Agilent Technologies

RF Intellectual Property Encoder

August 2005

Notice

The information contained in this document is subject to change without notice.

Agilent Technologies makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Agilent Technologies shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Warranty

A copy of the specific warranty terms that apply to this software product is available upon request from your Agilent Technologies representative.

Restricted Rights Legend

Use, duplication or disclosure by the U. S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DoD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

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

Acknowledgments

Contents

1 Introduction

RF Intellectual Property Encoder 1-1
Major Benefits 1-1
Major Features 1-1
Intended Audience 1-2
General Process 1-2
Use Model 1-3
About ADS Design Kits 1-4
What's in this Manual 1-5

2 Administrative Tasks

Hardware Requirements 2-1
Operating System Requirements 2-1
EDA Framework Requirements 2-1
License Requirements 2-1
Installing and Removing the RF IP Encoder 2-2
Configuring the Environment 2-3
Removing RFIP Encoder Engine 2-3
Important Information about RF IP Encoder 2-4
ADS Versions 2-4
Security 2-4
Technical Support 2-4

3 Getting Started Tutorial

Installing the Tutorial Examples 3-1
Creating an Encoded IP Library 3-2
Symbols in Encoded Libraries 3-8
Running the RF IP Encoder 3-9
Installing the Encoded IP Library 3-13
Installing a Debian Package File 3-13
Installing an Encoded Design Kit 3-14
Opening the Test Schematic Design 3-16
Adding the Encoded Designs 3-17
Editing Design Variables 3-21
Pushing into the Design Hierarchy 3-22
Performing a Simulation 3-22
Displaying Your Results 3-23
Verifying Your Results 3-24
Ending the Session 3-24

4 Encoding Schematic Designs and Netlists

Creating an Encoded IP Library 4-1
Libraries with Data File References 4-5

5 Working with Encoded Designs

Installing Encoded Debian Package Libraries 5-1
Installing Encoded IP Libraries in an Alternate Location 5-2
Viewing Packages 5-5
Using Encoded IP Libraries 5-5
Using Encoded Subcircuits 5-5
Using Encoded Netlists 5-6
Removing Encoded IP Libraries 5-12

6 Troubleshooting

Known Problems and Solutions 6-1

Glossary

Index

Chapter 1: Introduction

Protecting intellectual property (IP) of any form has always been a challenge. The EDA industry has its own unique set of challenges in that high priced encryption technology is not generally justified for protecting IP since most designs change rapidly. While different EDA vendors want to share their designs with their foundries for better efficiency, they typically do not want to share their intellectual property. The *RF Intellectual Property Encoder* (RF IP Encoder) was designed with this in mind and created specifically for vendors who wish to share their designs, but not their intellectual property. Encoded libraries can be distributed either as a debian package installation or as an ADS Design Kit.

RF Intellectual Property Encoder

The RF IP Encoder provides a mechanism for sharing a simulatable Advanced Design System (ADS) schematic design or netlist without divulging your actual design information.

Major Benefits

RF IP Encoder enables you to perform the following:

- protect your intellectual property
- share a simulatable ADS design or netlist without sharing your IP

Major Features

Key features of RF IP Encoder include the following:

- simple encoding process
- ease of packaging and delivery
- ability to include design variables

Intended Audience

This manual describes how to install and configure the RF IP Encoder product and assists both the model-provider and model-user in generating and utilizing encoded designs and netlists. Usage assumes basic familiarity with design and simulation in Advanced Design System.

The audience for this document includes:

- *System Administrators* (or CAD managers). Chapter 2, Administrative Tasks is intended to assist the system administrator or CAD manager with installing the RF IP Encoder product.
- *Model-Providers*. Chapter 3, Getting Started Tutorial is intended to familiarize the model-provider with the entire process of encoding designs and using encoded IP libraries. Chapter 4, Encoding Schematic Designs and Netlists is intended to quickly assist the model-provider in encoding and packaging an IP Library in ADS.
- *Model-Users*. Chapter 5, Working with Encoded Designs is intended to quickly provide instructions on using encoded IP libraries. This information is ordinarily needed by the model-user, however, model-providers may also need to use this when working with their customers. The Chapter 3, Getting Started Tutorial also demonstrates how to use an encoded library however the tutorial requires that you first build a simple library.

General Process

RF IP Encoder is accessible from the ADS **Tools** menu once the product is properly installed. The following steps describe a typical process for encoding a design or netlist with RF IP Encoder:

1. Create your IP Library.
2. Encode your IP Library into either a debian package or an ADS Design Kit.
3. Package your Encoded IP Library.
4. Send the encoded IP to your model user.
5. Model user installs your Encoded IP Library.
6. Encoded IP Library is used within another design.

Use Model

The diagram shown in Figure 1-1 describes the general use model for encoding schematic designs and netlists.

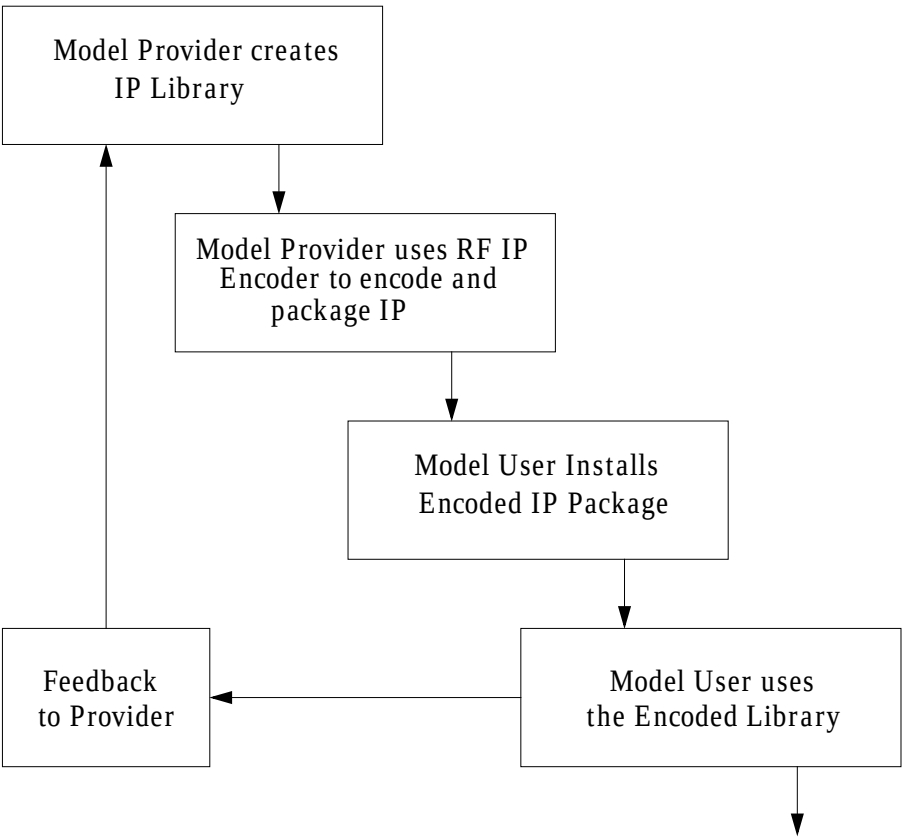


Figure 1-1. RF IP Encoder Use Model

- The Model-Provider groups schematic designs or netlist into a single ADS project.
- The Model-Provider then uses the RF IP Encoder product to create an *encoded IP library*. This library contains information that is the encoded version of the original schematic design(s) or netlist. For convenience, RF IP Encoder bundles

the files that represent the encoded IP library into an ADS package file that can be easily delivered to the Model-User (vendor's customer).

- The Model-User then installs the encoded IP library into their ADS installation. The encoded IP library appears in the library browser and the component palettes. The library is now accessible for use in simulations.

About ADS Design Kits

There are two options for distributing an encoded library. A library can be encoded into a debian package or it can be encoded and packaged as an ADS Design Kit. When you encode your library as an ADS Design Kit, you have the option of creating a zip file of the kit for ease of distribution. The zip file can be easily installed anywhere on a network or computer, it does not have to be installed in the ADS installation directory. Furthermore, installation is made easy because there is a simple user interface within the main window of ADS that automates the installation and configuration. The ADS Design Kit Installation and Setup manual explains how to install, manage and use design kits in ADS.

Another benefit that ADS Design Kits have over the use of a debian package is that you can include documentation about your design kit within the zip file. For example, if certain simulation controllers are required for a particular library, that can be specified in the design kit documentation and included with the design kit. For more information on providing documentation in a design kit, refer to the ADS Design Kit Development documentation.

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 showing you where you can get more information as you need it. This manual contains:

- Chapter 2, Administrative Tasks describes how to install and configure the software.
- Chapter 3, Getting Started Tutorial steps you through the process of creating an encoded IP library using two sample schematic designs, installing this encoded IP library into ADS and using the encoded schematic designs within a test schematic to perform a simulation. This simple Getting Started Tutorial enables you to immediately use the product.
- Chapter 4, Encoding Schematic Designs and Netlists is a quick reference for encoding designs and netlists.
- Chapter 5, Working with Encoded Designs assists the model-user with instructions on installing, viewing, removing and using encoded IP libraries.
- Chapter 6, Troubleshooting provides helpful information on known problems and solutions.

Chapter 2: Administrative Tasks

This chapter describes how to install and configure the software. You may require help from a UNIX or CAD Administrator to complete these tasks.

Hardware Requirements

Refer to the *Advanced Design System Installation* documentation for UNIX or PC systems.

Operating System Requirements

Refer to the operating system requirements for ADS. Although the product may work satisfactorily on later operating system versions, support for these later versions may not be available.

EDA Framework Requirements

The software requires ADS 2005A. Additional module licenses within each framework are also required as described in *License Requirements*.

License Requirements

In addition to ADS schematic and simulator licenses, the RF IP Encoder product license, *ads_encoder*, is required.

Installing and Removing the RF IP Encoder

Before you install the RF IP Encoder engine, ADS must be installed on your computer. For general installation issues, refer to the Advanced Design System installation manuals for UNIX or PC systems.

If you installed a previous version of RF IP Encoder, it is recommended that you remove the software. Instructions for removing the package files are included in "Removing RFIP Encoder Engine" on page 2-3.

The RF IP Encoder is shipped in two parts due to international restrictions on encoding technology. The user interface (UI) is available by using the Custom Installation option and is distributed on the Advanced Design System CD ROM. This enables you to access the UI; however, to be able to encode designs you will also need the proper *ads_encoder* license as well as the RF IP Encoder engine.

Note The *ads_encoder* license is only required for the encoding process. Using an encoded library in simulation does not require a license.

If an attempt is made to encode a design from the UI and the license or RF IP Encoder engine has not been installed, a warning dialog box will appear indicating the missing feature.

The RF IP Encoder engine is not provided on the standard Advanced Design System CD ROM. It must be downloaded from the *Agilent EEsof EDA Technical Support* website. To download the encoder engine software:

1. Access the *Agilent EEsof Knowledge Center* at:

   http://www.agilent.com/find/eesof-knowledgecenter

2. In the *Software Downloads* section, click *ADS*. Note that you are directed to the *Agilent EEsof Knowledge Center Login* page where you are required to enter a Username and Password.

3. After logging in, select your product version.

   Note The ADS version should correspond with the version of RF IP Encoder. For more information, refer to "ADS Versions" on page 2-4.

4. Select the *RF IP Encoder Engine* option.

5. Information on how to download and install the RF IP Encoder engine is provided near the bottom of the page. Select the appropriate version from the list and follow the instructions. The instructions will guide you to save and run the executable.

Configuring the Environment

This section describes how to configure your environment in order to use the hpeesofpkg command. The hpeesofpkg command uses shared libraries that are set in the $HPEESOF_DIR/bin/bootscript.sh script. Before attempting to use the hpeesofpkg commands, you should source the *bootscript.sh* file using one of the following commands:

```
. $HPEESOF_DIR/bin/bootscript.sh          (If using the Korn shell)

sh                                        (If using the C shell)
. $HPEESOF_DIR/bin/bootscript.sh
```

Note The above commands are only necessary if *SHLIB_PATH* for HP-UX or *LD_LIBRARY_PATH* for SunOS does not include the shared libraries required to run hpeesofsim.

Removing RFIP Encoder Engine

You can easily remove the RFIP Encoder engine by entering:

```
hpeesofpkg --remove rfip-encoder
```

Before attempting to use the hpeesofpkg command, refer to "Configuring the Environment" on page 2-3.

Note For further information on working with packages, refer to Chapter 5, Working with Encoded Designs.

Important Information about RF IP Encoder

This section provides important information about the RF IP Encoder version, security and technical support.

ADS Versions

Encoded libraries are version dependent and may not work with different versions of ADS. When using a different version of ADS, the library may need to be re-encoded using that particular version. In general, a library from a current version of ADS should be usable in a future version of ADS, but not in an earlier version.

Security

It is recommended that you share models generated by the RF IP Encoder only under the protection of a Non-Disclosure Agreement and through the use of private media distribution.

Technical Support

Agilent Technologies provides technical support to users of the RF IP Encoder who have also purchased the related support product. Support covers the use of the interface and the encoding process. The users of the RF IP Encoder are responsible for the support of the encoded models that they generate. Due to the sensitive nature of the encoded intellectual property, if a third party encounters problems arising from the use of an encoded model, technical assistance will be provided only to direct users of the RF IP Encoder product.

Chapter 3: Getting Started Tutorial

This tutorial guides you through the process of creating an encoded IP library using two sample schematic designs, installing this encoded IP library into ADS and using the encoded schematic designs within a test schematic to perform a simulation.

Installing the Tutorial Examples

This section provides information on how to install the tutorial example files.

The example projects for this tutorial are provided as *.zap* files. After the RF IP Encoder is installed, these .zap files are located in the installation directory under the *ComponentLibs/examples* subdirectories.

For example, if your installation directory is *C:\ADS2005A* on a Windows system, then the example files are located under *C:\ADS2005A\ComponentLibs\examples*. On a UNIX system, the example files are located in *$HPEESOF_DIR/ComponentLibs/examples*.

1. From the ADS Main window, select **File** > **Unarchive Project**. An *Unarchive Project* dialog box is displayed.

2. For each of the *.zap* files (*EncoderExample.zap* and *EncoderTest.zap*), set the *Unarchive File:* path to unarchive from the appropriate location. Then set the *To Directory:* path to a convenient working directory.

   The first project *(EncoderExample.zap*) contains the designs that will be encoded during the tutorial. The second project (*EncoderTest.zap*) contains test bench circuits for simulations that use the designs from the first project.

Creating an Encoded IP Library

After you have successfully installed the RF IP Encoder example projects, perform the following steps:

1. Open the project *EncoderExample_prj* that was previously unarchived.
2. Open the schematic design *simpleR.dsn* from the networks directory of the project. This file is a basic schematic design of a resistor with specified ports. The design includes a modifiable parameter, *Rx*, that allows the end user to vary the design parameter without viewing the encoded IP.

For more information on working with variables, refer to "*Creating Hierarchical Designs*" in the ADS *"Schematic Capture and Layout"* documentation.

Note Each of your schematic designs must include specified ports (that is, they must be defined as a Parametric Subnetwork) in order to encode the design. Be aware that anything included in your Parametric Subnetwork is not modifiable, with the exception of design variables. This includes, but is not limited to, sources and simulation control components.

Figure 3-1 displays the simple resistor network that includes the parameter *Rx* in the design.

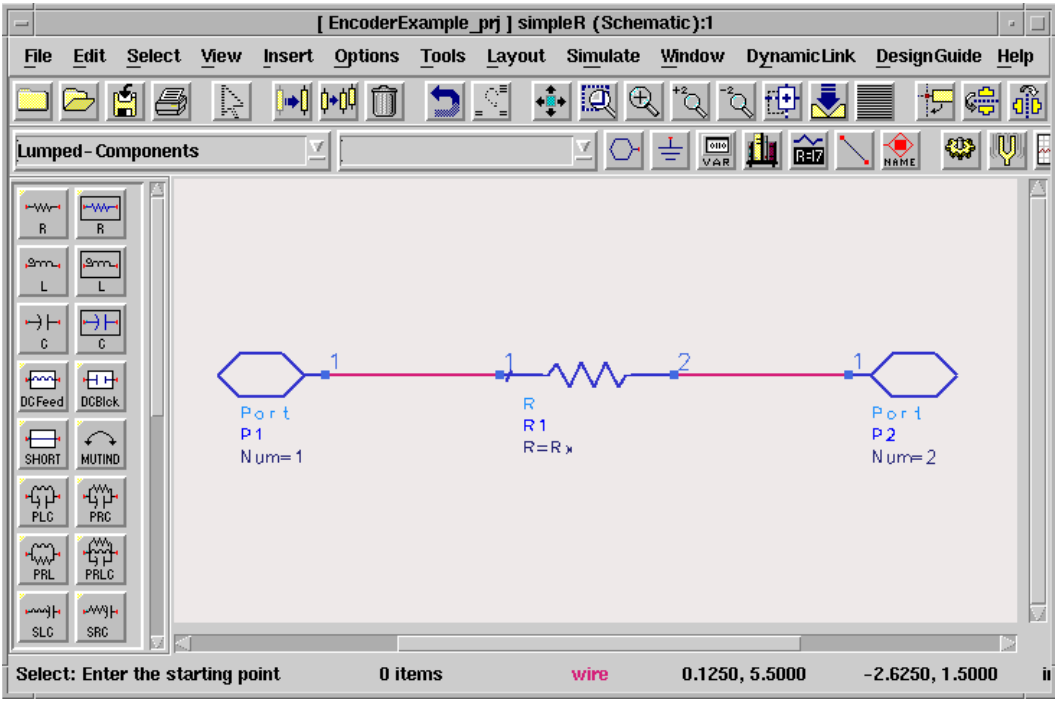


Figure 3-1. Simple Parametric Subnetwork

3. From the Schematic window, select **View** > **Create/Edit Schematic Symbol** to view the schematic symbol that represents your schematic design. Figure 3-2 shows the basic schematic symbol that represents the *simpleR.dsn* schematic design.

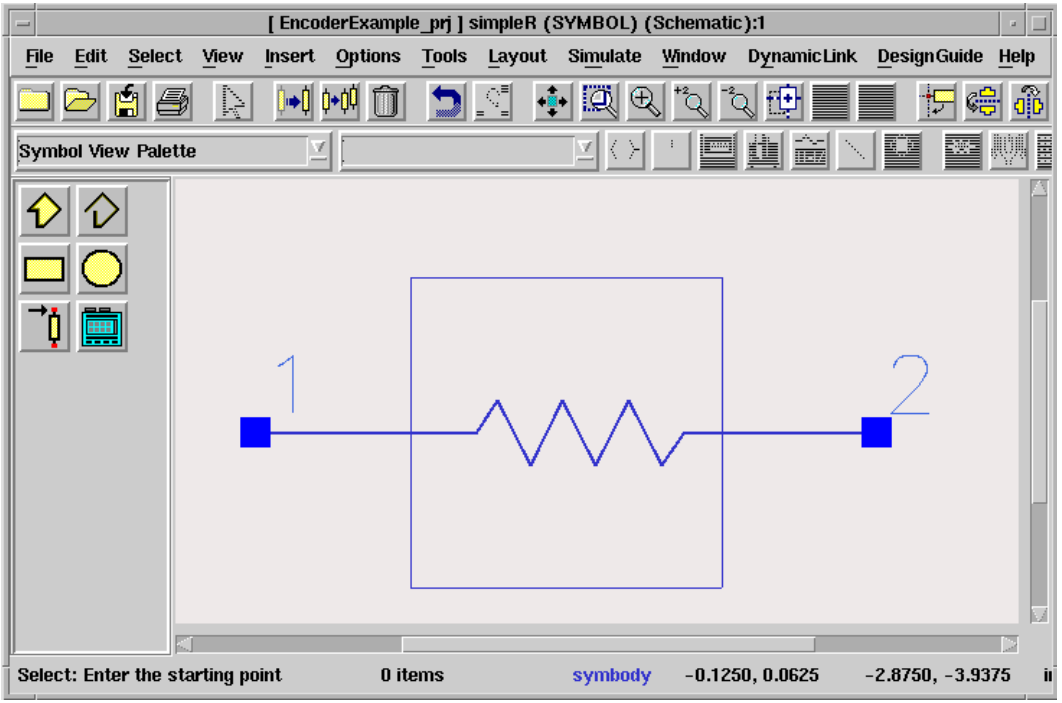


Figure 3-2. simpleR Schematic Symbol

Note If a symbol is not created for your design before you encode, ADS provides a default symbol to represent the encoded component. For more information, refer to "Symbols in Encoded Libraries" on page 3-8.

---

4. Select **File** > **Open Design** and then select the *GilCellMix.dsn* from the list of designs. Figure 3-3 displays the more complex design that will also be included in the encoded IP Library.

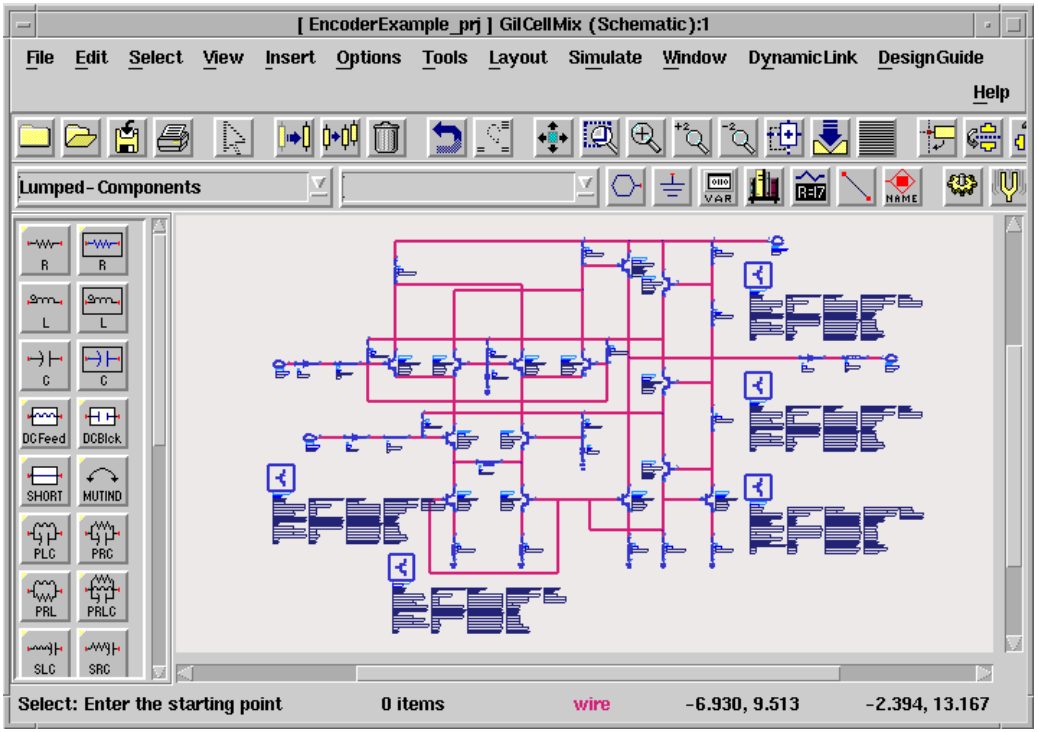


Figure 3-3. Mixer Schematic Design

5. From the Schematic window, select **View** > **Create/Edit Schematic Symbol** to view the schematic symbol that represents the mixer schematic design. Figure 3-4 shows the schematic symbol that represents the mixer schematic design.

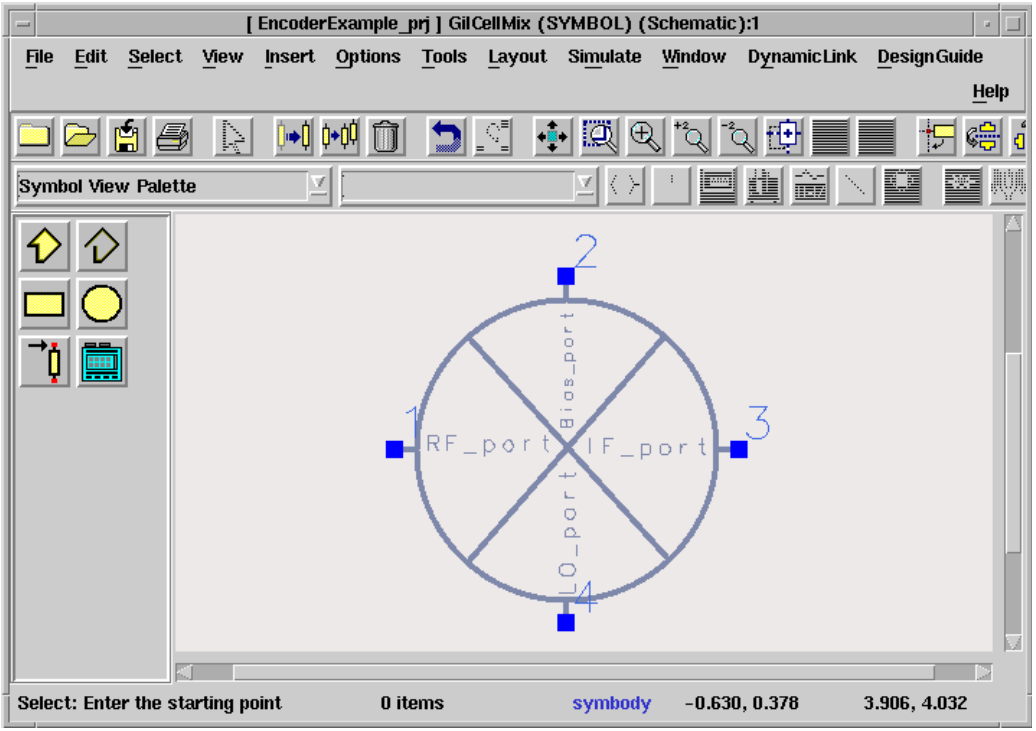


Figure 3-4. GilCellMix.dsn Schematic Symbol

Note If a symbol is not created for your design before you encode, ADS provides a default symbol to represent the encoded component. For more information, refer to "Symbols in Encoded Libraries" on page 3-8.

---

6. Select **View > Create/Edit Schematic** to return to the schematic view.
7. Select **File > Design Parameters**. The *Design Parameters* dialog box appears. Type in a description in the *Description* field. For this tutorial, use *Encoded Gilbert Cell Mixer* as shown in Figure 3-5. This description will be the Component Library description for the encoded component.
8. Click **OK** to close the Design Parameters dialog box.
9. Save the design to save any changes entered in the Design Parameters dialog.

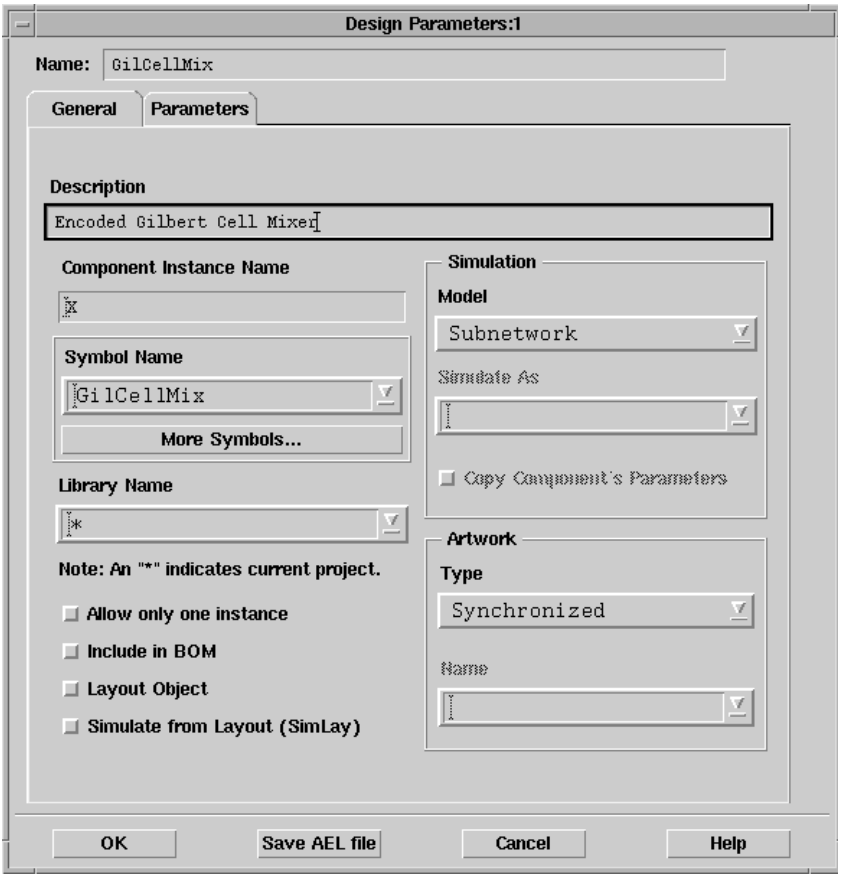


Figure 3-5. Setting the Encoded Component Description

Symbols in Encoded Libraries

There are a few different ways to define the symbol to be used with each library component. Be sure to note the associated limitations related to using symbols in design kits or encoded libraries.

- Using Pre-defined Symbols - Create or open a circuit in the schematic editor. From the schematic view of the circuit, select **File > Design Parameters**. In the *General* tab, define the symbol name by selecting a common pre-defined symbol from the drop-down list or by selecting *More Symbols*, which gives you access to all pre-defined library symbols in ADS, sorted by category. This method works best for encoded libraries and design kits.
- Schematics and Custom Symbols in One Design File - Create or open a circuit in the schematic editor. From the schematic view of the circuit, select **View > Create/Edit Schematic Symbol**. Draw a custom symbol and save the design. This will save the symbol information directly with the schematic design.

  This method works well for encoded designs but has limitations if the encoded design is made into a design kit. If the design kit is enabled at the project level, the user will get an error message when selecting the component from the palette and will not be able to place the component. If the design kit is enabled at any other level, this problem will not occur. Simply closing and re-opening the project will fix this.
- Schematics and Custom Symbols in Separate Design Files - From the schematic view of a new, empty design, select **View > Create/Edit Schematic Symbol**. Draw a custom symbol in the symbol view and save the file without entering anything in the schematic view.

  Create or open a circuit in the schematic editor. From the schematic view of the circuit, select **File > Design Parameters**. In the General tab, type the symbol name into the Symbol Name field, or select *More Symbols* and browse to select the name of the design file. The symbol name should be stored with the .dsn extension and no path (ex: myR_SYM.dsn).

Running the RF IP Encoder

To encode your designs using the RF IP Encoder perform the following steps:

1. From the Schematic window, select **Tools > Encode Designs**. A dialog box appears asking your permission to close all designs and schematic windows. Select *No* if you have unsaved changes that you need to save. If you select *Yes*, the *Create Encoded Library* dialog box appears.

2. Use the *Library Contents* section to select which schematic designs you want to encode. In the *Available Designs and Netlists* list box, highlight each individual schematic design you want to include in your library and then click **Add**. This adds each selection and displays the selected schematic designs in the *List of Designs and Netlists to Encode* list box.If you hold down the **CTRL** key while clicking the mouse you can make multiple selections. Clicking the mouse while holding down the **SHIFT** key will enable you to select everything between the first highlighted item and the last one selected. Double clicking the mouse on a selection will automatically move a single item to the Encode list box. Individual schematic designs can be removed from the *List of Designs and Netlists to Encode* by clicking **Remove**.

Note To include your own netlists in the *Available Designs and Netlists* list, place your netlists under the current project directory, *networks* subdirectory with a *.net* extension. For example, <*your_project*>/networks/<*your_netlist*>.net. Do not redistribute models from an outside source without permission.

---

3. By default, each encoded library on the palette is represented by an image of a padlock. To set a custom bitmap that will be more meaningful, first select the component in the right hand column labeled *List of Designs and Netlists to Encode*. This will make the *Set Bitmap* button available for selection. This is only available for schematic designs, not netlist files.

   Select the mixer design file first then click the **Set Bitmap (dsn only)** button and the bitmap selection dialog will appear. Some bitmaps that can be used are located in $HPEESOF_DIR/circuit/bitmaps as well as with the *DesignGuide Developer Studio*.

   Depending upon whether you are running on UNIX or the PC, select the appropriate Bitmap File entry field and enter the path:

Unix Bitmap File

<*Install Dir*>`/designguides/projects/dgstudio/ui/bitmaps/adsbmps/MIXER.BMP`

PC Bitmap File

<*Install Dir*>`\designguides\projects\dgstudio\ui\bitmaps\adsbmps\MIXER.BMP`

To use the browser, click the **Browse** button and change the filter to upper case BMP. Select the desired bitmap and close the bitmap selection dialogs.

Select the resistor component and enter the bitmap R.BMP from the same directory listed above. When a library is created, a copy of the bitmap will be made and a new name assigned to it, to match the name of the component. This file will be installed with the library, so the end user will not be required to have the complete set of ADS bitmaps installed if they are just using the library. If you do not select a bitmap for a particular platform the default padlock will be used when running ADS on that platform.

4. Use the *Library Information* section to define your new encoded IP library. The name *at_mixlib* should be used for this tutorial. Enter it into the *Library Name* field.

Note Valid library names contain only alphanumeric and underscore characters. All other punctuation characters are illegal. The first character must be alphabetic or an underscore. Names starting with the string *untitled* are not allowed.

Hint When selecting a library name for your custom library, choose a name that is unique and descriptive. This will help avoid name conflicts with other IP providers.

---

5. Enter a version number into the *Version Number* field. The default version value is *1*.

6. Enter a clear description of your library into the *Library Description* field.
7. The current working directory is the default for the *Destination Path* field. If you wish to change the path, enter a new path in the *Destination Path* field or click **Browse** to view and choose a currently available path. The encoded library package will be created in the specified directory.
8. Depending upon the package method that you would like to use, either select *Debian* for the Encoding Format or *Design Kit with no .zip file*. If you choose to create a zip file of an encoded Design Kit, refer to the ADS Design Kit Installation and Setup documentation for instructions on how to install a design kit from a zip file.
9. Enter your *Name*, *Company* and *Email Address* in the respective field within the *Encoded By* section. This will provide contact information about the person creating the encoded library.
10. After all of the appropriate information is entered, select the **Encode** button in the lower left-hand corner of the dialog box. An *Encoding Status* box appears.
11. Click **OK** to dismiss the *Encoding Status* dialog box.

Once the encoding process is complete, you should have one of the three results described below:

- A debian package file (extension = *.deb*).
- An ADS Design Kit with a zip file (extension = *.zip*).
- An ADS Design Kit directory structure without a zip file. This will be created under the path that you specified as the *Destination Path*.

For this example, we have created an encoded library called *at_mixlib.deb*.

For a debian package file (extension = *.deb*), a text file with information and instructions on working with a specific encoded design or netlist is automatically generated when an encoded IP library is created. This text file is in the form <*library_name*>.txt, where *library_name* is the unique name of your encoded IP library.

Important The debian package text file must be sent to the model-user along with your encoded IP library to help with package management issues.

For a Design Kit, information about the kit can be found in the design kits *doc* directory in a file called *about.txt*.

Installing the Encoded IP Library

The method used for installing your encoded IP library will depend on how the library was packaged.

- If your encoded IP library was packaged as a debian, the file can be easily installed, viewed or removed by following the information provided in "Installing a Debian Package File" on page 3-13. Note that you must have read/write permission in your ADS installation directory in order to install a debian package.

Note Before installing an encoded library, be sure to exit ADS. Also ensure the *$HPEESOF_DIR* environment variable (%HPEESOF_DIR% for PC) is set to your installed location of Advanced Design System and that *$HPEESOF_DIR/bin* (%HPEESOF_DIR%\bin for PC) is in your path. For more information, refer to "Installing and Removing the RF IP Encoder" on page 2-2.

- If your library was packaged as an ADS Design Kit, follow the steps provided in "Installing an Encoded Design Kit" on page 3-14.

Installing a Debian Package File

To install the encoded IP library package file, open a terminal window and change to the directory where your encoded library resides. Note that you must have read/write permission in your ADS installation directory in order to install a debian package. In the terminal window, enter the following:

```
hpeesofpkg -i at_mixlib.deb
```

Note To install an encoded IP library in an alternate location, use the following command: `hpeesofpkg -x <package_file.deb>`. For more information on using the *hpeesofpkg -x* command, refer to "Installing Encoded IP Libraries in an Alternate Location" on page 5-2.

Viewing the Package File

After installing the package, view the list of currently installed packages by typing the following in a terminal window:

```
hpeesofpkg --list
```

A list similar to the one below is displayed. Note that the encoded library *at-mixlib* is now in the list of installed packages (only partial list is shown).

```
Desired=Unknown/Install/Remove/Purge
| Status=Not/Installed/Config-files/Unpacked/Failed-config/Half-installed
|/ Err?=(none)/Hold/Reinst-required/X=both-problems (Status,Err:
uppercase=bad)
||/ Name            Version        Description
+++-===============================================================
ii  antennas-prop   2005A.day   Antennas and Propagation Library
ii  antennas-prop-d 2005A.100   Online Documentation for antennas-prop
ii  api             2005A.day   api files
ii  at-mixlib       1           Encoded Gilbert Cell Mixer
ii  basic           2005A.day   basic files
ii  behavioralmodel 2005A.100    Example files for BehavioralModels
```

The name of the package in the list may be different than the name of the package file. Generally speaking, the encoded package will have the same name as the file, except that characters are converted to lower-case (and underscores become dashes). When in doubt, the name of the package contained within a given package file can always be found by using the following hpeesofpkg command.

```
hpeesofpkg --info <package_filename>
```

The following command can be used to examine the contents of the package file:

```
hpeesofpkg --contents <package_filename>
```

For further information on working with packages, refer to Chapter 5, Working with Encoded Designs.

Installing an Encoded Design Kit

To install an encoded design kit from the ADS *Main* window:

1. Choose **DesignKit > Install Design Kits**. The *Install ADS Design Kit* dialog box appears.

2. In step one of the dialog box (*Unzip Design Kit*), click the **Unzip Design Kit Now** button to unzip a design kit. The *Unzip ADS Design Kit* dialog box appears enabling you to set the source and target directories for your design kit. Use the **Browse** buttons in the dialog to quickly locate the source and target. Click **OK** once the source and target are defined. The design kit zip file is unzipped in the specified location.

3. In step two of the dialog (*Define Design Kit*), notice that the Path, Name, Boot File, and Version are now all displayed automatically.

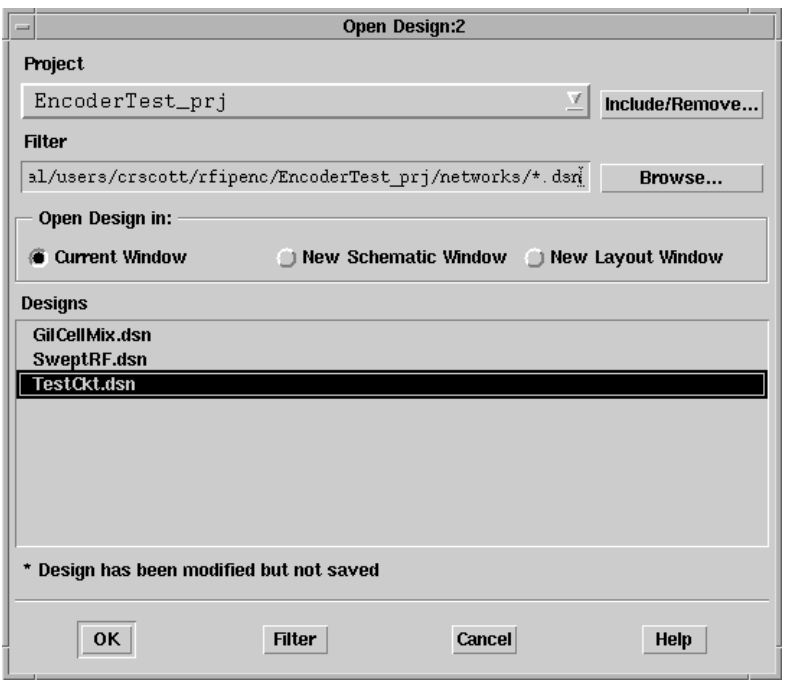
4. In step three of the dialog (*Select Installation Level*), select USER LEVEL for this tutorial.
5. Click **OK** in the Install ADS Design Kit dialog box. A confirmation message indicating that the design kit was successfully installed should appear.

For more information on installing design kits, refer to the ADS Design Kit Installation and Setup documentation.

Opening the Test Schematic Design

Use the following steps to open a test schematic design.

1. Restart ADS and attach to the *EncoderTest_prj*.
2. From the Schematic window, select **File** > **Open Design** to display the *Open Design* dialog box.



3. Click *TestCkt.dsn* from the *Designs* list. The *TestCkt.dsn* schematic already contains simulation components used for this example.

4. Click **OK** to open the *TestCkt.dsn* schematic in the ADS Schematic window.

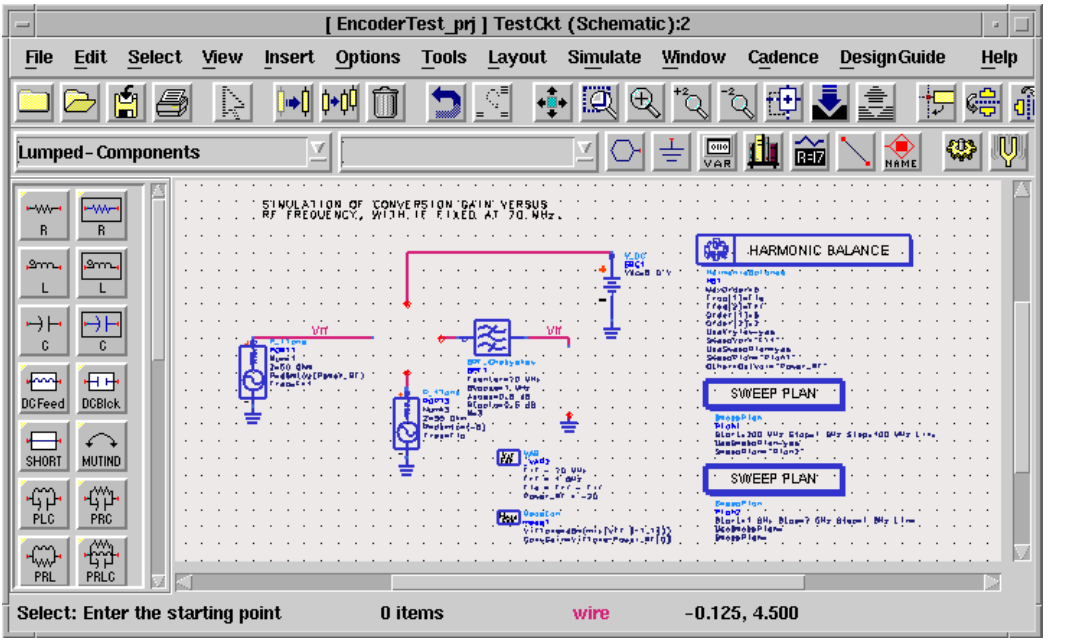


Figure 3-6. Test Circuit Schematic

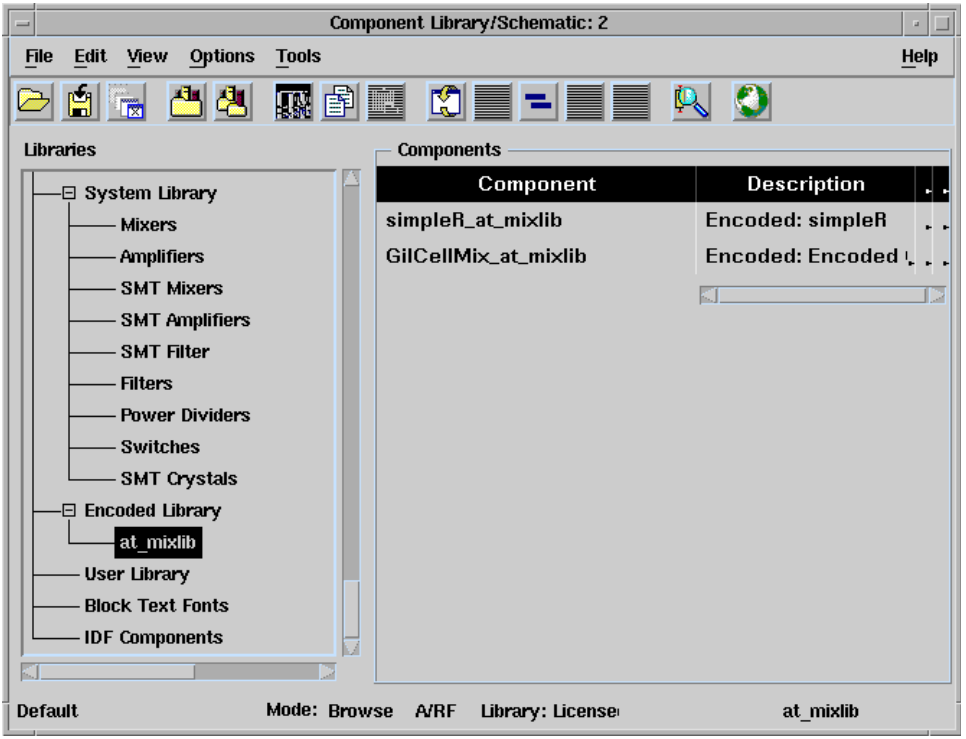
Adding the Encoded Designs

Newly encoded libraries will appear in the Library Browser under the category *Encoded Designs*.

Use the following steps to add the encoded designs into the Advanced Design System Schematic window.

1. From the Schematic window, select **Insert > Component > Component Library**.

   A *Component Library Schematic* dialog box appears, enabling you to select the components in the encoded IP library.



2. In the *Component* list box of this dialog, click the component you want to include in your schematic. In this case, click *GilCellMix_at_mixlib*.

Note *GilCellMix_at_mixlib* and *simpleR_at_mixlib* are the newly encoded designs in the *at_mixlib* encoded IP Library.

If you want a description name other than the default, the name must be specified while creating the encoded library. See "Creating an Encoded IP Library" on page 3-2 for more information.

Note Only encoded designs will appear in the *Component* list. Encoded netlists will not be listed.

---

3. An instance of the encoded design is attached to the cursor for you to place. In the ADS Schematic window, click the left mouse button to place the symbol as shown in Figure 3-7.

4. After the mixer is placed in the schematic, choose the *Cancel Command And Return To Select Mode* icon to proceed with the next step. 

Note While the *simpleR_at_mixlib* encoded design could be added to the schematic in the same manner as above, the following describes a different method.

---

5. Select the Component palette selection for *at_mixlib*. Click the *simpleR_at_mixlib* to place it on the schematic.

Note Only encoded designs will appear in the *component palette.* Placing the cursor over an encoded design in the palette displays a description of the encoded design. Encoded netlists will not appear in the palette.

---

6. After the resistor is placed in the schematic, choose the *Cancel Command And Return To Select Mode* icon to proceed with the next step. You will need to rotate the *simpleR_at_mixlib* component and place it in the circuit output as shown in Figure 3-7. The schematic is now complete and ready for simulation.

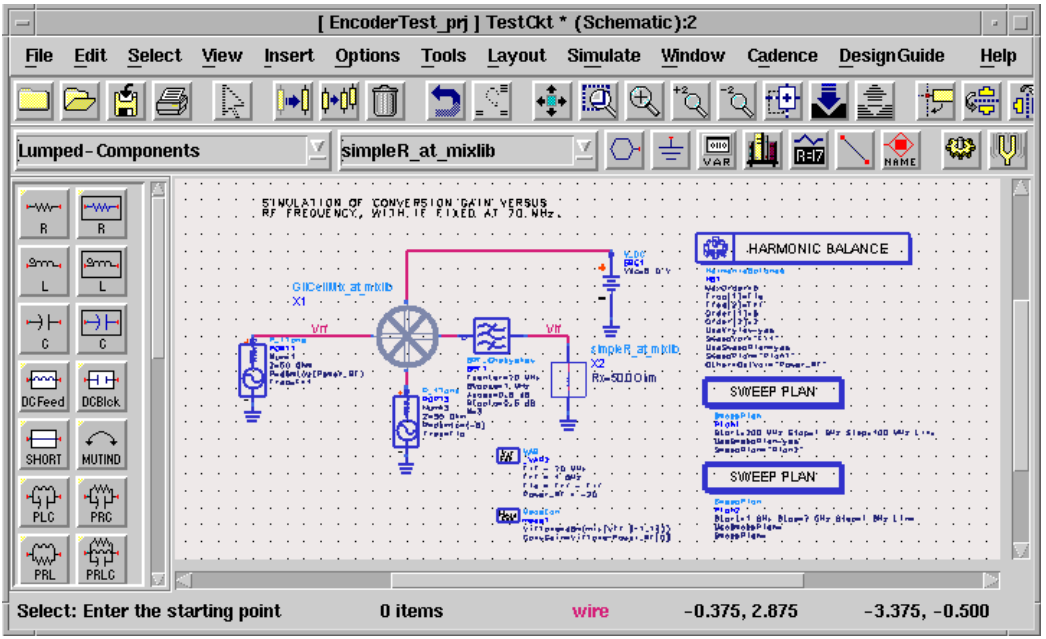


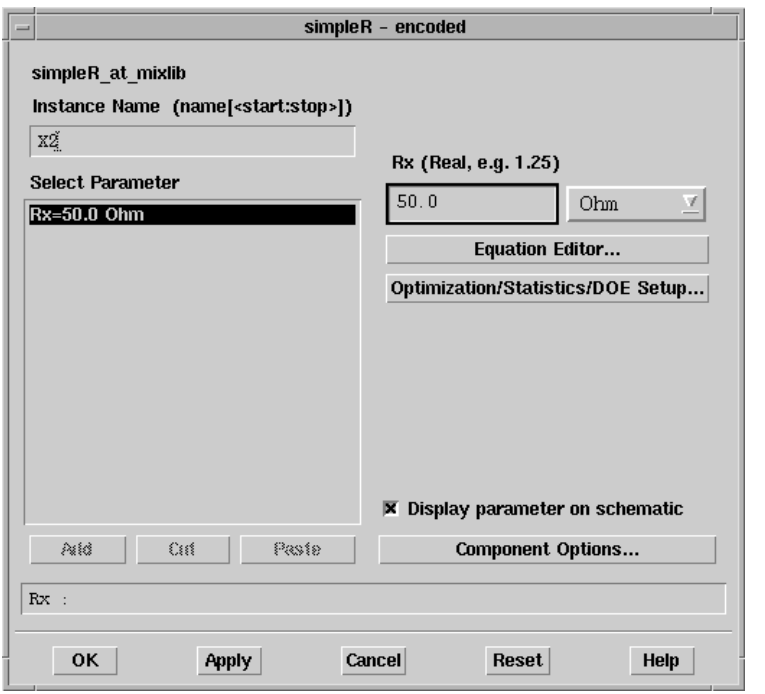
Figure 3-7. Adding the Encoded Designs

Editing Design Variables

Use the following steps to edit a design variable in the Advanced Design System schematic.

1. Select the X2 component *simpleR_at_mixlib*.
2. Select **Edit** > **Component** > **Edit Component Parameters** to edit the value of *Rx* or choose the *Edit Component Parameters* icon.

3. A dialog box appears enabling you to modify the value of Rx.



Pushing into the Design Hierarchy

Use the following steps to view a design deeper in the Advanced Design System schematic hierarchy.

1. In the ADS Schematic window, click the component you want to push into.
2. Choose the *Push Into Hierarchy* icon. 

Note If you select a component that is encoded, you will not be able to view the contents of the design.

---

3. Try to push into the *GilCellMix_at_mixlib* and *simpleR_at_mixlib* subnetworks. Note that you are not able to view the contents of the encoded designs and that an error dialog is displayed.

Performing a Simulation

Use the following steps to perform a simulation on the *TestCkt* schematic and view the results in a plotted Data Display.

1. From the Schematic window, select **Simulate > Simulate** or choose the *Simulate* icon. This netlists the top-level Advanced Design System 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 a few moments. Check this status window to see if any errors occurred during netlisting or simulation.

---

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

Displaying Your Results

Use the following steps to view the results of your simulation in a plotted Data Display.

1. Select *TestCkt* from the Dataset drop-down list if it is not already displayed.
2. From the Data Display window, select **Insert** > **Plot**, move the frame to an appropriate location within the window and click. The *Plot Traces & Attributes* dialog box appears. Similarly, you can choose the *Rectangular Plot* icon to drag and drop the plot frame.

3. Select *ConvGain* and then click **Add**. Select *dB* from the dialog box then click **OK**.
4. Click **OK** again to view the *Data Display*. The plot below shows the Conversion Gain, at 1.0GHz to be approximately 21.59 dB.

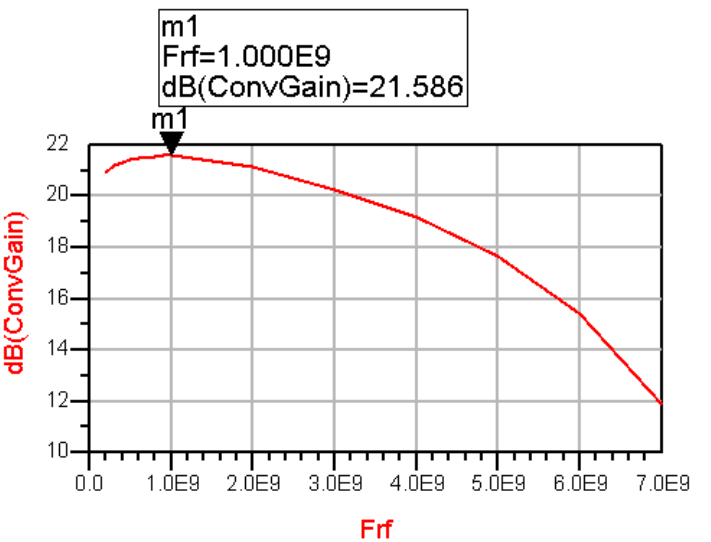


Figure 3-8. Data Display of Conversion Gain for TestCkt

For more information on displaying results, refer to the Data Display documentation.

Verifying Your Results

You can also verify the results of your simulation by running a simulation using the original schematic and comparing the results. The original schematic is called *SweptRF* and is also saved in *EncoderTest_prj*. This design uses a subcircuit that has not been encoded. Note that you can push into the Gilbert Cell Mixer in this design.

Ending the Session

Use the following steps to exit ADS.

1. Choose the Advanced Design System Schematic menu option **File** > **Exit Advanced Design System**. This displays a *Save Changes* dialog box that asks: `Do you want to save changes to design?`
2. Select **No To All** from the dialog box.

Congratulations... you have now successfully completed the Tutorial.

Chapter 4: Encoding Schematic Designs and Netlists

This section is intended to be a quick reference for encoding schematic designs and netlists. Most of this material is repeated in Chapter 3, Getting Started Tutorial.

Creating an Encoded IP Library

Follow the steps below to create an encoded IP library from one or more schematic designs and/or netlists.

1. From the schematic window, select **Tools > Encode Designs**. A dialog box appears asking your permission to close all designs and schematic windows. Select *No* if you have unsaved changes that you need to save. If you select *Yes*, the *Create Encoded Library* dialog box appears.
2. In the *Available Designs and Netlists* field, highlight each individual schematic design or netlist you want to include in your library and then click **Add**. This adds each selection and displays the selected schematic designs and netlists in the *List of Designs and Netlists to Encode* field. Similarly, individual schematic designs or netlists can be removed from the *List of Designs and Netlists to Encode* by clicking **Remove**.

   To include your own netlists in the *Available Designs and Netlists* list, place your netlists in the *networks* subdirectory of the current project directory. For example:

   <your_project>/networks/<your_netlist>.net

   All netlist files must have a *.net* extension to be recognized by the IP Encoder. To find netlists which are not in the *networks* directory, click the **Browse Netlists** button.

Note Encoded designs will appear on the palette and library browser. Encoded netlists will *not* appear on the palette and library browser. For more information, refer to "Using Encoded Netlists" on page 5-6.

3. By default, each encoded library on the palette is represented by an image of a padlock. To set a custom bitmap that will be more meaningful, first select the component in the right hand column labeled *List of Designs and Netlists to Encode*. This will make the *Set Bitmap* button available for selection. This is only available for schematic designs, not netlist files.

   With a design file selected, click the **Set Bitmap (dsn only)** button and the bitmap selection dialog will appear.

Enter the path to a bitmap. Libraries of ADS palette bitmaps are contained in:

```
$HPEESOF_DIR/circuit/bitmaps
```

and

```
$HPEESOF_DIR/designguides/projects/dgstudio/ui/bitmaps/adsbmps
```

To use the browser, click the **Browse** button. Some of the bitmaps have upper case names and some have lower case names. To see all selections, change the filter to *.*. Select the desired bitmap and close the bitmap selection dialogs.

When a library is created, a copy of the bitmap will be made and a new name assigned to it, to match the name of the component. This file will be installed with the library, so the end user will not be required to have the complete set of ADS bitmaps installed if they are just using the library.

4. Use the *Library Information* section to define your new encoded IP library. Choose a unique name for your encoded IP Library and enter it into the *Library Name* field.

Note Valid library names contain only alphanumeric and underscore characters. All other punctuation characters are illegal. The first character must be alphabetic or an underscore. Names starting with the string *untitled* are not allowed.

Hint When selecting a library name for your custom library, choose a name that is unique and descriptive to avoid possible confusion with other encoded libraries.

5. Enter a version number into the *Version Number* field. The default version value is *1*.
6. Enter a clear description of your library into the *Library Description* field.
7. Manually enter a path in the *Destination Path* field or click **Browse** to view and choose a currently available path. The encoded library package will be created in the specified directory.
8. Select an encoding format. A design kit style library is the recommended format. A debian format library must be installed manually in the ADS installation directory, which requires write permission in an area where most users do not have permission.

   A design kit style library has a lot more flexibility and support in ADS. It can be installed from the *DesignKit* menu in the ADS Main window, and can be installed anywhere on the user's machine. Select *Design Kit with .zip file* if your library will be complete as is or if you would like to test it immediately. If you would like to add more capability or customization, consult the Design Kit Development documentation for detailed instructions and examples, as well as instructions on how to manually package the file for distribution. Whether or not you select the .zip file option, the complete library in unzipped format will remain on your machine in the directory you specify.
9. Enter your *Name*, *Company* and *Email Address* in the respective field within the *Encoded By* section. This will provide contact information about the person creating the encoded library.
10. After all of the appropriate information is entered, select the **Encode** button in the lower left-hand corner of the dialog box. An *Encoding Status* box appears.
11. Click **OK** to dismiss the *Encoding Status* dialog box.

Once the encoding process is complete, you will have a debian package file (extension = *.deb*), ADS Design Kit with a zip file (extension = *.zip*), or an ADS Design Kit directory structure without a zip created in the path you specified as the *Destination Path*.

For a debian package file (extension = *.deb*), a text file with information and instructions on working with a specific encoded design or netlist is automatically generated when an encoded IP library is created. This text file is in the form <*library_name*>.txt, where *library_name* is the unique name of your encoded IP library.

Important The debian package text file must be sent to the model-user along with your encoded IP library to help with package management issues.

For a Design Kit, information about the kit can be found in the design kits *doc* directory in a file called *about.txt*.

Libraries with Data File References

The RF IP Encoder does not support the encoding of file-based components (such as S-parameter data file components). The Encoder will automatically move any referenced data file components into the Design Kit circuit/data directory. For Debian installations, the referenced data files will be copied to the ComponentLibs/data directory. The encoded library will reference these un-encoded data files in these new locations.

If encoded data files are required for the release of your design, please contact your Agilent Technologies technical support representative. For more information, refer to http://eesof.tm.agilent.com/support/.

Chapter 5: Working with Encoded Designs

This section will assist the model-user (typically an RF Board designer) with instructions on installing, viewing, using and removing encoded IP libraries. Your encoded IP library is created as either a standard ADS Add-on package or an ADS Design Kit. For information on installing or removing ADS Design Kits, refer to the ADS Design Kit Installation and Setup documentation. Before attempting to use the hpeesofpkg command, refer to "Configuring the Environment" on page 2-3.

Note Exit ADS *before* installing an encoded library that has been packaged as a debian.

Installing Encoded Debian Package Libraries

To install an encoded IP library package (.deb file) from a terminal window, type the following command:

```
hpeesofpkg -i <package_file.deb>
```

where <*package_file.deb*> is the name of the file containing the encoded IP library you wish to install (including the *.deb* extension). This will install the library in the *$HPEESOF_DIR/ComponentLibs* directory. Note that you are required to have write permission in the ADS installation directory to complete this process.

If you are an advanced user, you can refer to "Installing Encoded IP Libraries in an Alternate Location" on page 5-2 for instructions on installing your library in a location other than the *$HPEESOF_DIR/ComponentLibs* directory.

Note If a new version of an encoded IP library exists, you don't have to remove the old one to install the new one, the packager will automatically update over the old installation.

Installing Encoded IP Libraries in an Alternate Location

The `hpeesofpkg -x <package_file.deb>` command is used to install encoded IP libraries in an alternate location (a location other than the *$HPEESOF_DIR/ComponentLibs* directory).

Installation

1. Copy the encoded library package (*.deb*) into the directory in which the library is to be installed. For example, *<My_Lib_Directory>/<package_file>.deb*.
2. Open a terminal window and change to the installation directory.
3. At the UNIX prompt, enter the following command:

   `hpeesofpkg -x <package_file>.deb ./`

   This will install the files and associated sub-directories within `<package_file>.deb` into the current directory.
4. Update your *$HOME/hpeesof/config/de_sim.cfg* configuration file with the variable *USER_DSN_PATH* to point to the location where your encoded symbol files are stored. For example,

   `USER_DSN_PATH=<path to your encoded library>/circuit/symbols`

   Note If *USER_DSN_PATH* has already been defined, append the path to USER_DSN_PATH using a colon (:) as a separator to include the location where your encoded symbol files are stored.

   For more information on the USER_DSN_PATH variable, refer to *Typical Variables > Variables in de.cfg, de_sim.cfg > Table 1-5* in Chapter 1 of the Customization and Configuration documentation.

Configuration

1. If they do not already exist, create the following directories and sub-directories under your home directory:
   - hpeesof
   - hpeesof/config
   - hpeesof/circuit
   - hpeesof/circuit/config

Note You can find your home directory by viewing the *HOME* environment variable.

2. Copy the file *hpeesofbrowser.cfg* from the *config* sub-directory under the ADS installation directory (*$HPEESOF_DIR*) into *hpeesof/config* directory under your home directory.
3. Open *hpeesofbrowser.cfg* in a text editor.
4. Search for the line which starts with *HPVENDORLIB_BROWSER_PATH*.
5. Add ;<*My_Lib_Directory*>/ComponentLibs/records to the end of the line.
6. Save and close the edited *hpeesofbrowser.cfg* file.
7. Copy the *ADSlibconfig* file from the circuit/config sub-directory under the ADS installation directory (*$HPEESOF_DIR*) into *hpeesof/circuit/config* directory under your home directory.
8. Open *ADSlibconfig* in a text editor.
9. Type the following line at the end of the file:

   <*package_file*>
   <*My_Lib_Directory*>/ComponentLibs/models/<*package_file*>.library

Note Place the above text on a *single* line within the *ADSlibconfig* file.

10. Save and close the edited *ADSlibconfig* file.
11. Copy the *de_encode_master* file from *ComponentLibs/ael* sub-directory under the ADS installation directory (*$HPEESOF_DIR*) into *ComponetLibs/ael* under your local encodedlibs directory, <*My_Lib_Directory*>.
12. Open *de_encode_master.ael* in a text editor.
13. Type the following line at the end of the file:

    load("<*My_Lib_Directory*>/ComponentLibs/ael/<*package_file*>_encode_def.ael")

14. Save and close the edited *de_encode_master* file.

15. Copy the *de_sim.cfg* file from *config* sub-directory under the ADS installation directory (*$HPEESOF_DIR*) into *hpeesof/config* under your home directory.
16. Open *de_sim.cfg* in a text editor.
17. Search for the line which starts with *ANALOG_RF_SIMULATOR_AEL*.
18. Replace `{$HPEESOF_DIR}/ComponentLibs/ael/de_encode_master` with `<`*My_Lib_Directory*`>/ComponentLibs/ael/de_encode_master`.
19. Within *de_sim.cfg*, search for the line which starts with *DSP_SIMULATOR_AEL*.
20. Replace `{$HPEESOF_DIR}/ComponentLibs/ael/de_encode_master` with `<`*My_Lib_Directory*`>/ComponentLibs/ael/de_encode_master`.
21. Save and close the edited *de_sim.cfg* file.

Generate Files

1. From the your local encoded library directory, <*My_Lib_Directory*>, type the following:

   `hpedlibgen -in ComponentLibs/ael/<`*package_file*`>_encode_item.ael -out ComponentLibs/records/<`*package_file*`>.idf`

2. Press **ENTER** to generate files.

Viewing Packages

To view the list of currently installed packages, including your encoded IP libraries, open a terminal window and type the following:

```
hpeesofpkg --list
```

A list similar to the one below will be displayed (only partial list is shown).

```
Desired=Unknown/Install/Remove/Purge
| Status=Not/Installed/Config-files/Unpacked/Failed-config/Half-installed
|/ Err?=(none)/Hold/Reinst-required/X=both-problems (Status,Err:
uppercase=bad)
||/ Name            Version        Description
+++-===============================================================
ii  antennas-prop   1.9.1       Antennas and Propagation Library
ii  at-mixlib       1           1) Gilbert Cell Mixer, 2) Resistor
ii  cdma            1.9.1       CDMA
ii  circuit-encoder 1           Circuit Encoder
ii  controls-displa 1.9.1       Interactive Controls and Displays
ii  fixpt-analysis   1.9.1       Fixed Point Analysis
```

Using Encoded IP Libraries

Encoded IP libraries can be used in Advanced Design System just like any other library with two exceptions. If you are using an encoded subcircuit from a schematic design, you will not be able to push into its hierarchy. Also, there are special instructions for including the contents of an encoded netlist file in a simulation. For more information, refer to "Using Encoded Netlists" on page 5-6.

Using Encoded Subcircuits

To use an encoded subcircuit, first select the library to display in the component palette on the left hand side of the schematic window. Encoded components may be

displayed with a generic padlock icon. To identify the name of the component, hold the cursor over the padlock icon until the component name appears. Alternatively, you can select components in an encoded library from the library browser by choosing **Insert > Component > Component Library** in the schematic window or simply click the library tool button.

If an encoded component design has variable parameters, you are able to change the value of certain components within the design. This information is typically provided by the Model-Provider. For more information on variable parameters, refer to "*Creating Hierarchical Designs*" in the *Schematic Capture and Layout* documentation.

Using Encoded Netlists

Chapter 4, Encoding Schematic Designs and Netlists, described how to create an encoded library out of schematic subcircuits. It is also possible to create a library when you have an ADS netlist. This is a typical scenario for a foundry creating a design kit. In this section, you will learn how to encode a netlist that contains subcircuits, and how to use the encoded library in a simulation. It is also possible to encode a netlist that contains model cards and variables. For more information about this topic, refer to the *Design Kit Development* documentation.

Note The following steps assume that you have completed Chapter 3, Getting Started Tutorial and created a design kit. Note that this is an advanced topic and will require manual editing of AEL and other text files.

To create a netlist to encode:

1. Open the *EncoderExample_prj* project that was used in "Creating an Encoded IP Library" on page 3-2.
2. Create a new design and place an instance of *simpleR* and an instance of *GilCellMix* as shown in Figure 5-1. The location and connectivity is not important so feel free to place them anywhere and do not place any wires.

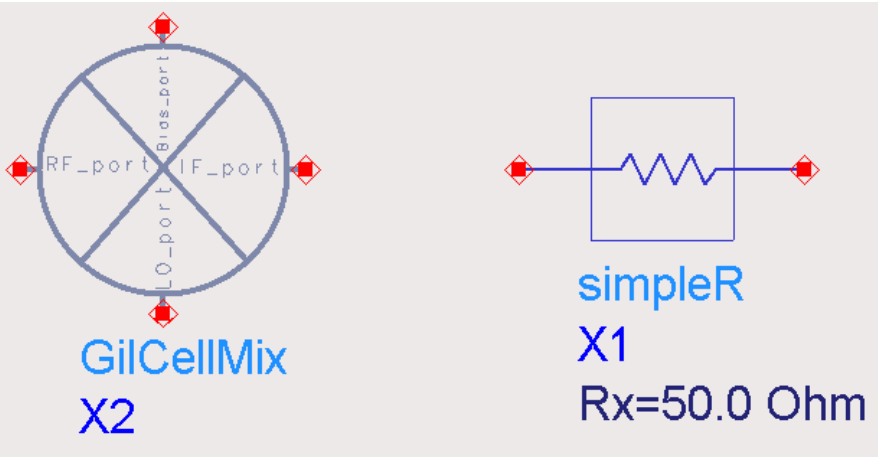
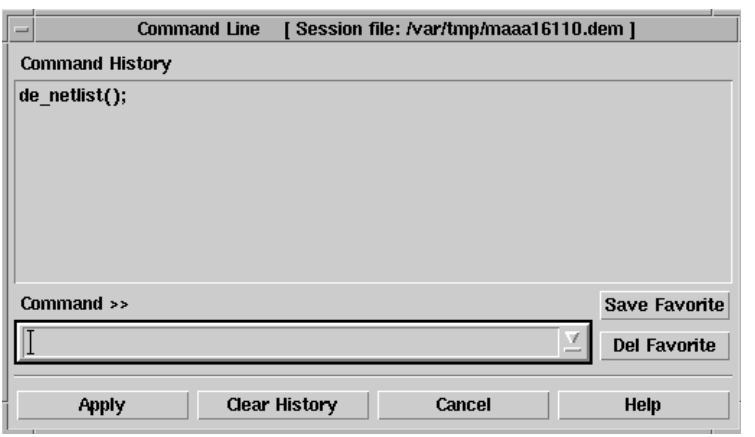


Figure 5-1. Simple Design

3. From the ADS Main window, choose **Tools > Command Line**.
4. In the **Command>>** field, enter the command **de_netlist();** The de_netlist(); command should appear in the Command History field after it has been entered.



5. From the ADS Main window, choose **Tools > Text Editor**. Use this or any other text editor to open the file called *netlist.log*. The netlist.log file can be found in the *EncoderExample_prj* project directory.

6. Edit the text file and remove the first line at the top that starts with the word "*Options*".
7. Remove the last two lines at the bottom of the netlist.log file. One line starts with "*simpleR:X*" and the other with "*GilCellMix:X*".
8. Save your netlist in the project *networks* subdirectory as *gilcellmix.net* (note the *.net* extension). Your *EncoderExample_prj/networks/gilcellmix.net* netlist file should look like the example shown below.

Example Netlist File

```
define simpleR ( _net16  _net15 )
parameters  Rx=50.0
R:R1  _net16 _net15 R=Rx Noise=yes
end simpleR

define GilCellMix ( _net12  _net6  _net22  _net15 )
;parameters
R:R6  _net9 _net28 R=20 Ohm Noise=no
R:R3  _net17 _net2 R=50 Ohm Noise=no
C:C3  0 _net2 C=1.0 uF
C:C1  _net12 _net10 C=1 uF
L:L1  _net10 _net21 L=0.5 nH Noise=yes
R:R1  _net13 _net21 R=50 Ohm Noise=no
R:R7  _net17 _net16 R=100 Ohm Noise=no
C:C4  _net15 _net7 C=1 uF
L:L2  _net7 _net16 L=0.5 nH Noise=yes
R:R10  _net6 _net1 R=400 Ohm Noise=no
R:R9  _net17 _net16 R=100 Ohm Noise=no
R:R2  _net13 _net3 R=50 Ohm Noise=no
R:R8  _net6 _net26 R=400 Ohm Noise=no
R:R11  _net6 _net24 R=800 Ohm Noise=no
R:R12  _net24 _net18 R=700 Ohm Noise=no
C:C5  _net14 _net25 C=1 uF
L:L3  _net25 _net22 L=0.5 nH Noise=yes
R:R13  _net18 _net4 R=500 Ohm Noise=no
R:R4  _net20 0 R=170 Ohm Noise=no
R:R5  _net23 0 R=170 Ohm Noise=no
R:R14  _net19 0 R=200 Ohm Noise=no
C:C2  0 _net3 C=1 uF
R:R15  _net11 0 R=1000 Ohm Noise=no
R:R17  _net5 0 R=45 Ohm Noise=no

"BJTM1":BJT8  _net1 _net16 _net8 Area=1 Mode=1 Noise=yes
"BJTM1":BJT6  _net26 _net2 _net8 Area=1 Mode=1 Noise=yes
"BJTM1":BJT5  _net1 _net2 _net27 Area=1 Mode=1 Noise=yes
```

```
"BJTM1":BJT7  _net26 _net16 _net27 Area=1 Mode=1 Noise=yes
"BJTM2":BJT4  _net8 _net3 _net28 Area=1 Mode=1 Noise=yes
"BJTM2":BJT3  _net27 _net21 _net9 Area=1 Mode=1 Noise=yes
"BJTM4":BJT9  _net6 _net1 _net14 Area=1 Mode=1 Noise=yes
"BJTM5":BJT14  _net14 _net11 _net5 Area=1 Mode=1 Noise=yes
"BJTM3":BJT10  _net6 _net24 _net17 Area=1 Mode=1 Noise=yes
"BJTM3":BJT11  _net17 _net18 _net13 Area=1 Mode=1 Noise=yes
"BJTM3":BJT12  _net13 _net4 _net11 Area=1 Mode=1 Noise=yes
"BJTM3":BJT13  _net4 _net11 _net19 Area=1 Mode=1 Noise=yes
"BJTM3":BJT1  _net28 _net11 _net23 Area=1 Mode=1 Noise=yes
"BJTM3":BJT2  _net9 _net11 _net20 Area=1 Mode=1 Noise=yes

model BJTM2 BJT NPN=1 PNP=0 Is=2.1482293E-16 Bf=1.30035647E+02
Nf=1.03E+00 Vaf=2.50E+01 Ikf=1.08500002E-02 Ise=7.56173434E-13
Ne=2.00E+00 Br=5.12295082E+00 Nr=1.00E+00 Ikr=5.600000013E-02
Isc=2.00640001E-12 Nc=2.00E+00 Rb=3.16646062E+01 Rbm=1.06309524E+01
Re=1.78571429E+00 Rc=3.75704756E+01 Imax=1.0 A Cje=1.93536003E-13
Vje=8.49999987E-01 Mje=3.99999994E-01 Cjc=1.04500002E-13
Vjc=7.49999989E-01 Mjc=4.99999993E-01 Xcjc=2.92440187E-01
Cjs=1.09200002E-13 Vjs=6.99999990E-01 Mjs=4.99999993E-01
Fc=7.99999988E-01 Xtf=3.35472224E+00 Tf=8.91292803E-12
Itf=2.17000005E-02 Ptf=1.80E+01 Tr=1.60000005E-09 Kf=0.00E+00
Af=1.00E+00 Ab=1.00 Fb=1.00 Iss=0 A Ns=1.00 Nk=0.50 Ffe=1.00 Lateral=0
RbModel=0 Approxqb=1 Tnom=25.0 Eg=1.11 Xtb=2.20E+00 Xti=8.00E+00

model BJTM1 BJT NPN=1 PNP=0 Is=1.07411147E-16 Bf=1.30035647E+02
Nf=1.03E+00 Vaf=2.50E+01 Ikf=5.42499974E-03 Ise=3.78086717E-13
Ne=2.00E+00 Br=5.12295082E+00 Nr=1.00E+00 Ikr=2.80000006E-02
Isc=1.16160000E-12 Nc=2.00E+00 Rb=6.12458791E+01 Rbm=1.91785714E+01
Re=3.57142857E+00 Rc=6.96232096E+01 Imax=1.0 A Cje=9.67680017E-14
Vje=8.49999987E-01 Mje=3.99999994E-01 Cjc=6.05000011E-14
Vjc=7.49999989E-01 Mjc=4.99999993E-01 Xcjc=2.52561980E-01
Cjs=8.68000015E-14 Vjs=6.99999990E-01 Mjs=4.99999993E-01
Fc=7.99999988E-01 Xtf=3.35472224E+00 Tf=8.91292803E-12
Itf=1.08500002E-02 Ptf=1.80E+01 Tr=1.60000005E-09 Kf=0.00E+00
Af=1.00E+00 Ab=1.00 Fb=1.00 Iss=0 A Ns=1.00 Nk=0.50 Ffe=1.00 Lateral=0
RbModel=0 Approxqb=1 Tnom=25. Eg=1.11 Xtb=2.20E+00 Xti=8.00E+00

model BJTM4 BJT NPN=1 PNP=0 Is=4.29644587E-16 Bf=1.30035647E+02
Nf=1.03E+00 Vaf=2.50E+01 Ikf=2.17000005E-02 Ise=1.51234685E-12
Ne=2.00E+00 Br=5.12295082E+00 Nr=1.00E+00 Ikr=1.11999998E-01
Isc=3.69600001E-12 Nc=2.00E+00 Rb=1.61795253E+01 Rbm=5.66269841E+00
Re=8.92857130E-01 Rc=2.27441087E+01 Imax=1.0 A Cje=3.87072007E-13
Vje=8.49999987E-01 Mje=3.99999994E-01 Cjc=1.92500003E-13
Vjc=7.49999989E-01 Mjc=4.99999993E-01 Xcjc=3.17506489E-01
Cjs=1.54000003E-13 Vjs=6.99999990E-01 Mjs=4.99999993E-01
Fc=7.99999988E-01 Xtf=3.35472224E+00 Tf=8.91292803E-12
Itf=4.34000010E-02 Ptf=1.80E+01 Tr=1.60000005E-09 Kf=0.00E+00
```

```
Af=1.00E+00 Ab=1.00 Fb=1.00 Iss=0 A Ns=1.00 Nk=0.50 Ffe=1.00 Lateral=0
RbModel=0 Approxqb=1 Tnom=25. Eg=1.11 Xtb=2.20E+00 Xti=8.00E+00

model BJTM5 BJT NPN=1 PNP=0 Is=4.08631847E-16 Bf=1.66178934E+02
Nf=1.03E+00 Vaf=2.50E+01 Ikf=2.32500005E-02 Ise=1.21824776E-12
Ne=2.00E+00 Br=5.12295082E+00 Nr=1.00E+00 Ikr=1.199999998E-01
Isc=4.08480002E-12 Nc=2.00E+00 Rb=5.91784277E+01 Rbm=1.41063123E+01
Re=8.33333321E-01 Rc=2.20144369E+01 Imax=1.0 A Cje=3.12320005E-13
Vje=8.49999987E-01 Mje=3.99999994E-01 Cjc=2.12750004E-13
Vjc=7.49999989E-01 Mjc=4.99999993E-01 Xcjc=2.21856636E-01
Cjs=1.65300003E-13 Vjs=6.99999990E-01 Mjs=4.99999993E-01
Fc=7.99999988E-01 Xtf=3.16117215E+00 Tf=7.97477712E-12
Itf=4.65000010E-02 Ptf=1.80E+01 Tr=1.60000005E-09 Kf=0.00E+00
Af=1.00E+00 Ab=1.00 Fb=1.00 Iss=0 A Ns=1.00 Nk=0.50 Ffe=1.00 Lateral=0
RbModel=0 Approxqb=1 Tnom=25. Eg=1.11 Xtb=2.20E+00 Xti=8.00E+00

model BJTM3 BJT NPN=1 PNP=0 Is=2.04315924E-16 Bf=1.66178934E+02
Nf=1.03E+00 Vaf=2.50E+01 Ikf=1.16250003E-02 Ise=6.09123888E-13
Ne=2.00E+00 Br=5.12295082E+00 Nr=1.00E+00 Ikr=6.00000013E-02
Isc=2.31840001E-12 Nc=2.00E+00 Rb=1.17166379E+02 Rbm=2.70221484E+01
Re=1.66666667E+00 Rc=3.58072512E+01 Imax=1.0 A Cje=1.56160003E-13
Vje=8.49999987E-01 Mje=3.99999994E-01 Cjc=1.20750002E-13
Vjc=7.49999989E-01 Mjc=4.99999993E-01 Xcjc=1.95445132E-01
Cjs=1.18900002E-13 Vjs=6.99999990E-01 Mjs=4.99999993E-01
Fc=7.99999988E-01 Xtf=3.16117215E+00 Tf=7.97477712E-12
Itf=2.32500005E-02 Ptf=1.80E+01 Tr=1.60000005E-09 Kf=0.00E+00
Af=1.00E+00 Ab=1.00 Fb=1.00 Iss=0 A Ns=1.00 Nk=0.50 Ffe=1.00 Lateral=0
RbModel=0 Approxqb=1 Tnom=25. Eg=1.11 Xtb=2.20E+00 Xti=8.00E+00

end GilCellMix
```

Now that you have a netlist file, encode the netlist as described in “Creating an Encoded IP Library” on page 4-1, creating a design kit named at_mixlib, in a different location than the original at_mixlib design kit that was created with the schematic subcircuits in the tutorial. Then follow the next steps to modify the design kit created in the tutorial. In this process, you will copy some files from the new at_mixlib design kit in the old mix_lib design kit, and then modify a few files. It is important that these 2 design kits have the same name. You may prefer to make a copy of the old design kit before you start modifying it. You should also make a copy of EncoderTest_prj.

1. From the new design kit, copy the file circuit/models/at_mixlib.library to the circuit/models directory of the old design kit.
2. In the old design kit, edit the file circuit/ael/at_mixlib_encode_item.ael. For each of the two item definitions, you will need to modify the *netlistFormat* and

netlistData information as shown in the example below. Note that the changes are shown in **bold**. After you have modified the file, be sure to save your changes.

```
set_simulator_type(1);
set_design_type(1);
create_item("GilCellMix_at_mixlib",                 //Name
            "Encoded Gilbert Cell Mixer - encoded" //Label
            "X",                                    //Prefix
            ITEM_NOT_ALL_PARM,                      //Attribute
            -1,                                     //Priority
            NULL,                                   //iconName
            "Component Parameters",                  //dialogName
            "",                                      //dialogData
            "#uselib \"at_mixlib\" , \"%d\"\n\"GilCellMix\":%t%#%44?0%:
            %31?%C%:_net%c%;%;%e %r%b%8?%29?%:%30?%p%:%k%?[%1i]%;=%p%;
            %;%e%e",                                //netlistFormat
            "gilcellmix_at_mixlib",                //netlistData
            ComponentAnnotFmt,                      //displayFormat
            "SYM_GilCellMix_at_mixlib",             //symbolName
            0,                                      //artWorkType
            NULL,                                   //artworkData
            ITEM_PRIMITIVE_EX                       //extraAttrib
);
set_simulator_type(1);
set_design_type(1);
create_item("simpleR_at_mixlib",                    //Name
            "simpleR - encoded",                    //Label
            "X",                                    //Prefix
            ITEM_NOT_ALL_PARM,                      //Attribute
            -1,                                     //Priority
            NULL,                                   //iconName
            "Component Parameters",                 //dialogName
            "",                                     //dialogData
            "#uselib \"at_mixlib\" , \"%d\"\n\"simpleR\":%t%#%44?0%:
            %31?%C%:_net%c%;%;%e %r%b%8?%29?%:%30?%p %:%k%?[%1i]%;=%p%;
            %;%e%e",                                //netlistFormat
             "gilcellmix_at_mixlib",                //netlistData
             ComponentAnnotFmt,                     //displayFormat
             "SYM_simpleR_at_mixlib",               //symbolName
             0,                                     //artWorkType
             NULL,                                  //artworkData
             ITEM_PRIMITIVE_EX,                      //extraAttrib

create_parm("Rx","",68608,"StdFormSet","1",prm("StdForm","50.0")));
```

3. To test these changes, start ADS and open the EncoderTest_prj used previously in the tutorial.
4. If you previously installed the design kit at_mixlib as described in "Installing an Encoded Design Kit" on page 3-14, disable or cut that design kit from the ADS Main window menu pick **DesignKit > Setup Design Kits**. For more information, refer to the *Design Kit Installation and Setup* documentation.
5. After the design kit has been disabled, close and restart ADS.
6. Now install the modified design kit as described in "Installing an Encoded Design Kit" on page 3-14.
7. Open the TestCkt design, and delete and replace the GilCellMix and simpleR components.
8. Simulate the circuit. The top few lines of netlist.log file in the project directory after simulation should resemble the following:

```
Options ResourceUsage=yes UseNutmegFormat=no
TopDesignName="/EncoderTest_prj/networks/TestCkt"
#uselib "at_mixlib" , "gilcellmix_at_mixlib"
"simpleR":X2  Vif 0 Rx=50.0 Ohm
#uselib "at_mixlib" , "gilcellmix_at_mixlib"
"GilCellMix":X1  Vrf _net1 _net2 _net5
```

Removing Encoded IP Libraries

You can remove an encoded IP Library that has been previously installed by entering:

```
hpeesofpkg --remove <package_name>
```

Where *package_name* is the name of the encoded IP library being removed. Close ADS before removing a library.

Note Ensure the encoded IP library package exists and that you have the correct spelling before attempting to remove it. See "Viewing Packages" on page 5-5 for information on how to view the existing packages.

Chapter 6: Troubleshooting

This chapter provides information on known problems and solutions to help you resolve unexpected issues.

Known Problems and Solutions

Problem: Third party customer is unable to use two packages created by two separate vendors.

Solution: When selecting a library name, choose a name that is unique and descriptive. Two encoded libraries with identical names will prevent a third party from using both simultaneously.

Problem: Encoded schematic designs do not work for my customer.

Solution: Encoded schematic designs may be version dependent. Verify the version that the library was created in.

Problem: Encoded libraries do not appear on the component palette.

Solution: Ensure that the unix or windows environment variable COMPL_DIR is set. This environment variable can be set to the same value as $HPEESOF_DIR (ADS install directory) if you have installed your encoded libraries in the default location.

Also make sure that the following configuration variables have not been modified.

Configuration Variable	Must Contain
SYSTEM_CIRCUIT_SYMBOLS	$HPEESOF_DIR/ComponentLibs/symbols
COMPONENT_LIBS_AEL	{$COMPL_DIR}/ComponentLibs/ael/ComponentLibs

These are set in $HPEESOF_DIR/config/de_sim.cfg but can be overwritten in $HPEESOF_DIR/custom/config/de_sim.cfg or $HOME/hpeesof/config/de_sim.cfg.

Additionally, for debian package installations, check for the presence of the appropriate .ctl, .idf, .rec files in the $HPEESOF_DIR/ComponentLibs/records directory. The path for these variables is set in hpeesofbrowser.cfg. The variable is called HPVENDORLIB_BROWSER PATH. For more information on analyzing configuration variables, refer to the configuration browser.

Problem: After debian package installation, the bitmap shown on the component palette requires updating.

Solution: You can specify a different bitmap file after the encoded libraries are installed. Change the bitmap file, <*design_name*>.bmp, under $HPEESOF_DIR/ComponentLibs/<*encoded_lib_name*>/bitmaps/ to one of your choice. If the name is changed, the reference to it in $HPEESOF_DIR/ComponentLibs/ael/<*encoded_lib_name*>/encode_def.ael also needs to be changed.

Problem: Custom bitmaps in a library created on a PC do not show up when the library is installed on unix. Or, custom bitmaps in a library created on unix, do not show up when the library is installed on a PC. The IP Encoder is not currently linked with a bitmap translator.

Solution: After the library is installed, it is possible to replace the default bitmaps if you have access to the proper bitmaps for the second platform. The bitmaps in a library installed in the default location are stored in $HPEESOF_DIR/ComponentLibs/<*libname*>/bitmaps/<*platform*>. The bitmap bears the same name as the component. Some bitmaps that are available to copy are found in: $HPEESOF_DIR/circuit/bitmaps and $HPEESOF_DIR/designguides/projects/dgstudio/ui/bitmaps/adsbmps.

Problem: When simulating with an encoded library, an error is generated about not finding the library.

Solution: The model may be missing from the configuration file, ADSlibconfig. A configuration line looks like this:

<*encoded_lib_name*> $HPEESOF_DIR/ComponentLibs/models/<*encoded_lib_name*>.library

This line needs to be in one of the following files. All three will be read and the contents combined.

- $HPEESOF_DIR/circuit/config/ADSLibconfig
- $HPEESOF_DIR/custom/circuit/config/ADSLibconfig
- $HOME/hpeesof/circuit/config/ADSLibconfig

Problem: When attempting to install an encoded library in the form of a debian package file, the following error is generated:

```
hpeesofpkg: operation requires read/write access to hpeesofpkg status area
```

Solution: You must have read/write permission in your ADS installation directory in order to install a debian package using the *hpeesofpkg* package command. Reset your permissions in the ADS installation directory.

Additional Known Issues

Netlist file must have the file extension .net.

No layout information is encoded. Fixed artwork will be copied over and included in the encoded library. AEL Macro Artwork and Synchronized Artwork will have to be a manual step. Refer to the chapter "*Design Kit Development for Layout*" in the "*Design Kit Development*" documentation.

There is a report of a failure during encoding when using some built-in ADS symbols.

Glossary

Add-On

A debian package file (.deb) that contains additional capabilities for Advanced Design System.

ADS (Advanced Design System)

This is the Agilent Technologies Advanced Design System.

AEL (Application Extension Language)

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

callback

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

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.

DES (Data Encryption Standard)

Data Encryption Standard (DES) is a widely-used method of data encryption developed in 1977.

EDA (Electronic Design Automation)

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

encoded IP (encoded intellectual property)

A protected version of the contents of a schematic design or netlist.

encoded IP library (encoded intellectual property library)

A collection of encoded designs or netlists that appear on the ADS palettes as well as in the library browser.

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.

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, WinNT.

package

A special file archive that allows the easy distribution, installation and removal of Add-Ons for the Advanced Design System.

RFIC

Radio Frequency Integrated Circuits.

RF IP Encoder (Radio Frequency Intellectual Property Encoder)

The product that takes *schematic designs and netlists* and generates *package* files that contain *encoded IP libraries* which in turn contain *encoded IP*. The RF IP Encoder product provides a mechanism for sharing simulatable ADS schematic designs and netlists without divulging your actual design information.

schematic design

A schematic circuit design created using the Advanced Design System.

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.

Index

A

ADS
- exiting, 3-24
- versions, 2-4

AEL, 5-6

C

commands
- de_netlist(), 5-7
- hpeesofpkg, 3-13, 5-1, 5-2, 5-12

D

data
- viewing, 3-23

design kits, 1-1, 1-2, 1-4, 3-8, 3-12, 3-13, 3-14, 4-4, 4-5, 5-1, 5-6, 5-10, 6-3

design variables, 3-2, 3-21

designs
- encoding, 3-9

E

encoded IP libraries, 1-4, 2-4, 3-1, 3-9, 3-13, 3-17, 5-1, 5-6
- descriptions, 3-19
- hierarchy, 3-22
- new versions, 5-1

encoding
- designs, 3-9
- netlists, 3-10, 5-6, 5-10

example projects, 3-1

F

files
- .deb, 3-12, 5-1
- .txt, 3-12, 4-5
- .zap, 3-1
- .zip, 3-12, 4-4
- AEL, 5-6
- at_mixlib.library, 5-10
- at_mixlib_encode_item.ael, 5-10
- netlist.log, 5-8, 5-12
- opening, 3-4
- package, 3-12, 5-1
- viewing files, 3-14

H

hierarchy
- schematic, 3-22

L

libraries
- encoded IP, 2-4, 3-1, 3-9, 5-1
  - creating, 3-2
  - installing, 3-13
  - unique names, 3-11

licenses, 2-1

N

netlist.log, 5-8, 5-12

netlistData, 5-11

netlistFormat, 5-10

networks subdirectory, 5-8

P

package files, 3-12, 5-1
- removing, 2-3
- viewing, 3-14

ports
- specified, 3-2

projects
- examples, 3-1
- unarchiving, 3-1

S

schematic hierarchy, 3-22

security, 2-4

subnetworks
- parametric, 3-2

symbols, 3-3
- encoded symbol files, 5-2
- viewing, 3-5

V

variables
- design, 3-2, 3-21
- editing, 3-21
- USER_DSN_PATH, 5-2

versions
- ADS, 2-4
- encoded IP libraries, 5-1

viewing
data, 3-23

Z

zap files, 3-1
zip files, 1-4, 3-12, 4-4