ECE 429 Laboratory 8
Carry-Ripple Addition III

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-08**

- In-Class students
  - ECE-429-01 (every Monday 1:50 - 4:30 PM.) -11/02/2015
  - ECE-429-02 (every Wednesday 6:25 - 9:05 PM.) -11/04/2015
  - ECE-429-03 (every Friday 1:50 - 4:30 PM.) -10/30/2015
- 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, you will finish the design of a 4-bit carry-ripple adder using the full adder built in Lab 6 and 7.

2. Preparing for the Laboratory (Pre-Lab)

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

- Review the carry-ripple architecture as discussed in Lab 6.
- Translate the following two sets of inputs to binary
  - 11+5
  - 4+10

3. Lab Instructions

Create a Verilog file 'adder4.v' using the following template for the 4-bit carry-ripple adder design. You may reuse the Verilog code for the full adder in Lab 6.

```
module adder_4bit(A, B, S, CO);
    input [3:0] A, B;
    output [3:0] S;
    output CO;
    // Fill in your code here
endmodule
```

Your code can be a simple "assign" statement as follows:

```
assign {CO, S} = {1'b0, A} + {1'b0, B};
```

Of course, you can come up with different Verilog code, e.g. you can cascade four full adders to build a 4-bit adder.

Create a stimulus file 'adder4 test.v' using the following template to validate your adder's functionality by computing 11 + 5 and 4 + 10. You need to convert the decimal numbers into binary numbers. For example, "A = 4'b1011" represents "A=11".

```
module stimulus;
    // Declare variables for stimulating input
    reg [3:0] A, B;
    // All outputs declared with wire
    wire [3:0] S;
    wire CO;
    // Instantiate the design block counter
    adder_4bit a(A, B, S, CO);
    // Stimulate the Input Signals
    initial
    begin
        // Compute 0+0
        #10 A=4'b0000; B=4'b0000;
        #5  $display("%d+%d=%d %b", A, B, S, CO);
        // Add your test vectors here to compute and display
        // 11+5 and 4+10 plus 3 more additions of your choice
    end
endmodule
```

Simulate your Verilog files as follows and validate the correctness of the results:

```
username@host:~%  source /import/scripts/ece429.cshrc
username@host:~%  source /import/scripts/synopsys2012.cshrc
username@host:~%  verilog  adder4  test.v  adder4.v
```

Create the schematic and the layout of a 4-bit carry-ripple adder hierarchically from your full adder design built in Lab 6 and 7. Verify the schematic using equivalence checking and the layout using LVS. Here are a few hints:

- The 4-bit carry-ripple adder should take A and B as inputs and output S and C [4].
- C [0], i.e. the ci of the first full adder, should be connected to GND.
- To simplify equivalence checking, name the inputs as A<0>, A<1>, A<2>, A<3>, B<0>, B<1>, B<2>, B<3>, and the outputs as S<0>, S<1>, S<2>, S<3>, CO in both the schematic and the layout.

Perform a post-layout simulation of your carry-ripple adder design after parasitic extraction. When you design your ".sp" file, you must make sure that the order of signals in the line which instances your "adder_4bit" must be consistent with the order of pins in "adder_4bit.pex.netlist". Otherwise, the connection is incorrect, and you will not have correct simulation results.

Use the excitation A + B=11 + 5 → 4 + 10. Measure the delays from A<0> to S<3> and CO. You need to think how to measure the delays, what is the trigger (TRIG) and what is the target (TARG), and write measure statements properly.

4. 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 delays and the propagation delay. Discuss your lab by addressing at least the following question.

- How is the functionality of your adder design validated/verified at various abstraction levels (i.e. Verilog, transistor schematic, layout)?

Finally, attach screenshots of your Verilog files and simulation results, the schematic and the layout of your 4-bit carry-ripple adder, and the LVS report and the equivalence checking report.