ECE 551 Homework #5

Due: Nov. 30th @ 11:00 am

This homework assignment is to be completed in groups. Work with the members of your project group and submit one copy with each of your names on it.

Please Note: To receive full credit, you should use the following best practices in your homework assignments**:**

- Follow the coding style guidelines outlined in the course and previous homework
- All figures & graphs should include a number and caption (handwritten is OK).
- When printing waveforms, ensure that the relevant signal transitions are clearly visible, and label them if necessary.
- *All code should be typed*. Please do not submit handwritten code.
- **You must always turn in a printout of all Verilog code used with each problem unless explicitly told not to do so.**

[1] The Wild World of Synthesis - *(30pts)*

(a) (*4pts)* In this problem you will modify the parameterized adder you designed in HW 4, Problem 2 (select one module among your group members).

Change the default value of BITWIDTH to 168. Add a clock signal to the module and **register both the inputs** (a, b, c_in) **and the outputs** (sum, c_out). This means that all inputs to the top-level module should be loaded directly into flip-flop inputs and all outputs should come directly from flip-flop outputs. (You can infer your flip-flops using a clock-triggered always block; you don't need to use structural flip-flops.) There should be no combinational logic between your input ports and input registers and between your output ports and output registers.

Submit your updated module code.

(b) *(6pts)* Write a synthesis script that does the following:

- Begins by running the command: **remove_design –designs**
- Reads and elaborates the files you need to synthesize your module from part (a)
- Creates a 200 MHz clock and sets it as a don't touch network
- Sets the area constraint to **0.**
- **Operating Environment**: Sets drive cell to DDRVLS33, operating conditions to NOM, wire load to B1X1, input and output delays to 0.
- Compiles the design using low area effort (use **compile,** not **compile_ultra)**
- Prints the Area and Timing reports

You can use the sample script on the course website as a reference on the commands you need to use to achieve these requirements. If you need additional help on the command syntax, try performing the action in the Design Vision GUI, following to the instructions in the tutorial. When you perform an action in the GUI, the corresponding script command is echo to the Log.

Submit your script.

(c) *(16pts)* Using the script from part (b), fill in the columns of Table 1 with the area results from your area report. You should list the Combinational ("comb"), Non-Combinational ("ff"), and Total areas in the table. (The other important component of area is the Net Interconnect area)

You will need to change your clock period each time you run the script to match the clock frequencies in the table.

- For the "low" column use the script as described in (b).
- For the "uniquify" column, add the **uniquify** command before compiling.
- For the "ungroup" column, add the **ungroup -all** command before compiling (do not include uniquify).
- For the "ultra" column, use **compile_ultra** with default options instead of compile (do not include ungroup and uniquify; compile-ultra ungroups automatically if needed).

If the synthesis fails to meet timing constraints (negative slack) at a given frequency, put "FAILED" in the corresponding area cells.

Clock (MHz)	**low**			**uniquify**		
	comb	**ff**	**total**	**comb**	**ff**	**total**
50						
100						
150						
200						
250						
Clock (MHz)	**ungroup**			**ultra**		
	comb	**ff**	**total**	**comb**	**ff**	**total**
50						
100						
150						
200						
250						

You do not need to submit the area & timing reports themselves.

(d) *(4pts)* Briefly comment on:

- How does the area taken up by the registers compare to the adder logic?
- How did the various synthesis options affect your area results?
- Did you notice any trends as the timing requirement got tighter?
- Did you notice a difference in how long it took for synthesis to finish?

One thing to note: Just because a specific option worked well or poorly with your adder doesn't mean that option always works well or poorly. Certain types of designs are more likely to benefit from each of the options.

[2] Milestone 1.5 - *(42pts)*

Note: Although this problem is designed to help you stay on track with your Project, it will still count towards your Homework grade.

(a) *(20pts)* Synthesize your TX module from the project. If your TX module has changed since Milestone 1, upload a copy of the updated version to the HW5 drop box. You do not need to turn in a printout of the TX code.

Write a self-checking testbench similar to the one you created in Homework 4 that compares the output of the pre-synthesis version of your TX with the post-synthesis netlist. (You can probably just modify the testbench you were using for Project Milestone 1).

Create a separate error signal for each output of the TX and an overall error signal called "any_error" that is a logical OR of your other error signals.

If you find any errors between the pre-synthesis and post-synthesis versions, go back and modify your pre-synthesis code until the post-synthesis results match the pre-synthesis results (other than the normal delays exhibited by synthesized logic) *and* the post-synthesis version passes all of your tests.

Submit your self-checking testbench (on paper) and print a wave window showing your pre-synthesis output compared to your post-synthesis output (if necessary, include more than one wave window to make sure the all output signals and error signals are clearly visible). If your error signals still go high in the fixed module, include an explanation of why this happens.

(b) *(20pts)* Repeat part (a) for the RX module.

(c) *(2pts)* If you needed to make any changes to your pre-synthesis module to eliminate performance errors between the pre-synthesis and post-synthesis versions, summarize them. If your post-synthesis version worked properly the first time you tried it, say "No problems detected".