Project 3: Synthesis with Design Compiler

Objectives: Use the Design Compiler (DC) to synthesize your design and generate netlist Verilog code.

Software setup:

1. You will need to connect to the RedHat system remotely: walle.ece.pdx.edu. If you are on campus, you can remote access through Ubuntu PC in the Intel lab. If you are off campus, check: https://cat.pdx.edu/services/network/vpn-services/ to install PSU VPN on your system. Then install a remote access software support x-server such as MobaXterm or putty with xming.

2. Remote Login to walle.ece.pdx.edu with your ECE username and password.

3. Read the DC_Tutorial to set up DC and DC_User_guide for common commands. Please read the tutorial first and install DC in your UNIX account. If you have any question about the installation process, please email your problems to the CAT team: support@cat.pdx.edu.

Note: choose osu05_stdcells.db as your target library, generate the netlist file in Verilog.

Task 1:

a) Design a 4-bit grey code adder.
b) The adder has three components: two 4-bit grey-to-binary converters, a 4-bit binary adder, and a 5-bit binary-to-grey code convertor.
c) Model this design with SV as a combinational block.
d) Write one test bench to verify the SV model.
e) Synthesize the design and show the complexity (gate counts), area, and power of synthesized circuits using report command from DC. (Choose osu05_stdcells.db as your target library, generate the netlist file in Verilog)
f) Go back to your simulator, and simulate the synthesized Verilog netlist by including std05_stdcell.v as the standard cell library.
g) By simulation, verify the SV model and the synthesized Verilog netlist with the same testbench.

Task 2:

a) Design a finite state machine (FSM) with at least five states. You decide the inputs and outputs. Draw the state diagram of your FSM. Make sure that your design be different from those used by other students. If your design is identical to some design used by other students, further investigation will be conducted and you will be asked to revise your design and redo the work.
b) Model your FSM with SV using regular case statements.
c) Model your FSM with SV using reversed case statements.
d) Write one test bench to verify both models.
e) Synthesize both models and compare the complexity (gate counts), area, and power of synthesized circuits using report command from DC.
f) By simulation, verify the SV model and the synthesized Verilog netlist with the same testbench.

Task 3:

a) Synthesize your simulation model for the Traffic Light Controller in Project 2.
b) Use your testbench to verify if your simulation model is equivalent to your synthesized circuit.