

# Synopsys Design Compiler Tutorial

## Addendum to GWU tutorial for SMU students

T. Manikas

Updated 12/9/20

Note: This document supplements the Synopsys Design Compiler material originally developed by T. Farmer and W. Gibb at George Washington U. (GWU) for their ECE 128 course in Spring 2011 (file name is "lab2\_synopsys\_dc"). I have modified some of their material to cover items specific to Southern Methodist U. (SMU) computing resources. Please use their tutorial as the main document, but follow my specific instructions below for given sections of their tutorial.

### Part I: OVERVIEW

Synopsys Design Compiler (SDC) is an RTL compiler. An RTL compiler takes an RTL version of a design (such as Verilog) and transforms (compiles) the RTL by mapping the design to components in a standard cell library (such as logic gates). The mapping decisions are performed to meet various design objectives (area, timing, power).

The basic steps for RTL compilation are the following:

1. Analyze and Elaborate
2. Apply Design Constraints
3. Optimize and Compile
4. Inspect Results

### Part II: PREPARATION

1. Log into your Lyle Unix account and create a working directory for SDC. For example, Farmer and Gibb use the directory **lab2** for their tutorial.
2. From my RTL Compiler website [http://lyle.smu.edu/~manikas/CAD\\_Tools/SDC/SynopsysDesignCompiler.html](http://lyle.smu.edu/~manikas/CAD_Tools/SDC/SynopsysDesignCompiler.html), copy the file `.synopsys_dc.setup` into your new directory. The file name has a period in front of it – this is not a typo. To view the file in your directory, use the command `ls -a`.
3. Create the following subdirectories in your work directory: **work**, **src**, **db**, and **reports**.
4. Copy the following files from my RTL Compiler website into your **src** directory:
  - a. `osu05_stdcells.v`
  - b. `halfadder.v`
  - c. `fulladder.v`
  - d. `fulladder_tb.v`
5. Copy the following file from my RTL Compiler website into your **db** directory:
  - a. `osu05_stdcells.db`

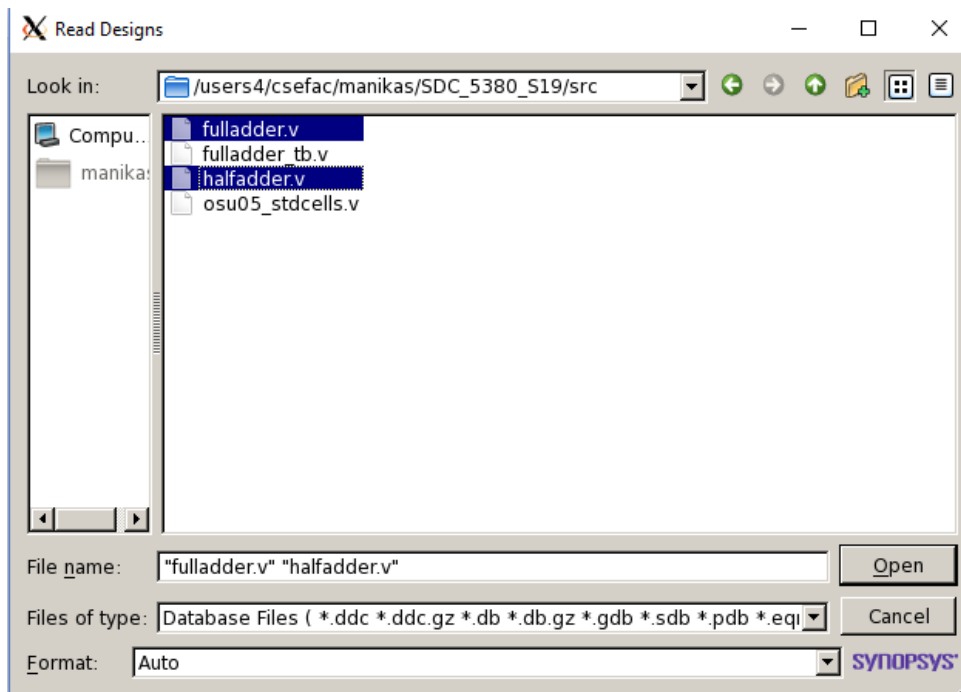
### Part III: STARTING THE DESIGN COMPILER

#### Ignore Step 1 as this is specific to GWU

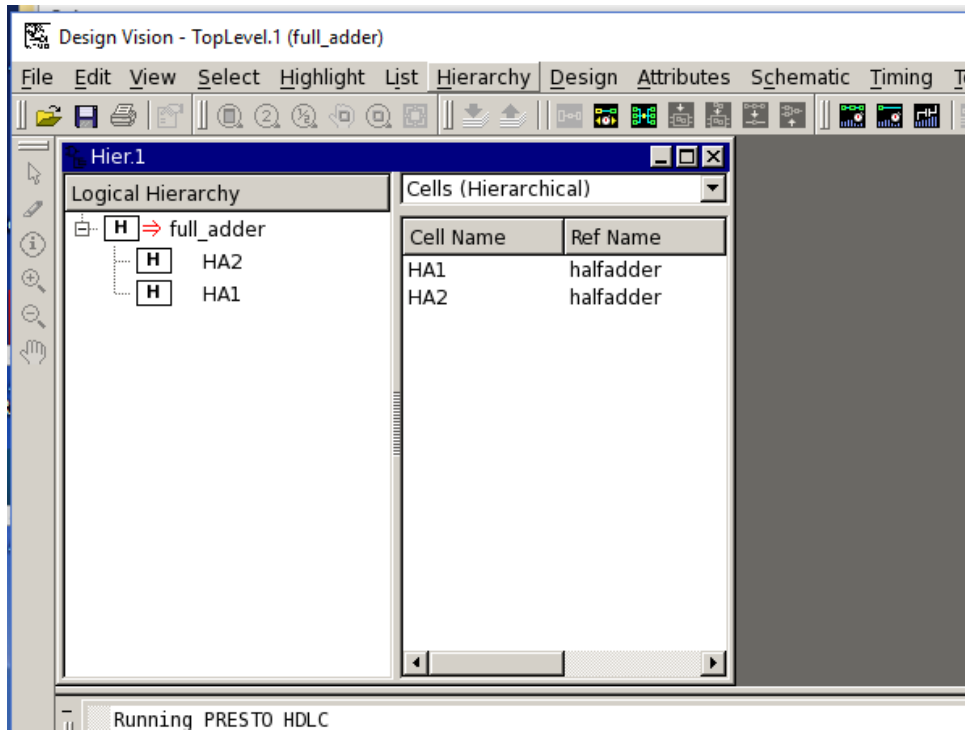
For Step 2, make sure that you are in your main working directory that you created for Step 1 of Part II.

**Skip Steps 5 and 6 of the tutorial (File->Analyze, File->Elaborate).** Instead, do the following to load your Verilog files “fulladder.v” and “halfadder.v” into Design Compiler:

1. Select **File->Read**
2. In the “Read Designs” pop-up window, double-click on the **src** directory
3. In the **src** directory, first select the file “fulladder.v”, then select the file “halfadder.v”. Note that the top file in the hierarchy (fulladder.v) should be the *first* file selected, as shown in the figure below:



4. Click on the Open button. You should get the following hierarchy window in Design Compiler:



#### Part IV: VIEWING THE SCHEMATIC – SKIP THIS PART

- **NOTE: DO NOT OPEN A SCHEMATIC UNTIL YOU REACH PART VI OF THE TUTORIAL (INSPECTING YOUR RESULTS)**
- **If you attempt to do the items in Part V with an open schematic, Design Compiler will crash.**

#### Part V: APPLYING CONSTRAINTS

Follow all the steps in this section of the GWU tutorial, except as noted below.

For Step 3, follow part (a) since the full adder does not have a clock.

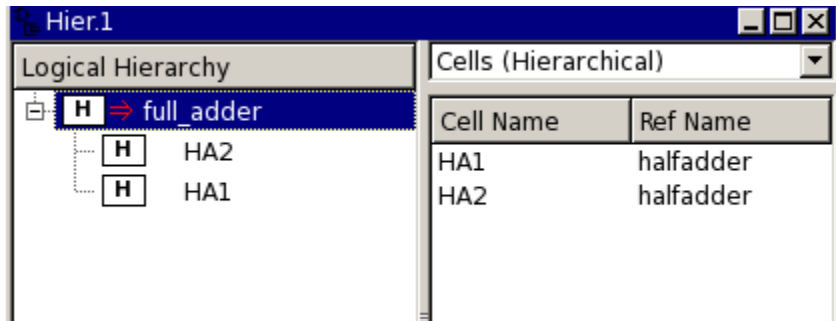
[NOTE: if you have a design with a clock (e.g., sequential logic), then you would follow part (b)]

Continue with part (c) for Step 3, and the remaining steps of the GWU tutorial.

## Part VI: INSPECTING YOUR RESULTS

Before doing Step 1 in the GWU tutorial, do the following setup steps to allow you to view the schematic:

1. From the menu, choose **Hierarchy->New Logical Hierarchy View**
2. In the Hierarchy window, click on the top level of the logic hierarchy (full\_adder):



Then follow the instructions in Step 1 for viewing the schematic of the full adder.

Follow all the remaining steps in this section of the GWU tutorial.

## PART VII: RE-SIMULATING THE SYNTHESIZED DESIGN

After the design has been synthesized, you will want to check the synthesized Verilog code using your testbench.

1. From the menu, choose: **File->Save as**
  - a. Navigate to your **src** subfolder
  - b. Enter the name **fulladder\_syn.v**
  - c. **Note: do not call the new file the same name as your original Verilog file (fulladder.v). If you do, it will overwrite your original file.**
2. Exit Design\_Vision.
3. At the Unix prompt, change to your **src** subdirectory and run Xcelium on your synthesized Verilog code: **xmverilog osu05\_stdcells.v fulladder\_tb.v fulladder\_syn.v**

Recall that we introduced delays during synthesis, so you will see delays on your outputs in Xcelium.