Agilent Technologies

Advanced Design System 2011.01

Feburary 2011
Encrypted HSPICE Flow

Acknowledgments

SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE 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).

Encrypted HSPICE

Note
Please consider using the native *HSPICE Compatibility* (hspice) support available in ADS to read HSPICE netlists directly into ADS instead of the Encrypted HSPICE flow. Encrypted HSPICE is not a generic solution and is **targeted only for signal integrity verification**. It inherits all of the limitations of the HSPICE simulator and has some limitations itself; in particular, it isn't compatible with an ADS Ptolemy Cosimulation. Thus, *HSPICE Compatibility* (hspice) is superior in most cases. To use HSPICE Compatibility, you must however request that your IC vendor supply you with HSPICE netlists that are encrypted using the ADS *RF Intellectual Property Encoder* (rfipenc), and use them for your ADS simulations. You can send a copy of this letter to your vendor to explain the process.

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. The Encrypted HSPICE flow involves using an actual HSPICE simulator license and models encrypted with the Synopsys encryption key for channel verification in the ADS environment. This feature has been designed specifically for the high-speed, serial-link signal integrity designer whose primary flow is ADS, including Momentum, Broadband Spice Model Generator, FEM Simulator, transmission line models, and measured data such as S-parameter files. A typical flow is to develop transmitter and receiver models and then to perform a final verification step using encrypted HSPICE models from the IC vendor.

Encrypted HSPICE Background

IC vendors that distribute models to their customer generally protect their intellectual property (IP) by using either Input/Output Buffer Information Specification (IBIS) models, or encrypted HSPICE models. IBIS models tend to be faster; however, encrypted HSPICE models are generally considered to be more accurate. Therefore, supporting encrypted HSPICE models is an important aspect in the channel design/verification process. Although ADS can run *unencrypted HSPICE netlists natively* (hspice), only Synopsys has access to the decryption key that "unlocks" encrypted models. If your IC vendor is unwilling to follow the recommend procedure of *re-encrypting their models using our encryption key* (rfipenc), then it is necessary to use the Encrypted HSPICE flow described here.

(For more information on ADS support for the Input/Output Buffer Information Specification, refer to the *IBIS Models* (ibis) documentation.)

Licensing Requirements

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

W2302 ADS Transient Convolution Element License

A signal integrity verification license line item *ads_si_verification* is required for Encrypted HSPICE flow. This license line item is included in W2302 ADS Transient Convolution Element and the bundles that contain it. (If you are using the older product structure, the license was part of E8828A Signal Integrity Toolkit module.)

Note
You may want to check the availability of your license using your ADS license configuration tool before proceeding. Look for the *ads_si_verification* increment line. For more information on ADS licensing, refer to *Setting Up Licenses* in your *Windows Installation* (instalpc) or *UNIX and Linux Installation* (install) 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/ .

Summary Table: Running HSPICE Netlists in ADS

Netlist type	Flow	License(s) required	Pros	Cons
Unencrypted HSPICE	*HSPICE Compatibility* (hspice)	ADS Transient Convolution	Faster than HSPICE	None
HSPICE netlist encrypted with EEsof encryption key	*HSPICE Compatibility* (hspice)	ADS Transient Convolution	Faster than HSPICE	IC Vendor might be unwilling to encrypt their HSPICE netlist with EEsof encryption key
HSPICE netlist encrypted with Synopsys encryption key	*Encrypted HSPICE* (hspicecosim)	ADS Transient Convolution plus Synopsys HSPICE	Models are widely available	Cosimulation overhead; Ptolemy not available

Following the Encrypted HSPICE Flow

The encrypted HSPICE flow includes four main steps:

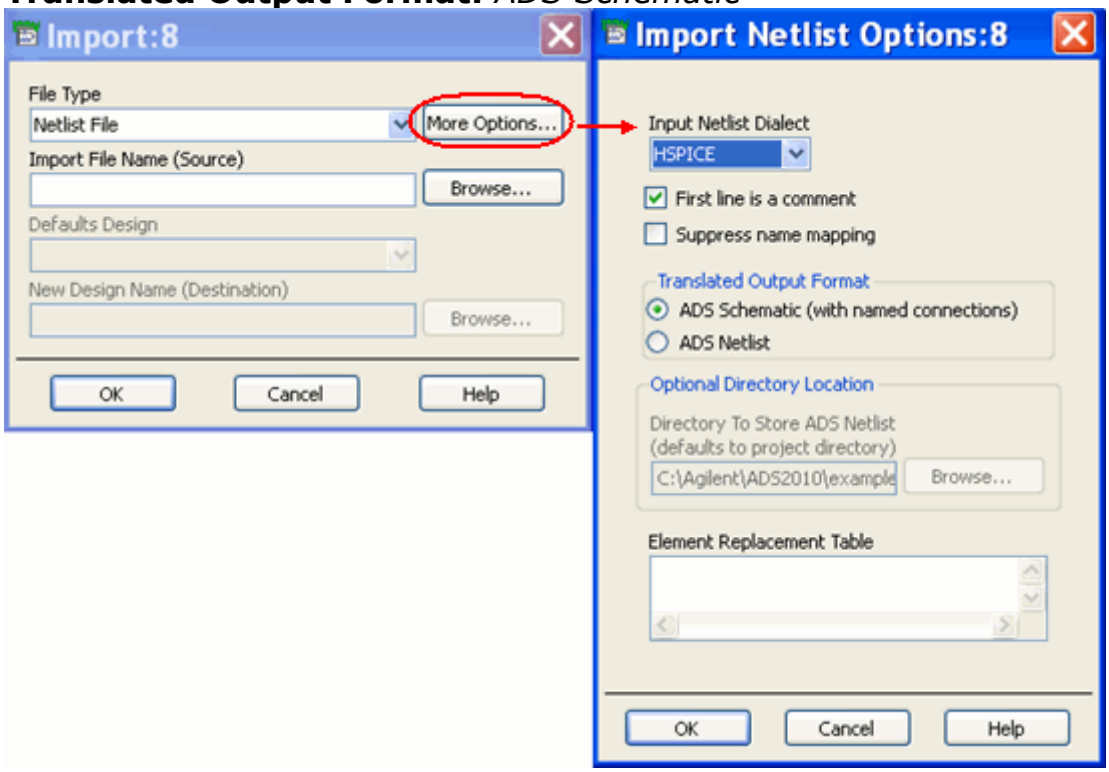
- Importing Encrypted HSPICE
- Creating Your Design
- Simulating the Design
- Evaluating Results

Importing an Encrypted HSPICE Subcircuit

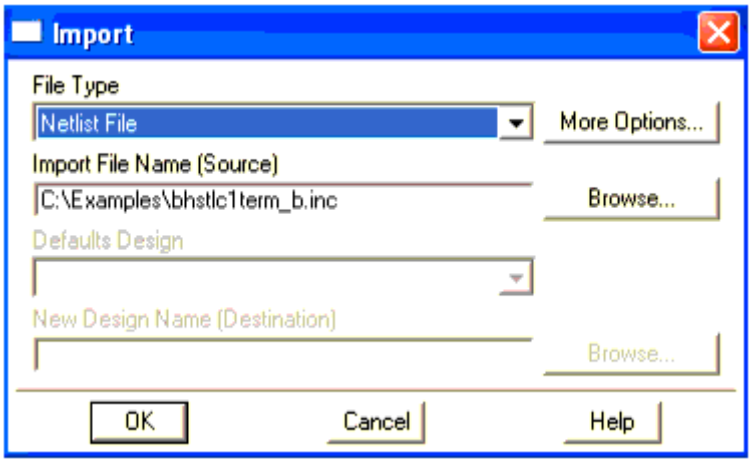
Importing an encrypted HSPICE subcircuit actually involves pointing to the encrypted HSPICE netlist fragment. A typical import provides an ADS representation of all of the files and then the external files are no longer needed. However, with the encrypted HSPICE flow, a placeholder is constructed that 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 Schematic 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:
    1. **Input Netlist Dialect:** *HSPICE*
    2. **Translated Output Format:** *ADS Schematic*



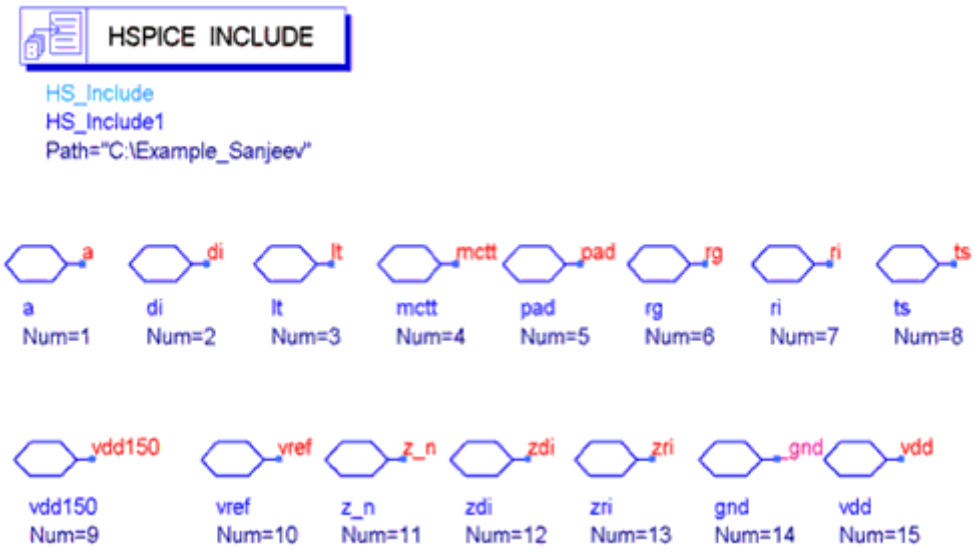
4. In the *Import* dialog box, specify the name and location of the encrypted HSPICE netlist and then click **OK** .



An ADS design will be created with a pin interface and an *HS_INCLUDE* component as shown in ADS Design with Pin Interface and HSPICE INCLUDE Component. For more information on the HS_Include, refer to The HS_Include Component.

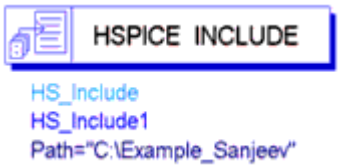
Note that the name of the new ADS design is the same name as the original encrypted HSPICE netlist.

Figure: 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. 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* (cktsimhstran) 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). 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* (cktsimhstran) 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

- FEM Simulator
- Transmission line models, such as those documented in *Multilayer Interconnects* (ccdist)
- 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.
- 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.

Modifying HSPICE Kits for use in ADS

Some HSPICE kits must be modified into a form that ADS can use.
The example procedure below is provided to help you with modifying the Cyclone kit from Altera Corporation. The Altera HSPICE kit can be downloaded from:
http://www.altera.com/support/software/download/hspice/hsp-index.html
The intent of this procedure is to package the encrypted portions of the HSPICE kit into modular subcircuits, which can then be used by the ADS netlist import and simulation processes. After downloading and unpacking the HSPICE kit:

1. Open the test netlist (e.g. Cyclone_1p5_8ma_inc.sp).
2. Find the subcircuit definition that wraps the encrypted content (CYCLONE_SINGLE_ENDED_IO, line 186).
3. Create a new file using the same name as the subcircuit, with an extension of ".inc" (cyclone_single_ended_io.inc).
4. Copy the ".subckt" to ".ends" lines (lines 186-192) to the new file. To comply with HSPICE conventions, make the first line of the file a comment (e.g. start it with "*").
5. Replace the ".include" line (line 6) with the contents of the indicated file (encrypted_cyclone_single_ended_io.inc).
6. For any files that are referenced in ".include" lines in the test netlist (e.g. './models/Cyclong_typ_model.inc' and 'encrypted_c6851.inc'), copy those ".include" lines to the subcircuit file being created (cyclone_single_ended_io.inc). Insert the ".include" lines either just before the ".subckt" line or just after the ".ends" line.

7. Within an ADS schematic window,
   - Choose **File > Import** .
   - Select *Netlist File* as the File Type and then click **More Options** .
   - Select *HSPICE* from the Input Netlist Dialect drop-down list.
   - Then select the new ".inc" subcircuit that was created in steps 3 through 6 above (cyclone_single_ended_io.inc). The import process should create a PDE design file for the imported subcircuit (cyclone_single_ended_io).
8. In a new PDE design, place an instance of the subcircuit (cyclone_single_ended_io). Now design the channel, and place an HS_Tran controller to enable the encrypted HSPICE simulation.
9. When simulating, the HSPICE simulator reports errors indicating "no definition for rnnmd" and "no definition for rptmd". These are global variables that are referenced by the encrypted subcircuits. Edit the HS_Tran controller, and select the "Options" tab. In the "Global Parameters" section, define values for "rptmd" and "rnnmd". Save the new values and simulate again.

The netlist importer expects encrypted HSPICE files to be in exactly this format:

```
.subckt ...
.param ...
.prot ...
.ends
```

That is, a single encrypted section inside of a subcircuit definition, with the associated pin names and parameter names and values. Any deviation from this - for example, unencrypted lines inside or outside of the subcircuit definition, multiple encrypted sections, etc. - will cause trouble for the netlist importer.