

Lab 2

Due 02/13/2013 1:30 pm

Synthesis Using Synopsys Design Compiler

Log on to a LINUX machine using your account.

1. Make a directory "5470_tutorial" at your choice of location and enter it.
 - a. `mkdir 5470_tutorial`
 - b. `cd 5470_tutorial`
2. Download the example in the "5470_tutorial" directory that you just created.
3. Use the following commands to uncompress lab2.tgz
 - a. `gunzip lab2.tgz`
 - b. `tar -xvf lab2.tar`
 - c. `cd lab2`
4. You will notice the following directory getting created that contains the input design data and the design constraints: **s35932/** - This is the test example for a sequential benchmark with over 10,000 gates. The benchmark contains the design input data needed for running synthesis and place & route flow for the design. Note: These files must be present for any design that is needed to be synthesized and placed & routed.
5. Below is a brief description of the file related to the s35932 benchmark inside the directory:
 - a. **s35932.v** - This is the RTL functional description of the design that we intend to synthesize and place & route.
6. There are also script files located at one level above the s32932 directory. These include:
 - a. **env.tcl** - this file sets up the different path variables that will be used later on to read the design data (e.g., location of design files, gate libraries). You can read the explanation of each "TCL variable" inside the file itself.
 - b. **dc.tcl** - this is the script that contains the "TCL commands" for doing the synthesis using Synopsys Design Compiler.
7. We also need some standard-cell library files which are required during synthesis and for standard-cell placement, routing and timing analysis. For this library, we need the following files:
 - a. **your_library.lib** - this is the timing (technology) library containing delay/power characteristics of the gates (cells) used for mapping in synthesis. The gate-level Verilog netlist will be mapped using the cells specified in this library. For the synthesis purposes we have already converted this file into another format which is required by Synopsys (**your_library.db**) which we will also read along with the input design. For this course we will be using a Commercial Standard Cell 90nm library obtained through Faraday's academic program.
8. We covered '**How to run Synopsys Design Compiler**' in the Lab1. Here are a few additional steps you can follow.
 - a. To start the Synopsys Design Compiler type: **dc_shell**
 - b. If you wish to use the graphical interface "Design Vision" of Synopsys Design Compiler, you can type "**gui_start**" at the dc_shell prompt.
 - c. To see details about a particular dc_shell command (e.g. compile), you can use the following two techniques:
 - c.i. Type "**man compile**" at the dc_shell prompt.

- c.ii. Use the Help->Man Pages->Commands feature from Design Vision and check details on the particular command.
9. To synthesize the design, type the following command at the dc shell prompt: ***source dc.tcl***. This runs the entire synthesis flow. Alternately, **you can also run each command in dc.tcl one at a time to understand the flow better (recommended)**.
10. We provide a timing constraint to the tool (in the form of clock period) which is given as a command inside the 'dc.tcl' script. Synthesis is done to ensure this timing constraint is satisfied and total area is minimized.
11. Try reading the INFO messages being printed on the terminal output by the tool during the synthesis procedure. You will find that there are three phases involved in the compilation stage and there will be multiple iterations at each of these sub-phases to improve the quality of the design. These three sub-phases are given below:
 - a. **Beginning Mapping Optimizations:** Notice that the tool tries to improve the worst delay seen after initial mapping. As visible in the printed table, WORST NEG SLACK [the most negative slack] decreases to 0 after multiple iterations.
 - b. **Beginning Delay Optimization Phase:** Since the timing constraints have been already met (worst negative slack is zero), this does not impact the results much.
 - c. **Beginning Area Recovery Phase:** Here the synthesis tool tries to meet/optimize the area constraints specified in the input file. As visible, the area of the design decreases with no negative impact on timing.
12. In the end, the RTL Verilog netlist ".s35932/s35932.v" is synthesized using the library and saved in the file "s35932_netlist_synopsys.v". This file is located at the \$designDir (i.e. ./s35932/) as specified in 'env.tcl'. You can also find the other reports written to the \$outDir(i.e. ./outputDir/). Some of the important reports are:
 - a. area_report.txt - Specify the combinational area of the synthesized netlist.
 - b. timing_report.txt - Specify the timing of the design obtained after doing the synthesis to a gate-level netlist.
 - c. latch_report.txt - Reports if any latches were inferred during synthesis.
 - d. power_report.txt – Reports the dynamic and leakage power.

Read through the following commands used for synthesis in the script file:

The dc.tcl is the file which has all the commands needed for synthesis of the design.

1. ***source env.tcl*** : This command sources the env.tcl file. Thus the commands in this file will be executed. The **env.tcl file** sets up the location parameters (which basically are tcl variables). The location parameters to be set are designDir, libDir, outDir. In short, it specifies the location of the input design data, input libraries required, timing constraints imposed and the directory to dump the intermediate results. Commands in env.tcl are as follows:
 - a. ***set designDir ./s35932*** : Set the following parameter to point to the location where the design data is located. Design data includes gate level netlist (.v), design floorplan (.fp and .fp.spr), design configuration file (.conf file), timing constraint file (.sdc).
 - b. ***set libDir /opt/software/cadence/library***: This command sets the path to where the library is located.
 - c. ***set outDir ./outputDir*** : Set the following variable to point to the directory where you want to store the intermediate results. The intermediate results include the design saved at various stages of the flow. e.g. You may want to save the design after placement or routing or clock tree synthesis or in between while doing timing optimizations.

2. ***set target_library "\$your_library.db"*** : This is the command to load the timing library (.db). The lib library is the one with the logic and timing for each cell. In the provided example, the library is mentioned in the '.db' format which is the internal format used by Design Compiler. Though we have already converted cell library to the required format for you, any industry standard liberty format of the technology library (i.e. .lib format) can be converted to corresponding '.db' format using following steps:
 - a. `dc_shell> read_library <.lib file>`
 - b. `dc_shell> write_library`
3. ***set_wire_load_model -name G50K*** : Here we are specifying which wire-load model to use for the capacitance estimation of the wires/interconnects. This is very important to understand the concept of the wire-load models. During synthesis, we do not have exact estimates of the wire-capacitance as the design is not placed/routed. Wireload models provide the estimates of the capacitance of the interconnects based on the number of the fan-outs. You will find that later when we actually place and route the design, we will run into timing problems as the estimates of the wire-load models were inaccurate and the synthesis engine used those inaccurate estimates of wire capacitance to optimize the design timing.
4. ***analyze -format verilog "\$designDir/s35932.v"***: Analyzes the verilog file s35932.v. Used for initial syntax and semantic checking. Also making sure that only synthesizable verilog constructs have been used in the input design. For Example, most of the synthesis tools cannot synthesize unbounded loops.
5. ***elaborate s35932_bench*** : This command is used to elaborate the design. It builds a design from the intermediate format of a Verilog module, a VHDL entity and architecture, or a VHDL configuration. In a multimodule design, the argument of the elaborate command is the top level module.
6. ***set_max_area 55000***: Setting the area constraint that synthesis tool must target to meet. Note that by default timing constraints have priority over the area constraint. So Synthesis tools will first try to optimize the design for meeting the timing constraints and later try to improve the area consumption under the timing constraints.
7. ***create_clock blif_clk_net -period 0.6***: This is used to create a clock for the design with a period of 0.6ns. Here are actually specifying the clock frequency at which we want our design to run after fabrication. You can play with setting different clock periods and running the synthesis tool. If you set very tight constraints, you will find that tool takes a lot of time to optimize the design to meet those constraints.
8. ***set_input_delay 0 -max -clock blif_clk_net [all_inputs]***: This is used to set the delay at input ports as 0 relative to the clock signal.
9. ***set_output_delay 0 -max -clock blif_clk_net [all_outputs]***: This is used to set the delay at output ports as 0 relative to the clock signal.
10. ***check_design > \$outDir/check_design.txt***: This command checks the current design for consistency. It stores the output in a file called check_design.txt in the outDir folder.
11. ***check_timing > \$outDir/check_timing.txt***: This command is used to check for possible timing problems in the current design. It stores the output in a file called check_timing.txt in the outDir folder.
12. ***uniquify***: This command is used to remove multiply-instantiated hierarchy in the current design by creating a unique design for each cell instance.
13. ***compile -map_effort medium -ungroup_all***: This is used to compile and map the design. This is the command where the synthesis tool actually do the technology mapping using the cell library that we have specified and try to meet the design constraints of area and timing.
14. ***write -format verilog -hierarchy -output \$designDir/s35932_netlist_synopsys.v***: The write command is used to generate the structural code or netlist. The netlist report generated by the write command is written to the verilog file s35932_post_syn.v in the designDir directory.

15. ***report_area*** > ***\$outDir/area_report.txt***: This reports the summary of the area of each component of the current design and exports the result to the *area_report.txt* file in the *outDir* directory.
16. ***report_power*** > ***\$outDir/power_report.txt***: This reports the power consumed by the design and exports the result to the *power_report.txt* file in the *outDir* directory.
17. ***report_constraint -all_violators*** > ***\$outDir/violator_report.txt***: This is used to displays constraint-related information about a design and the result is stored in the *violator_report.txt* in the *outDir* directory.
18. ***report_timing*** > ***\$outDir/timing_report.txt*** : It displays timing information about a design. It exports the result to the *timing_report.txt* file in the *outDir* directory.
19. Finally, to exit from the SDC compiler type the following command at the *dc_shell* prompt: ***exit***

Turn in all the reports generated and submit through Canvas.