ECE 429 Laboratory 5
Hierarchical Design and Formal Verification

Announcements:

1. **Teaching Assistants**
   Lab-1: Yunlong Zhang (yzhan167@hawk.iit.edu), Office Hours: 11:30 AM. - 1:30 PM. at SH309 on Mondays
   Lab-2: Shuai Li (sli97@hawk.iit.edu), Office Hours: 11:30 - 1:30 PM. at SH309 on Fridays

2. **Time Frame for Lab-05**

- In-Class students
  - ECE-429-01 (every Monday 1:50 - 4:30 PM.) -10/14/2015
  - ECE-429-02 (every Wednesday 6:25 - 9:05 PM.) -10/07/2015
  - ECE-429-03 (every Friday 1:50 - 4:30 PM.) -10/09/2015
- Internet Students
  - Every Monday (Report due: every Monday at 4:30 PM.)

3. **Account Administrator**
   If you have account issues or remote access problems, please contact Mr. Upendra Gandhi (support@ece.iit.edu).

1. Introduction

In this lab assignment you will be introduced to hierarchical design and formal verification techniques that are essential for complex circuit designs. We will continue to practice layout for the 2-input NAND gate we designed in Lab 4. Then we will build a complete design of a 2-input AND gate from previous designs including both schematic and layout. The overall design flow is summarized as follows.

- Schematic and layout design for individual cells (NAND gate and inverter)
  - Layout verification via LVS
  - Timing and power estimation via schematic and post-layout simulations
- Hierarchical schematic and layout design for the circuit (AND gate)
  - Layout verification via LVS
  - Schematic verification via equivalence checking

- Timing and power estimation via either schematic and post-layout simulations or estimations from individual cells

2. Formal Verification and Equivalence Checking

Due to the extremely high cost of chip fabrication, it is a must nowadays to verify that a chip will work correctly before sending the design out for tapeout. Functional verify-cation is one of the many important verification tasks that validates the logic functionality of the design, e.g. an AND gate design should output ’1’ only if both inputs are ’1’. While one can always simulate the design with different input patterns in order to validate the outputs against correct values, he/she has to face the dilemma when there are more than tens of inputs - it is way too time-consuming to simulate all possible input patterns, but without simulate all of them, there is no guarantee to catch all errors if there is any.

To resolve the dilemma, formal verification techniques have been proposed that can provide a proof of functional correctness against a full specification or a group of desired properties using formal methods of mathematics. The by-product when the correctness is disproved could be further used to identify the errors in the system. While active researches are still needed for many formal verification techniques, some of them are mature enough so that commercial tools are readily available and are adopted by most, if not all, leading hardware design companies.

You will be introduced to one such formal verification techniques, called equivalence checking, in this lab. As its name suggests, equivalence checking proves or disproves that the two designs of interests exhibit the same functionality. It makes sense for hardware designs since most hardware design flows involve designs at various abstraction levels that should have the same functionality. Moreover, it helps to detect any error during manual or automated design transformations that target only at better performance without incurring any functional change.

In some sense, LVS is one example of equivalence checking - it shows where the layout is exactly the same as the schematic, though some other factors including transistor sizes are also verified. As you may already notice, LVS doesn’t guarantee any kind of correctness for your schematic and when the complexity of your schematic increases, it will become more difficult to make it right. Therefore, we need to specify designs at a higher abstraction level to verify the correctness of the schematics.

Since all our designs are digital designs in this course, we can specify the functionality in boolean logic in addition to the transistor-level schematic. For example, the following Verilog code specifies a 2-input AND gate called ‘and2’ that has two inputs ’a’ and ’b’ and one output ’f ’.

Don’t worry if you don’t have any previous experience with Verilog since we will only use the code as it is to verify the schematic and no modification is required. By

using the Formality ESP tool, we will be able to verify a schematic as a SPICE netlist against a Verilog model as mentioned above.

```
module and2(a,b,f);
    input a,b;
    output f;
    assign f=a&b;
endmodule
```

3. Preparing for the Laboratory (Pre-Lab)

Please make sure you have the following prepared before you come to lab.

- Sketch a stick diagram for a static CMOS 2-input NAND gate.
- Sketch a 2-input AND gate design using a NAND gate and an inverter.
- Familiarize yourself with Tutorial III: Hierarchical Design and Formal Verification.

4. Lab Instructions

Make sure the transistors (PMOS and NMOS) in your 2-input NAND gate design of Lab 4 have the minimum width (90nm) and length (50nm). Create the layout of the NAND gate in Virtuoso in the same cell and verify the layout using LVS.

Follow Tutorial III to create a complete design of a 2-input AND gate including schematic and layout. Verify your AND gate layout using LVS against its schematic and verify the AND gate schematic using equivalence checking against the Verilog model.

Perform schematic simulations using the three excitation settings from Lab 4 and a load capacitance of 1fF attached to the output of the AND gate. Compute the rising propagation delay and the falling propagation delay of your AND gate design.

5. Deliverables

The requirement of the lab/project reports with a template can be found on the Blackboard. *NEVER share your writings/screenshots with others.* We prefer to receive reports electronically as either .pdf or .doc files through the Assignments section on the Blackborad.

Follow the template for lab/project reports posted on the Blackboard to structure your lab report. Briefly describe the tasks you have accomplished. Report your measurements of propagation delays and the corresponding input transitions. Discuss your lab by

addressing at least the following questions.

- Is there any other way to design an AND gate than combining an inverter with a NAND gate? Which one will you prefer?
- Do the input transitions leading to the propagation delays match your expectations? Why or why not?
- Is there any relationship between the propagation delays you measured in this lab for the AND gate and those you measured in Lab 2 for the inverter and in Lab 4 for the NAND gate?
- What is the difference between a schematic and a Verilog model?

Finally, attach screenshots of your NAND gate layout and LVS report, AND gate schematic and equivalence checking report, and AND gate layout and LVS report.