Synthesis Tools Installation Notes
Version A-2007.12

December 10, 2007

These installation notes present information about installing Synopsys synthesis tools in the following sections:

To ensure a successful installation, complete the following procedures before beginning the installation process:

- Create the Synopsys root directory.
- Define the `SYNOPSYS` environment variable.

See also http://www.synopsys.com/install for additional installation and licensing information.

Synthesis Tools

This installation note covers the following Synopsys synthesis tools:

- Automated Chip Synthesis
- BSD Compiler
- Design Compiler
- Design Vision
- DesignWare
- DFT Compiler
- HDL Compiler (Presto Verilog)
- HDL Compiler (Presto VHDL)
- Library Compiler
- Module Compiler
- Power Compiler
- VHDL Compiler

If you have purchased any of these tools, you must install the synthesis tools suite.

Media Availability and Supported Platforms

The synthesis tools are are available available by EST download upon initial software release, and at a later date on DVD (or CD depending on image size). Obtain the appropriate binary executable files based on the operating system you need. Table 1-1 shows the supported platforms for the A-2007.12 release. Most synthesis tools install on all platforms.

Note:
You must download a common file and a platform file for synthesis.

Table 1-1 Supported Platforms and Keywords

Platform	Operating system	Synopsys platform keyword
AMD Opteron	Red Hat Enterprise Linux v4, v5[1]	amd64 (64-bit mode) linux (32-bit mode)
EMT64T	SUSE Enterprise Linux 9, 10	suse64 (64-bit mode) suse32 (32-bit mode)
IA-32 (X86)	Red Hat Enterprise Linux v4, 5[1]	linux (32-bit mode)
IBM RS/6000	AIX 5.3	rs6000 (32-bit mode) aix64 (64-bit mode)
Sun SPARC	Solaris 9, 10[1]	sparcOS5 (32-bit mode) sparc64 (64-bit mode)

1. Binary-compatible hardware platform or operating system. Note, however, that binary compatibility is not guaranteed. See http://www.synopsys.com/products/platforms/a-foundation.html for latest information.

Disk Space and Memory Requirements

The synthesis tools have the following minimum memory requirements:

- Physical Memory – 128 MB (1 GB recommended)
- Swap space – 256 MB (2 GB recommended)

For large designs, the expected amount of required memory is approximately 1 million bytes per 2,000 gates.

The disk space requirement varies depending on the platform and tool selected for installation. During the installation process, Synopsys Installer displays the required disk space.

Swap Space Requirements for Synthesis Tools

The amount of swap space required by the synthesis tools depends on the size and type of each circuit design.

Use the following formula to help you determine the minimum amount of available swap space required for HDL designs:
16.3 MB + (5.9 x (size of the design in K gates)).

For example, a 5K-gate design requires 16.3 + (5.9 x 5) = 45.8 MB of available swap space.

Accessing Memory Beyond 2 GB With 32-Bit Synthesis Tools

Synthesis tools can extend memory beyond 2 GB. Note that available memory is calculated as space not used by the OS, the windowing system, or other applications. The following synthesis tools can extend memory beyond 2 GB:

- Design Compiler
- DFT Compiler
- HDL Compiler (Presto Verilog)
- HDL Compiler (Presto VHDL)
- Power Compiler

  Note:

  Available memory is space not used by the OS, the windowing system, or other applications.

To access memory beyond 2 GB,

1. Make sure your server has Solaris 9 (or later) loaded.
2. Make sure your server has at least 4 GB of memory (physical and swap space) available.

   Note:

   Physical memory equals data size plus stack size, and stack size is used before data size. Therefore setting stack size to a large value causes problems for designs that need to go over 2 GB. If you set the stack size too high, you cannot get enough memory for your data. To check the settings, use the `limit` command at the system prompt.
3. Make sure the system you are using does not have restrictions that prevent you from using more than 2 GB of memory.
4. Create unlimited data size in the shell that you are using: C, Bourne, Korn, or Bash. If there are system-wide limits on the data size you can create, you can remove them or override them. You can do this in one of two ways:

   - Enter one of the following commands:

For the C shell,

```
% limit datasize 3800000
```

For the Bourne, Korn, or Bash shell,

```
# ulimit -S -d 3800000
```

- Modify the kernel of your server. This approach allows everyone using your server to extend memory beyond 2 GB.

Installing the Software

The synthesis tools use the Synopsys Installer tool, which allows you to use a text script or a graphical user interface (GUI). For information about downloading the Synopsys Installer and synthesis, see the document *Installing Synopsys Tools* at http://www.synopsys.com/install.

To install synthesis tools, follow the procedures described in *Installing Synopsys Tools*. This document provides a Synopsys media installation script. The synthesis tools are installed in a similar manner. They cannot be installed over an existing Synopsys product, including a prior versions. You must create a new directory for to install synthesis products.

Configuring the Synthesis Tools

Configuring the synthesis tools is described in the following sections:

- Setting Up the User Environment
- Setting the SNPSLMD_LICENSE_FILE Environment Variable

Setting Up the User Environment

A platform-independent wrapper script is provided for the synthesis tools. This script automatically determines the operating system platform at runtime, which simplifies the setup required to use the synthesis tools.

The platform-independent wrappers are located at $SYNOPSYS/bin and include the following new options:

```
-32bit|-64bit
```

Note:
If you select an executable file that is not available, you will automatically be switched to an available platform based on your current environment. A warning message will not be issued.

To set up a new synthesis tools user, add the directory for the synthesis executable files to the PATH environment variable.

- If you are using the C shell, add the following line to the .cshrc file:

```
set path=($SYNOPSYS/bin $path)
```

- If you are using the Bourne, Korn, or Bash shell, add the following line to the .profile, .kshrc, or .bashrc file:

```
PATH=$SYNOPSYS/bin:$PATH
export PATH
```

When you install the synthesis files, a copy of the synthesis setup file is placed in $SYNOPSYS/admin/setup/.synopsys_dc.setup. The .synopsys_dc.setup file contains the system defaults for the synthesis tools.

Setting the SNPSLMD_LICENSE_FILE Environment Variable

You must install the SCL software and define the SNPSLMD_LICENSE_FILE variable before you can verify the synthesis installation.

For information about downloading SCL, installing SCL, or setting the license variable, see *Installing Synopsys Tools* at http://www.synopsys.com/install.

Configuring the Browser for Design Vision Online Help

The online Help system is a browser-based HTML Help system designed for viewing in Firefox and Mozilla Web browsers. You can use any of the following browsers:

- Firefox version 1.5 or 2.0
- Mozilla version 1.7

Although online Help might work in Internet Explorer version 6.0 or later, it is not tested in this browser.

Important:

When you use online Help from within the GUI, the directory containing the browser executable file must be in the search path specified by your UNIX or Linux $PATH variable.

The default browser is Firefox. To use a different browser, change the `gui_online_browser` variable in your .synopsys_dv_gui.tcl setup file. You can also change the browser for the current session by setting this variable from within the GUI. For example,

```
set gui_online_browser "mozilla"
```

The choices are `firefox` or `mozilla`.

Installing Optional Tools

The synthesis media installation script automatically installs most of the synthesis tools. However, the following tools require manual setup or installations:

- Power Compiler VPOWER
- SoCBIST

Installing Power Compiler VPOWER

VPOWER is the Power Compiler interface to VCS, the Cadence Verilog-XL and NC-Verilog simulators, and the MTI Verilog simulator. VPOWER contains user tasks that allow you to monitor toggle activity during simulation and to output the information in a form readable by Power Compiler. To use VPOWER, link the user tasks to the executable file of your simulator.

The following sections describe the steps for static-linking VPOWER with Verilog-XL and VCS simulators only. For information about linking VPOWER with other simulators, see the *Power Compiler User Guide*.

Verilog-XL Simulator

The following procedure describes how to link VPOWER to a version of the Verilog-XL simulator that contains the standard features you normally use at your site and includes the toggle count utilities needed for Power Compiler.

Note:

You must perform this installation on a machine that has access to your Verilog-XL simulator vendor distribution.

Consult your Verilog system administrator to obtain the following information before beginning the VPOWER installation:

- The directory path to your Verilog .o, .a, and .h files
- The directory location of your central Verilog distribution, for obtaining a current site copy of the veriuser.c file

This installation requires modification of your veriuser.c file. By obtaining a current site copy of the veriuser.c file, you can be sure to include any current site modifications when you modify this file.

To install VPOWER,

1. Change to the Synopsys vpower directory.
2. Modify a copy of your site veriuser.c file.
3. Link the VPOWER user tasks to the simulation executable file.
4. Copy the linked executable file.

The following sections describe these steps.

Changing to the Synopsys power Directory. All directories listed are relative to the root of the vpower directory: $SYNOPSYS/auxx/syn/power/vpower.

To change to the Synopsys vpower directory,

1. Make sure the $SYNOPSYS environment variable is set.

   ```
   % echo $SYNOPSYS
   ```

   If it is not set, set it to the correct value.

   ```
   % setenv SYNOPSYS root_directory
   ```

2. Change to the Synopsys vpower directory.

   ```
   % cd $SYNOPSYS/auxx/syn/power/vpower
   ```

Modifying the veriuser.c File. To modify the veriuser.c file to define the new toggle count utilities,

1. Change to the vxl/vxl.sample directory, and review the sample veriuser.c file, which shows the edits you will have to make.

   ```
   % cd vxl/vxl.sample
   ```

2. Copy your current site version of veriuser.c into the sample directory. To copy veriuser.c, you must know the directory location of your central Verilog distribution.

   ```
   % cp site_location_dir_path/veriuser.c .
   ```

   By using a current site copy of veriuser.c, you ensure that any existing customizations are included in the VPOWER installation.

3. As shown in the sample veriuser.c file, make the following changes in your current site copy of veriuser.c:

   - Add the following line:

     ```
     # include "tc_extern.h"
     ```

   - Add the following user tasks:

```
{usertask, 0, 0, 0, tc_set, tc_set_sync, "$toggle_set", 1},
{usertask, 0, 0, 0, tc_start, 0, "$toggle_start", 1},
{usertask, 0, 0, 0, tc_stop, 0, "$toggle_stop", 1},
{usertask, 0, 0, 0, tc_reset, 0, "$toggle_reset", 1},
{usertask, 0, 0, 0, tc_compatibility, 0, "$toggle_count", 1},
{usertask, 0, toggle_report_check, 0, toggle_report, 0, "$toggle_report", 0},
{usertask, 0, 0, 0, read_lib_saif, tc_lib_sync, "$read_lib_saif", 1},
{usertask, 0, 0, 0, read_rtl_saif, tc_set_sync, "$read_rtl_saif", 1},
```

   - Comment out the following line:

     ```
     char *veriuser_version_str = "";
     ```

4. Save your modified veriuser.c file.

5. Exit your text editor and remain in the sample directory to link the executable file.

Linking User Tasks to the Simulation Executable File. VPOWER provides two ways to link the user tasks to your simulator executable file: by using the vconfig utility or by using a UNIX makefile. Each method links your simulator to the VPOWER user tasks. Choose the method that you find familiar or comfortable.

Using vconfig to Link the Executable File. The vconfig utility creates a script called cr_vlog. The cr_vlog script links your Verilog-XL simulator's executable file to the VPOWER user tasks. You must define the name of the executable file created by cr_vlog, for example, verilog_toggle.

To use the vconfig method to link your executable file,

1. Use your vconfig utility or an equivalent utility to generate the cr_vlog script or an equivalent script.

2. In the script, set an environment variable pointing to the directory of the generated library archive. For example (if you are using Solaris 7 or later),

   ```
   setenv PPLILIB "../../lib-sparcOS5/libvpower.a"
   ```

3. In cr_vlog, look for the line that includes the math libraries:

   ```
   -lm \
   ```

4. Add a line above this line to include the libvpower.a library. For example,

   ```
   $PPLILIB \
   -lm \
   ```

5. Run cr_vlog.

   ```
   % cr_vlog
   ```

   This script links your executable file to the VPOWER user tasks and creates the customized executable file called verilog_toggle. For details about linking the programmable language interface (PLI) by using the vconfig utility, see the *Power Compiler User Guide*.

Proceed to "Copying the Linked Executable File."

Using a Makefile to Link the Executable File. Using the UNIX `make` command, you can use a makefile to link your Verilog-XL executable file to the VPOWER user tasks. The makefile creates a modified executable file called verilog_toggle.

Two makefiles exist: Makefile.sol and Makefile.hp.

To use the makefile method to link your executable file,

1. Using a text editor such as vi, edit the appropriate makefile to set variable values for `VERILOG_LIB` and `VERILOG_INC`.

   Modify the lines in the makefile to read according to your data. For example, enter

   ```
   VERILOG_LIB = path1
   VERILOG_INC = path2
   ```

   where *path1* is the path to your Verilog distribution .o and .a files, and *path2* is the path to your Verilog distribution .h files.

   The `VERILOG_LIB` variable must point to the directory path of the vlog.o and omnitasks.o files. The `VERILOG_INC` variable must point to the directory path of the acc_user.h and veriuser.h files.

2. Save the modified makefile and exit your text editor.

3. Use the make utility to link the executable file.

   % **make -f Makefile.*platform***

   The *platform* extension is sol or hp.

   The `make` command uses the modified makefile to link your executable file, creating a customized executable file called verilog_toggle.

Copying the Linked Executable File. After you create your customized executable file, change the permissions so that the file is not writable, and copy it to a directory suitable for group access.

Enter the following commands at the UNIX prompt:

```
% chmod ogu-w verilog_toggle
```

This removes write access to other, group, and user.

```
% cp verilog_toggle site_verilog_bin_location
```

This copies the file to the site_verilog_bin_location directory for group access.

VCS Simulator

The following procedure describes how to link VPOWER to a version of VCS that contains the standard features you normally use at your site and includes the toggle count utilities needed for Power Compiler.

Note:
The PLI library has been tested with VCS version 3.0 and later versions.

To install VPOWER,

1. Change to the Synopsys vpower directory.
2. Modify a copy of the PLI table file.
3. Compile the simulation executable file.

The following sections describe these steps.

Changing to the Synopsys vpower Directory. All directories listed are relative to the root of the vpower directory: $SYNOPSYS/auxx/syn/power/vpower.

1. Make sure the `$SYNOPSYS` environment variable is set.

   % **echo $SYNOPSYS**

If it is not set, set it to the correct value.

```
% setenv SYNOPSYS synthesis_root_directory
```

2. Change to the Synopsys vpower directory.

```
% cd $SYNOPSYS/auxx/syn/power/vpower
```

Modifying the PLI Table File. To modify the PLI table file (vpower.tab) to define the new toggle count utilities,

1. Change to the vcs/vcs.sample directory, and review the sample vpower.tab file, which shows the edits you will have to make.

```
% cd vcs/vcs.sample
```

2. Make the necessary changes to the vpower.tab file.

Compiling the Simulation Executable File. VCS is a compiled simulator, so you must compile your designs along with VCS libraries to make a simulation executable file. To add PLI functionality to the simulation executable file, you need to link an extra PLI library when you compile your designs.

For Solaris the appropriate PLI library is

../../lib-sparcOS5/libvpower.a

You normally get a VCS simulation executable file by entering the following command at the UNIX prompt:

```
% vcs -Mupdate your_verilog_design_files compiler_options
```

To link with the PLI library, enter

```
% vcs -Mupdate  -P $SYNOPSYS/auxx/syn/power/vpower/vcs/vcs.sample/vpower.tab \
  your_verilog_design_files compiler_options \
  $SYNOPSYS/auxx/syn/power/vpower/lib-sparcOS5/libvpower.a
```

This generates an executable file called simv that includes PLI functionality.

Note:

You can copy vpower.tab and libvpower.a into any file locations that are convenient for you.

Using SoCBIST

To insert SoCBIST into your design, you need DFT Compiler, which installs with the synthesis tools. You also need TetraMAX if you want to use the SoCBIST pattern generation functionality. You can install TetraMAX as an overlay on the synthesis tools or as a stand-alone installation. For required SoCBIST environment variables, see “Setting Up the User Environment”.

Verifying the Synthesis Tools Installation

To verify installation,

1. Make sure you are in a directory where you have read/write privileges.

   ```
   % cd $HOME
   ```

2. Invoke the synthesis tools on a licensed machine. For example, invoke Design Compiler, Library Compiler, or Design Vision by entering one of the following commands:

   ```
   % $SYNOPSYS/bin/dc_shell-xg-t

   % $SYNOPSYS/bin/lc_shell

   % $SYNOPSYS/bin/design_vision
   ```

Note:

You can verify other synthesis tools by using the preceding command. Simply replace the executable file name with the name of another synthesis tool.

If you get the correct prompt, or if a GUI appears, the installation was successful.