Synopsys Design Compiler Tutorial

ECE 551 - Design and Synthesis of Digital Systems
Spring 2002

This document provides instructions, modifications, recommendations and suggestions for performing the ***Synopsys Design Compiler Tutorial***. You will be viewing this tutorial on-line as you execute it using ***Design Compiler***. A PDF version of the tutorial broken down into chapter files with hyperlinks to sections can be accessed in the CAE UNIX system at:

/afs/engr.wisc.edu/apps/eda/synopsys/doc/online/synth/dctut/toc.pdf

NOTE: Links will only work when viewed in Adobe Acroreader, not the Netscape Acroread plug-in.

A complete version for printing can be accessed at:

/afs/engr.wisc.edu/apps/eda/synopsys/doc/online/synth/print/dctut.pdf

1. Operating System

You will be working only on Sun workstations at CAE. Ignore all portions of the tutorial that describe the use of Windows NT.

2. Choosing an Interface to Use (p. 1-6)

As part of ***Design Compiler***, Synopsys provides a graphical interface called ***Design Analyzer*** and a command line interface call **`dc_shell`**. The **`dc_shell`** supports two scripting languages – **`dcsh`**, which uses the Synopsys language, and **`dctcl`**, which uses **Tcl** (Tool Command Language). It is recommended that ***Design Analyzer*** be used for most of the synthesis and optimization processes. The **`dc_shell`** is preferable for a standardized synthesis methodology or optimization of large designs. To use both interfaces simultaneously, open the ***Command Window*** from ***Design Analyzer*** and use **`dc_shell`** commands. Also, while using **`dc_shell`**, it is recommended that **`dcsh`** script be used because it is easier to understand if you do not know the **Tcl** scripting language.

3. Chapters 2-4

These chapters are a prologue to the actual tutorial, which begins in Chapter 5. An overview of ***Design Compiler*** is given in these chapters. Only in the case of an extreme time crunch is it recommended that you skip these chapters.

4. Creating Tutorial Directories (pp. 5-4)

Before you actually begin using ***Design Compiler***, you will need to copy the tutorial files. They can be copied using the following command (which replaces Step 2 on p. 5-4):

```
cp -r /afs/engr.wisc.edu/apps/eda/synopsys/doc/syn/tutorial .
```

Proceed to change to your tutorial directory and continue the tutorial.

The remainder of Chapter 5 is more information about files that will be used as part of the tutorial. Chapter 6 is a description of the design that will be synthesized and subsequently optimized.

5. Invoking Design Compiler

Be sure you are in your tutorial directory before you invoke either of the following (because the setup files are in this directory):

Design Analyzer: Type **`design_analyzer`** on the command line.

dc_shell: Type **`dc_shell`** on the command line. This launches **`dc_shell`** in the **`dcsh`** mode. Typing **`dc_shell –tcl_mode`** will launch **`dc_shell`** in the **`dctcl`** mode.

6. Reading in a Hierarchical Design (p. 7-6)

Throughout the tutorial, the use of VHDL is to mean HDL. Therefore, VHDL stands for Verilog HDL as well as VHDL. In ECE 551, we analyze and synthesize Verilog. Therefore, make sure that you read Verilog source files from the **`./tutorial/verilog`** directory. If you refer to page 7-3, the difference is spelled out. However, the bullet there is somewhat ambiguous. It says "Use the VHDL analyze and elaborate procedures to read your designs." This sentence does NOT mean that you choose or click on VHDL while analyzing or elaborating Verilog files. All it means is that the same steps to analyze and elaborate VHDL files are used for Verilog files. However, when reading these steps, substitute Verilog(verilog) for VHDL(vhdl) and **`v`** for **`vhd`**. Note that **.v** stands for Verilog whereas **.vhd** stands for VHDL.

When using **`dc_shell`**, the format for the read command changes. If you refer to p. 7-49, it shows how to read VHDL files. In order to read the corresponding Verilog file from the verilog directory, type:

```
read –format verilog {"./verilog/synopsys.v"}
```

7. Not-So-Trivial Details

a) Although the tutorial may seem a massive undertaking, it is likely to take only from 2 to 4 hours depending on your experience and your interest in establishing a strong foundation for your use of ***Design Compiler***. The tutorial looks at multiple ways of doing things. We recommend that you do the ***Design Analyzer*** approach first through to completion while keeping the Command Window open so that you can study the dc_shell equivalents to the GUI operations (On the File menu, choose **Setup** → **Command Window** to launch the ***Command Window***.). This will speed up the learning curve when you go back to complete the dc_shell portion of the tutorial.

b) Chapters 7, 8, and 9 are the core of the tutorial. You should try to fully understand and complete these steps of the tutorial. Chapter 10 is just an extension of the ideas on Chapters 7 through 9.

c) ***Appendix A*** shows how to write scripts for performing the same steps covered in the tutorial. You will find scripts extremely useful in the future when you are working on your project with more elaborate designs.

8. Technology Libraries

For this tutorial, you are using dummy technology libraries provided by Synopsys (**`class.db`**). In homework problems and the project, we will use LSI Logic's technology libraries, **`lcb11p`**. To use the technology libraries provided by LSI Logic, a different **`.synopsys_dc.setup`** file is needed. These files are provided in the Tutorials folder on the website and on the CAE UNIX systems:

Nominal Delay
/pong/usr5/e/ece551/public_html/technology_libraries/lsi_nom/.synopsys_dc.setup

Worst Case Delay
/pong/usr5/e/ece551/public_html/technology_libraries/lsi_wc/.synopsys_dc.setup

To use either of these libraries, re-launch ***Design Compiler*** from the directory in which the **`.synopsys_dc.setup`** file is stored. Since each of the libraries contain different standard cells, synthesized HDL designs will differ.