ECE 429 Laboratory 6
Carry-Ripple Addition I

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-06**

- In-Class students
  - ECE-429-01 (every Monday 1:50 - 4:30 PM.) -10/19/2015
  - ECE-429-02 (every Wednesday 6:25 - 9:05 PM.) -10/21/2015
  - ECE-429-03 (every Friday 1:50 - 4:30 PM.) -10/16/2015
- 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

We will spend the next three lab assignments to build a circuit that implements binary addition. We will first focus our effort on the schematic design of a circuit that performs one-bit addition, i.e. a full adder. Then will design a layout for the full adder and verify its correctness against its schematic. Finally, we will apply hierarchical design to build a 4-bit carry-ripple adder and verify it against a Verilog model.

The overall design flow is summarized as follows.

- Lab 6: schematic design of a full adder
- Lab 7: layout design of a full adder
- Lab 8: hierarchical schematic and layout design for a 4-bit carry-ripple adder

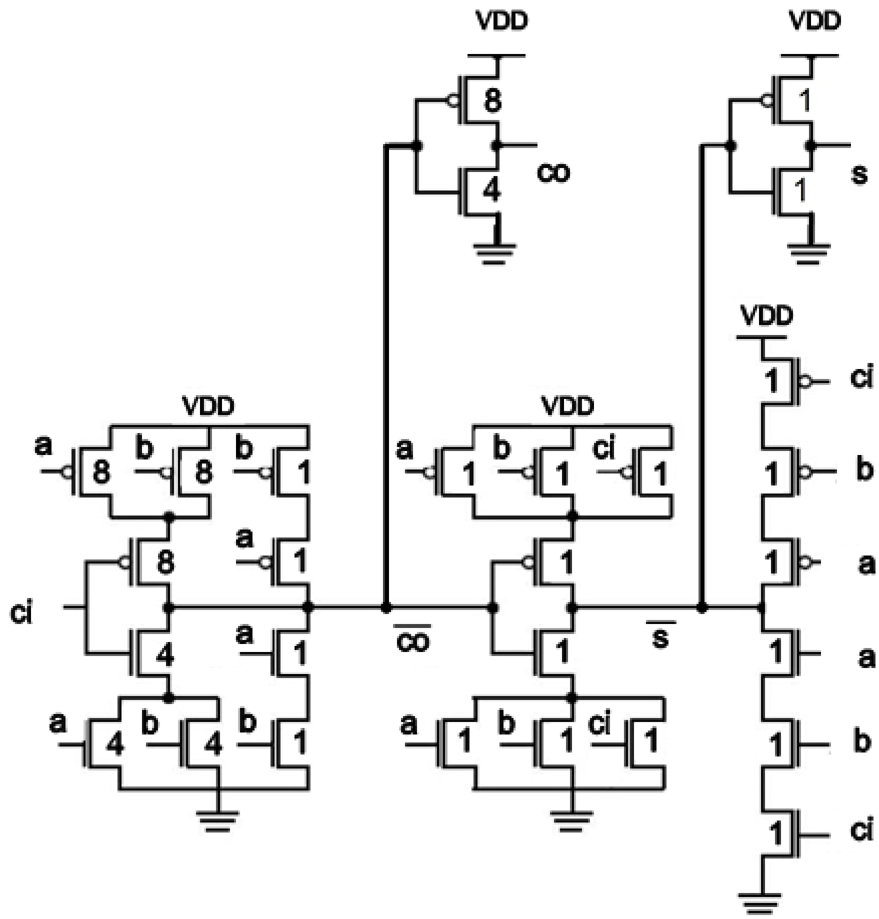

Figure 1: Schematic of Mirror Adder

2. Binary Addition and Full Adder Design

Addition of binary numbers is performed in a bit-by-bit approach. For two operands A and B, let their ith bits be A[i] and B[i] respectively. Note that we assume the least significant bit of an operand is its 0th bit. The summation S, whose bits are S[i], can be computed using a carry-ripple architecture as follows:

$$C[0] = 0,$$
$$S[i] = A[i] \text{ XOR } B[i] \text{ XOR } C[i], \ \forall i \geq 0,$$
$$C[i+1] = A[i]\,B[i] + A[i]\,C[i] + B[i]\,C[i], \ \forall i \geq 0,$$

where for all $i \geq 0$, C [i + 1] is the carry generated by the addition of the ith bits of the operands.

The above equations suggest that one can perform a binary addition for an arbitrary number of bits using a circuit that computes the summation and the carry for one bit at a

time. To be more specific, this circuit will compute the summation bit s and the carry output co from the bits a and b of the operands and the carry input ci generated from the addition of the previous bits. This circuit is called a full adder and the relationship between its input and output are defined as follows.

co = a b + a ci + b ci,
s = a XOR b XOR ci.

Several architectures have been proposed for the implementation of the full adder circuit. The primary target is to reduce the number of transistors required and also to increase the speed of the addition operation. One of the most successful designs (in static CMOS) is the mirror adder shown in Fig. 1 (also see Fig. 11.4 from the textbook). The mirror adder design exploits the following two sets of equations of s and c to construct the pull-up and pull-down networks that are symmetrical to each other.

pull-up network: co = (a + b) (ab + ci), s = (a + b + ci) (a b ci + co),
pull-down network: co = a b + (a + b) ci, s = a b ci + (a + b + ci) co.

Note that this symmetrical design also simplifies layout design since both nMOS and pMOS diffusions and all their internal connections should be the same except the heights of the diffusions (widths of transistors).

The following Verilog code can be used to verify the functionality of any full-adder design.

```
module
    adder(a,b,ci,s,co);
    input  a,b,ci;
    output  s,co;
    assign  s=a^b^ci;
    assign co=(a&b)|(b&ci)|(ci&a);
endmodule
```

3. Preparing for the Laboratory (Pre-Lab)

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

- Familiarize yourself with the mirror adder schematic. Try a few input patterns on the mirror adder and see how the pull-up and pull-down networks work.

4. Lab Instructions

Design the schematic of the full adder according to Figure 1 of the lab instruction. Note that the size of each transistor is provided in Figure 1. “1” means the minimum transistor size, W=90nm, L=50nm. “4”: W=360nm, L=50nm. “8”: W=720nm, L=50nm. You must set the transistor sizes correctly.

Create a test circuit and generate the SPICE netlist (lab6.sp) for simulation. Set the load capacitance of “co” and “s” to be 1fF.

Create the verliog file (adder.v) for the full adder using the code provided in section 2.

Do equivalence checking using Formality ESP (see Tutorial 3). In the first step (Reference), use “adder.v” as the design file. In the second step (Implementation), use “lab-6.sp” as the design file, and disselect the option "Show only possible top designs". Then you will be able to choose your full adder design.

Attach a load capacitance of 1fF to co and s. Compute the propagation delay from ci to co under a load by performing schematic simulations using the following two excitation settings:

- a b ci=010 → 011 and 011 → 010
- a b ci=100 → 101 and 101 → 100

Specifically, please use the following PULSE for “ci” in your simulation:

v3 ci 0 PULSE 0 1.1 0 10p 10p 490p 1n

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 addersing at least the following questions.

- It is known that the most timing-critical path in a full adder is from ci to co. For the circuit to compute co, why the transistors are sized differently and why ci is used as the inner inputs?

Finally, attach screenshots of your mirror adder schematic and equivalence checking report.

6. Bonus Work

You can obtain an additional 20% in the grade of this lab if you can build a model for the propagation delay from ci to co. Assuming there is a load capacitance of 1fF attached at s. Design a set of experiments to collect necessary data and fit them into a model similar to that in Chapter 4.4.5. You should discuss your approach, present the experimental results and the obtained model, and attach relevant screenshots to your lab report for grading.