

VLSI
October 25, 2025
Winston Har 873637451

Lab 5: Full Adder

Introduction:

In this lab, the goal was to build a full adder from the ground up, applying the principles of schematic design, physical layout, and simulation. This was a two-part process. The first part involved designing the fundamental building blocks: a 2-input NAND gate, a NOT gate (inverter), and a 2-input XOR gate. For each of these gates, I was required to create the schematic using 6u/2u MOSFETs, design a standard-cell physical layout, create a custom symbol, and simulate its logical behavior to verify functionality. A key requirement was to create all layouts within a standard cell frame of a consistent height (70 units) to allow them to be seamlessly placed end-to-end.

In the second part, I used these verified gates to construct the full adder schematic and its corresponding composite layout by arranging the individual cell layouts. The final step was to perform a transient simulation on the complete full adder to confirm it correctly performed binary addition for all possible input combinations, thus validating the entire design flow from basic gates to a complex arithmetic circuit.

Lab work:

Lab 5 – Directions

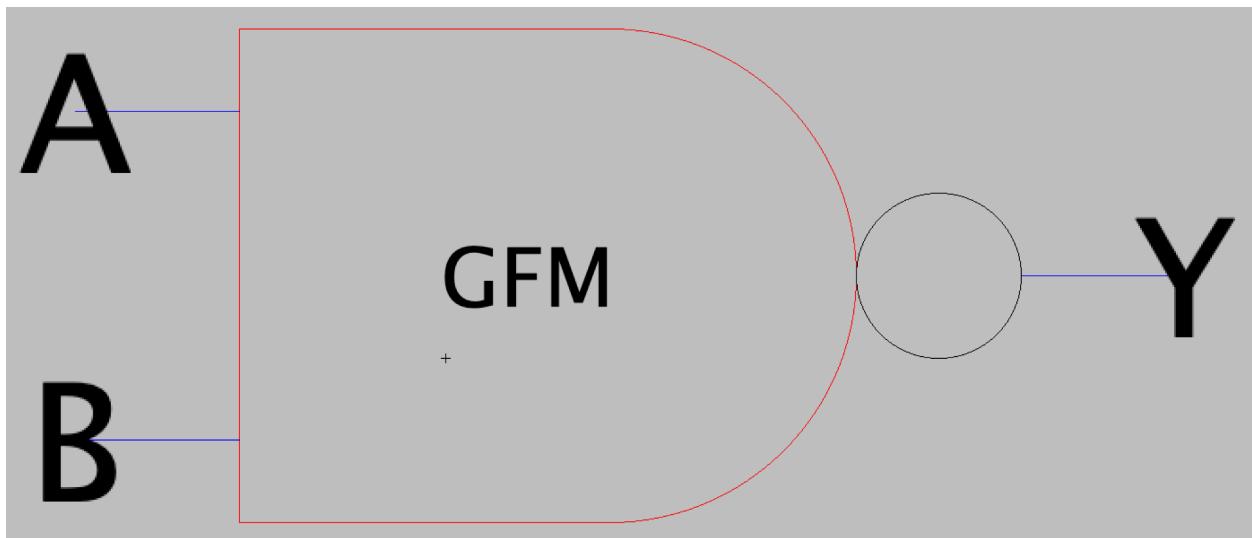
Part 1

Draft schematics, layouts, and symbols for a 2-input NAND gate and a 2-input XOR gate using 6u/2u MOSFETs (both NMOS and PMOS)

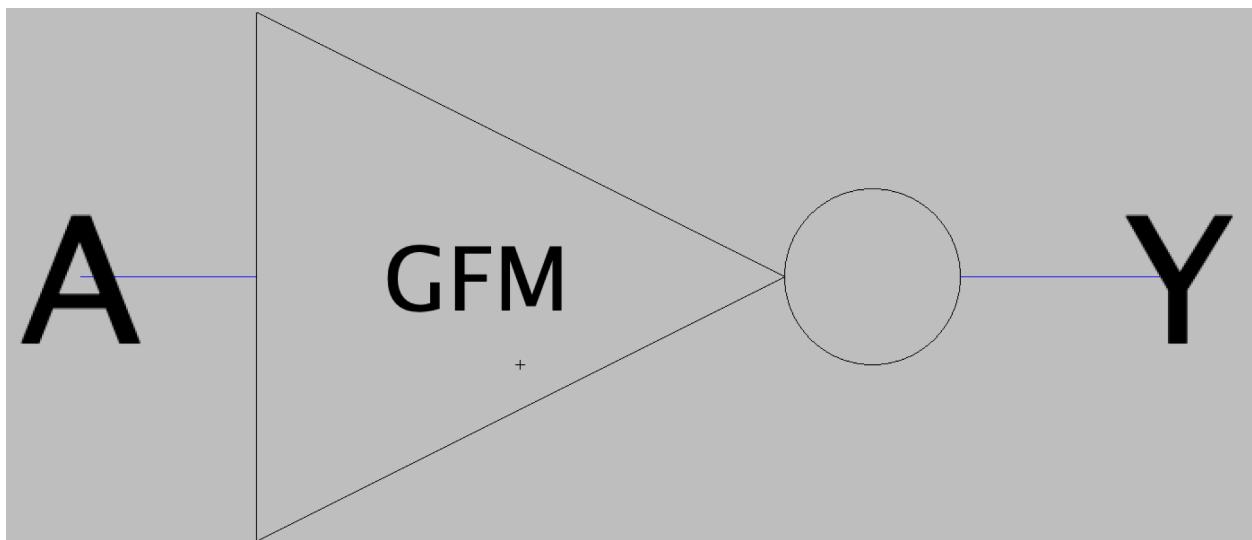
- Ensure that your symbol views are commonly used symbols (not boxes) for these gates with your initials in the middle of the symbol

Screenshots of icons

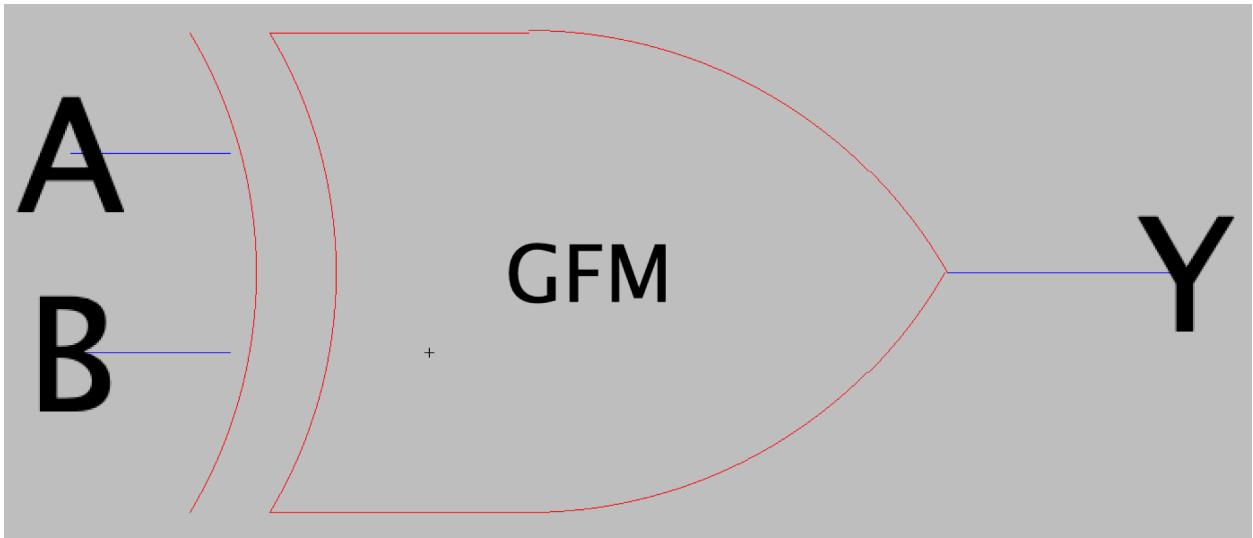
NAND



NOT



XOR



- Ensure all layouts in this lab use standard cell frames that snap together end-to-end for routing vdd and gnd.

To do this I created the NAND cell first. After creating the layout I ended up with a height of 70 units. I then created all the other cells (XOR and NOT) to also be 70 units height.

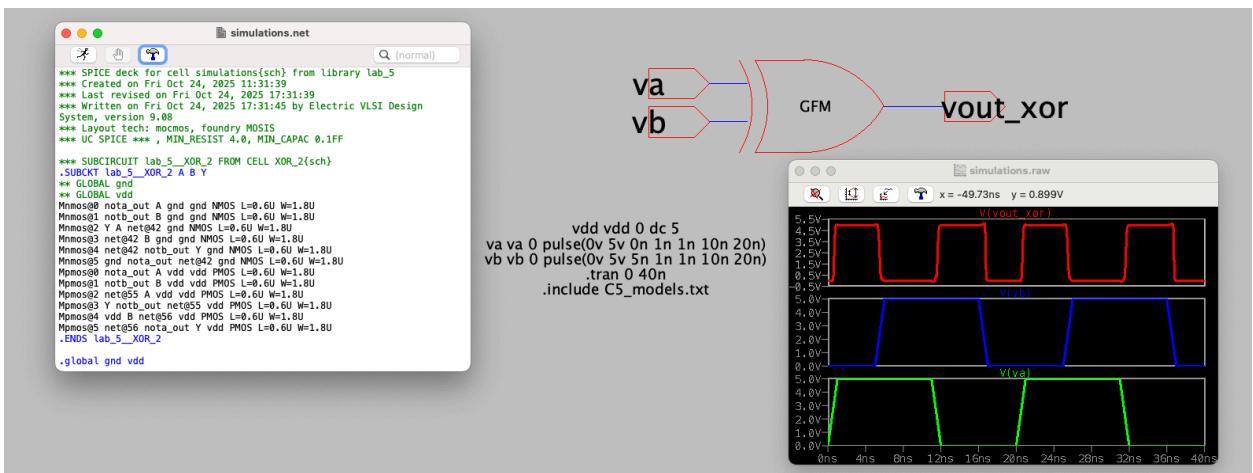
- Ensure gate inputs, outputs, vdd, and gnd are all routed in metal1

This was done, although sometimes a little janky as I had to use a metal-2-metal-1 connector just to step down the wires, especially for the end layout for the full adder.

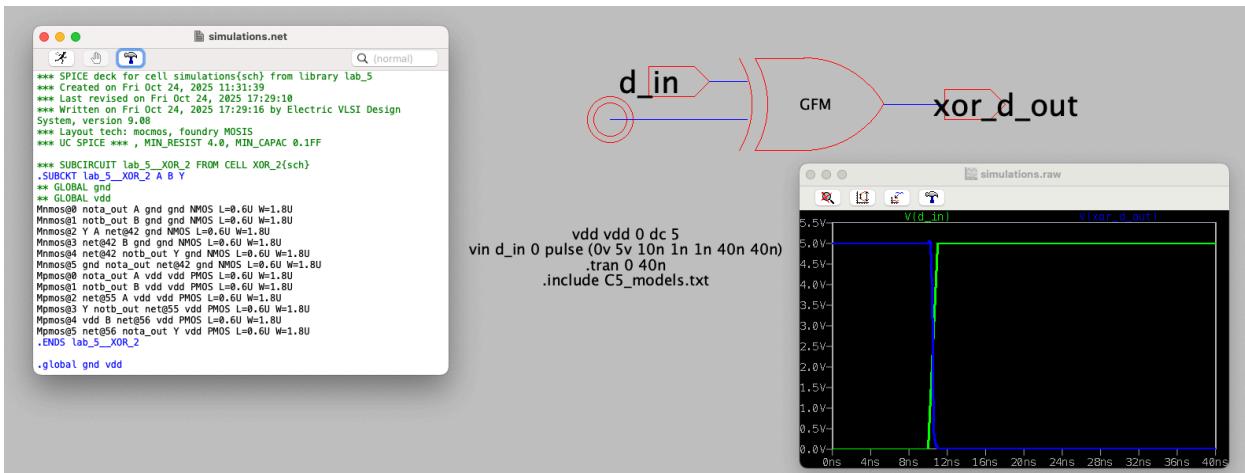
- Simulate the logic operation of the gates for all 4 possible inputs (00, 01, 10, 11).

Screenshots of Simulations

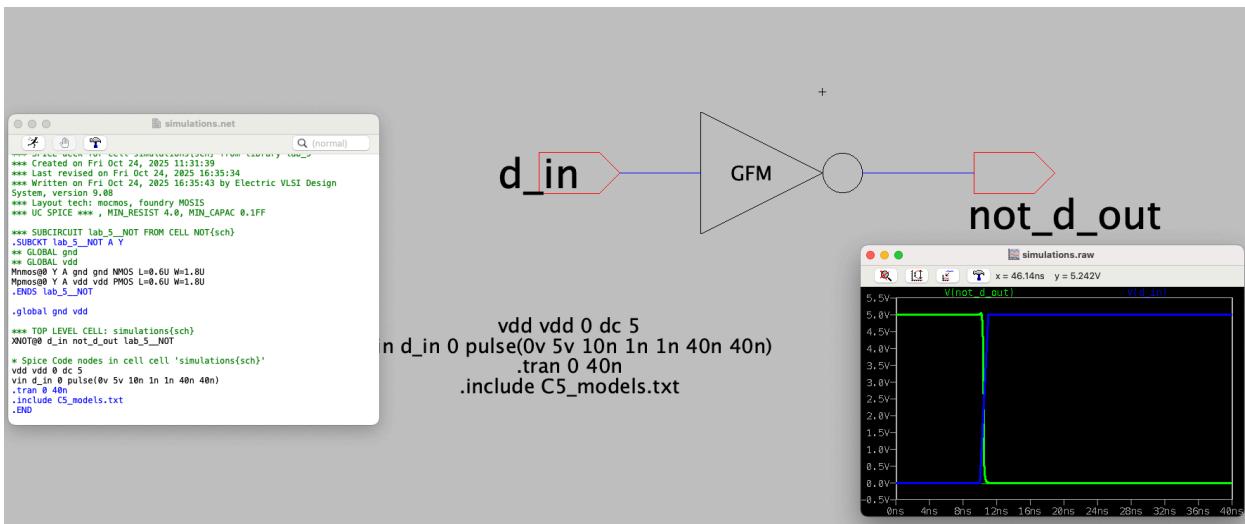
Xor 4 input test simulation



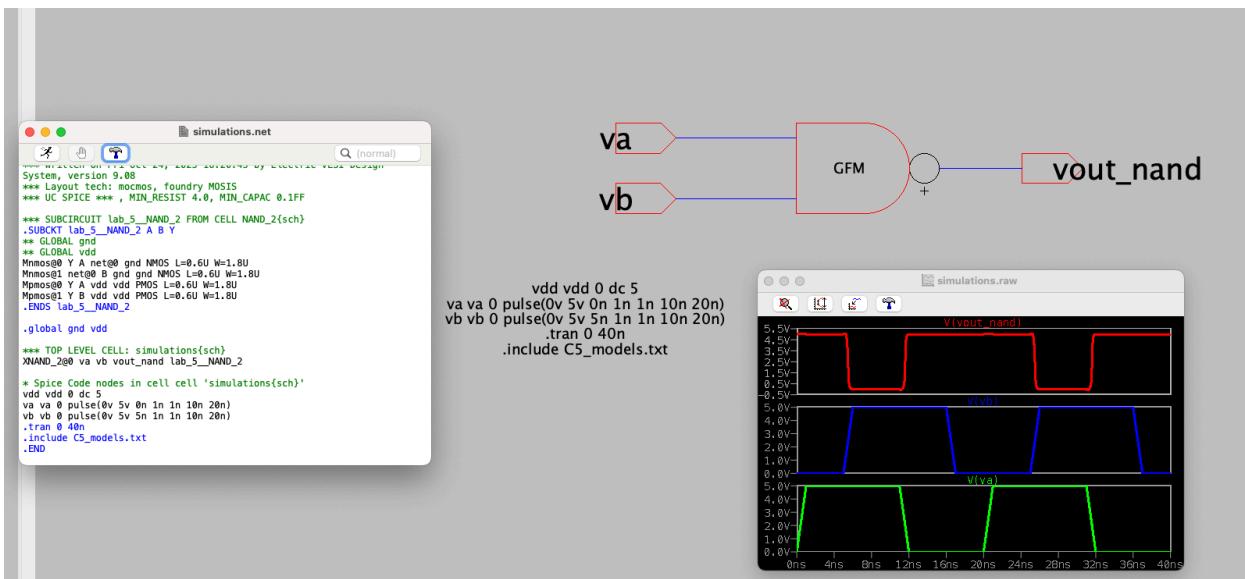
Xor crossover simulation



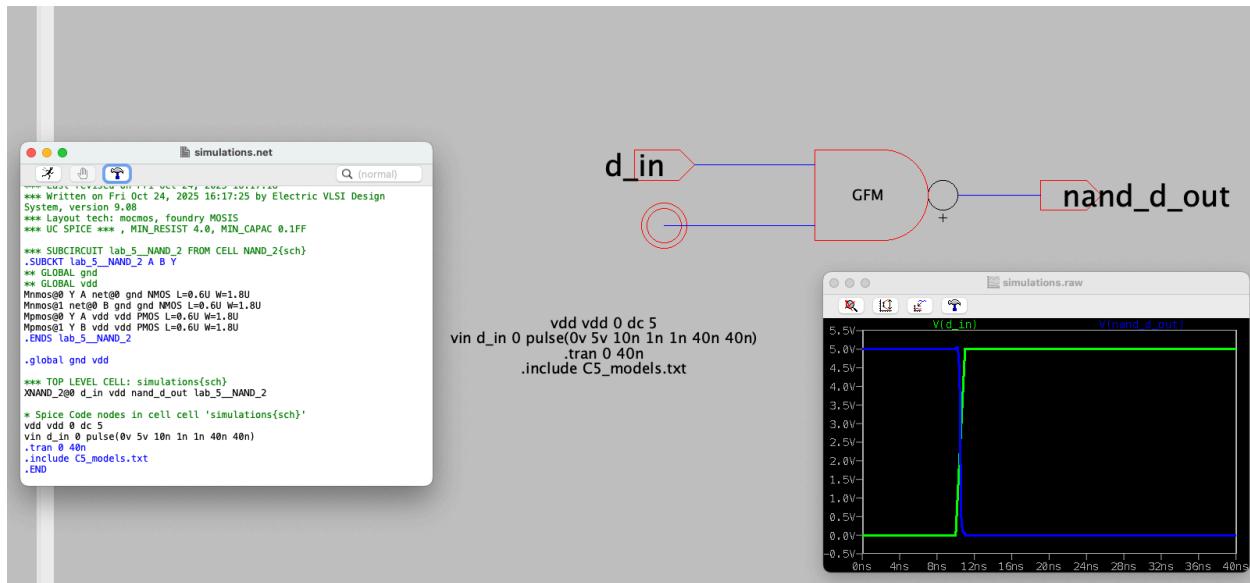
Not 4 input test



NAND 4 input test



Nand crossover test



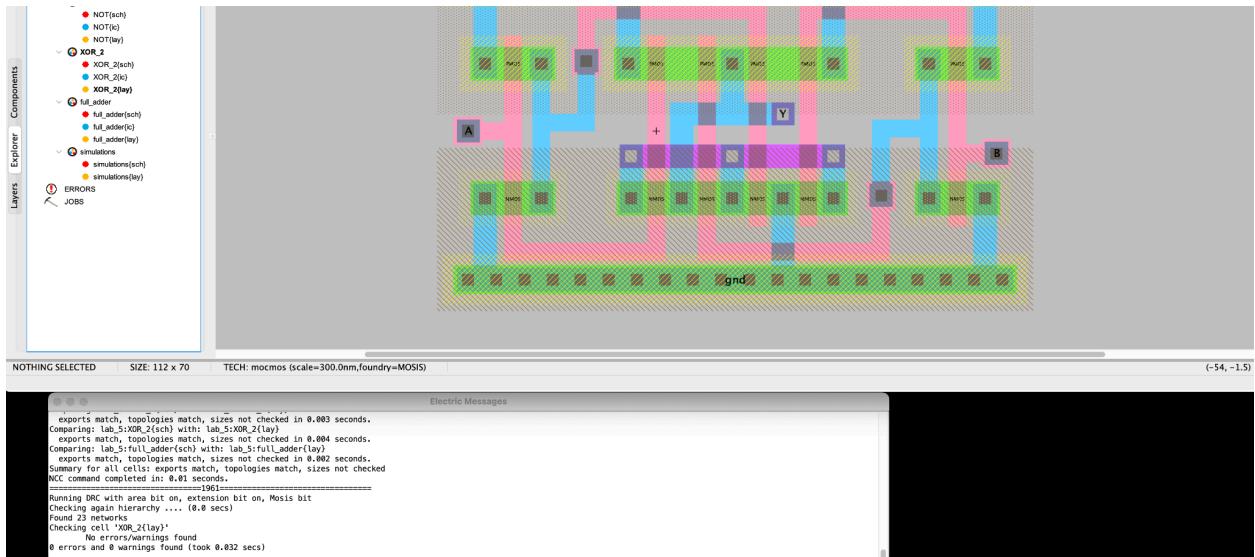
- Comment on how timing of the input pulses can cause glitches in the output of a gate.

When designing logic gates, the timing of input signals is critical. Even if the logic is functionally correct, a "race condition" can occur if inputs that are supposed to change simultaneously arrive at the gate at slightly different times. This can cause a brief, unintended pulse at the output, known as a glitch or hazard.

Lets say in an XOR gate, if the inputs are supposed to switch from (0,1) to (1,0), the output should remain high. However, if the first input changes slightly before the second, the gate might momentarily see an intermediate state of (1,1) or (0,0). Both of these states result in a low output for an XOR gate. This would cause the output to briefly dip low before returning high, creating a glitch. These glitches are often very short but can cause significant problems in larger synchronous systems if they occur near a clock edge, potentially leading to incorrect data being latched. Careful layout to balance signal path delays can help minimize these effects, but they are an inherent risk in combinatorial logic.

- DRC and NCC your designs
- Screenshots for DRC and NCC of Cells**

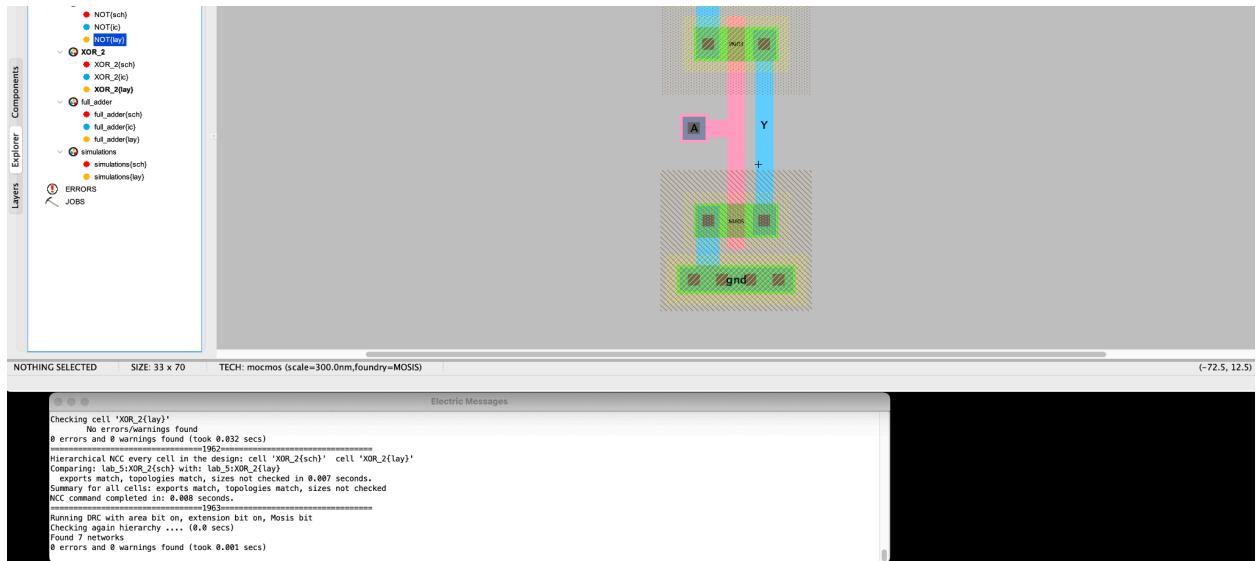
XOR DRC



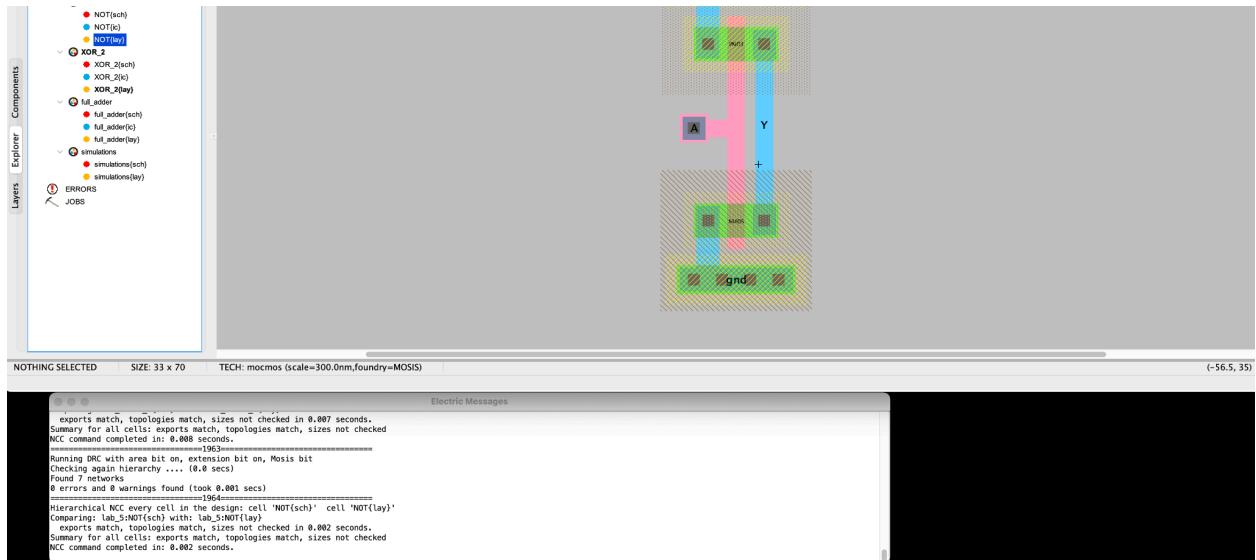
XOR NCC



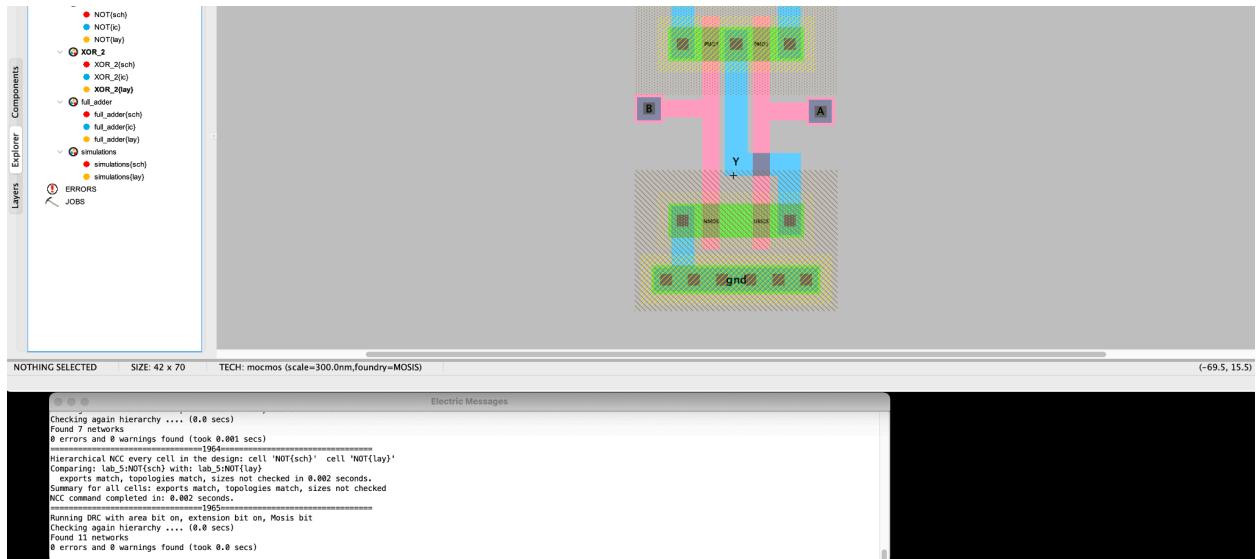
NOT DRC



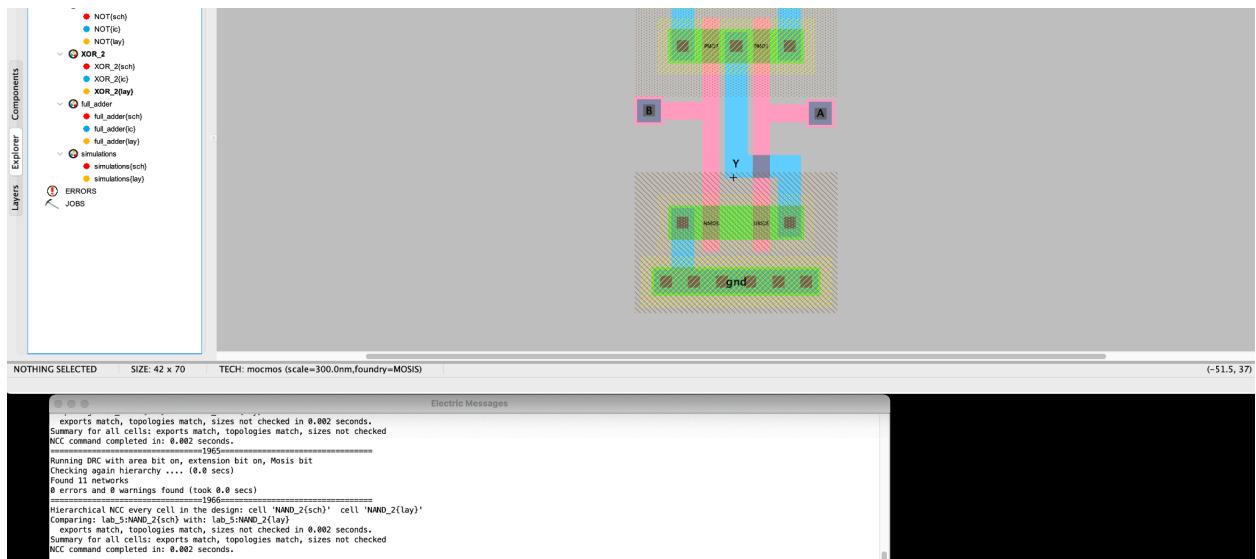
NOT NCC



NAND DRC



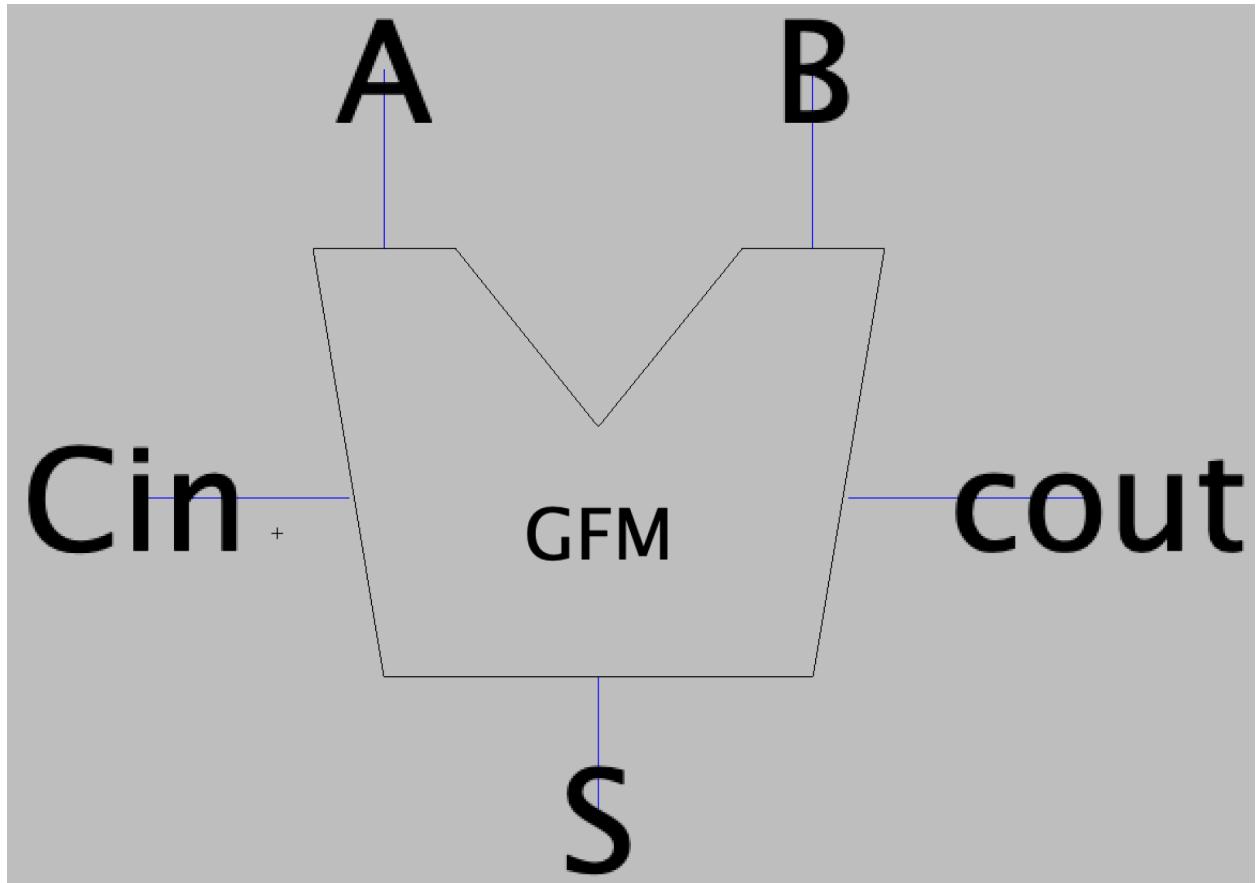
NAND NCC



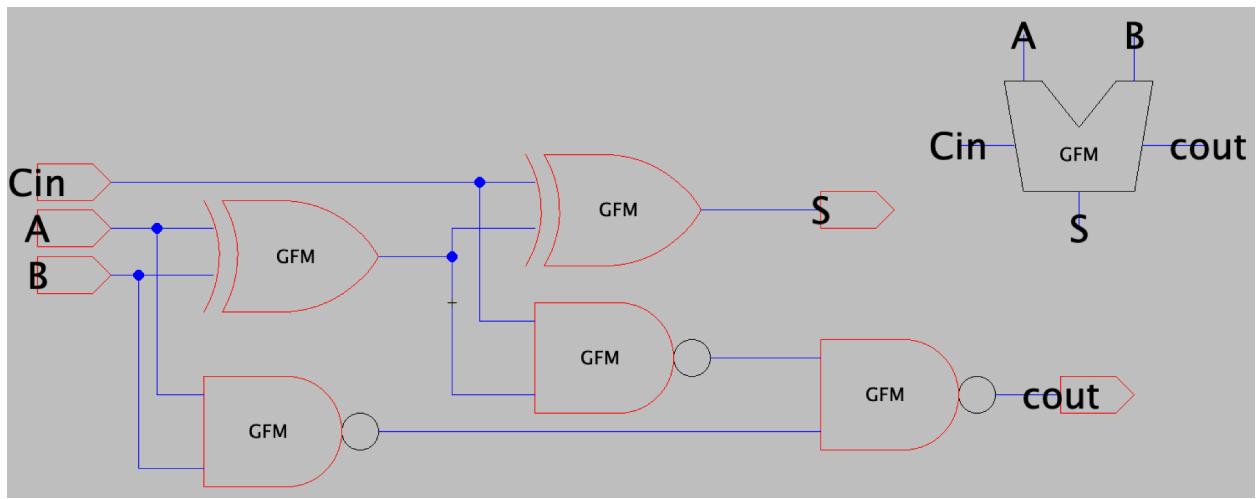
Part 2

Using the gates from Part 1, draft the schematic, layout, and symbol of a full adder (truth table and circuit below).

Full adder symbol

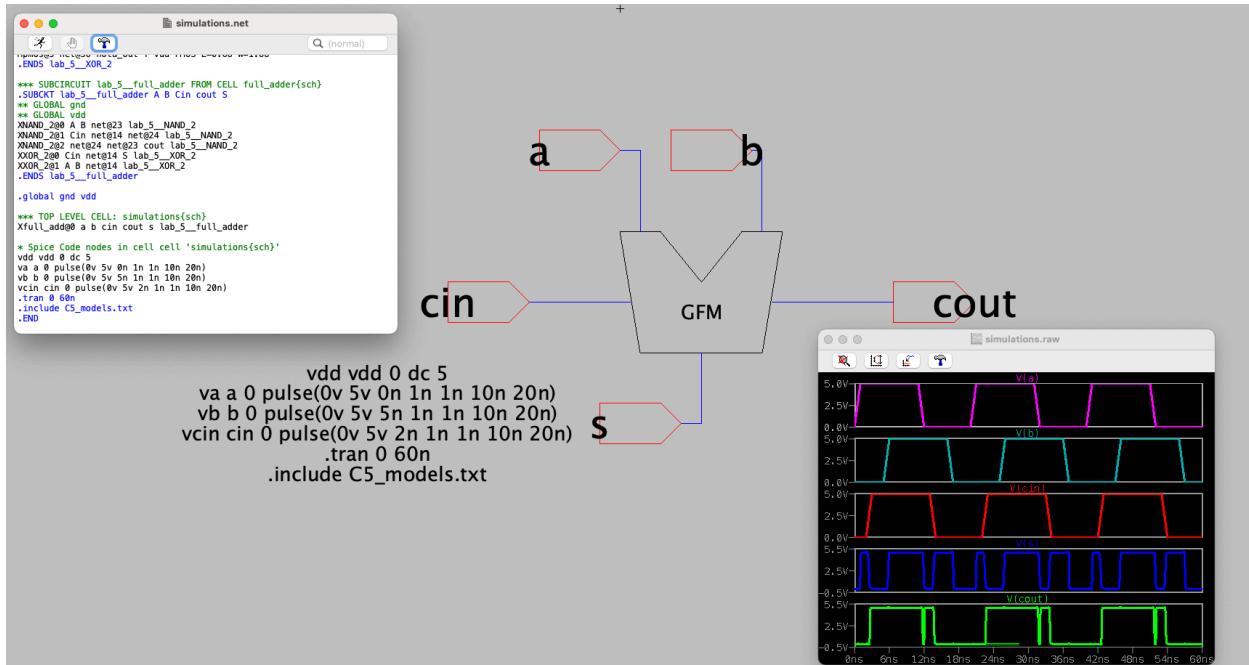


Full adder schematic



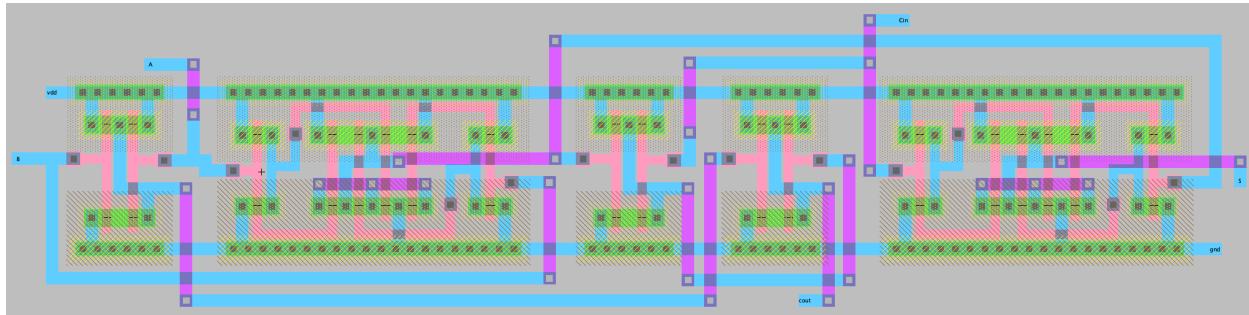
- Simulate the operation of the full adder

Full Adder Sim



- Layout the full adder by placing the 5 gates end-to-end so that vdd and gnd are routed

Full Adder Layout



- Full adder inputs and outputs can be on metal 2 but not metal 3

For this I decided to stick to the metal 1 for the full adder inputs/outputs, to stay consistent with the rule from the cells. When making the layout I just pieced the layouts from the cells I already had. This was in opposition from the instructional slides, I believe, as I noticed that vdd nwell for the left most nand and xor were connected (I instead just jumped the gap with a metal 1 wire).

- DRC and NCC your designs

Screenshots DRC and NCC for full adder

Full adder layout DRC

Nothing selected | Size: 418.5 x 100 | TECH: mocomos (scale=300.0nm,foundry=MOSIS) | (-58, -61.5)

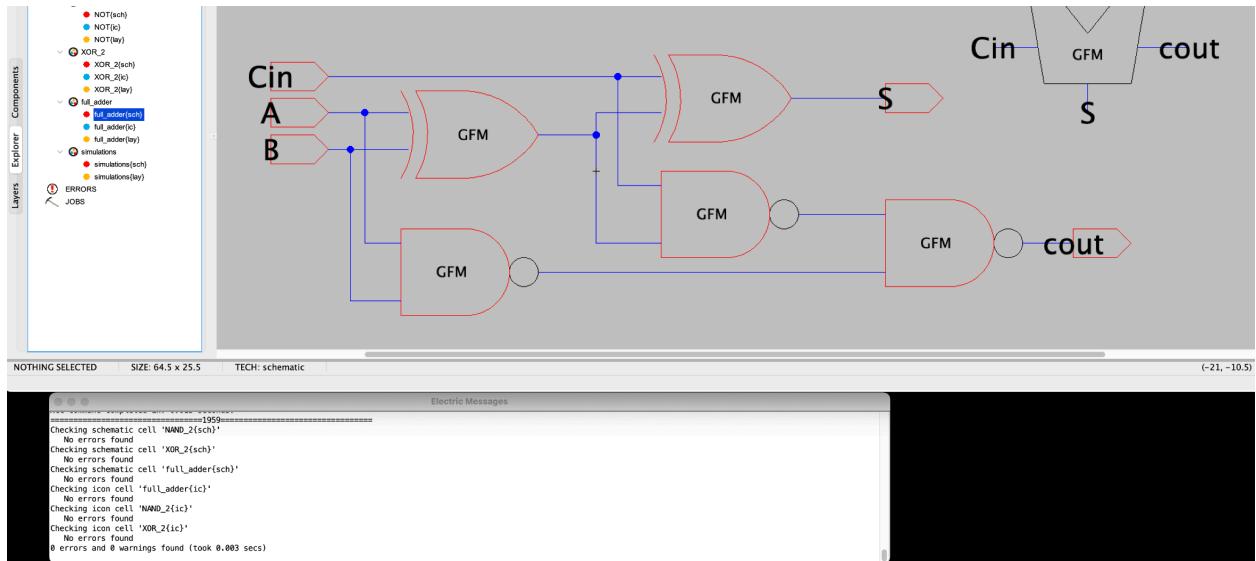
```
No errors/warnings found
Checking cell 'full_adder[lay]'
No errors/warnings found
0 errors or 0 warnings found (took 0.056 secs)
=====
@/users/winstomhar/Desktop/OU computer science/vlsi/Projects/lab_5/lab_5.jelib written
=====
Running DRC with area bit on, extension bit on, Mosis bit
Checking again hierarchy ... (0.001 secs)
Found 11 networks
0 errors and 0 warnings found (took 0.001 secs)
```

Full adder layout NCC

Nothing selected | Size: 418.5 x 100 | TECH: mocomos (scale=300.0nm,foundry=MOSIS) | (3.5, 86)

```
0 errors and 0 warnings found (took 0.001 secs)
=====
Hierarchical NCC every cell in the design cell 'full_adder(sch)' cell 'full_adder[lay]'
Comparing: lab_5:NAND_2(sch) with: lab_5:NAND_2[lay]
exports match, topologies match, sizes not checked in 0.004 seconds.
Comparing: lab_5:XOR_2(sch) with: lab_5:XOR_2[lay]
exports match, topologies match, sizes not checked in 0.002 seconds.
Comparing: lab_5:full_adder(sch) with: lab_5:full_adder[lay]
exports match, topologies match, sizes not checked in 0.003 seconds.
Summary for all cells: exports match, topologies match, sizes not checked
NCC command completed in: 0.013 seconds.
```

Full adder schematic DRC



Conclusion:

Overall, Lab 5 was an excellent practical application of digital logic design principles, taking the process from individual transistors all the way to a complex functional unit like a full adder. Building the basic NAND, NOT, and XOR gates in Part 1 provided a solid foundation. The constraint of using a standard cell height was particularly insightful, as it forced me to think about layout planning and how individual cells would eventually integrate into a larger system. It really demonstrated the importance of modular design in VLSI.

The most challenging part was laying out the full adder in Part 2. Stitching the five cells together required careful routing to connect the outputs of one gate to the inputs of another. I had to adjust the y output for the xor to be able to get everything to fit into a small-ish area while not staggering anything but after a little tweaking it was fine. Seeing the final simulation of the full adder work correctly for all eight input combinations was very satisfying. It successfully validated the entire design flow, from the individual gate schematics to the final composite layout.