Lab 7: Logic Synthesis, Static Timing Analysis, and Automatic Place and Route

ECEN 454-503

Alexia Perez, 127008512

**Purpose:** This lab serves to introduce students to design vision, and its capabilities with a Verilog module. Design vision can take a Verilog module and translate it into a netlist, in which a clock signal and delays can be assigned, and maximum and minimum paths through the module can be calculated. It also introduces students to synthesized netlists and standard cell libraries to place and route circuits on a die in a program called *innovus*.

**Procedure:**

***Part A:***

1. Open design vision and adjust the settings to source the correct libraries. Analyze and elaborate the cruisecontrol.v module.
2. Save the design and the attribute settings.
3. Create a symbol view and set the drive strength on all the input ports to 0.0335. Set the load on all the output ports to 3.
4. Set the operating conditions as ‘typical’ within the library specified in step 1.
5. Save the design and attribute settings.
6. Set the clock constraints to have a period of 25 and a pulse width of 12.5. Set the output delay to be 5 for max rise and min rise.
7. Check the design and use the uniquify command if needed.
8. Save the design as “ccs\_attributes\_b4\_compile.db”
9. Compile the design and set the map effort to medium. Check the “allow boundary conditions.”
10. Create the report constraints and report area files.
11. Save the optimized netlist and report the register count.

***Part B:***

1. Close design vision and open primetime for static timing analysis. Set up the search and link path and load the design. Set the capacitance for the output ports, and the driving strength for the input ports.
2. Create the clock with a period of 10 and a pulse width of 5. Use the check\_timing command to show possible timing constraints and set the max and min input delays to 4 and 0, respectively. Set the output delay to 5.
3. Generate the path-based timing reports for the max and min paths of the module.

***Part C:***

1. Make a directory in the root directory for this lab. Copy the synthesized netlist created in the previous lab and place it in this directory.
2. Download the appropriate files from the lab website and add them to the directory.
3. Open innovus.conf and modify it to include the top level module for the cruise control circuit.
4. Launch innovus using the specified command.
5. Import the design to innovus. Then, specify the floorplan to have an aspect ratio of 0.9 and a core utilization of 0.7. Specify the core margins to be 30 for all.
6. Place the power rails around the die and specify metal3 for the top and bottom layers and metal2 for the left and right layers. Modify all metal width to 10 and change the offset to center in channel.
7. Add power stripes with a width of 5, and a first/last shape of 70.
8. Route the power grid entirely through the die. Next, place the standard cells. Note the number of standard cells used in your design.
9. Route the required wires with the timing driver concurrent routing feature.
10. Print out a copy of the layout generated.

**Results:**

***Part A:***

1. The area report and constraints report are displayed below. The combinational area is reported to be 8543, the non-combinational to be 1920, and the total cell area to be 10463. The design also has no violated constraints.

Area:

Table

Description automatically generated

Constraints:

Text

Description automatically generated

1. Below is the synthesized netlist for the “cruisecontrol” module. It lists every gate (and, flip flop, not, etc.) or intermediate net needed to create this module as a circuit.

Text

Description automatically generatedText

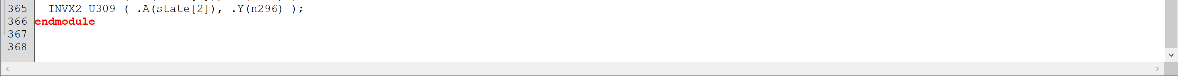
Description automatically generated

Table

Description automatically generated with medium confidenceText

Description automatically generated

Text

Description automatically generated with medium confidenceGraphical user interface, text

Description automatically generated with medium confidenceGraphical user interface, text

Description automatically generated

Within the synthesized net list above, there are no DFFSR registers, but there are 4 DFFPOSX1 registers (the state register, speed register, cruisespeed register, and cruisectrl register).

***Part B:***

1. Below is the max paths report for a slack lesser than 5. This shows the least possible slack lesser than 5 considering only the setup paths. According to the report, the lowest slack achievable is 3.79.

A picture containing graphical user interface

Description automatically generated

Chart

Description automatically generated with low confidence

1. Below is the min paths report for a slack lesser than 3. This shows the least possible slack lesser than 3 considering only the hold time paths. According to the report, the lowest slack achievable is 0.13.

A picture containing chart

Description automatically generated

A picture containing graphical user interface

Description automatically generated

***Part C:***

1. For this lab, the total wire length was 9027 µm, and the number of vias was 1436. The screenshot of this information is shown below:

Text

Description automatically generated

The number of standard cells reported was 264. The screenshot of this information is shown below:

Graphical user interface, text

Description automatically generated with medium confidence

1. Lastly, the full layout generated by the program is displayed below.

A picture containing text, display

Description automatically generated

**Conclusion:** This lab helped me better understand how to use Design Vision, the graphical interface to the Synopsys family of logic synthesis tools. I became familiar with the basics of synthesis using Design Vision through the simple “cruise control” design example. After completing the synthesis portion, I performed pre-layout static timing analysis of my synthesized design, and the defined constraints for it and check the timing of all the paths in it. In the final part of this lab, I learned how to use the synthesized Verilog netlist that I previously generated to "Place and Route" the cruise control logic circuit on a die.