Harris

Introduction to CMOS VLSI Design (E158)

Lab 2: ALU Design

In this lab, you will design a full adder and combine it with your gates from Lab 1 to form a 1-bit ALU.

1. Full Adder

The objective of this lab is to complete a design of a nontrivial circuit, a full adder. There are many ways to approach this problem. You may assemble your adder from various simpler logic gates or may design it as a single complex cell. Refer to your textbooks for further references on full adders. Warning: at least one of the figures in Principles of CMOS VLSI Design regarding adders is incorrect. Be certain you understand a design before blindly copying it!

Copy your lab1_xx library to lab2_xx for this lab.

Draw schematics of the adder in fulladder{sch}. Name the inputs *a*, *b*, and *c* and the outputs *s* and *cout* to match the fulladder{ic} provided. It is wise to simulate your schematic to verify it works before proceeding to the time-consuming layout process.

Draw a layout of your adder in fulladder{lay}. The layout should obey the same constraints as your gates from Lab 1:

- power and ground run horizontally in Metal 2 on a 80λ center-to-center spacing
- all transistors, wires, and well contacts fit between the power and ground lines
- all transistors should be within 100λ of a well contact
- avoid long routes in diffusion
- *a* and *b* appear on left side of layout in meta1
- *s* appears on right side of layout in metal1
- *c* appears near the bottom of a cell in metal1
- *cout* appears near the top of the cell directly above *c* in metal1
- metal 2 use is acceptable, but leave space for at least six horizontal metal2 lines to run over the top of the cell

Remember that you will have to connect *cout* of one adder to *c* of the next when you assemble a ripple carry adder next lab. Therefore, don't place any obstructions that would prevent the connections.

Simulate your adder layout and verify it with DRC, ERC, and NCC.

2. 1-bit ALU

Look at the alu{sch} 1-bit Arithmetic / Logic Unit (ALU) schematic provided. It defines a 1-bit ALU like that of Figure 4.17 of Computer Organization and Design, sans overflow detection. It is missing the AND and OR gates. Complete the schematic and simulate to verify correct operation.

Then look at the alu{lay} layout. It is missing the AND, OR, and full adder gates. Add these gates and wire them together. Be sure to leave at least three metal2 tracks free for over-the-cell routing when you assemble the entire datapath, just as you left tracks free over your full adder. Be sure to export *cin*, *cout*, and *set*.

Simulate your ALU layout and verify it with DRC, ERC, and NCC.

3. What to Turn In

Please provide a hard copy of each of the following items:

1. Please indicate how many hours you spent on this lab. This will not affect your grade, but will be helpful for calibrating the workload for the future.
2. What was unclear in this lab writeup? How would you change it to run more smoothly next time?
3. A printout of your fulladder schematic (and any subcells, if applicable).
4. A printout of your fulladder layout.
5. Simulation waveforms demonstrating correct operation of the fulladder layout.
6. A printout of your alu layout.
7. Simulation waveforms demonstrating correct operation of the alu layout.
8. What is the verification status of your layouts? Does they pass DRC? ERC? NCC?

Optimization Contest

A prize will be given for the best full adder design. This primarily means smallest area. However, easy access to inputs and outputs and clean layout are also important.

Extra Credit

As you are probably aware by now, Electric has plenty of bugs and idiosyncrasies. A major goal of this class is to improve the stability and ease-of-use of Electric. Please email your bug reports directly to Prof. Harris in the format described in Lab Manual 1.