



Agilent Technologies

Encrypted HSPICE Flow

May 2007

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.

© Agilent Technologies, Inc. 1983-2007.
395 Page Mill Road, Palo Alto, CA 94304 U.S.A.

Acknowledgments

Mentor Graphics is a trademark of Mentor Graphics Corporation in the U.S. and other countries.

Microsoft[®], Windows[®], MS Windows[®], Windows NT[®], and MS-DOS[®] are U.S. registered trademarks of Microsoft Corporation.

Pentium[®] is a U.S. registered trademark of Intel Corporation.

PostScript[®] and Acrobat[®] are trademarks of Adobe Systems Incorporated.

UNIX[®] is a registered trademark of the Open Group.

Java[™] is a U.S. trademark of Sun Microsystems, Inc.

SystemC[®] is a registered trademark of Open SystemC Initiative, Inc. in the United States and other countries and is used with permission.

MATLAB[®] is a U.S. registered trademark of The Math Works, Inc.

Contents

1 Encrypted HSPICE

Encrypted HSPICE Background	1-1
Licensing Requirements	1-2
Signal Integrity Verification License	1-2
HSPICE Simulation License	1-2
Following the Encrypted HSPICE Flow	1-3
Importing an Encrypted HSPICE Subcircuit	1-3
Creating Your Design	1-7
Simulating the Design	1-7

Index

Chapter 1: Encrypted HSPICE

The encrypted HSPICE flow involves using the HSPICE simulator with encrypted models for channel verification in the Advanced Design System (ADS) environment. This feature has been designed specifically for the high-speed, serial-link Signal Integrity designer whose primary flow is ADS. These designers typically develop transmitter and receiver models and would like to perform a final verification step using vendor supplied encrypted HSPICE models (see [“Encrypted HSPICE Background” on page 1-1](#)).

If you prefer to create your channel models and designs within the ADS framework using tools such as, Momentum, Broadband Spice Model Generator, EMDS for ADS, transmission line models, or measured data such as S-parameter files, then the encrypted HSPICE flow may be a good tool for you.

The encrypted HSPICE flow is a solution developed for time-domain channel verification using an HSPICE simulator in the ADS design flow. A subset of the most useful ADS passive models and sources are supported within the encrypted HSPICE flow and the feature is simple to use and setup in the ADS environment.

Note that this feature is not a generic solution and is targeted only for Signal Integrity verification. The feature inherits all of the limitations of the HSPICE simulator and will not work in a cosimulation environment; in particular, an ADS Ptolemy Cosimulation.

Encrypted HSPICE Background

Companies that distribute models that include their intellectual property (IP) generally deliver their models as either Input/Output Buffer Information Specification (IBIS) models, HSPICE models, or encrypted HSPICE models. IBIS models tend to be faster; however, HSPICE models are generally considered to be more accurate. Therefore, supporting encrypted HSPICE models was an important aspect in the channel design/verification process. As an example, if a designer is using HSPICE models and the organization’s IP is being delivered outside of the company, this designer is almost always required to use an encrypted HSPICE format. Because of this important need to protect the IP of various companies, the Agilent Technologies Signal Integrity solution has assumed that at least part of the circuit design could include components in the form of encrypted HSPICE models; therefore, driving the need for an encrypted HSPICE flow.

For more information on the Input/Output Buffer Information Specification, refer to the [IBIS Models](#) documentation.

Licensing Requirements

In addition to your standard ADS licenses, the following product licenses are required.

Signal Integrity Verification License

A signal integrity verification license is required to use the ADS tools for encrypted HSPICE flow.

`ads_si_verification`

Note The *ads_si_verification* license is included by default in the majority of Signal Integrity and high-speed ADS bundles. You may want to check the availability of your license using your ADS license configuration tool before proceeding. For more information on ADS licensing, refer to *Setting Up Licenses* in your [Windows Installation](#) or [UNIX and Linux Installation](#) documentation.

HSPICE Simulation License

You must have the correct license for running the HSPICE simulator. The HSPICE Simulator and license are available from Synopsys, Inc. For more information on HSPICE licensing, contact Synopsys, Inc. or visit the Synopsys Web site at <http://www.synopsys.com/>.

Following the Encrypted HSPICE Flow

The encrypted HSPICE flow includes four main steps:

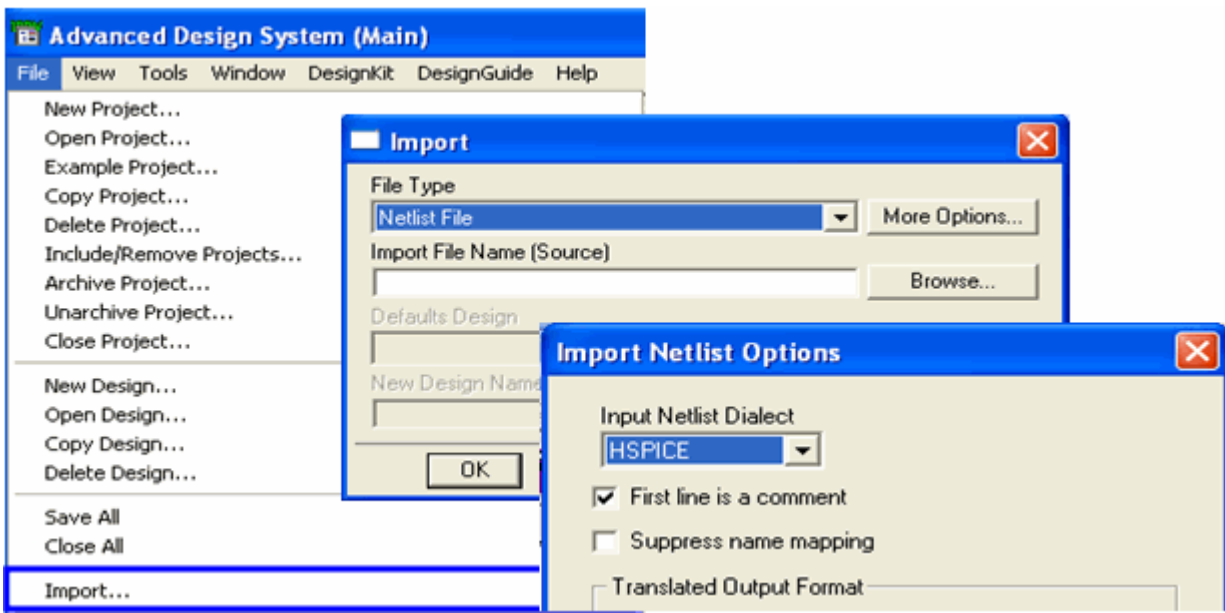
- Importing Encrypted HSPICE
- Creating Your Design
- Simulating the Design
- Evaluating Results

Importing an Encrypted HSPICE Subcircuit

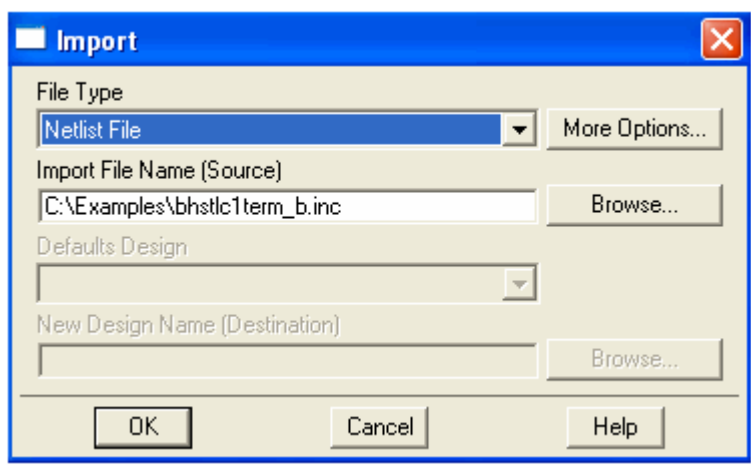
Importing an encrypted HSPICE subcircuit actually involves pointing to the encrypted HSPICE netlist fragment. A typical import provides an ADS representation of all of the files and then the external files are no longer needed. However, with the encrypted HSPICE flow, a placeholder is constructed that points to the actual file so that the location of the file can be passed off to the HSPICE simulator later.

To import your encrypted HSPICE netlist into an encrypted HSPICE subcircuit,

1. From the ADS Main Window, choose **File > Import**. The Import dialog box appears.
2. Click the **More Options** button to specify the netlist import options.
3. Using the *Import Netlist Options* dialog box, specify the following options:
 - **Input Netlist Dialect:** *HSPICE*
 - **Translated Output Format:** *ADS Schematic*



4. In the *Import* dialog box, specify the name and location of the encrypted HSPICE netlist and then click **OK**.



An ADS design will be created with a pin interface and an *HS_INCLUDE* component as shown in [Figure 1-1](#). For more information on the *HS_Include*, refer to [“The HS_Include Component”](#) on page 1-6.

Note that the name of the new ADS design is the same name as the original encrypted HSPICE netlist with the exception of the *.dsn* extension.

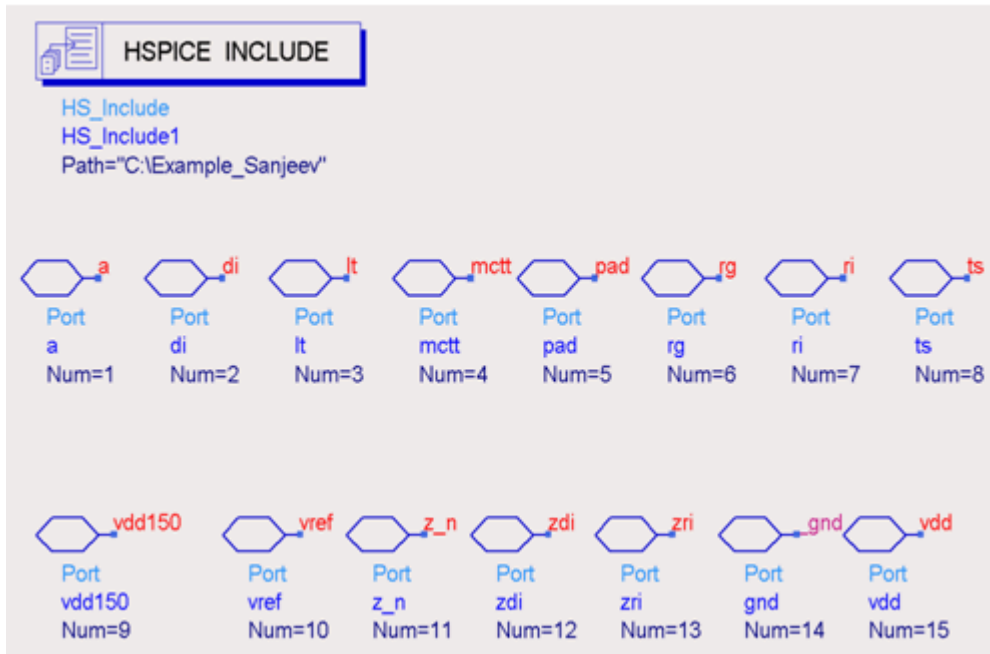
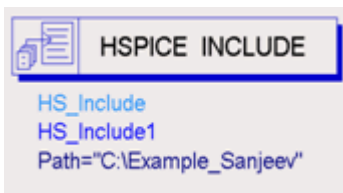


Figure 1-1. ADS Design with Pin Interface and HSPICE INCLUDE Component

The HS_Include Component

The HSPICE INCLUDE (*HS_Include*) component is designed to instruct the simulator on how to use and apply a particular subcircuit as an encrypted HSPICE subcircuit. The component has only one parameter called, *Path*. The HSPICE INCLUDE component's Path parameter is used to provide the parent directory of where the encrypted HSPICE file can be found.



Note The HSPICE_INCLUDE (HS_Include) component is automatically configured when you import an encrypted HSPICE netlist as described in [“Importing an Encrypted HSPICE Subcircuit” on page 1-3](#). The only reason you may want to modify the HSPICE_INCLUDE component's Path variable is if the original HSPICE source file is moved to a new location after you have performed the encrypted HSPICE netlist import.

To modify the path to the encrypted HSPICE netlist,

1. Double click the HSPICE INCLUDE component. The Incorporate HSPICE netlist dialog box appears.
2. In the Path field, modify the path to the parent directory of where the encrypted HSPICE file is located.
3. Click **OK** to change the path to the HSPICE INCLUDE component.

The encrypted HSPICE file must use the same name as the subcircuit that it contains with an added “.inc” extension. The ADS design, and thus the ADS subcircuit definition, must all have the same name as the encrypted HSPICE subcircuit.

Note that the HSPICE INCLUDE component is ignored by all ADS simulation controllers with the exception of the HSPICE TRANSIENT (*HS_Tran*) controller. For more information on this controller, refer to the [Encrypted HSPICE Simulation](#) documentation.

Note Both the HSPICE INCLUDE (HS_Include) and HSPICE TRANSIENT (HS_Tran) controller are available under the *Signal Integrity - Verification* component palette.

Creating Your Design

The next step involves building your circuit using the imported encrypted HSPICE subcircuit (see [“Importing an Encrypted HSPICE Subcircuit” on page 1-3](#)). Note that the name of this subcircuit will be the same as the encrypted HSPICE netlist file.

Note For the transient controller, use the *HS_Tran* controller. This is a different controller from the ADS transient controller. For more information, refer to the [Encrypted HSPICE Simulation](#) documentation.

For a channel design, most of the ADS tools that you would normally use for creating a design are available. For example,

- Momentum
- Broadband Spice Model Generator
- EMDS for ADS
- Transmission line models, such as those documented in the *Multilayer Interconnects* section of the Distributed Components manual
- Use of measured data, such as S-parameter files
- Lumped Components
- Time Domain Sources

Simulating the Design

Before attempting to simulate your design, ensure that,

- You have the proper license available for running the HSPICE simulator. For more information, refer to [“Licensing Requirements” on page 1-2](#).

- You have *hspice* added to your \$PATH environment variable. If you need to add *hspice* to your path, you will need to restart ADS after making the change.

The output from the HSPICE simulation is available to you as the simulation runs enabling you to monitor progress. Once the simulation is complete, a dataset is generated and the Data Display is open displaying the simulation results. This information can be used to evaluate your results.

Note Simulation results returned from HSPICE will have all names lowercased.

Index

- A**
 - ADS
 - Ptolemy Cosimulation, 1-1
 - tools, 1-7
 - transient controller., 1-7
 - C**
 - channel verification
 - time-domain, 1-1
 - component
 - palette, 1-7
 - cosimulation, 1-1
 - E**
 - encrypted HSPICE
 - flow, 1-1, 1-3
 - models, 1-1
 - environment variables
 - \$Path, 1-8
 - F**
 - file extensions
 - .inc, 1-6
 - H**
 - HS_Include, 1-6
 - HS_Tran, 1-6
 - HSPICE
 - include, 1-6
 - licensing, 1-2
 - models, 1-1
 - simulation controller, 1-6
 - hspice
 - environment variable, 1-8
 - I**
 - IBIS models, 1-1
 - importing
 - subcircuits, 1-3
 - incorporate HSPICE netlist, 1-6
 - intellectual property, 1-1
 - L**
 - licensing, 1-2, 1-7
 - limitations, 1-1
 - N**
 - netlist options, 1-3
 - P**
 - parameters
 - Path, 1-6
 - pin interface, 1-4
 - S**
 - signal integrity
 - verification license, 1-2
 - T**
 - time-domain
 - channel verification, 1-1
-

