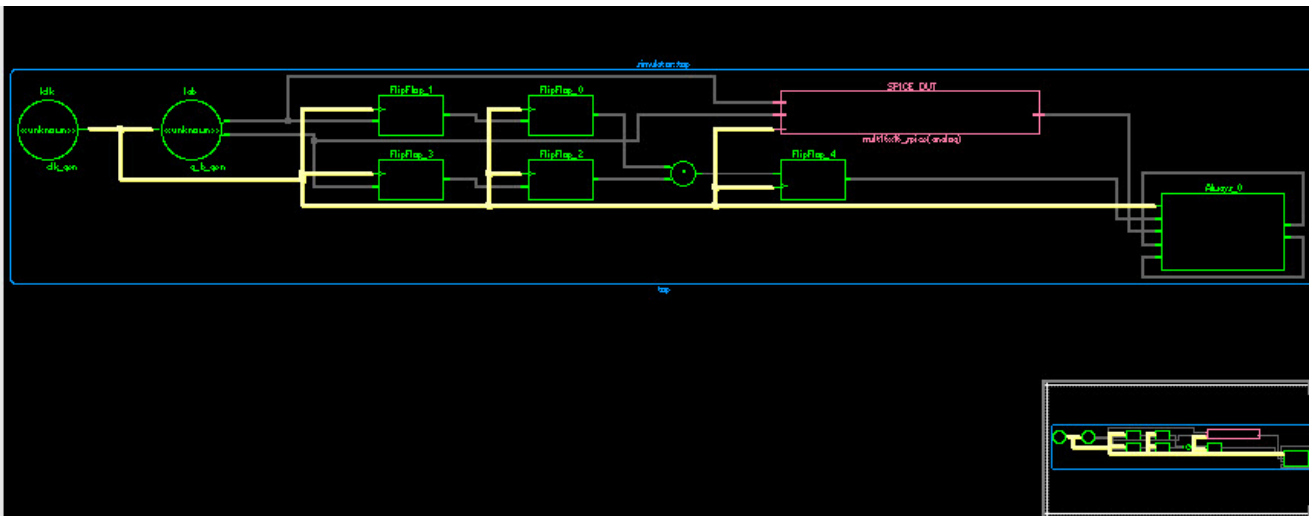
msline	port: t1 inout port: b1 inout port: t2 inout port: b2 inout	n3port	port: t1 inout port: b1 inout port: t2 inout port: b2 inout port: t3 output port: b3 inout	pmos	port: S inout port: D inout port: G inout
mtline	port: in1 inout port: out1 inout port: inref inout port: outref inout	n4port	port: t1 inout port: b1 inout port: t2 inout port: b2 inout port: t3 output port: b3 inout port: t4 inout port: b4 inout	pmos4	port: S inout port: D inout port: G inout port: B inout
delay	port: p inout port: n inout port: ps inout port: ns inout	diode	port: PLUS inout port: MINUS inout	iprobe	port: PLUS inout port: MINUS inout
fracpole	port: PLUS inout port: MINUS inout	schottky	port: PLUS inout port: MINUS inout	switch	port: N+ inout port: N- inout port: NC+ inout port: NC- inout
vsource	port: PLUS inout port: MINUS inout	zener	port: PLUS inout port: MINUS inout	fourier	port: p inout port: n inout

isource	port: PLUS inout port: MINUS inout			fourier2ch	port: p inout port: n inout port: pr inout port: nr inout
---------	---	--	--	------------	---

Tracing the Critical Signal Path

Schematic Tracer flattens the design and has the capability to display the critical signal path exclusively even at various hierarchy levels of a design. This feature allows you to focus on specific signals and debug it to find the root cause of a problem.

The critical signal path is highlighted in yellow when a signal is clicked.



Checking the Current Distribution

Tracing the current flow and understanding its direction is essential to debug an analog net. Tracing currents in large designs can be a time consuming and effort-intensive task.

SimVision MS has a Browse Currents sidebar that provides a clear understanding of the current distribution while saving a lot of debugging time. Probing the current distribution in SimVision MS does not require you to set up current probes or save ports before the simulation starts.

To open the Browse Currents sidebar, do the following in Schematic Tracer:

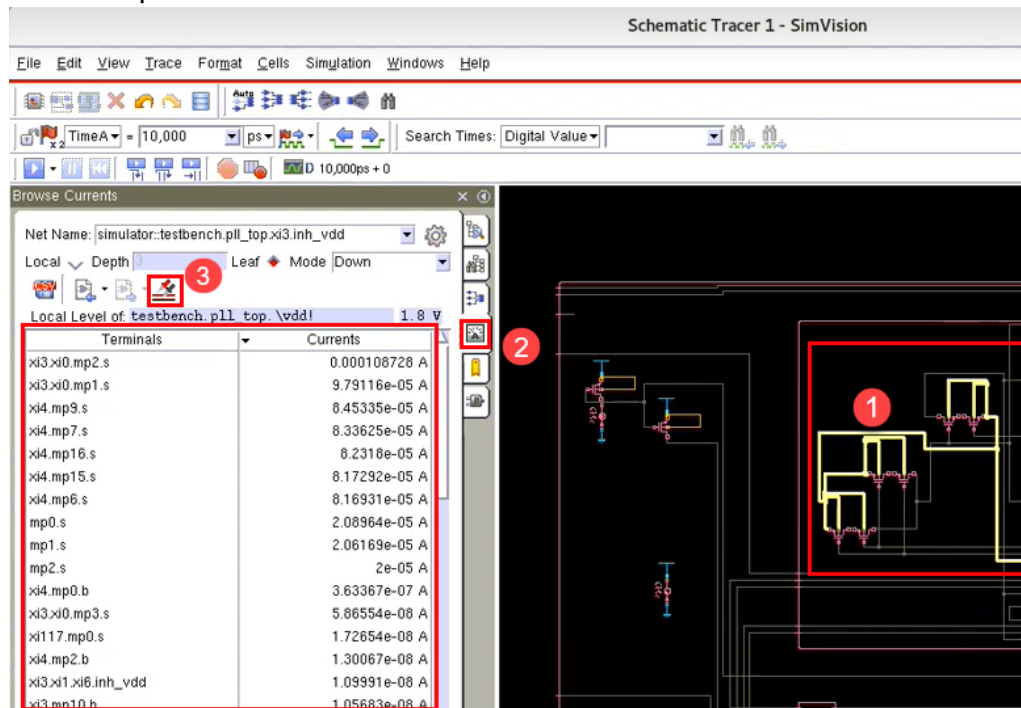
1. Select an analog net in the schematic.
2. Click *Browse analog current on a given analog net* () on the left bar to open the *Browse*

Currents sidebar.

3. Click *Browse the current for the selected net* () in the *Browse Currents* sidebar to browse the currents for the selected net.

SimVision MS queries Spectre and displays a list of terminals and currents on the selected net. The voltage of the net is shown above the terminals and currents table.

An example is shown below.



- **Dynamic current versus static current:** Due to the different methods used by Spectre to calculate the static and dynamic currents, a small discrepancy is expected for some large net. Refer to Spectre Classic Simulator, Spectre APS, Spectre X, and Spectre XPS User Guide for more information.
- **Batch mode versus debugging mode:** When SimVision MS is enabled, the simulation runs in the debugging mode. In this mode, some optimization techniques are turned off. It may cause numerical discrepancy. However, it should be within tolerance levels.

Related Topics

- [Understanding the Current Flow Direction](#)
- [Sorting Currents](#)
- [Checking the Current Distribution](#)
- [Checking the Current Distribution at Different Levels in the Hierarchy](#)

Sorting Currents

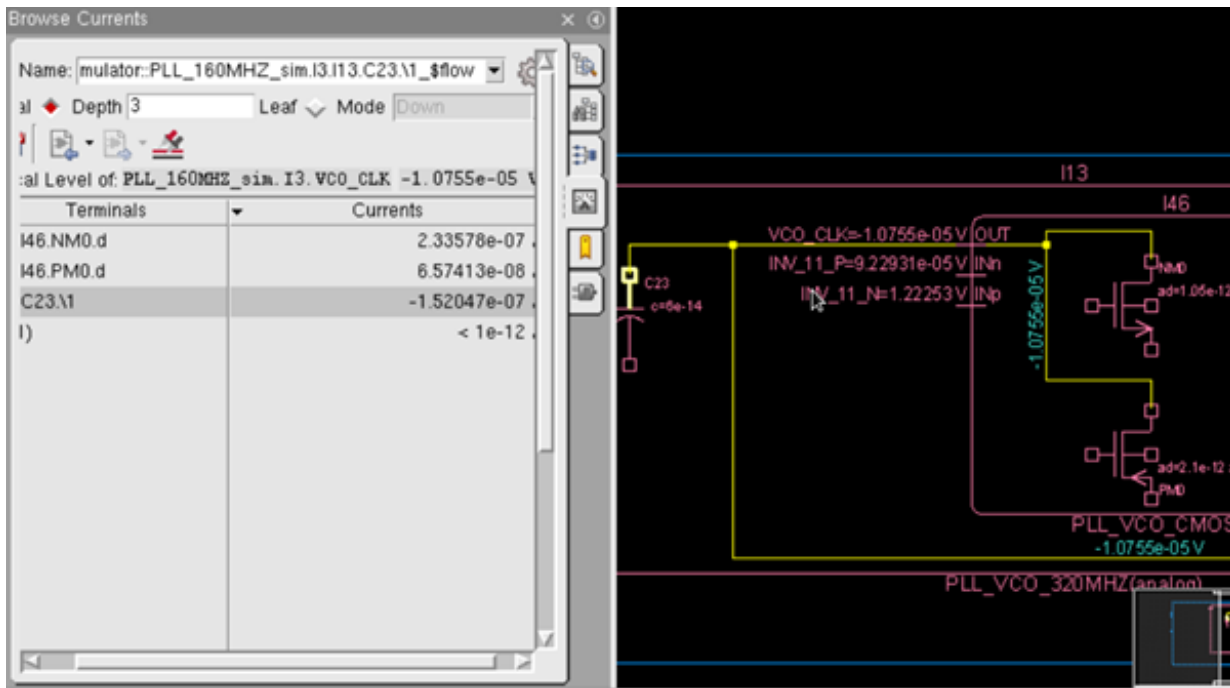
In the Browse Currents sidebar, you can sort the table in following two ways:

- Click the *Terminals* column header to sort the table alphabetically based on the terminal label.
- Click the Currents column header to sort the table based on absolute values of the currents.

Understanding the Current Flow Direction

According to Kirchhoff's Current Law, the sum of currents on a nodes is equal to 0. However, Spectre notates currents in a different way. Spectre solver notates the current value reporting on a subcircuit or a primitive as a positive value '+' when the current enters the port and a negative value '-' when the current flows out in the reverse direction. SimVision MS follows current direction conversion model of Spectre. Therefore, in the Browse Currents sidebar, the current value of a terminal is positive if current is coming in and negative if current is going out.

Consider the following example screenshot. Two current values in the table are positive indicating inflow of current and one is negative indicating outflow of current.



Related Topics

[Checking the Current Distribution](#)

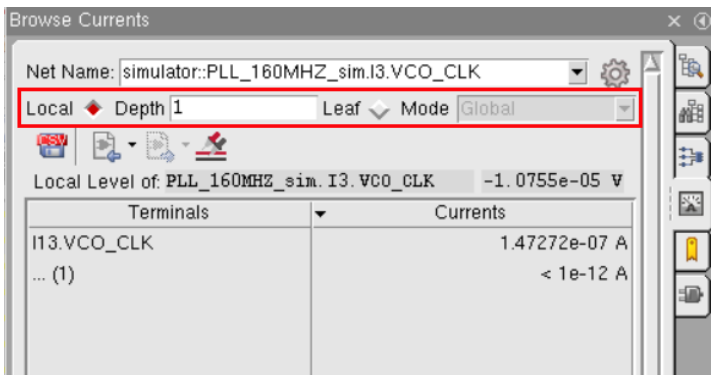
[Tracing the Critical Signal Path](#)

[Checking the Current Distribution at Different Levels in the Hierarchy](#)

Checking the Current Distribution at Different Levels in the Hierarchy

The Browse Currents sidebar lets you view all the current values at instance ports or terminals connected to the selected net at various hierarchical levels of the design.

The following figure shows the default settings used by the tool to define the scope for the display of current values:

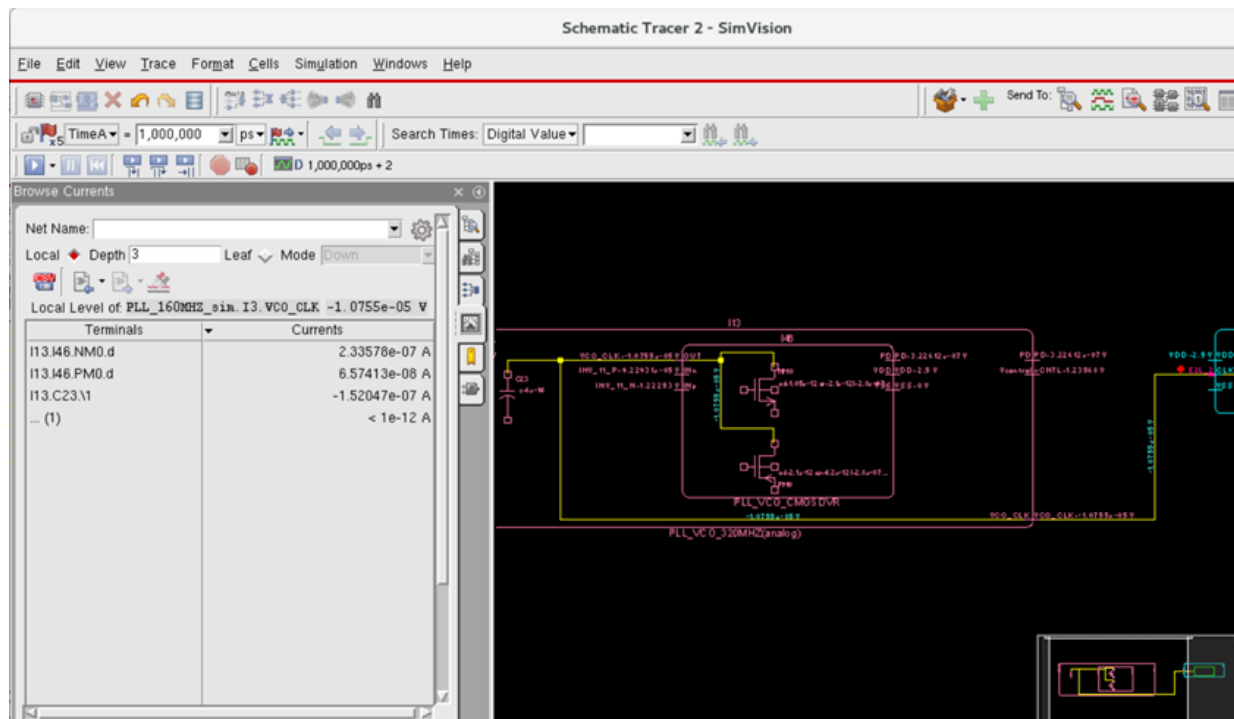


By default, *Depth* is set to 1, which implies that SimVisionMS displays currents for the hierarchy level displayed in the schematic.

You can check current distributions at different levels in the hierarchy in the following ways:

- By changing the depth in the *Depth* field. Increasing the depth value displays currents from the specified levels down in the design hierarchy.

The following example shows the current values displayed from 3 levels down in the hierarchy:



- By enabling the *Leaf* option to view currents at the leaf level of the selected net. Along with this option, you can also choose a mode from the *Mode* drop-down list to either check the net

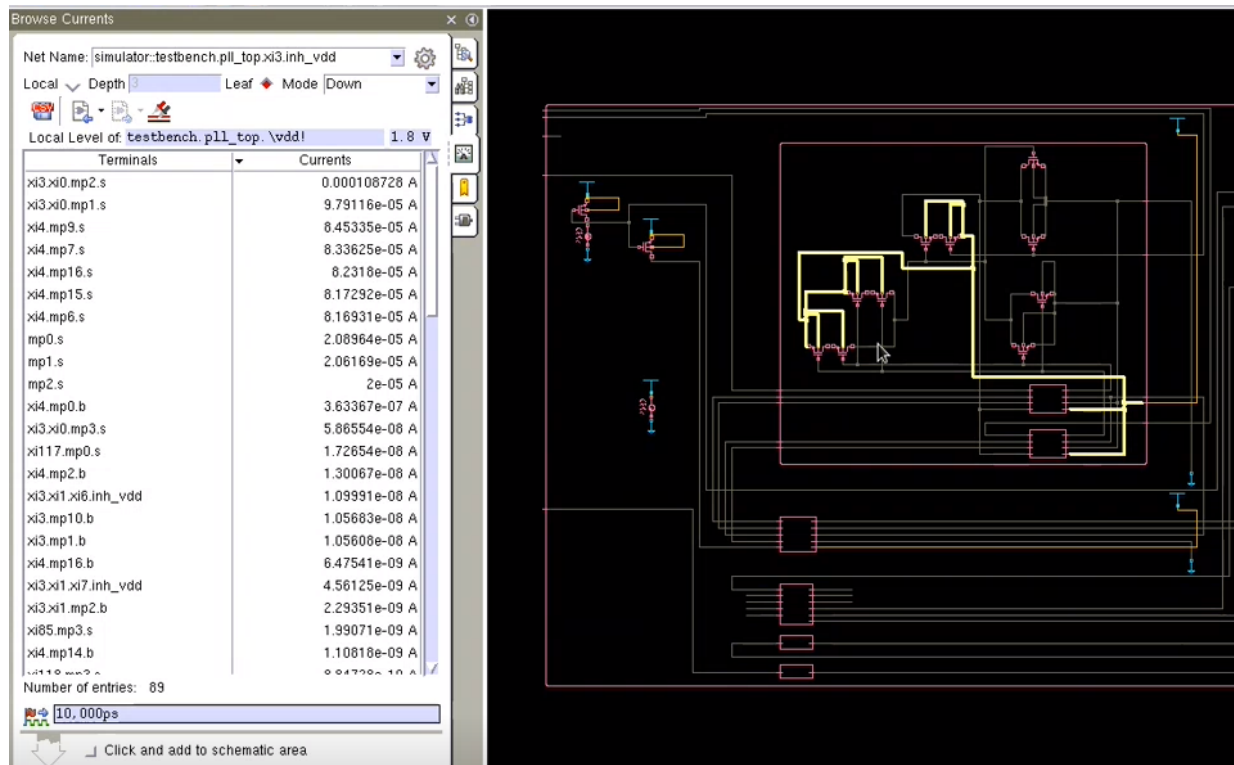
globally at all hierarchy levels or only below the current hierarchy level.

Leaf Modes

The Browse Currents sidebar provides two modes for leaf-level checking:

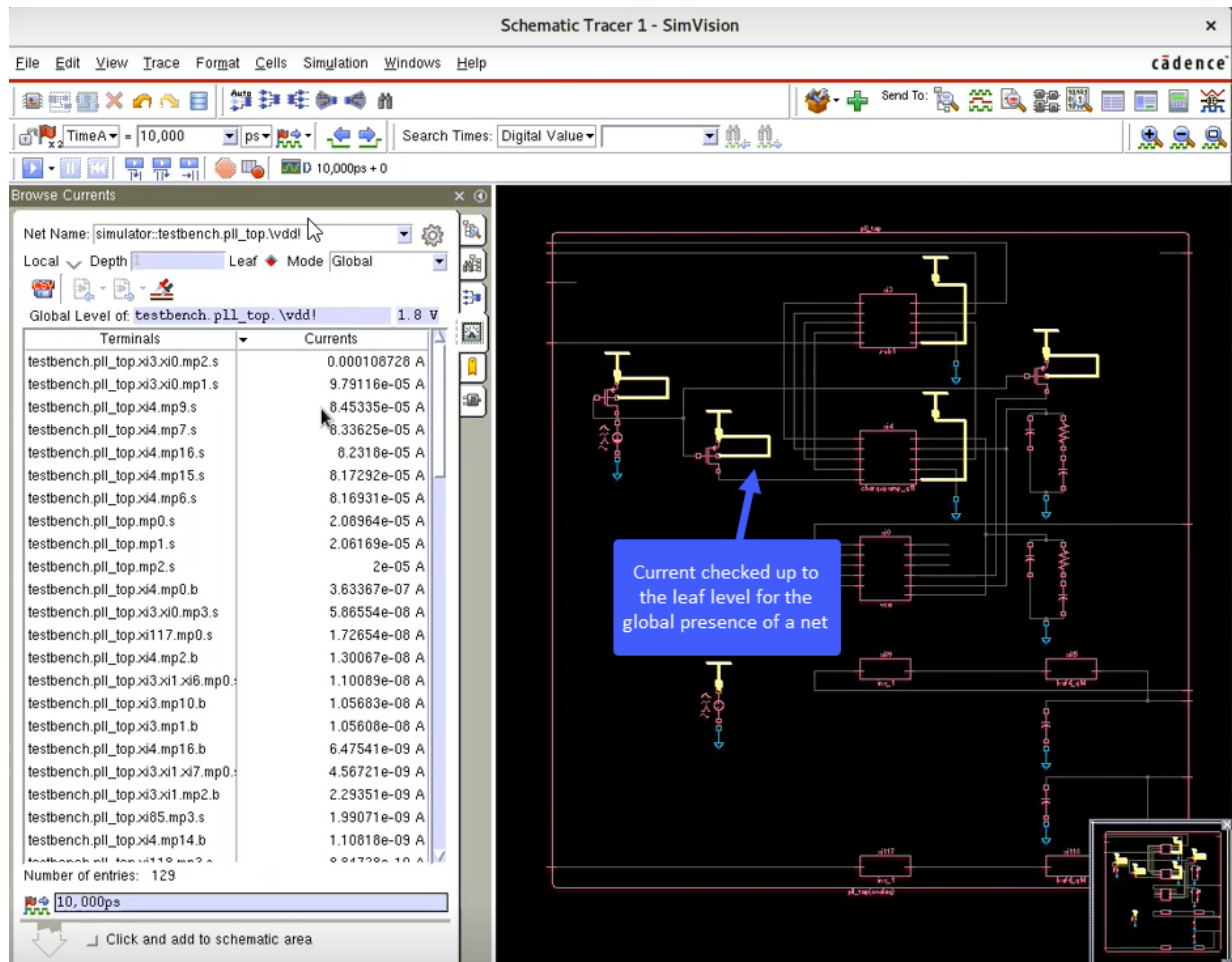
- *Down*: In this mode, the current distribution to the leaf level is checked below the current level in the hierarchy.

In the following example, currents are checked for net `inh_vdd` down to the leaf level inside the current instance only.



- *Global*: In this mode, the current distribution to the leaf level of the selected net is checked globally in all instances at all hierarchy levels, up or down, in the complete design.

In the following example, currents are checked for net `vdd` to the leaf level globally in the design.



✓ For a clear visual identification of a net in the schematic, choose *Format – Color Through Hierarchy* and choose a color from the palette.

Related Topics

- [Understanding the Current Flow Direction](#)
- [Sorting Currents](#)

Exporting Terminal and Current Data to a CSV File

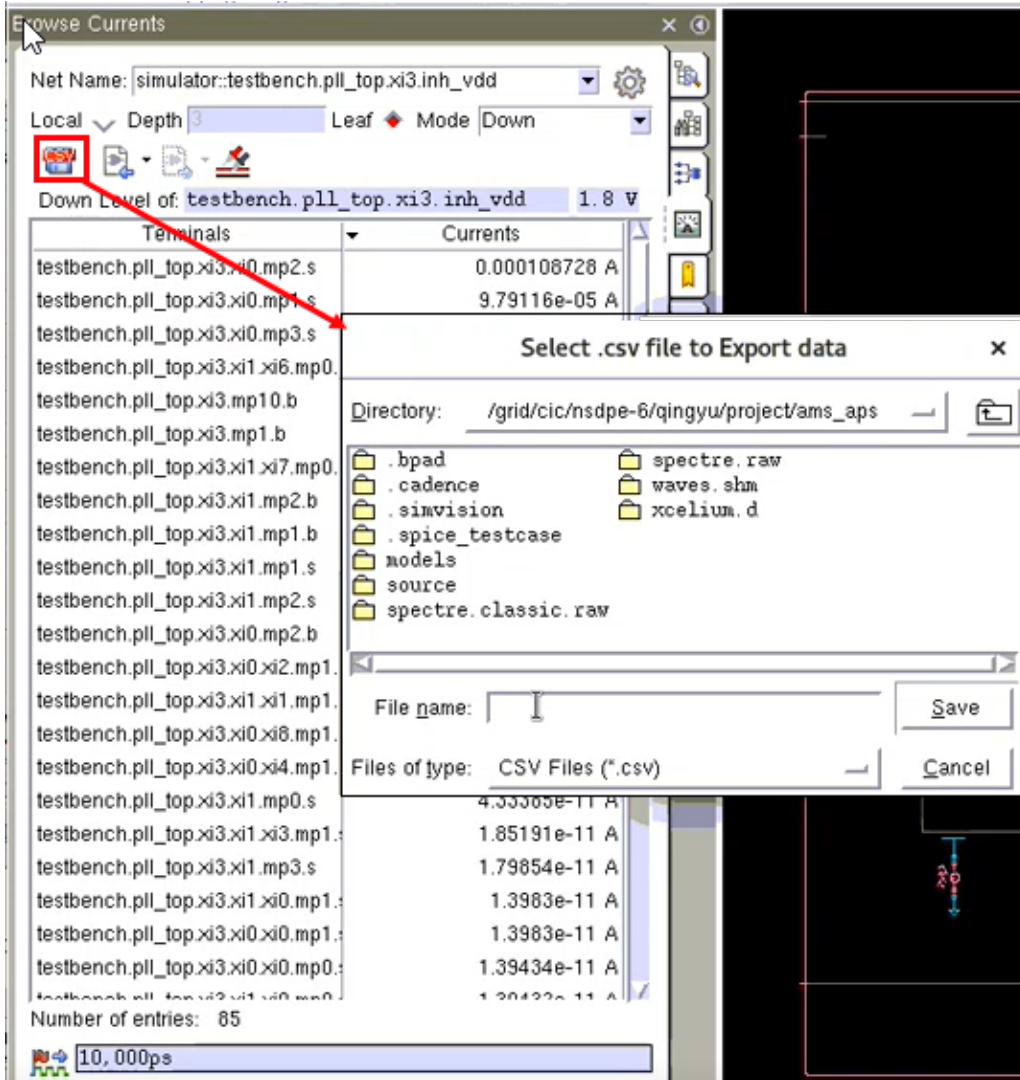
You can save the terminal and current details for the selected net in a .csv file for further analysis and comparison to be done later.

To export the terminal and current details displayed in the Browse Currents sidebar of

To export the terminal and current details displayed in the Browse Currents sidebar of Schematic Tracer:

1. Click *Export displayed current values to .csv file* in the Browse Currents sidebar.

The Select .csv file to export data form is displayed.



2. Browse to the directory where you want to save data.
3. Specify a file name and click *OK*.

The terminals and their current values are exported from the table to the specified .csv file. The header of the CSV file also contains information about the data source .

Related Topics

- [Checking the Current Distribution](#)

Cross-Selecting Instances and Nets

Many of the applications that are part of the AMS Designer software support cross-selecting. Cross-selecting means selecting an instance or net in one application and having the same instance or net automatically selected in another application. For example, you can highlight an instance in the Virtuoso Schematic Editor and have the same instance appear highlighted in the SimVision Schematic Tracer window. Similarly, you can highlight a net in the SimVision Design Browser and have the same net appear highlighted in the Waveform window.

See the following topics for more information:

- [Cross-Selecting Instances](#)
- [Cross-Selecting Nets](#)

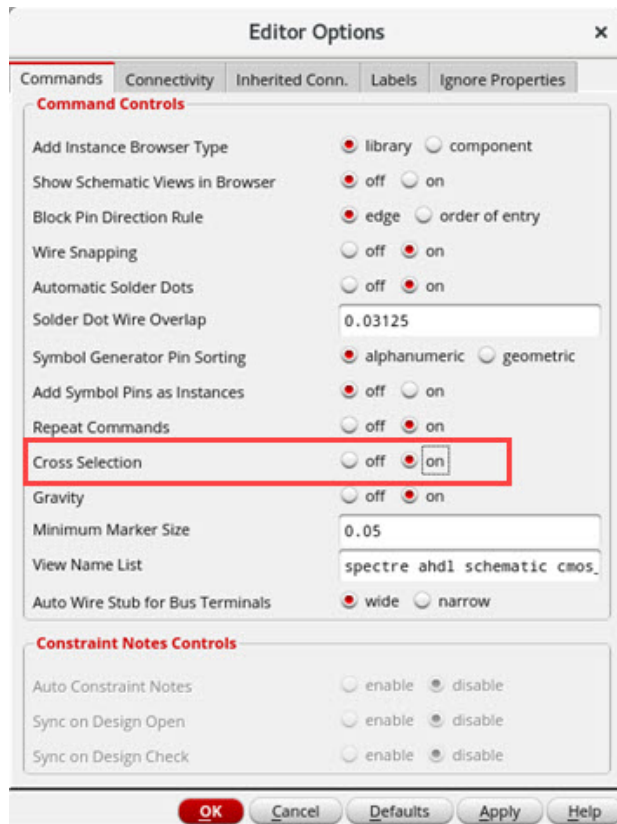
Cross-Selecting Instances

In AMS Designer, the applications that support cross-selecting instances are SimVision, the Cadence [hierarchy editor](#), and the Virtuoso® Schematic Editor. To enable cross-selecting in the hierarchy editor and the schematic editor, do the following:

1. In the Cadence hierarchy editor window, choose *View - Tree*.
Instances in the design appear on the *Tree View* tab.

 You can view cell bindings instead of instances by choosing *View - Parts Table*.

2. In the Virtuoso® Schematic Editor window, choose *Options - Editor*.
The Editor Options form appears.



3. Turn on *Cross Selection*.
4. Click *OK*.

Applications Supported for Cross-Selection

By setting cross-selection options in the Editor Options form, you can cross-probe module instances as summarized in the following table. The first entry in each cell, marked with_p, refers to probing primitives. The second entry, marked with_m, refers to probing modules.

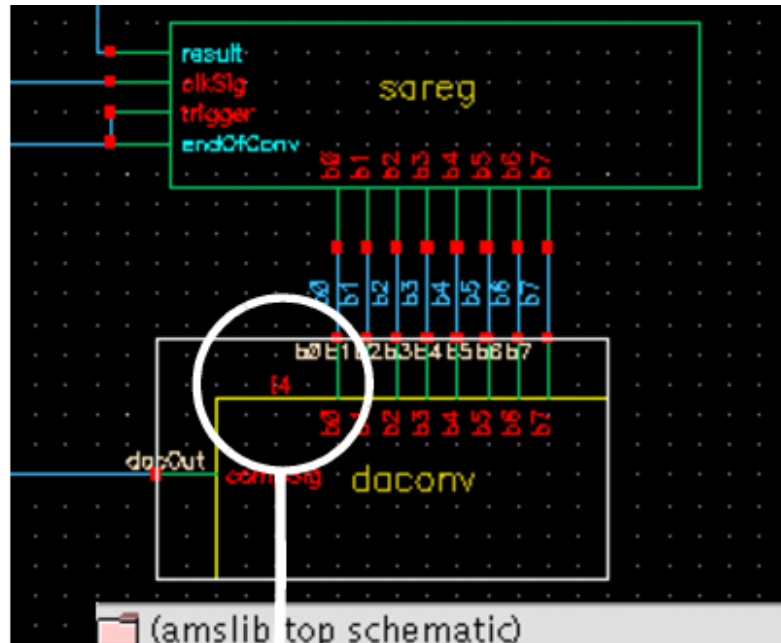
		Responding Application			
		Hierarchy Editor	Schematic Editor	SimVision	
Initiating Application	Hierarchy Editor		Yes _p Yes _m	No _p Yes _m	
	Schematic Editor	Yes _p Yes _m		No _p Yes _m	

	SimVision	No _p	Yes _m	No _p	Yes _m	
--	-----------	-----------------	------------------	-----------------	------------------	--

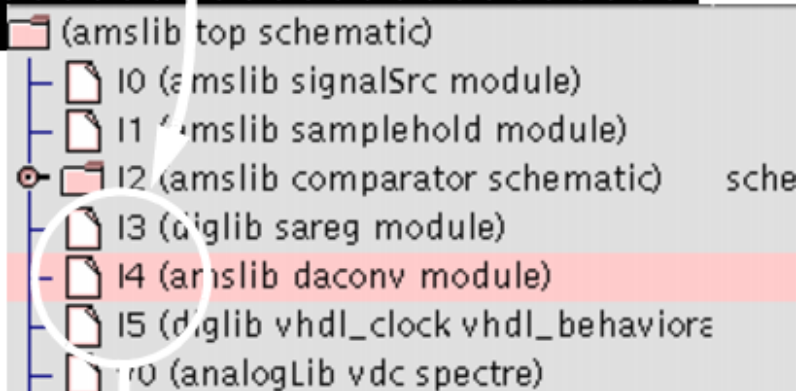
For example, if you select an instance in the schematic editor, the same instance is automatically selected in SimVision. Similarly, if you highlight a primitive in the hierarchy editor, the corresponding primitive in the schematic editor appears highlighted as well, and vice versa.

The following illustrates the cross-selecting that occurs when the `daconv` component (instance I4) is selected in the Schematic Editing window.

Select instance I4 in the Schematic Editing window.



Instance I4 highlights in the hierarchy editor...



...and in the SimVision windows.

```

18 sareg (* integer library_binding = "diglib")
19   .endOfConv( endOfConv ), .b5( b5 ), .b6( b6 ),
20   .b0( b0 ), .clkSig( clkSig ), .b4( b4 ),
21   .result( compOut ), .trigger( endOfConv )
22
23 daconv #( refVolt(5.000000)) (* integer library_binding = "amslib")
24   I4 ( .b2( b2 ), .b5( b5 ), .b6( b6 ),
25   .compSig( dacOut ), .b7( b7 ), .b0( b0 ),

```

Cross-Selecting Nets

In AMS Designer, the applications that support cross-selecting nets are SimVision, the schematic editor, and the Waveform window. You can enable cross-selecting in the schematic editor by following steps 2 through 4 in "[Cross-Selecting Instances](#)". With these options set, you can cross-probe nets as summarized in the following table.

Initiating Application	Responding Application			
		Schematic Editor	SimVision	Waveform
	Schematic Editor		Yes	Yes
	SimVision	Yes		Yes
	Waveform	No	Yes	

For example, if you select a net in the schematic editor, the same net automatically highlights in the SimVision window. However, if you select a net in the Waveform window, the schematic editor window does not reflect the selection.

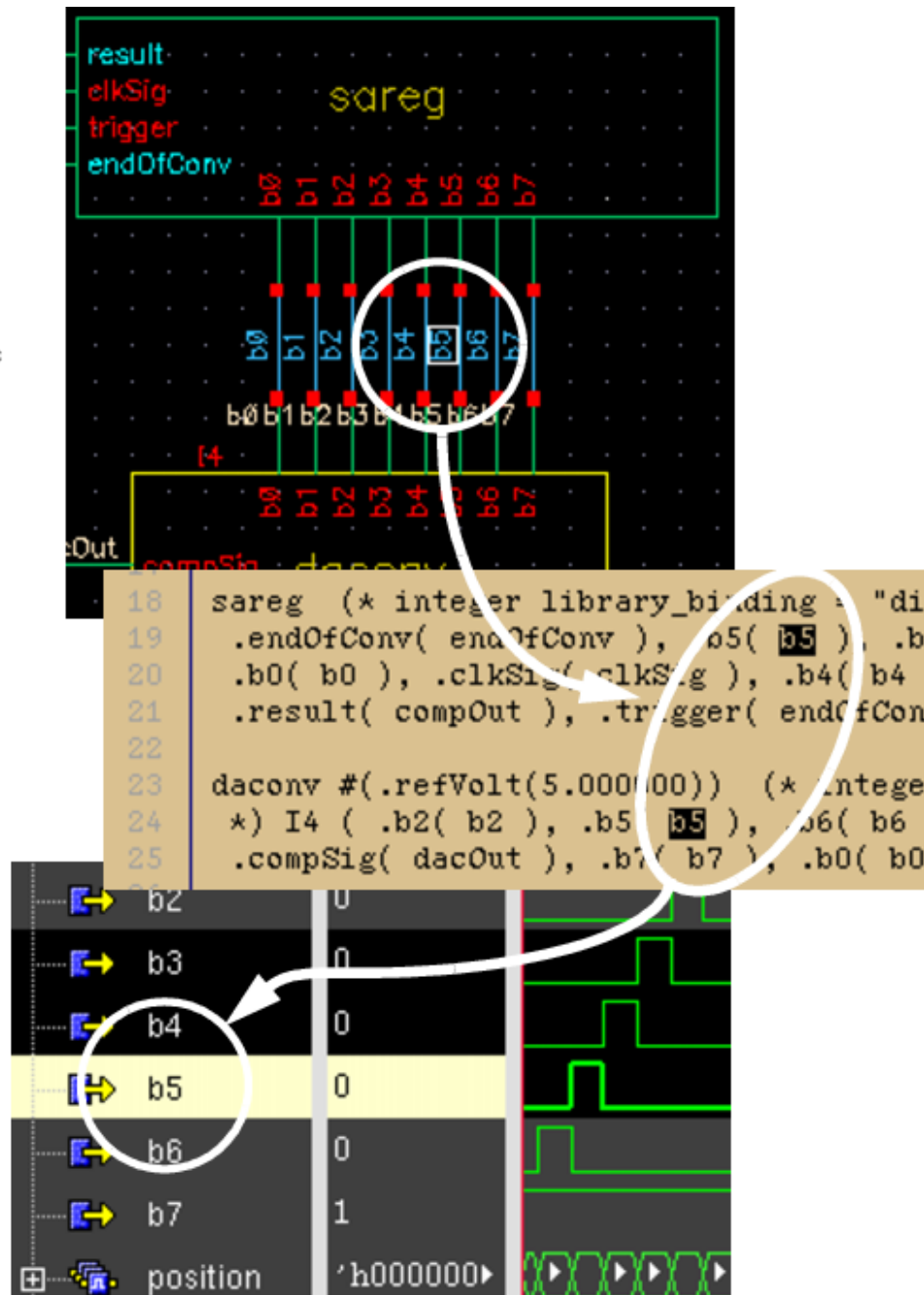
You cannot cross-probe global signals with AMS Designer.

The following illustrates the cross-selecting that occurs when the `b5` net is selected in the Schematic Editing window.

Select net b5 in the Schematic Editing window.

Net b5 highlights in the SimVision window...

...and in the Waveform window.

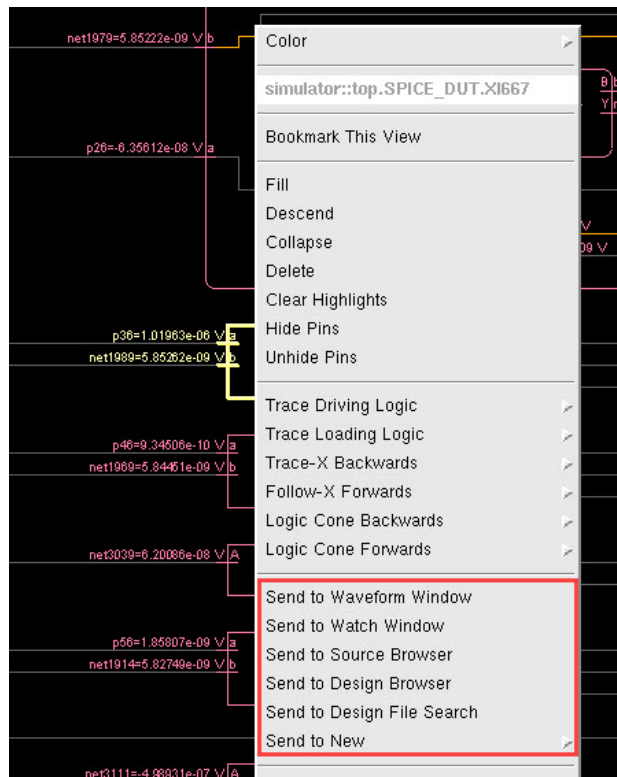


Viewing Instances in Other Windows

To cross-select and view the instances displayed in Schematic Tracer in the other SimVision MS windows, do the following:

1. Select and right-click the required instance in Schematic Tracer.

2. Select one of the following options:

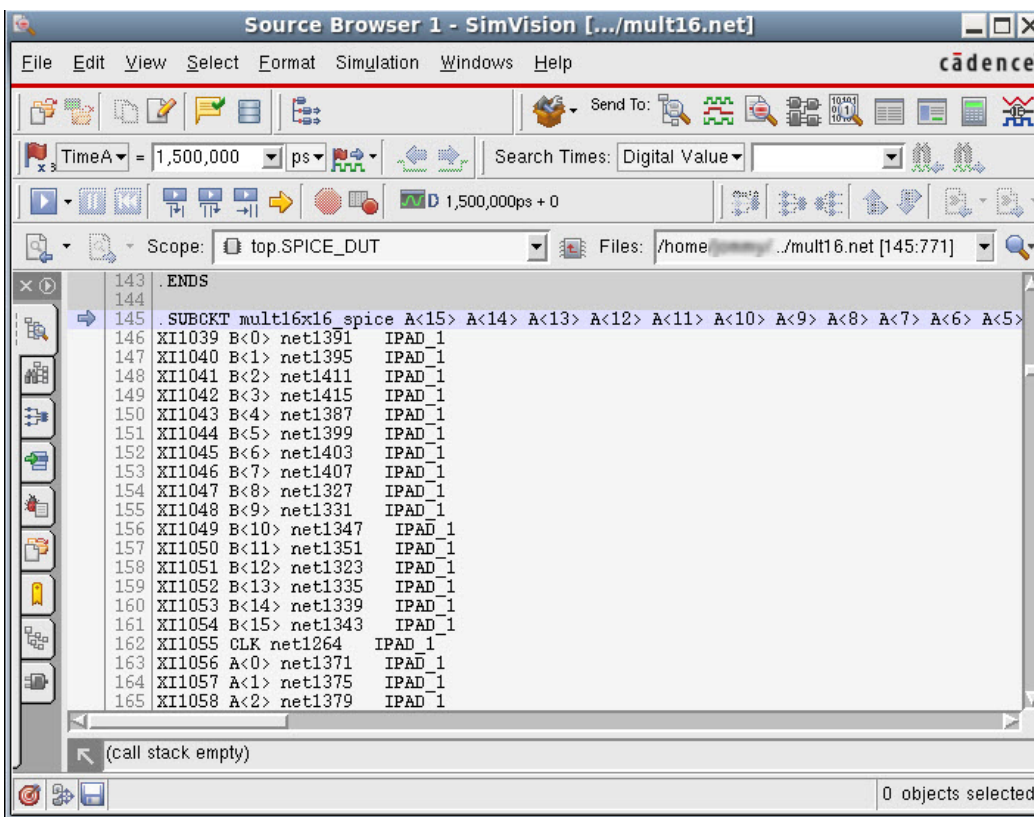


The instance will be opened in the selected window.

Working with Source Browser

The Source Browser window in SimVision MS displays the source code for digital instances and netlists for the analog instances of mixed-signal designs. In Source Browser, you can define the breakpoints of the connect modules to debug step-by-step. Similarly, you can debug analog instances with the help of the automatic instance line identification in the netlist.

Source Browser can be invoked using the  icon on the toolbar. It can also be invoked using the right-click option, *Send to Source Browser*, from the other Simvision MS windows. When launched, Source Browser looks like the following image.



The standard toolbars and menus perform the common functions available in all SimVision MS windows, such as controlling the simulation.

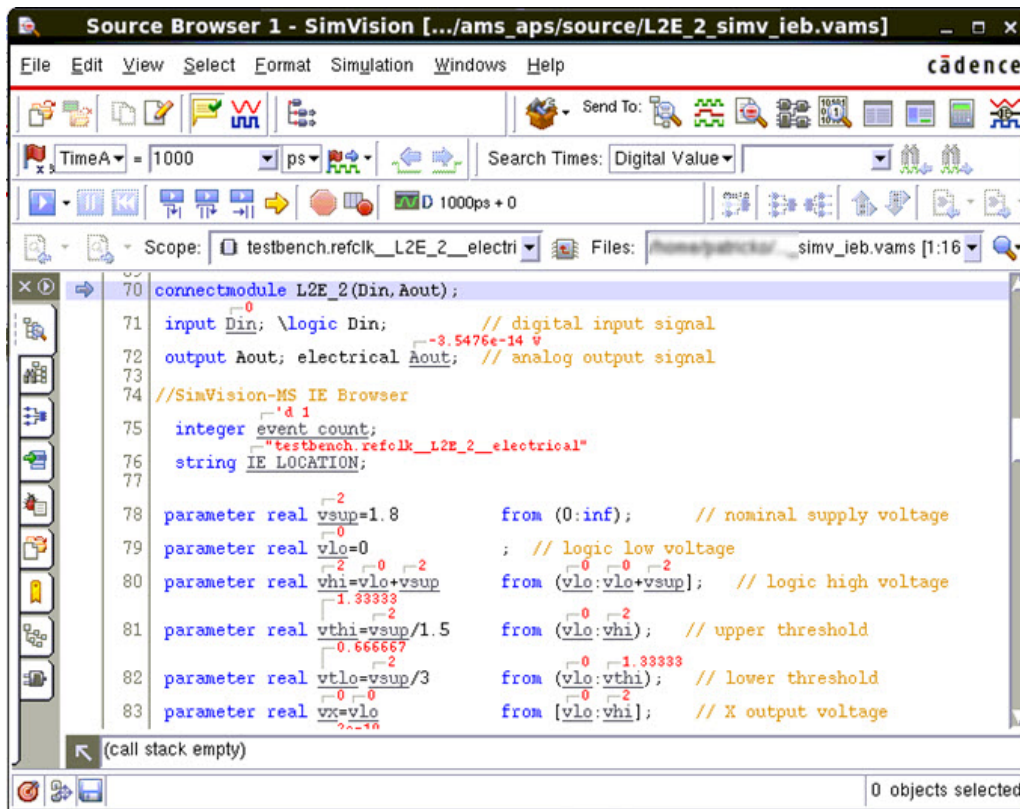
This chapter contains the following topics:

- [Viewing the Analog Source](#)

- Debugging the Connect Module Source
 - Auto Scope Focus
 - Defining Breakpoints
- Editing a Source File Using Your Own Editor
- Searching for a Line Number in the Source Code
- Searching for a Text String in the Source Code

Debugging the Connect Module Source

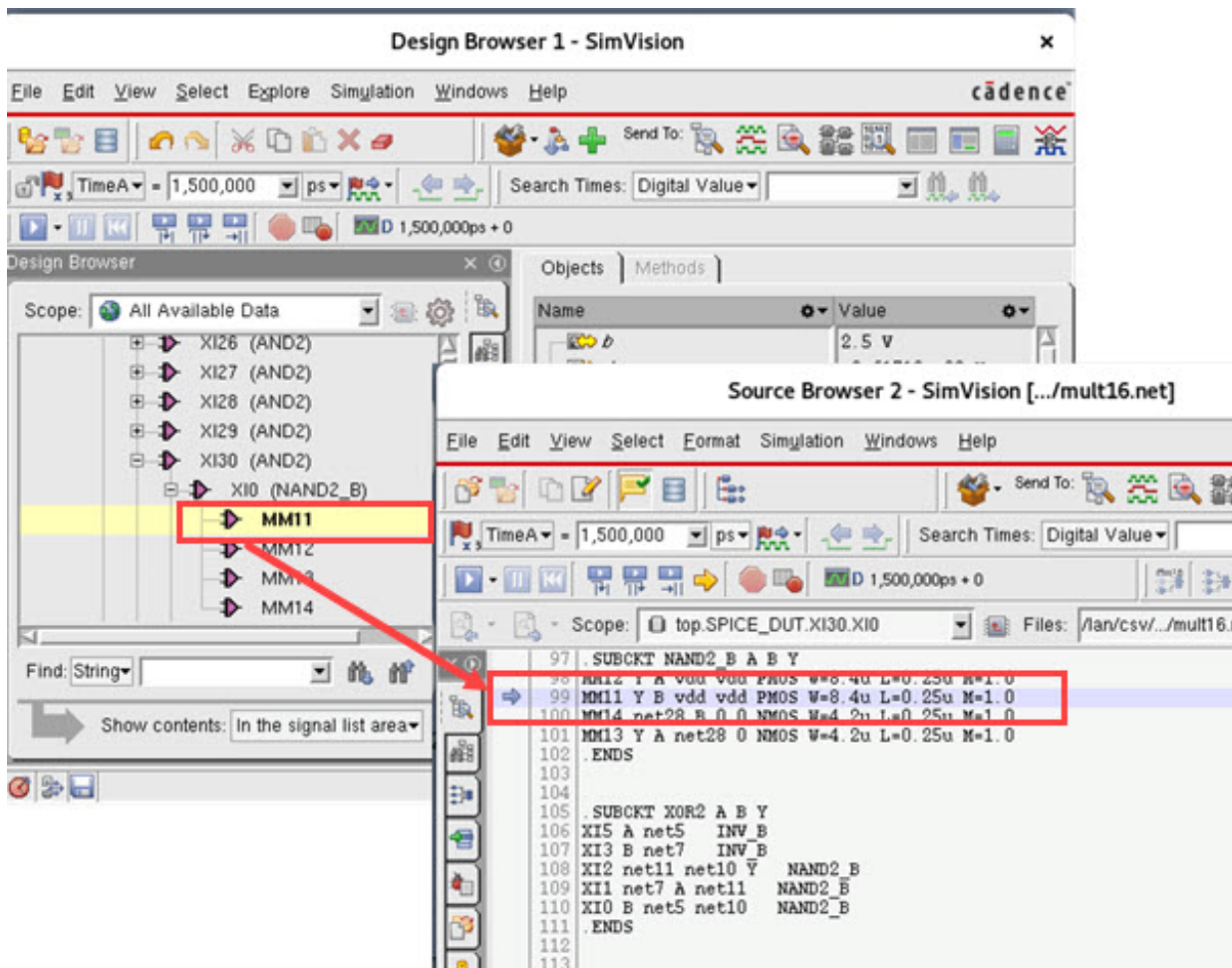
To view the source code of connect modules, select the connect module in Mixed Net Browser and click  icon on the toolbar. Source Browser will display the detailed code of the selected connect module. Click  to enable back annotation in Source Browser.



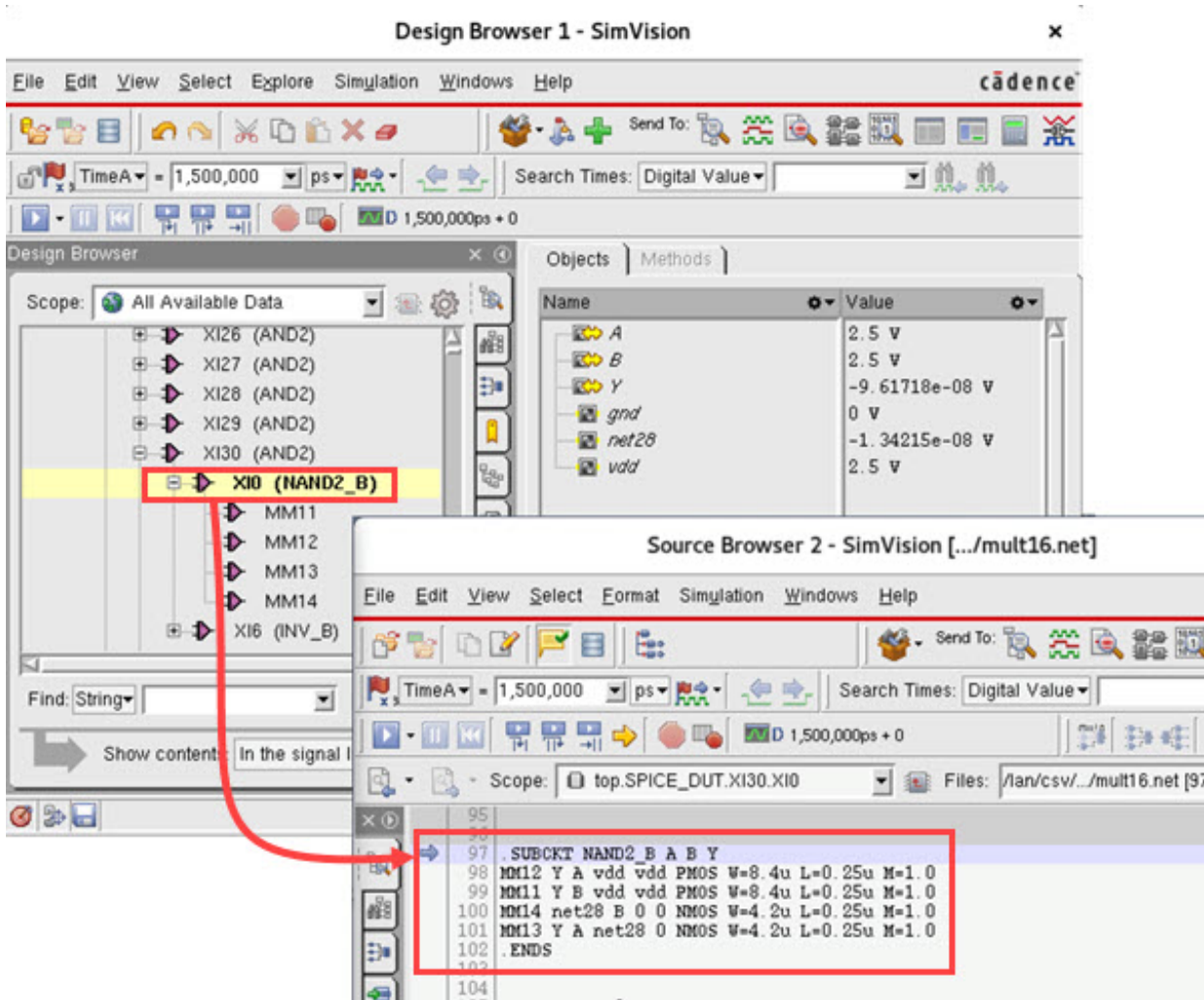
Auto Scope Focus

Auto scope focus, which is cross selection between Design Browser and Source Browser enhances the debugging experience for Spectre and SPICE netlists.

For example, when a MOS device is send from Design Browser to Source Browser, the exact line of code will be highlighted in the Source Browser. This can be seen in the image shown below.

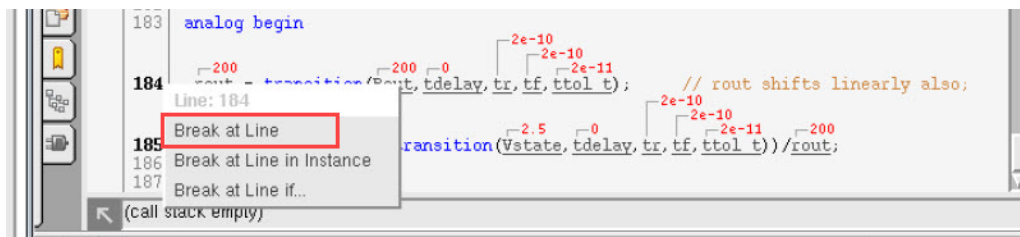


For a Spectre/SPICE subcircuit instance the behavior will be different. The auto scope will display the subcircuit definition as shown in the image below.

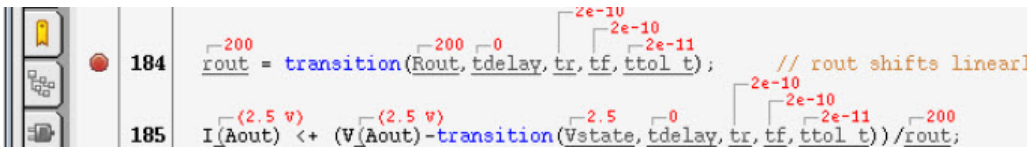


Defining Breakpoints

To define breakpoint for step-by-step debugging, right-click the required line in Source Browser and select *Break at line*.



A red dot is displayed on the left of the line number to indicate that the breakpoint is applied:



```
184 rout = transition(Rout, tdelay, tr, tf, ttol t); // rout shifts linearl
185 I(Aout) <+ (V(Aout)-transition(Vstate, tdelay, tr, tf, ttol t))/rout;
```

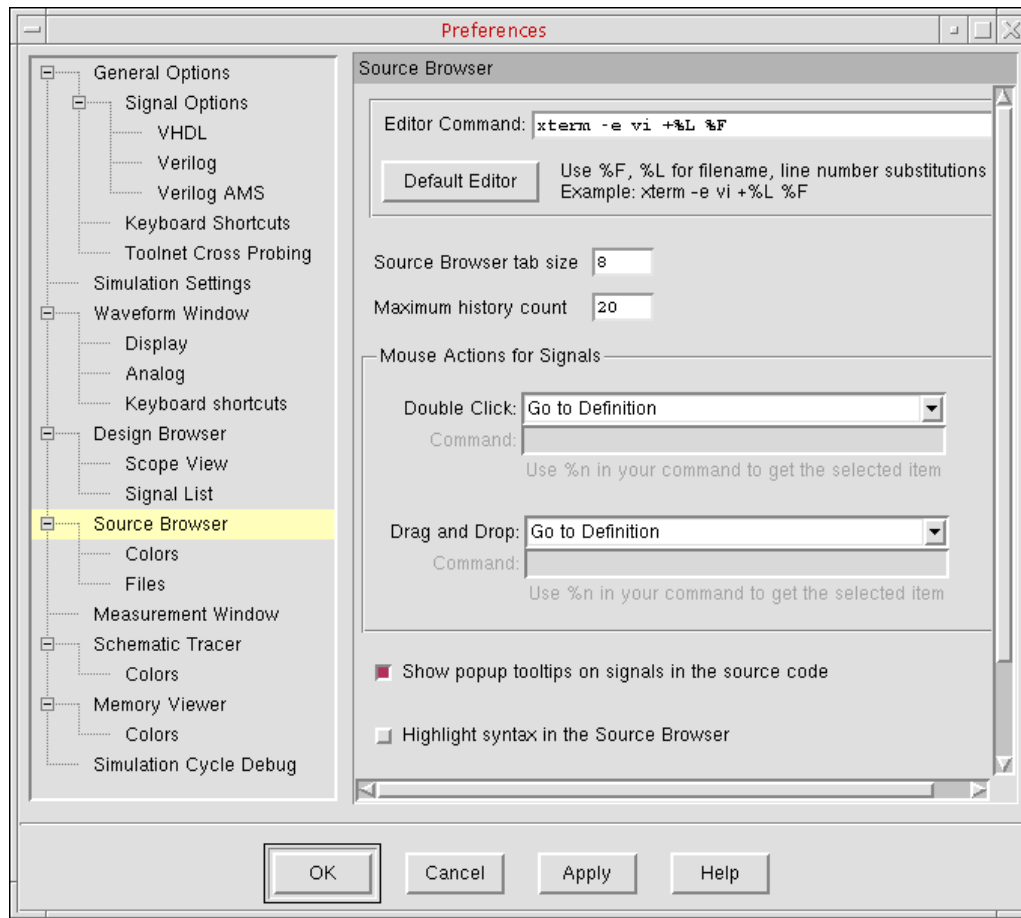
 Breakpoints can be defined only on line numbers that are in bold font.

All the defined breakpoints can be viewed by clicking the *Show Breakpoints Page* () button.

Editing a Source File Using Your Own Editor

vi is the default editor for editing source files. However, you can set any graphical text editor as your default editor. To specify the command to start the source file editor and specify a default editor, do the following:

1. In any SimVision MS window, choose *Edit - Preferences*.
2. On the Preferences form, select **Source Browser** in the list on the left side of the window.



3. In the Editor Command field, type the command to start the text editor.
You can also specify an editor other than the default vi editor.
4. Click *OK*.

To start the editor to edit the current file in the Source Browser, do the following:

In the Source Browser window, click the *Edit Source* () button on the toolbar:

The source file appears in your editor.

See also "[Accessing Design Source Code](#)" in the *Using the Source Browser* book.

Searching for a Line Number in the Source Code

To find a particular line number in the source code, do the following:

1. In the SimVision Source Browser window, choose *Edit - Go to Line*.

2. In the Line field on the form that appears, type the line number you want.
3. Click *OK*.

An arrow appears and points to the specified line in the code.

For more information, see "[Accessing Design Source Code](#)" in the Using the Source Browser book.

Searching for a Text String in the Source Code

To search for a text string in the source code, do the following:

1. In the SimVision Source Browser window, choose [Edit - Text Search](#).
2. Use the Text Search form to specify the search string and parameters.
3. Click *Find Next*.

Viewing the Analog Source

To view the analog source, select an analog instance in the any of the SimVision MS windows and click  icon on the toolbar. Source Browser will display the source of the selected instance.

Having information about the file that contains the selected instance and line number where it lies in the code gives you a powerful debugging ability.

Using SimVision with AMS Simulator Working with Source Browser

The image displays two windows from the Cadence SimVision environment:

- Schematic Tracer 2 - SimVision:** Shows a circuit schematic. A red box labeled '1' highlights a specific instance within the circuit.
- Source Browser 1 - SimVision [.../ams_aps/./source/analog/./Gates.sp]:** Shows the source code for the instance highlighted in the schematic. A red box labeled '3' highlights the line number where the instance is located. A red arrow points from this box to the schematic window.

Annotations:

- Red box '1' in the schematic window points to the instance location.
- Red box '2' in the schematic window points to the toolbar.
- Red box '3' in the source browser window points to the line number where the instance is located.
- Red text 'File that contains the instance' points to the source browser window.
- Red text 'Line number where the instance is located' points to the line number in the source browser window.

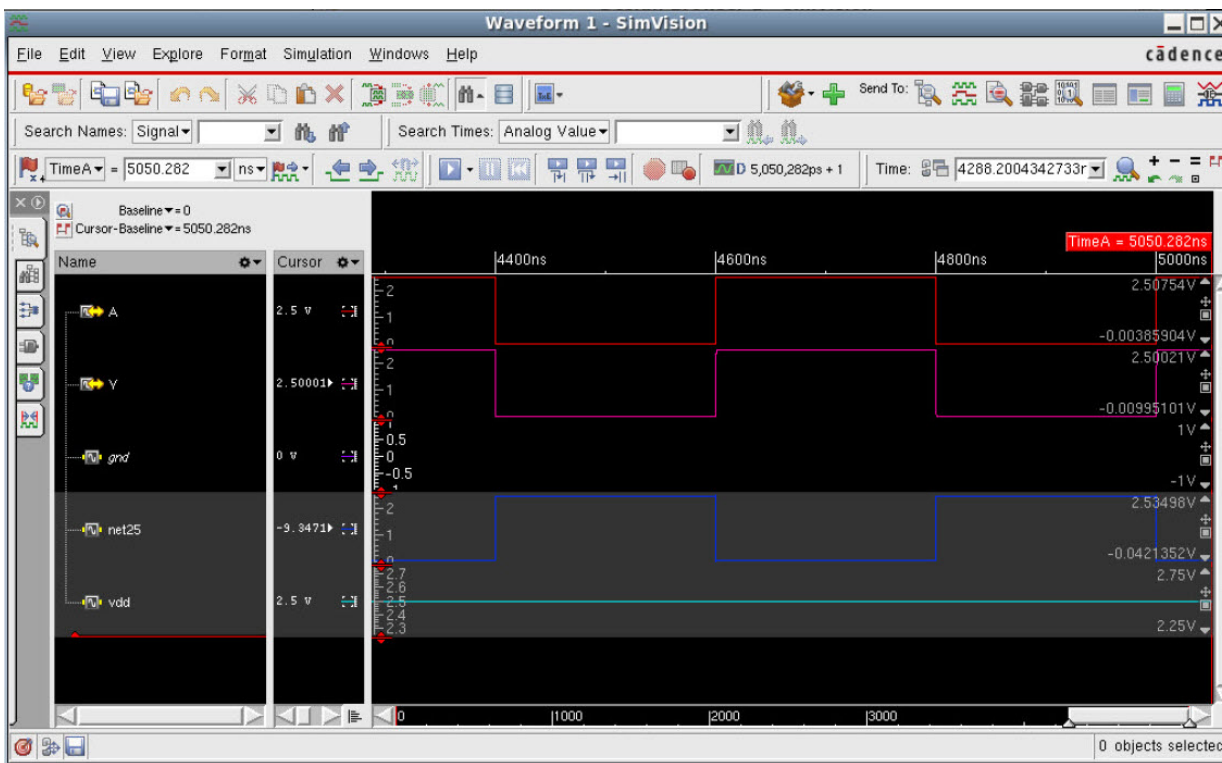
Source code snippet from the Source Browser:

```
137 .subckt ain_v_2 a y inh_vss inh_vdd
138 n1 y a inh_vss inh_vss rmos1 1=180e-9 v=+8.00000000e-07 m=+1.00000000e+00
139 mp1 y a inh_vdd inh_vdd pmos1 1=180e-9 v=+8.00000000e-07 m=+1.00000000e+00
140
141
142
143
144 .ends ain_v_2
145
146 * file name: amspll_ainv_schematic.s
147 * subcircuit for cell: ain_v
148 * generated for: hspices
149 * generated on dec 1 12:01:14 2005.
150
151 * terminal mapping: a = a
152 * y = y
153 * gnd! = inh_vss
154 * vdd! = inh_vdd
155 .subckt ain_v_3 a y inh_vss inh_vdd
156 nml y a inh_vss inh_vss rmos1 1=180e-9 v=+8.00000000e-07 m=+1.00000000e+00
157 mp1 y a inh_vdd inh_vdd pmos1 1=180e-9 v=+1.60000000e-06 m=+1.00000000e+00
158
159
160 .ends ain_v_3
161
162 * end of subcircuit definition.
```

Working with the Waveform Window

The Waveform window lets you view the waveforms that represent the signal transitions during a simulation.

The Waveform window can be invoked using the  icon on the toolbar. It can also be invoked using the right-click option, *Send to Waveform Window*, from the other Simvision MS windows. When launched, the Waveform window looks like the following:



This chapter contains the following topics:

- [Displaying Waveforms in the Waveform Window](#)
 - [Opening the SimVision Waveform Window](#)
 - [Using the Waveform Window to View a Post-Simulation Database](#)
 - [Using the Waveform Window for Interactive Waveform Display during Simulation](#)
- [Using Analog Mnemonic Maps in SimVision](#)

- [Creating Analog Mnemonic Map](#)
- [Applying Analog Mnemonic Map](#)
- [Limitations of Analog Mnemonic Maps](#)
- [Example of Creating a Mnemonic Map](#)
- [Searching Values for RNM or Electrical Signal](#)
- [Sending Signals to ViVA](#)
- [Enabling Real Value Probe Filtering](#)

Displaying Waveforms in the Waveform Window

You can use the SimVision Waveform window to display and analyze waveforms. How you plot a signal depends on the kind of probing that you use.

- If you use Tcl probing for analog and digital objects,
 - a. Select the net in either Schematic Tracer, or the Design Browser.
 - b. Choose *Windows - Waveform* in Design Browser to plot the signal in an existing Waveform window, or choose *Windows - New - Waveform* to plot the signal in a new Waveform window.
- If you probe objects from the analog simulation control file,
 - a. Start, but do not finish, the simulation.
For example, you might choose *Simulation - Advance - Timepoint* in the Design Browser.
 - b. Select the net in either the Schematic Editing window or Design Browser.
 - c. Choose *Windows - Waveform* in the Design Browser window to plot the signal in an existing Waveform window, or choose *Windows - New - Waveform* to plot the signal in a new Waveform window.

To plot signals after the simulation finishes, follow the procedures described in Chapter 4, "Accessing the Design Hierarchy," and Chapter 6, "Managing Simulation Databases," in the *SimVision User Guide*.

Opening the SimVision Waveform Window

When working with waveforms, there are two use models:

- Using a Waveform window to [view a post-simulation database](#) after the simulation ends.
- Using a Waveform window for [interactive waveform display](#) so that you can view the waveforms as the simulation progresses.

Using the Waveform Window to View a Post-Simulation Database

 When you use SimVision to view a post-simulation database, you are using it in "PPE mode". For more information, see the *SimVision User Guide*.

To view the waveforms using SimVision after the simulator creates the database, do the following:

1. Open the viewer by typing

```
simvision -waves
```
2. Choose *File - Open Database*.
3. In the Open Database browser, navigate to the `.trn` file for your data.
4. Click *Open*.
5. In the Waveform window, choose *Windows - New Design Browser*.
6. In the Design Browser, select the signals you want to display.
7. Click *Send selected object(s) to target waveform window* to add the signals to the Waveform window.

Using the Waveform Window for Interactive Waveform Display during Simulation

To view the waveforms as the simulation progresses, do the following:

1. [Preselect the signals you want to view](#).
2. [Run the simulation](#).

See the following books for details:

- [Using the Design Browser](#)
- [Running SimVision in Simulation Mode](#)

If you do not preselect objects, you can use the Design Browser to probe signals and add them to the Waveform window.

Using Analog Mnemonic Maps in SimVision

Mnemonic maps associate signal values with strings, colors, line shapes, and icons. For example, you can define a mnemonic map to associate a name with each state of a state machine, or you can define a mnemonic map to display all x values with a background color of your choice.

When you create an analog mnemonic map, you define not only the text that is displayed for a signal value, but also a background color of the waveform in the SimVision Waveform window.

- [Creating Analog Mnemonic Map](#)
- [Applying Analog Mnemonic Map](#)
- [Limitations of Analog Mnemonic Maps](#)
- [Example of Creating a Mnemonic Map](#)

Creating Analog Mnemonic Map

You can create analog mnemonic maps in one of the following methods:

- **Use the SimVision TCL command.**

You must use the `-analog` option with the `mmap` new SimVision command. The syntax is:

```
mmap new -analog -name label -contents {  
{- range { <lowValue> <highValue>} -label "name" -bgcolor}
```

You can define one or more ranges for a given mnemonic map entry using `-range {<lowValue> <highValue>}`. Each entry is specified as a list with a range, a label, and background color that specifies how a waveform is drawn when its value matches one of the value ranges for the mnemonic map entry.

So, for a signal set with a map with value in one of the value ranges, a box with a label is displayed in the Waveform trace.

For more information on the `mmap` command, refer to the *SimVision Tcl Commands Guide*.

- **Define analog mnemonic maps in a script file and source it to SimVision.**

In the SimVision GUI, choose *File – Source Command Script*, and in the *Select SimVision Command Script* dialog box select the command file with Analog Mnemonic Maps.

Applying Analog Mnemonic Map

To apply a mnemonic map to a signal, perform one of the following methods:

- In the SimVision Waveform window, then click *Format – Radix/Mnemonic* and choose the analog mnemonic map.
- In the SimVision Waveform window, right-click the signal value in the *Cursor* column and choose *Radix/Mnemonic – analog mnemonic map*.

For more information on the SimVision interface, see the *SimVision: Using the Waveform Window User Guide*.

Limitations of Analog Mnemonic Maps

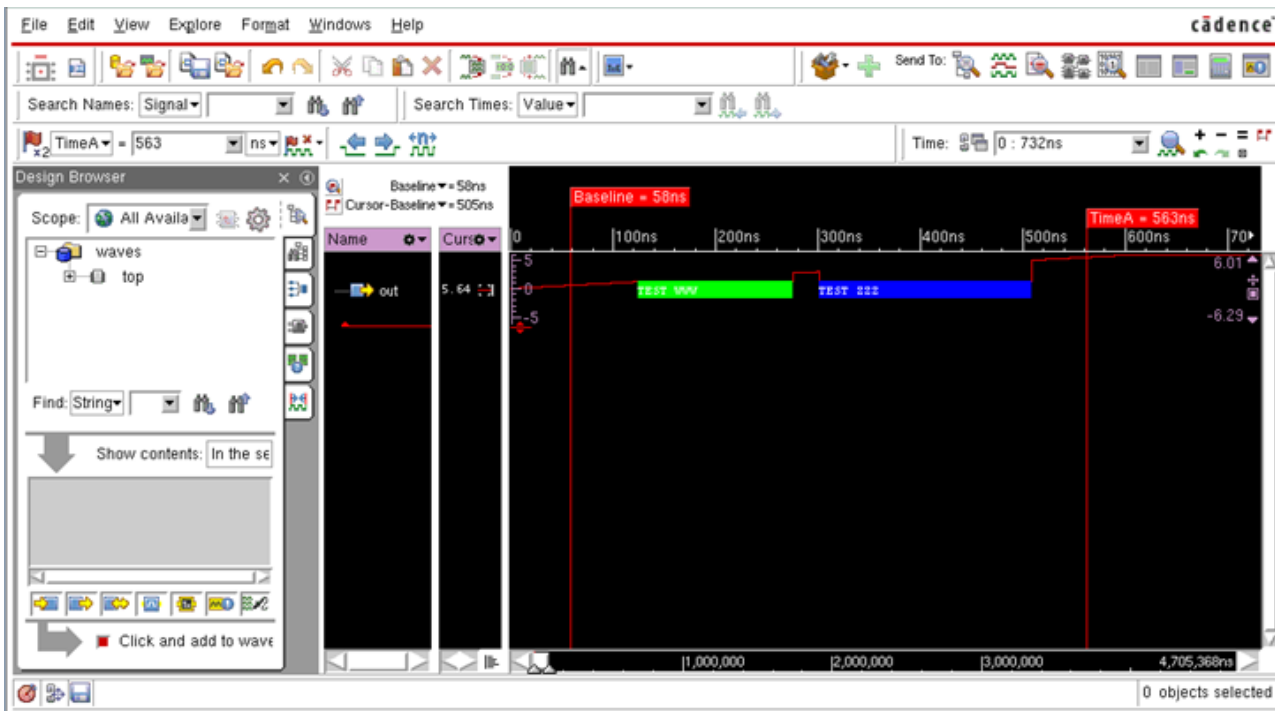
The following limitations exist with the analog mnemonic map feature:

- Analog mnemonic maps are not listed in the *Mnemonic Maps* tab of the *Properties* window.
- Analog mnemonic maps cannot be applied to non-analog traces such as integers.
- Analog Mnemonic Maps cannot be applied on composite types such as SV UDN wreal/vams, wreal/var real, and SV UDT/UDN structures. You can only select members of a composite type (individually or as a group) and then apply the analog mnemonic map to those selected signal(s).
- You cannot apply an analog mnemonic map to a group of signals from a SV UDT/UDN structure, if one or more of the signals in that group are of a non- analog type (logic type or integer type).
- Only the `-label`, `-bgcolor` options of the `mmap` Tcl command are supported. The other `mmap` options documented in the *SimVision Tcl Commands Guide* are not supported.

Example of Creating a Mnemonic Map

The following example illustrates using the `mmap` TCL command. It creates an analog mnemonic map, `Analog Mnemonic Map` and creates two entries for the mnemonic map. Each entry is specified with a range, a label, and the background color for the mnemonic map.

```
mmap new -analog -name "Analog Mnemonic Map" -contents {  
{-range {1.25 2.76} -label "TEST WWW" -bgcolor green}  
{-range {3.01 5.09} -label "TEST ZZZ" -bgcolor blue}
```

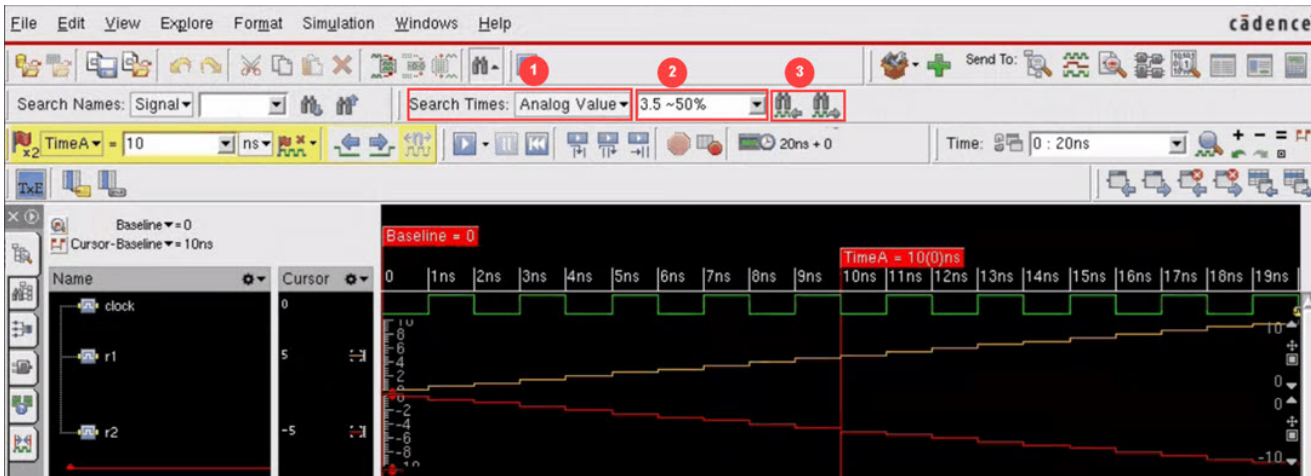


Searching Values for RNM or Electrical Signal

In the Waveform window, you can find real values on a selected waveform and also the search for a range, a center or tolerance, and more. This feature enhances the usability of SimVision with real-valued signals and is particularly relevant for mixed-signal designs.

To search for values on a waveform,

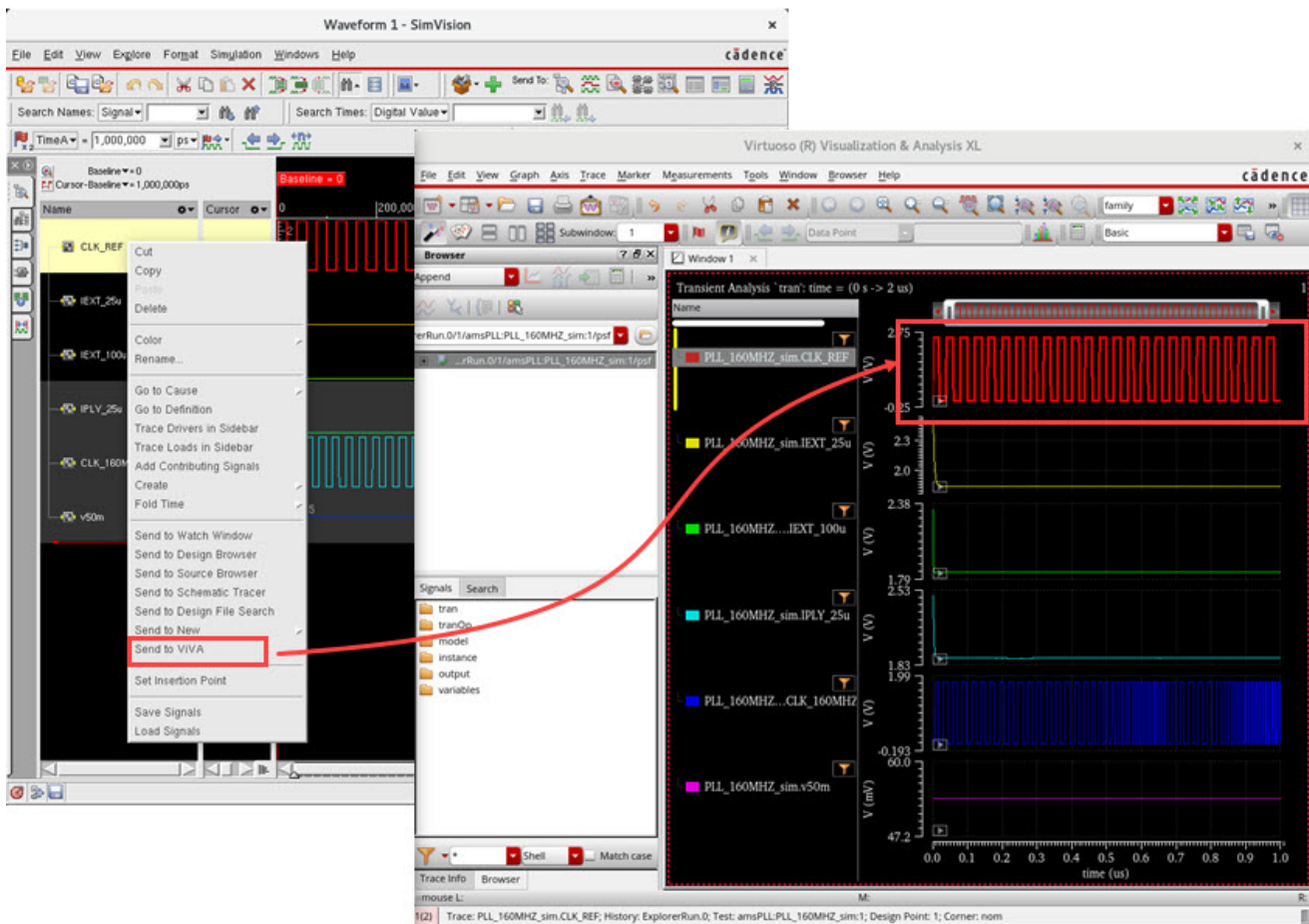
1. Select a value type from the *Search Times* drop-down.
2. Specify a value or range in the input field.
3. Click the search button.



Sending Signals to ViVA

SimVision MS provides the ability to send the signals from SimVision to Virtuoso Visualization and Analysis XL (ViVA). Given that ViVA is an analog-centric tool, it provides many features to work with the analog signals.

To view the signals in ViVA, select the signal on the right pane of the Waveform window and select the  icon on the toolbar. Alternatively, right-click the signal and select *Send to ViVA*.



This launches ViVA and the selected signal is displayed in ViVA.

Enabling Real Value Probe Filtering

The AMS Designer simulator allows you to specify a precision (delta value) for real number probes to remove recorded values whose delta is less than the precision specified. This helps reduce the waveform database size for real number model simulations and improve simulation performance; and, still maintain the simulation model accuracy.

You can enable real value filtering for all real value probes using the `-shm_filter_group <precision>` global option, with `xrun` or `xmsim` commands.

When the `-shm_filter_group` option is used, the delta from the previously dumped value is computed. If this delta is at or greater than the precision tolerance, then the value is dumped to the waveform database, otherwise it is skipped. This reduces the waveform database size by not recording values less than the specified precision.

The value for `precision` can be integer, float or scientific notation. When an integer value is

specified it represents the number of points after the decimal place; so, an integer value of 2 would represent a delta value of 0.01. When float or scientific notation are used its value is used as the delta value.

 This option does not filter probe values computed by the analog solver (electrical signals).