



Agilent Technologies

ADS 2008
January 2008
RFIC Dynamic Link

Advanced Design System 2008

© Agilent Technologies, Inc. 2000-2008

395 Page Mill Road, Palo Alto, CA 94304 U.S.A.

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Agilent Technologies, Inc. as governed by United States and international copyright laws.

Acknowledgments

Mentor Graphics is a trademark of Mentor Graphics Corporation in the U.S. and other countries. Microsoft®, Windows®, MS Windows®, Windows NT®, and MS-DOS® are U.S. registered trademarks of Microsoft Corporation. Pentium® is a U.S. registered trademark of Intel Corporation. PostScript® and Acrobat® are trademarks of Adobe Systems Incorporated. UNIX® is a registered trademark of the Open Group. Java™ is a U.S. trademark of Sun Microsystems, Inc. SystemC® is a registered trademark of Open SystemC Initiative, Inc. in the United States and other countries and is used with permission. MATLAB® is a U.S. registered trademark of The Math Works, Inc.. HiSIM2 source code, and all copyrights, trade secrets or other intellectual property rights in and to the source code in its entirety, is owned by Hiroshima University and STARC.

Errata The ADS product may contain references to "HP" or "HPEESOF" such as in file names and directory names. The business entity formerly known as "HP EEsof" is now part of Agilent Technologies and is known as "Agilent EEsof". To avoid broken functionality and to maintain backward compatibility for our customers, we did not change all the names and labels that contain "HP" or "HPEESOF" references.

Warranty The material contained in this document is provided "as is", and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license. Portions of this product include the SystemC software licensed under Open Source terms, which are available for download at <http://systemc.org/>. This software is redistributed by Agilent. The Contributors of the SystemC software provide this software "as is" and offer no warranty of any kind, express or implied, including without limitation warranties or conditions or title and non-infringement, and implied warranties or conditions merchantability and fitness for a particular purpose. Contributors shall not be liable for any damages of any kind including without limitation direct, indirect, special, incidental and consequential damages, such as lost profits. Any provisions that differ from this disclaimer are offered by Agilent only.

Restricted Rights Legend If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as "Commercial computer software" as defined in DFAR 252.227-7014 (June 1995), or as a "commercial item" as defined in FAR 2.101(a) or as "Restricted computer software" as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to Agilent Technologies' standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Contents

- About RFIC Dynamic Link
 - ◦ Advanced Design System
 - Virtuoso Schematic Composer
 - RFIC Dynamic Link
 - - RFIC Dynamic Link Use Model
 - What's in this Manual
- Configuring RFIC Dynamic Link
 - ◦ System Requirements
 - - Hardware Requirements
 - - Software Requirements
 - - License Requirements
 - Installing RFIC Dynamic Link
 - Configuring the Cadence Installation
 - - Customizing a Cadence Installation
 - - Customizing Multiple Cadence Installations
 - - Avoiding Modifications to Your Cadence Installation
 - Configuring the Software
 - - Configuring the UNIX Environment
 - - Configuring the ADS Install Directory
 - - Modifying the RFIC Dynamic Link-Cadence Initialization File
 - - Modifying the Configuration File
 - Managing Projects and Designs
- Getting Started with RFIC Dynamic Link
 - ◦ Setting up the Examples Directory
 - Starting the Cadence Design Framework
 - - Opening a Cadence Composer Schematic
 - Linking with Advanced Design System
 - Opening a Test Schematic Design
 - Adding a Symbol of the Cadence Cellview
 - Adding Design Variables
 - Adding Model Files
 - Performing a DC Simulation
 - Annotating DC Operating Points to a Selected Cellview
 - Saving Node Voltages and Pin Currents with RFIC Dynamic Link
 - - Additional Notes:
 - Performing an S-parameter Simulation
 - - Displaying Your Results
 - Performing a Parameter Optimization
 - - Verifying Your Results
 - Ending the Session
- Using RFIC Dynamic Link
 - ◦ Launching Advanced Design System
 - Adding an Instance of a Cadence Design
 - Pushing into the Design Hierarchy
 - Using Design Variables
 - - Adding and Editing Design Variables

Advanced Design System 2008

- Updating Cadence Design Variables
- Adding Model Files
- - Adding a Netlist File Include Component
 - Accessing the Netlist File Include Dialog
 - Summarizing the Netlist File Include Component
- Annotating a Cellview
- - Annotating DC Voltages to a Cadence Cellview
 - Annotating DC Currents to a Cadence Cellview
 - Displaying Voltages or Currents from a Previous Simulation
 - Creating Symbols for Hierarchical Subcircuits with cdsTerm
- Netlisting, Simulating, and Displaying Data
- - Netlisting and Simulating a Design
 - Viewing Netlists
 - - Viewing Netlists from Advanced Design System
 - Viewing Netlists from the Cadence Schematic Window
 - Net and Instance Name Mapping
 - Expression Name Mapping
 - Using Global Nodes
- Running a DSP and Analog/RF Cosimulation with RFIC Dynamic Link
- - RFIC Dynamic Link DSP and Analog/RF Cosimulation Example
- Tuning and Optimizing Designs
- - Tuning Cadence Instance Parameters and Design Variables
 - - A Dynamic Link For Cadence Tuning Example
 - Optimizing Designs
 - - Statistical Variations
 - Updating the Cadence Cellview
- Using Additional Features of RFIC Dynamic Link
- - Freezing Selected Subcircuits
 - - Setting the Freeze Parameter
 - Generating a Cadence Subcircuit Netlist
 - Setting the netlistFile Parameter
 - Using "Freeze" Mode to Simulate a Design in ADS Standalone
 - Compatibility between Advanced Design System and Cadence Tools
 - - Support for Duplicate Pin Names
 - Using Buses
 - Setting up Unnamed Nets
 - Support for pPar and iPar
 - Using Inherited Connections
 - Using S-parameter File Devices from analogLib
 - Encoding Cadence Designs for RFIC Dynamic Link
 - - Encoding Process Summary
- Using Switch Views, Stop Views and the Hierarchy Editor
- - Expanding Hierarchy with the Dynamic Link Netlister
 - Using the Hierarchy Editor with RFIC Dynamic Link
 - - Placing the config view in ADS
- Troubleshooting RFIC Dynamic Link
- - Known Problems and Solutions
 - Installation and Use Checklist
- Command Reference for RFIC Dynamic Link
- - ADS Schematic window DynamicLink Menu

- Cadence Schematic Window DynamicLink Menu
- CIW Tools > ADS Dynamic Link Menu Item
- Glossary for RFIC Dynamic Link

About RFIC Dynamic Link

Agilent Technologies and Cadence Design Systems both offer powerful EDA design tools. Many of today's design engineers prefer to use a combination of these tools to take advantage of the strengths of both design environments. Because of this desire to use multiple tools, Agilent Technologies has developed the RFIC Dynamic Link for Cadence. The Dynamic Link enables both top-down and bottom-up design and simulation in Advanced Design System (ADS) using IC designs from the Cadence database.

Advanced Design System

Advanced Design System has been developed specifically to simulate the entire communications signal path. This unique solution integrates the widest variety of proven RF, DSP, and electromagnetic design tools into a single, flexible environment. Building on years of expertise developing new technologies for our EDA tools, Advanced Design System provides a broad range of high-performance capability. This makes it easy to explore design ideas, then model the electrical and physical design of the best candidates.

Virtuoso Schematic Composer

The Virtuoso Schematic Composer from Cadence Design Systems is a hierarchical design entry tool used by RFIC circuit designers. Useful for both analog and digital designs, the database created is accessible by the Cadence simulation and physical layout tools. The tool supports multi-sheet schematics, including cross-referencing, symbol creation, automatic HDL cell template generation, global nets and hierarchical property definition for most database objects. The tool also provides hierarchical checking of connectivity, consistency of different cell representations and label attachments.


RFIC Dynamic Link

RFIC Dynamic Link is an EDA framework integration software product based on Inter-Process Communication (IPC), rather than data file translation, maximizing data integrity and ease of use. This manual describes how to install and configure the RFIC Dynamic Link product and assists you in designing and analyzing analog mixed-signal and RF circuits via Dynamic Link. [Getting Started Tutorial](#) is provided to help you quickly get started with using the RFIC

Dynamic Link. For information on Library Customization, refer to the RFIC Dynamic Link Library Guide.

RFIC Dynamic Link Use Model

The RFIC Dynamic Link use-model coincides with that of both Cadence and ADS, with only a few exceptions. Essentially, the Affirma Analog Circuit Design Environment user interface is replaced with Advanced Design System and all of its functionality. The Affirma features that are not directly replaced by ADS are provided on the DynamicLink pull-down menu in the Cadence Virtuoso Schematic window.

 **Note**
If the DynamicLink pull-down menu does not appear in the Cadence Virtuoso Schematic window, choose Tools > ADS Dynamic Link in the Cadence schematic window.

Usage assumes basic familiarity with the Cadence IC Design Framework II (DFII) , including Virtuoso schematic capture and Affirma Analog Circuit Design Environment , as well as basic familiarity with design and simulation in the Advanced Design System.

Additional Information

- Wherever a shell variable is set, this manual uses the K-shell syntax.
- Unless otherwise mentioned, assume case sensitivity.
- Terminology used for Agilent Technologies and Cadence EDA Tools is frequently different. For example, a project in ADS is similar to a library in Cadence and design in ADS is similar to a cellview in Cadence.

What's in this Manual

The goal of this manual is to help you get started, providing relevant examples that teach you how to use the software, and show you where you can get more information as you need it. This manual contains:

- [Configuring RFIC Dynamic Link](#) describes the system requirements and how to install and configure the software.
- [Getting Started Tutorial](#) steps you through the process of simulating a circuit using components from the Dynamic Link analogLib library. Other examples are also included to help you become more familiar with the product.
- [Using RFIC Dynamic Link](#) provides information on launching ADS from a Cadence Schematic window, performing some basic operations and closing the Dynamic Link between ADS and Cadence.

- [Netlisting, Simulating, and Displaying Data](#) describes the procedures for netlisting and simulating a design as well as viewing the netlist from either ADS or a Cadence schematic window. Information on net, instance and expression name mapping is also provided.
- [Running a DSP and Analog/RF Cosimulation with RFIC Dynamic Link](#) provides an example of how to run an ADS Ptolemy Cosimulation using RFIC Dynamic Link.
- [Tuning and Optimizing Designs](#) provides information on tuning and optimizing designs using the ADS tuning and optimization capabilities.
- [Using Additional Features of RFIC Dynamic Link](#) includes a collection of Dynamic Link features such as "Freezing" selective subcircuits. Compatibility features covering support for Duplicate Pin Names, Bus-ports, Buses and Bundles, Unnamed Nets and pPar and iPar are also discussed.
- [Using Switch Views, Stop Views and the Hierarchy Editor](#) provides information on using switch views, stop views and the Hierarchy Editor in Dynamic Link.
- [Troubleshooting](#) provides information that can help you resolve common problems with RFIC Dynamic Link.
- [Command Reference](#) describes the function of each of the menu selections available for using RFIC Dynamic Link.

Configuring RFIC Dynamic Link

This chapter describes system requirements and how to install and configure the software. You may require help from a UNIX or EDA Administrator to complete these tasks. For general system requirements, refer to [Check the System Requirements](#) in the UNIX and Linux Installation documentation.

System Requirements

This section provides a reference to the minimum hardware and operating system requirements and also describes the EDA Framework and License requirements necessary for using the RFIC Dynamic Link.

Hardware Requirements

For information on the minimum hardware requirements for installing and using RFIC Dynamic Link, refer to [Check the System Requirements](#) in the UNIX and Linux Installation documentation.

Software Requirements

The specific RFIC Dynamic Link version requires that you use a compatible Advanced Design System version. For example, if you are using RFIC Dynamic Link 2006A, you must also use ADS 2006A.

Dynamic Link supports Cadence DFII versions on UNIX & Linux operating system versions from vendors which run this Cadence software. For a summary of supported platforms, refer to [Check the System Requirements](#) in the UNIX and Linux Installation documentation, or contact Cadence Design Systems Inc.

Note that the following Cadence context files must be present under your Cadence installation directory in order to run RFIC Dynamic Link:

```
<Cadence Installation Dir>/tools/dfII/etc/context/spectrei.cxt
```

```
<Cadence Installation Dir>/tools/dfII/etc/context/asimenv.cxt
```

If either of these files is missing, please contact Cadence Design Systems Inc. for information on how to install it.

License Requirements

In addition to your standard Advanced Design System licenses, the following additional product licenses are required.

RFIC Dynamic Link License

- trans_idf

Cadence Licenses

- OASIS_Simulation_Interface
- 34510 - Affirma(TM) analog design environment
- 300 - Virtuoso(R) layout editor (if using layout)



Note


You must purchase all required Cadence licenses from Cadence Design Systems.

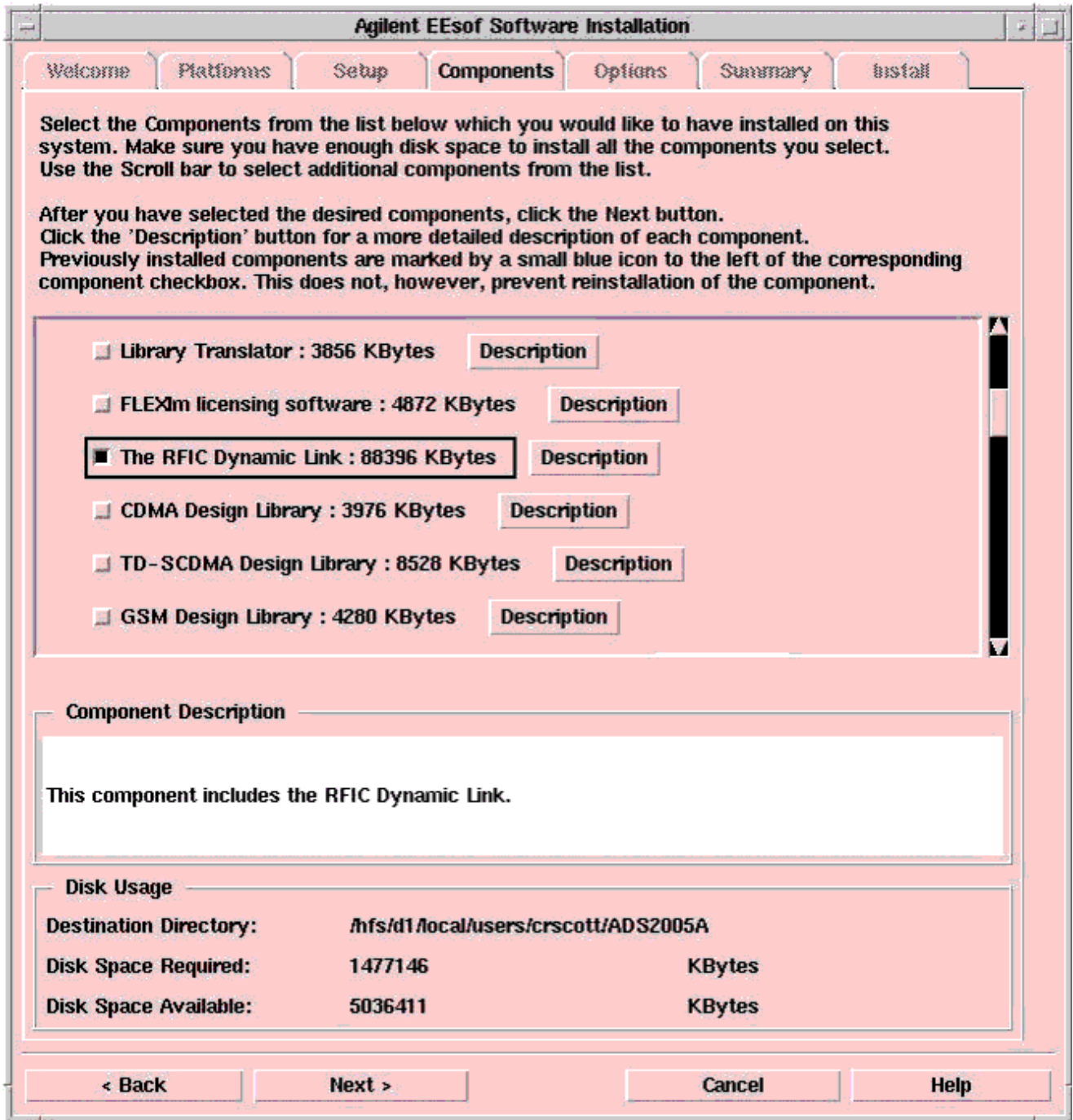
Installing RFIC Dynamic Link

The RFIC Dynamic Link installation procedure continues to be improved to make it easier for you to install and configure.

To install RFIC Dynamic Link:

1. Follow instructions in the [UNIX and Linux Installation](#) documentation to run the SETUP utility and load the install program.
2. After the Agilent EEsof Installation Manager starts, you are prompted to select one of the following installation options:
 - Typical - If you choose a Typical installation, the RFIC Dynamic Link will not be installed.
 - Complete - If you choose a Complete installation, the RFIC Dynamic Link will be automatically installed.
 - Custom - If you choose a Custom installation, you must select both Simulators and Design Entry and The RFIC Dynamic Link components from the scroll-down list in the Agilent EEsof Software Installation dialog box.

 **Note**
The Simulators and Design Entry component is the basic ADS software, including the Design Environment, Data Display, and Analog/RF Systems and Signal Processing simulators. This is a minimum requirement for RFIC Dynamic Link.



3. After the installation is complete, you will need to run the idfConfigCadence script to configure Cadence for use with RFIC Dynamic Link. This script is located under:
\$HPESOF_DIR/bin/idfConfigCadence

Note
For ADS 2001, the Cadence configuration script was located at \$HPESOF_DIR/idf/config/configCadence.


For more information on the idfConfigCadence script, refer to [Customizing a Cadence Installation](#).

For more information on installation procedures, refer to the["../install/wwhelp.htm" UNIX and Linux Installation] documentation.

Configuring the Cadence Installation

After the installation procedure (see [Installing RFIC Dynamic Link](#)), you will need to run the idfConfigCadence script if you want to customize your Cadence installation for Dynamic Link. This script configures the Cadence install directory for use with ADS by adding or modifying five files.

If you prefer not to modify your existing Cadence installation(s), refer to the section on [Avoiding Modifications to Your Cadence Installation](#).

 **Note**
You must have write access to your Cadence install directory in order to run the idfConfigCadence script.

The following files are created:

<Cadence Installation Dir>/tools/dfII/etc/tools/ads/.cdsenv

<Cadence Installation Dir>/tools/dfII/etc/tools/ADSsim/.cdsenv

<Cadence Installation Dir>/tools/dfII/etc/tools/adsDL/.cdsenv

<Cadence Installation Dir>/share/cdssetup/hierEditor/templates/ads

<Cadence Installation Dir>/tools/dfII/etc/skill/hnl/ads.ile

<Cadence Installation Dir>/tools/dfII/etc/skill/si/caplib/ads.ile

An additional file, <Cadence Installation Dir> /tools/dfII/etc/tools/auCore/.cdsenv, is edited to add ads to the Tool Filter list of simulators.



Note

The files described in this section are created or modified based on detailed instructions provided in the Cadence OASIS Direct Integrator's Guide . The information in the Cadence manual __ describes how to integrate a simulator into the Cadence Analog Design Environment using the Cadence direct toolkit .

Important While deviating from the information described in the Cadence OASIS Direct Integrator's Guide may work for RIFC Dynamic Link, the files have been specifically set up to follow the Cadence documentation. Agilent Technologies does not support deviations from this information.

For more information, refer to the Cadence documentation:

- OASIS Integrator's Guide , Product Version 4.4.5, December, 1999 - Pages 4-12, 4-13, 6-2 and 6-6.
- OASIS Direct Integrator's Guide , Product Version 4.4.6, September, 2001- Pages 67, 101 and 105.

Customizing a Cadence Installation

Before running the idfConfigCadence script, set HPEESOF_DIR to your ADS installation directory and set your PATH to include Cadence software.

The idfConfigCadence command uses the following general syntax:

```
idfConfigCadence {-h | -ls | -rm}
```

Simply entering idfConfigCadence at the command line with no options runs the script to configure Cadence for RFIC Dynamic Link. The following table shows a list of the options and definitions used by the idfConfigCadence command.

Option Definitions for idfConfigCadence

Option	Definition
-h	This option can be used to display the idfConfigCadence help file.
-ls	This option can be used to list the Cadence configuration files for ADS/RFIC Dynamic Link and indicate whether Cadence is ready for ADS RFIC Dynamic Link or not.
-rm	This option removes the Cadence configuration for ads/RFIC Dynamic Link by deleting the files previously installed with the idfConfigCadence script and removing

ads from the Cadence tool Filter. For more information on the Cadence Tool Filter, refer to your Cadence documentation.

Understanding the idfConfigCadence Script

When the idfConfigCadence script is executed without any options, the script performs the following operations:

- Copies
`$HPPEESOF_DIR/idf/config/adsCdsenvFile`
to
`<Cadence Installation Dir>/tools/dfII/etc/tools/ads/.cdsenv`
- Copies
`$HPPEESOF_DIR/idf/config/ads.hierEd`
to
`<Cadence Installation Dir>/share/cdssetup/hierEditor/templates/ads`
- Creates two new empty files called ads.ile in the following locations:
`<Cadence Installation Dir>/tools/dfII/etc/skill/hnl/ads.ile`
`<Cadence Installation Dir>/tools/dfII/etc/skill/si/caplib/ads.ile`

Customizing Multiple Cadence Installations

The idfConfigCadence script configures the Cadence installation directory for use with Dynamic Link. However, if a new version of the Cadence software is subsequently installed or you have multiple Cadence installations, neither will have the Dynamic Link configuration. In this case, you would need to run the idfConfigCadence script as described in [Configuring the Cadence Installation](#) for each Cadence installation.

Avoiding Modifications to Your Cadence Installation

If you do not want to change your Cadence installation or create a Cadence shadow directory tree for RF Design Environment (RFDE) or RFIC Dynamic Link, there is an alternative approach available. You will need to set the following three environment variables before starting Cadence Design Framework II:

```
CDS_LOAD_ENV=CSF
```

Advanced Design System 2008

```
ossSimUserHnlDir=$HPEESOF_DIR/idf/ads_site/hnl
```

```
ossSimUserSiDir=$HPEESOF_DIR/idf/ads_site/si
```

In addition to the above, you will need to add the following path to your setup.loc file:

```
$HPEESOF_DIR/idf/ads_site
```

Sourcing the setCSF.ksh (or setCSF.csh) sample script under \$HPEESOF_DIR/bin is one method of accomplishing the tasks above. Assuming you are using the Korn shell, the following is a list of steps for using the sample scripts to setup the environment for RFDE or Dynamic Link in order to work with your original Cadence installation:

1. Change to a directory where you have write permission. For example,

```
cd /tmp
```

2. Set your HPEESOF_DIR to your RFDE or ADS installation root directory.
3. Add \$HPEESOF_DIR/bin to your path. For example,

```
PATH=$HPEESOF_DIR/bin:$PATH
```

4. Ensure your Cadence and Agilent licenses are set correctly for your environment.
5. Ensure that the Cadence <cadence_installation_root> /tools/bin and <cadence_installation_root> /tools/dfII/bin directories are in your PATH.
6. Set the CDS_LOAD_ENV, ossSimUserHnlDir, ossSimUserSiDir, and setup.loc files appropriately. For example, if using the Korn Shell,

```
. $HPEESOF_DIR/bin/setCSF.ksh
```

If you are using the C Shell, use the following command,

```
source $HPEESOF_DIR/bin/setCSF.ksh
```

7. Ensure that you have included RFDE Libraries. For example,

```
echo INCLUDE \${HPPEESOF_DIR}/cdslibs/rfde.lib > cds.lib
```

8. Ensure that the RFDE and Dynamic Link startup file ads.ini will be loaded upon starting Cadence Design Framework II. For example,

```
cp ${HPPEESOF_DIR}/idf/examples/.cdsinit .
```

9. Launch Cadence by entering `icms&` or `msfb&`.

If Cadence fails to locate the RFDE or Dynamic Link OASIS files under your Cadence installation, the software will look for these OASIS files under `HPPEESOF_DIR/idf/ads_site`.

The alternate search location will be used because of the way `CDS_LOAD_ENV`, `ossSimUserHnldir`, `ossSimUserSiDir`, and `setup.loc` are set in step 6 above. RFDE and Dynamic Link will still operate normally with your original Cadence installation. This may be the preferred approach when you have multiple Cadence installations, in which case you would need to add RFDE OASIS files to every individual installation or create a shadow directory for each installation.

According to the Cadence Application Infrastructure User Guide, Product Version 5.01, Cadence looks for the Cadence setup search file, `HPPEESOF_DIR/idf/ads_site/cdssetup/setup.loc`, if `HPPEESOF_DIR` is defined in the environment; otherwise, it will look for `<your_installation_directory>/share/cdssetup/setup.loc`.

For more information about `CDS_LOAD_ENV`, `CSF`, `setup.loc`, or OASIS, refer to your Cadence documentation or contact your Cadence customer support representative.

Configuring the Software

This section describes the various aspects of configuring and/or modifying the software to provide additional flexibility.

Configuring the UNIX Environment

There are several UNIX environment variables relevant to Dynamic Link. These are described in the following table.

UNIX Environment Variables

Environment Variable	Description
IDF_CONFIG_FILE	The name of the configuration file (only the file name, not the entire path). Default value is <code>idf.cfg</code> .
IDF_ADS_PROJ_DIR	If this environment variable is defined, ADS will attach to the specified project when RFIC Dynamic Link launches ADS.

IDF_DEBUG_MODE	If set to TRUE debugging will be turned on and additional log messages will be written to the CIW; the log files mps.log and emx.log are also created. Default value is FALSE.
IDF_LOG_FILE	Name of the file (only the file name, not the entire path) to which ADS messages, normally written to stderr are redirected. Default value is idf.log .
PATH	The UNIX path variable.

Configuring the ADS Install Directory

The installation procedure configures the ADS install directory by automatically adding, modifying or replacing some files and/or directories.

The following files and/or directories are created, modified or replaced:

```

$HPEESOF_DIR/bin/idfmp
$HPEESOF_DIR/bin/idf
$HPEESOF_DIR/bin/idfConfigCadence
$HPEESOF_DIR/idf
$HPEESOF_DIR/idf/ael
$HPEESOF_DIR/cdslibs
$HPEESOF_DIR/idf/config
$HPEESOF_DIR/idf/examples
$HPEESOF_DIR/idf/skill
$HPEESOF_DIR/circuit/symbols/idfSymbol.dsn
$HPEESOF_DIR/tools/lib/dpkg/info/idf\*
```

Modifying the RFIC Dynamic Link-Cadence Initialization File

The RFIC Dynamic Link-Cadence initialization file (.cdsinit) is located in the directory \$HPEESOF_DIR/idf/config/.cdsinit . Append it to or load it from the first available .cdsinit file from the list below:

- <Cadence Installation Dir> /tools/dfII/local/.cdsinit
- ./cdsinit
- \$HOME/.cdsinit

Modifying the Configuration File

Dynamic Link comes with the default configuration file \$HPEESOF_DIR/idf/config/idf.cfg . This file is used to set various site-specific or user-specific options. It is searched for and read sequentially from the following locations in the

order given, so that settings in files read later override those of earlier files:

```
$HPEESOF_DIR/idf/config/
```

```
$HOME/hpeesof/config/
```

```
./
```

The name of the configuration file can be set via the UNIX environment variable `IDF_CONFIG_FILE`. By default the configuration file is named `idf.cfg`.

The configuration file consists of lines in the form `< parameter > = < value >`. The various parameters that can be set in the configuration file are listed below, with brief descriptions and an example for each. If no configuration file is found or some parameters are not set, internal default values are used. Note that wherever a file name is required for a configuration parameter value, it may be specified with a path prefix that is a UNIX environment variable value or using standard UNIX conventions such as `~`. A complete example configuration file can be found at the end of this section.

- **Switch View List:** If an instance has none of the views listed in the switch view list, the netlister reports an error. The default is `ads sch.model schematic`. Example:
`IDF_SWITCH_VIEW_LIST = "ads sch.model schematic"`
- **Stop View List:** The netlister identifies primitives with a stop list. When it reaches a view that is listed in both the switch view and stop view lists, the instance is netlisted and no expansion occurs below this level. There is normally no reason to change the Stop View list to anything other than `ads`. Example:
`IDF_STOP_VIEW_LIST = "ads"`
- **Project Path:** This indicates the path to the ADS project to attach to upon starting ADS via RFIC Dynamic Link.

Note
Unless Cadence and ADS are installed on the same workstation where you run RFIC Dynamic Link, using this parameter is not recommended.

Example:

```
IDF_ADS_PROJ_DIR = /tmp/ads_prj
```

- **Netlist Filter:** When you have site customization that is not performed by the supplied netlister, this option enables you to specify the name of the program or script used to post-process the netlist generated by the Dynamic Link netlister. Example:
`IDF_NETLIST_FILTER = "$HPEESOF_DIR/bin/myfilter"`
If you want to change all "vss!" in the netlist to 0 (zero or the ground in ADS), the script (`myfilter`) would look similar to the one below:

```
sed -e s/\ "vss!\ "/0/g
```
- **Netlist Suffix:** This is the suffix for the Advanced Design System netlist file generated by the Dynamic Link netlister for each sub-circuit. The default is `.net`. Example:
`IDF_NETLIST_SUFFIX = ".net"`

Advanced Design System 2008

- **Debug Mode:** This option enables you to turn debugging messages on or off. These messages appear in order to help you determine the cause and/or location of problems.
By default, debugging is turned off. To enable debugging, set this option to TRUE . Example:
`IDF_DEBUG_MODE = TRUE`
- **Symbol Generation:** This option enables you to specify whether to generate a missing symbol using the Cadence symbol generator or the Advanced Design System symbol generator. The Cadence symbol generator is used as the default. Example:
`IDF_CADENCE_SYMBOL = FALSE`
- **User AEL Files:** Users can define their own AEL functions and load them into the ADS environment via a list of comma separated file names. These files get loaded just after the Dynamic Link environment is initialized.
Example:
`IDF_USER_AEL_FILES = "file1.ael, $HOME/file2.atf"`
- **Expression Mapping:** This causes sub-strings in Cadence expressions to be mapped to corresponding sub-strings in ADS expressions in the netlist file and/or in the design variable values used. Example:
`IDF_EXPR_MAP = "foo bar"`
- **Freezing Subcircuit Netlists:** When this variable is set to yes (or TRUE), all Cadence subcircuits for which a netlist already exists are not re-netlisted. The default is FALSE . Example:
`IDF_FREEZE_NETLISTS = TRUE`



Note

The Freeze parameter can be set to yes (TRUE) or no (FALSE).

To freeze selected subcircuits, see [Freezing Selected Subcircuits](#).

- **Message Timeout:** This specifies the timeout period in seconds for message actions initiated in ADS to complete in DFII. The default is 45; however, you will need to set it to a value larger than the default for netlisting large Cadence circuits. Example:
`IDF_MSG_TIMEOUT = 200`
- **Other Parameters:** The following parameters should not be altered.
`IDF_AEL_FILES = "globals.atf, utils.atf, commands.atf, callbacks.atf, symbol.atf"`
`IDF_PDE_EXEC = hpeesofde`
`IDF_PDE_ARGS = "-env de_sim"`
`IDF_SCALE_FACTOR = 2.0`

Example Configuration File

```
IDF_SWITCH_VIEW_LIST = "ads sch.model schematic"
```

```
IDF_NETLIST_FILTER = "~/bin/mynetlistfilter"
```

```
IDF_NETLIST_SUFFIX = ".net"
```

Managing Projects and Designs

Your Cadence designs will remain in their original locations. They are not copied, translated, or otherwise modified. When ADS starts up in Dynamic Link mode, it puts you in a project directory defined by the `IDF_ADS_PROJ_DIR` environment variable. If this variable is not defined, only the ADS Main window will open and no project will be attached. This is the recommended option.

To change this behavior, you can do one of the following:

- Specify your own startup project directory by defining `IDF_ADS_PROJ_DIR` in your UNIX environment or via the configuration file `idf.cfg`.
- After the Advanced Design System has come up, go to the Main window and open a new or existing project (File > New Project or File > Open Project).

For more information on projects and design files, refer to [Managing Projects and Designs](#) in the Schematic Capture and Layout documentation.

Getting Started with RFIC Dynamic Link

This tutorial steps you through the process of simulating a circuit using components from the Agilent version of the analogLib library. Other examples are also included to help you become familiar with the product.

Both the drop-down menus and icons are described to help familiarize you with the Advanced Design System environment.

Setting up the Examples Directory

To set up the examples directory, first ensure that:

- The `HPEESOF_DIR` environment variable is set to your ADS installation directory
- `$HPEESOF_DIR/bin` is in your `PATH`
- Your environment is set up to run the Cadence `icms` or `msfb` tool

From any directory of your choice, enter:

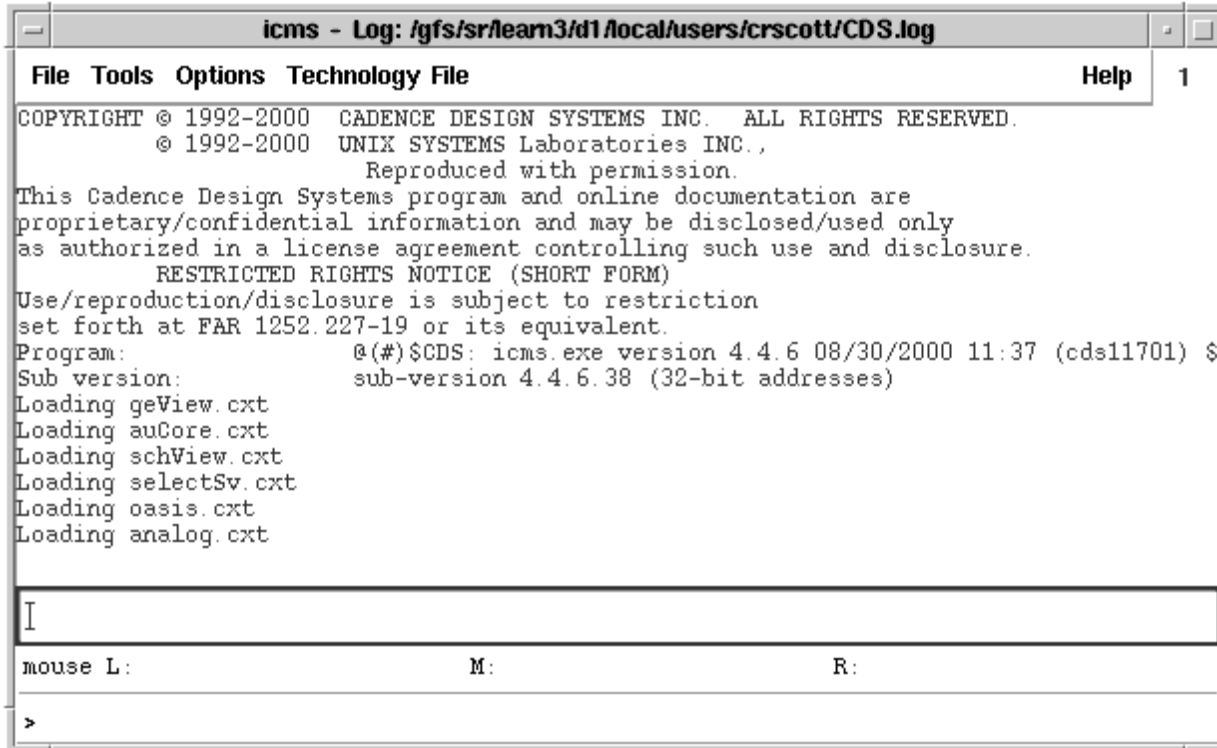
```
cp -r $HPEESOF_DIR/idf/examples ./
```

cd examples

Note
 This must be done before attempting the Getting Started Tutorial. The cds.lib file under the examples directory defines libraries provided by Dynamic Link.
 The .cdsinit file under this directory loads the Dynamic Link .cdsinit file which then loads the context files required to run Dynamic Link.

Starting the Cadence Design Framework

Ensure that you are in the examples directory then launch Cadence Design Framework II by typing the appropriate command (typically icms or msfb). The Cadence Command Interpreter Window (CIW) appears.



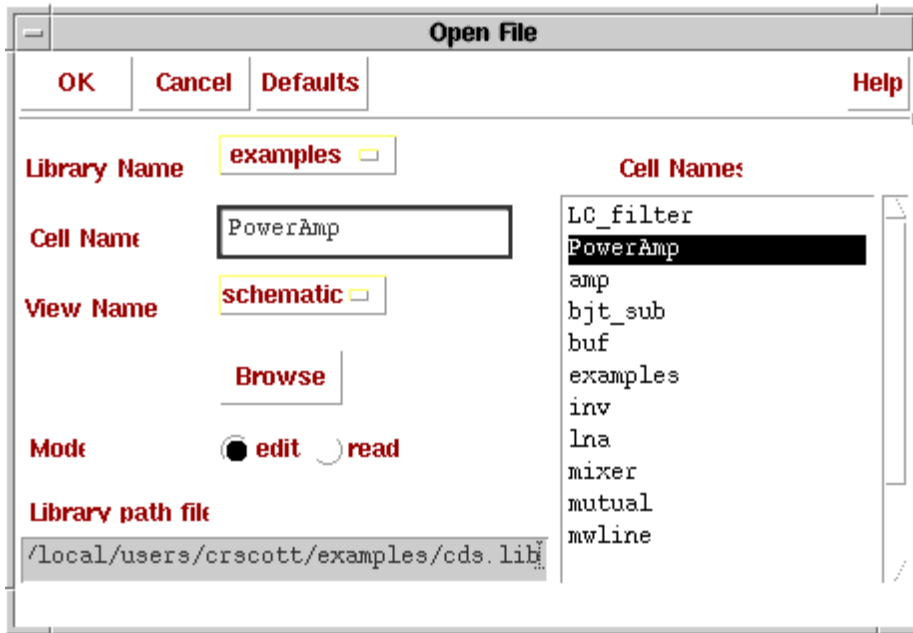
Cadence CIW Window

Note
 Non-standard or customized start-up scripts for Cadence Design Framework II may not be supported. If you have difficulties, contact your system administrator.

Opening a Cadence Composer Schematic

To open a schematic cellview in Cadence Composer:

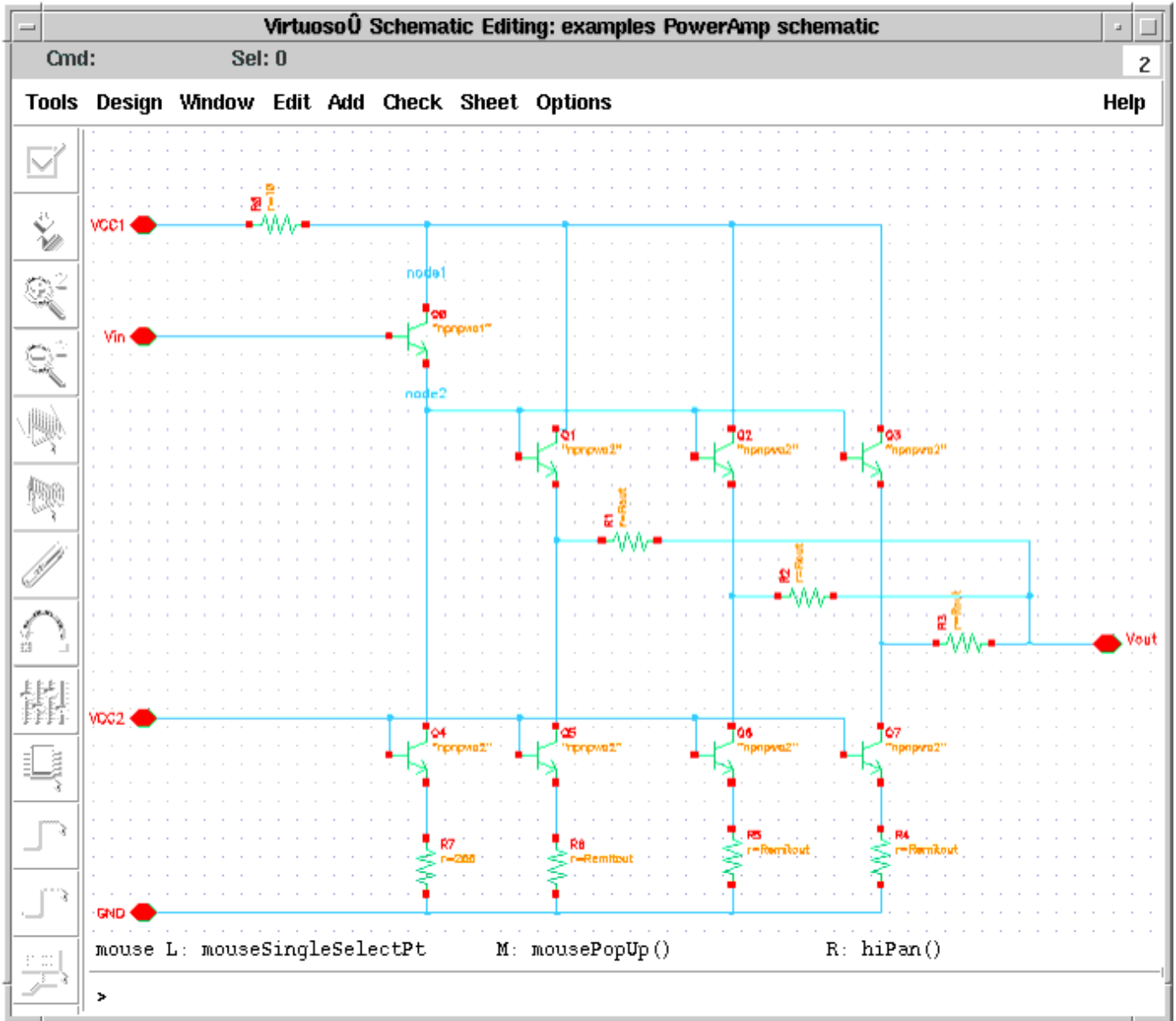
1. Choose File > Open from the Cadence CIW. The Cadence Open File form appears.



2. Select examples from the Library Name drop-down list.
3. Click PowerAmp in the Cell Names list. This sets the Cell Name field to PowerAmp.
4. Select schematic from the View Name drop-down list if not already selected.
5. Select the edit Mode if not already selected.

Note
To open a file in edit mode you must have write permission.

6. Click OK . The Cadence examples, PowerAmp schematic cellview appears.



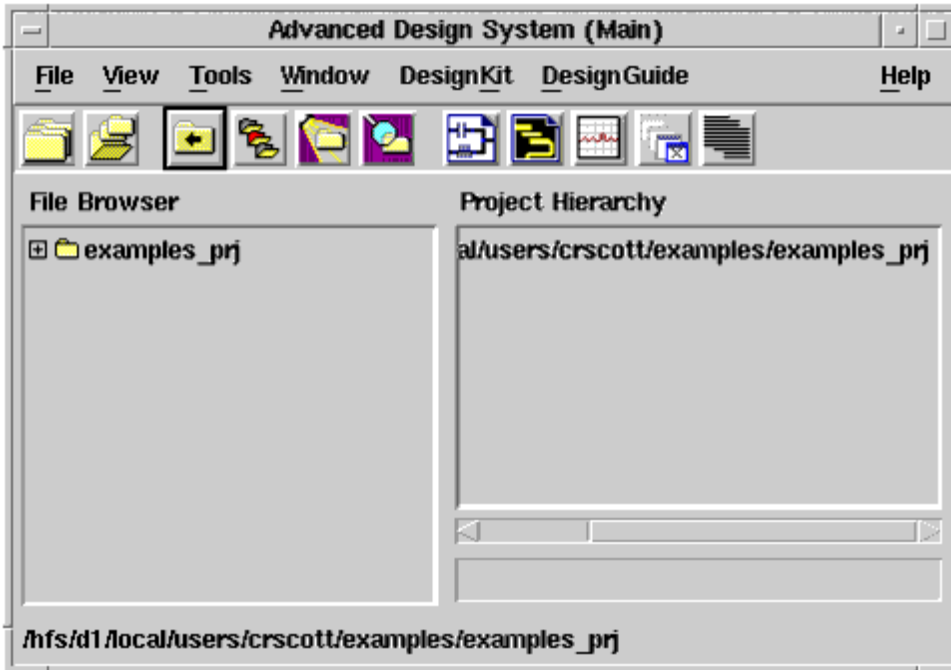
Cadence Virtuoso Schematic Composer Window

Linking with Advanced Design System

To link the Cadence design environment to Advanced Design System:

Advanced Design System 2008

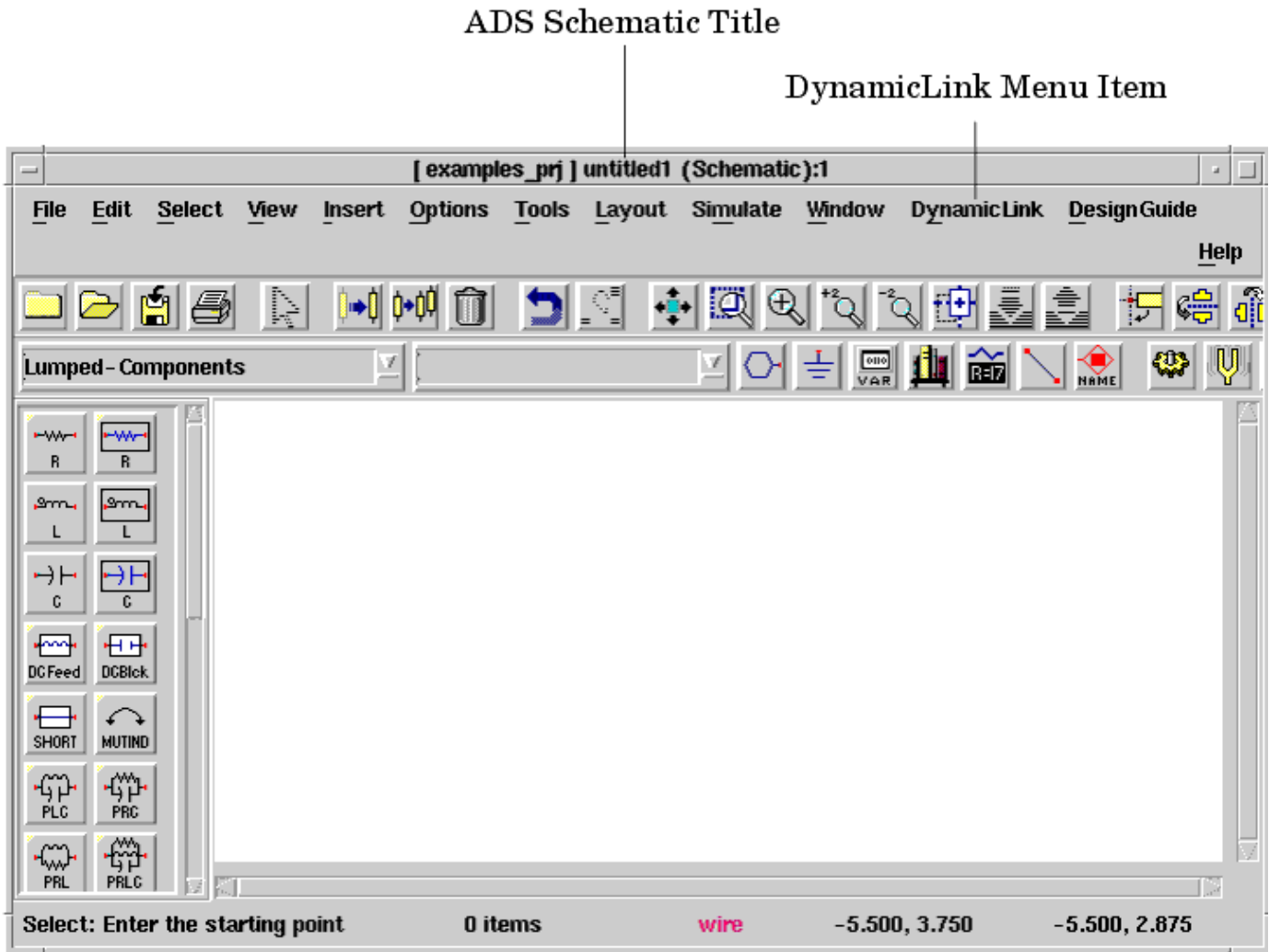
1. Choose Tools > ADS Dynamic Link from the menu bar in the Cadence Schematic window. In a few moments, the Advanced Design System Main window appears in the upper-left hand corner of your display.



Advanced Design System Main Window

Note
Depending on your system, it may take a few moments for the ADS windows to appear. View the Cadence CIW window for the link status.

2. From the ADS Main window, choose File > Open Project and select the examples_prj project. The ADS Schematic window displays a DynamicLink menu item and is automatically titled untitled1 (see the following figure).



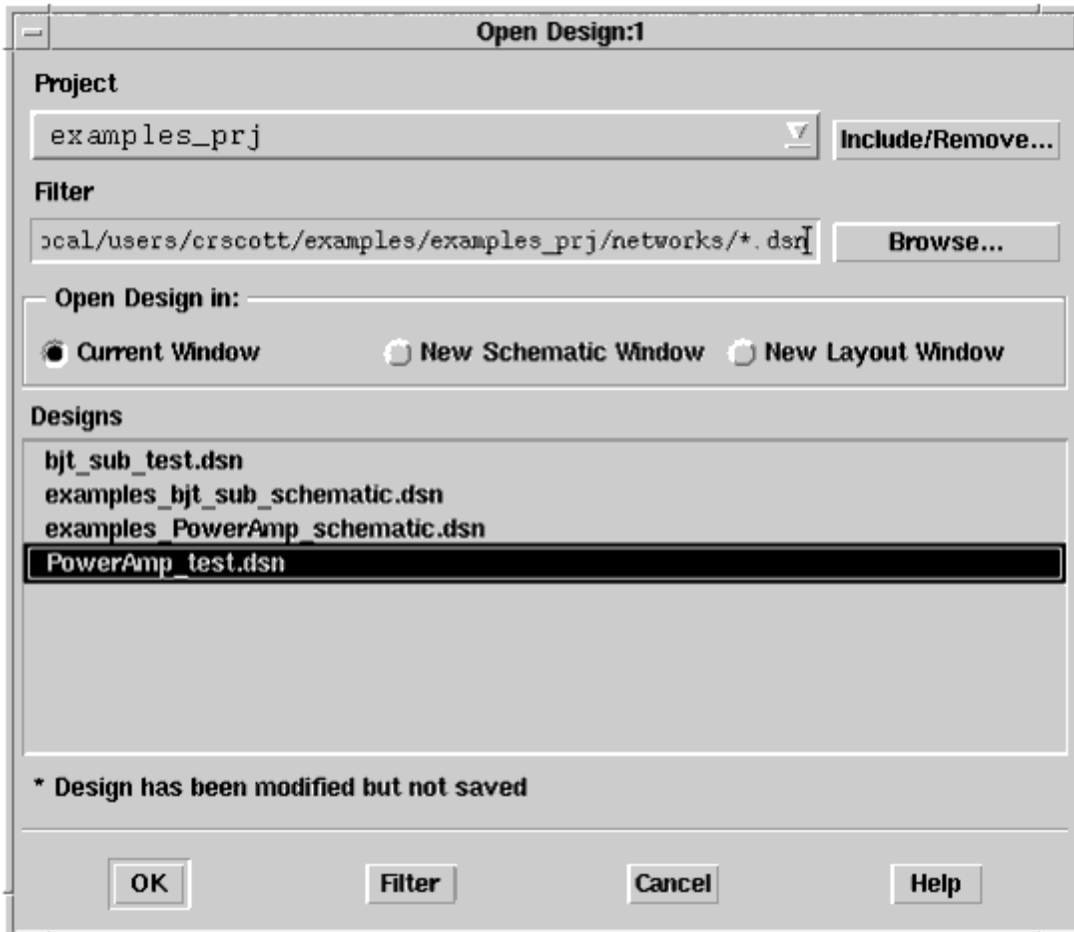
Advanced Design System Schematic Window

At this point, ADS and the Cadence Design Framework II are working together.

Opening a Test Schematic Design

To open a test schematic design:

1. Choose File > Open Design in the ADS Schematic window to display the Open Design dialog box. Use this dialog box to select the design you wish to simulate.

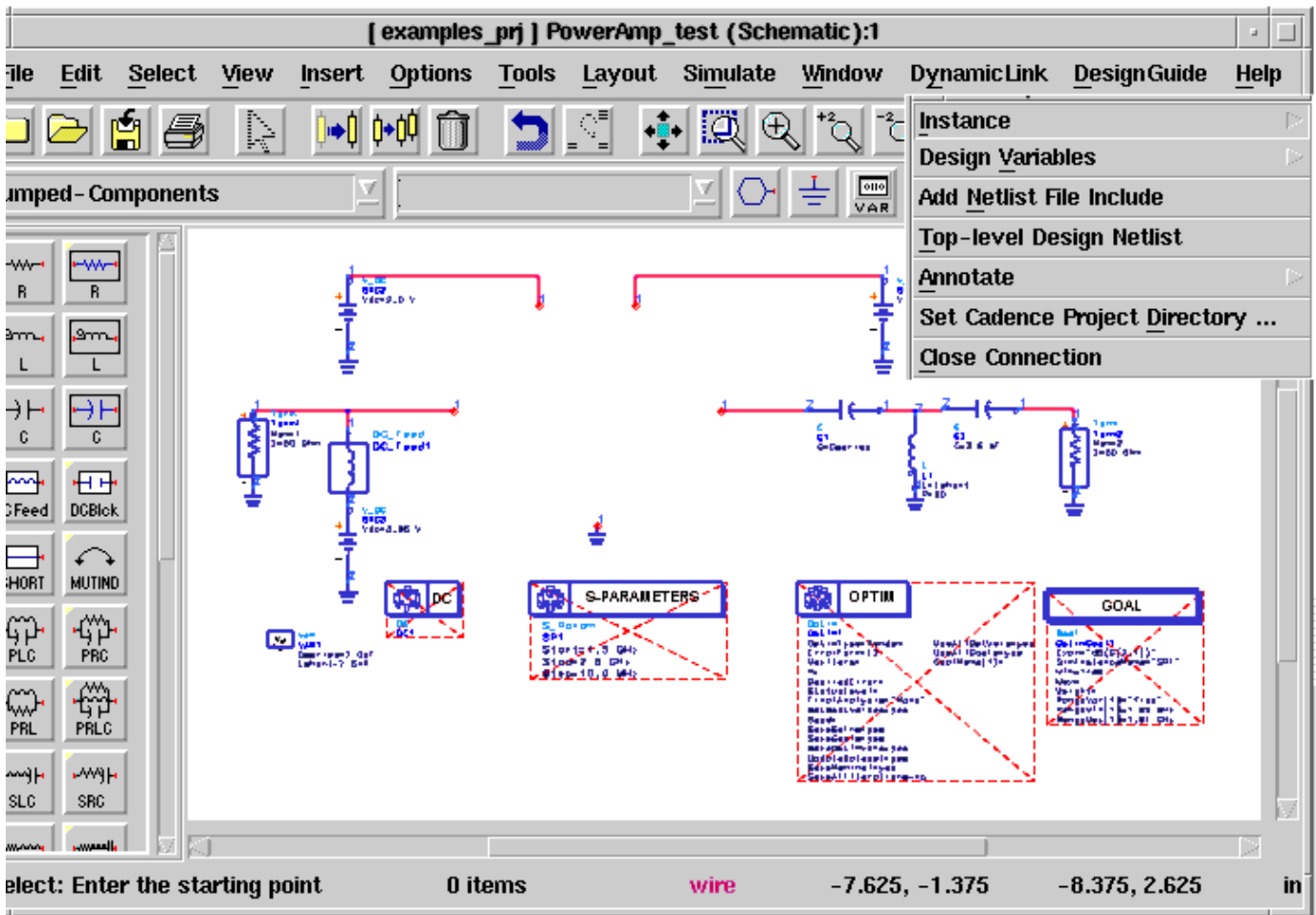


2. The examples_prj selection in the Project drop-down list is set by default in the Open Design dialog box.
3. Select PowerAmp_test.dsn from the Designs list. The PowerAmp_test.dsn schematic contains simulation components that can be selectively activated or deactivated.
4. Click OK to include the PowerAmp_test.dsn schematic in the ADS Schematic window.

Adding a Symbol of the Cadence Cellview

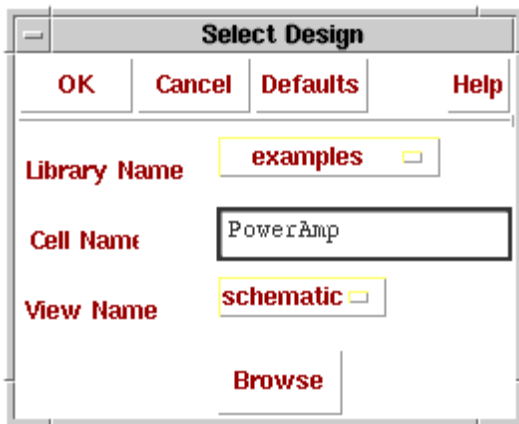
To add a symbol of the Cadence cellview in the Advanced Design System Schematic window:

1. Choose DynamicLink > Instance > Add Instance of Cellview in the ADS Schematic window.



Adding a Cellview

A Select Design dialog box appears, enabling you to select the Cadence Cellview to simulate.



2. In the Cell Name field of this dialog, verify the entry or type the name of the Cellview you want to simulate (in this case PowerAmp). Alternatively, you can use the Browse button and library manager to select the name.

3. Click OK . A symbol of your Cadence Cellview is automatically generated.
4. An instance of the symbol is attached to the cursor for you to place. In the Advanced Design System Schematic window, click the left mouse button to place the symbol as desired.
5. You may continue placing more instances of the same cellview, or, in this case, choose the Cancel Command And Return To Select Mode icon to proceed with the next step. Similarly, you may place instances of other Cadence designs.



Adding Design Variables

To add the design variables from the Cadence Cellview to the ADS schematic window choose DynamicLink > Design Variables > Get Design Variables .

This places a corresponding VAR component on the ADS schematic containing the design variables from Cadence (i.e. Rcc, Rout, Remitin and Remitout).



VAR

VAR1

Cseries=7.0pF

Lshunt=7.5nH

Rout=12

Rcc=10

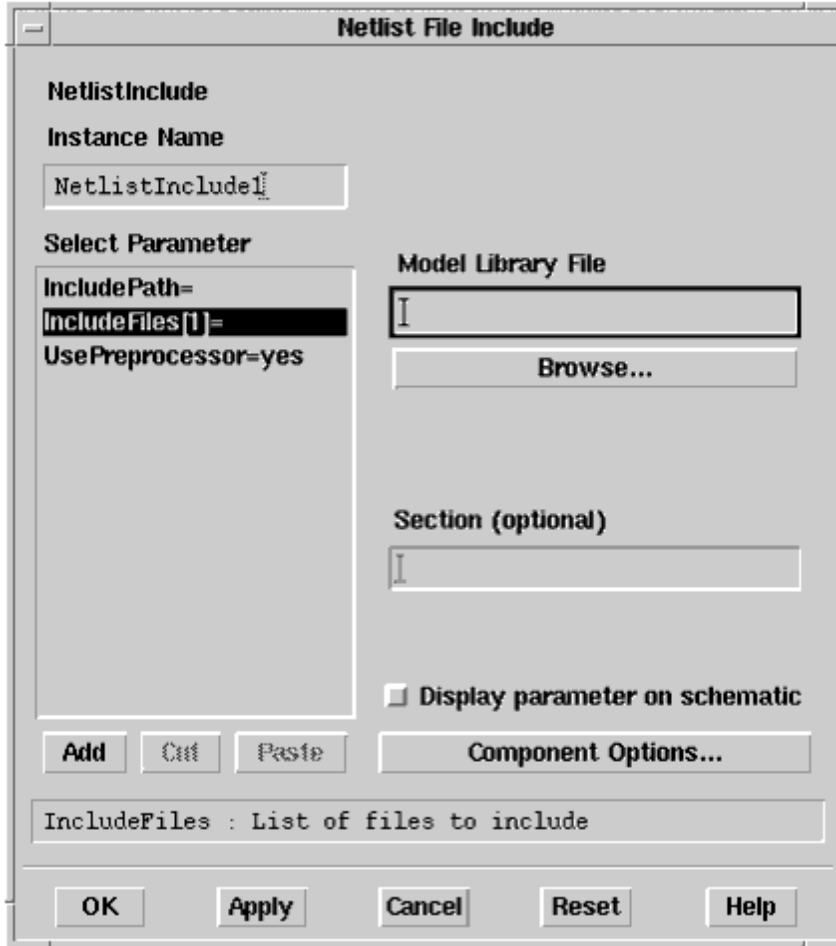
Remitin=200

Remitout=163

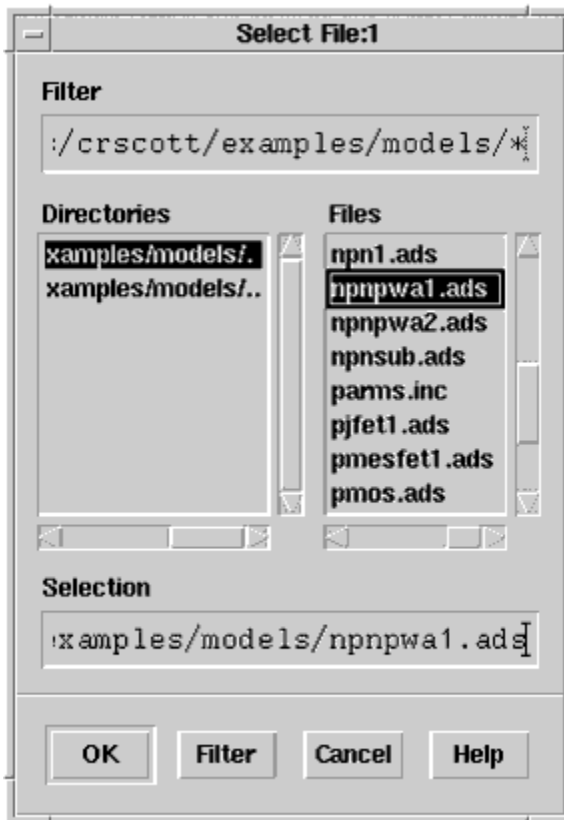
Adding Model Files

To add a model file

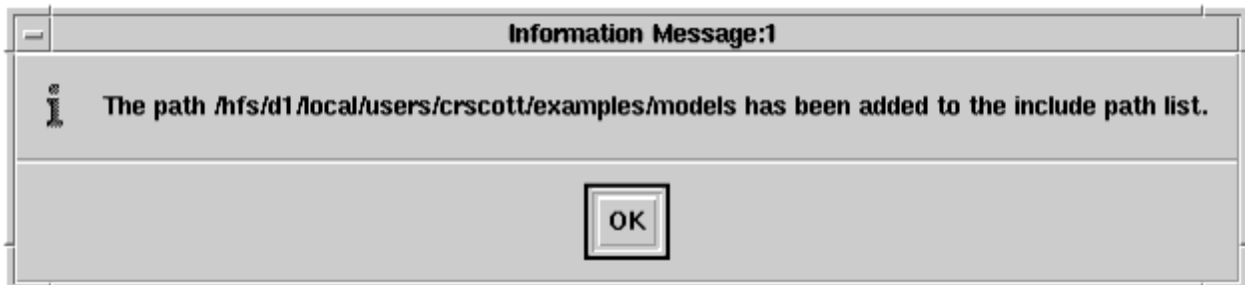
1. Choose DynamicLink > Add Netlist File Include
2. Place the NetlistInclude component in an open area on the schematic.
3. Double click the include component icon. The Netlist File Include dialog box appears.



4. In the Netlist File Include dialog box, click IncludeFiles in the Select Parameter list box. The Model Library File field appears in the dialog box.
5. Click Browse just below the Model Library File field to locate the first model library file. The Select File dialog box appears.



6. In the Select File dialog box, use the Directories field to locate the models directory.
`<your_current_working_dir> /examples/models`
 This sets the path for the location of the model library files.
7. In the Select File dialog box, use the Files field to locate and click the nnpwa1.ads model file, then click OK . An information message appears stating that a new path has been added to the include path list.



- Click OK in the Information Message dialog box. You are returned to the Netlist File Include dialog box.
8. In the Netlist File Include dialog box, notice that the Model Library File field now contains the nnpwa1.ads model file. Click Apply to add the nnpwa1.ads model file.
 9. Click Browse again to locate the second model library file. The Select File dialog box appears.
 10. In the Select File dialog box, use the Files field to locate and click the nnpwa2.ads model file, then click OK . You are returned to the Netlist File Include dialog box.
 11. In the Netlist File Include dialog box, notice that the Model Library File field now contains the nnpwa2.ads model file. Click Add to add the nnpwa2.ads model file.
 12. The Select Parameter field should now contain the information below.
`IncludeFiles[1]=nnpwa1.ads`

IncludeFiles[2]=nnpwa2.ads.

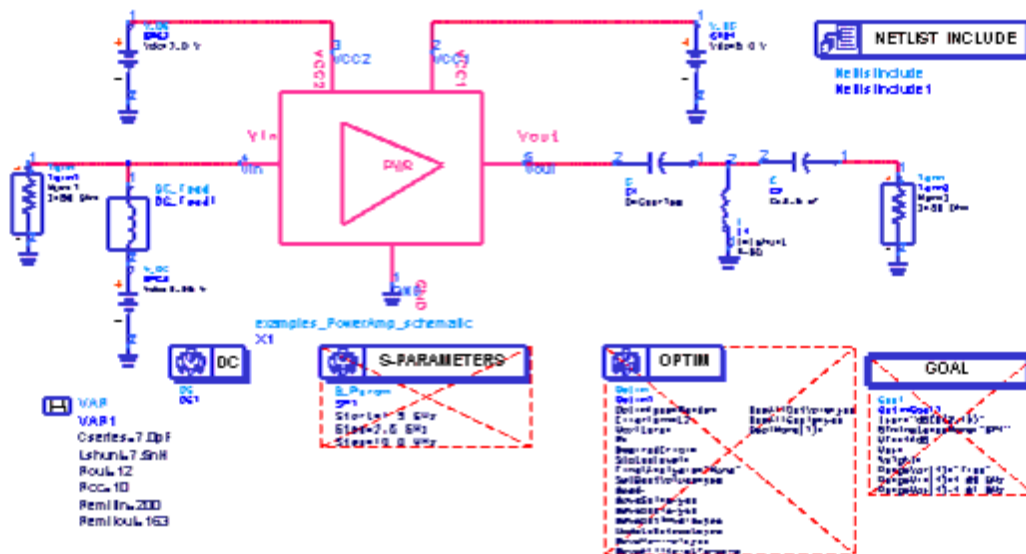
- Click OK in the Netlist File Include dialog box.

For more information on the Netlist File Include Component, refer to [Adding Model Files](#).

Performing a DC Simulation

To run a DC simulation on an ADS schematic and then annotate the results to the Cadence Composer Window:

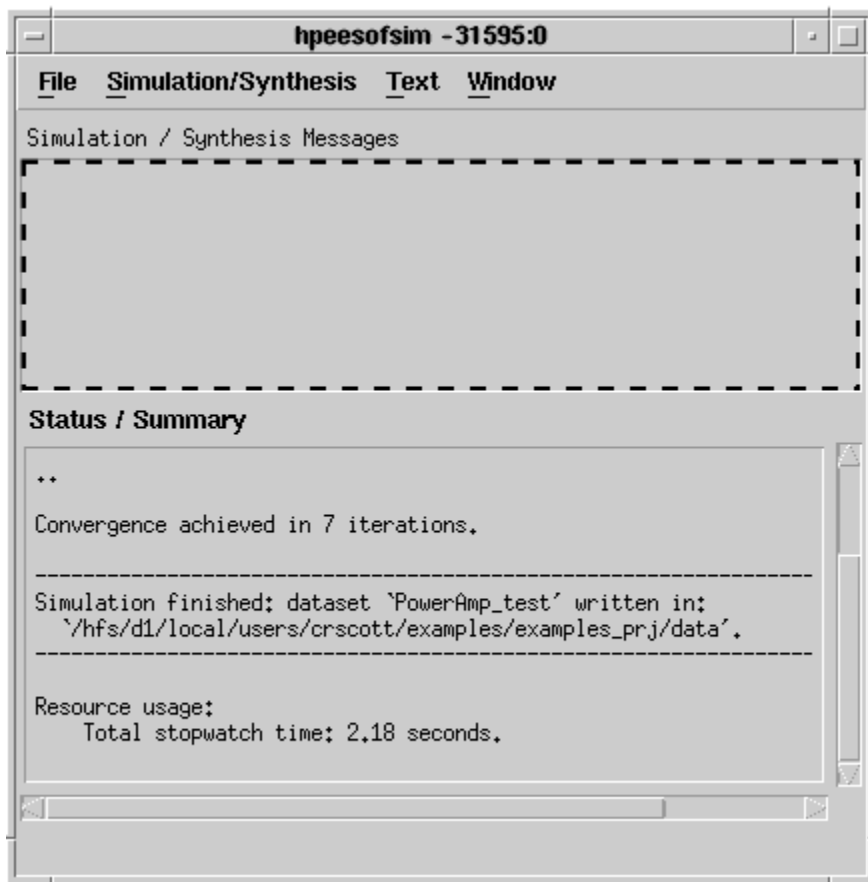
- Choose Edit > Component > Deactivate/Activate then click the DC component in the ADS Schematic window to toggle and activate the component. Alternatively, you can choose the Deactivate/Activate Components icon to activate the DC component.



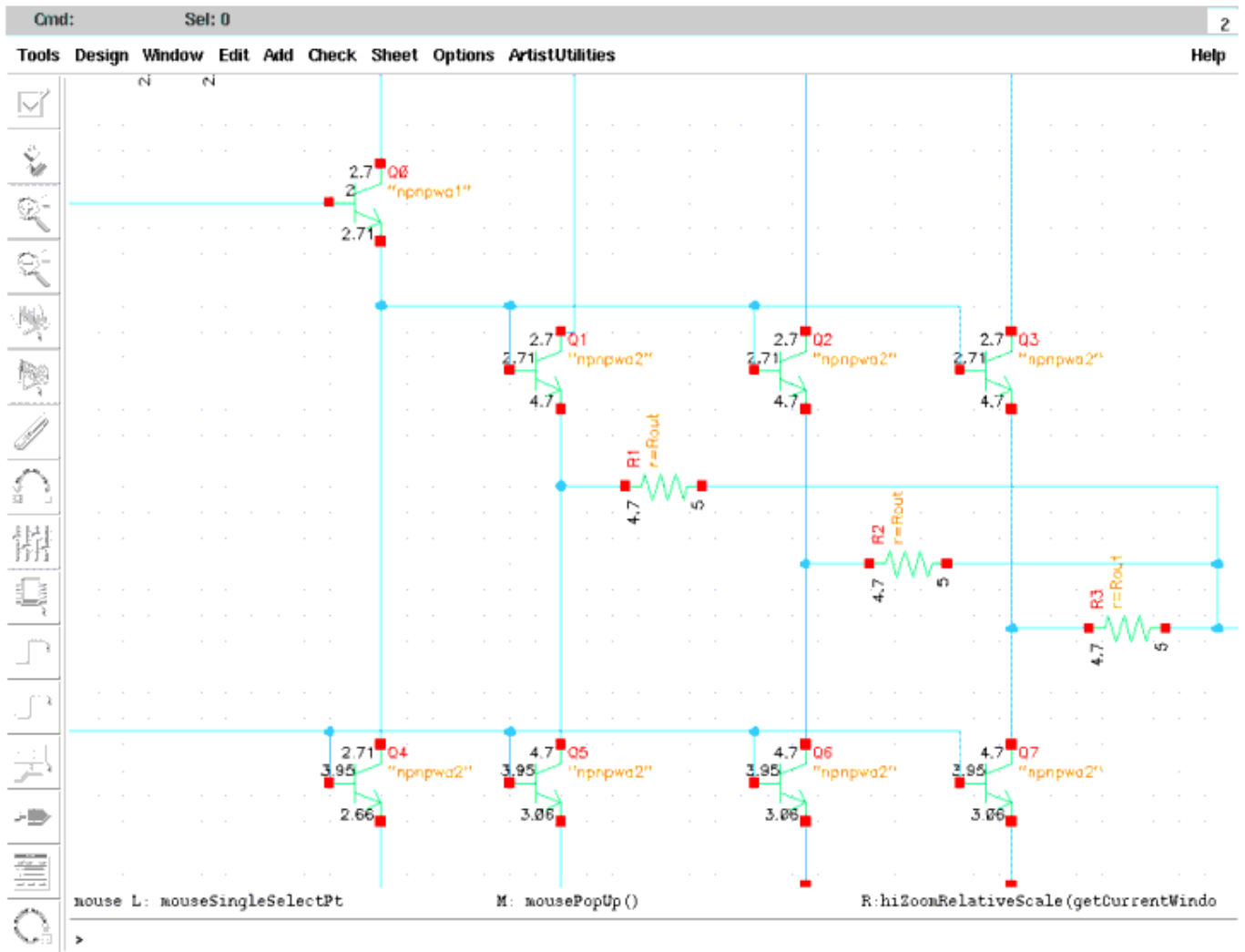
Activating Components in an ADS Schematic Window

- Choose Simulate > Simulate or choose the Simulate icon to run a simulation. A simulation dialog box appears in your display.





3. After the simulation is complete, a Data Display window titled PowerAmp_test automatically appears. Close this window using the File > Close Window menu option.
4. Click the PowerAmp schematic symbol in the ADS Schematic window.
5. Choose DynamicLink > Annotate > Annotate DC Solution to Selected Cellview . This displays the DC node voltages on the Cadence schematic.



Cadence Schematic with Annotated Display

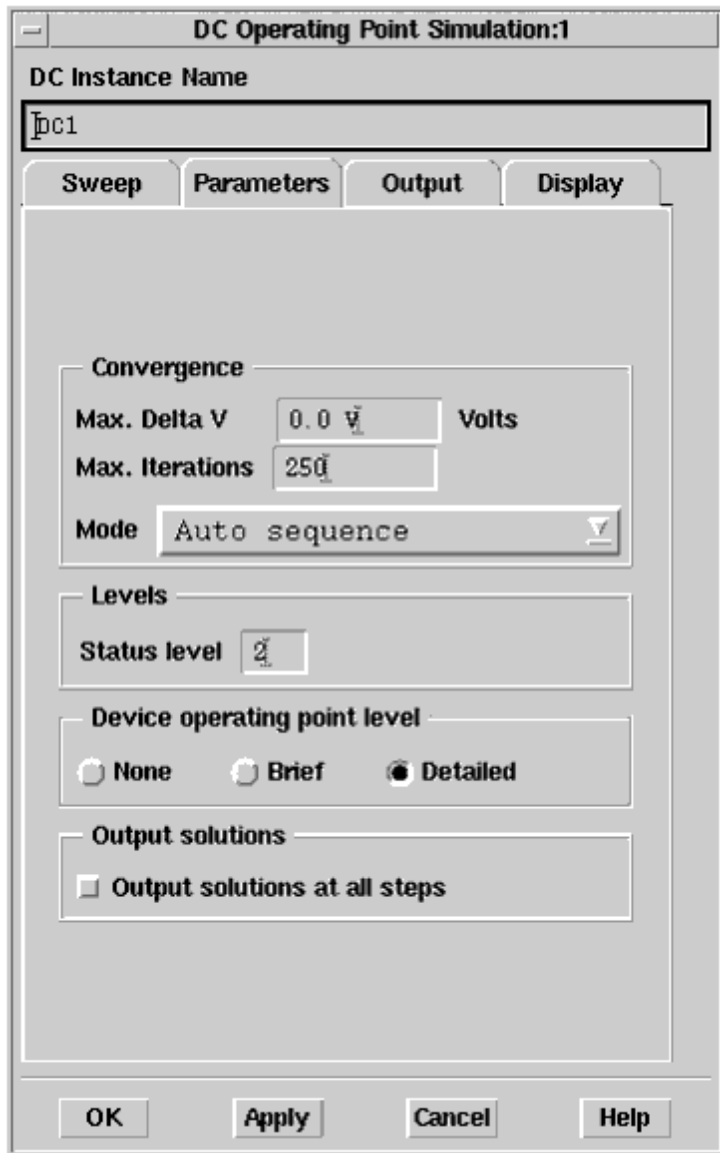
Annotating DC Operating Points to a Selected Cellview

Any simulation that includes a DC analysis produces DC operating point information for most active and some passive devices in the circuit. This data includes currents, power, voltages, and linearized device parameters of the selected device.

To run a DC Operating Point Simulation on an ADS schematic and then annotate the results to the Cadence Composer Window:


1. Double-click the DC Simulation component in the ADS schematic window. The DC Operating Point Simulation

dialog box appears.

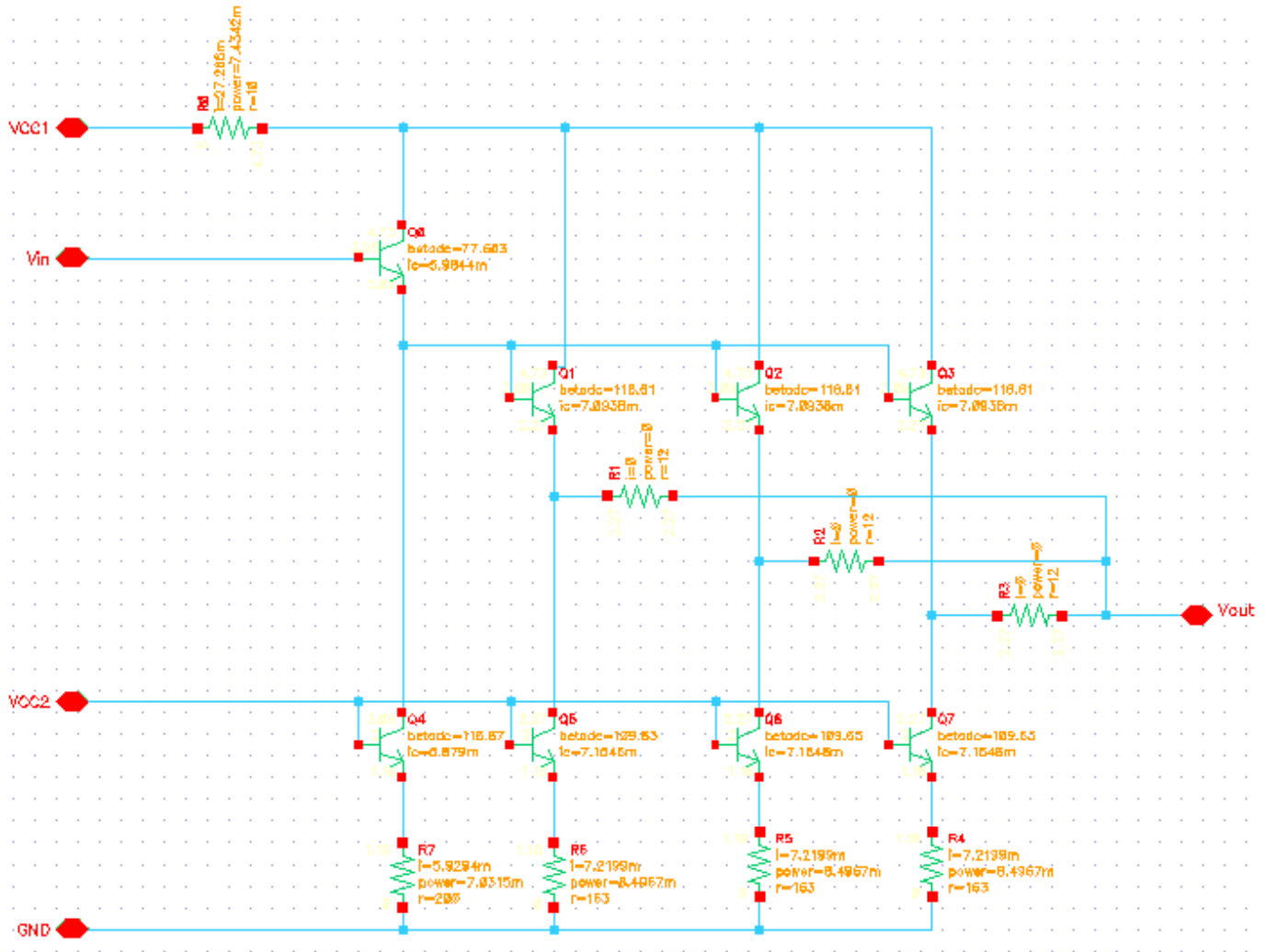


2. Click the Parameters tab in the DC Operating Point Simulation dialog box.
3. Click Detailed in the Device operating point level section of the DC Operating Point

Note
For a subset of the detailed DC Operating Point Simulation information that covers most common parameters, click Brief in the Device operating point level section of the DC Operating Point Simulation dialog box.

4. Choose Simulate > Simulate or choose the Simulate icon to run a simulation. A simulation dialog box appears in your display. 
5. After the simulation is complete, a Data Display window titled PowerAmp_test automatically appears. Close this window using the File > Close Window menu option.
6. Click the PowerAmp schematic symbol in the ADS Schematic window.
7. Choose DynamicLink > Annotate > Annotate Operating Points to Selected Cellview . This displays the DC

operating point values for each component on the Cadence schematic.



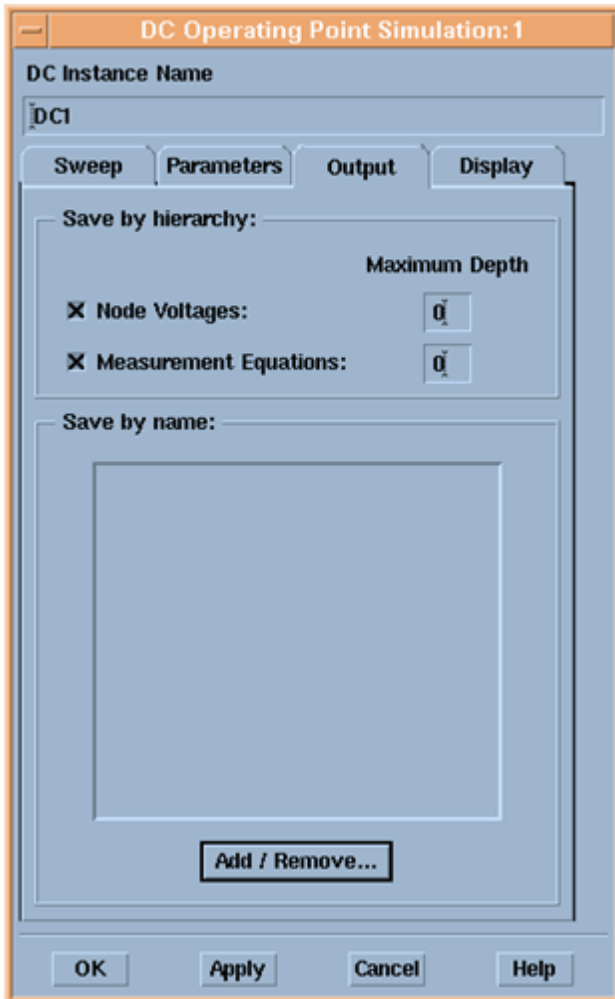
Annotated Cellview with DC Operating Points

Saving Node Voltages and Pin Currents with RFIC Dynamic Link

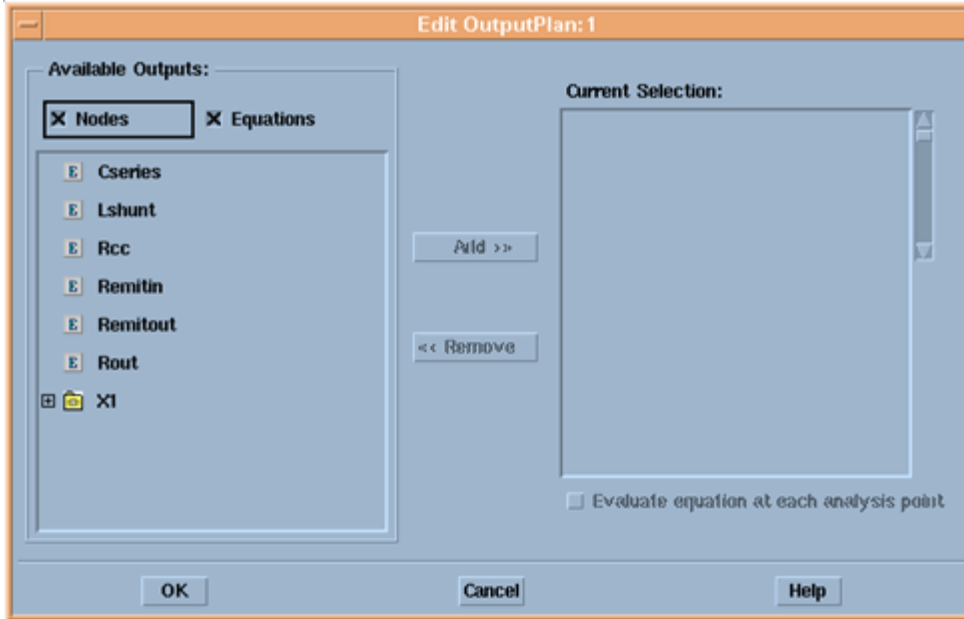
With the RFIC Dynamic Link Probing feature, you can select nodes or pins of interest in a Cadence subcircuit and display their voltages or currents in an ADS Data Display window.

The following is an example session using the Dynamic Link Probing feature with the PowerAmp example.

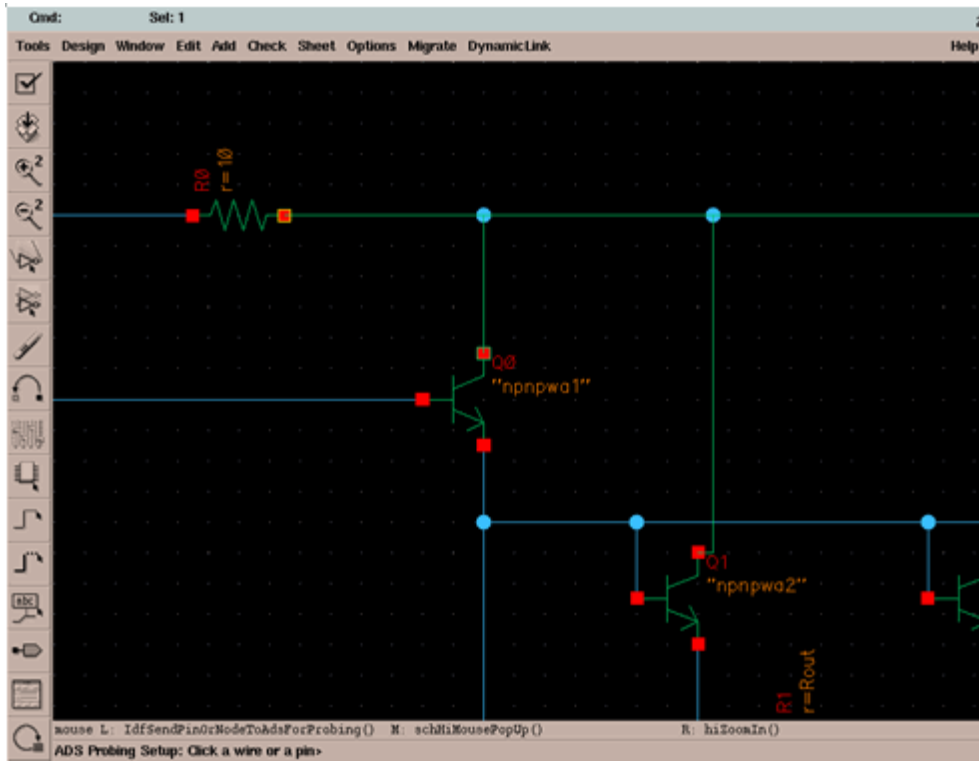
1. Double click the DC Simulation controller component in the ADS schematic window to open the DC Operating Point Simulation dialog box.



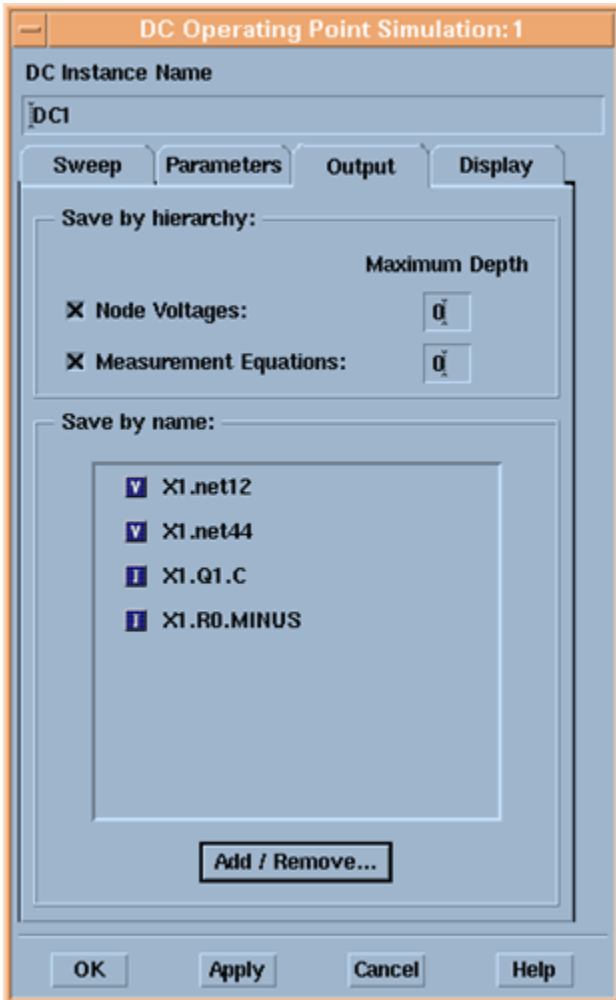
2. Click the Output tab in the DC Operating Point Simulation dialog box.
3. Click Add/Remove in the Output section of the DC Operating Point Simulation dialog box to open the Edit OutputPlndialog box.





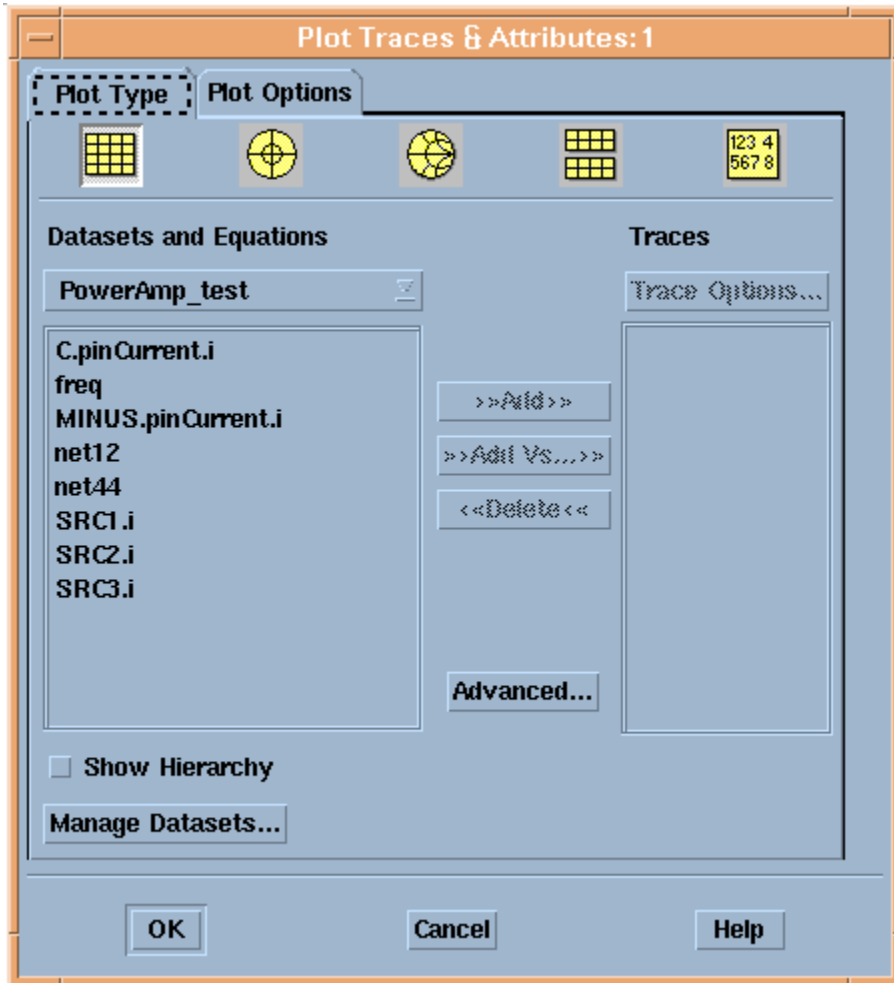
4. Click the + sign to the left of the X1 component in the Available Outputsfield to start Dynamic Link Node Probing setup. This raises the Cadence schematic window containing the PowerAmp cellview. The prompt in the Cadence schematic window is changed to: ADS Probing Setup: Click a wire or a pin>.
5. Click the right pin of the R0 resistor on the upper-left region of the schematic window. "X1.R0.MINUS.pinCurrent.i" appears in the Current Selection field on the right side of the ADS Edit OutputPlan dialog box.
6. Click the collector pin at the top of the Q1 BJT. "X1.Q1.C.pinCurrent.i" appears in the Current Selection field on the right side of the ADS Edit OutputPlan dialog box.
7. Click the vertical net connected from above Q1 to the collector pin of the Q1 BJT. "X1.net44" appears in the Current Selection field on the right side of the ADS Edit OutputPlan dialog box.
8. Click the vertical net connected from below Q1 to the emitter pin of the Q1 BJT. "X1.net12" appears in the Current Selection field on the right side of the ADS Edit OutputPlan dialog box.



9. Click OK in the ADS Edit OutputPlan dialog box to close the dialog box. The DC Operating Point Simulation dialog box should now appear similar to the following figure.



10. Click OK in the DC Operating Point Simulation dialog box to instruct ADS to save currents for X1.R0.MINUS and X1.Q1.C in addition to voltage values for X1.net12 and X2.net44 after DC simulation.
11. Click the Simulate icon or select Simulate > Simulate menu item in the ADS schematic window to run a DC Simulation. The ADS Data Display window appears automatically once the ADS DC Simulation is complete. 
12. Click the List icon on the left of the Data Display window, then click a desired location in the ADS Data Display window to bring up the Plot Traces & Attributes dialog box. 



13. In the Plot Traces & Attributes dialog box, press the Ctrl key and click the following four items on the left, then click >>Add>> in the middle to add these four items to the Traces field on the right.

C.pinCurrent.i
 MINUS.pinCurrent.i
 net12
 net44

14. Click OK to close the Plot Traces & Attributes dialog box. The currents of R0.MINUS and Q1.C pins and voltages of net12 and net44 are immediately listed in the ADS Data Display window.


Additional Notes:

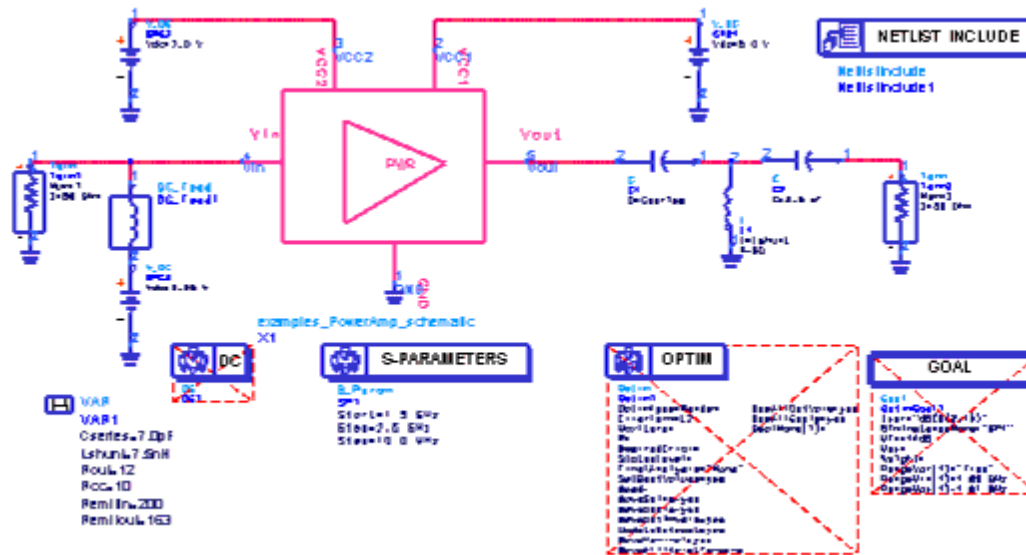
- When the Add/Remove button in the DC Operating Point Simulation dialog box is clicked to bring up the Edit OutputPlan dialog box, as described in step 6 above, all of the named nodes and subcircuit components at current circuit level appear in the Available Outputs field on the left. You can select a named node from the Available Outputs field and then click Add>> to add it to the Current Selection field on the right. You can also remove a named node from the Current Selection field by clicking <<Remove after selecting it.

- Clicking the + sign to the left of a subcircuit component in the Available Outputs field displays all the named nodes and subcircuit components within that subcircuit. In step 7 above, nothing was displayed initially because there was no named node or subcircuit in the PowerAmp cellview.
- After a Cadence schematic window is raised as a result of clicking the + sign mentioned above, any named or unnamed node in the Cadence subcircuit hierarchy can be selected by clicking the left mouse button at any point on the wire. To select a node in Cadence subcircuit, choose Design > Hierarchy > Descend Edit menu item in the Cadence schematic window, click OK in the form popped up, and then click a wire of interest. Valid selections are limited to the current window and the current circuit hierarchy.
- During the Node Probing operation, the left mouse button in the Cadence Virtuoso schematic window is mapped to a Dynamic Link SKILL procedure. Do not bind any function to the left mouse button during this period. Any bindkey function previously mapped to the left mouse button will not work until the tune mode ends.


Performing an S-parameter Simulation

To perform an S-parameter Simulation on an ADS schematic:

1. Deactivate the DC component using the Deactivate/Activate toggle. 
2. Activate the S-parameter component using the Deactivate/Activate toggle. The ADS schematic is now ready to simulate an S-parameter.



Activating Components in an ADS Schematic Window

3. Choose Simulate > Simulate or choose the Simulate icon to run the S-parameter simulation. 


After the simulation is complete, a Data Display window titled PowerAmp_test automatically appears.

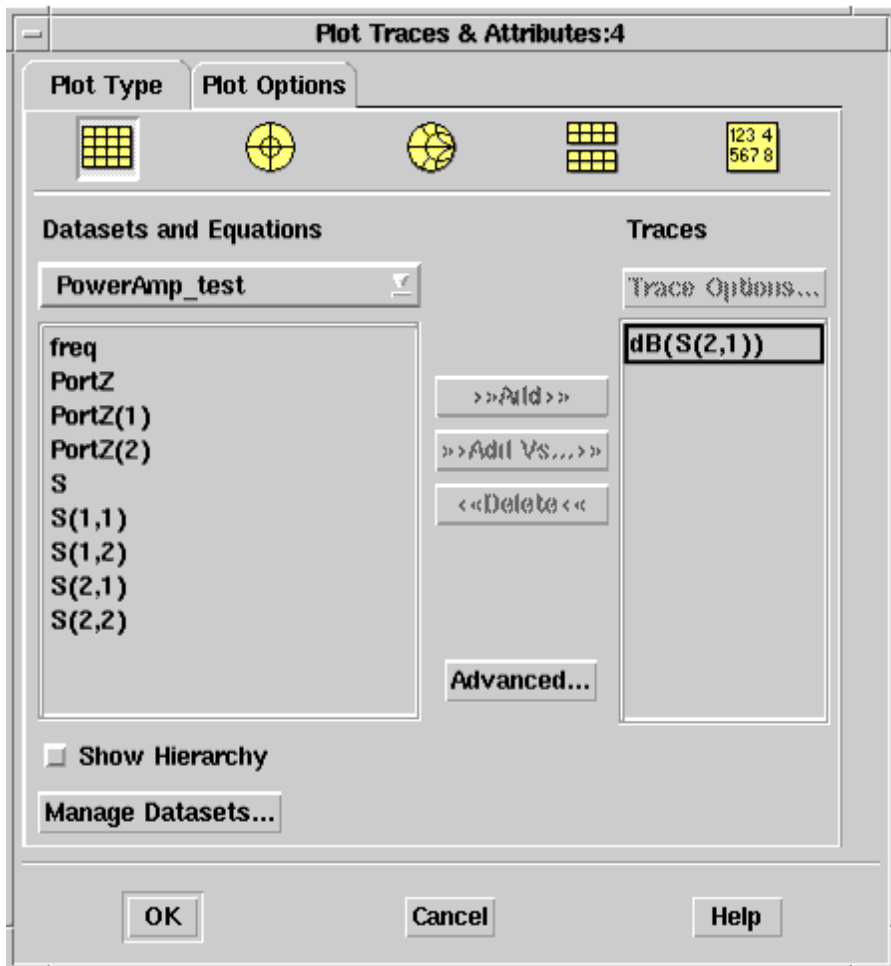
Note
The amplifier used in this tutorial is not a traditional microwave amplifier for use by itself. It is meant to be

used as part of an RFIC, where it takes a small voltage signal and converts it into a current for driving a larger-signal amplifier, off chip. It has not been designed to have S 11 and S 22 less than 1. More simulations are run on the amplifier in the ADS example file, IQ_Mod_from_MDS_prj.zap, which you can download from the Agilent EEs of EDA external Web site. The example described in this document is only illustrating the RFIC Dynamic Link for Cadence.

Displaying Your Results

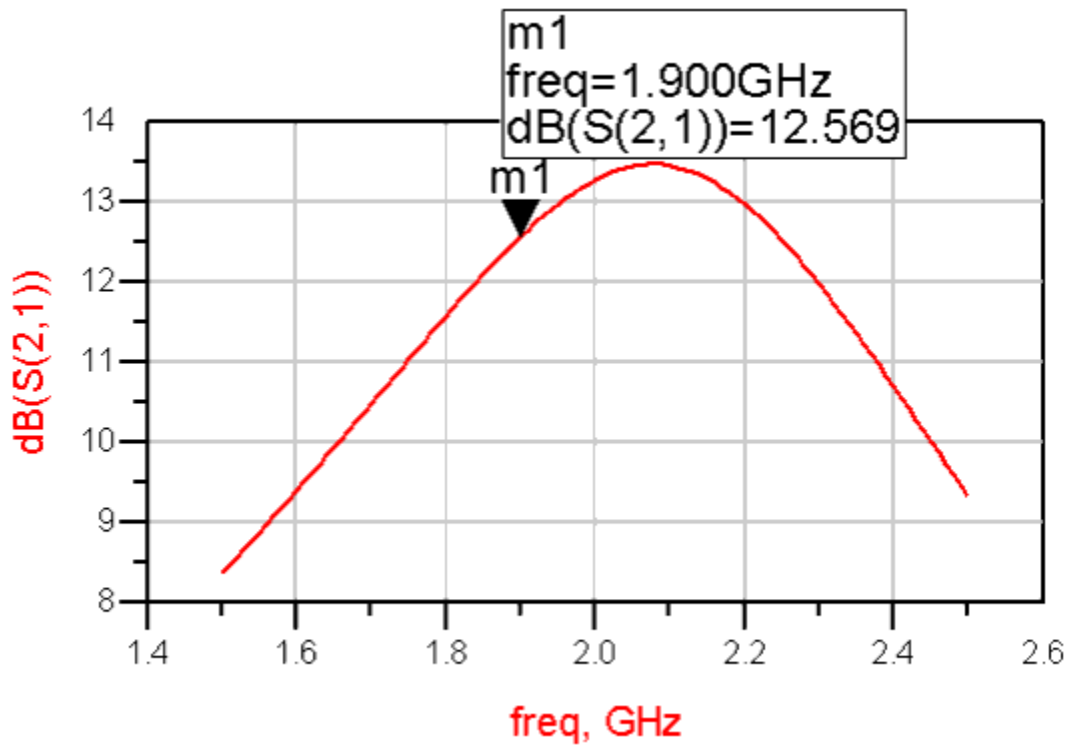
To view the results of your simulation in a plotted Data Display:

1. Select PowerAmp_test from the Default Dataset ___ drop-down list if not already selected.
2. From the Data Display window, choose Insert > Plot , move the frame to an appropriate location within the window and click. This anchors a frame for your plot. Similarly, you can choose the Rectangular Plot icon to drag and drop the plot frame. The icon is a small square with a grid pattern, representing a rectangular plot.
3. The Plot Traces & Attributes dialog box appears. Select S(2,1) and then click Add . Select dB from the dialog box then click OK .



4. Click OK again to view the Data Display .

Note
The following figure shows the forward gain, $S(2,1)$, at 1.9 GHz to be approximately 12.57 dB. By varying the value of Remitout in this case, you can modify the circuit to achieve the desired results. After you have completed the tutorial, take some time to experiment with different values of Remitout .



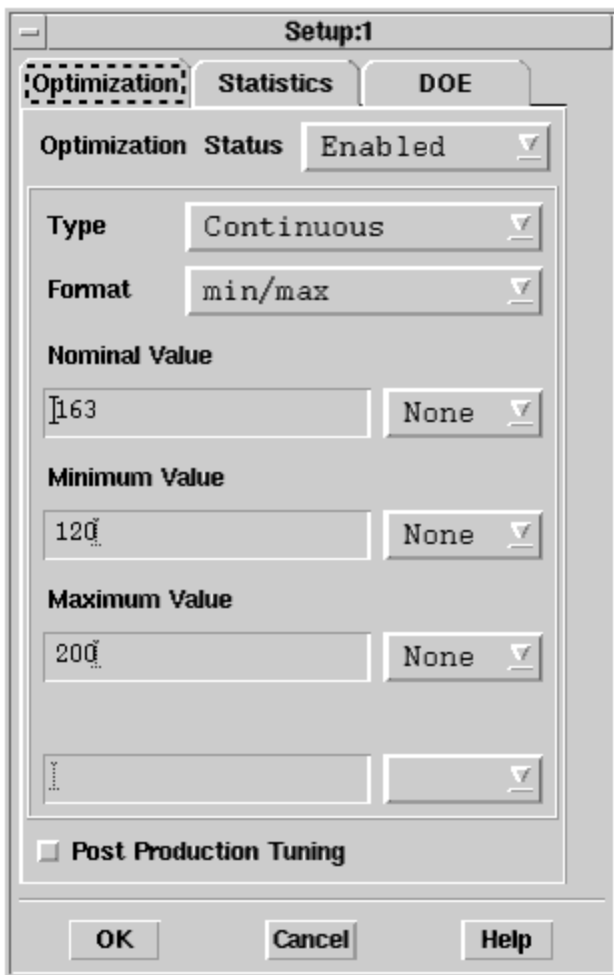
Data Display with S-parameter Simulation Results

Performing a Parameter Optimization

To optimize the parameters of Var1 in the ADS schematic:

1. Choose the Deactivate/Activate Components icon then place the cross hair over the Nominal Optimization component to toggle and activate the component. Repeat this step for the Goal component. The ADS schematic is now ready to optimize the S-parameter.





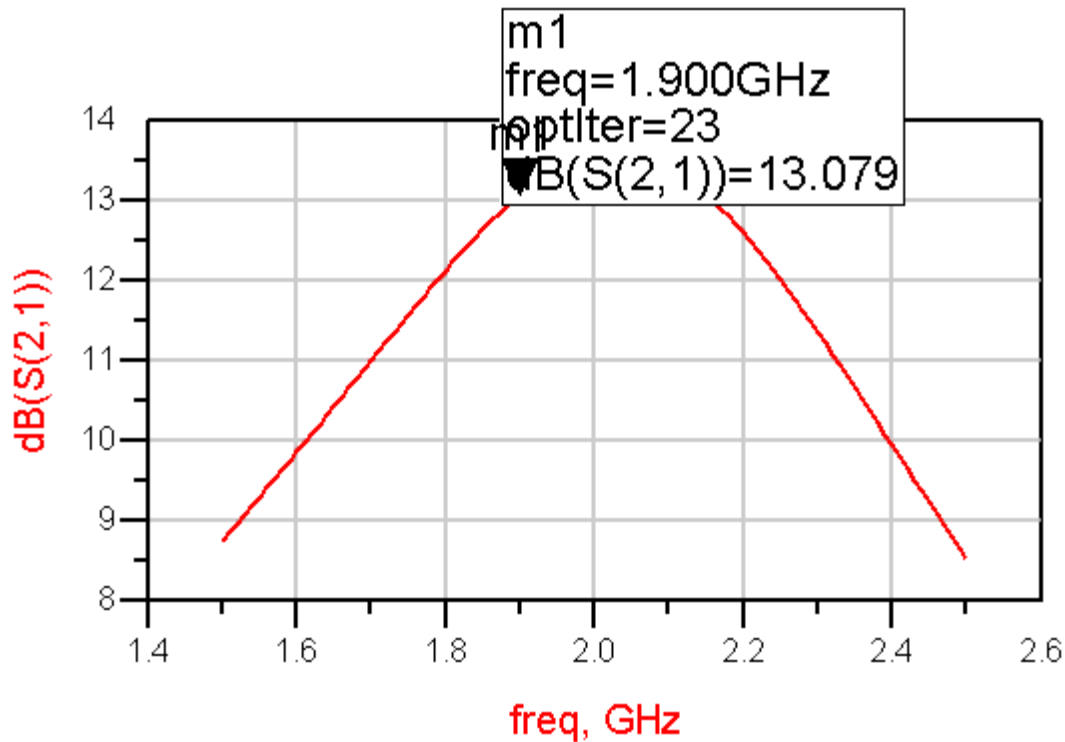
7. Click OK twice, once in the Setup dialog box and once in the Variables and equations dialog box. Note that the Remitout parameter on the ADS schematic now displays:
Remitout= 163 opt{120 to 200}
8. Choose Simulate > Simulate or choose the Simulate icon. This netlists each Cadence subcircuit in the Affirma Analog Circuit Design Environment, as well as the top-level ADS schematic, and starts the Advanced Design System simulator.



Note
A simulation status window appears, reporting the status of the simulation; depending on your system, this may take some time. Check this status window to see if any errors occurred during netlisting or simulation.

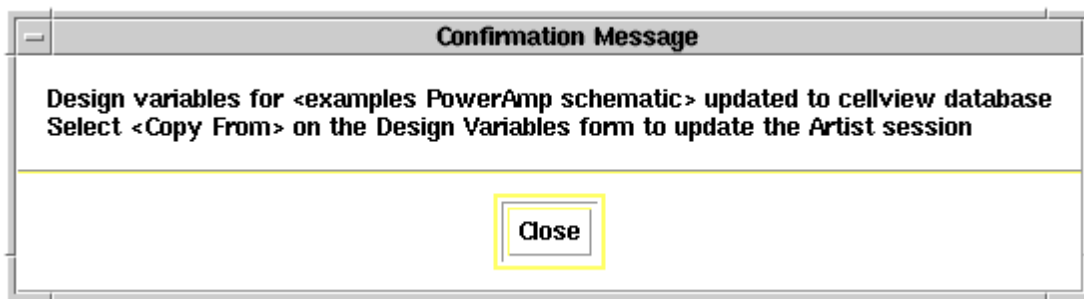
After the simulation is complete, a Data Display window titled PowerAmp_test automatically appears. For information on configuring the Data Display, refer to [Displaying Your Results](#).

Note
The following figure shows the optimized forward gain, $S(2,1)$, at 1.9 GHz to be approximately 13.08dB as set by the Goal component in the ADS Schematic window.



Data Display with Optimized Simulation Results

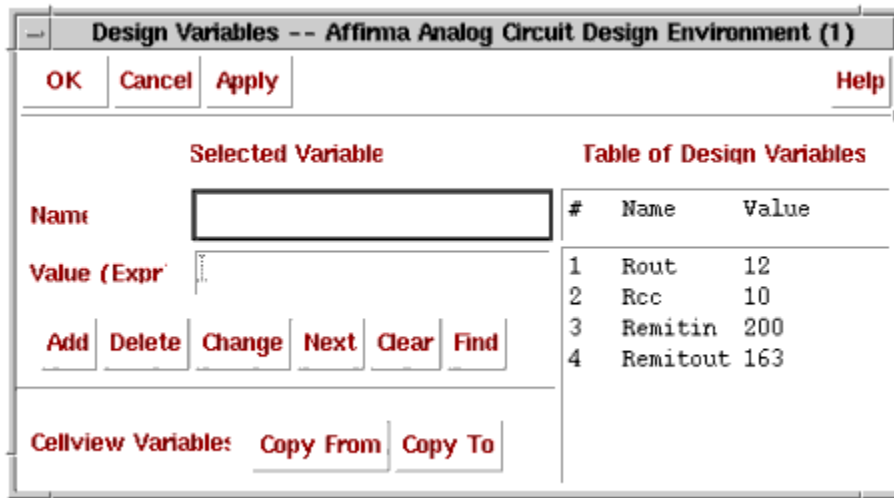
9. Choose Simulate > Update Optimized Values to update the optimized values. This changes the value of Remitout in the VAR component to the optimized value.
10. Choose DynamicLink > Design Variables > Update Design Variables to Cellviews to update the optimized value to the Cadence cellview. A Confirmation Message dialog box appears.



11. Click the Close button.
12. From the Cadence menu bar, choose DynamicLink > Design Variables . A Design Variables form appears.

Note
 If the DynamicLink pull-down menu does not appear in the Cadence Virtuoso Schematic window, choose Tools > ADS Dynamic Link in the Cadence schematic window.

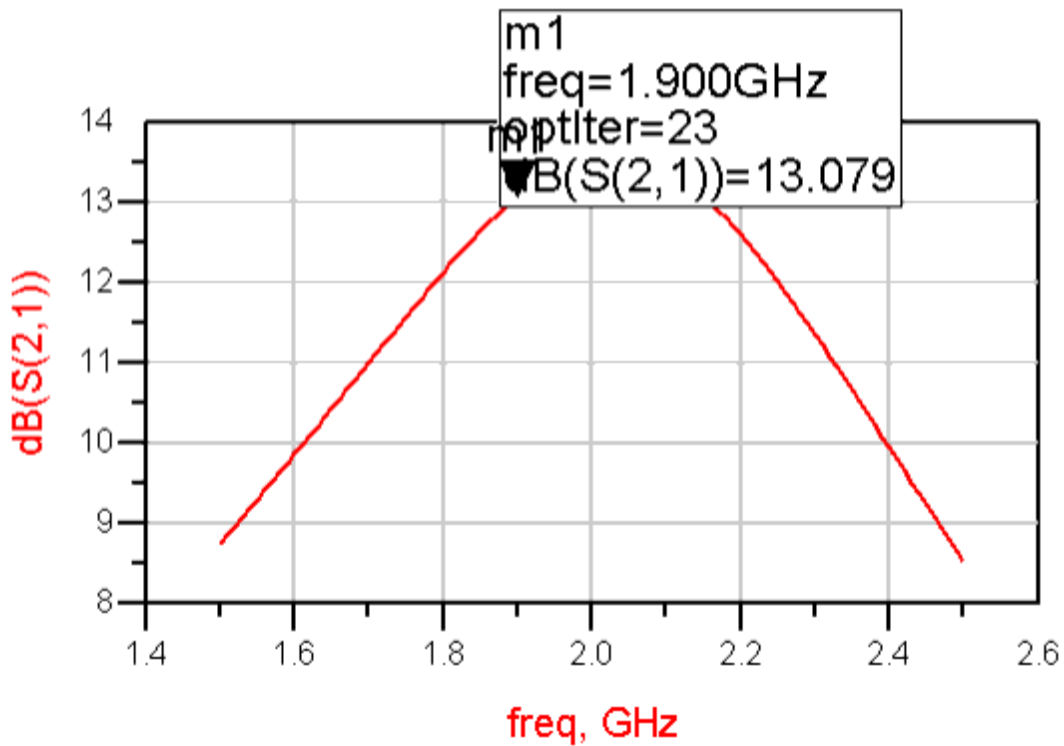
13. Click Copy From in the Cellview Variables section. This enables you to update the design variables to the Artist session.



Verifying Your Results

You may now verify the results of your optimization by using the optimized value of Remitout in an S-parameter simulation. Set the value of Remitout = 120 (approximate optimum value) and repeat the steps in [Performing an S-parameter Simulation](#) with the new value of Remitout.

Note
The following figure again shows the optimized forward gain, $S(2,1)$, at 1.9 GHz to be approximately 13.08 dB as set by the Goal component in the ADS Schematic window. By using the optimized value of Remitout, you have verified the optimum desired results of the circuit.



Data Display with Optimized Simulation Results

Ending the Session

Use the following steps to exit the Dynamic Link environment and close both ADS and Cadence.

1. Choose the Advanced Design System Schematic menu option DynamicLink > Close Cadence Connection . This closes ADS and terminates the link between Cadence and the Advanced Design System.
2. Exit from the Cadence CIW.
3. Congratulations... you have now successfully completed the Tutorial.

Using RFIC Dynamic Link

This chapter describes the procedures for:

- [Launching Advanced Design System](#)
- [Adding an Instance of a Cadence Design](#)
- [Pushing into the Design Hierarchy](#)
- [Using Design Variables](#)
- [Adding Model Files](#)
- [Annotating a Cellview](#)

Launching Advanced Design System

To run Advanced Design System from the Cadence Schematic window using RFIC Dynamic Link:

1. In the Cadence Schematic window, open the desired cellview.
2. Choose Tools > ADS Dynamic Link from the Cadence Schematic window. The Advanced Design System Main window appears in the upper left corner of your screen followed, to the right, by an empty ADS Schematic window (this may take some time).

The Cadence schematic window displays a DynamicLink pull-down menu. This menu provides some familiar, useful Affirma Analog Circuit Design Environment interface functionality. For further information about these options, consult your Cadence documentation.

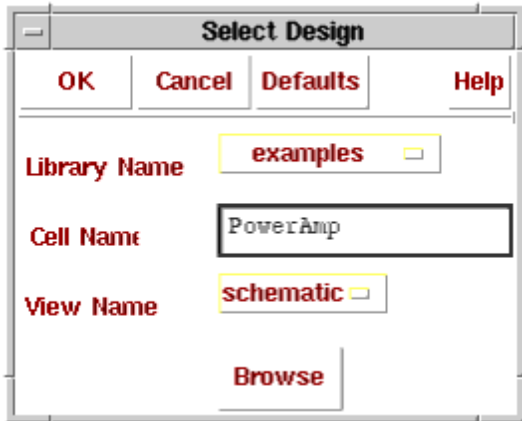


Note
The terminal output (stderr) of ADS gets redirected to the file idf.log in the directory in which the Cadence framework is started. Once the link is started, subsequently opening a Cadence design will not involve the overhead of re-starting ADS, but you will need to select Tools > ADS Dynamic Link just to see the DynamicLink pull-down menu.

Adding an Instance of a Cadence Design

To add an instance of a Cadence design to an ADS test schematic, choose DynamicLink > Instance > Add Instance of Cellview .

A dialog box appears, allowing the selection of a Cadence design.



If a symbol already exists for the design in Cadence, the symbol geometry is duplicated in ADS; otherwise the Cadence symbol generator is automatically invoked to generate a Cadence symbol, which is then automatically duplicated in ADS.

If Cancel is selected, or if the configuration file has the entry,

```
IDF_CADENCE_SYMBOL = FALSE
```

the ADS symbol generation is automatically invoked (as opposed to the Cadence symbol generation). This generates the symbol in ADS.

Note
The generated symbol can be edited and modified if needed. If aesthetics are a concern, it is recommended that the symbol be manually created in Cadence and then automatically replicated in ADS as described above. The symbol of the Cadence design is given the following nomenclature <library> _ <cell> _ <view> . For example, examples_PowerAmp_schematic . There is a skeleton schematic of the Cadence design that also gets created in ADS. This is used as a placeholder to link with the actual Cadence design; you do not need to edit this.

Pushing into the Design Hierarchy

To view a design deeper in the Advanced Design System schematic hierarchy:

1. Select the component you want to push into in the ADS Schematic window.

2. Choose the Push Into Hierarchy icon . This downward arrow icon is located below the Cadence menu item in the tool bar.



3. If the selected component is a Cadence cellview instance, the corresponding Cadence cellview is opened in the Cadence design editor. If this view is already open, it is simply raised to the top of the window stack.

Using Design Variables

This section describes how to add and edit design variables in Advanced Design System and also update your Cadence design variables.

Cadence Affirma Analog Circuit Design Environment design variables are intended to be global in the context of a particular Artist session or Cellview. When you select DynamicLink > Design Variables > Get Design Variables , these variables are automatically mapped to corresponding variables in a VAR component in the Advanced Design System schematic. This mapping ensures that these variables can be used for optimization or statistical analysis in ADS. All the design variables for each Cadence design are put into a single VAR component. Each time the menu item DynamicLink > Design Variables > Get Design Variables is selected, this component is updated with the most recent values from Cadence.

Adding and Editing Design Variables

To add or edit a design variable for the ADS schematic:


1. From the Cadence schematic menu bar, choose DynamicLink > Design Variables . The Cadence Design Variables form appears.

Selected Variable		Table of Design Variables	
Name	Value (Expr)	#	Name Value
Remitout	120.037opt{120to200}	1	Rout 12
		2	Rcc 10
		3	Remitin 200
		4	Remitout 120.0...


For more information one using this form, refer to Design Variables and Simulation in your Cadence

documentation.

2. In the ADS Schematic window, choose DynamicLink > Design Variables > Get Design Variables . This places a corresponding VAR component on the ADS schematic containing the design variables from Cadence. If the VAR component already exists it is updated only with variables and values that are not already there.

 **VAR**
VAR1
Cseries=7.0pF
Lshunt=7.5nH
Rout=12
Rcc=10
Remitin=200
Remitout=120.037 opt{ 120 to 200 }


VAR Block corresponding to Design Variables

 **Note**
There is no way to distinguish a design variable of the same name coming from different Cadence cellviews. If a variable has different values in different cellviews, the value sent to ADS is chosen arbitrarily.
Non-alphanumeric characters, like parentheses, in variable expressions must be preceded by a backslash (`\`).

Updating Cadence Design Variables

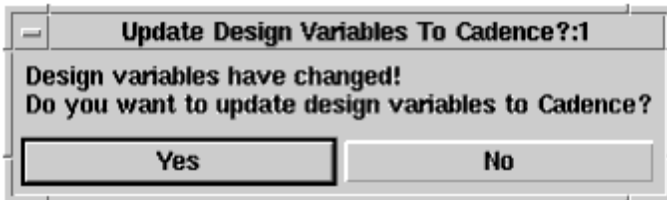
To update your Cadence design variables:

1. In the Advanced Design System Schematic window, choose DynamicLink > Design Variables > Update Design Variables .

 **Note**
Design variables can be used for optimizations and sweeps in ADS. Variables used for this purpose should not be assigned a value or expression in Cadence; the Value (Expr) text entry box should be left blank.

Closing the Cadence Connection

If you have changed your design variables in Advanced Design System and attempt to close the Cadence connection before updating your design variables in Cadence, an Update Design Variables To Cadence message will appear prompting you to update design variables.



Adding Model Files

This section describes how to use the Netlist Include Component in RFIC Dynamic Link. The Netlist Include component is provided as a means of utilizing external files in your design.

For more information on the Netlist Include component, refer to Chapter 5 of the ["../ccsim/wwhelp.htm"Introduction to Circuit Components] documentation.

Adding a Netlist File Include Component

To place an instance of the Netlist Include Component :

1. From the top-level ADS schematic window, choose DynamicLink > Add Netlist File Include . An instance of the NetlistInclude is attached to your cursor.
2. Move the cursor to where you want to place the component, then single click. A Netlist Include symbol is placed on the schematic as shown in the following figure.

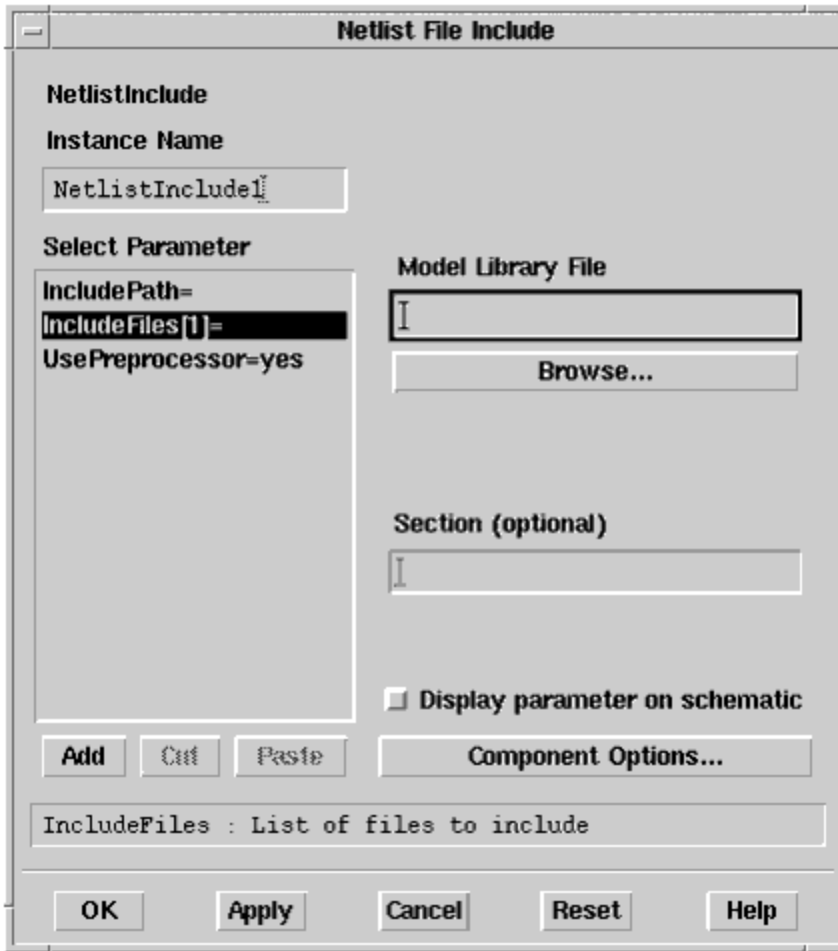


NetlistInclude
NetlistInclude1

Note
 Only one Netlist Include Component can be placed in a Dynamic Link design. This is to ensure that files are not multiply included (this cause's redefinition errors within the ADS simulator).

Accessing the Netlist File Include Dialog

To access and edit information in the Netlist File Include component, double click the Netlist Include component symbol. The Netlist File Include dialog box appears.



Select Parameter

The Select Parameter list box displays a list of three parameters that enable you to create your include definition. Refer to each of the sections listed for detailed information on defining these parameters.

- Set the path to where your model files are located ([IncludePath](#)).
- Select the model files to include (see [IncludeFiles](#)).
- Enter an optional section designator (see [Section \(optional\)](#)).
- Determine the preprocessor setting (see [UsePreprocessor](#)).

IncludePath

The includePath parameter is a space delimited search path that is used to locate included model files. The include path needs to be set up for the simulation machine in order to work properly. However, there is an issue with this. The netlister searches through the include path to find files, and then outputs the values as expanded full paths (the simulator requires this). If the expanded full path on the netlisting machine is different from the expanded full path of the simulation machine, the simulator will not find the file to be included. If you want to do remote simulations, ensure that the expanded full path of your included file is the same on the netlisting machine and the simulation machine. Note that, in directory names, path prefixes such as '.', '..', '~', and '\$' all have the usual UNIX interpretation.

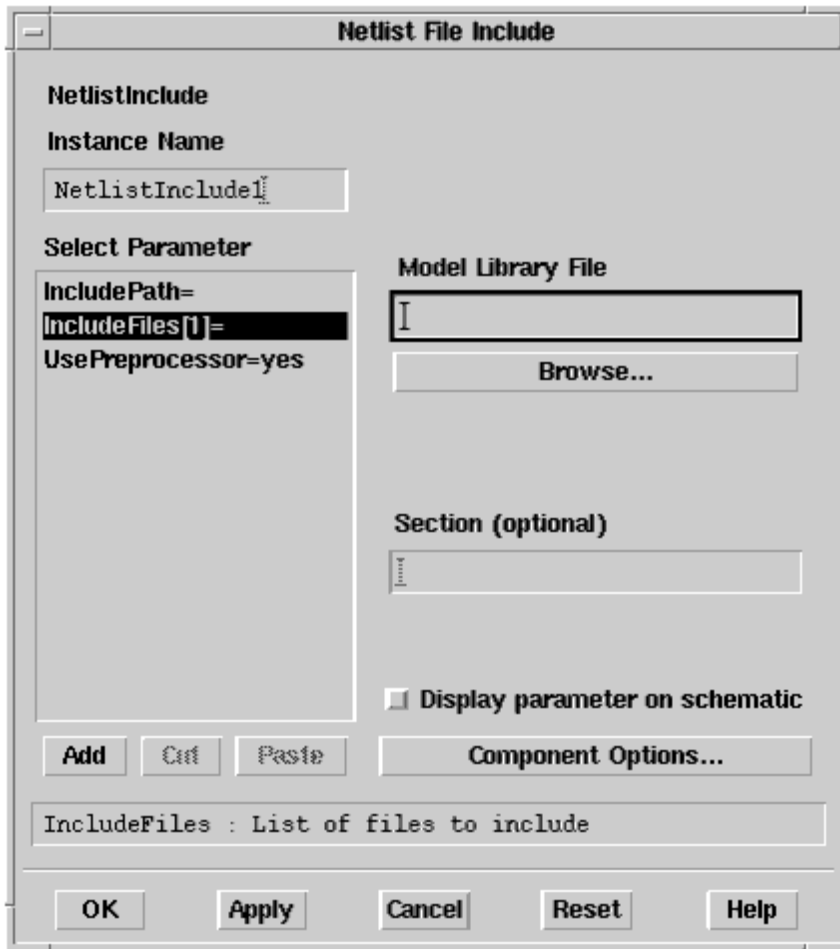
To enter a group of include paths:

1. Click the includePath= parameter in the Select Parameter list box.
2. Enter the name of the search path in the Search Path for included files (space delimited) field separating each search path by a space.

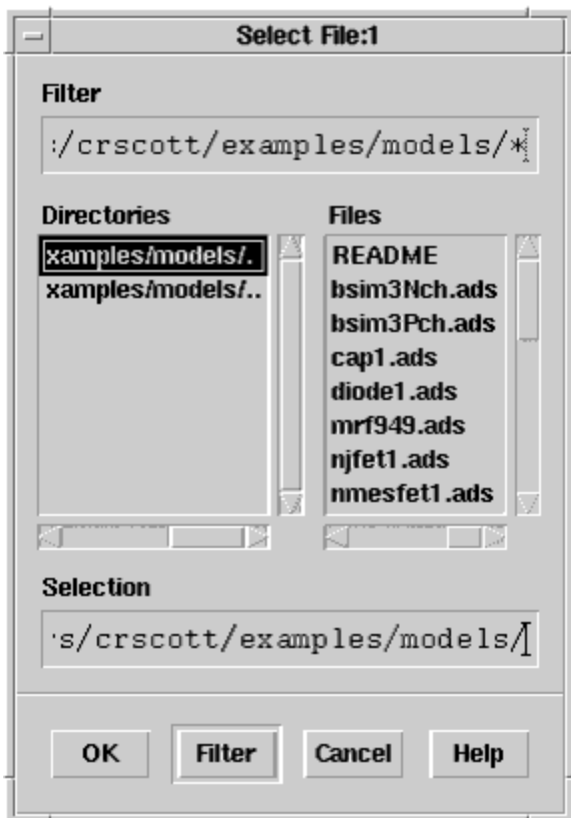
IncludeFiles

This parameter enables you to build a list of model files that you want to include. To specify the path and filename of a model file to include:

1. Click the IncludeFiles[n]= in the Select Parameter list box. This activates the Model Library File selection field.



2. Click the Browse button. The Select File dialog appears.



3. Double-click as needed to locate the directory containing your model file or enter the full path and file name in the Selection field. Click OK to return to the Netlist File Include dialog.
4. Once selected, the filename of the model file is displayed in the Library Model File field. Note that the path is appended to the includePath parameter and the file name is added to the IncludeFile parameter definition.
Example:

includePath= Path1 Path2 ... ModelFilePath

IncludeFiles[n]= filename

5. To add additional model files, click Add . This creates additional model file parameter definitions in the Select Parameter list box. Repeat steps 1 through 4 to define the path and file name. You can continue adding model file parameters as needed. You can also use the Cut and Paste buttons to move or delete any model file parameters.

Section (optional)

You may only have a single file for each IncludeFiles[n] parameter - unlike the prior parameters - this is not a space delimited list. Each model file can have a Section designator. This enables you to include only a portion of a model file

Advanced Design System 2008

for corner analysis , provided your model file has been set up properly. The section designator is optional; if it is left empty, the entire file will be included (provided it has no dependencies on needing a particular section set up). To properly set up a model file to utilize the section directive, you must bracket the sections using `#ifdef <section>/#endif` C-Pre-Processor(CPP) directives. The netlister automatically defines and undefines a variable with the name section before and after the `#include` statement. As an example, if you wanted to have a file with corner cases, and had a Nominal section, you would make the file as follows:

```
#ifdef Nominal
; Nominal section
R:R1 in out R=50
#endif
```

If the same library file is named, with a different section, a single `#include` is generated, with multiple `#define` statements around it.

UsePreprocessor

The UsePreprocessor parameter defaults to " yes" . Setting this parameter to yes causes a preprocessor directive (`#include`) to be used which results in faster behavior by not copying the full text of the file. This parameter can also be set to no to provide a slower, but more compatible, full-text inclusion behavior.

To set the UsePreprocessor parameter,

1. Click the UsePreprocessor=yes parameter in the Select Parameter field of the Netlist File Include dialog box. The Parameter Entry Mode pull-down menu appears.
2. Click the Parameter Entry Mode pull-down menu in the Netlist File Include dialog box and select the appropriate option.



The UsePreprocessor parameter was developed for delivering increased speed to old ADS designs that use the gminiInclude , spiceInclude or csvInclude component and for delivering increased flexibility to old ADS designs using the idfInclude component.

Display parameter on schematic

The Display parameter on schematic check box in the Netlist File Include dialog box is used to list the individual parameters and their associated values on the schematic. If you want to display the parameters, activate the check box.

Component Options

The Component Options dialog box enables you to change the visibility of the component parameters on a schematic and/or reference items in hierarchical designs. To access the Component Options, click the Component Options button on the Netlist File Include dialog box. A Component Options dialog box appears.



Changing the Visibility of Component Parameters on a Schematic

You can change the visibility status of all parameters of the Netlist File Include component through the Component Options dialog box.

- Set All-Displays all parameters for this component on the schematic. Use this option to display all, or almost all, parameters for this component. To display most-but not all-parameters, select Set All and then go back and turn off the display of individual parameters as desired.
- Clear All-Clears the display of all parameters for this component from the schematic. Use this option to turn off the display of all, or almost all, parameters for this component. To display a small subset of parameters, select Clear All and then go back and turn on the display of individual parameters as desired.

Referencing VAR Data Items and Model Items in Hierarchical Designs

The Scope option applies to the VAR (Variables and Equations) data item and most model items (such as R_Model, BJT_Model, BSIM3_Model). Exception: it does not apply to multi-layer models. Scope indicates the levels, from a hierarchical standpoint, that recognize the expressions defined in the VAR data item or model item.

- Nested-VAR or model item expressions are recognized within the design containing the VAR or model item, as well as within any subnetworks (designs at lower levels) referenced by the design containing the VAR or model item.
Global-VAR or model item expressions are recognized throughout the entire design, no matter what level in the design hierarchy the VAR or model item is placed.

Summarizing the Netlist File Include Component

Advanced Design System 2008

For all of the Netlist File Include component parameters, a single include statement is netlisted for each file. The netlister checks to see if a file has already been output, to avoid having multiple definitions of files. The precedence is that model files are output first, so that the segment directives can be placed around the #include .

If you are using the Netlist File Include component, it is not putting out #ifdef <file> statements to further ensure that files are not multiply included. If you use a Netlist File Include component, you should not additionally use other file include components to avoid multiple inclusions which will cause a simulator redefinition error.

Example:

Parameter settings

```
includePath=". ./models"
definitionFiles="functions.def"
stimulusFiles="vccdef.stim"
modelLibraryFiles\[1\]="resistor.lib Nominal"
```

Netlist File Output (Note that . is /users/default/default_prj in this example):

```
#define Nominal
#include "/users/default/default_prj/models/resistor.lib"
#undef Nominal
#include "/users/default/default_prj/models/functions.def"
#include "/users/default/default_prj/models/vccdef.stim"
```

It is worth noting that, once the Netlist File Include component is netlisted, the simulator makes no differentiation between definition, stimulus, or model files. Each file will generate the #include __ statements.

You may want to use the IncludeFiles parameter for all of your files so that you can put corner case statements into all of your model files.

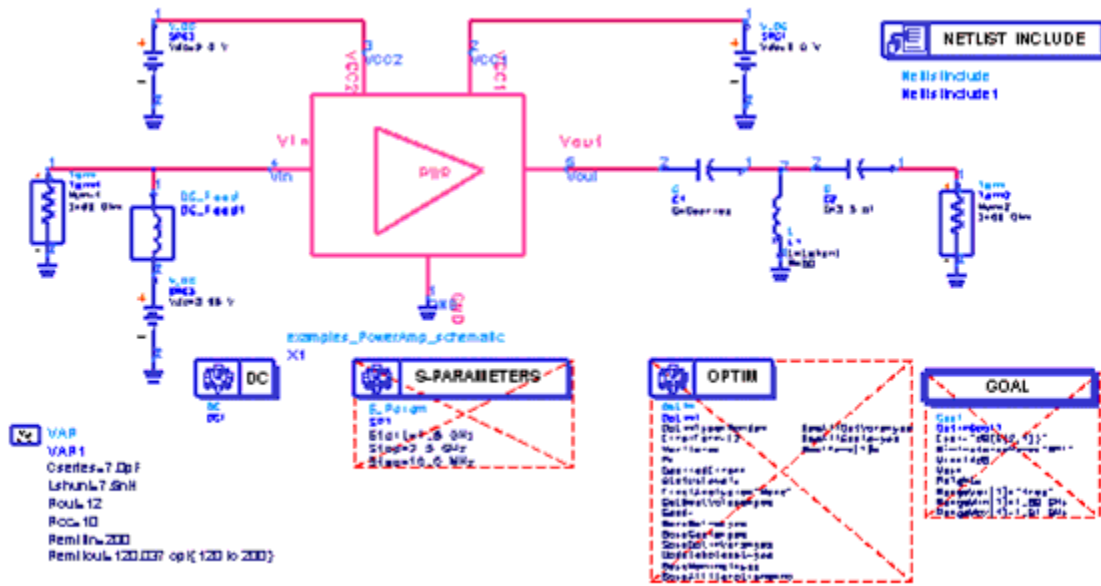
Annotating a Cellview

This section describes how to annotate your simulation results in Advanced Design System to a Cadence cellview.

Annotating DC Voltages to a Cadence Cellview

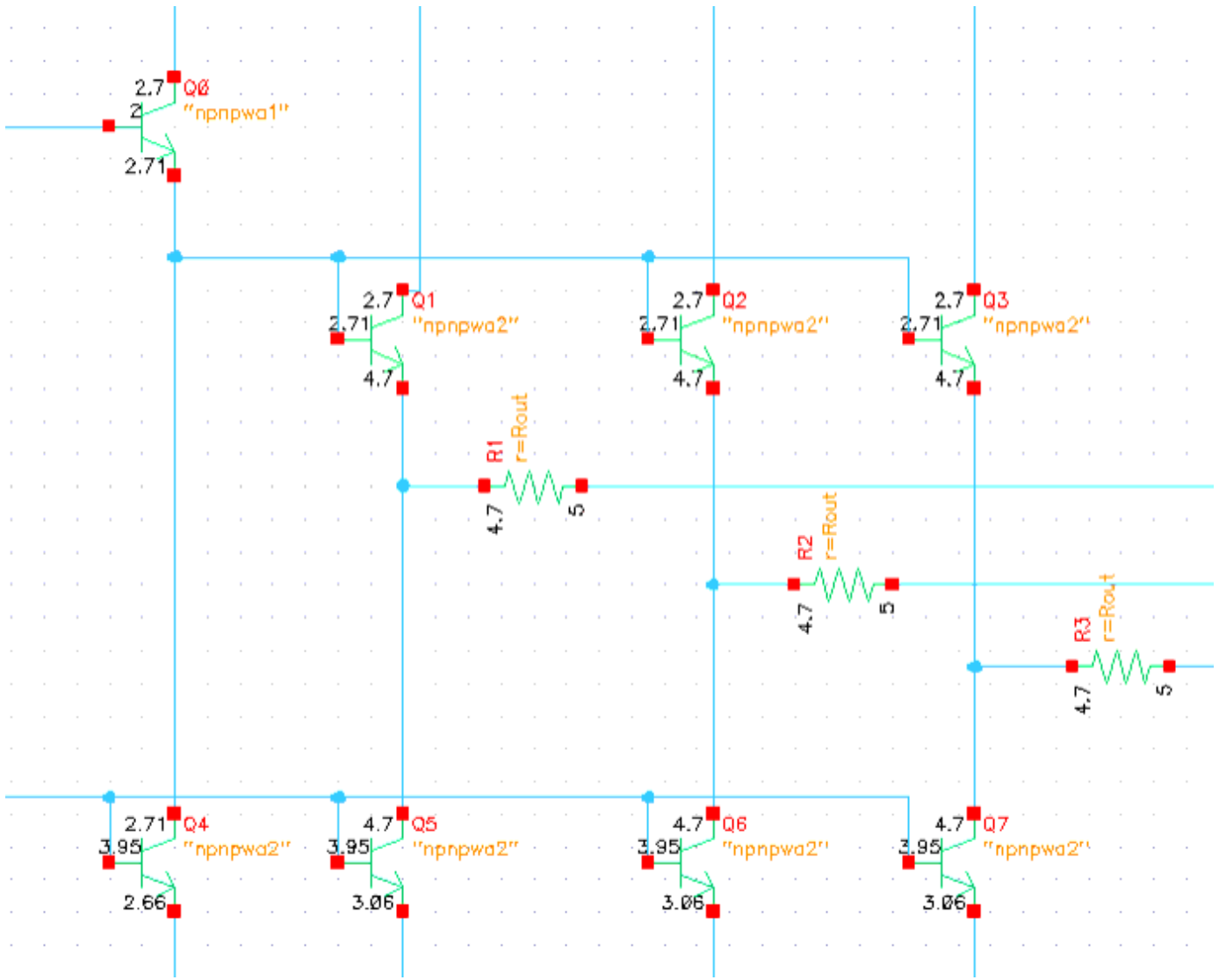
To annotate a DC voltage solution in Advanced Design System to the Cadence cellview:

1. In Advanced Design System, set up and simulate your schematic. This schematic must contain a DC Simulation Component as shown in [Example setup for DC Simulation](#).



Example setup for DC Simulation

2. Select the schematic symbol in the ADS schematic that represents the Cadence circuit you want to back annotate. For example, the amplifier block in the preceding figure.
3. From the ADS Schematic window, choose DynamicLink > Annotate > Annotate DC Solution to Selected Cellview .
4. The voltages are then displayed on the Cadence schematic as shown in the following figure.



DC Voltage Annotation on the Cadence Schematic

Annotating DC Currents to a Cadence Cellview

To annotate a DC current solution in Advanced Design System to the Cadence cellview:

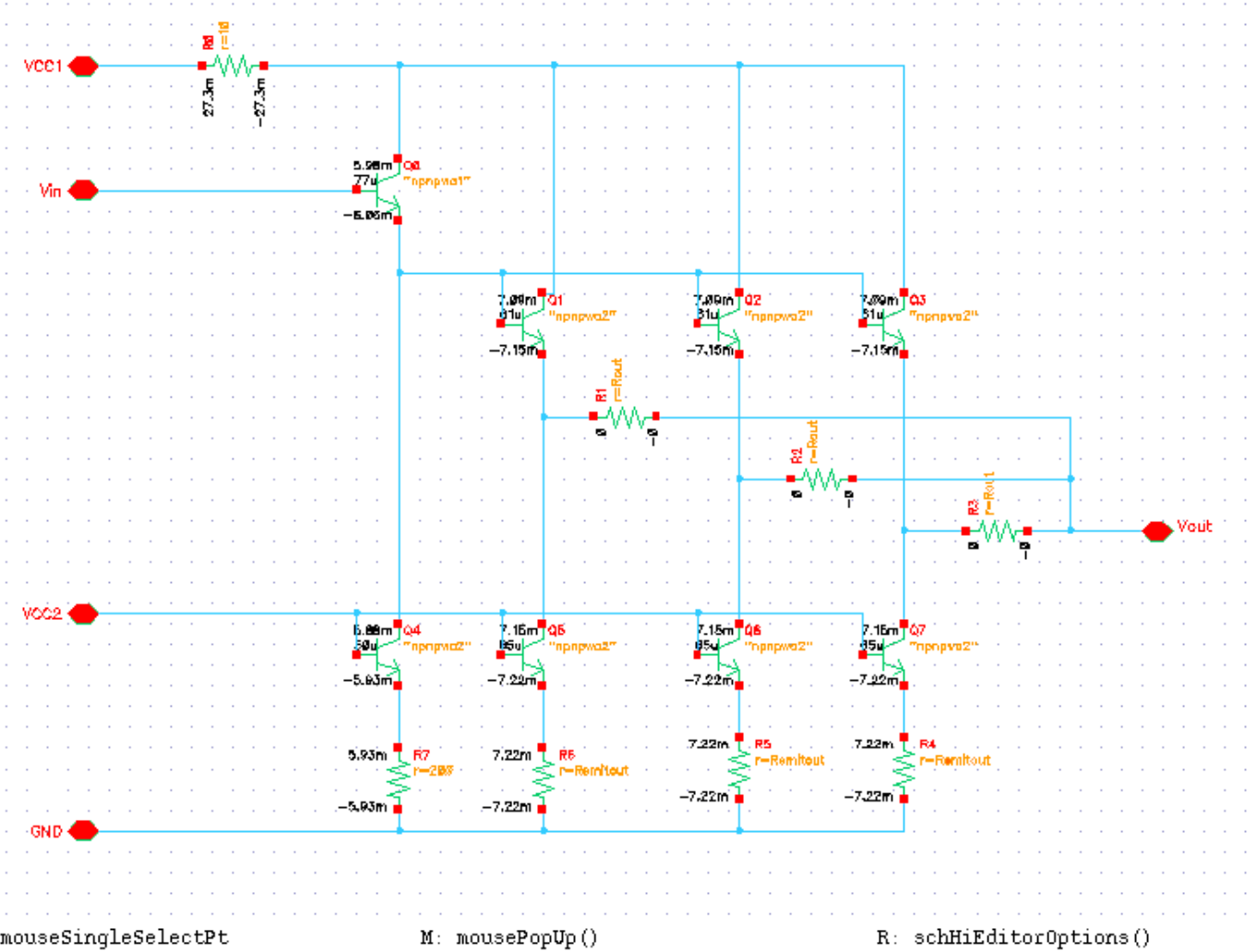
1. First annotate the DC voltages as described in [Annotating DC Voltages to a Cadence Cellview](#).
2. From the Cadence schematic window, choose Edit > Component Display . The Edit Component Display Options form appears.
3. Click an instance in the Cadence Schematic window. For this example, Q0 was selected. Note that the title of the Edit Component Display Options form changes to include the component selected. In this case, the title Edit Component `Q0' Display appears.

- Click the terminal checkbox from the Select Label options. Notice that the form now displays a Terminal Labels section. This section shows that DC and voltage is currently selected as seen in [Edit Component `Q0' Display form showing DC Voltage](#).

The image shows a dialog box titled "Edit Component `Q0' Display". At the top, there are buttons for "OK", "Cancel", "Previous", "Next", and "Help". Below these are several options: "Select Labe" with radio buttons for "terminal" (checked), "parameter", and "instance"; "Auto Redraw" with radio buttons for "yes" (checked) and "no"; and buttons for "Save", "Load", "Attach", "Detach", and "Set Simulation Data Directory". A section titled "Terminal Labels" contains "Apply To" with radio buttons for "library" (checked), "cell", and "instance"; "Terminal Labels" with radio buttons for "none", "pin name", "net name", "voltage" (checked), and "current"; and another set of radio buttons for "DC" (checked) and "transient".

Edit Component `Q0' Display form showing DC Voltage

- Click the currents checkbox in the Terminal Labels section of the Edit Component Display form to display currents instead of voltage. The next figure shows the design with the DC currents annotated.
- Click OK to clear the Edit Component `Q0' Display form.



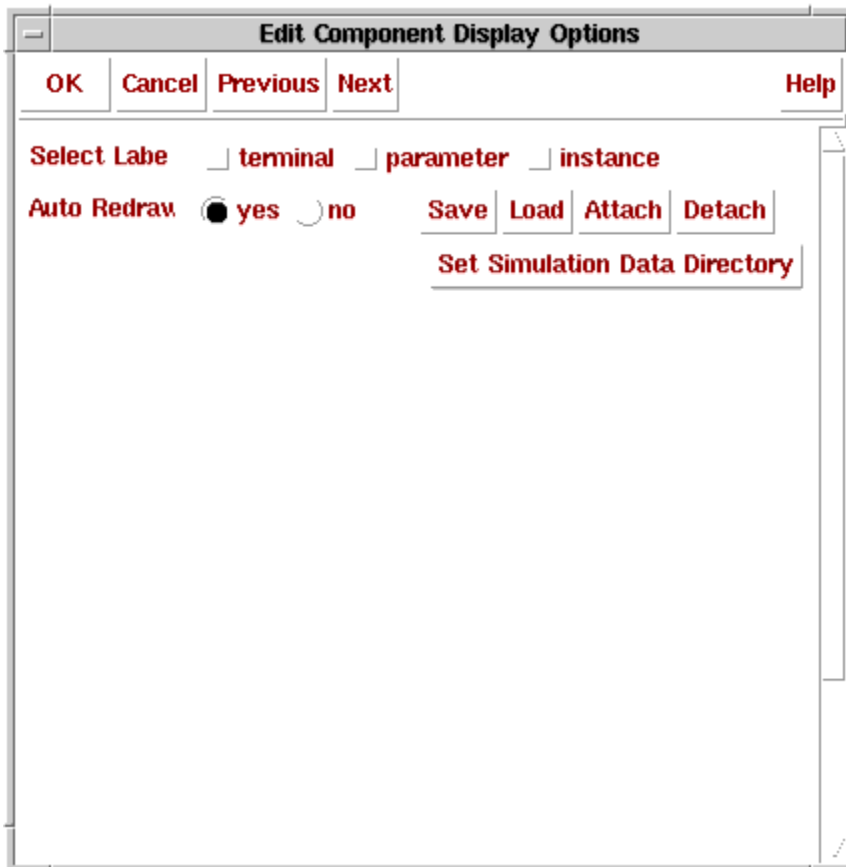
DC Current Annotation on the Cadence Schematic

Note
 If you do not select an instance, the library that your changes will apply to will be the library that contains the schematic (i.e. examples for the PowerAmp example). Since the primitives (i.e. the res and npn cells) are not in the schematic's library, you will not see any changes to the annotation if you do not first select an instance from the proper library. The Edit Component Display form enables you to control how labels are displayed for each library, cell and instance in the design.

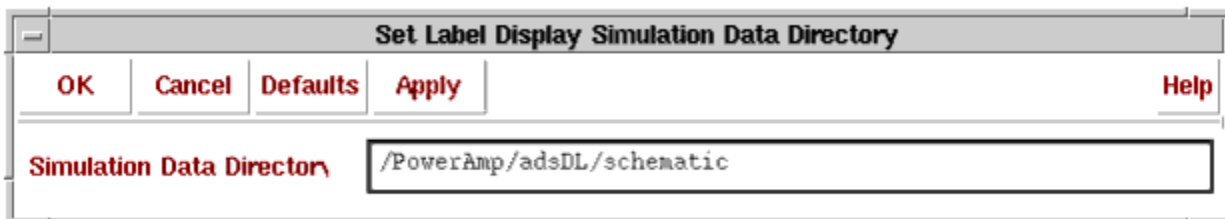
Displaying Voltages or Currents from a Previous Simulation

To display voltages or currents from a previous simulation:

1. Before displaying voltages or currents you must have annotated a DC solution to the schematic in a prior Cadence session. Follow the instructions for annotating a DC solution (see [Annotating a Cellview](#)) to a schematic if you have not already done so.
2. From the Cadence Schematic window, choose Edit > Component Display . The Edit Component Display Options form appears.



3. Click Set Simulation Data Directory . The Set Label Display Simulation Data Directory form appears.



4. Enter the full path to the Data Directory. This is everything up to the psf directory. The psf directory contains Cadence formatted data. The structure for the path name is, < Cadence_project_dir >/< cell_name >/< tool_name >/< view >
The path for the Data Directory used in the example for [DC Voltage Annotation on the Cadence Schematic](#) was, ~/simulation/PowerAmp/adsDL/schematic
The annotation code looks in the psf directory.
5. Click OK in the Set Label Display Simulation Data Directory form.
6. Click OK in the Edit Component Display Options form.

Creating Symbols for Hierarchical Subcircuits with cdsTerm

If you want to annotate a hierarchical Cadence design, you must create a Cadence symbol for the design. To create symbols for hierarchical subcircuits using cdsTerm :

1. From the Cadence Schematic window, choose Design > Create Cellview > From Cellview . The Cellview From Cellview form appears.

2. In the Cellview From Cellview form, ensure the following settings are correct:
 - From View Name - schematic
 - To View Name - symbol
 - Tool / Data Type - Composer-Symbol
 Click OK. The Symbol Generation Options form appears, assuming a symbol does not already exist.
3. In the Symbol Generation Options form, click the Edit Labels checkbox. Your form will display the Label options.
4. In the Symbol Generation Options form, select analog pin annotate from the Label Choice pull-down menu. The Name field should now display cdsTerm("(pinname)").
5. Select all pins from the Apply To drop-down menu and click Add . This generates a new label rule that creates a cdsTerm for each pin. You may alter the location if you choose. The form with all appropriate option settings is shown in the following figure.

[learn2] Symbol Generation Options [SHARED]

OK Cancel Apply Help

Library Name: example Cell Name: hierarchySymbolCreation View Name: symbol

Pin Specifications:

Pin Specification	Attributes
Left Pins: in	List
Right Pins: out	List
Top Pin:	List
Bottom Pin:	List

Load/Save Edit Attributes Edit Labels Edit Properties

Label: 1. {pinName} (bottom) Add Delete Modify

Name: cdsTerm("{pinName}") Label Choice: analog pin annotate

Apply To: all pins Label Type: ILLabel

Orientation: R270 Font Height: 0.0625

Justification: centerRight Font Style: stick isOverba

Location: upperCenter

Symbol Generation Options to create a symbol with cdsTerms on each pin

For more information on the Symbol Generation Option form, refer to your Cadence documentation.

Netlisting, Simulating, and Displaying Data

This chapter describes the procedures for netlisting and simulating a design as well as viewing the netlist from either Advanced Design System or a Cadence Schematic window. Information on net, instance and expression name mapping is also provided.

Netlisting and Simulating a Design

Netlisting automatically occurs when you simulate your schematic in Advanced Design System. The complete netlist is sent to the simulator, stored in memory, and written to a netlist.log file in the project directory of ADS. You can view the netlist in the netlist.log file as needed.

To netlist and simulate a schematic in Advanced Design System:

1. In the ADS Schematic window, choose the Simulate icon or choose the menu item Simulate > Simulate .



A Simulation window appears, indicating the netlisting status and listing any errors encountered. If the netlisting is successful, the design is then simulated; otherwise, act on the errors displayed in the Simulation window and repeat step one above.

2. For information on configuring and viewing the simulation results in the Data Display window, refer to [Data Display Basics](#) in the [Data Display](#) documentation.

Viewing Netlists

This section describes how to view the top-level netlist from Advanced Design System as well as how to view an ADS subnetwork netlist for a Cadence design from a Cadence Schematic window.

Viewing Netlists from Advanced Design System

To generate and display the entire top-level ADS netlist, select DynamicLink > Top-level Design Netlist .

The screenshot shows a window titled "Hpeesofeedit: [./netlist.log]". The window contains a netlist for a circuit named "PowerAmp". The netlist includes component definitions, pin connections, and resistor values. The status bar at the bottom indicates "Hpeesofeedit (TM) 1.0 150.200" and "1, 1".

```
Options ResourceUsage=yes UseNutmegFormat=no TopDesignName="/a/new/sr/learn2/d1/local/users/crscor

; Library name: examples
; Cell name: PowerAmp
; View name: schematic
#ifndef inc_examples_PowerAmp_schematic
#define inc_examples_PowerAmp_schematic
define examples_PowerAmp_schematic ( GND VCC1 VCC2 Vin Vout)
nnpwa2:Q7 net6 VCC2 net37 0
nnpwa2:Q6 net9 VCC2 net35 0
nnpwa2:Q5 net12 VCC2 net13 0
nnpwa2:Q4 net15 VCC2 net31 0
nnpwa2:Q3 net44 net15 net6 0
nnpwa2:Q2 net44 net15 net9 0
nnpwa2:Q1 net44 net15 net12 0
nnpwa1:Q0 net44 Vin net15 0
R:R7 net31 GND R=200
R:R6 net13 GND R=Remitout
R:R5 net35 GND R=Remitout
R:R4 net37 GND R=Remitout
R:R3 net6 Vout R=Rout
R:R2 net9 Vout R=Rout
R:R1 net12 Vout R=Rout
R:R0 VCC1 net44 R=10 Tnom=25
```

Viewing Top-Level ADS Netlist

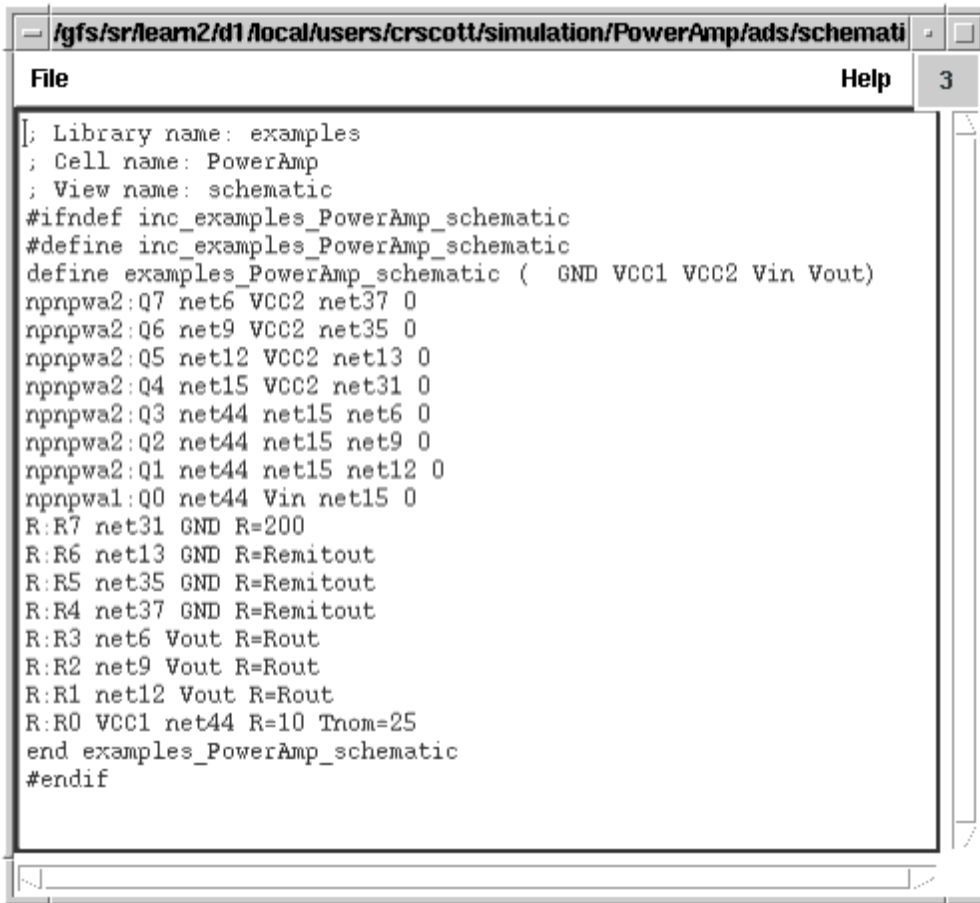
Viewing Netlists from the Cadence Schematic Window

To generate and display the ADS subnetwork netlist for the Cadence design displayed in a particular Cadence Schematic window:

1. From the menu bar, select DynamicLink > Subcircuit Netlist . Netlisting progress is displayed in the Cadence CIW.

Note
If the Dynamic Link pull-down menu does not appear in the Cadence Virtuoso Schematic window, choose Tools

> ADS Dynamic Link > Add Dynamic Link menu to all schematic windows in the Cadence Command Interpreter Window (CIW).



```

- |gfs/sr/learn2/d1/local/users/crscott/simulation/PowerAmp/ads/schemati
File Help 3
[; Library name: examples
; Cell name: PowerAmp
; View name: schematic
#ifndef inc_examples_PowerAmp_schematic
#define inc_examples_PowerAmp_schematic
define examples_PowerAmp_schematic ( GND VCC1 VCC2 Vin Vout)
nprnpwa2:Q7 net6 VCC2 net37 0
nprnpwa2:Q6 net9 VCC2 net35 0
nprnpwa2:Q5 net12 VCC2 net13 0
nprnpwa2:Q4 net15 VCC2 net31 0
nprnpwa2:Q3 net44 net15 net6 0
nprnpwa2:Q2 net44 net15 net9 0
nprnpwa2:Q1 net44 net15 net12 0
nprnpwa1:Q0 net44 Vin net15 0
R:R7 net31 GND R=200
R:R6 net13 GND R=Remitout
R:R5 net35 GND R=Remitout
R:R4 net37 GND R=Remitout
R:R3 net6 Vout R=Rout
R:R2 net9 Vout R=Rout
R:R1 net12 Vout R=Rout
R:R0 VCC1 net44 R=10 Tnom=25
end examples_PowerAmp_schematic
#endif

```

Viewing a Subnetwork Netlist

2. A log window pops up, displaying the netlist results. Once you have viewed the results, you can select File > Close Window to exit this window.

Net and Instance Name Mapping

Since Advanced Design System nomenclature rules differ from those of Cadence, nets, instances, etc. must be properly mapped. This mapping is done automatically as part of the netlisting function. The mapping rules are as follows.

- Advanced Design System keywords used as net or instance names are mapped by appending an underscore (_)

to the name.

Name	Map
then	then_
else	else_
elseif	elseif_
endif	endif_
equals	equals_
notequals	notequals_
and	and_
not	not_
or	or_
global	global_
model	model_
define	define_
end	end_
parameters	parameters_

- Any non-alphabetical character (e.g. not a-z) in a net or instance name is mapped to an under bar (_).
- Advanced Design System uses a single name space for all names, regardless of object type (net, instance, etc.). This may necessitate name mapping in addition to the above.

Expression Name Mapping

Most Cadence Analog Expression Language (AEL) expressions contain constants, functions and suffixes with equivalents in ADS. In most cases the names of these equivalents are identical, requiring no mapping. As far as possible, Cadence expressions are pre-evaluated in the Cadence environment, prior to netlisting and prior to getting design variables from Cadence. This leaves only a few built-in function names to map (i.e. names that are not identical in the two environments).

Cadence	ADS
complex	cmplx
fabs	abs
log	ln
log10	log

Some built-in operator and function names in the Cadence Affirma Analog Circuit Design Environment as yet do not

map to anything in the ADS environment.

Non-Mapping Operators

Cadence	Description
-	unary minus
~	unary one's complement
%	modulo
<<	left shift
>>	right shift
&	bitwise AND
	bitwise OR
^	bitwise XOR
?:	conditional expression

For these non-mapping functions, custom equivalents in ADS need to be written and mapped until they are available as built-ins in ADS. Custom mapping is enabled via the configuration file option `IDF_EXPR_MAP`. For more information, refer to Expression Mapping in [Modifying the Configuration File](#).

Non-Mapping Functions

Cadence	Description
acosh	inverse hyperbolic cosine
asinh	inverse hyperbolic sine
atanh	inverse hyperbolic tangent
aelCheckRange	determines if a number falls within a range
conjugate	complex conjugate
floor	floor of a real number
ceil	ceiling of real number
mag	magnitude
db10	10 times log10
db20	20 times log10

Using Global Nodes

Cadence designs typically use implicit global nodes (names ending in !) for substrate power and ground connections. This notation is now supported by Advanced Design System. If the exclamation point suffix is used, a globalnode does not need to be placed in the ADS schematic.

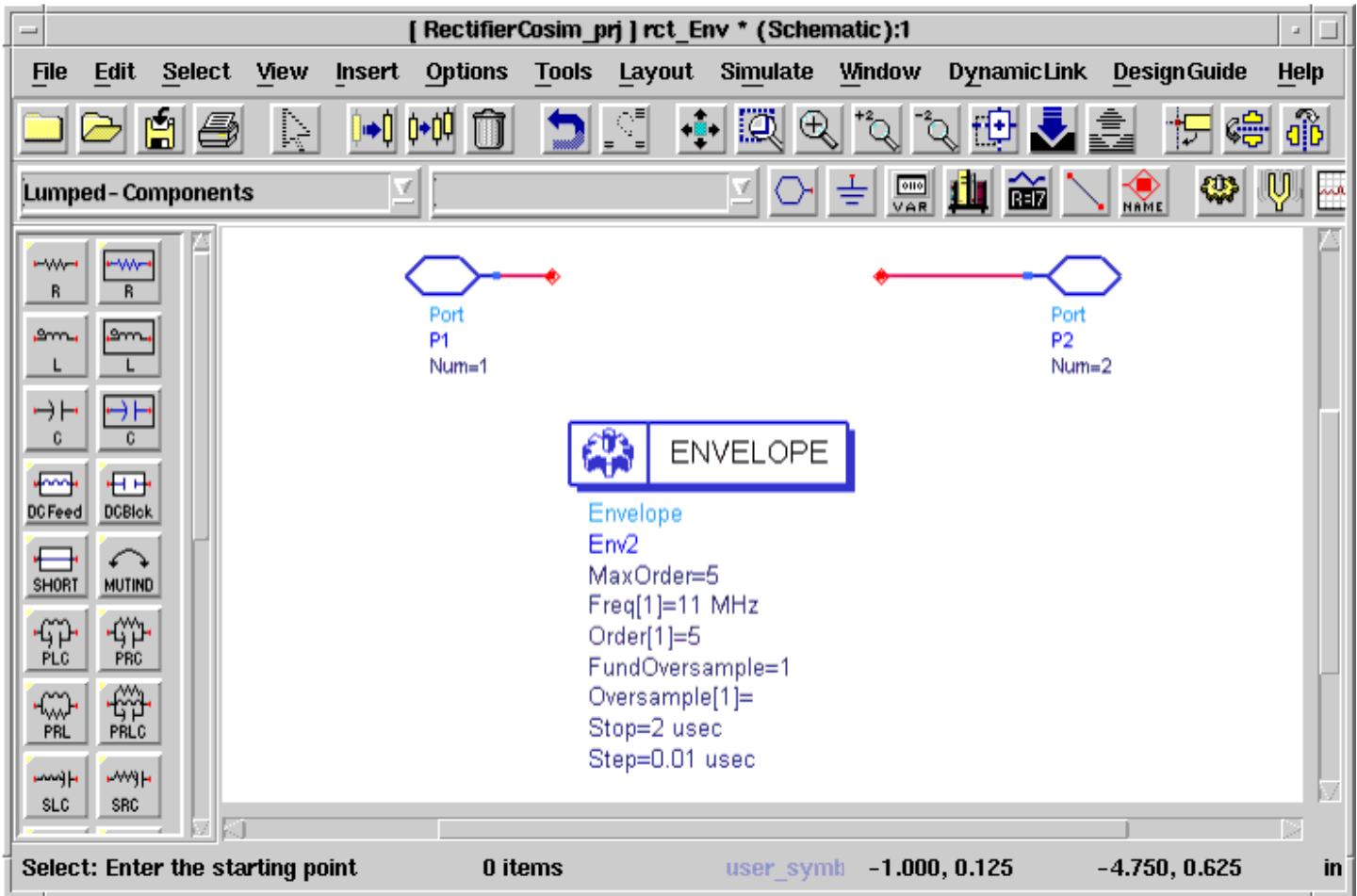
Running a DSP and Analog/RF Cosimulation with RFIC Dynamic Link

This chapter describes how to run cosimulation of behavioral Digital Signal Processing (DSP) designs along with Analog/RF circuit designs. Using RFIC Dynamic Link, you can perform a cosimulation with DSP designs in Advanced Design System (ADS) and your Analog/RF designs in Cadence. To run a cosimulation using RFIC Dynamic Link, you will need to create an intermediate layer ADS design. This intermediate design must contain a dummy ADS design representing the Cadence cellview in ADS and also include either an ADS Transient (Tran) or Envelope simulation controller.

For more detailed information about ADS Cosimulation, refer to the [ADS Ptolemy Simulation](#) documentation.

RFIC Dynamic Link DSP and Analog/RF Cosimulation Example

The following example uses the RectifiedCx design (see the following figure) provided in the ADS Communication Systems RectifierCosim_prj example project to demonstrate running a DSP and Analog/RF cosimulation using RFIC Dynamic Link.

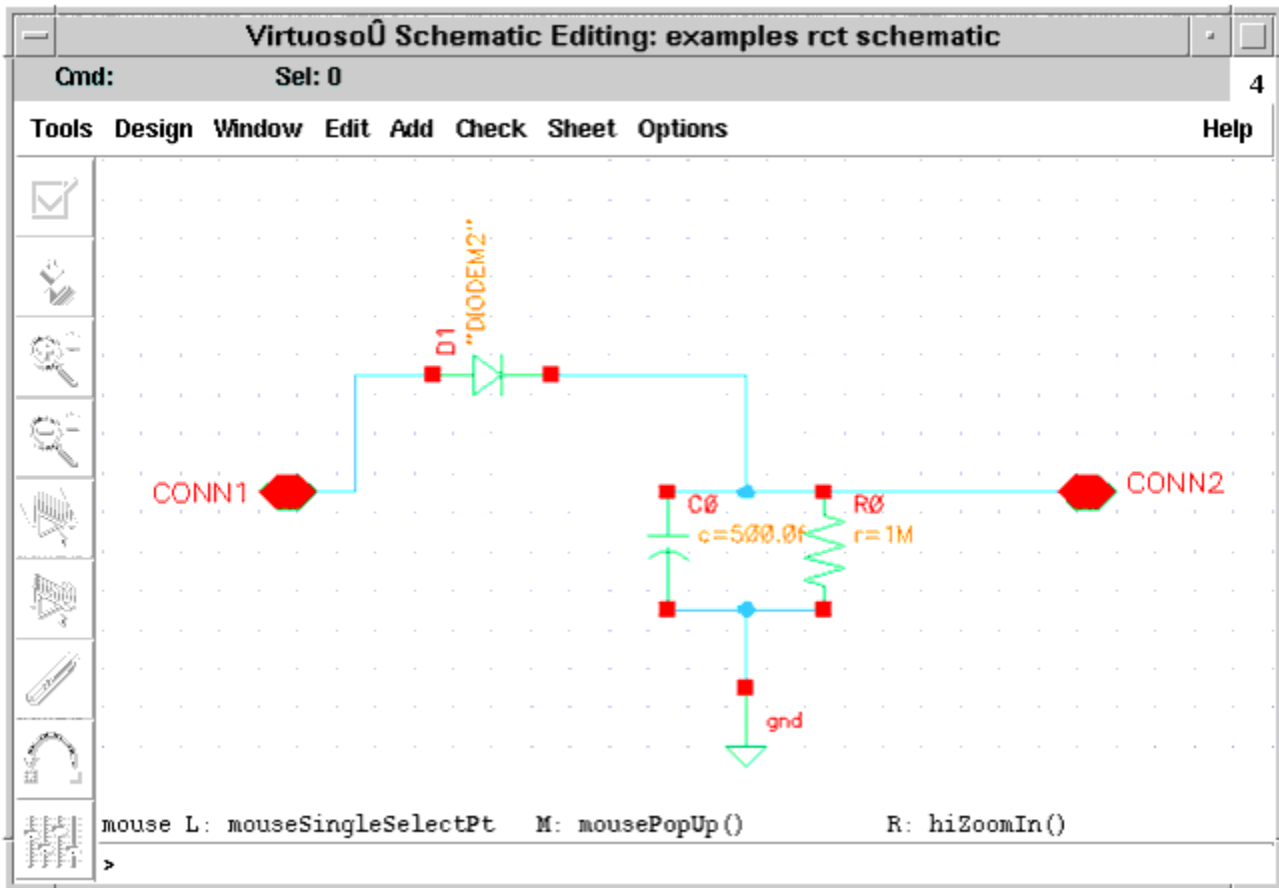


The rct_Env_Cadence Intermediate Design

Before you begin, ensure you are set up to run the RFIC Dynamic Link examples as described in [Setting up the Examples Directory](#).

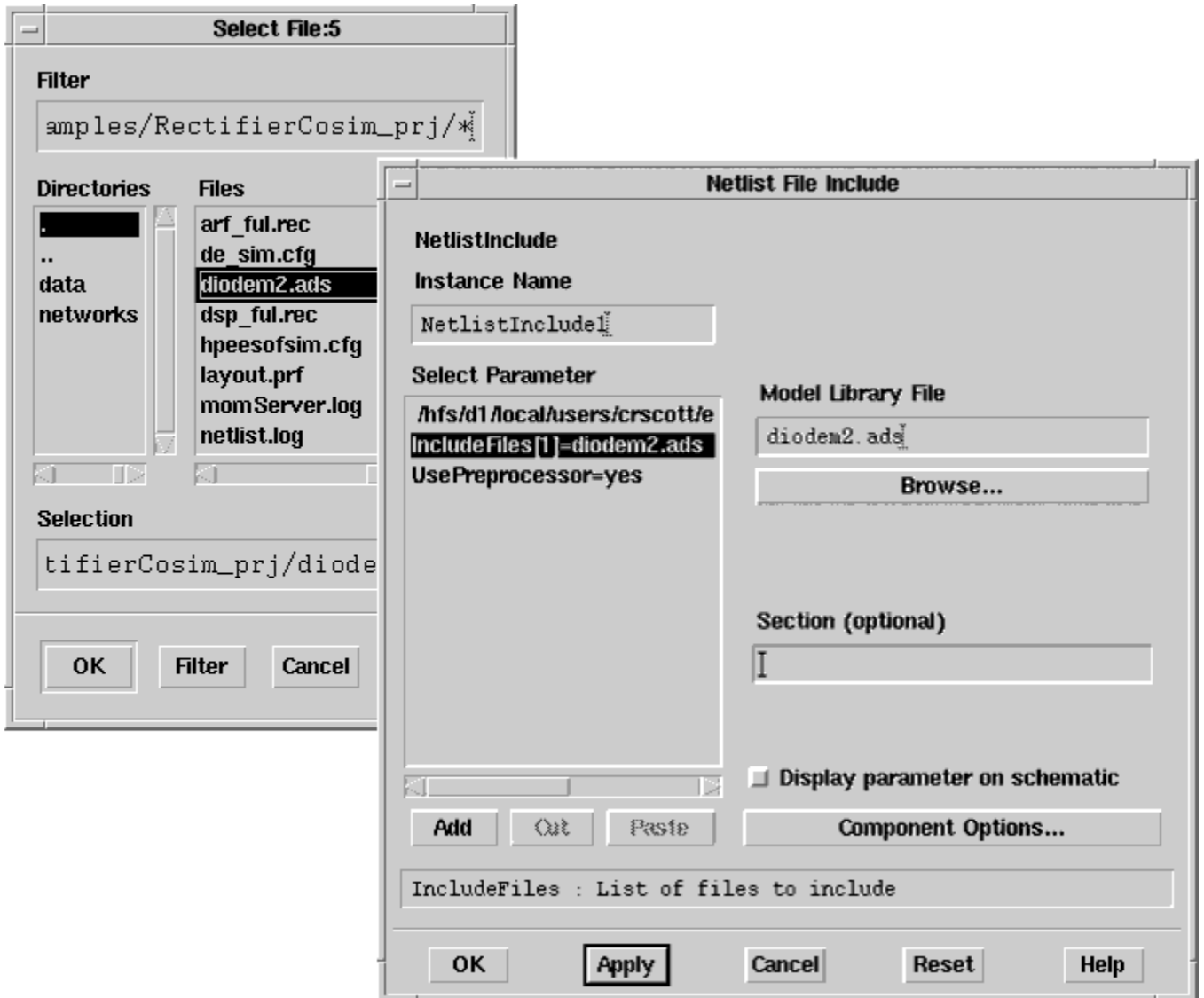
1. Open the rct cellview under the examples library. This cellview is a duplicate of the Advanced Design System rct_Env.dsn in the ADS example project \$HPEESOF_DIR/examples/Com_Sys/RectifierCosim_prj.

Note
As an alternative to performing steps 2 to 4 below, you can simply execute the get_cosim_example.sh script in the examples directory at your system prompt.



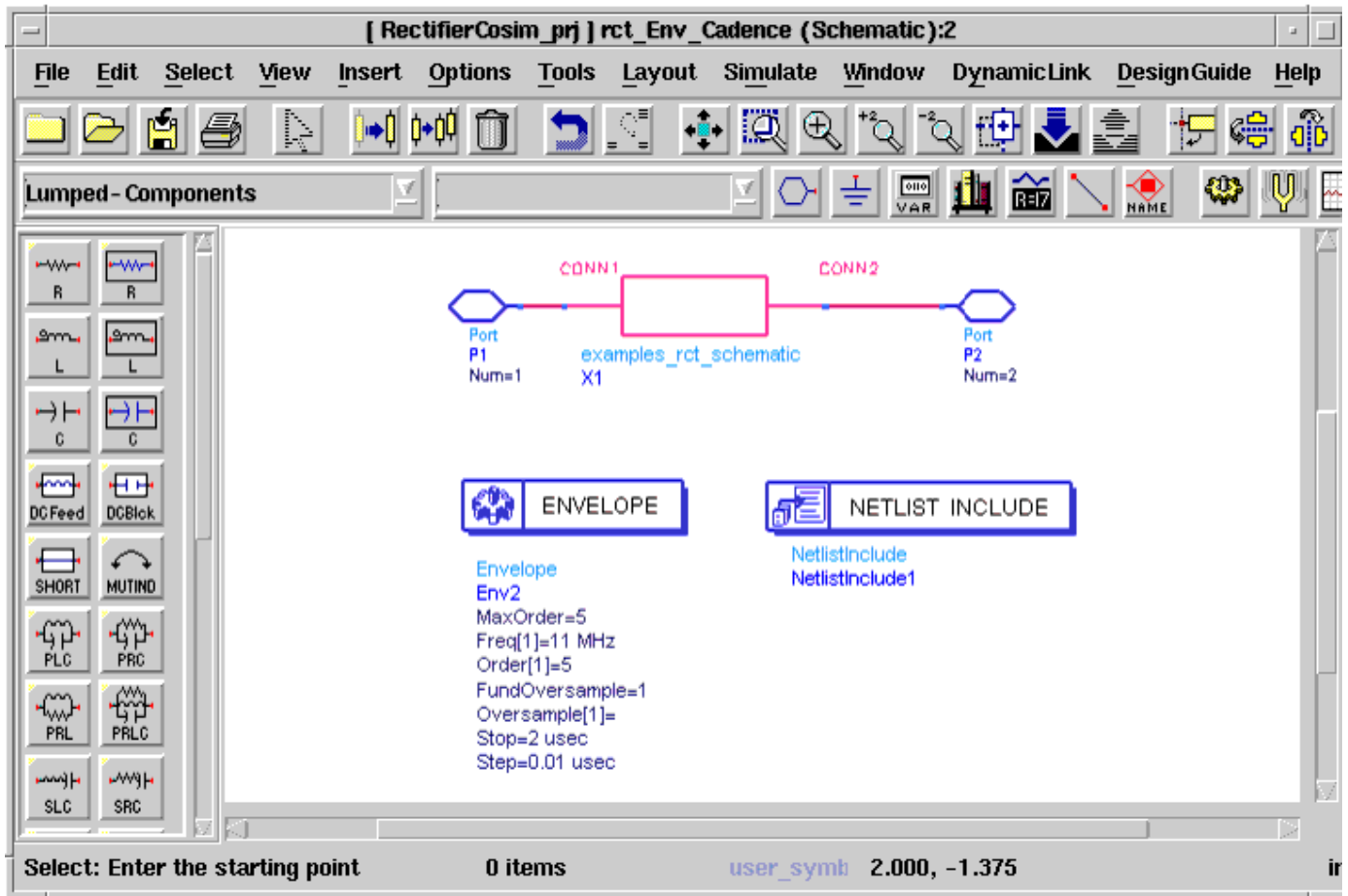
The rct Cell in the examples Library

2. Copy the ADS example project `$HPEESOF_DIR/examples/Com_Sys/RectifierCosim_prj` to a local directory.
3. Copy `$HPEESOF_DIR/idf/examples/examples_prj/networks/rct_Env_Cadence.*` to the local `RectifierCosim_prj/networks` directory. This is the intermediate ADS design for this example.
4. Copy `$HPEESOF_DIR/idf/examples/models/diodem2.ads` to the local `RectifierCosim_prj` directory. This model file contains the same ADS netlist as generated for the diode `DIODEM2` in the original `rct_Env` subcircuit.
5. Start ADS by selecting `Tools > ADS Dynamic Link` in the Cadence schematic window.
6. Open the local `RectifierCosim_prj` from the ADS Main window, then open the `rct_Env_Cadence` ADS design.
7. Choose `DynamicLink > Instance > Add Instance of Cellview` to place the `examples_rct_schematic` design between the two ports.
8. Add a `NetlistInclude` component to the `rct_Env_Cadence` design by choosing `DynamicLink > Add Netlist File Include`.
9. Double click the `NetlistInclude` component and apply the `diodem2.ads` model file to the `IncludeFiles` parameter.



Adding the NetlistInclude Component

- Click OK to update the NetlistInclude component. The modified rct_Env_Cadence design should now appear similar to the following figure.

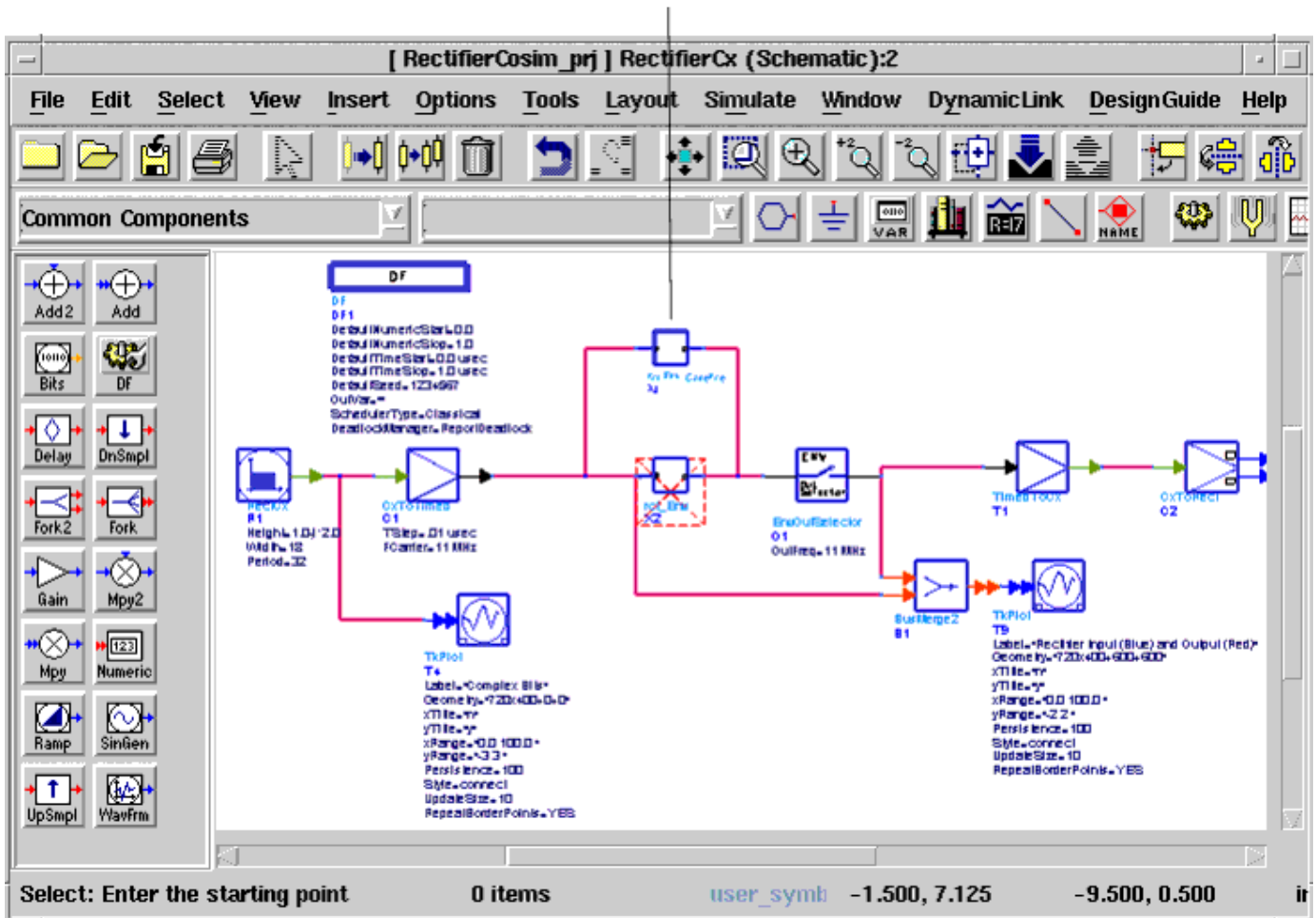


The Modified rct_Env_Cadence Design

- Replace the rct_Env subcircuit component with rct_Env_Cadence in the RectifiedCx design by choosing Insert > Component > Component Library and then selecting the rct_Env_Cadence from the Library Browser.

Note
If a Failed to locate the component definition error occurs, you may need to create a symbol. Open the rct_Env_Cadence design and select View > Create/Edit Schematic Symbol to create a new symbol.

rct_Env_Cadence



The RectifiedCx Design with rct_Env_Cadence

12. Choose Simulate > Simulate in the ADS schematic window of RectifierCx to simulate the design in ADS. Simulation results of the modified RectifierCx design with rct_Env_Cadence subcircuit should be identical with that of the original RectifierCx Design where the rct_Env subcircuit was included.

Tuning and Optimizing Designs

This chapter provides information on tuning and optimizing designs using the Advanced Design System tuning and optimization capabilities.

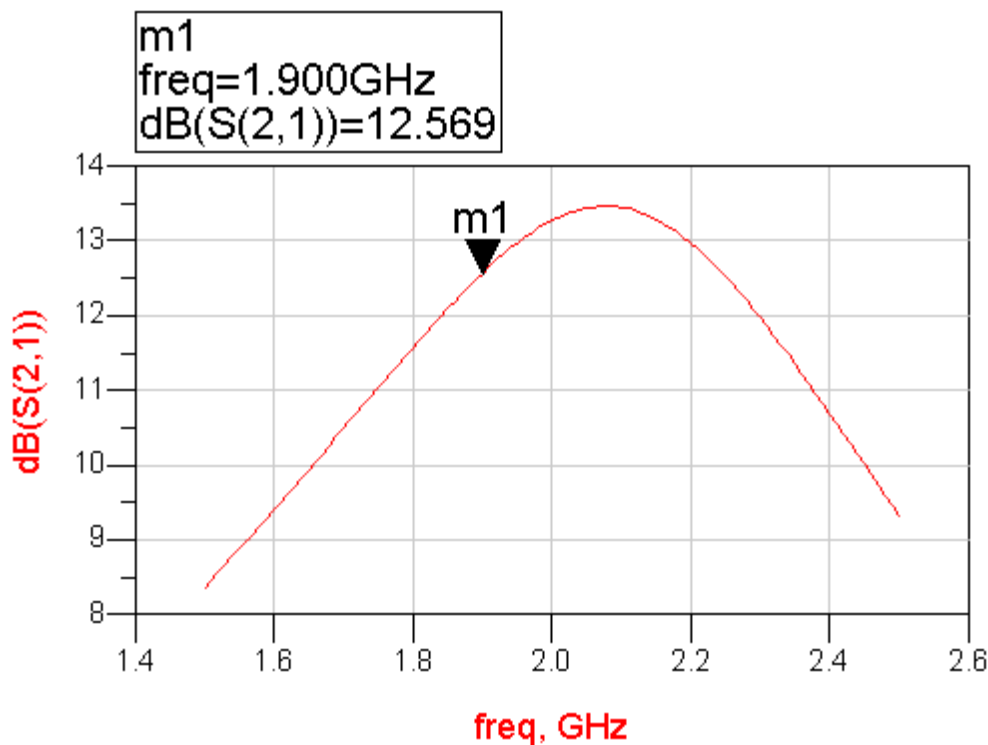
Tuning Cadence Instance Parameters and Design Variables

The ADS tuning capability enables you to change one or more design parameter values and see its effect on the output without re-simulating the entire design again from the beginning. Dynamic Link extends the ADS tuning function to instance parameters and design variables in Cadence subcircuits. For more information on using the ADS tuning feature, refer to Tuning in the["../optstat/wwhelp.htm" Tuning, Optimization, and Statistical Design] documentation.

This section uses the PowerAmp example used in [Getting Started Tutorial](#) for demonstrating tuning of Cadence instance parameters and design variables.

A Dynamic Link For Cadence Tuning Example

1. Follow the steps listed in [Performing an S-parameter Simulation](#) in the [Getting Started Tutorial](#). Plot S(2,1) in the ADS Data Display window after ADS S-parameter simulation is completed. The S(2,1) plot should resemble the following figure.

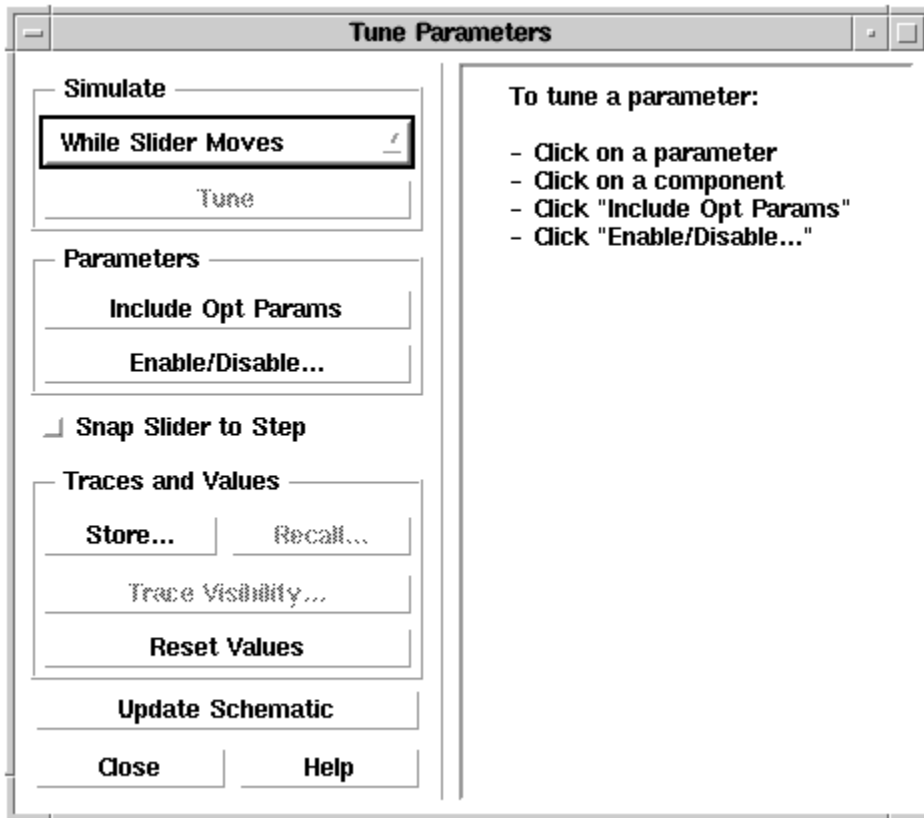


Data Display with S(2,1) of the PowerAmp_test Design.

2. In the ADS Schematic window, choose Simulate > Tuning or click the Tune Parameters icon (tuning fork) in the toolbar.



- Wait for the initial analysis to complete. The Tune Parameters dialog box appears as shown in the following figure.

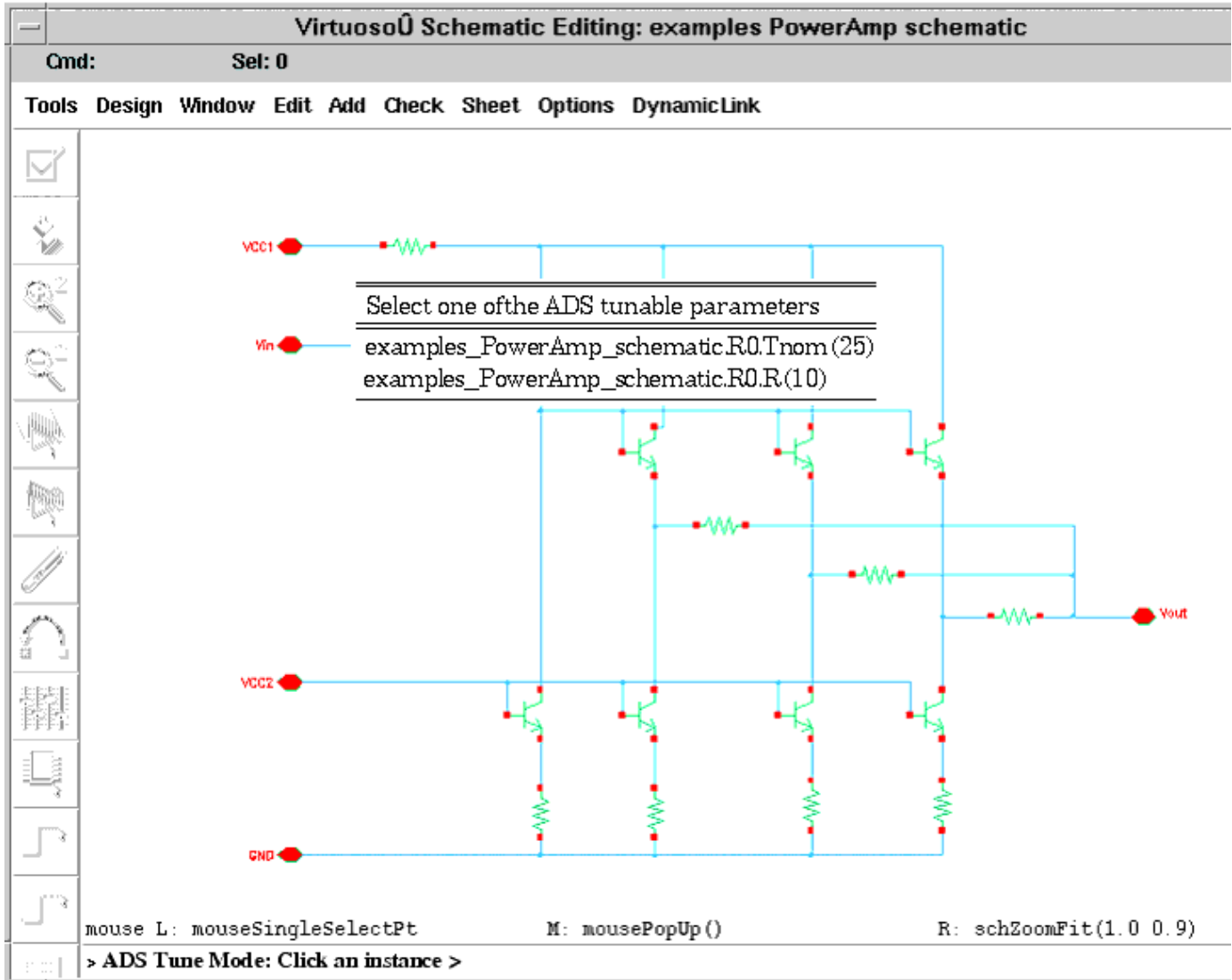


The Initial ADS Tune Parameters Dialog Box.

Note that the prompt at the bottom of the Cadence schematic window says:

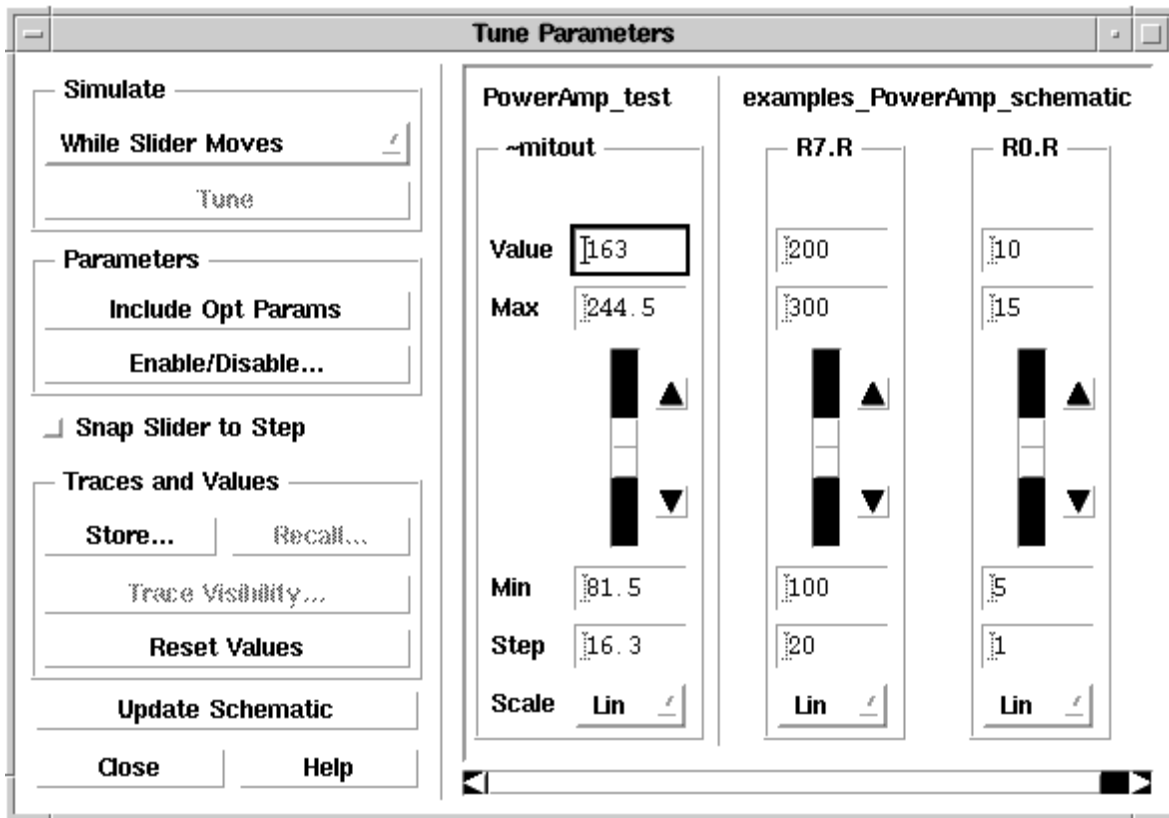
ADS Tune Mode: Click an instance>

- Move your cursor over resistor R7 in the Cadence schematic window and click the left mouse button (left click). A slider control for R7.R immediately appears in the ADS Tune Parameters dialog box. Note that only integer and floating point parameters are tunable. R7 contains only one tunable parameter R (r in Cadence spectre), the resistance.
- Left click resistor R0 in the Cadence schematic window. A menu pops up beneath R0 as illustrated in the following figure. The popup menu is displayed because there is more than one tunable parameter in R0 , R and Tnom . Cadence instance parameters are sent to ADS one at a time.



Select Tunable Parameter Pop-up Menu in the Cadence Virtuoso Schematic Window.

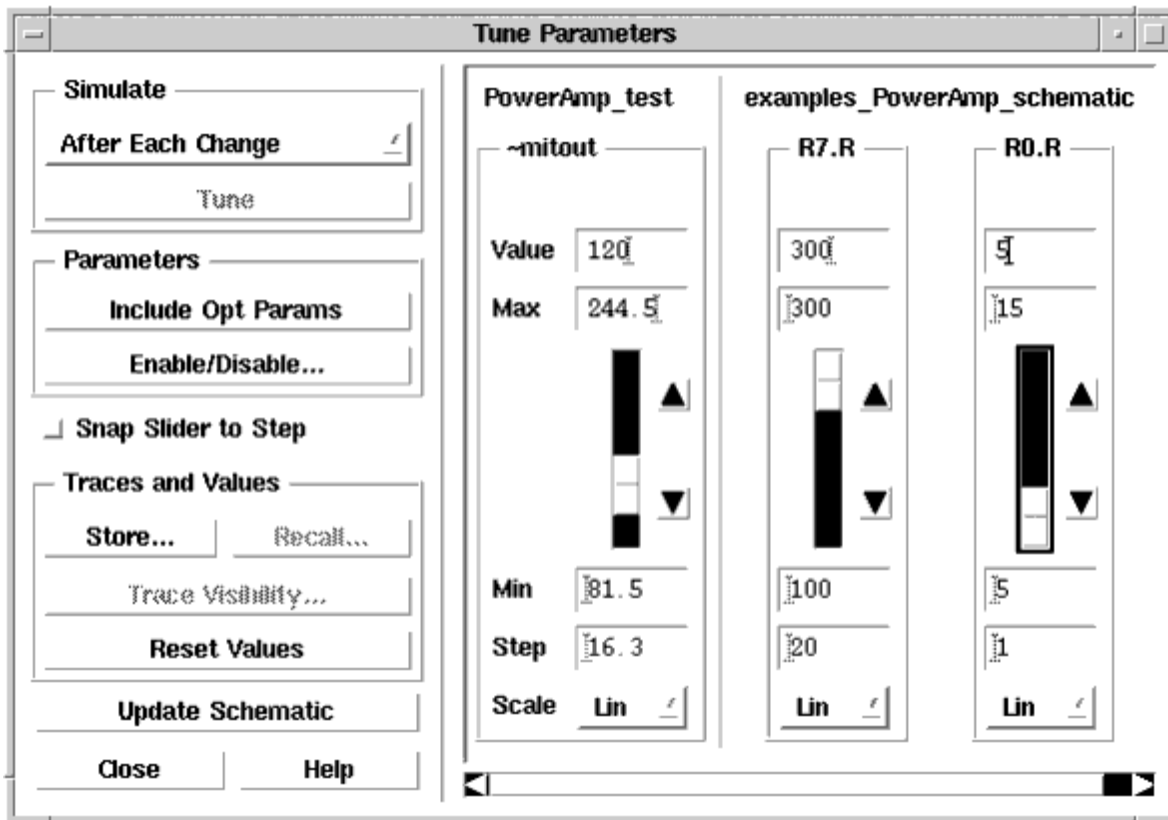
6. Select examples_PowerAmp_schematic.R0.R(10) in the pop-up menu. This creates a slider control for R0.R in the ADS Tune Parameters dialog box.
7. In the ADS Schematic window, click the design variable Remitout in the VAR1 block. The following figure shows the ADS Tune Parameters dialog box with VAR1.Remitout , R7.R , and R0.R being selected. Recall that VAR1.Remitout is a design variable originated from the Cadence subcircuit.



ADS Tune Parameters Dialog Box

Notice the distinction that a Cadence design variable is selected directly from the VAR1 block in the ADS Schematic window, while an instance parameter is selected from the Cadence Schematic window.

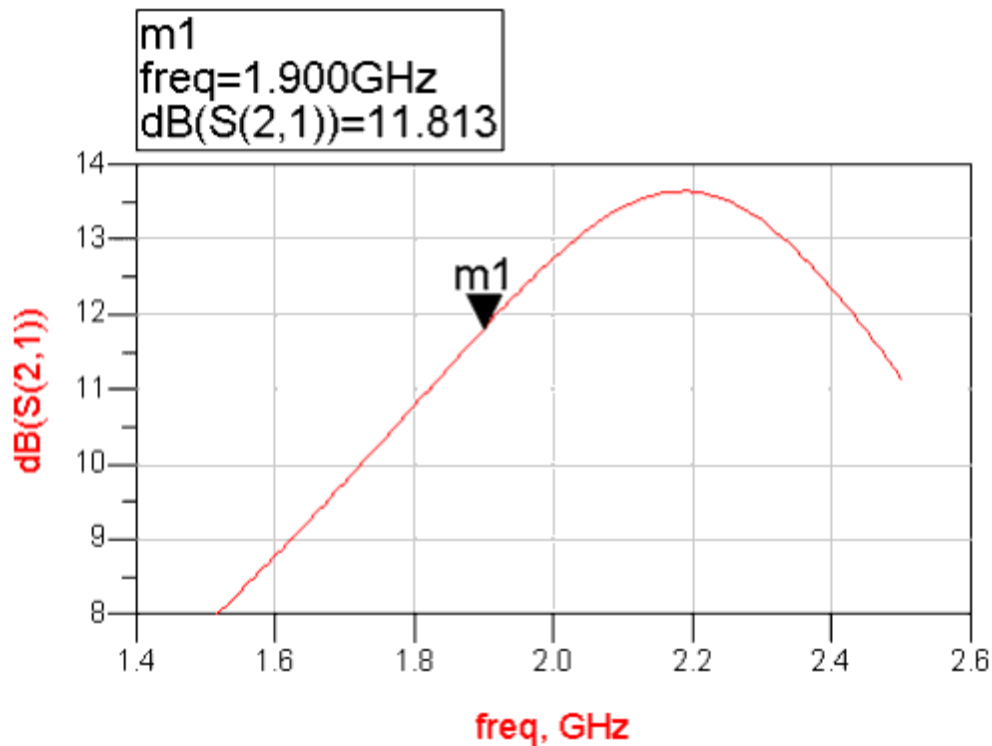
8. In the ADS Tune Parameters dialog box, select a tune analysis mode from the Simulate drop-down list. This tells ADS when you want tuning to occur. For this example, choose After each change if not already selected. After you finish with all of the steps in this example, try each tuning analysis method (After Each Change , After Pressing Tune , While Slider Moves) to see which one works best for you.
9. Drag the slider for examples_PowerAmp_schematic.R7.R to 300 (Ohms). You also can change the tunable parameters by doing the following:
 - Click the up or down arrows.
 - Type the value directly into the box.
 Observe the results in Data Display window each time you release the mouse button after dragging a slider or changing a value in the Tune Parameters dialog box.
10. Drag the slider for examples_PowerAmp_schematic.R0.R to 5 (Ohms).
11. Finally, enter a value for PowerAmp_test.VAR1.Remitout of 120 (Ohms). The Tune Parameters dialog box should now appear similar to the following figure.



ADS Tune Parameters Dialog Box (With VAR1.Remitout , R7.R , and R0.R values Changed).

The next figure shows the resultant $S(2,1)$ curve displayed in the Data Display window after changing VAR1.Remitout , R7.R , and R0.R in the Tune Parameters dialog box. You can compare this to the initial results in the figure [Data Display with \$S\(2,1\)\$ of the PowerAmp_test Design](#).

12. You can click the Reset Values button to restore all the sliders to their original values. The Update Schematic button in the Tune Parameters dialog box enables you to write the instance parameter values currently displayed in the dialog box into the Cadence Schematic window. For Cadence design variables, such as VAR1.remitout , you still need to select DynamicLink > Design Variables > Update Design Variables to Cellviews in the ADS schematic window and then follow the instruction in the Confirmation Message form to complete the update. Because there is no undo function for the Update operation, do not click the Update Schematic button if you do not want to change values in the Cadence subcircuit.



Results of S(2,1)

(as R7.R , R0.R and VAR1.Remitout are changed in the Tune Parameters dialog box)

- Click the Close button in the Tune Parameters dialog box to end the tuning session. The prompt at the bottom of the Cadence Virtuoso schematic window returns to its default greater than sign (>).

Note
For more information on using the ADS tuning feature, refer to [Tuning in Advanced Design System](#) in [Tuning, Optimization, and Statistical Design](#).

This example demonstrated three types of tuning operations in Dynamic Link:

- Clicking a Cadence instance with a single tunable parameter causes that parameter to be sent to ADS for tuning.
- Clicking a Cadence instance with multiple tunable parameters results in a pop-up menu being displayed. Selecting an item from the pop-up menu causes the parameter associated with that menu item to be sent to ADS for tuning.
- After obtaining Cadence design variables by selecting DynamicLink > Design Variables > Get Design Variables in the ADS schematic window, clicking a Cadence design variable in the ADS VAR1 block sends that variable for tuning.

All the above three types of operation act like a toggle switch. Selecting an item already on the Tune Parameters dialog box removes it from the dialog box.

You can descend down a Cadence design hierarchy to find an instance parameter for tuning. You can then return to higher level Cadence design hierarchy to select another instance parameter.

Note
During the tuning operation, the left mouse button in the Cadence Virtuoso schematic window is mapped to a Dynamic Link SKILL procedure. Do not bind any function to the left mouse button during this period. Any bindkey function previously mapped to the left mouse button will not work until the tune mode ends.

If you have a problem while tuning and need to discontinue the operation, enter `IdfMpsTuneEnd` in the Cadence CIW input area. This will end the Dynamic Link Tune Mode operation.

Optimizing Designs

Performance optimization enables you to specify a range of device or component values. The software can then automatically compute the nominal values that best meet your performance goals or specifications. A family of optimizers come with ADS, each with a different mathematical effect or use. For more information on performance optimization, refer to Performing Nominal Optimization in the["../optstat/wwhelp.htm" Tuning, Optimization, and Statistical Design] documentation.

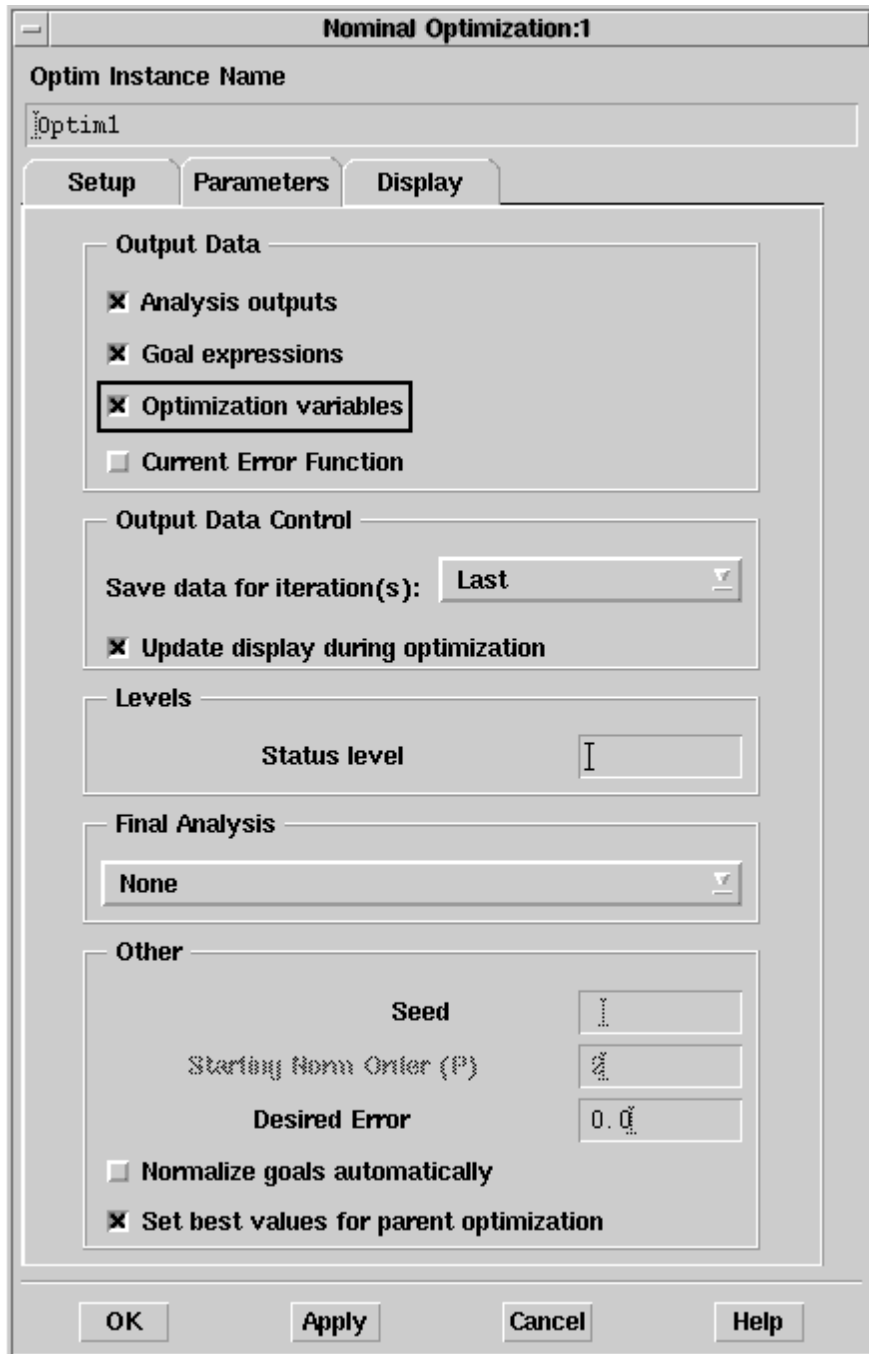
To optimize a design in the Advanced Design System:

1. In the Schematic window containing the design you want to optimize, choose Optim/Stat/Yield/DOE from the component palette. ** The Optim/Stat/Yield/DOE palette is displayed.

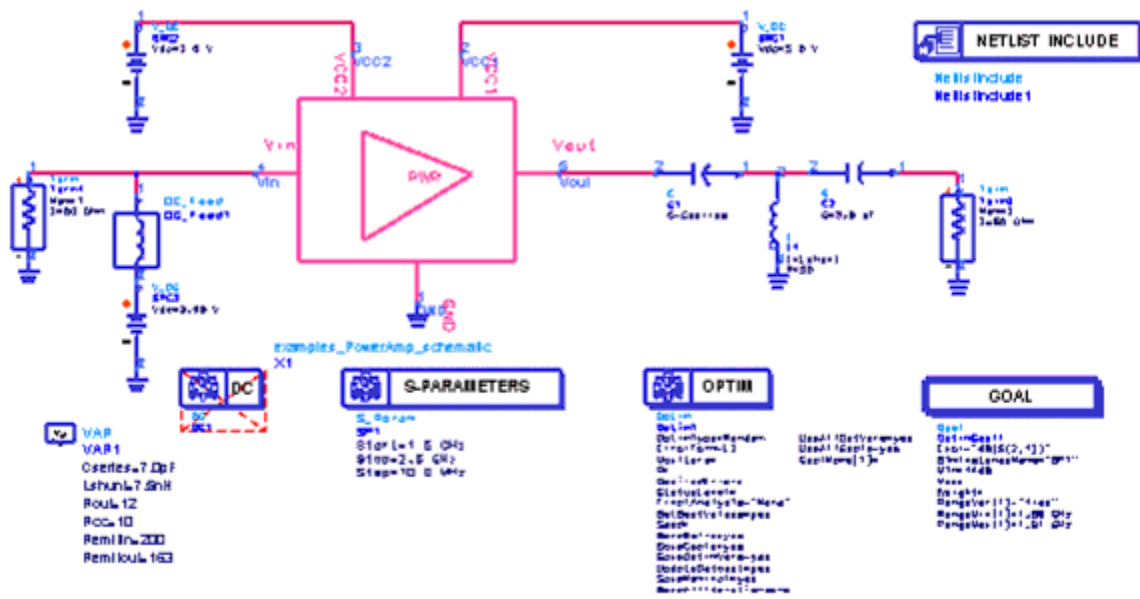


The Optim/Stat/Yield/DOE Component Palette

2. Set the options (Goal, Nominal Optimization, etc.) as desired. When a Nominal Optimization component is first added, you need to enable the output to the dataset. Click the Nominal Optimization component and choose Edit > Component > Edit Component Parameters . ** This brings up a Nominal Optimization dialog box. Select the Parameters tab and ensure that the Analysis outputs and Optimization variables are checked. Click OK .



3. Proceed with the Advanced Design System optimization.



An Optimization Setup

Statistical Variations

Both the Monte Carlo and Yield simulation controllers include options for specifying statistical variations. Note that process and mismatch variations are only used for ADS RFIC Dynamic Link.

Statistical Variation Options

option	Description
stat{...}	ADS syntax for statistical variations.
process †	For process only analysis, select the process checkbox and unselect mismatch checkbox. For process and mismatch analysis, select both the process and mismatch checkboxes.
mismatch †	For mismatch only analysis, unselect the process checkbox and select mismatch checkbox. For process and mismatch analysis, select both the process and mismatch checkboxes.

† These options use Spectre syntax and are provided only for use with RFIC Dynamic Link. For more information on process and mismatch variations, refer to the section on Monte Carlo Analysis in Chapter 6 of the Cadence _Virtuoso Spectre Circuit Simulator User Guide, Product Version 5.1.41_ .

Updating the Cadence Cellview

Once the optimum value of a variable is computed by an ADS simulation, you can update the value to the Cadence cellview. To update the Cadence cellview:

1. In the Schematic window, choose DynamicLink > Design Variables > Update Design Variables to Cellviews .

Note
Only the nominal values of variables get updated to Cadence; any range values are ignored. Variables to be optimized should not be assigned a value in Cadence; they may be assigned a nominal value and a range in ADS.

Using Additional Features of RFIC Dynamic Link

This chapter describes some of the additional features provided by the RFIC Dynamic Link. Some of the issues related to compatibility between Advanced Design System and the Cadence tools are also discussed in this chapter.

Freezing Selected Subcircuits

The RFIC Dynamic Link Freeze mode enables you to keep Cadence from generating a new netlist each time you simulate in Advanced Design System. This helps to avoid unnecessary time-outs caused by re-netlisting a large Cadence subcircuit.

Setting the Freeze Parameter

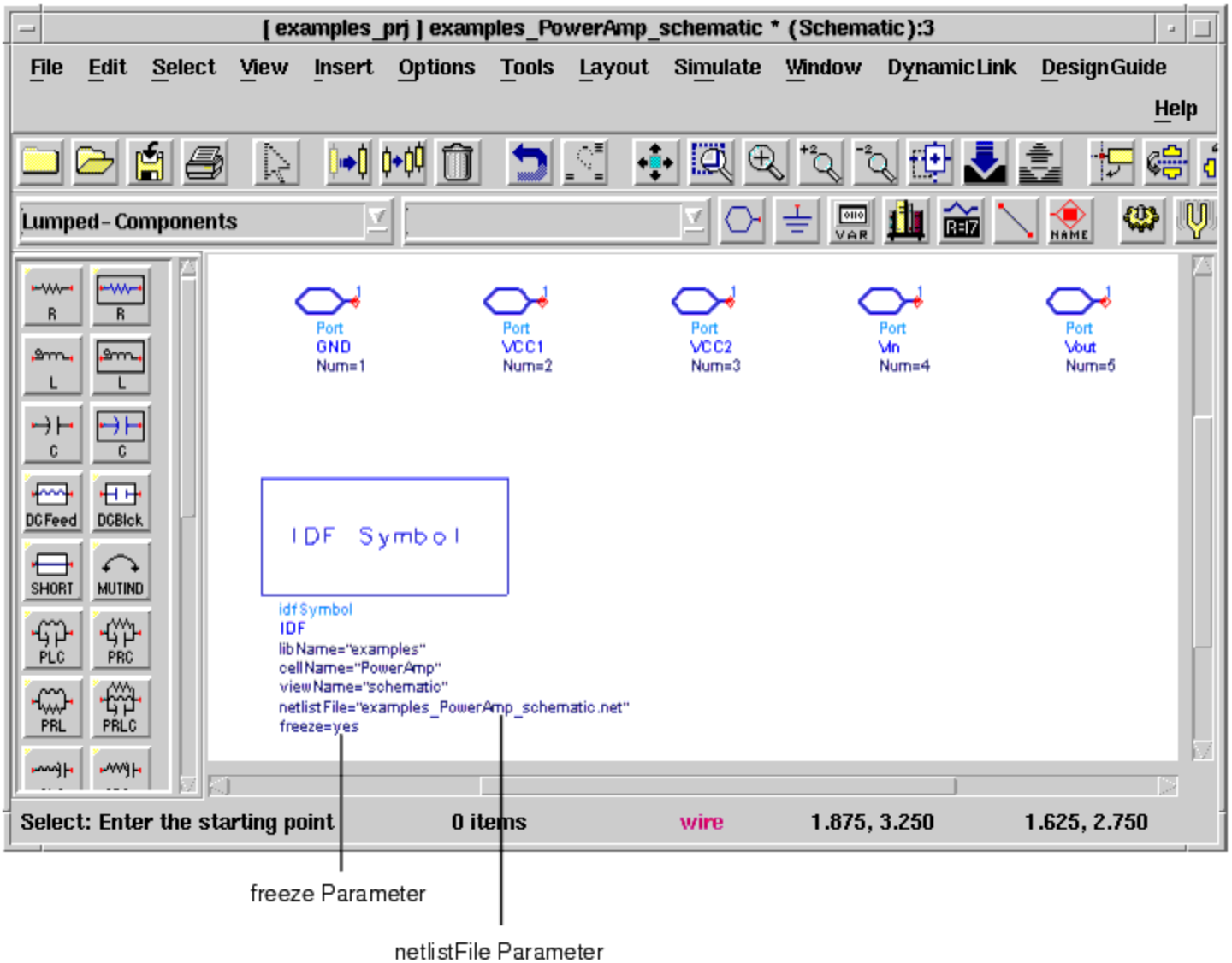
If you want to keep a Cadence Cellview from being netlisted and a netlist already exists:

1. Edit the ADS dummy design for that Cellview which would have a name of the form < lib >< _cell >< _view >.dsn. In this example, the name of the file is examples_PowerAmp_schematic.
2. Double click the _idfSymbol_ in the schematic dummy design (see the following figure) and set the freeze parameter to "yes" or "TRUE". Note that the idfSymbol contains the library name, cell name, view name and netlist file information



Freeze Parameter Setting in idfSymbol

Refer to the example schematic in the following figure for the location of the freeze and netlistFile parameter settings. To turn the freeze parameter off, set the parameter to "no" or "FALSE". The default value for the freeze parameter is "FALSE".



Defining the Freeze Parameter

To freeze all Cadence subcircuits, see [Modifying the Configuration File](#).

Generating a Cadence Subcircuit Netlist

In order to run Dynamic Link in Freeze mode, the Cadence subcircuit netlist must exist. If a Cadence subcircuit netlist does not exist, you can generate a new Cadence subcircuit netlist before running your ADS Simulation. Choose

DynamicLink > Subcircuit Netlist from your Cadence Schematic window.

Note
If the Dynamic Link pull-down menu does not appear in the Cadence Virtuoso Schematic window, choose Tools > ADS Dynamic Link > Add Dynamic Link menu to all schematic windows in the Cadence Command Interpreter Window (CIW).

This generates the Cadence subcircuit netlist. You can now run your ADS Simulation in Freeze mode.

Note
If you set the Freeze parameter to TRUE but the Cadence subcircuit has never been netlisted, the Cadence subcircuit will automatically be netlisted the first time an ADS Simulation is attempted.

Setting the netlistFile Parameter

The netlistFile parameter is used to specify the location of the ADS netlist of a frozen Cadence Cellview. On the dummy (placeholder) design, set the netlistFile parameter to point to the appropriate netlist file. For example:

```
netlistFile="examples_PowerAmp_schematic.net"
```

The default location for storing the ADS netlist of a frozen Cadence Cellview is:

```
<current_ADS_project_directory> /networks/
```

If you want to copy the netlist file elsewhere, set the netlistFile parameter to point to the full path and file name of the new location. For example:

```
netlistFile="/tmp/my_design.net"
```

Refer to the example schematic in [Defining the Freeze Parameter](#) for the location of the netlistFile parameter setting.

Using "Freeze" Mode to Simulate a Design in ADS Standalone

You can run Advanced Design System standalone (without Cadence DFII or RFIC Dynamic Link) using a frozen netlist from an earlier RFIC Dynamic Link session (see [Freezing Selected Subcircuits](#)). The parameter Freeze=TRUE must be set on all dummy designs in ADS that represent Cadence cellviews.


Advanced Design System 2008

While RFIC Dynamic Link is not designed to operate on a PC, you can take an ADS netlist (created using Dynamic Link on your UNIX workstation) of your Cadence cellview and copy it to your PC for an ADS simulation. This type of operation is typically done for board design. Once you have done some minor configuration, you can then add simulation and control components externally to your design and resimulate on the PC.

To set up and simulate your Cadence cellview in Advanced Design System on a standalone Windows machine:

1. Install RFIC Dynamic Link on your PC.
2. For each Cadence CellView, copy the netlist, AEL and design files into your working project directory's networks subdirectory. For instance, copy the following example PowerAmp schematic files into your project directory:

```
examples_PowerAmp_Schematic.net  
examples_PowerAmp_Schematic.ael  
examples_PowerAmp_Schematic.dsn
```

3. If the Cadence CellView contains design variables, you will need to manually enter them into a VAR block in the top level ADS schematic. To do this:
 - Choose Data Items from the Component Pallet.
 - Click the Var Eqn block to add the component and use the cursor to place an instance on the schematic. You may continue placing more instances of the Var Eqn block, or choose the Cancel Command And Return To Select Mode icon to proceed with the next step. 
 - Enter the appropriate values into Var Eqn block.

It is important to note that any changes made to the design on your standalone machine will not be reflected in your original Cadence cellview. While you may add simulation and control elements externally, the fundamental design should not be changed if you want it to match your original Cadence design.

Compatibility between Advanced Design System and Cadence Tools

Some of the features provided by the RFIC Dynamic Link include support for compatibility issues related to differences between Advanced Design System and the various Cadence tools. This section addresses several of these compatibility issues.

Support for Duplicate Pin Names

It is typical for the top (chip) level schematic to have multiple pins for the same signal, usually power and ground connections. The netlister lists duplicate I/O ports only once in the subnetwork definition and likewise for the nets connected to an instance of the subnetwork. However, the netlister in ADS (which does the top-level netlisting), writes out the multiple connections to ports with the same name, causing a conflict to be reported by the ADS simulator

while parsing the final netlist. To eliminate this conflict, when the symbol generator encounters duplicate pin names, it draws only one pin with a given name and issues a warning message. However, duplicate pins at lower levels in the Cadence schematic hierarchy are allowed, because no ADS symbol is involved.

Using Buses

For an example on using buses in RFIC Dynamic Link, refer to [Getting Started Tutorial](#) and use the PowerAmp2 Cadence cellview in place of the PowerAmp cellview. Then use the PowerAmp2_test ADS design in place of the PowerAmp_test design. The PowerAmp2 example is the same as the PowerAmp example except that it uses bus wires.

For details on creating buses in ADS, refer to Chapter 3 of the ["../usrguide/wwhelp.htm"Schematic Capture and Layout] documentation. __ For details on using buses within Cadence Design Framework II, refer to your Cadence documentation.

Setting up Unnamed Nets

In ADS, unnamed nets begin with an _net prefix followed by an integer. All other net value are written out to the output dataset during simulation. By default, Cadence tools use the prefix net followed by an integer. By default, the dataset can get very large. To avoid this, set the Net Name Prefix in the Cadence schematic to _net instead of net . To set the default Net Name Prefix in the Cadence schematic:

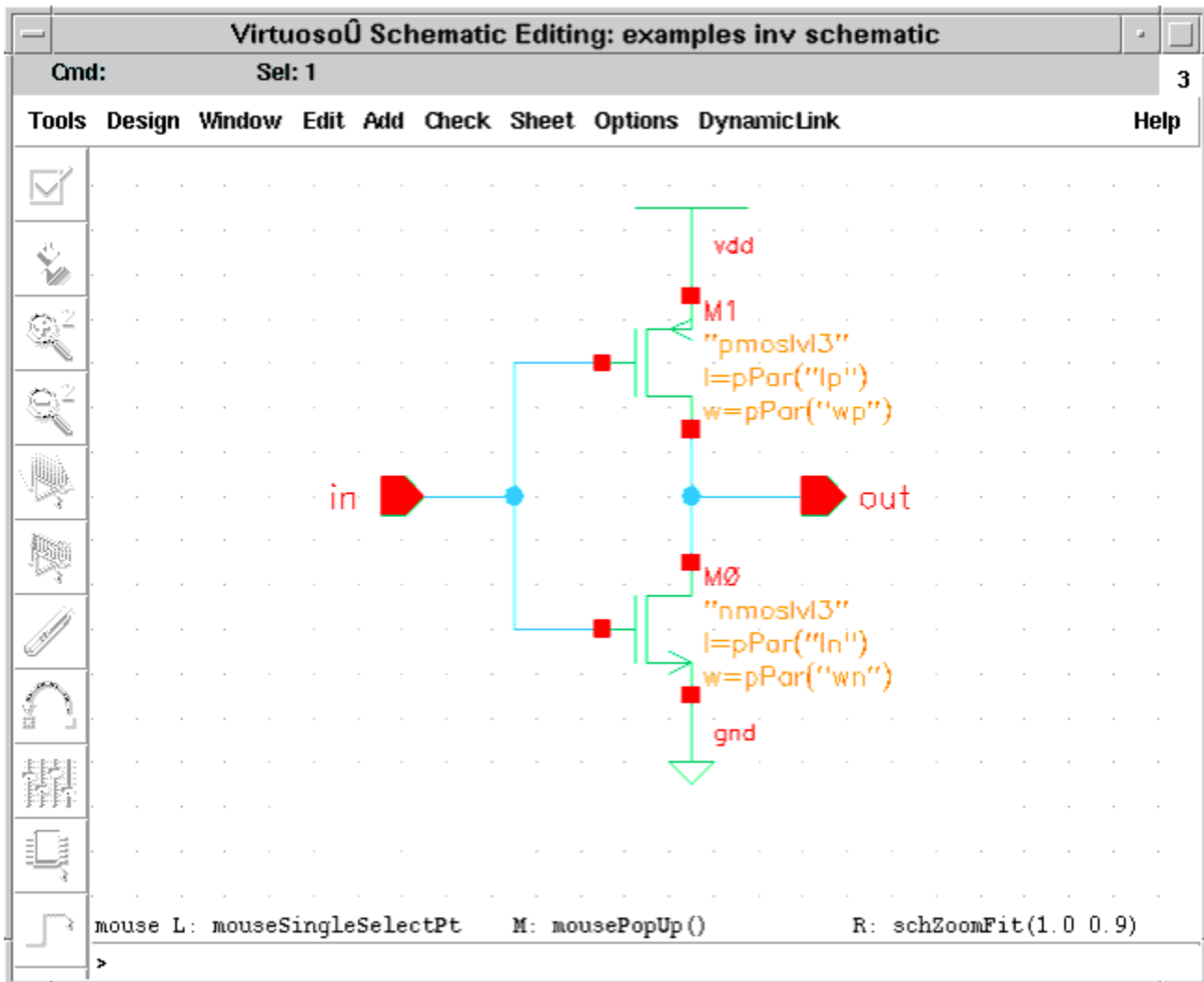
- Choose Options > Editor .
- In the Editor Options dialog box, enter _net in the Net Name Prefix field
- Click OK .
- Choose Design > Check and Save to save each related Cadence schematic.

Support for pPar and iPar

This section describes the general use of parent parameters (pPar) and instance parameters (iPar).

pPar()

The following figure shows an example of an inverter design that contains two CMOS transistors (M0 & M1).



Composer Schematic showing use of pPar()

This Composer circuit contains instances whose parameters are defined in terms of parent parameter values using pPar() . The parameters in this case are defined as,

$l = \text{pPar}("ln")$

$w = \text{pPar}("wn")$

for M0, and

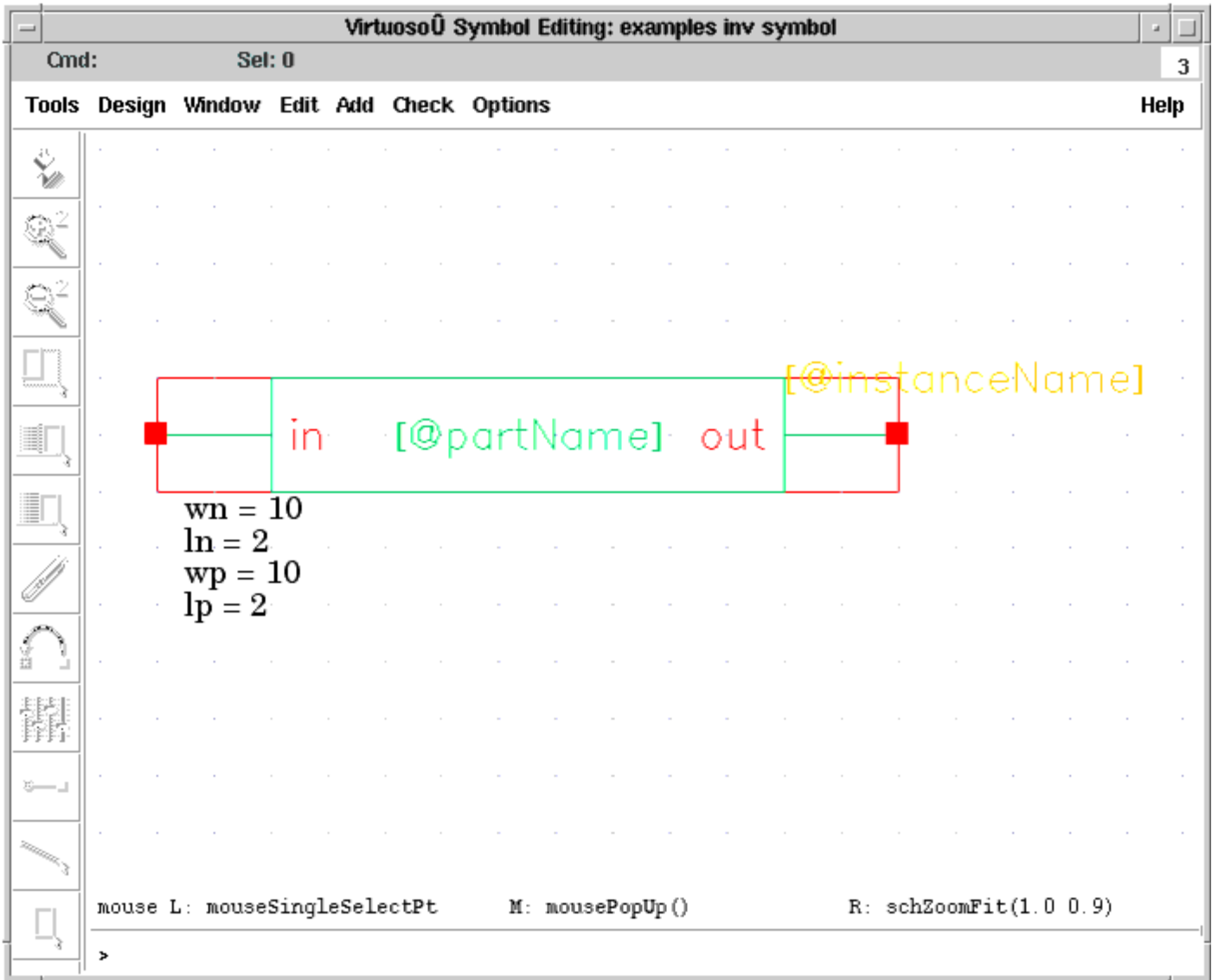
```
l = pPar("lp")
```

```
w = pPar("wp")
```

for M1.

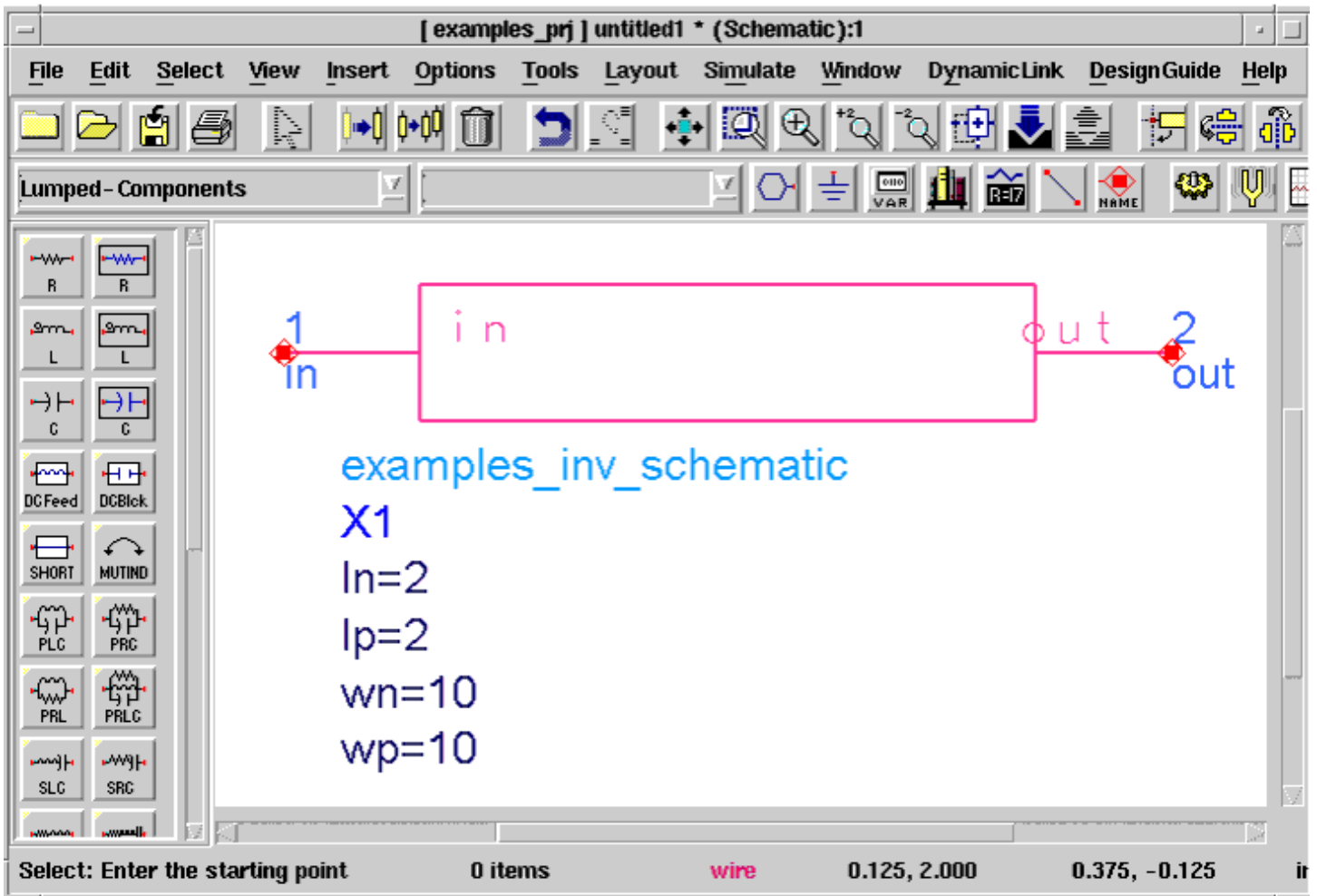
This inverter circuit also has an associated symbol view in Cadence Composer. The symbol view shown in the following figure is equivalent to a black box that displays the input, output and instance properties for the circuit in the previous figure.

The default values of w_n , l_n , w_p and l_p are displayed in the symbol view along with the associated symbol for the device.



Composer Symbol View showing default values for pPar()

The Composer symbol is instantiated in ADS via the Dynamic Link where the instance properties also appear in ADS Schematic (see the following figure).



Inherited Symbol in ADS

The parameter values are reflected in the netlist that is sent to the simulator. These values can be viewed and edited using the Cadence Edit Component CDF form (see the following figure).

If the Cadence menu option Design > Create Cellview > From Cellview is used, the CDF for the schematic will be set up automatically. The Cadence software will traverse the hierarchy looking for pPar statements and automatically generate parameters. It will also set up netlisting data for known simulators.

If you are modifying an existing schematic and you have already generated a symbol for your schematic, it may be necessary to manually add the netlisting data for ADS. If this is the case, do the following:

1. Go to the Simulation Information section of the Edit Component CDF form (see the following figure) and click Edit

Edit Component CDF

CDF Selection **Cell** **Library** **CDF Type** Base

Library Name examples

Cell Name inv

File Name

Component Parameters

Name	Add	Edit	Move	Select To Delete
In	2			┘
lp	2			┘
wn	10			┘
wp	10			┘

Simulation Information

ads "in" "out"

Edit Component CDF Form

- When the Edit Simulation Information dialog (see the following figure) appears, change the Choose Simulator setting to "ads".

Edit Simulation Information

OK Cancel Apply Help

Choose Simulator **ads**

netlistProcedure IdfSubcktCall

otherParameters

instParameters **wn ln wp lp**

componentName subcircuit

termOrder "in" "out"

termMapping

propMapping

typeMapping

uselib

Edit Simulation Information Dialog

3. In the netlistProcedure field enter IdfSubcktCall .
4. In the instParameters field enter the parameters you wish to have netlisted for ADS. For the inv circuit, this means entering "wn ln wp lp". The parameter order does not matter.
5. Change the componentName field entry to "subcircuit" or leave it blank.
6. Set the termOrder field to the order you wish to netlist the terminals in for ADS. You should have one entry for each terminal on your design. For the inv circuit, this is set to "in" "out". You can also quote the names, but it is not necessary.
7. If you wish to back annotate currents, make the appropriate entry in the termMapping field. This is done by specifying the name of a terminal, followed by the ADS pin number it will be. For the inv example, termMapping entry would be "nil in ": P1" out ": P2"".

For more information on editing simulator information, refer to [Creating the Netlist Interface](#) in the [Cadence Library Integration](#) documentation.

iPar()

Similarly, the Dynamic Link supports the use of iPar . For any given instance, you can define an instance parameter as

a function of another parameter of the same instance. For example, if the parameter w in the figure [Composer Schematic showing use of pPar\(\)](#) were defined as $w=2*iPar("l")$, then if $l=10$, then $w=20$.

Using Inherited Connections

Inherited connections used in Dynamic Link must all be resolved within the Cadence hierarchy. For example, if you create a schematic in Cadence called `test`, that contains instances that have inherited connections with them, such as `nmos` in `analogLib`, the default connectivity is used in `test` if no netset properties have been placed on instances in the hierarchy of the top level circuit. If you have hierarchy above the top level Cadence schematic placed in the ADS design environment, you cannot place netset properties on those instances.

Using S-parameter File Devices from analogLib

For information on S-parameter file components in the `analogLib` library, refer to [Using S-parameter File Devices from analogLib](#) in the [analogLib Components](#) documentation.

Encoding Cadence Designs for RFIC Dynamic Link

This example demonstrates the use of Advanced Design System's RF IP Encoder command-line interface with RFIC Dynamic Link. The Cadence and ADS designs provided in the RFIC Dynamic Link Tutorial are adopted here. The PowerAmp Cadence design and the ADS model files that are used in the design will be encoded using the RF IP Encoder, as a component in an ADS library. The encoded component will eventually be used in the `PowerAmp_test` ADS design for a simulation. For more information, refer to the [RF Intellectual Property Encoder](#) documentation.

Familiarity with the Cadence environment and the RFIC Dynamic Link tutorial is assumed. The RFIC Dynamic Link tutorial is provided in [Getting Started Tutorial](#).

The following is a list of steps to encode a Cadence design and then use the encoded design in an ADS simulation via RFIC Dynamic Link:

1. Make a copy of the Dynamic Link examples directory and start Cadence from the directory created. For example, enter the following commands:

```
cp -r $HPEESOF_DIR/idf/examples /tmp
cd /tmp/examples
icms&
```

See [Setting up the Examples Directory](#) for details.

2. Open the PowerAmp cellview of the examples library in the Cadence schematic window.
3. Choose Tools > ADS Dynamic Link menu item in the Cadence schematic window to start ADS RFIC Dynamic Link.
4. Choose DynamicLink > Subcircuit Netlist menu item in the Cadence schematic window to generate the netlist for the PowerAmp Cadence design.
5. Choose File > Save As in the netlist window and save the netlist file as examples_PowerAmp_schematic . For demonstration purpose, it is more convenient to name the netlist file exactly as examples_PowerAmp_schematic here, i.e. name the netlist file as <libName> _ <cellName> _ <viewName> with no extension.
6. Close the Cadence schematic window and the netlist window.
7. Open the examples_PowerAmp_schematic file in an ACSCII text editor.
8. Insert the two model files examples/models/nnpwa1.ads and examples/models/nnpwa2.ads between the end and the #endif statements in the examples_PowerAmp_schematic file as shown below. Save your changes and quit the editor.

```

...
end examples_PowerAmp_schematic
model npnpwa1 BJT NPN=1 PNP=0 Bf=130 Ikf=0.01085 \
  Ise=7.56E-13 Ne=2 Vaf=25 Nf=1.03 Tf=8.91E-12 \
  Xtf=3.35 Vtf=0 Itf=0.0217 Ptf=18 Xtb=2.2 \
  Approxqb=1 Br=5.123 Ikr=0.056 Isc=2.01E-12 \
  Nc=2 Var=0 Nr=1 Tr=1.6E-9 Eg=1.11 Is=2.15E-16 \
  Imax=1 Xti=8 Tnom=25 Nk=0.5 Iss=3.28E-13 Ns=1 \
  Cjc=1.045E-13 Vjc=0.75 Mjc=0.5 Xcjc=0.2292 \
  Fc=0.5 Cje=1.935E-13 Vje=0.85 Mje=0.4 \
  Cjs=1.092E-13 Vjs=0.7 Mjs=0.5 Rb=317 Irb=0 \
  Rbm=10.63 Re=1.79 Rc=37.6 Af=1 Ab=1 \
  Fb=1 Ffe=1 Lateral=0
model npnpwa2 BJT NPN=1 PNP=0 Bf=166 Ikf=0.02325 \
  Ise=1.218E-12 Ne=2 Vaf=25 Nf=1.03 Tf=7.97E-12 \
  Xtf=3.16 Vtf=0 Itf=0.0465 Ptf=18 Xtb=2.2 \
  Approxqb=1 Br=5.123 Ikr=0.12 Isc=4.08E-12 \
  Nc=2 Var=0 Nr=1 Tr=1.6E-9 Eg=1.11 Is=4.08E-16 \
  Imax=1 Xti=8 Tnom=25 Nk=0.5 Iss=4.96E-13 Ns=1 \
  Cjc=2.13E-13 Vjc=0.75 Mjc=0.5 Xcjc=0.22 \
  Fc=0.5 Cje=3.123E-13 Vje=0.85 Mje=0.4 \
  Cjs=1.65E-13 Vjs=0.7 Mjs=0.5 Rb=59.2 Irb=0 \
  Rbm=14.1 Re=0.833 Rc=22 Af=1 Ab=1 \
  Fb=1 Ffe=1 Lateral=0
#endif

```

9. Make a temporary working directory. For example, enter

```

mkdir /tmp/myip
cd /tmp/myip

```

10. Copy the netlist fragment to be encrypted to the temporary directory. Make sure the filename is the name that will be used to get the fragment out of the library and it does not contain an extension. For example, enter

```

cp /tmp/examples/examples_PowerAmp_schematic /tmp/myip/

```

11. Create a new file named 'library.cfg' that contains the following three lines:

Advanced Design System 2008

```
Library name: PowerAmp_schematic
Library size: 2
examples_PowerAmp_schematic examples_PowerAmp_schematic
```

12. Create another file named `ADSlibconfig' that contains the following line:

```
PowerAmp_schematic /tmp/myip/PowerAmp_schematic.library
```

13. Make sure LD_LIBRARY_PATH (Linux and SunOS) or SHLIB_PATH (HP-UX) is set to include shared libraries required to run ADS. For example, entering the following command at the Korn shell prompt will accomplish this:

```
. bootscripts.sh
```

14. Make sure PowerAmp_schematic.library does not already exist by entering:

```
rm PowerAmp_schematic.library
```

15. Enter the `hpeesofencode' command to create PowerAmp_schematic.library.

16. Edit or create one of three ADSlibconfig files to add a line for mapping the library name to the full path to the .library file. The line to add for this tutorial should read:

```
PowerAmp_schematic /tmp/myip/PowerAmp_schematic.library
```

The library mapping line can be added to any of the three ADSlibconfig files listed below:

```
$HOME/hpeesof/circuit/config/ADSlibconfig
$HPEESOF_DIR/custom/circuit/config/ADSlibconfig
$HPEESOF_DIR/circuit/config/ADSlibconfig
```

17. Choose File > New > Cellview in the Cadence CIW to create a new examples_PowerAmp_schematic symbol in the examples library, that is, enter the following information in the Create New File form:

Library Name:	examples
Cell Name:	examples_PowerAmp_schematic
View Name:	symbol

Note that selecting Composer-Symbol in the Tool cyclic button will fill in symbol as the View Name.

18. In the Cadence Symbol Editor window, choose Add > Import Symbol to import the PowerAmp symbol in the examples library. Save the symbol and close the Symbol Editor window.

Advanced Design System 2008

19. Use the Cadence CDF Editor or `cdfDump()` and `load()` SKILL commands to enter the following ads simInfo (Simulation Information) for the `examples_PowerAmp_schematic` cellview:

netlistProcedure	ADSSimCompPrim
termOrder	(GND VCC1 VCC2 Vin Vout)
uselib	PowerAmp_schematic

Note that the library name entered in the `uselib` field must be exactly the same as the name entered in the `ADSlibconfig` file.

20. Create an ads view by copying the `examples_PowerAmp_schematic` symbol view to an ads view. One way is to press and hold the middle mouse button at the symbol in the Cadence Library Manager and select Copy in the popup menu. Enter ads in the View field of the To group in the Copy View form and click OK .
21. Create a cellview in the Cadence schematic window. For example, enter the following information in the Create New File form:

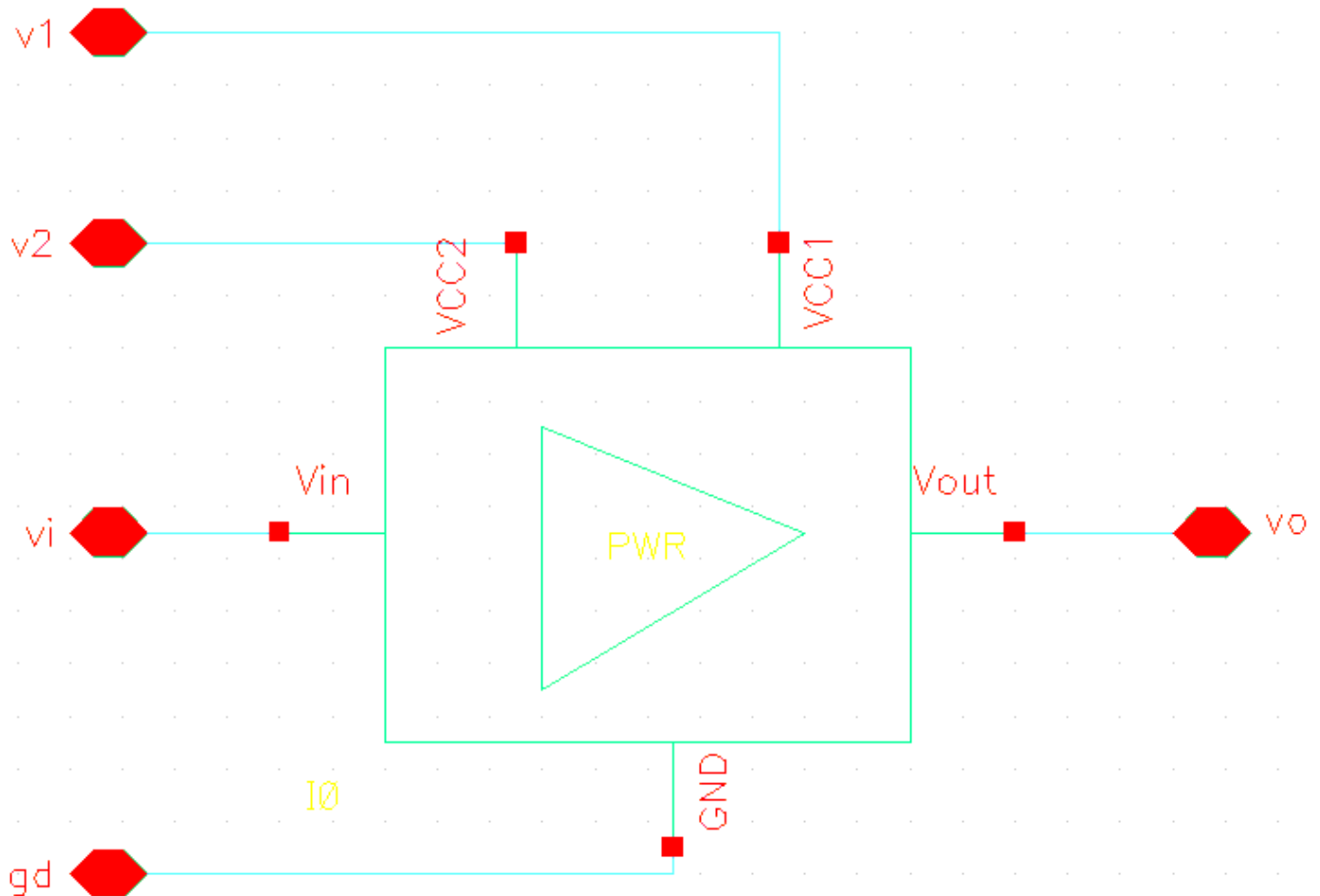
Library Name:	examples
Cell Name:	trojan
View Name:	schematic

22. Place an `examples_PowerAmp_schematic` instance in the schematic window. Add five Input/Output pins and connect one to each pin of the instance as shown below.

Virtuoso® Schematic Editing: examples trojan schematic

Sel: 0

Window Edit Add Check Sheet Options DynamicLink



23. Choose Tools > ADS Dynamic Link in the Cadence schematic window.

24. Choose DynamicLink > Design Variables in the Cadence schematic window in which the trojan cellview is displayed. Add the following four design variables:

Rout	12
Rcc	10
Remitin	200
Remitout	163

It may be necessary to click the Copy To button in the Design Variables form.

25. Choose Design > Create Cellview > From Cellview to create a symbol view for the examples/trojan/schematic cellview. Save the symbol created and close the Symbol Editor window.
26. Click the Check and Save icon in the schematic window.
27. Open the PowerAmp_test ADS design in the examples_prj ADS project.
28. Choose DynamicLink > Instance > Add Instance of Cellview to place the examples_trojan_schematic instance in the PowerAmp_test ADS design. Make an appropriate connection for each of the five pins as demonstrated in the Dynamic Link Tutorial. Make sure the v1 pin, which connects to the VCC1 pin of the examples_PowerAmp_schematic Cadence cellview, is connected to the SRC1 instance on the right side of the PowerAmp_test ADS design.
29. Choose DynamicLink > Design Variables > Get Design Variables to add the Rout, Rcc, Remitin, and Remitout Cadence design variables to the VAR1 block in the ADS schematic window.
30. Activate the DC and S_PARAMETERS simulation controllers and start an ADS simulation.

i Note
 Unlike in the Dynamic Link Tutorial, the model cards need not be included in a NetlistInclude component since they have already been included in the encoded PowerAmp_schematic library.

31. Plot S(2,1) to ensure the simulation results are the same as illustrated in the Dynamic Link Tutorial. If the simulation completed without error but the output plot looks different, check the pin order of the Cadence instance.
32. Choose DynamicLink > Top-level Design Netlist in the ADS schematic window and observe the following lines in the netlist file:

```
define examples_trojan_schematic \( gd v1 v2 vi vo \  

#{anchor:1109884:index:uselib}uselib "PowerAmp_schematic", "examples_PowerAmp_schematic"  

examples_PowerAmp_schematic:I0 gd v1 v2 vi vo  

end examples_trojan_schematic  

...  

examples_trojan_schematic:X1 0 net51 net50 net54 net52
```

These lines are equivalent to the combination of the subcircuit call, the subcircuit definition of the examples_PowerAmp_schematic, and the model card include statements in the netlist of the Dynamic Link Tutorial.

33. Choose DynamicLink > Close Connection to end the Dynamic Link session. Exit Cadence.

Encoding Process Summary

When selecting the DynamicLink > Subcircuit Netlist menu item in a Cadence schematic window, Dynamic Link defines the Cadence subcircuit name as <libName> _ <cellName> _ <viewName> , which can be renamed as desired in an editor. The key is that a Cadence component with the same name as the subcircuit, with an ads view and appropriate uselib, must be created. The name must contain an underscore character. The whole string with the underscore character will be the encoded component name. The sub-string after the underscore will be the name of the new ADS library that contains the encoded component.

If the Simulation > Netlist > Recreate menu item in the RF Design Environment (RFDE) Analog Design Environment is used to create the initial netlist, a subcircuit definition (define <subckt_name> and end statements) must be inserted

Advanced Design System 2008

and undesired netlist fragments such as, Options and outputPlan, must be removed.

The encoded design will not work in spectre or other simulators integrated into the Cadence design environment.

In summary, here is the list of tasks to perform for creating and using an encoded Cadence design with Dynamic Link or RFDE:

1. Create a subcircuit netlist for the Cadence design to be encoded. The subcircuit name must contain an underscore character. For the ensuing context, assuming ABC_XYZ is the subcircuit name.
2. Merge model cards to be encoded with the above netlist file and save the file as the same name as the subcircuit with no suffix, i.e. ABC_XYZ.
3. Create a `library.cfg' file in a temporary working directory with the following three lines:

```
Library name: XYZ
Library size: 2
ABC_XYZ ABC_XYZ
```

4. Create another file named `ADSlibconfig' under the same temporary directory with the following line:

```
XYZ _<full_path_to_temporary_directory>_ /XYZ.library
```

5. Create a Cadence symbol with as many input/output pins as the encoded subcircuit. This symbol should bear the same name as the subcircuit, i.e. ABC_XYZ.
6. Enter `. bootscript.sh` to set up shared library paths for ADS tools.
7. Enter ``hpeesofencode'` in the temporary directory to encode the ABC_XYZ netlist file into XYZ.library.
8. Copy the Cadence symbol view to create an ads view to indicate it being a primitive component in RFIC Dynamic Link and RFDE.
9. Add an ads simInfo section to the CDF of the Cadence component. Enter ADSsimCompPrim as the netlistProcedure, the encoded library name (i.e. XYZ) as uselib, and appropriate termOrder entries that match the subcircuit definition.
10. Create a Cadence subcircuit to encapsulate an instance of the above primitive component. This step is not necessary for RFDE.
11. Instantiate an instance of the Cadence subcircuit cellview in the ADS schematic window via RFIC Dynamic Link to use the encoded Cadence design in ADS; or, simply instantiate an instance of the Cadence primitive component (i.e. the ABC_XYZ symbol) to use the encoded Cadence design in RFDE.
12. To distribute the encoded design to RFIC Dynamic Link users, deliver the encoded library (XYZ.library), the primitive cellview (ABC_XYZ), and the Cadence wrapper subcircuit created for encapsulating the primitive cellview. For RFDE users, only the first two items are necessary. The original Cadence design and its associated model cards are protected in the encoded library.

Note that the user of the encoded library will need to add an entry with appropriate full path to define the encoded library in one of the user's three ADSlibconfig files listed below:

```
$HOME/hpeesof/circuit/config/ADSlibconfig
$HPEESOF_DIR/custom/circuit/config/ADSlibconfig
$HPEESOF_DIR/circuit/config/ADSlibconfig
```

Note

The contents of these ADSlibconfig files must contain only ASCII text. Some Windows applications may introduce invisible encoding characters without your knowledge. These non-ASCII encoding characters will cause a library name to be interpreted incorrectly resulting in a failure that may be very difficult to diagnose. It is highly recommended that you use an ASCII text editor such as vi or emacs to edit these files.

If you will deliver the library to users who will only use it on a PC, then instantiate the Cadence wrapper instance in an ADS schematic window via Dynamic Link, and then set its Freeze Mode to yes after generating its ADS netlist. You can create an ADS project with a design containing this instance and its design variables (in VAR1), place the encoded library under this ADS project, and then send this ADS project to your users. You will not need to send the Cadence library for PC (Dynamic Link Freeze Mode) users. The ADS design should contain:

```
<lib> _ <cell> _ <view> .dsn,  
<lib> _ <cell> _ <view> .ael, and  
<lib> _ <cell> _ <view> .net
```

Note that the <lib> _ <cell> _ <view> .net file is the Cadence wrapper circuit's Freeze Mode netlist.

Using Switch Views, Stop Views and the Hierarchy Editor

This chapter provides information on using switch views , stop views and the Hierarchy Editor . A switch view is a list that describes the views to use and their priority. A stop view is a list that designates when to stop moving down in hierarchy. In other words, if a view is in the switch view list, and also in the stop view list, it won't traverse any lower in hierarchy.

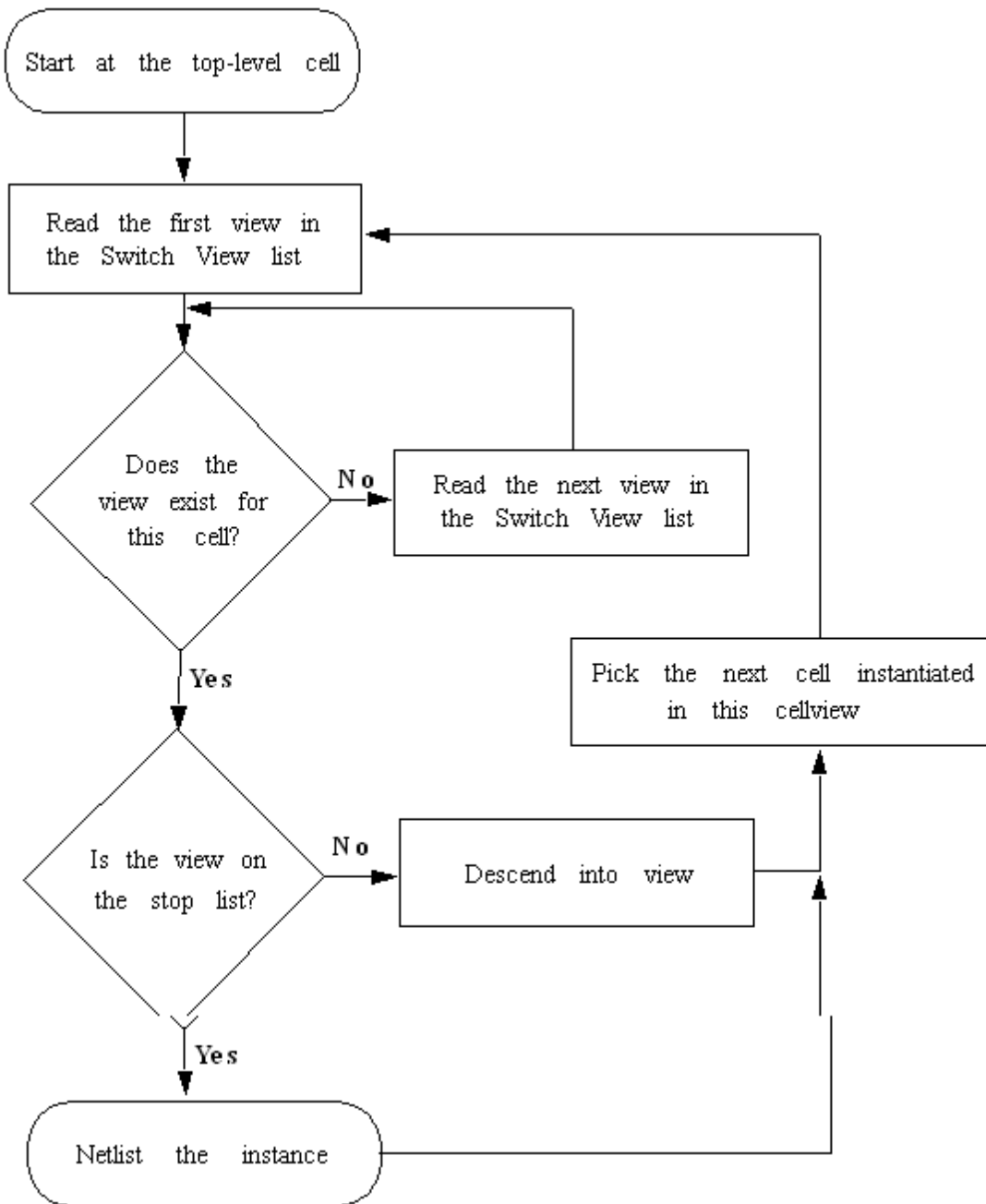
RFIC Dynamic Link supports the concept of using switch views and stop views . Dynamic Link also supports Cadence's Hierarchy Editor tool, which enables more detailed specification of switch views and stop views than the standard Artist forms. Switch views and stop views are utilized by the netlister to expand hierarchy. The Hierarchy Editor enables you to override the switch view list and stop view list for each instance in a schematic's hierarchy.

Note

The information provided in this chapter refers to using the Cadence Hierarchy Editor tool with the RFIC Dynamic Link and Advanced Design System. For more detailed information on using the Cadence Hierarchy Editor tool exclusively, refer to your Virtuoso Analog Circuit Design Environment User Guide . For more detailed information on expanding hierarchy in the Cadence environment, refer to the section on How the Netlister Expands Hierarchy in your Cadence documentation.

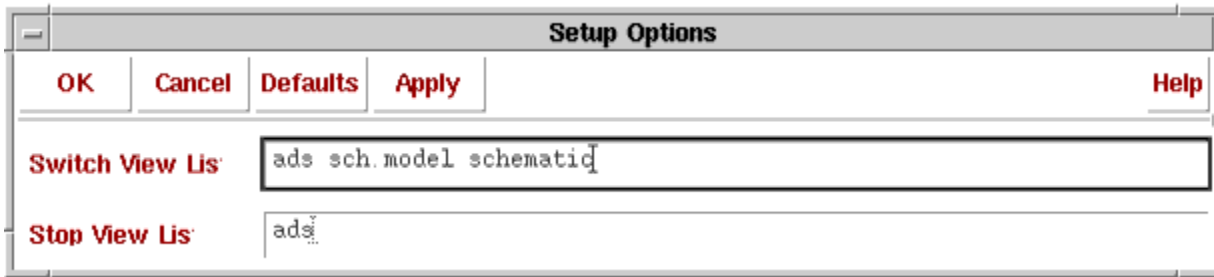
Expanding Hierarchy with the Dynamic Link Netlister

The flowchart shown in the following figure describes the general process used by the RFIC Dynamic Link netlister to move through and expand the hierarchy in Dynamic Link.



Netlist Hierarchy Expansion

In the Cadence schematic window, choose the DynamicLink > Setup Options menu item to access the Setup Options form. The Setup Options form enables you to designate a switch view list, and a stop view list.

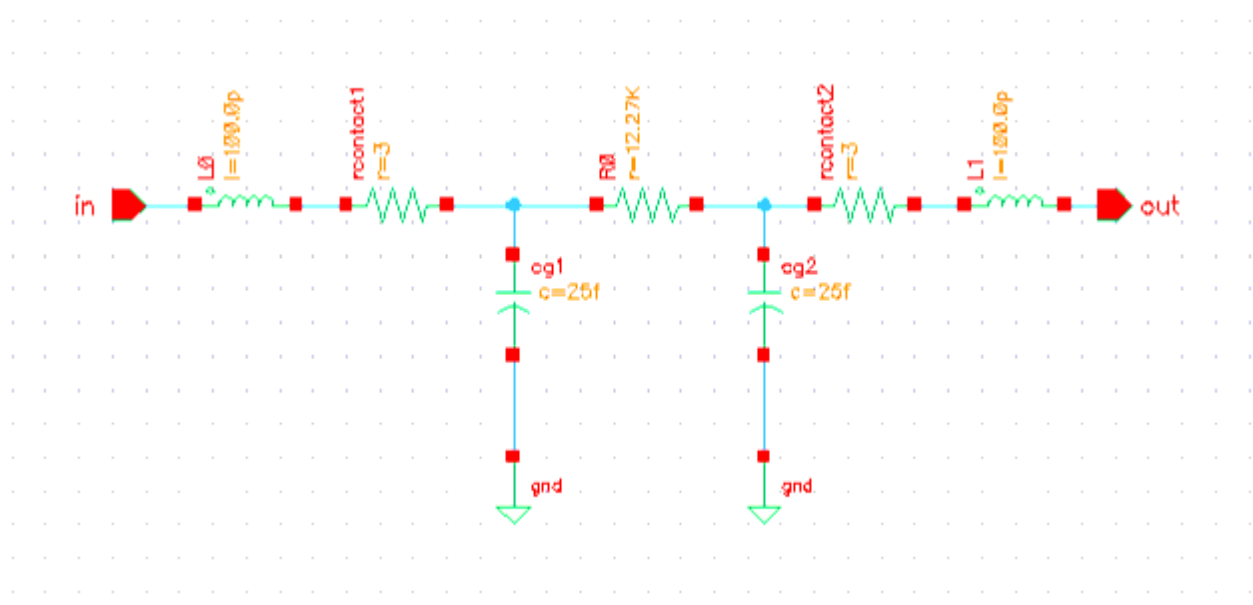


Setting a Switch View List and a Stop View List.

Each instance in a hierarchy has a master cell, that contains a number of views for that cell. Typically views are schematic, symbol, layout, extracted, etc. The switch view list is used to enable you to designate the priority of each view for hierarchical netlisting. The stop view is used to designate whether a view in the switch view list should be traversed for hierarchy. A stop view is implied to have no hierarchy. For an example of this, refer to the design hierarchy_bottom in the library examples that is provided with the Dynamic Link installation. In this example, the cell hierarchy_bottom has two alternate schematic views, schematic_ideal and schematic_parasitics (see the following two figures).

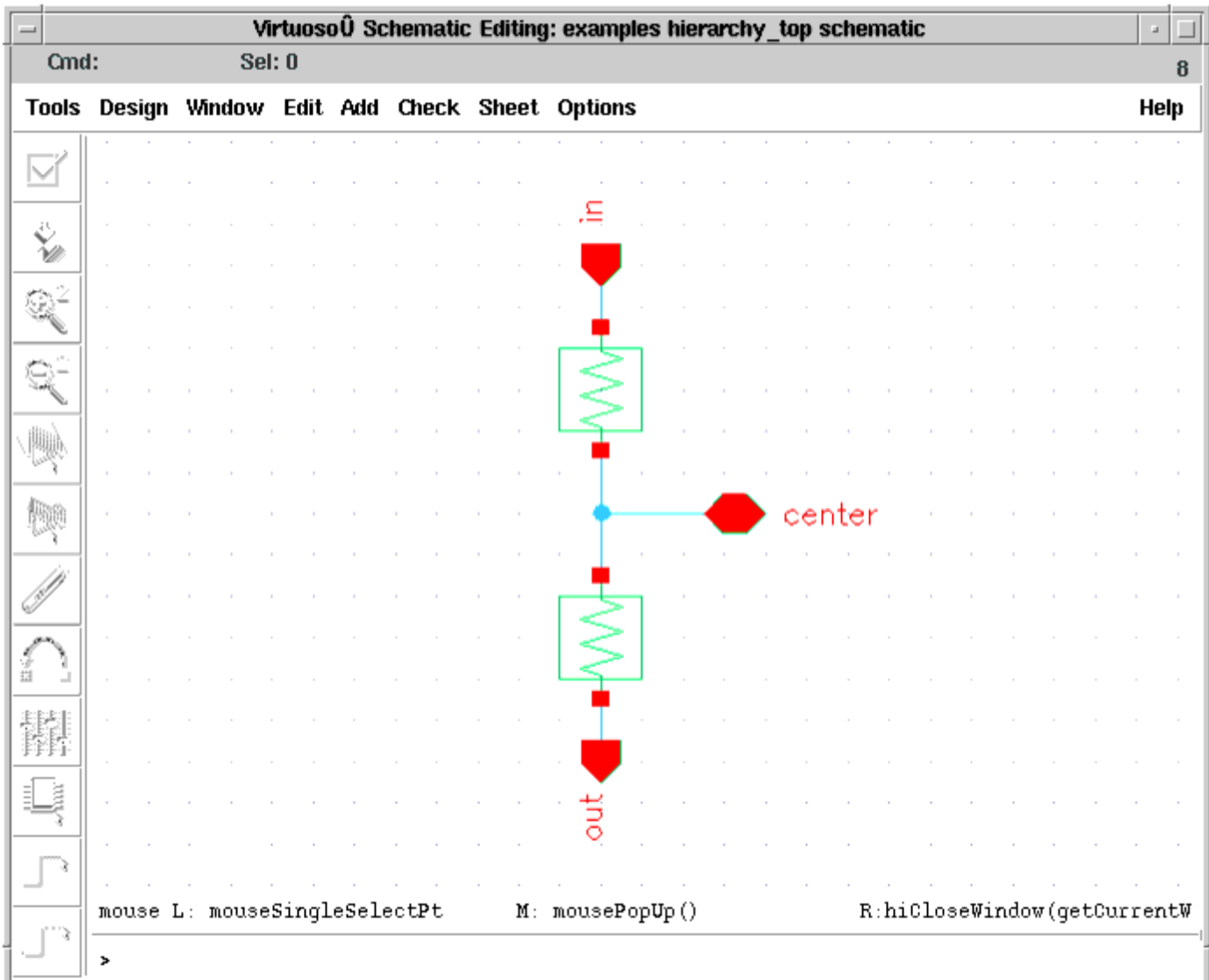


Alternate schematic view schematic_ideal for the example cell hierarchy_bottom .



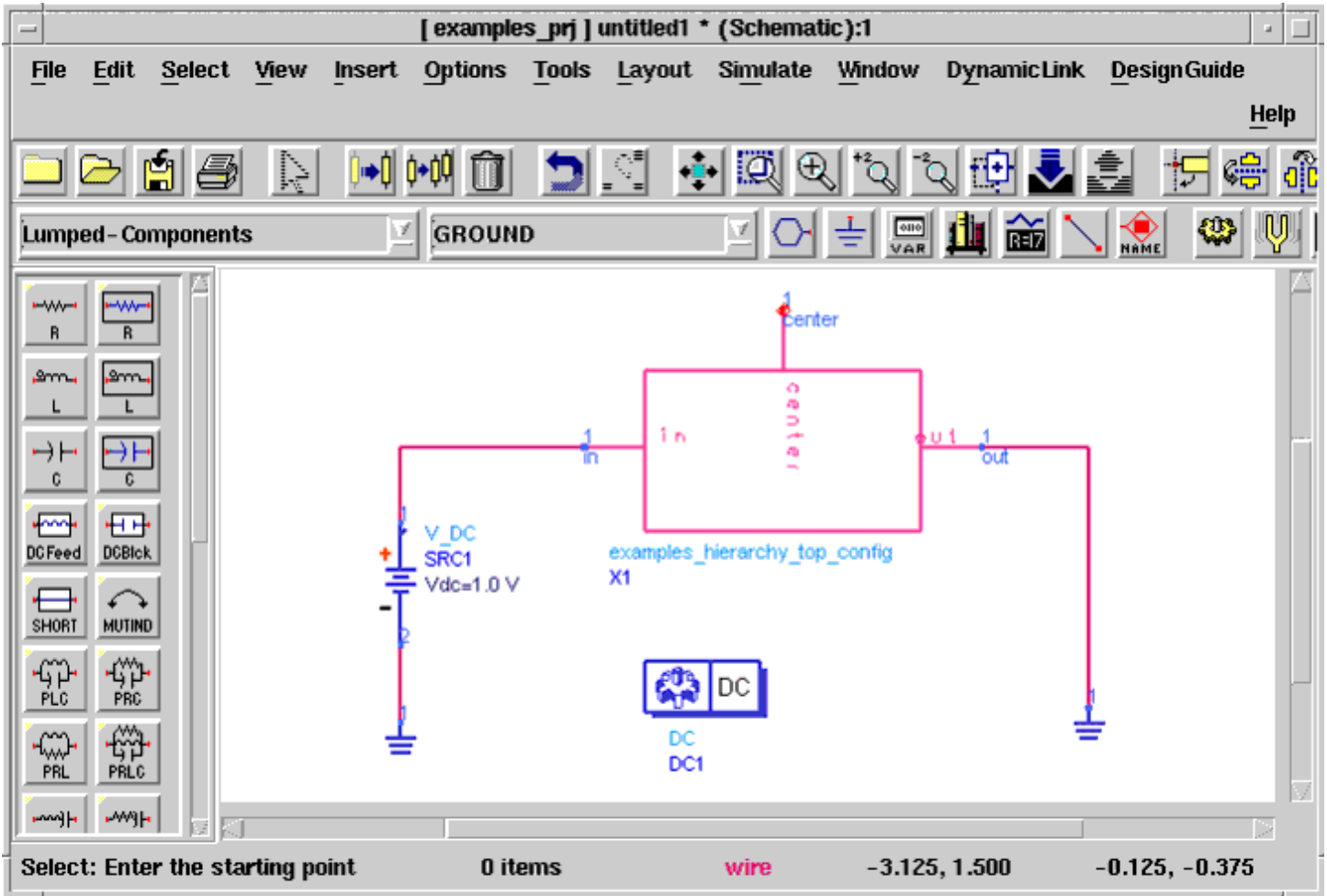
Alternate schematic view schematic_parasitics for the example cell hierarchy_bottom .

Two instances of the cell hierarchy_bottom have been placed in the schematic hierarchy_top (see the following figure) in the examples .



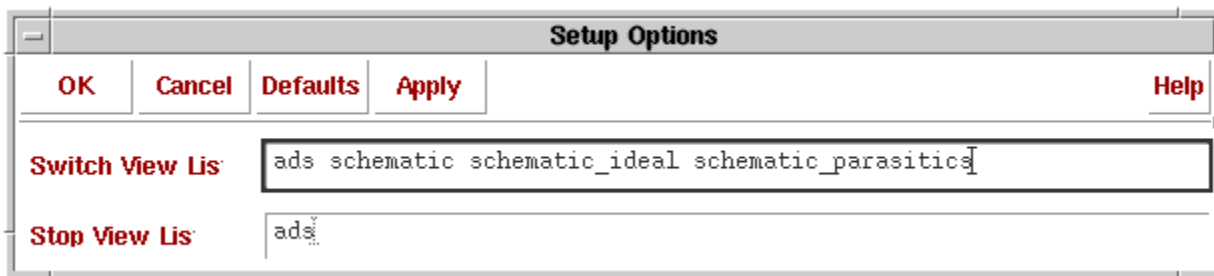
The schematic view for the example cell hierarchy_top

This cell is in turn placed in the ADS example project design hierarchyTest , with the schematic view chosen as the Cadence view (see the following figure). Because there are two alternate schematics, it becomes necessary to tell the netlister which one to use when you netlist the design hierarchy_top . The switch view list is used to tell the netlister which one you want to use.



The cadence cell hierarchy_top placed in ADS

The following figure shows how the setup options have been changed to designate that schematic_ideal takes priority.



Switch View List with schematic_ideal taking priority

Referring to the Hierarchy Expansion flowchart in the figure [Netlist Hierarchy Expansion](#), when an instance of hierarchy_bottom is encountered, the following occurs:

1. There is no ads view, so the next switch view is looked at.
2. There is no schematic view, so the code continues to the next switch view.
3. There is a schematic_ideal view. Since schematic_ideal is not in the stop view list, it is opened, so that a subcircuit can be netlisted for it.
4. With hierarchy_bottom as the top cell, now the first instance, and only instance encountered, is the analogLib res component. The switch view list is consulted.
 - Is there an ads view? Yes, there is.
 - Is the ads view a stop view? Yes, it is.

This means that the res component will not be opened as a schematic with hierarchy. Instead, it is meant to be a simulator primitive. Either a built in simulator component exists, which will be used, or a subcircuit definition has been included into the simulator that defines what the component is. A single component line is output for it, and the netlister continues on.

The following figure shows the resulting netlist from netlisting hierarchy_top with schematic_ideal having higher priority than schematic_parasitics . Note that the instances and subcircuit definition have been highlighted.

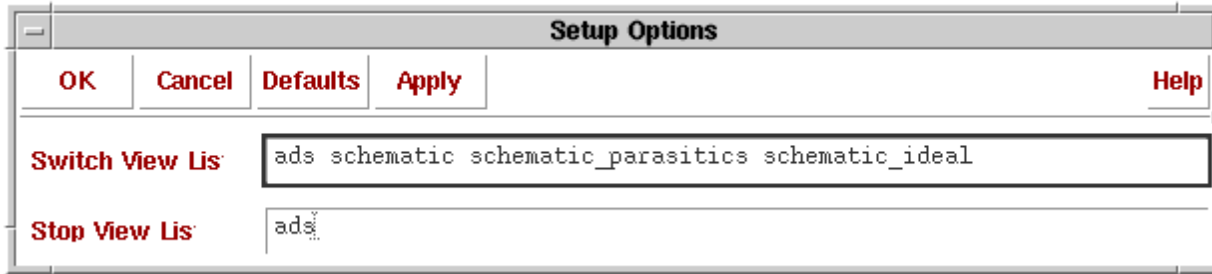
```

[learn2] /gfs/sr/learn2/d1/local/users/crscott/simulation/hierarchy_top/ads/schematic/netlist/netlist
File Help 6
; Library name: examples
; Cell name: hierarchy_bottom
; View name: schematic_ideal
#ifndef inc_examples_hierarchy_bottom_schematic_ideal
#define inc_examples_hierarchy_bottom_schematic_ideal
define examples_hierarchy_bottom_schematic_ideal ( in out)
R:R0 in out R=12.27K
end examples_hierarchy_bottom_schematic_ideal
#endif
; Library name: examples
; Cell name: hierarchy_top
; View name: schematic
#ifndef inc_examples_hierarchy_top_schematic
#define inc_examples_hierarchy_top_schematic
define examples_hierarchy_top_schematic ( center in out)
examples_hierarchy_bottom_schematic_ideal:XI7 center out
examples_hierarchy_bottom_schematic_ideal:XI6 in center
end examples_hierarchy_top_schematic
#endif

```

Netlist of the example cell hierarchy_top with schematic_ideal taking priority

If you prefer to use the schematic_parasitics schematic, the hierarchy_top schematic does not need to be changed. Instead, in the setup options form, you change the switch view list so that schematic_parasitics comes before schematic_ideal as shown in the following figure.



Switch View List with schematic_parasitics taking priority over schematic_ideal

Referring to the flowchart in the figure [Netlist Hierarchy Expansion](#) again, when an instance of hierarchy_bottom is encountered, the switch view list is checked. There is no ads view or schematic view. There is a schematic_parasitics view, which is not a part of the stop view list. This results in the netlister opening the schematic_parasitics view, which is netlisted as a subcircuit. The netlister then goes on to the next instance, without ever checking for a schematic_ideal view (in point of fact, it is not necessary to put schematic_ideal in the switch view list, since it will not be used). The following figure shows the resulting netlist with the setup options from the previous figure.

Netlist of example hierarchy_top with schematic_parasitics taking priority

In all of the above examples, the stop view list was set to ads . This is the recommended stop view list to use for the RFIC Dynamic Link netlister. If you consult the netlist flowchart in the figure [Netlist Hierarchy Expansion](#), you will notice that it is not necessary for the stop view to be ads , any view can be used to designate that a part is a primitive. In the future, the RFIC Dynamic Link netlister will be expanded, so that alternate simulation definitions can be used based on which stop view is encountered (e.g. ads vs. ads_ptolemy). At present, the netlister always uses the simulation definition defined for ads in the cell's CDF, no matter which stop view is encountered.

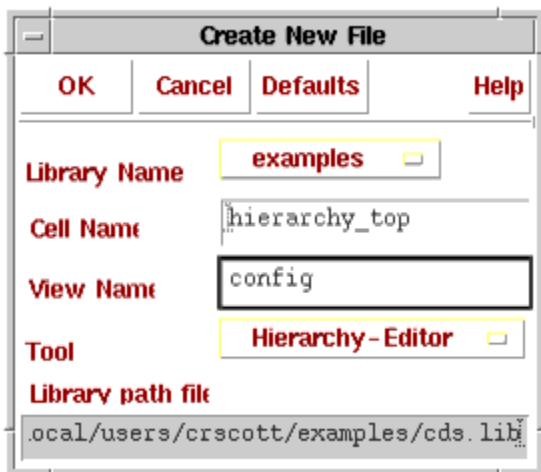
Using the Hierarchy Editor with RFIC Dynamic Link

The Hierarchy Editor is a stand alone Cadence tool that enables switch views and stop views to be designated for each

instance in a schematic hierarchy. For information regarding the hierarchy editor in general, consult your Cadence Hierarchy Editor User Guide .

When the Hierarchy Editor is used, a cell view specific to that tool is generated. The view itself is actually a text file, and cannot be opened directly in anything other than the Hierarchy Editor. The default view name for the Hierarchy Editor is config . A config view has a top cell view that it points to. All other information regarding switch views and stop views is then created based on the top cell view that is being pointed to. It is not necessary for the Hierarchy Editor view to be a part of the cell that it is pointing to, although in most cases this will be the simplest way to organize things.

The following figure displays the Create New File form used to create a Hierarchy Editor view. In this case, a Hierarchy Editor view is being made for the example cell hierarchy_top (see the figure [The schematic view for the example cell hierarchy_top](#)). The goal is to set up this config view so that, during netlisting, it is possible to specify that one instance of hierarchy_bottom should use the schematic_ideal view, while the other instance uses the schematic_parasitics view.



Creating a Hierarchy-Editor view

After the new view is created, the Hierarchy Editor tool is started. For a new view, the New Configuration form appears as shown in the following figure.

New Configuration

Top Cell

Library: Cell: View:

Global Bindings

Library List:

View List:

Stop List:

Description

OK Cancel Use Template... Help

New Hierarchy Editor Configuration

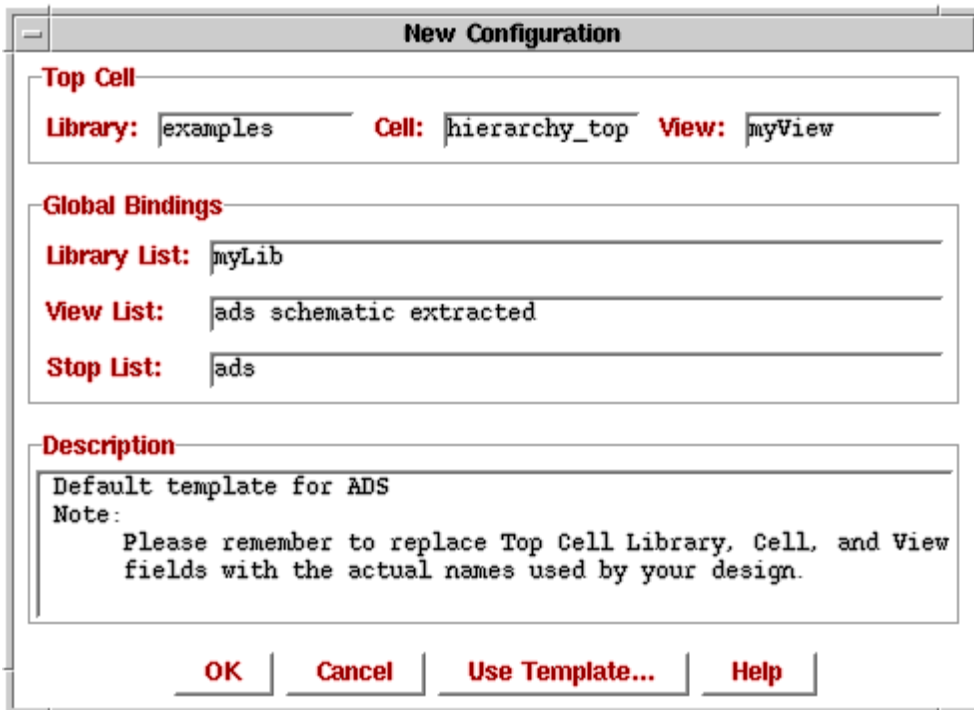
Notice that all fields are left blank initially, except the Library and Cell fields which were specified in the Create New File form shown in the figure [Creating a Hierarchy-Editor view](#). At this point, it is necessary to designate the top cell view, as well as the switch view list and stop view list. The Library List can normally be left blank. For more information on the Library List field, refer to your Cadence Hierarchy Editor User Guide .

It is necessary to fill in the Top Cell view. The top cell view should be a standard CDB view. From the standpoint of RFIC Dynamic Link, this means either a view that is edited with Virtuoso-schematic, or Virtuoso-layout. If you use a layout view, it should be an extracted view of some sort (i.e. the layout will contain connectivity and instances that can be traced back to schematic equivalents). Typically, you will put in either schematic or extracted as the view name.

The Library List, View List, and Stop List can be filled in by using a template. Templates contain default settings for particular simulators. To select a template, click the Use Template button in the New Configuration form. The Use Template form appears.



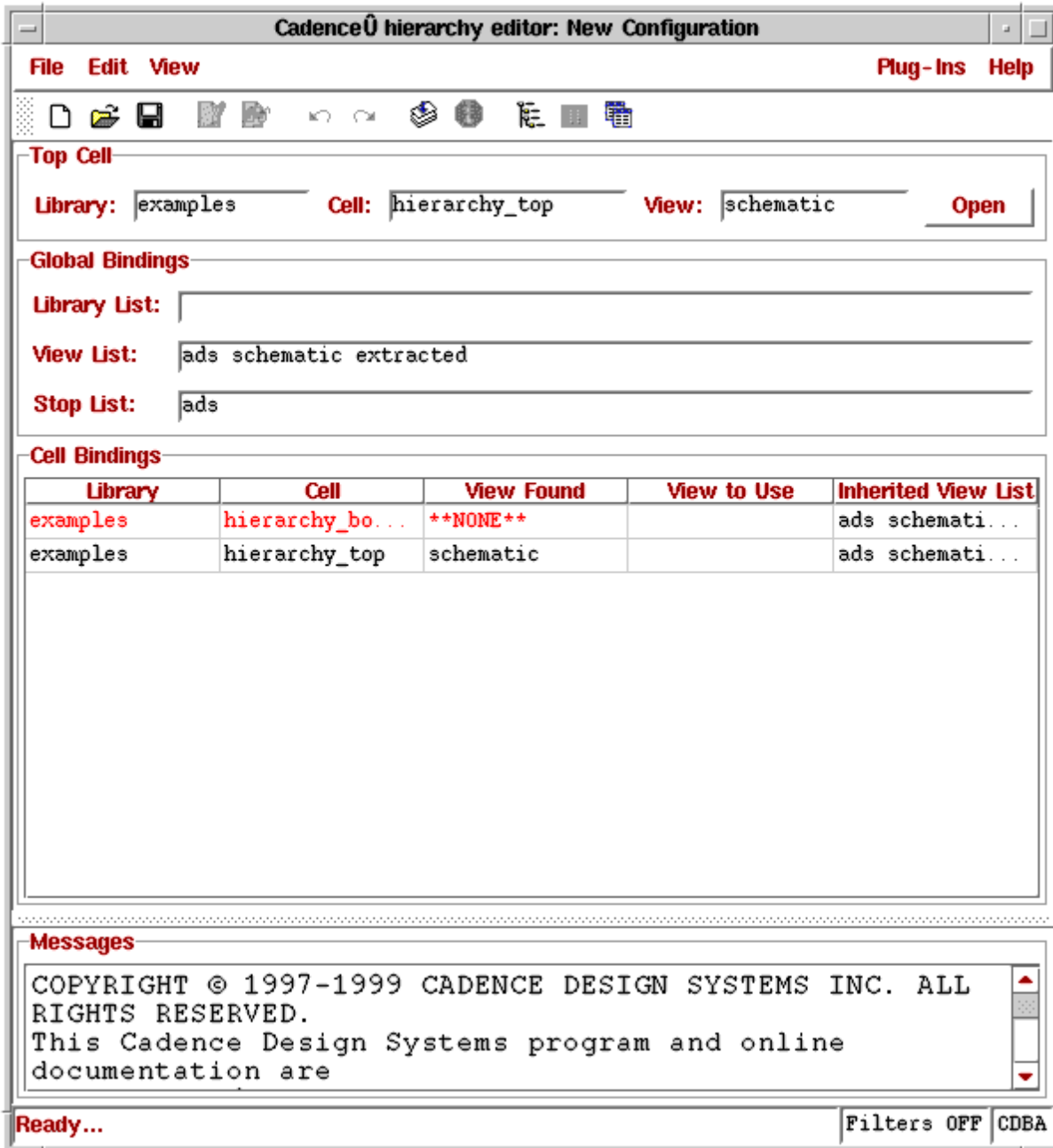
In the following figure, the ads template was selected, which has filled the View List in as ads schematic extracted , and the stop list as ads . These names represent the default view names for the tools that the Dynamic Link netlister supports. During netlisting, the view list and stop list will override the switch view list and stop view list that is specified using the Dynamic Link setup options dialog. Also, the Hierarchy Editor view will not consult the Dynamic Link options dialog to fill in the switch view list or stop view list, it is a completely separate tool, and must be set up independently.



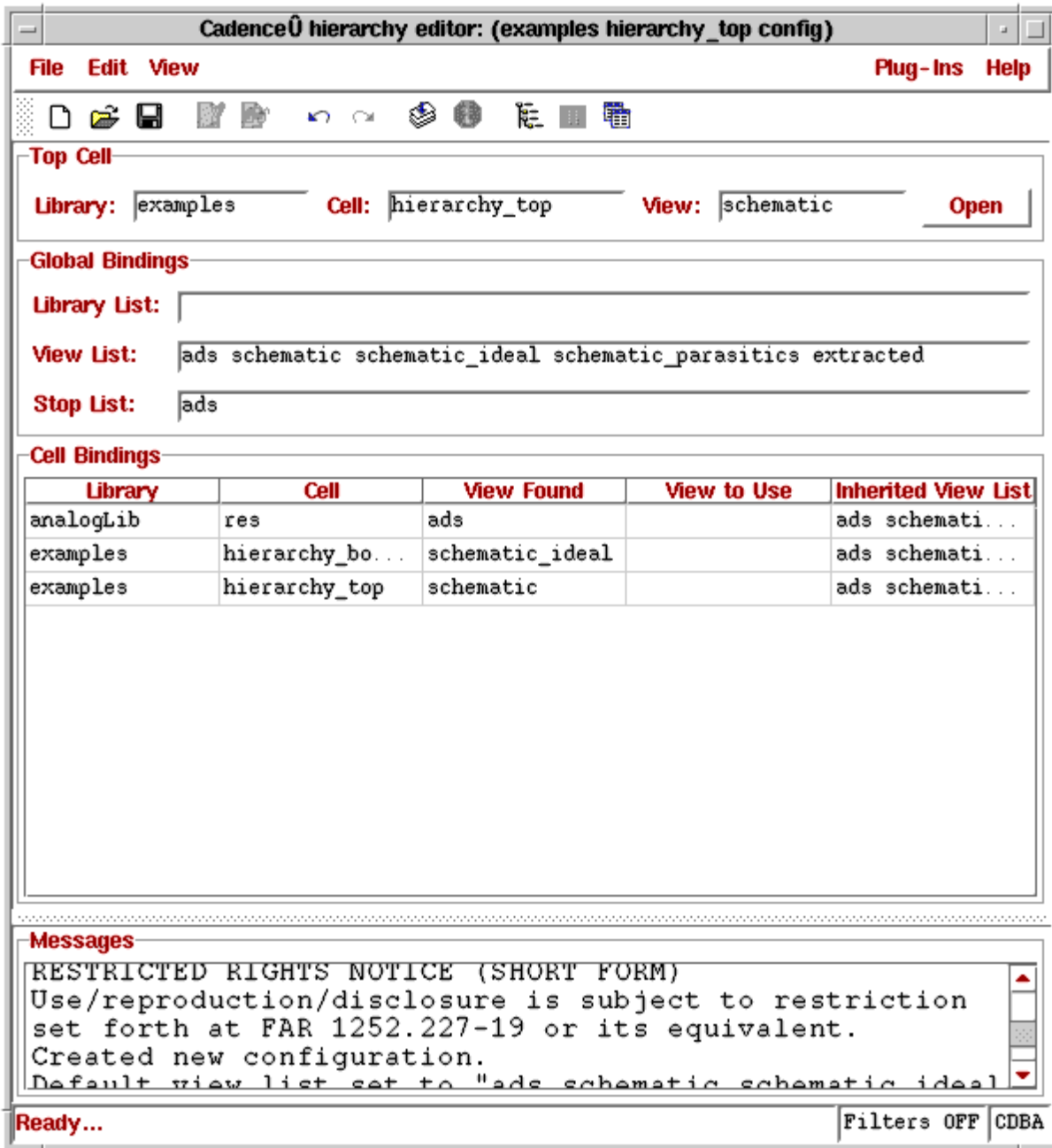
New Hierarchy Editor Configuration, using ADS template

When the new configuration is accepted, the main Hierarchy Editor window will be filled out. The main window shows the current Top Cell setting, as well as the global Library List, View List, and Stop List. In addition, the Hierarchy Editor will expand the hierarchy of the top cell, and show which views will be used for each instance within the top cell's hierarchy. When nothing has been overridden, the expansion will follow the flow chart shown in the figure [Netlist](#)

[Hierarchy Expansion](#). The following figure shows how the example hierarchy_top schematic view was expanded using the global bindings. The Cell Bindings shows the two cells that were found, hierarchy_top and hierarchy_bottom . You may need to select View > Update to see the hierarchy_bottom Cell in the Cell Bindings list. The view found for hierarchy_top was schematic, this is a special case. Because it is the top cell, the library, cell, and view found will always be the same as what is specified for the top cell, regardless of the view list and stop list. The cell hierarchy_bottom says that the view found was ****NONE****. Because hierarchy_bottom has the views schematic_ideal , schematic_parasitics , and symbol , this is accurate. None of hierarchy_bottom 's views are in the global view list. If netlisting were attempted at this point, an error would result. The cell bindings give a visual indication of this.

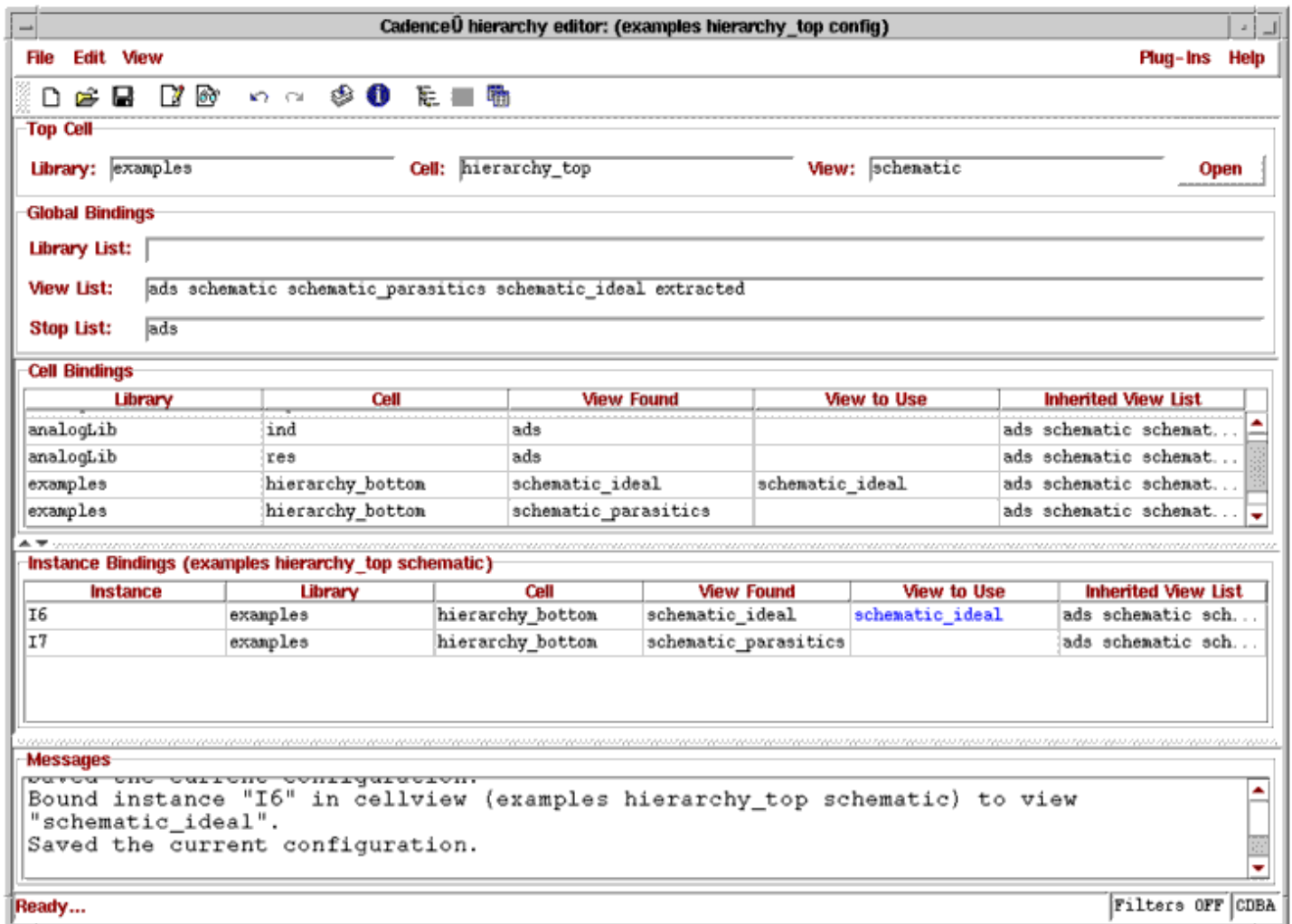


In the following figure, the global view list has been changed, so that it now includes schematic_ideal and schematic_parasitics . The cell bindings are now updated to reflect the new hierarchy expansion. Because schematic_ideal precedes schematic_parasitics in the global view list, the expansion now says that, for hierarchy_bottom , schematic_ideal is the view found. Also, the analogLib res cell has been found, because hierarchy_bottom was expanded. This expansion happens because schematic_ideal is not listed in the global stop view list. Since res has an ads view, the expander decides that ads will be the view used for the res cell.



So far, all of this demonstrates is that the Hierarchy Editor can be used to see how expansion would occur for a specified view list and stop list. That's nice, but it doesn't add any functionality over what was provided by the Setup Options form. What is needed is a way of overriding the view list. As it happens, the Hierarchy Editor is capable of doing just that.

In the following figure, the switch view list has been changed, so that schematic_parasitics is now first in the switch view list.



Notice that the instance bindings table is now displayed. When hierarchy_top is selected, the instance table shows all

of the instances in the schematic. As it happens, hierarchy_top contains two instances of hierarchy_bottom , I2 and I3 (see the figure [The schematic view for the example cell hierarchy_top](#)). If the global switch view list is used, both instances will expand to use schematic_parasitics. In this case, we have chosen to change the default behavior. In the view to use field, schematic_ideal has been specified. The view found field now specifies schematic_ideal instead of schematic_parasitics . Now, when the cell hierarchy_top is netlisted with this configuration, it will be necessary to expand the hierarchy for both schematic_parasitics and schematic_ideal. The netlister keeps track of the proper component name for the netlist.

The figure below shows the resultant netlist that is created using this configuration.

```

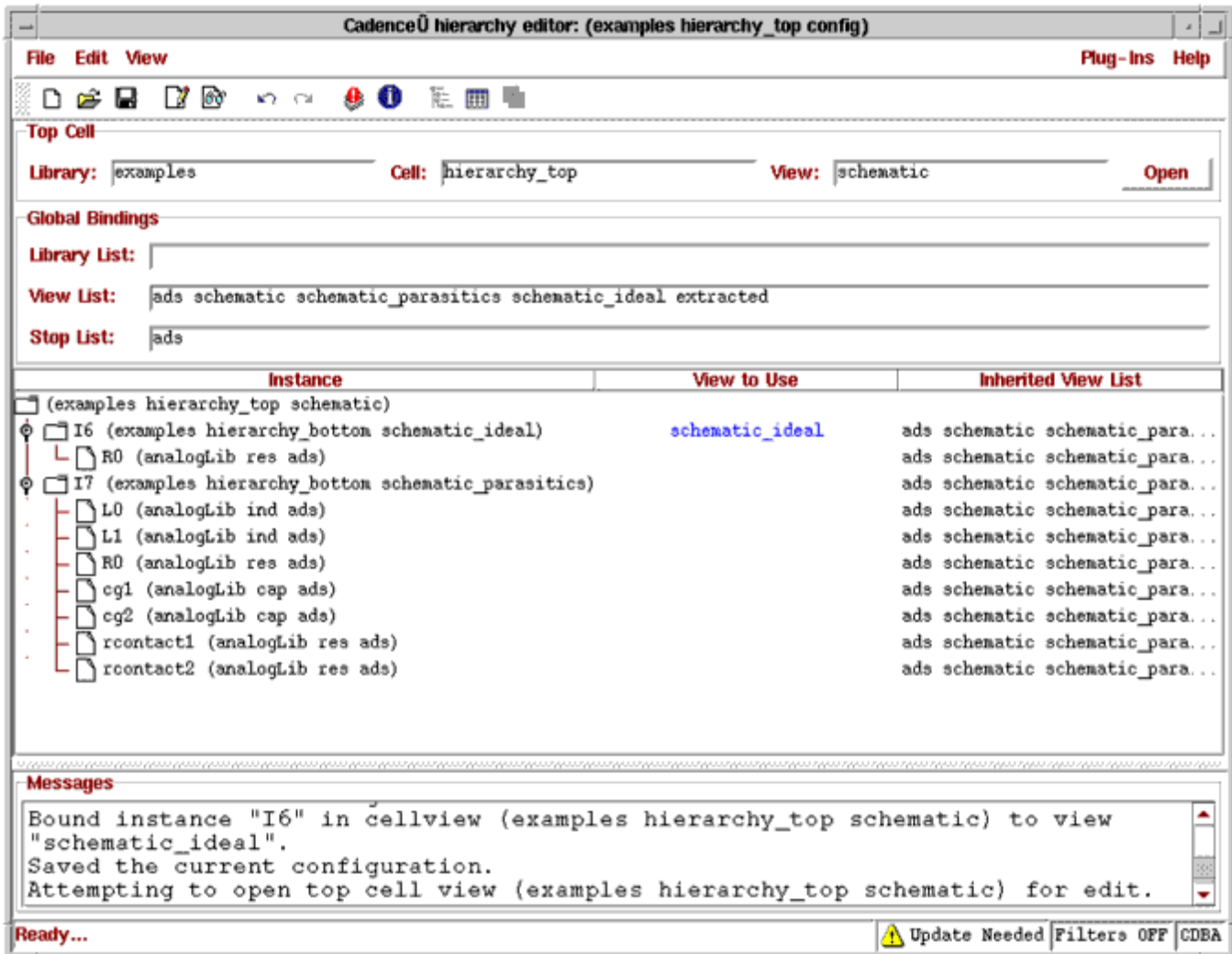
- [learn2] /gfs/sr/learn2/d1/local/users/crscott/simulation/hierarchy_top/ads/config/netlist/netlist
File Help 6

; Library name: examples
; Cell name: hierarchy_bottom
; View name: schematic_parasitics
; Inherited view list: ads schematic schematic_parasitics schematic_ideal
; extracted
#ifndef inc_examples_hierarchy_bottom_schematic_parasitics
#define inc_examples_hierarchy_bottom_schematic_parasitics
define examples_hierarchy_bottom_schematic_parasitics ( in out)
L:L1 net11 out L=100.0p
L:L0 in net5 L=100.0p
C:cg2 net15 0 C=25f
C:cg1 net13 0 C=25f
R:R0 net13 net15 R=12.27K
R:rcontact2 net15 net11 R=3
R:rcontact1 net5 net13 R=3
end examples_hierarchy_bottom_schematic_parasitics
#endif
; Library name: examples
; Cell name: hierarchy_bottom
; View name: schematic_ideal
; Inherited view list: ads schematic schematic_parasitics schematic_ideal
; extracted
#ifndef inc_examples_hierarchy_bottom_schematic_ideal
#define inc_examples_hierarchy_bottom_schematic_ideal
define examples_hierarchy_bottom_schematic_ideal ( in out)
R:R0 in out R=12.27K
end examples_hierarchy_bottom_schematic_ideal
#endif
; Library name: examples
; Cell name: hierarchy_top
; View name: schematic
; Inherited view list: ads schematic schematic_parasitics schematic_ideal
; extracted
#ifndef inc_examples_hierarchy_top_config
#define inc_examples_hierarchy_top_config
define examples_hierarchy_top_config ( center in out)
examples_hierarchy_bottom_schematic_parasitics:XI7 center out
examples_hierarchy_bottom_schematic_ideal:XI6 in center
end examples_hierarchy_top_config
#endif

```

Netlist using configuration detailed in figure 10-15.

An alternate way of viewing the overridden expansion is to look at the tree view, as opposed to the cell and instance binding tables. This is shown in the following figure.



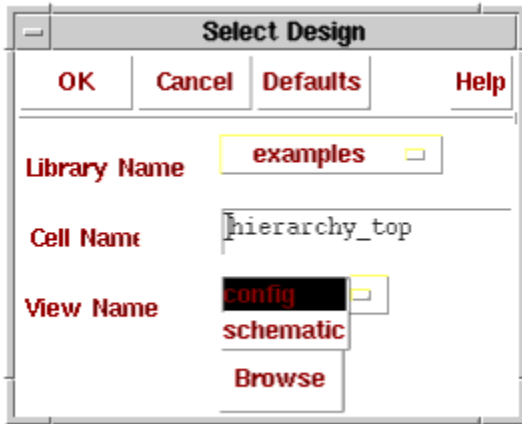
Tree view of hierarchy_top configuration with I2 set to use schematic_ideal.

The Hierarchy Editor makes it possible to override the hierarchy expansion on any instance. It is also possible to override the global switch view list on any instance. This allows you to change the expansion options for one tree of a hierarchy, while leaving all other trees intact.

Placing the config view in ADS

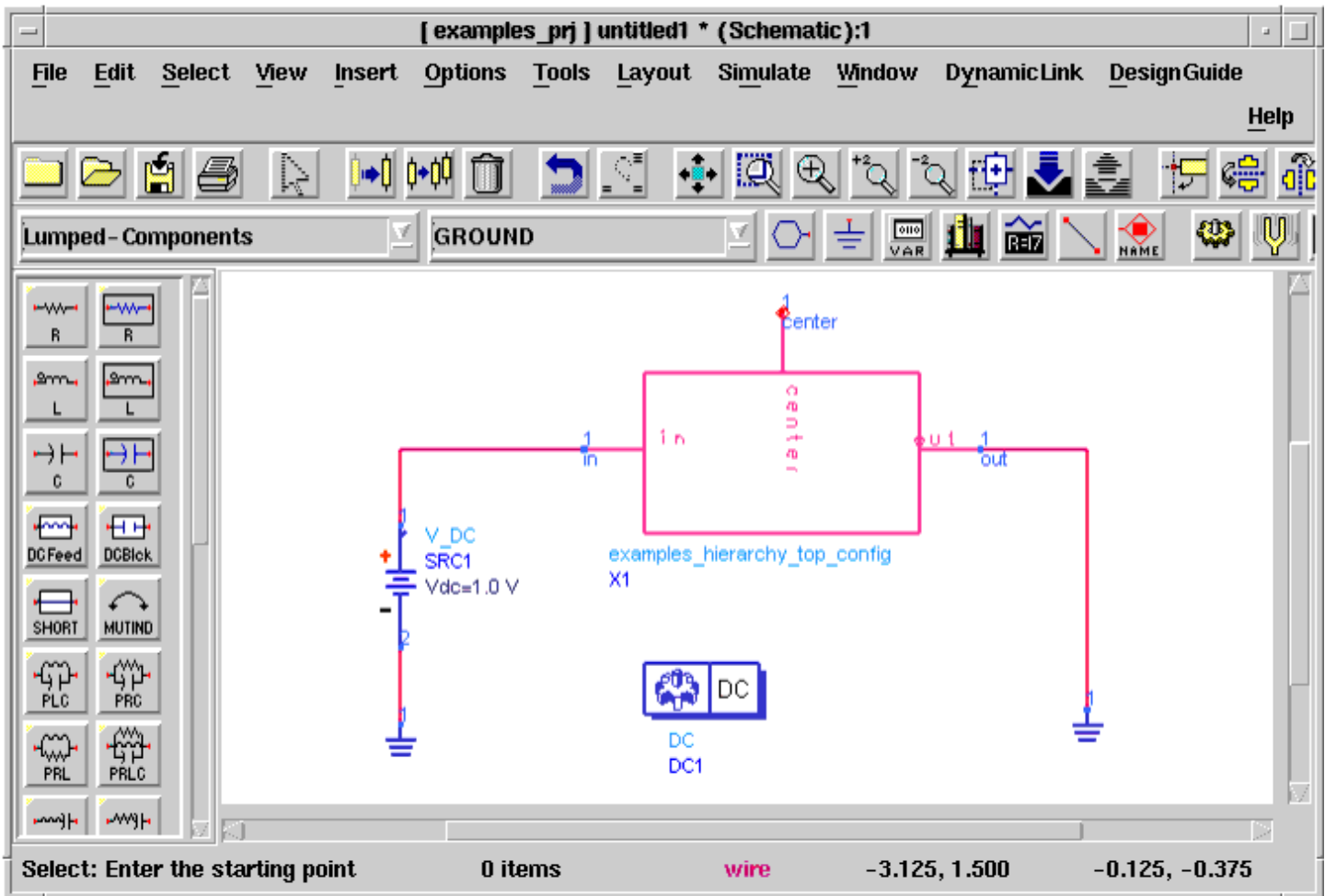
In order for a Hierarchy Editor configuration to be used during netlisting, it is necessary to place the Hierarchy Editor view in the ADS schematic. This is done by selecting the DynamicLink > Instance > Add Instance of Cellview menu

option in ADS. When the Select Design dialog appears, set the view name to the Hierarchy Editor view. In the hierarchy_top example, this is done by selecting the view name config (see the following figure).



Selecting the Hierarchy Editor view for placement in ADS.

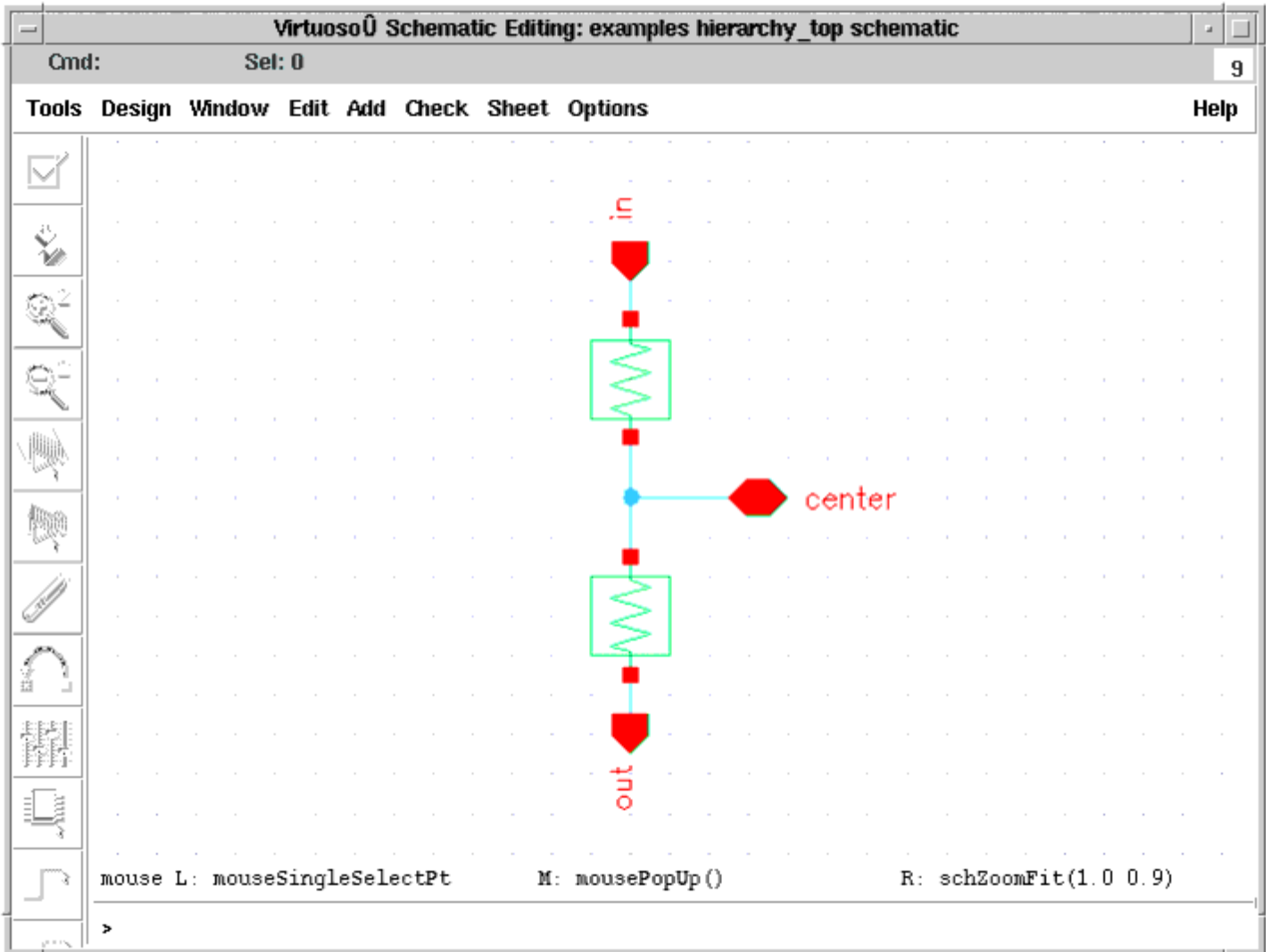
A new symbol is generated for the config view, and you are then be able to place it. In the hierarchy_top example, the symbol graphics are inherited from the hierarchy_top cell. This results in a symbol that looks identical to the hierarchy_top schematic symbol for ADS. However, this symbol is now linked to the Hierarchy Editor view, as opposed to being linked to the schematic view.



Hierarchy Editor test bench with config view placed

The previous figure shows the new test bench where the config view is used instead of the schematic view. If the Cadence instance is selected, and you descend into it's hierarchy, the top cell view for the configuration will be opened. The configuration being used will be indicated in the title of the schematic/layout window that is opened. The following figure shows the hierarchy_top schematic that will be opened. Note that the title indicates that the Config in use is examples hierarchy_top config . This is important, as it means that hierarchy expansion will obey that particular configuration.

Note also that the top cell view does not need to remain constant. Thus, if you wish to do a simulation where the top cell schematic is used, and then do another simulation wherein an extracted view is used, you do not need to make multiple ADS symbols and then swap them. You can make a single Hierarchy Editor configuration, and swap the top cell view name. Once the test bench is set up, you do not need to modify anything in ADS to change your simulation. For the hierarchy_top example, a simulation would result in the netlist shown in the figure [Netlist using configuration detailed in figure 10-15.](#), based on the configuration shown in the figure [Hierarchy Editor with an instance view overridden.](#)



Schematic window attached to a Hierarchy Editor configuration

Troubleshooting RFIC Dynamic Link

This chapter provides troubleshooting information that can help you resolve common problems is also provided in this appendix.

All errors, warnings, and other messages are directed to the Cadence CIW. When a new message is written to the CIW, the window is raised to the top of your window stack so that new messages are always visible. Error messages

may also be logged in a file, idf.log.

Some known problems and solutions are listed in the following section. You may find this information helpful in determining how to resolve a particular problem however, if you're unable to resolve a problem with the RFIC Dynamic Link using the information provided, contact Agilent EEsof EDA customer support.

Known Problems and Solutions

Problem: By default, ADS does not create its own private color map, which may lead to unpredictable color behavior and/or menu buttons in place of icons.

Solution: Try one or more of the following:

- Set HPEESOF_COLORMAP = private in the ADS configuration file \$HOME/hpeesof/config/hpeesof.cfg or \$HPEESOF_DIR/config/hpeesof.cfg .
- Set CDS_NUM_USER_COLORS = 16 in your .profile or .cshrc file.
- Restart Dynamic Link after exiting all other color-intensive applications.

Problem: When you remake a symbol, even when the old symbol is deleted, ADS does not allow you to create another symbol with the same name.

Solution: Click the instance in the ADS schematic window, then choose DynamicLink > Instance > Update Instance of Cellview . If the problem persists:

- Delete the <lib> _ <cells> _ <view> symbol in the ADS Design.
- Save the ADS design.
- Exit ADS.
- Delete all the <ADS_proj_dir> /networks/ <lib> _ <cells> _ <view> .* files.
- Restart ADS.

Problem: *Error* Could not find `nlpglobals/ads' in library `basic'. The nlpglobals' cell view is required. Netlisting aborted. This error occurs either while netlisting in Analog Artist or after clicking Simulate in the ADS.

Solution: Copy the spectre view in the nlpglobals cell to create an ads view.

Advanced Design System 2008

Problem: There is no distinction between a design variable X from Design A and a design variable X from a different Design B .

Solution: Use unique design variable names for different designs, unless you really intend them to be the same variable, in which case there's no problem.

Problem: Symbol generation via Cadence symbol duplication does not reproduce arcs.

Solution: Use line segments instead of arcs.

Problem: Could not spawn master program . This message appears in your parent terminal window upon attempting to use the ADS link.

Solution: Ensure that \$ HPEESOF_DIR/ bin is in your PATH and that \$ HPEESOF_DIR/ bin/idfmp is a valid executable. If this does not work, ask your UNIX System Administrator to reboot your system or otherwise determine if a socket address is in use.

Problem: The UNIX environment does not set up properly when an in-house script for DFII is used.

Solution: Starting DFII using an in-house script may not set up the UNIX environment properly for RFIC Dynamic link. Work with your System Administrator to ensure that you understand what environment variables need to be set in the in-house script and modify your script accordingly.

Installation and Use Checklist

This section provides a checklist that can be used to help you resolve problems with RFIC Dynamic Link. You can use the questions below to help determine what the cause of a particular problem might be.

- Is RFIC Dynamic Link installed? If yes, the following files/directories should exist:
 - \$HPEESOF_DIR/bin/idf
 - \$HPEESOF_DIR/bin/idfmp
 - \$HPEESOF_DIR/circuit/symbols/idfSymbol.dsn
 - \$HPEESOF_DIR/idf/ael/
 - \$HPEESOF_DIR/idf/skill/5.0.0/ads.ini
 - \$HPEESOF_DIR/idf/skill/5.0.0/ads.al
 - \$HPEESOF_DIR/idf/skill/5.0.0/ads.cxt

- \$HPEESOF_DIR/idf/skill/4.4.6/ads.ini
- \$HPEESOF_DIR/idf/skill/4.4.6/ads.al
- \$HPEESOF_DIR/idf/skill/4.4.6/ads.cxt
- \$HPEESOF_DIR/idf/skill/4.4.5/ads.ini
- \$HPEESOF_DIR/idf/skill/4.4.5/ads.al
- \$HPEESOF_DIR/idf/skill/4.4.5/ads.cxt
- \$HPEESOF_DIR/idf/config/
- \$HPEESOF_DIR/cdslibs/5.0.0/analogLib/
- \$HPEESOF_DIR/cdslibs/4.4.6/analogLib/
- \$HPEESOF_DIR/cdslibs/4.4.5/analogLib/
- \$HPEESOF_DIR/cdslibs/5.0.0/basic/
- \$HPEESOF_DIR/cdslibs/4.4.6/basic/
- \$HPEESOF_DIR/cdslibs/4.4.5/basic/
- \$HPEESOF_DIR/idf/examples/
- Is Cadence configured for RFIC Dynamic Link? Run `idfConfigCadence -h` (RFIC Dynamic Link 2002) or `$HPEESOF_DIR/idf/config/configCadence -h` (RFIC Dynamic Link 2001).
- Is HPEESOF_DIR set? Run `idfenv` (RFIC Dynamic Link 2002 or later).
- Is \$HPEESOF_DIR/bin in your \$PATH?
- Is icms in your \$PATH?
- Are cds_root and cdsd both in your \$PATH?
- Is PATH, AGILEESOF_LICENSE_FILE or LM_LICENSE_FILE set in ~/.cshrc, ~/.profile, ~/.kshrc, or other scripts sourced by these scripts? Execute `echo $SHELL` or `finger` to display your login shell.
- Is your ADS license.lic file in the default \$HPEESOF_DIR/licenses/ directory or is it set in \$AGILEESOF_LICENSE_FILE?
- Is there a trans_idf (RFIC Dynamic Link 2002) or Idf_c_interface (previous versions of RFIC Dynamic Link) feature in the ADS license.lic or license.dat file? Has this feature expired?
- Which item in the following list defines the Cadence license file?
 - \$CDS_LIC_FILE
 - <Cadence_install_dir> /share/license/clients
 - \$LM_LICENSE_FILE
 - <Cadence_install_dir> /share/license/license.dat
- Does the Cadence license contain the valid features listed below?
 - OASIS_Simulation_Interface
 - 34510
 - 300 (only if using layout)
- Are all seats of the above ADS and Cadence licenses taken?
- Does \$HPEESOF_DIR/idf/config/.cdsinit exist?
- Is \$HPEESOF_DIR/idf/config/.cdsinit loaded in <Cadence_installation_directory>/tools/dfII/local/.cdsinit?
- If the above file does not exist, is \$HPEESOF_DIR/idf/config/.cdsinit loaded in ./cdsinit?
- If the above files do not exist, is \$HPEESOF_DIR/idf/config/.cdsinit loaded in \$HOME/.cdsinit?
- If none of the .cdsinit files exists, `cp $HPEESOF_DIR/idf/examples/.cdsinit ./`
- Does the .cdsinit file that loads RFIC Dynamic Link's .cdsinit file change PATH, CDS_INST_DIR, LM_LICENSE_FILE?
- Does the icms , icfb , or msfb script change your PATH or LM_LICENSE_FILE? Run `idfenv` at the UNIX prompt, start Cadence, enter `system("idfenv")` in the Cadence CIW, and then compare the output of `idfenv` before and after the Cadence script is executed.
- What is in the ./cds.lib ? Do all of the libraries appear in blue in Cadence Library Path Editor form? If you are using the Agilent Technologies version of analogLib and analogLib appears in red, check to see if IDF_CDS_VERSION is used in cds.lib and it is not defined in the effective .cdsinit file.
- Does an ads view exist for the components in use? Use the Cadence Library Manager to check to see if the

analogLib/cap cell contains an ads view.

- Does the ads simInfo section exist in the Cadence Component Description Format (CDF) for the components in use? Is all of the information in the ads simInfo correct? Use the Edit Component CDF form (Tools > CDF > Edit) or `cdfDump("<lib>" "/tmp/<cell>.cdf" ?cellName "<cell>" ?edit t)` function to see if the ads simInfo section exists in a cellview (leave out the `"?cellName "<cell>"` for dumping the library CDF instead of the cell CDF).
- Does the ads view exist for the global cellview nlpglobals under the basic library? If not, open the Cadence Library Manager, select symbol or spectre view, press middle mouse button, choose Copy from the popup list, and copy the symbol or spectre view to the ads view.
- If RFIC Dynamic Link failed to create an ADS netlist, can the Cadence netlister create a spectre netlist? Select Tools > Analog Environment from Cadence schematic window, select Setup > Simulator/Directory/Host from the Affirma DE window and ensure that spectre is selected as the Simulator, then select Simulation > Netlist > Create to generate a spectre netlist.
- Has the Cadence symbol changed since the instance was placed in ADS? Select the instance in the ADS 2002 schematic window, then choose DynamicLink > Instance > Update Instance to Cellview to recreate ADS symbol and remove the `<Cadence_project_dir> / <cell> / directory` to force Dynamic Link to regenerate the netlist. With RFIC Dynamic Link 2001, remove `<lib> _ <cell> _ <view> .*` under `<ads_prj> /networks/`, delete `<Cadence_project_dir> / <cell>` , then delete the instance in the ADS schematic and replace the instance.
- Is the `de_sim.cfg` file under the ADS project directory corrupted?
- Is there any local ADS customization in `$HOME/hpeesof/config/de_sim.cfg`? Search for AEL .
- Is there site-wide ADS customization in `$HPEESOF_DIR/config/de_sim.cfg` or `custom/config/`? For example, the line that reads `" + dynamic_link"` in the `de_sim.cfg` file includes the file `dynamic_link.cfg` .
- If some components require models, are the associated ADS model files included with the `modelLibraryFiles` parameter in the `idfInclude` component?
- For debugging purposes, if the `./idf.log` file appears to be shorter than expected for debugging, is `IDF_DEBUG_MODE` set to TRUE in the UNIX environment?
- If the current GMT in `./idf.log` and `~/CDS.log` do not match, is there a lock on `~/CDS.log` ? Is it `~/CDS.log.1` or `~/CDS.log.2` that is the Cadence log file for this session?
- Are there any suspicious options set in any of the following four Cadence `.cdsenv` files?
 - `<cds> /tools/dfII/etc/tools/ <application> /.cdsenv`
 - `<cds> /tools/dfII/local/.cdsenv`
 - `$HOME/.cdsenv`
 - `./cdsenv`
 The above files contain user preference options.
- If DC Operating Point Back Annotation failed, check the following in the order given:
 - `<ads_prj> /data/ <ads_dsn_name> .ds`
 - `<ads_prj> /psf/ <lib> {} _<cell> {} _<view> /dcOpInfo.info`
 - `<Cadence_proj_dir> / <cell> /adsDL/ <view> /psf/dcOpInfo.info`
 - View the contents of the last file to see if there are any reasonable numbers.
- If DC Back Annotation failed, check the following in the order given:
 - `<ads_prj> /psf/ <lib> {} _<cell> {} _<view> /dcOp.dc`
 - `<Cadence_proj_dir> / <cell> /adsDL/ <view> /psf/dcOp.dc`
 - View the contents of the last file to see if there are any reasonable numbers.

Command Reference for RFIC Dynamic Link

This appendix describes the function of each menu selection provided in Advanced Design System (DynamicLink Menu), the Cadence Schematic window (Tools > ADS Dynamic Link & DynamicLink Menus) and the Cadence CIW (Tools > ADS Dynamic Link Menu item) while using the RFIC Dynamic Link.

ADS Schematic window DynamicLink Menu

Instance > Add Instance of Cellview...

Add a symbol of a Cadence design to the current Advanced Design System Schematic window. For example, see [Adding a Symbol of the Cadence Cellview](#).

Instance > Update Instance of Cellview

Discards <ads_prj> /networks/ <lib> _ <cell> _ <view> .dsn and <ads_prj> /networks/ <lib> _ <cell> _ <view> .ael, regenerates an ADS symbol from the Cadence symbol, and deletes the \$HOME/simulation/ <cell> directory. Deleting the \$HOME/simulation/ <cell> directory forces ADS to recreate a new netlist the next time ADS requires a netlist for a Cadence subcircuit.

Design Variables > Get Design Variables

Add design variables from a Cadence Cellview to an Advanced Design System Schematic window. For example, see [Adding Design Variables](#).

Design Variables > Update Design Variables to Cellview

Update the value of the Advanced Design System schematic design variables to the Cadence Cellview. For example, see [Updating Cadence Design Variables](#).

Add Netlist File Include

Enables duplication of the Definition, Stimulus, and Model Library File include used in Cadence/Affirma. For more information, refer to [Adding Model Files](#).

Top-level Design Netlist

Generate and display the Advanced Design System subnetwork netlist for the Cadence design displayed in a particular Composer window. For example, see [Viewing Netlists from Advanced Design System](#).

Annotate > Annotate DC Solution to Selected Cellview

Display the DC node voltages, generated by the ADS simulator, on the Cadence Schematic Window. For example, see [Performing a DC Simulation](#).

Annotate > Annotate Operating Points to Selected Cellview

Display the DC operating points, generated by the ADS simulator, on the Cadence Schematic Window. For example, see [Annotating DC Operating Points to a Selected Cellview](#).

Set Cadence Project Directory

This is the Cadence project directory and not the ADS project directory. The subcircuit netlist for a Cadence cellview is saved as:

```
<Cadence_Proj_Dir> / <cellName> /adsDL/ <viewName> /netlist/netlist
```

The Cadence subcircuit netlist will be merged with the top-level ADS netlist under the ADS project directory and used for Dynamic Link simulation. DC back annotation data for Dynamic Link is saved in the PSF files under:

```
<Cadence_Proj_Dir> / <cellName> /adsDL/ <viewName> /psf
```

Close Connection

Closes ADS and terminates the link between Cadence and ADS. For example, see [Ending the Session](#).

Cadence Schematic Window DynamicLink Menu

Setup Options ...

Set parameters such as Switch View List and Stop View List. The default values for this dialog are obtained from the configuration file. For more information, consult your Cadence documentation.

New Design ...

Select a new design to include in the Cadence Cellview. Set parameters such as Library Name, Cell Name and View Name. The dialog also offers a browse feature to search for existing designs. For more information, consult your Cadence documentation.

Design Variables ...

Modify the Component Description Format (CDF) information for a component so that it works with ADS. For example, see [Using Design Variables](#). For more information, consult your Cadence documentation.

Save State ...


Save a state of a simulation. The simulation name, variables, model path, outputs and environment options are all available selections. For more information, consult your Cadence documentation.

Load State ...

Load an Analog Artist state. For more information, consult your Cadence documentation.

Subcircuit Netlist

Generate and display the Advanced Design System subnetwork netlist for the Cadence design displayed in a particular Cadence Schematic window. For example, see [Viewing Netlists from the Cadence Schematic Window](#). For more information, consult your Cadence documentation.

 **Note**
If the Dynamic Link pull-down menu does not appear in the Cadence Virtuoso Schematic window, choose Tools > ADS Dynamic Link > Add Dynamic Link menu to all schematic windows in the Cadence Command Interpreter Window (CIW).

CIW Tools > ADS Dynamic Link Menu Item

There are two additional ADS sub-menu items accessible from the Cadence CIW.

Start ADS Dynamic Link

Launch Advanced Design System without opening a Cadence cellview.

Add Dynamic Link menu to all schematic windows

Adds the Dynamic Link menu item to the Cadence schematic window with only the Subcircuit Netlist menu item enabled. This option enables you to create a Cadence subcircuit netlist without launching Advanced Design System.

Glossary for RFIC Dynamic Link

ADS (Advanced Design System)

Advanced Design System 2008

Advanced Design System is an EDA System for high-frequency circuit and system design.

AEL (Cadence Analog Expression Language)

In the Cadence context, AEL is the syntax and API (available in Skill or C) to support full or partial expression evaluation for repetitious circuit simulation.

AEL (ADS Application Extension Language)

This is a C-like interpretive programming language to configure, customize and enhance the Advanced Design System design environment.

Affirma Analog Circuit Design Environment

Cadence's interface for analog circuit design and analysis in versions 4.4.5 and 4.4.6.

bindkeys

Settings used to map individual keystrokes to a particular function within the software.

callback

A function or expression that gets evaluated when certain events occur; for example, clicking on a menu item.

CDF (Component Description Format)

The CDF is Cadence's mechanism to interactively define and evaluate parameters and attributes for individual components and designs.

CIW (Command Interpreter Window)

The CIW is Cadence's command window.

colormap

Indexed color table where each entry is a combination of R, G, and B pixel intensity values for UNIX X-windows display. Table size (number of colors) per software application is limited by the number of display bits per pixel, commonly eight.

DFII (Design Framework II)

Cadence's overall IC design environment.

EDA (Electronic Design Automation)

Software and services that give customers a distinct advantage by improving time-to-market, quality and productivity in the design of electronic products.

GUI (Graphical User Interface)

The interface between the user and the application.

HB (Harmonic Balance Simulation)

An iterative method of analysis that is based on the assumption that for a given sinusoidal excitation, there exists a steady-state solution that can be approximated to satisfactory accuracy using a finite Fourier series.

iPar()

The function used in an AEL expression for a parameter which is a function of another parameter of the same instance. For example, for MOSFET instances we might use $AD=iPar("w")*5u$.

IPC (Inter-Process Communication)

The protocol for passing messages between two or more processes.

OASIS

Open Analog Simulation Integration Socket. The procedural interface for simulator integration into the Cadence simulation environment.

optimization

Mechanism by which a simulator finds the optimal value of a global parameter within a user-supplied range of values.

OS (Operating system)

Such as HP-UX, Solaris, or Win2000.

pPar()

The function used in a Cadence AEL expression for a parameter which is a function of some parameter of the parent instance. For example, for CMOS inverters we might use $W=pPar(wp)$ (where wp is a parent instance parameter) on one of the pull-up FETs, enabling use of the same inverter symbol for different size inverters.

PSF

Parameter Storage Format. This is a Cadence-defined file format for storing complex structured data.

Ptolemy

A design environment that supports simultaneous mixtures of different computation models. Ptolemy, named after the second-century Greek astronomer, mathematician, and geographer, was developed at the University of California at Berkeley. For detailed information, refer to the ADS Ptolemy Simulation documentation.

RF Design Environment (RFDE)

Advanced Design System 2008

RF Design Environment provides a more tightly integrated EDA solution that enables RF/mixed-signal IC designers to simulate their designs directly in the Cadence environment using the ADSSim RF simulator. This enables the RF/MS IC customer to take advantage of complementary features provided by both Agilent Technologies and Cadence Design Systems.

RFIC Dynamic Link (Dynamic Link)

RFIC Dynamic Link (Dynamic Link) for Cadence is an EDA framework integration software product. The product enables both tops-down and bottoms-up design and simulation in Advanced Design System(ADS) using IC designs from the Cadence database. RFIC Dynamic Link is based on IPC rather than data file translation maximizing data integrity and ease of use.

SKILL

Cadence's C/lisp-like interpretive programming language for framework and database integration.

testbench

Top-level schematic used to analyze a sub-circuit using a circuit simulator.

tuning

Mechanism by which a simulator can quickly re-simulate a circuit using new values for a number of parameters without having to re-input the netlist and recreate its data structures.

Virtuoso Analog Design Environment

Cadence's analog design and simulation environment for the Virtuoso custom design platform. It is a task-based environment for simulating and analyzing full-custom, analog, and RF IC designs.