Advanced Design System 2011.01

Feburary 2011
RF Intellectual Property Encoder

Acknowledgments

POSSIBILITY OF SUCH DAMAGE.

7-zip - 7-Zip Copyright: Copyright (C) 1999-2009 Igor Pavlov. Licenses for files are: 7z.dll: GNU LGPL + unRAR restriction, All other files: GNU LGPL. 7-zip License: This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful,but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA. unRAR copyright: The decompression engine for RAR archives was developed using source code of unRAR program.All copyrights to original unRAR code are owned by Alexander Roshal. unRAR License: The unRAR sources cannot be used to re-create the RAR compression algorithm, which is proprietary. Distribution of modified unRAR sources in separate form or as a part of other software is permitted, provided that it is clearly stated in the documentation and source comments that the code may not be used to develop a RAR (WinRAR) compatible archiver. 7-zip Availability: http://www.7-zip.org/

AMD Version 2.2 - AMD Notice: The AMD code was modified. Used by permission. AMD copyright: AMD Version 2.2, Copyright © 2007 by Timothy A. Davis, Patrick R. Amestoy, and Iain S. Duff. All Rights Reserved. AMD License: Your use or distribution of AMD or any modified version of AMD implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA Permission is hereby granted to use or copy this program under the terms of the GNU LGPL, provided that the Copyright, this License, and the Availability of the original version is retained on all copies.User documentation of any code that uses this code or any modified version of this code must cite the Copyright, this License, the Availability note, and "Used by permission." Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included. AMD Availability: http://www.cise.ufl.edu/research/sparse/amd

UMFPACK 5.0.2 - UMFPACK Notice: The UMFPACK code was modified. Used by permission. UMFPACK Copyright: UMFPACK Copyright © 1995-2006 by Timothy A. Davis. All Rights Reserved. UMFPACK License: Your use or distribution of UMFPACK or any modified version of UMFPACK implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA Permission is hereby granted to use or copy this program under the terms of the GNU LGPL, provided that the Copyright, this License, and

the Availability of the original version is retained on all copies. User documentation of any code that uses this code or any modified version of this code must cite the Copyright, this License, the Availability note, and "Used by permission." Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included. UMFPACK Availability: http://www.cise.ufl.edu/research/sparse/umfpack UMFPACK (including versions 2.2.1 and earlier, in FORTRAN) is available at http://www.cise.ufl.edu/research/sparse . MA38 is available in the Harwell Subroutine Library. This version of UMFPACK includes a modified form of COLAMD Version 2.0, originally released on Jan. 31, 2000, also available at http://www.cise.ufl.edu/research/sparse . COLAMD V2.0 is also incorporated as a built-in function in MATLAB version 6.1, by The MathWorks, Inc. http://www.mathworks.com . COLAMD V1.0 appears as a column-preordering in SuperLU (SuperLU is available at http://www.netlib.org). UMFPACK v4.0 is a built-in routine in MATLAB 6.5. UMFPACK v4.3 is a built-in routine in MATLAB 7.1.

Qt Version 4.6.3 - Qt Notice: The Qt code was modified. Used by permission. Qt copyright: Qt Version 4.6.3, Copyright (c) 2010 by Nokia Corporation. All Rights Reserved. Qt License: Your use or distribution of Qt or any modified version of Qt implies that you agree to this License. This library is free software; you can redistribute it and/or modify it under the
terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA Permission is hereby granted to use or copy this program under the terms of the GNU LGPL, provided that the Copyright, this License, and the Availability of the original version is retained on all copies.User
documentation of any code that uses this code or any modified version of this code must cite the Copyright, this License, the Availability note, and "Used by permission." Permission to modify the code and to distribute modified code is granted, provided the Copyright, this License, and the Availability note are retained, and a notice that the code was modified is included. Qt Availability: http://www.qtsoftware.com/downloads Patches Applied to Qt can be found in the installation at:
$HPEESOF_DIR/prod/licenses/thirdparty/qt/patches. You may also contact Brian Buchanan at Agilent Inc. at brian_buchanan@agilent.com for more information.

The HiSIM_HV source code, and all copyrights, trade secrets or other intellectual property rights in and to the source code, is owned by Hiroshima University and/or 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 documentation 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 U.S. Government Restricted Rights. Software and technical data rights granted to the federal government include only those rights customarily provided to end user customers. Agilent provides this customary commercial license in Software and technical data pursuant to FAR 12.211 (Technical Data) and 12.212 (Computer Software) and, for the Department of Defense, DFARS 252.227-7015 (Technical Data - Commercial Items) and DFARS 227.7202-3 (Rights in Commercial Computer Software or Computer Software Documentation).

About RF Intellectual Property Encoder

The *RF Intellectual Property Encoder* (RF IP Encoder) lets you create an encrypted version of an ADS circuit. For example, it is useful to IC vendors who wish to promote their products by sharing a simulatable model of their IC with their prospective customers. RF IP Encoder uses encryption technology to let the IC vendor do this without revealing the intellectual property of the underlying design. Models encoded in this way can be distributed as an encrypted ADS Design Kit to the customers of the IC vendor. The customer uses the Kit to help design the IC into their system and verify IC and system performance: models users can *run* the "black box" model, but not "see inside" the box. The design need not have been created in ADS. For example, it's possible to use *HSPICE Compatibility* (hspice) feature to import and then encrypt an HSPICE netlist for native simulation in ADS.

Major Features

Key features of RF IP Encoder include the following:

- Simple encoding process
- Ease of packaging and delivery
- Ability to include design variables

Note
The RF Intellectual Property Encoder can be used with RFIC Dynamic link. For more information, see *Encoding Cadence Designs for RFIC Dynamic Link* (dynlnkug).

Intended Audience

This documentation 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). *Configuring RF Intellectual Property Encoder* (rfipenc) is intended to assist the system administrator or CAD manager with installing the RF IP Encoder product.

- *Model-Providers*. *Getting Started Tutorial* (dynlnkug) is intended to familiarize the model-provider with the entire process of encoding designs and using encoded IP libraries. *Encoding Schematic Designs and Netlists* (rfipenc) is intended to quickly assist the model-provider in encoding and packaging an IP Library in ADS.

- *Model-Users*. *Working with Encoded Designs* (rfipenc) 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 *Getting Started Tutorial* (dynlnkug) also demonstrates how to use an encoded library however the tutorial requires that you first build a simple library.

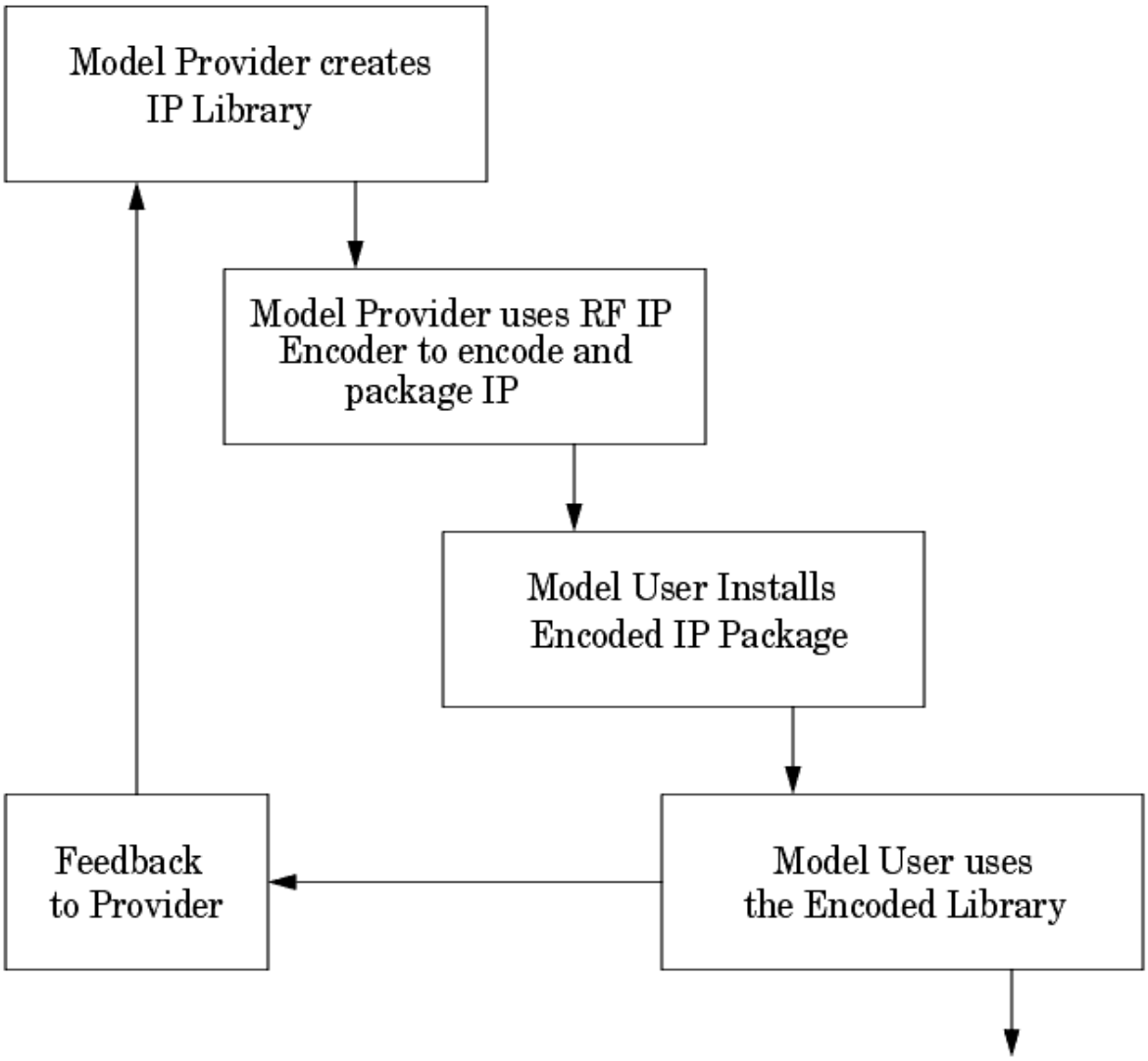
General Process

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 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 *RF IP Encoder Use Model* (adshbapp) describes the general use model for encoding schematic designs and netlists.



RF IP Encoder Use Model

- The Model-Provider groups schematic designs or netlist into a single ADS Workspace.
- 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

A library can be encoded into 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.

Another benefit of using ADS Design Kits 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* (dkarch) documentation.

What's in this Documentation

The goal of this documentation 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 documentation contains:

- *Configuring RF Intellectual Property Encoder* (rfipenc) describes how to install and configure the software.
- *Getting Started Tutorial* (dynlnkug) 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.
- *Encoding Schematic Designs and Netlists* (rfipenc) is a quick reference for encoding designs and netlists.
- *Working with Encoded Designs* (rfipenc) assists the model-user with instructions on installing, viewing, removing and using encoded IP libraries.
- *Troubleshooting RF Intellectual Property Encoder* (rfipenc) provides helpful information on known problems and solutions.

Configuring RF Intellectual Property Encoder

This section 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 2006A or later.

License Requirements

In the present product structure, the RF IP Encoder product increment line, *ads_encoder*, is included in W2200 ADS Core. (In the older ADS product structure , the RF IP Encoder was licensed separately as the now superceded E8894 RF IP Encoder.)

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 documentation 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.

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 included in the standard Advanced Design System web image. It must be downloaded separately from Agilent EEsof EDA Web .

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

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.

Note
For further information on working with packages, refer to *Working with Encoded Designs* (rfipenc).

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.

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 *Getting Started Tutorial* (dynlnkug).

Creating an Encoded IP Library

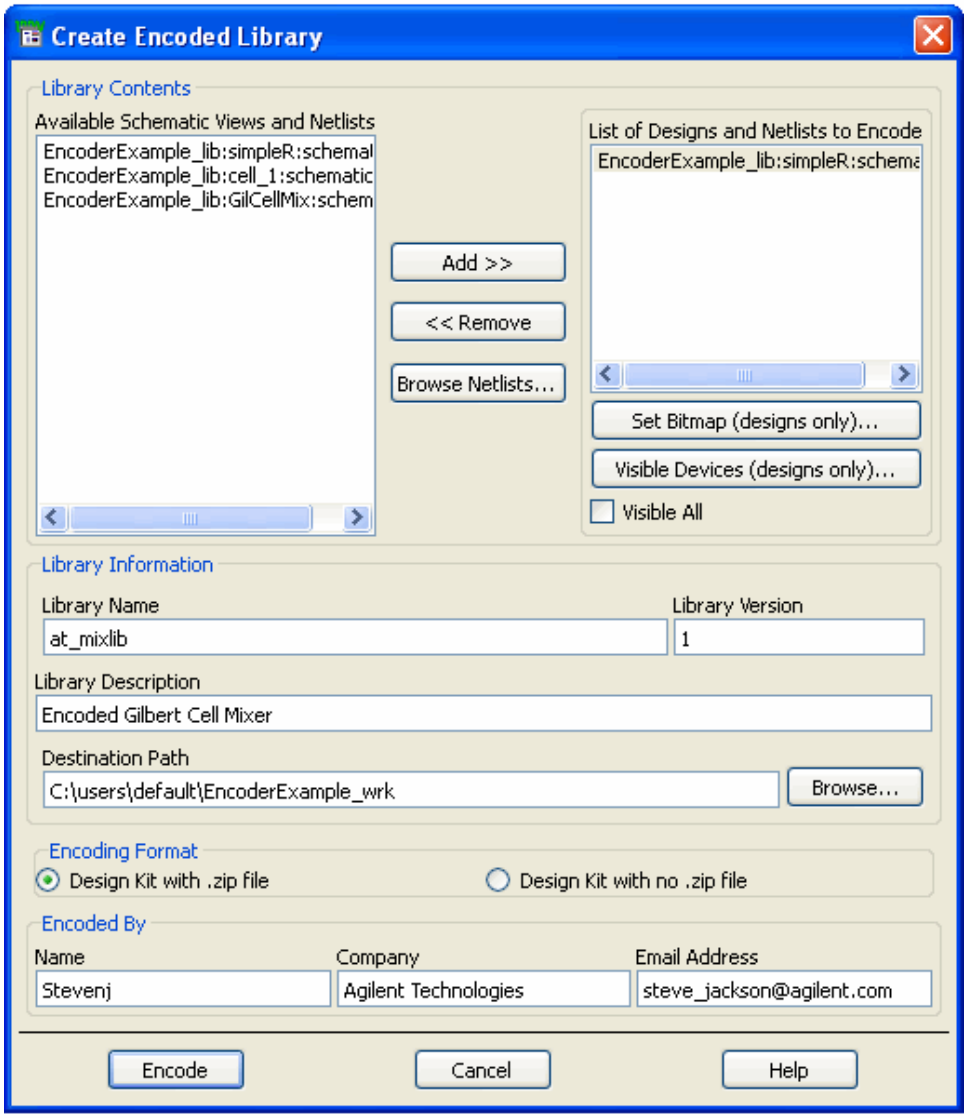
Follow the steps below to create an encoded IP library from one or more schematic designs and/or netlists.

1. Remove any simulation controllers present on the schematics of the designs that are to be encoded.
2. 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.
3. In the **Available Schematic Views 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**.

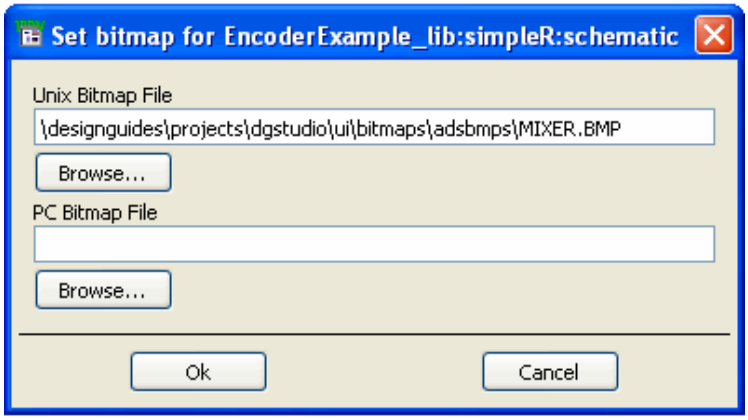
   To include your own netlists in the *Available Schematic Views and Netlists* list, place your netlists in the *networks* subdirectory of the current workspace directory. For example:
   <your_workspace>/networks/<your_netlist>.net
   All netlist files must have a *.net* extension to be recognized by 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* (rfipenc).

   In case of designs that have several levels of hierarchy (i.e., designs with other subcircuits included in them), you can encode the entire design by selecting the top level design file for encoding. However, if a design is encoded in this manner, only the top level ports are accessible in the encoded component. If you wish to access a subcircuit individually, then that subcircuits file must be selected and encoded explicitly. In this case, two different encoded components are formed (one for top level and another for subcircuit) each of which are complete in themselves.



4. 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 (designs 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.

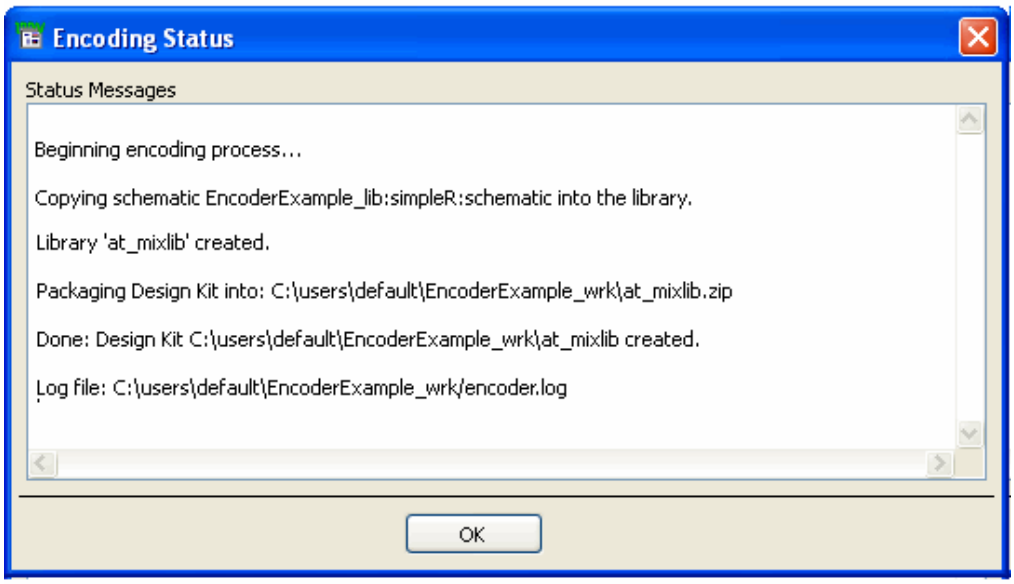
5. Use the **Library Information** section to define your new encoded IP library. Choose a unique name for your encoded IP Library and enter it into the **Library Name** field.

> **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.

6. Enter a version number into the **Library Version** field. The default version value is *1*.
7. Enter a clear description of your library into the **Library Description** field.
8. 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.
9. Select an encoding format.

   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.
10. 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.
11. 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.



12. Click **OK** to dismiss the *Encoding Status* dialog box.

Once the encoding process is complete, you will have an 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 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. The encoded library will then reference these un-encoded data files in this new location.

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/ .

Getting Started with RF Intellectual Property Encoder

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

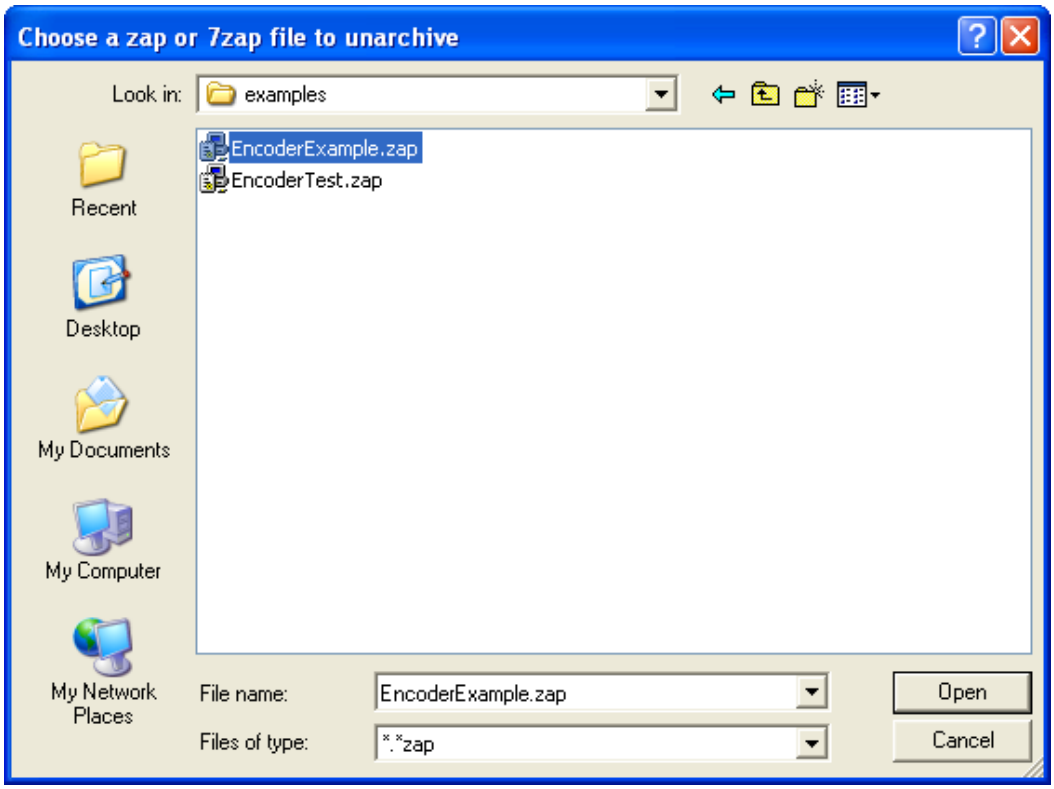
This section provides information on how to install the tutorial example files.

The example for this tutorial are provided as .zap files.

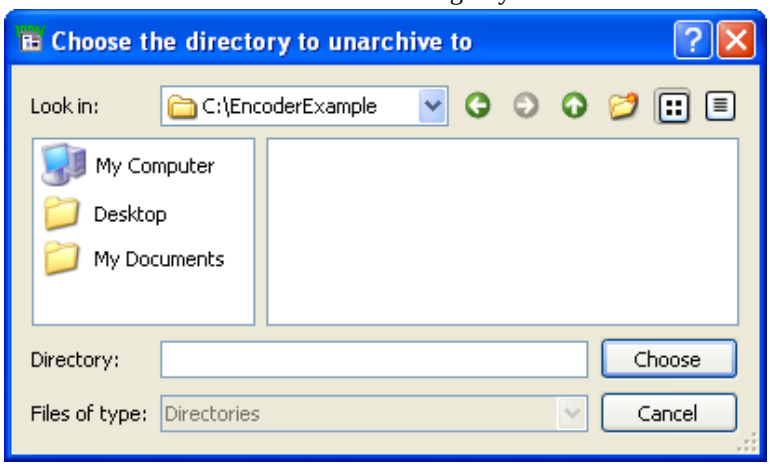
After the RF IP Encoder is installed, these .zap files are located in the installation directory under the *ComponentLibs/examples* subdirectories.

For example, if your installation directory is *C:\ADS2011_01* on a Windows system, then the example files are located under *C:\ADS2011_01\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 Workspace or Project** . An Unarchive Workspace dialog box is displayed.



2. Select the *.zap* file to be archived (*EncoderExample.zap* and *EncoderTest.zap*).
3. Then select the directory where the *.zap* file must be archived.

3. 

4. click **Choose** to unarchive the selected *.zap* file.

The first example *(EncoderExample.zap*) contains the designs that will be encoded during the tutorial.

The second example (*EncoderTest.zap*) contains test bench circuits for simulations that use the designs from the first example.

Creating an Encoded IP Library

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

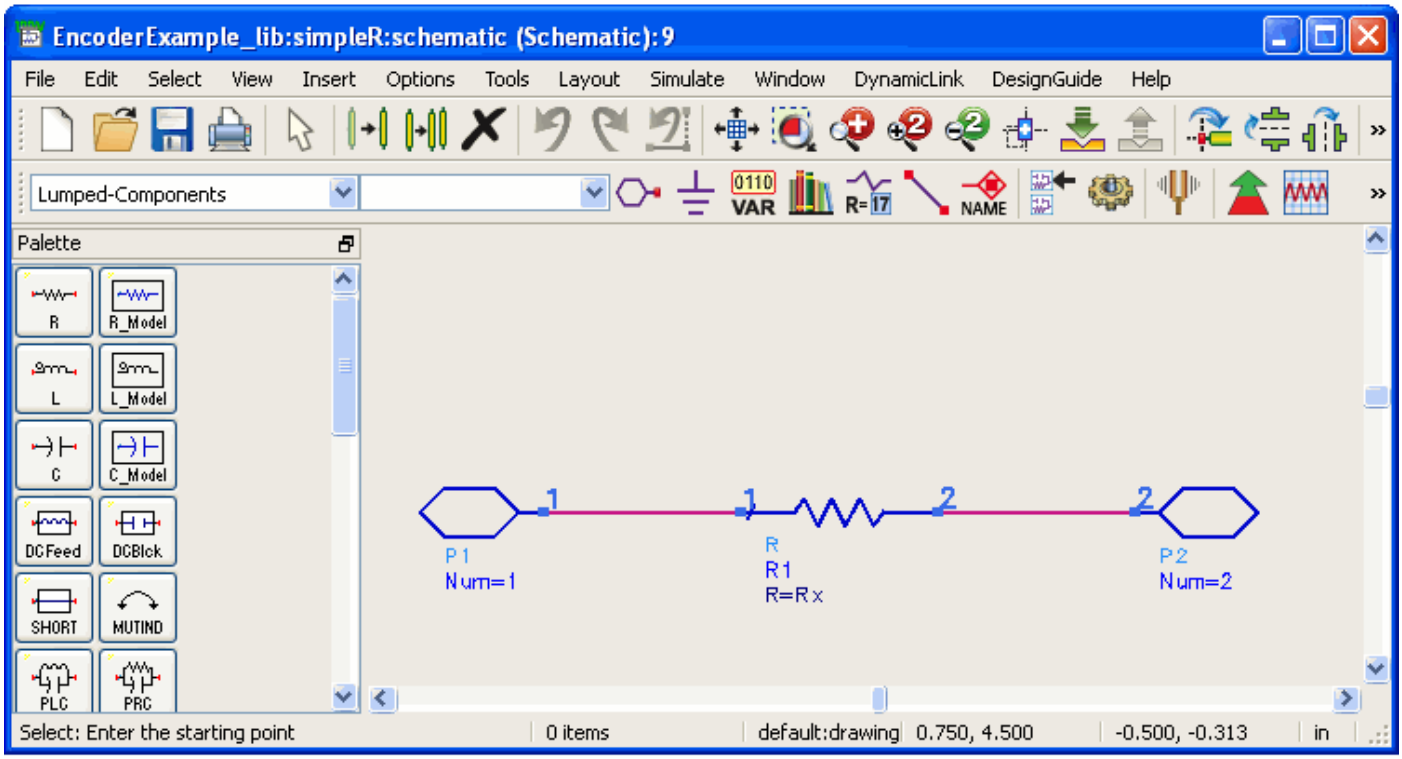
1. Open the workspace *EncoderExample_wrk* that was previously unarchived.
2. In the ADS main window, click **File > Open > Schematic** to display the Open Cell View dialog box. In the Open Cell View dialog box, select **SimpleR** in the Cell category.
3. Click **OK** to open the *simpleR* schematic from the old_networks directory of the Workspace. This file is a basic schematic of a resistor with specified ports. This 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 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.

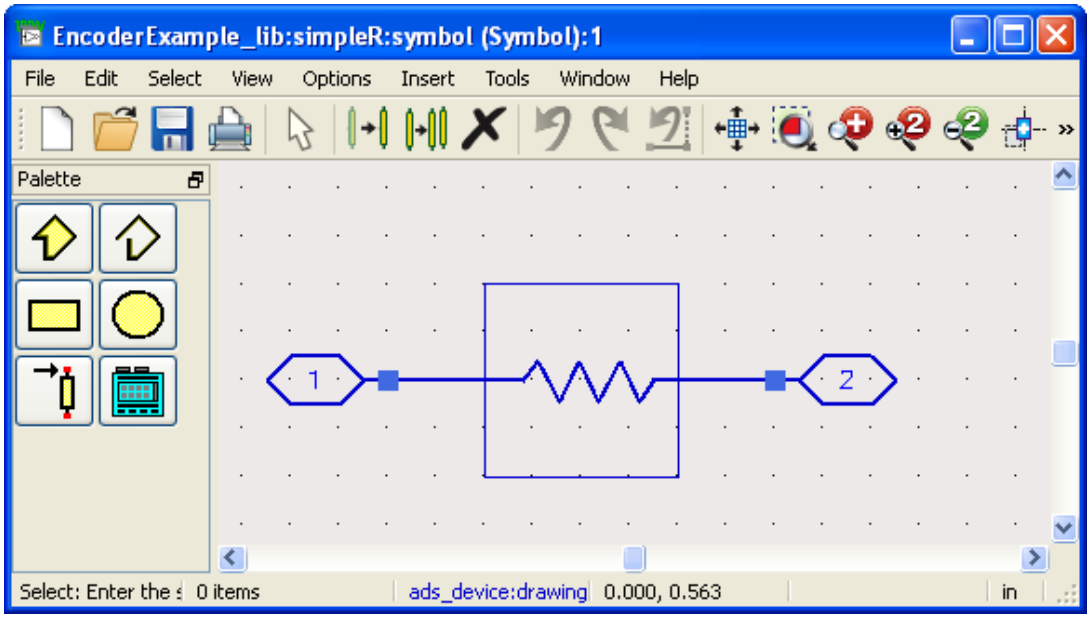
   The following figure displays the simple resistor network that includes the parameter *Rx* in the Schematic.

   Simple Parametric Subnetwork



4. From the ADS Main window, right-click the symbol under a schematic and click **Open** to view the schematic symbol that represents your schematic design. The following figure shows the basic schematic symbol that represents the *simpleR* schematic design.

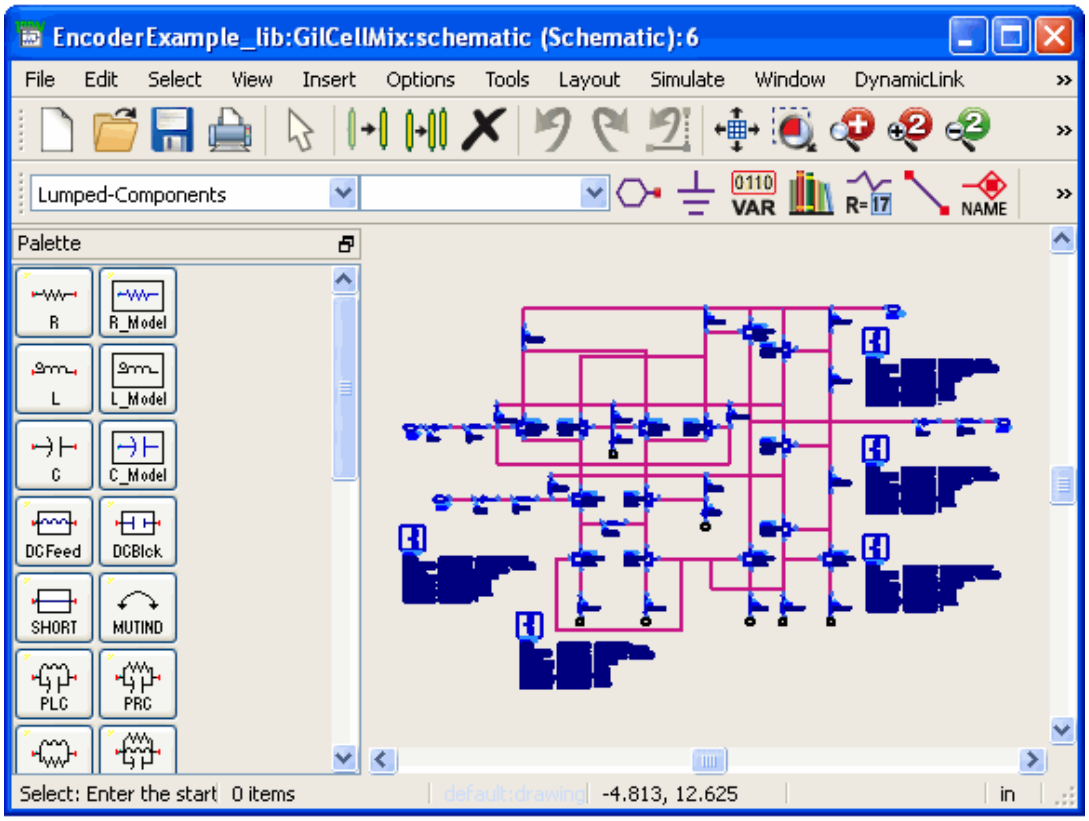
simpleR Schematic Symbol



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.

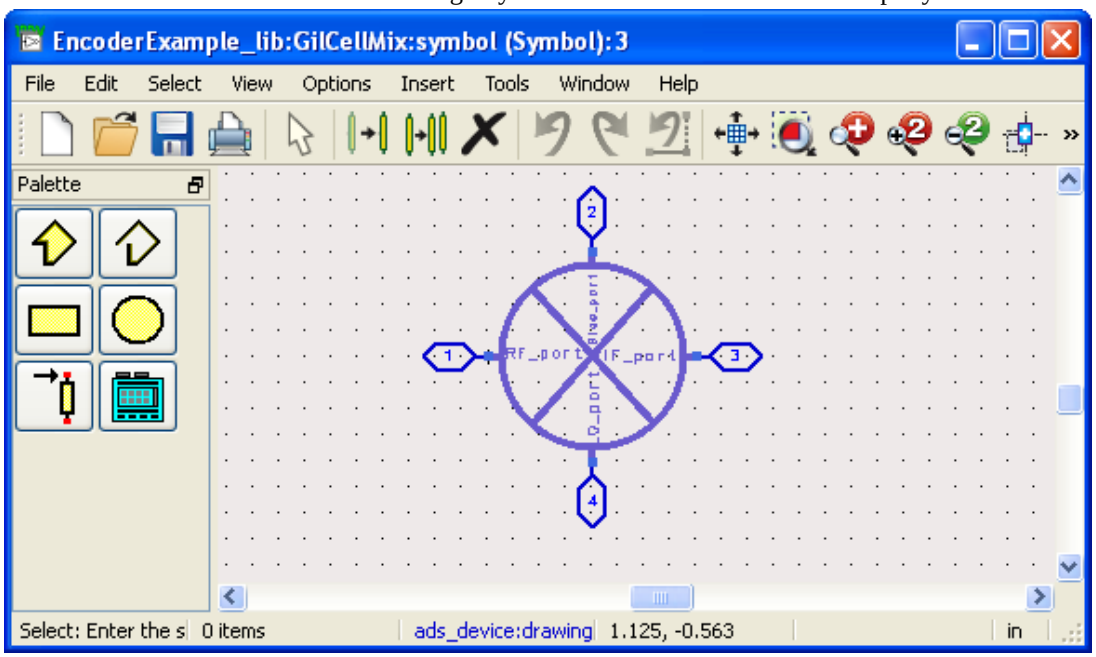
5. Select **File > Open > Schematic** to display the Open Cell View dialog box. In the Open Cell View dialog box, select **GilCellMix** in the Cell category. The following figure displays the more complex design that will also be included in the encoded IP Library.

Mixer Schematic Design



6. From the ADS Main window, right-click the symbol under a schematic and click **Open** to view the schematic symbol that represents the mixer schematic design. The following figure shows the schematic symbol that represents the mixer schematic design.

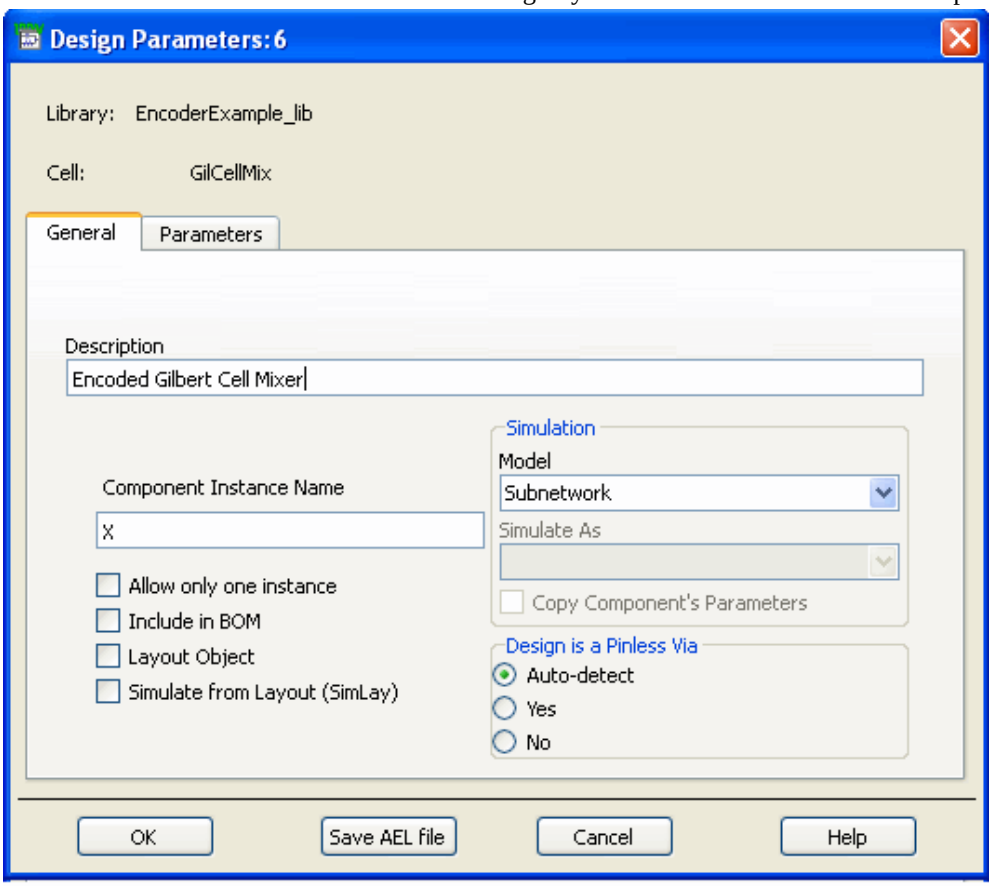
GilCellMix Schematic Symbol



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.

7. Select **View > Create/Edit Schematic** to return to the schematic view.
8. 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 the following figure. This description will be the Component Library description for the encoded component.
9. Click **OK** to close the Design Parameters dialog box.
10. Save the design to save any changes entered in the Design Parameters dialog box.

Setting the Encoded Component Description



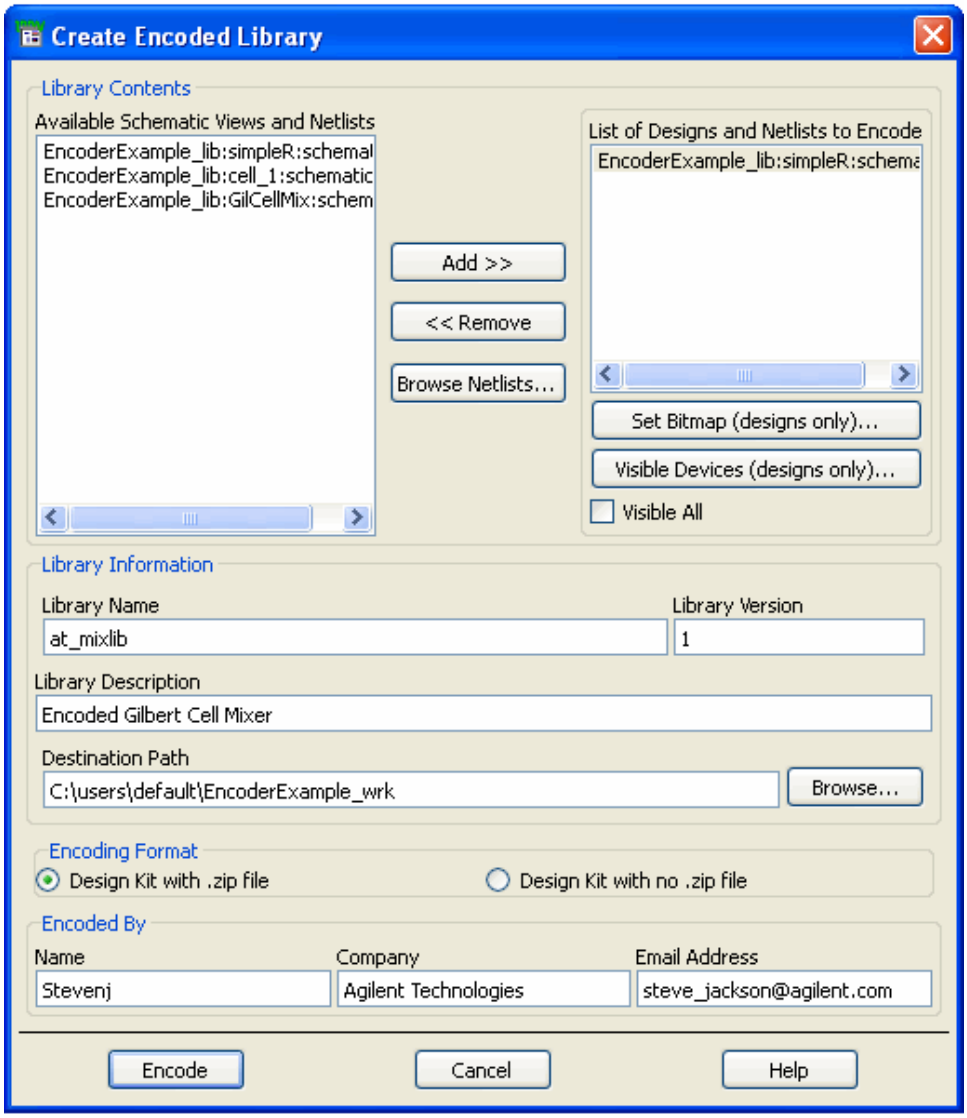
Symbols in Encoded Libraries

You can use a Pre-defined symbol to define the symbol to be used with each library component. The Pre-defined symbols can be used to create or open a circuit in the schematic editor. From the ADS main window, right-click a symbol and select **Open Symbol** to view the details of the selected symbol.

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.

1. 

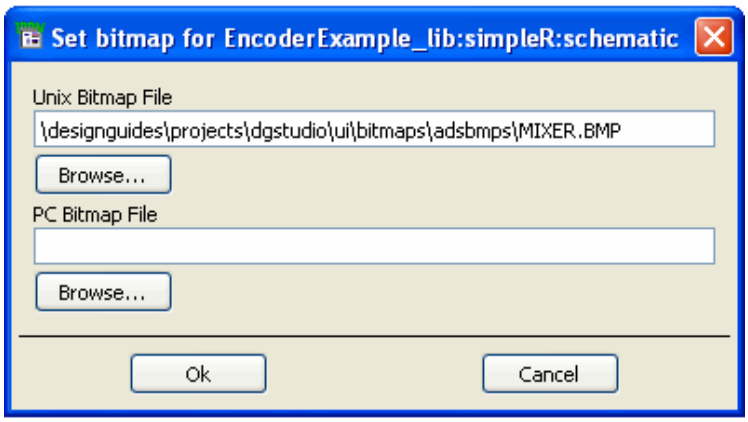
2. Use the *Library Contents* section to select which schematic designs you want to encode. In the **Available Schematic Views and Netlists** list box (on the left), highlight each individual schematic design you want to include in your library and then click **Add** . This adds each selection and displays the selected schematic designs in the **List of Designs and Netlists to Encode** list box (on the right). 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 selecting them and then clicking **Remove**.

> **Note**
> To include your own netlists in the **Available Designs and Netlists** list, place your netlists under the current workspace directory, in the *networks* subdirectory, with a *.net* extension. For example, <*your_workspace*> /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 button is

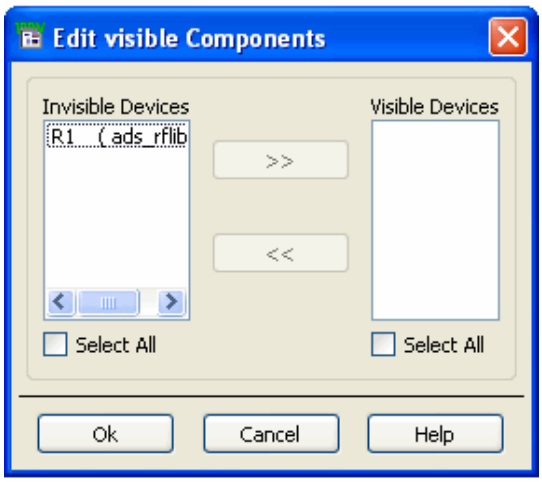
only available for schematic designs, not netlist files).

Select the mixer design file first, then click the **Set Bitmap (designs 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. By default, all the circuits and the devices being encoded have their DC device operating points suppressed. However, you have control over that and can change it. An easy way to do this is to click on the **Visible All** checkbox. This will ensure that all the devices in all the circuits that are being encrypted will have their DC device operating points accessible to the end users. Such DC device operating point information can be seen in the Data Display after running a DC simulation. Note that this is optional; it is not a required step to encode your designs.
5. If you want finer control over which devices in your circuits have their DC device operating points accessible, select a design in the list on the right, and then click on the **Visible Devices (designs only)...** button. In the dialog box that opens up, you can now select one or more devices.

5. 

6. 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.

7. Enter a version number into the **Library Version** field. The default version value is *1*.
8. Enter a clear description of your library into the **Library Description** field.
9. 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.
10. Select *Design Kit with no .zip file* .
11. 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.
12. 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.
13. Click **OK** to dismiss the Encoding Status dialog box.

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

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

Glossary for RF Intellectual Property Encoder

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.

Troubleshooting RF Intellectual Property Encoder

This section 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.

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`

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.

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

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.Before attempting to use the `hpeesofpkg` command, refer to *Configuring the Environment* (rfipenc).

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.

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

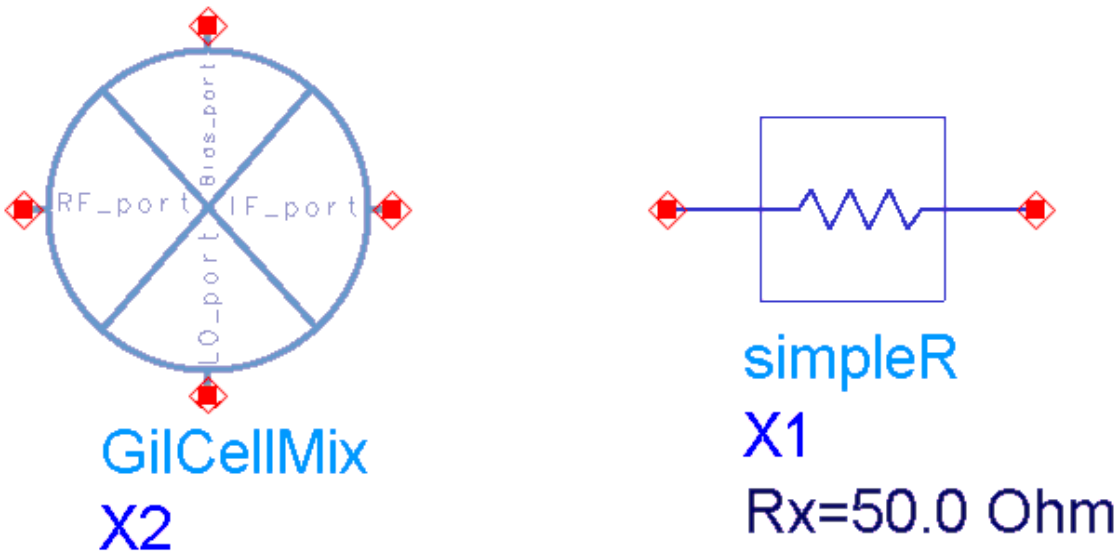
Encoding Schematic Designs and Netlists (rfipenc), described how to create an encoded library out of schematic subcircuits. It is also possible to create a library when you have an ADS netlist. This is a typical scenario for a foundry creating a design kit. In this section, you will learn how to encode 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 the *Getting Started Tutorial* (rfipenc) and created a design kit. Note that this is an advanced topic and will require manual editing of AEL and other text files.

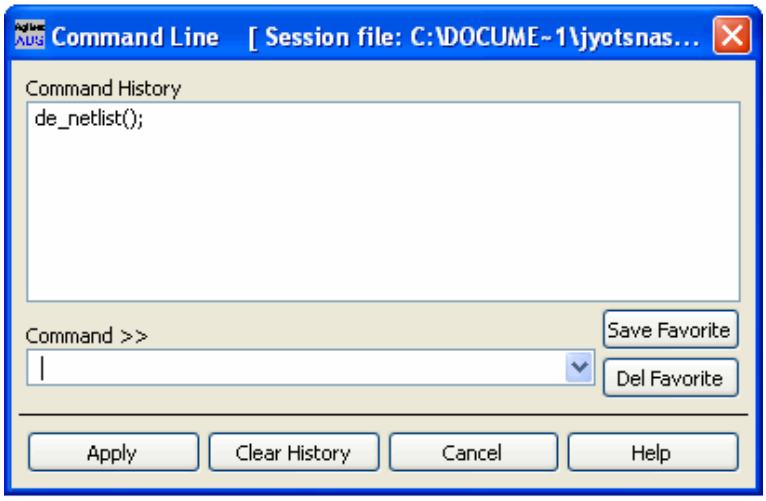
To create a netlist to encode:

1. Open the *EncoderExample_wrk* Workspace used in *Creating an Encoded IP Library* (rfipenc).
2. Create a new Schematic and place an instance of *simpleR* and an instance of *GilCellMix* as shown in Simple Design. The location and connectivity is not important so feel free to place them anywhere and do not place any wires.

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_wrk* workspace 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 workspace *networks* subdirectory as *gilcellmix.net* (note the *.net* extension). Your *EncoderExample_wrk/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* (rfipenc), 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_wrk.

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_wrk used previously in the tutorial.
4. If you previously installed the design kit at_mixlib, disable or cut that design kit from the ADS Main window menu pick **DesignKit > Manage Favorite Design Kits**.
5. After the design kit has been disabled, close and restart ADS.
6. Now install the modified design kit.
7. Open a test design, and delete and replace the GilCellMix and simpleR components.
8. Simulate the circuit.

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 for information on how to view the existing packages.