Experiment 3: Introduction to ASIC Synthesis

Objective

- To familiarize with ASIC synthesis tools.
- To interpret area, power and timing reports.
- To control the synthesis outputs through timing constraints

Background:

In the ASIC design flow, the Verilog register transfer level (RTL) code captures the ASIC's architecture and functionality. ASIC synthesis is a process which maps the RTL code to a gate level netlist (GLN) built from *standard cell libraries* of a logic family (eg. 180 nm CMOS). Since standard cell libraries are real circuit implementations of logic gates, the synthesis tool can estimate hardware metrics of the architecture such as area, power and timing from the GLN. A synthesis tool takes in three inputs: the RTL code, library file, and a constraint file and generates the GLN, area, power and timing reports.

Tools/Library used

- Synopsys Design Compiler (DC)
- 180 nm CMOS library from Semi-Conductor Laboratory (SCL)

Experimental Procedure:

1. Run the sample script provided. Modify the script for the following designs and generate the synthesis reports and outputs.

    a. 4-bit Fibonacci sequence generator
    b. 4-bit Fibonacci sequence generator with ripple carry adder

2. Identify the critical path of the design from the timing report and verify whether it is the expected path. Take a note of the netlist, area, power and the slack obtained in the timing report.

3. Reduce the clock timing constraint in steps (eg. reduce by 0.1 to 1ns) and rerun the synthesis tool. Note down the new netlist, area and power results and plot their variation with respect to the timing constraint. Stop the process once slack becomes negative. What is the maximum achievable frequency of this design for the given library?

4. Perform a gate level simulation to verify the functionality of the synthesized netlist.

5. Repeat steps 1-4 for all the designs. Also generate the results for a 32-bit Fibonacci sequence generator. Update the timing constraints accordingly.