Lab 4: From Boolean Equation to Silicon

Brittany Duffy
Partner: Jingxuan Sun
EE 330- Integrated Electronics
Lab Section B
Professor Randy Geiger
2/14/13

Introduction

During this lab, I implemented a three input NAND gate as well as a given Boolean function. First, we created a layout design using parameterized cells (pcells) to make an inverter. We then went through the full design flow of the three input NAND gate. Afterwards, we implemented a prearranged Boolean function with given area and pin constraints.

Pre-Lab

Please see the attached pre-lab on the next page.

Procedure

1. Layout of an inverter using pcells
    a. Click on Create → Instance and select NCSU_Techlib_ami06
    b. Choose nmos and pmos in the cell
    c. Edit the width and length to desired values
    d. Complete the inverter layout using pcells
    e. Passes the LVS
2. Layout of a logic function
    a. Using pcell to create the 3-input NAND gate
    b. Build the NAND gate in schematic view, create symbol for it, and build it in Layout view. Make sure it passed the LVS
    c. Using NAND gate and NOT gate to build the logic function in schematic view and build it in Layout view. Make sure it passed the LVS.
    d. Make a symbol for the logic function
    e. Make and check the test-bench to ensure proper functionality

Results & Analysis

Part 1: Layout of an inverter using pcells

Below you will find images of the inverter schematic, layout, and LVS Test. In figure 3, you can see that the net-lists have matched, and the LVS test passed.

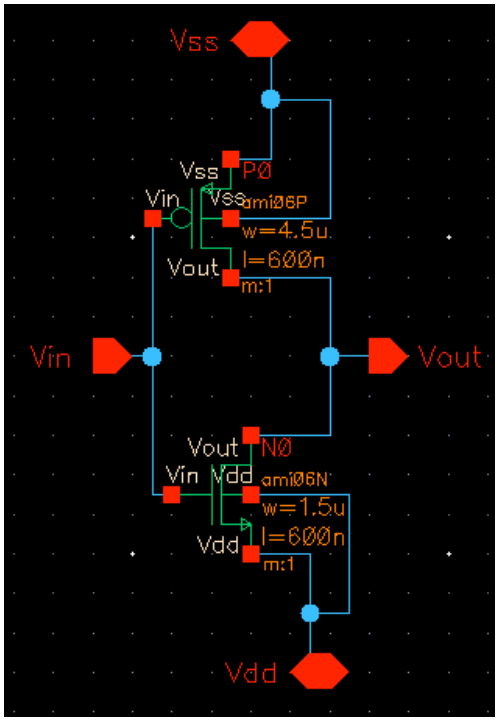

Figure 1: Inverter Schematic

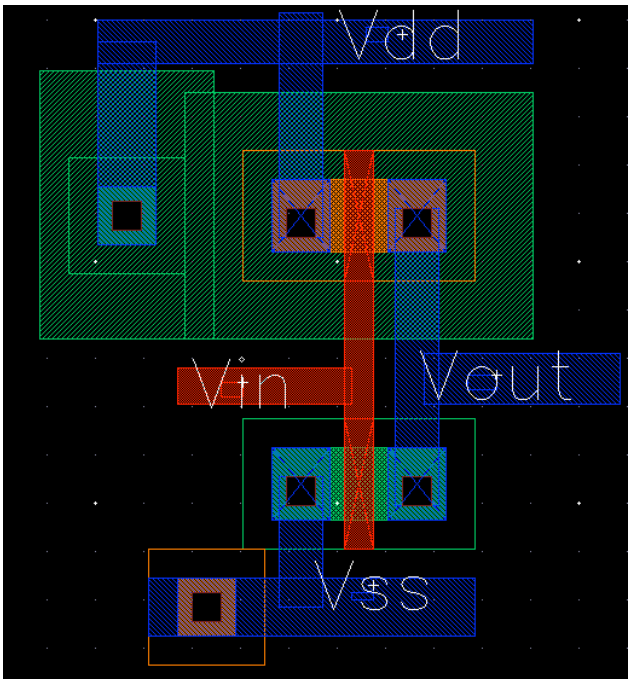

Figure 2: Inverter Layout

The net-lists match.

		layout	schematic
		instances	
un-matched	0	0	
rewired		0	0
size errors	0	0	
pruned		0	0
active		30	30
total		30	30
		nets	
un-matched	0	0	
merged		0	0
pruned		0	0
active		20	20
total		20	20
		terminals	
un-matched	0	0	
matched but different type		0	0
total		6	6

Figure 3: Net-lists match & LVS test passed

Part 2: Layout of logic function

Part 2.1: 3-Input NAND/NOR Gate

For this lab, I implemented a 3-input NAND gate while my partner chose to create the 3-input nor gate. The figures below show the schematic view, the layout, symbol, test bench, and test bench response of the 3-input NAND gate. In figure 6, you can see the LVS test was successful.

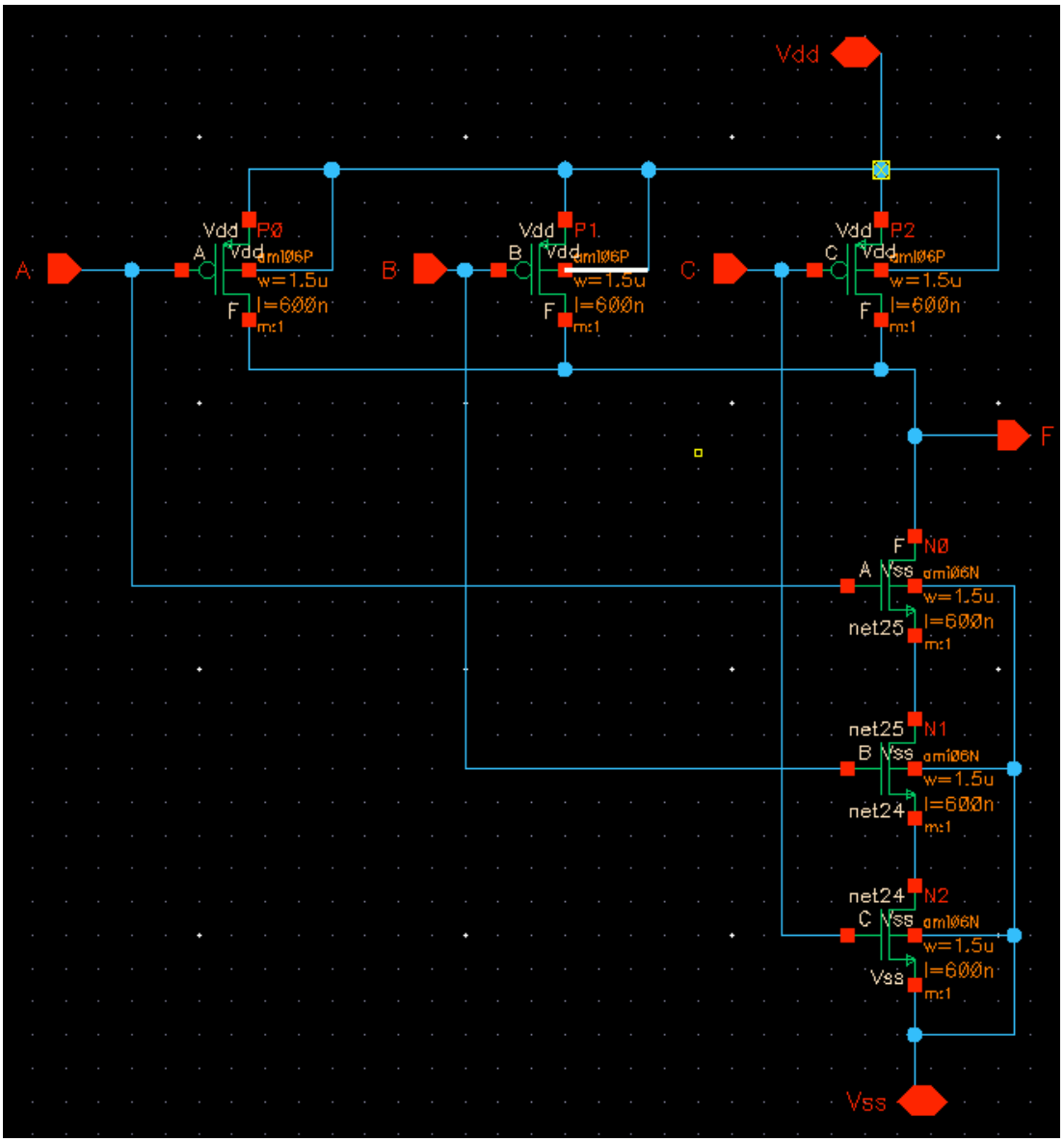

Figure 4: 3-Input NAND Gate Schematic

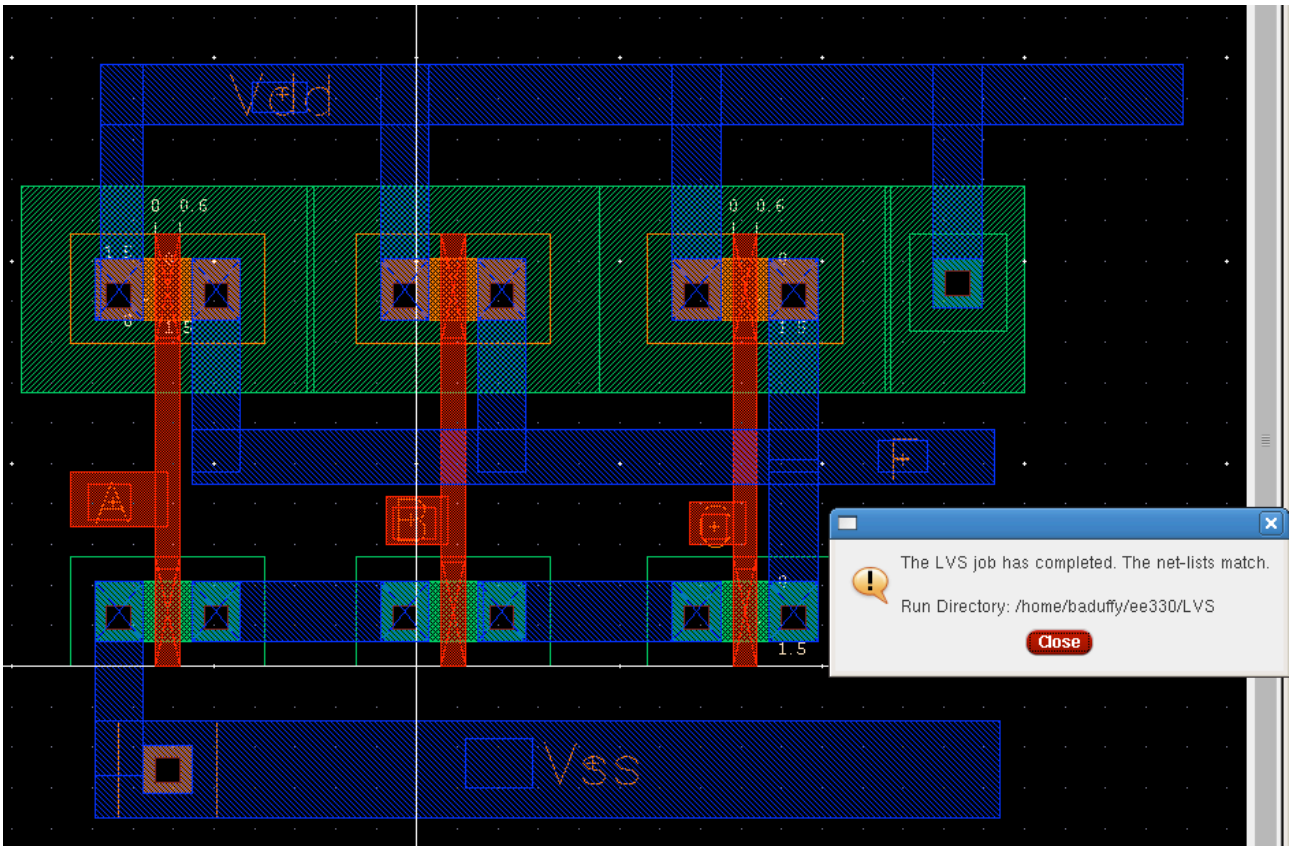

Figure 5: 3-Input NAND Gate Layout

The net-lists match.

		layout	schematic
		instances	
un-matched	0	0	
rewired		0	0
size errors	0	0	
pruned		0	0
active		6	6
total		6	6
		nets	
un-matched	0	0	
merged		0	0
pruned		0	0
active		8	8
total		8	8
		terminals	
un-matched	0	0	
matched but different type		0	0
total		6	6

Figure 6: Net-lists match & LSV test passed

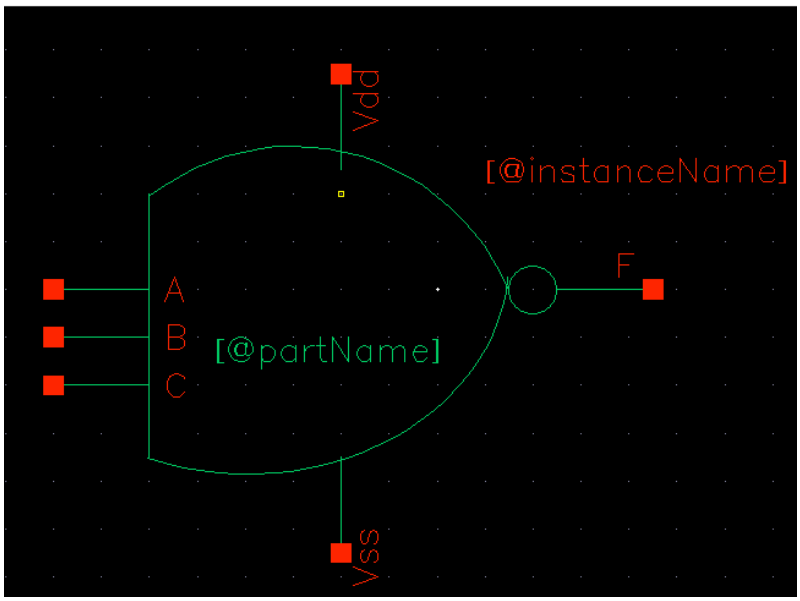

Figure 7: 3-Input NAND Gate Symbol

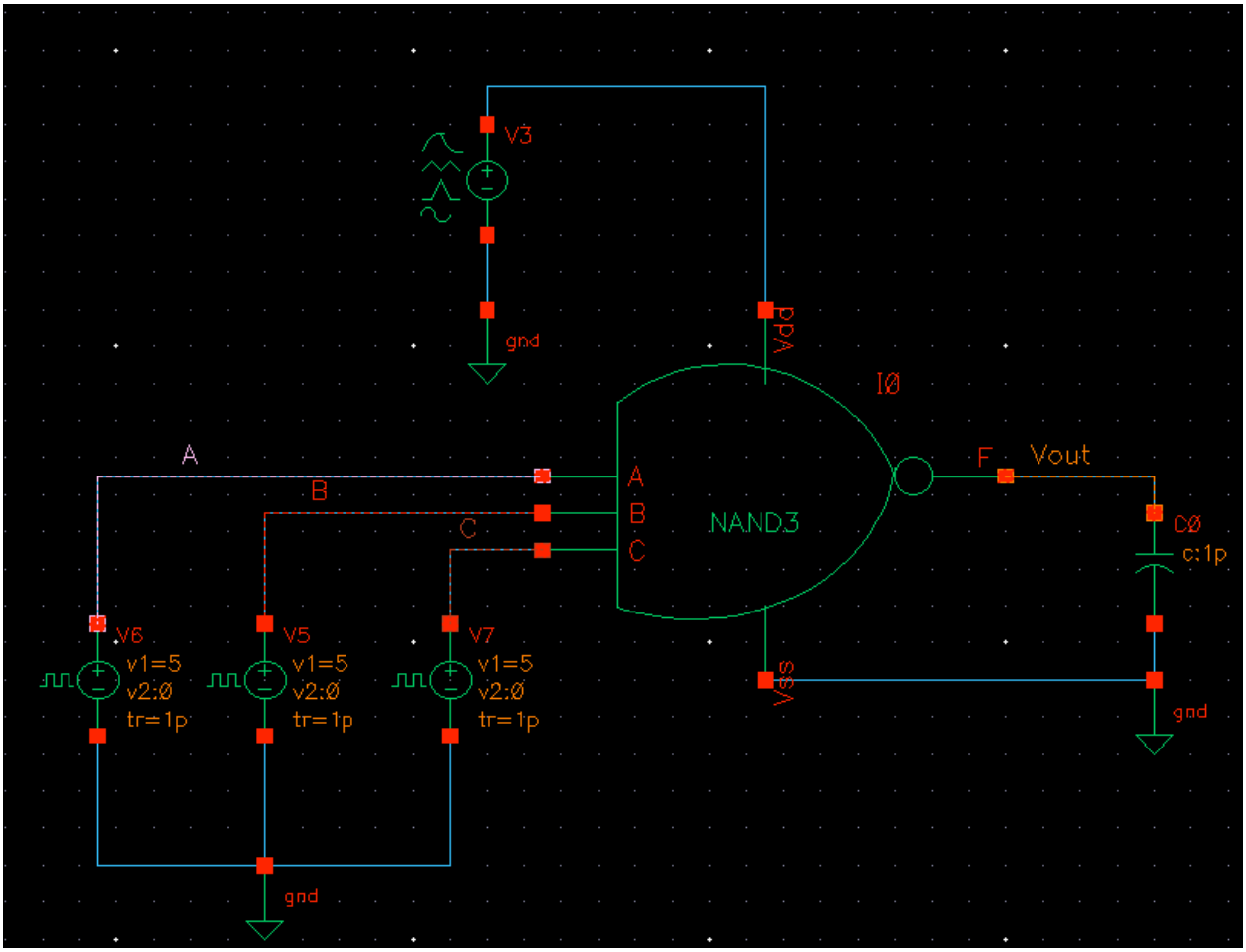

Figure 8: 3-Input NAND Gate Test Bench

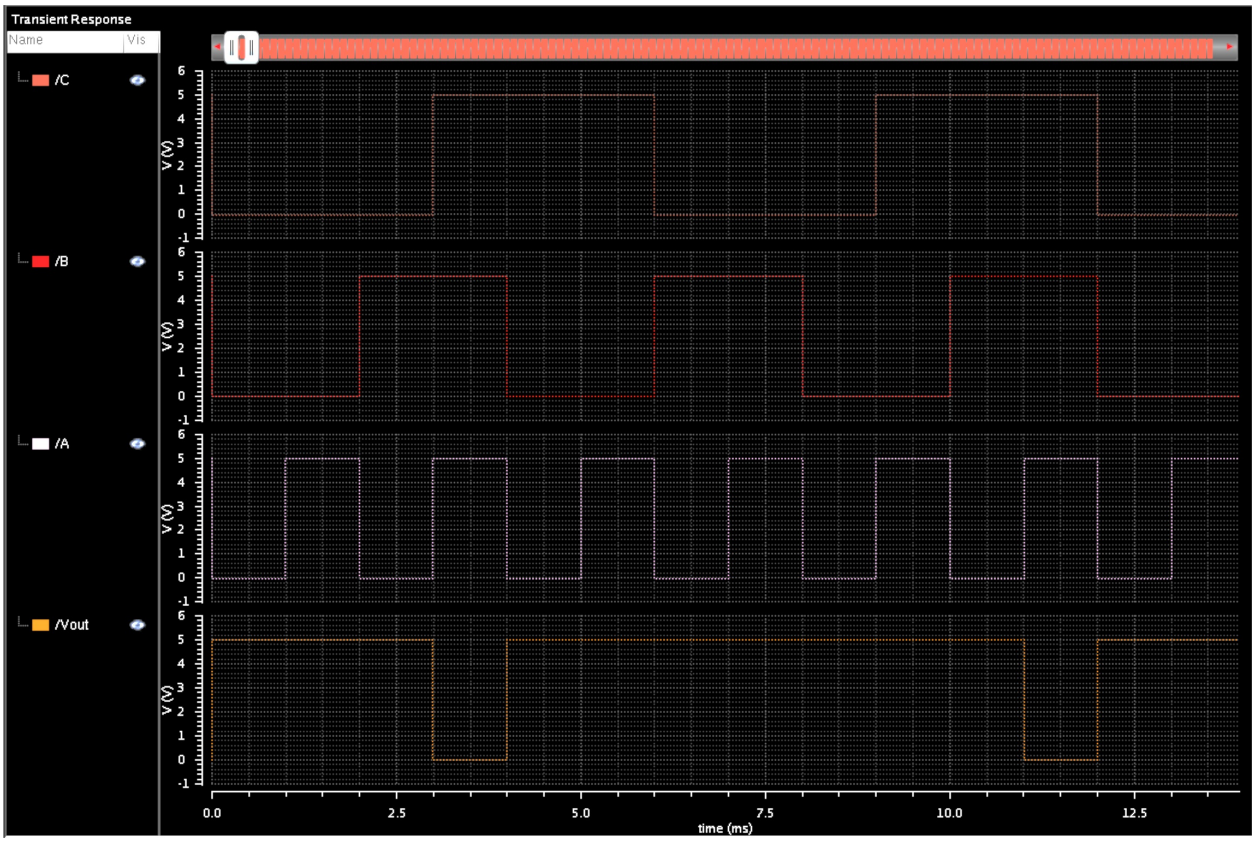

Figure 9: 3-Input NAND Test Bench Response

Analysis: The simulation result is consistent with the truth table found in the pre-lab.

Part 2.2: Boolean Function

In this section, I implemented the Boolean function: $F = \overline{A}BC + A\overline{B}C + AB\overline{C}$

Using NAND and NOT gate to represent: $F = \overline{\overline{\overline{A}BC} * \overline{A\overline{B}C} * \overline{AB\overline{C}}}$

Use the inverter from Part 1 and the NAND gate from Part 2.1, I created the above Boolean function with the entire design flow (schematic, layout, LVS, symbol, test bench). Below you will find all images gathered. Figure 13 shows that the net-lists match and the LVS was successful.

GET SCHATICCCCC Figure 10: Boolean Function Schematic

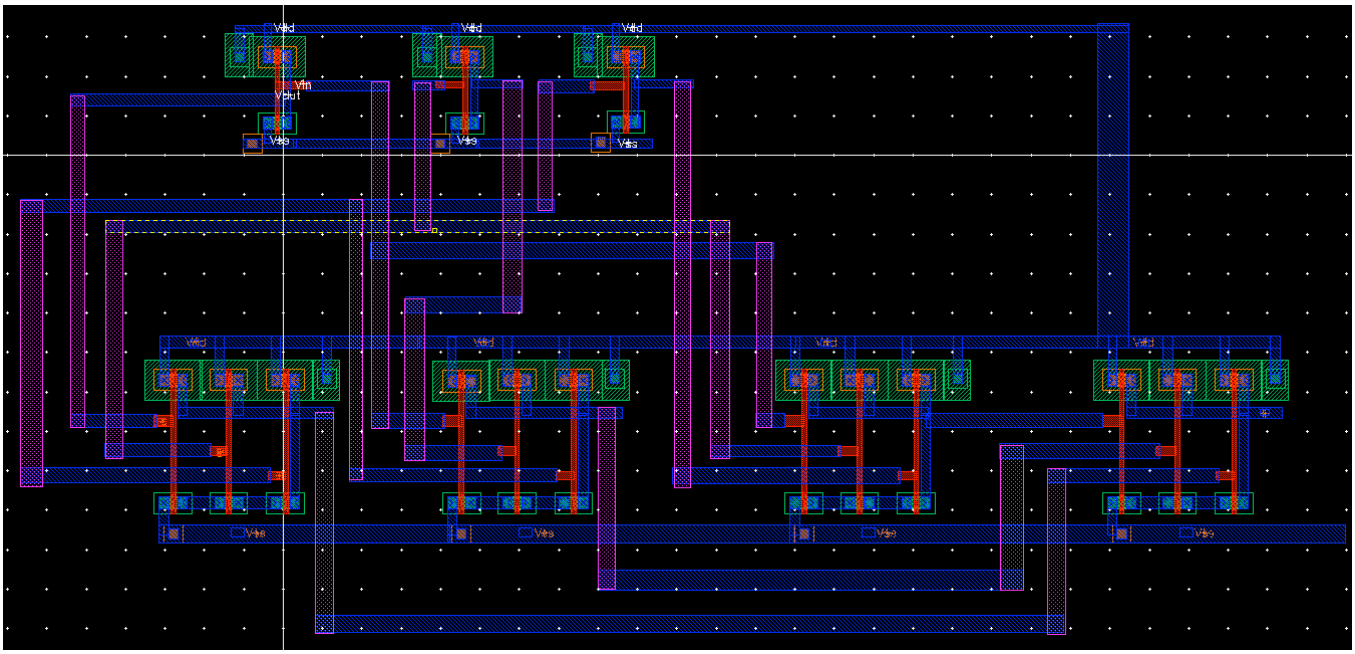

Figure 11: Boolean Function Layout

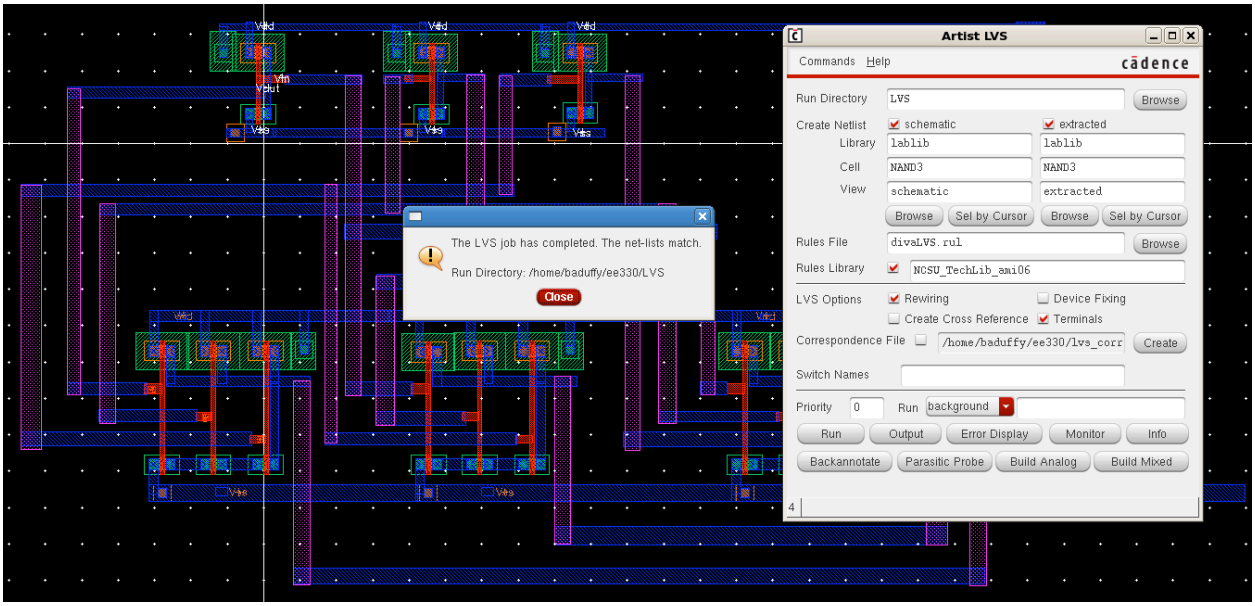

Figure 12: LVS job successful

The net-lists match.

	layout	schematic
instances		
un-matched	0	0
rewired		0
size errors	0	0
pruned		0
active		30
total		30
		0
		0
		30
		30
nets		
un-matched	0	0
merged		0
pruned		0
active		20
total		20
		0
		0
		20
		20
terminals		
un-matched	0	0
matched but different type		0
total		6
		0
		6

Figure 13: Net-lists Match

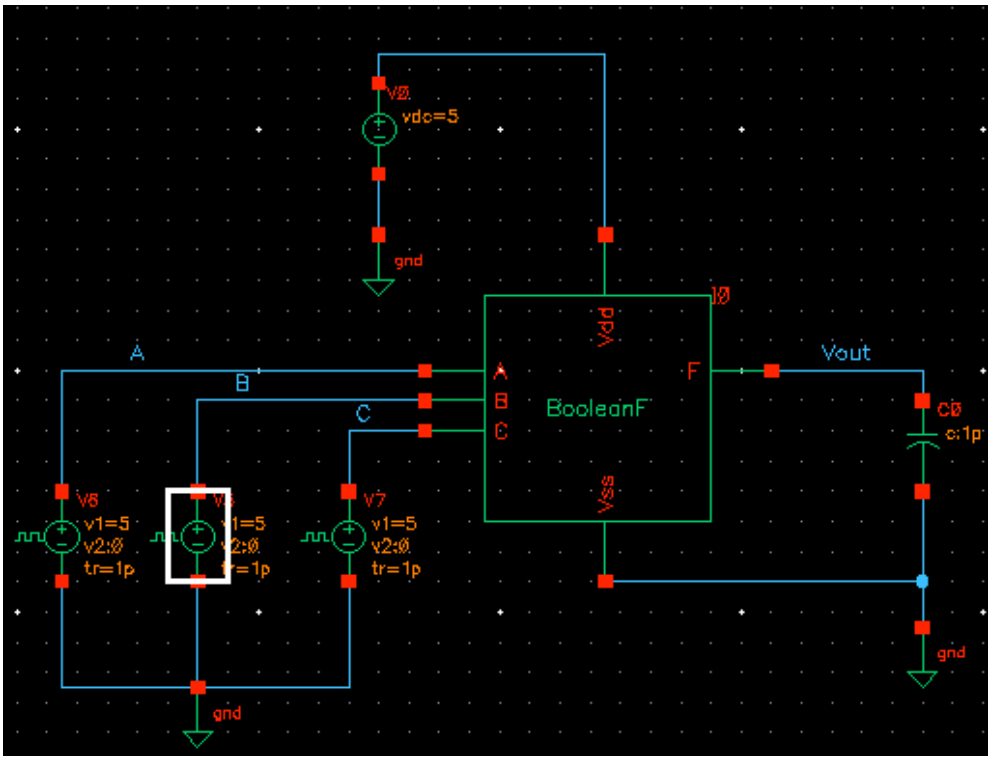

Figure 14: Boolean Function Test Bench

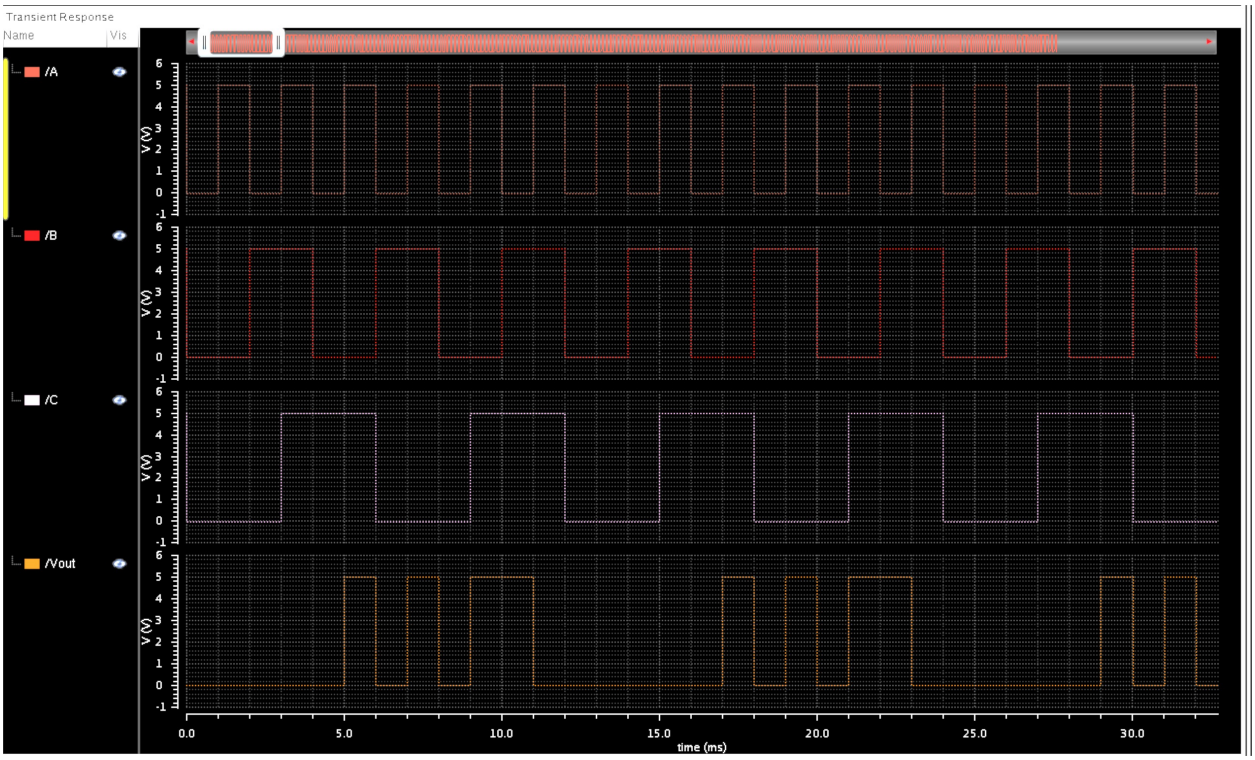

Figure 15: Boolean Function Test Bench Response

Analysis: The simulation result is consistent with the truth table found in the pre-lab.

Conclusion

During this lab, we have learned how to use the pcell to create the layout. This is a handy way to use a NMOS and PMOS. Setting the width and length of the channel for MOS is the only aspect needed to be done for the pcell. When drawing the layout for the logic function, it should become habit to check and save the implementations; you never know when Cadence will crash. Also, the design ruler and the DRC check are very important. Patience and precision were the key factors in success during this laboratory.