

EECS 119

Project 3

December 9, 2024

Alex Danielian

1. Introduction:

In this lab, I will be designing a 4-bit Full Adder/Subtractor. The circuit receives the inputs of two 4-bit numbers, $A = A_3A_2A_1A_0$ and $B = B_3B_2B_1B_0$. Depending on the control input, the circuit produces either $A+B$ or $A-B$. The subtraction portion was done using XOR gates that functioned off of Control and B, with Control being inverted and routed to C_{in} of the first 1-Bit Adder for the complete two's complement. C_{in} lines were cascaded over each 1-Bit Adder for correct carry-in. The layout of the 4-Bit Full Adder was constructed using a standard cell layout with 4 segments, two being mirrored and two normal, all layed out vertically. The GND of the first adder was connected to the GND of the second mirrored adder, whose VDD was connected to the VDD of the third normal adder, where that ones GND was connected to the GND of the fourth mirrored adder.

2. Theory:

The approach taken for the design of the schematics and layouts was the simplest and cleanest looking one. There are two types of components that make up the 4-Bit Full Adder/Subtractor. These are 1B FA blocks and the 1B FA with Control Block. These two blocks are nearly identical except for the CBlock having Control jumped into A and the normal 1B FA block not having that. The Control signal for each block is routed to an XOR gate with the B input so that when Control is High, B will be complemented and the Control that is jumped into A0 will provide the +1 needed to get the two's complement. All the gates were designed using a standard cell approach, simplified, and made neat for the final combination which is the 4B FA. Here, I took one 1B FA control block and assembled it with three other 1B FA blocks vertically, merging VDD and GND rails to provide a really neat looking, square 4B FA that saves space, is dense without much crosstalk, and readable. The Full Adders were constructed using the following design: NAND-XOR-NAND-XOR-NAND. This configuration provided the neatest approach in my opinion where many signals were able to be

routed while preserving the metal4 layer for connections that are made in the 4B FA layout. A control block was simply the same design but with an XOR at the beginning of the sequence.

3. Design [XOR and NAND]:

To begin with our preliminary designs, we will look at the XOR and NAND gates. Beginning with the XOR gate schematic, I used two inverters to begin the sequence and coupled those to a network that evaluated the following equation: $A\bar{B} + \bar{A}B = AxorB$. Regarding the layout, what was needed were two inverters in succession followed by the circuit! It was really simple to do since all the inputs of the sub-network's PUN were linked which allowed me to find the Euler's Path that worked with both the PUN and PDN $[B - \bar{B} - A - \bar{A}]$.

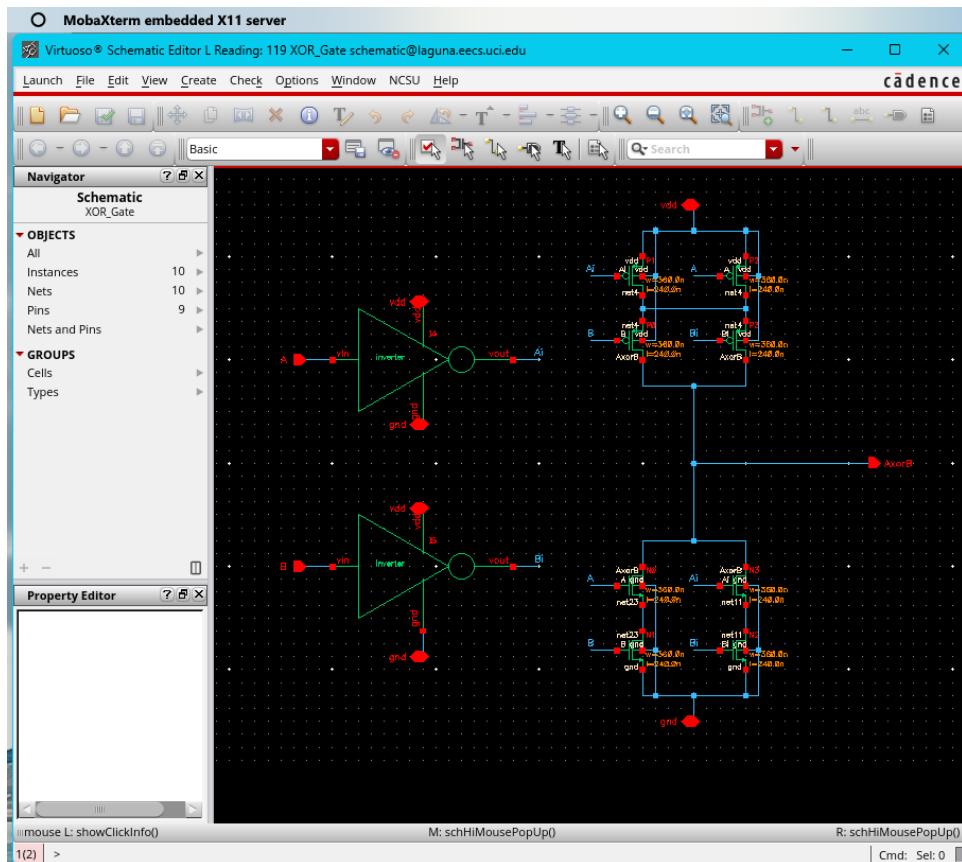


Figure 1: XOR Gate Schematic

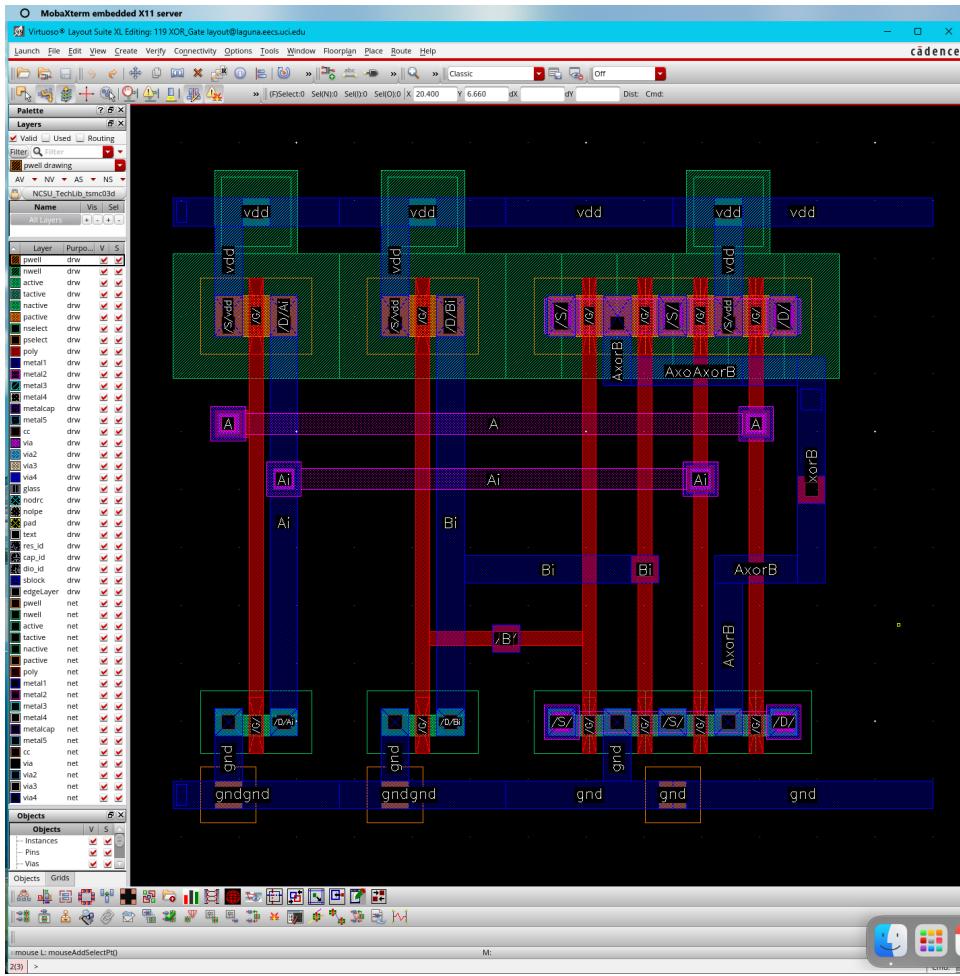


Figure 2: XOR Gate Layout

4. Design [Full Adder]:

Let's move on to the Full Adder. In this schematic, I used a sequence of XOR and NAND gates to create a 1-Bit Full Adder in order to conserve space in comparison to other approaches which used a bundle of NANDs. This schematic created the outputs for the Sum and Cout. S was calculated using the XOR output of A, B with Cin, and Cout was done with NANDing and XORing A and B then NANDing again with Cin and NANDing that output with AnandB to obtain it. The layout was standardized and designed with a standard cell approach by continuing the sequence of gates horizontally. This was because verticality was only meant to be used in the final 4-Bit Adder to prevent complications.

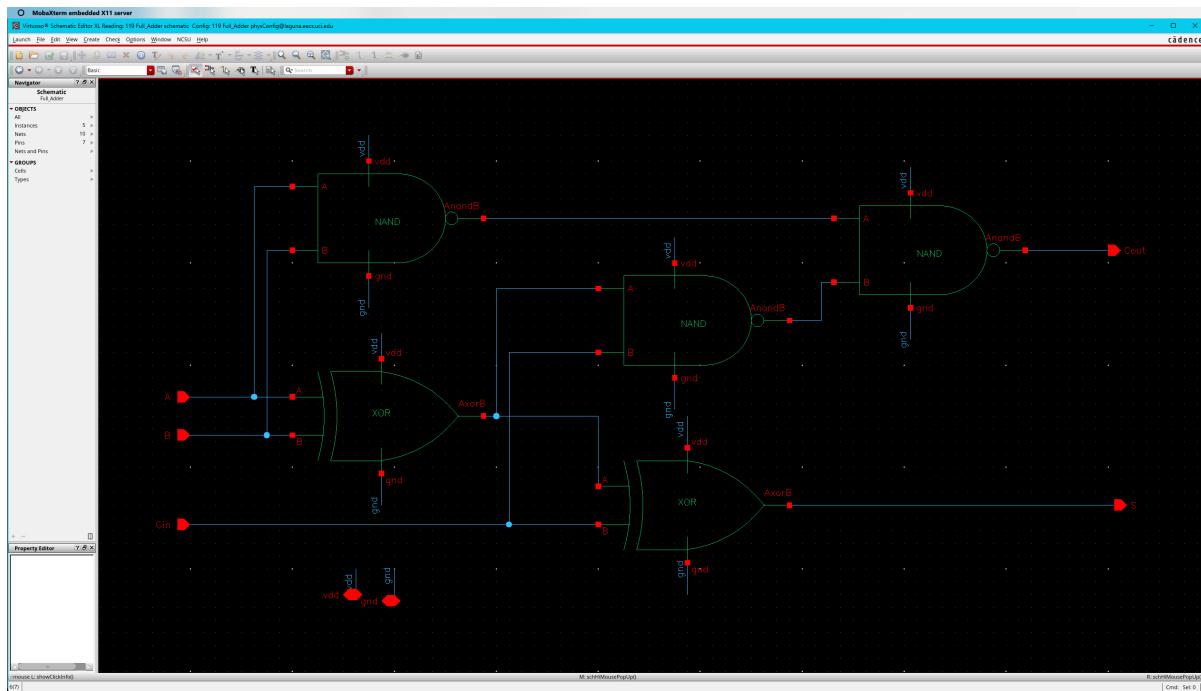


Figure 3: Full Adder Schematic

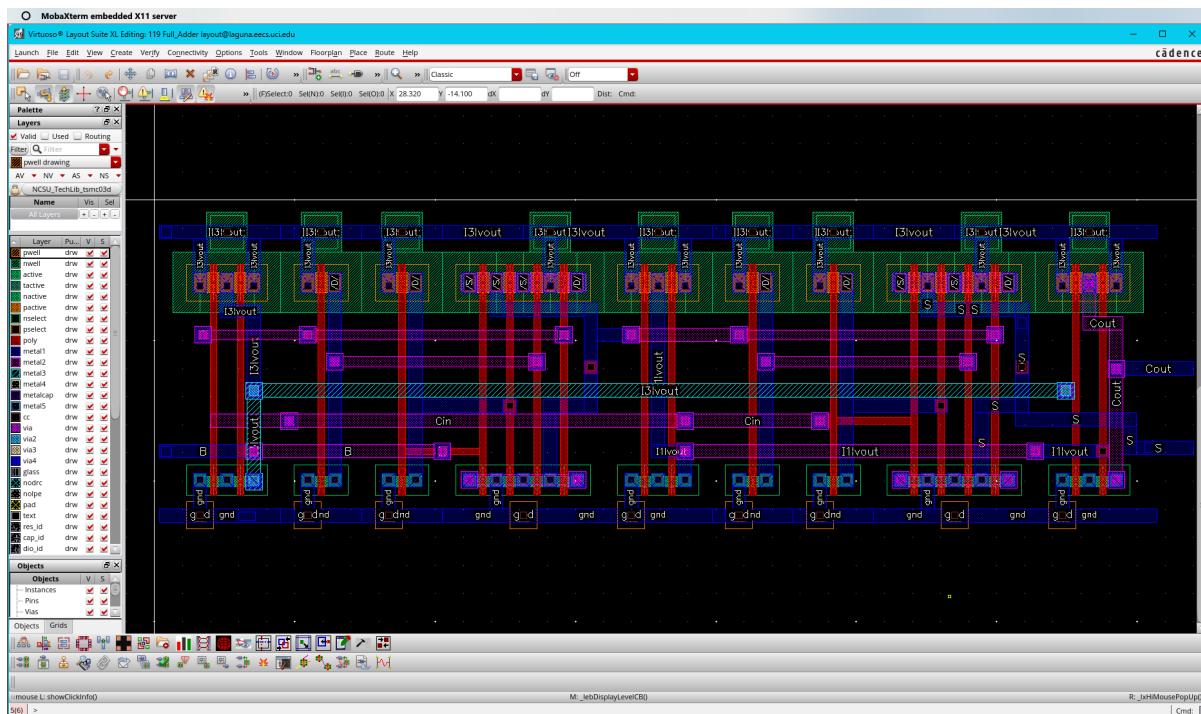


Figure 4: Full Adder Layout

5. Design [1B FA and CBlock]:

Onto the complete adder with Control input, its quite simple in that the schematic and layout are similar to the Full Adder except with the addition of an XOR gate at the beginning of each of them.

6. Design [4 Bit Full Adder]:

There are two types of components that make up the 4-Bit Full Adder/Subtractor. These are 1B FA blocks and the 1B FA with Control Block. These two blocks are nearly identical except for the CBlock having Control jumped into A and the normal 1B FA block not having that. The Control signal for each block is routed to an XOR gate with the B input so that when Control is High, B will be complemented and the Control that is jumped into A0 will provide the +1 needed to get the two's complement.

7. Operations:

The operations were all done successfully. To reference the equation that is being inputted, refer to Control HIGH for subtraction and Control LOW for addition. Necessary inputs are prioritized since I could not fit everything in the viewer, but I did merge inputs when both had Voltage = 0 at certain instances. In these outputs, the only small issue I realized was when Cout was high when it shouldn't be in instances where the output was zero. Otherwise, everything was correct as I saw it and aligned with the outputs that I expected.

8. Conclusion:

In conclusion, this lab was really hard, both in the amount of work required to complete it and the amount of debugging necessary to get things working. I stumbled across a variety of issues in my approaches to the layout and schematic design in the beginning, but as I used and got better with Cadence my design approaches to schematics and layouts became much more streamlined and elegant. Overall, every circuit worked and it was great seeing how my modular creations were combined into one giant solution.

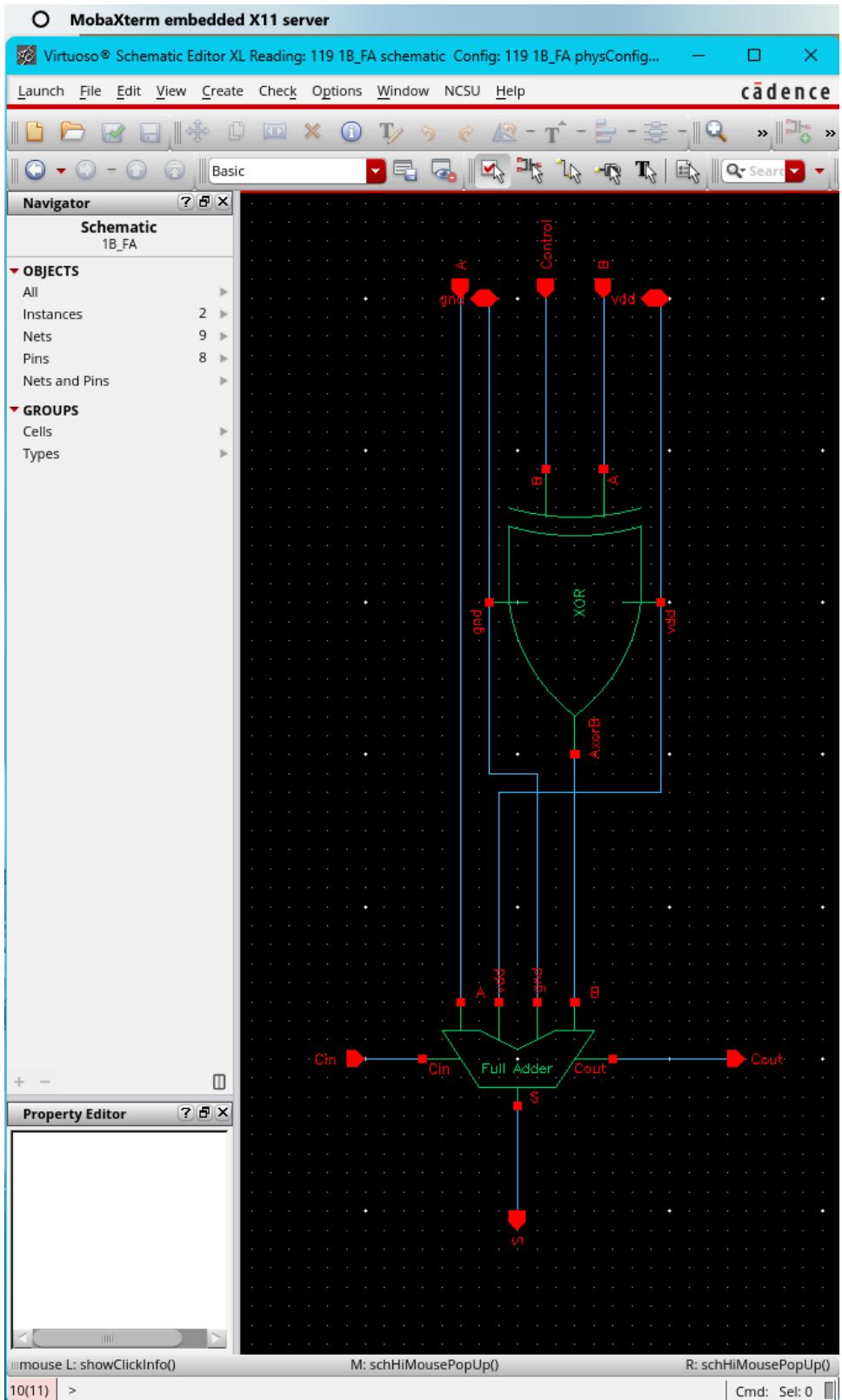


Figure 5: 1B FA Schematic

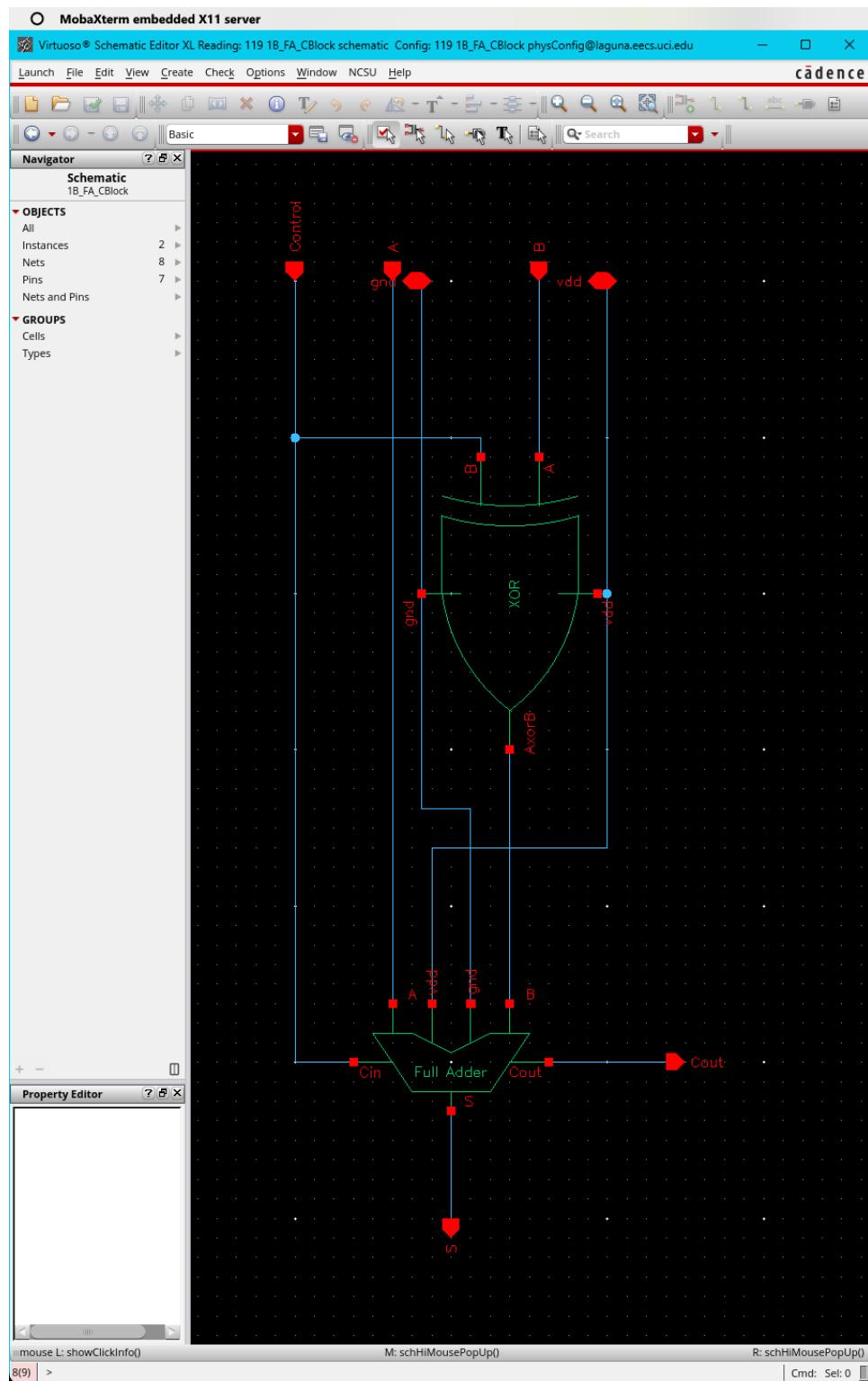


Figure 6: 1B FA with CBlock Schematic

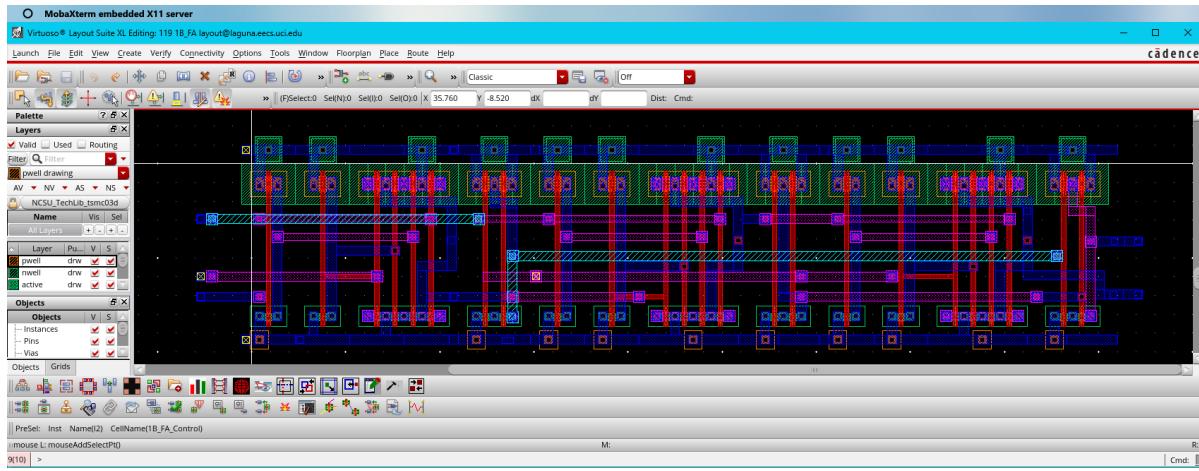


Figure 7: 1B FA Layout



Figure 8: 1B FA with CBlock Layout

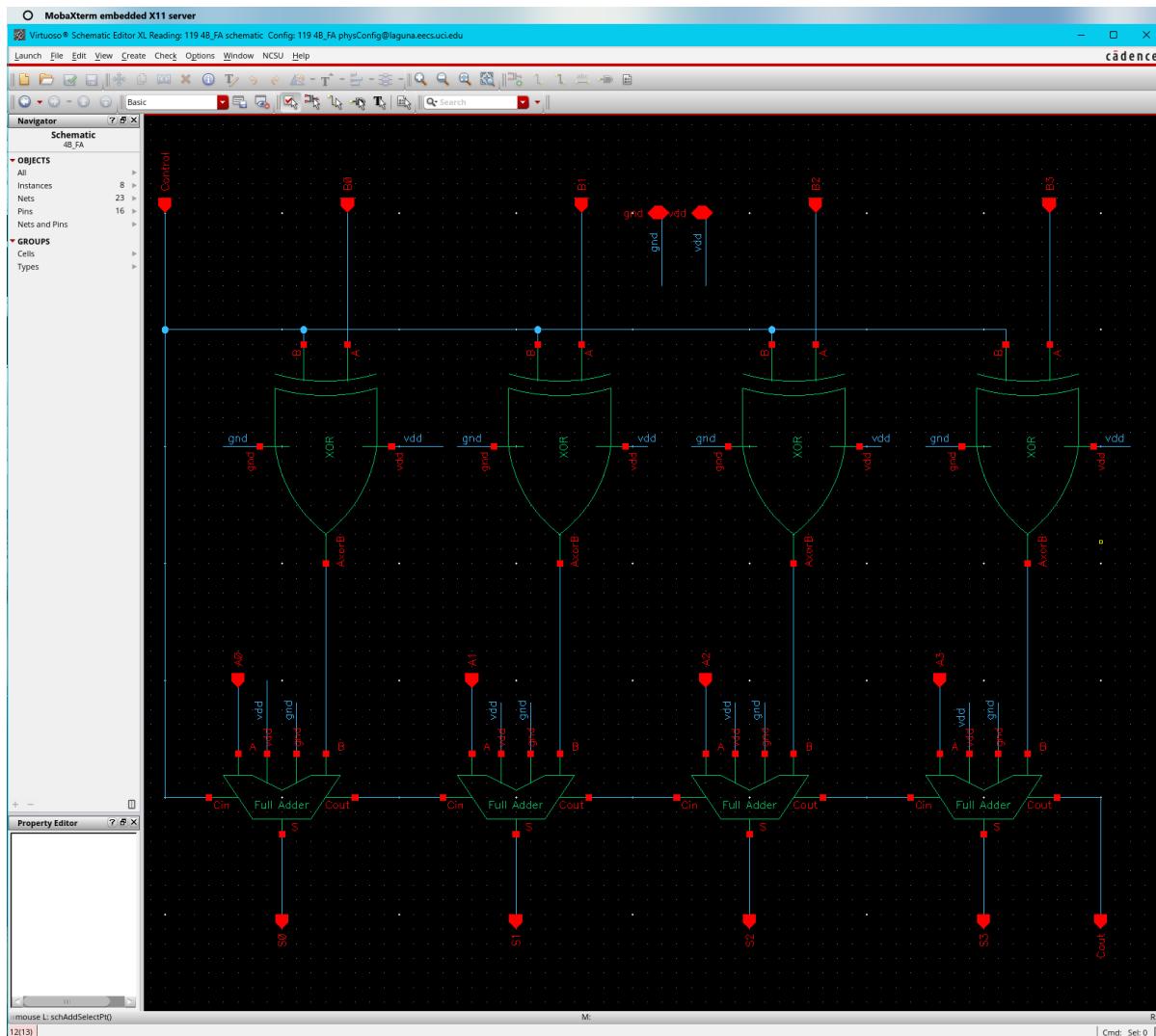


Figure 9: 4B FA Schematic

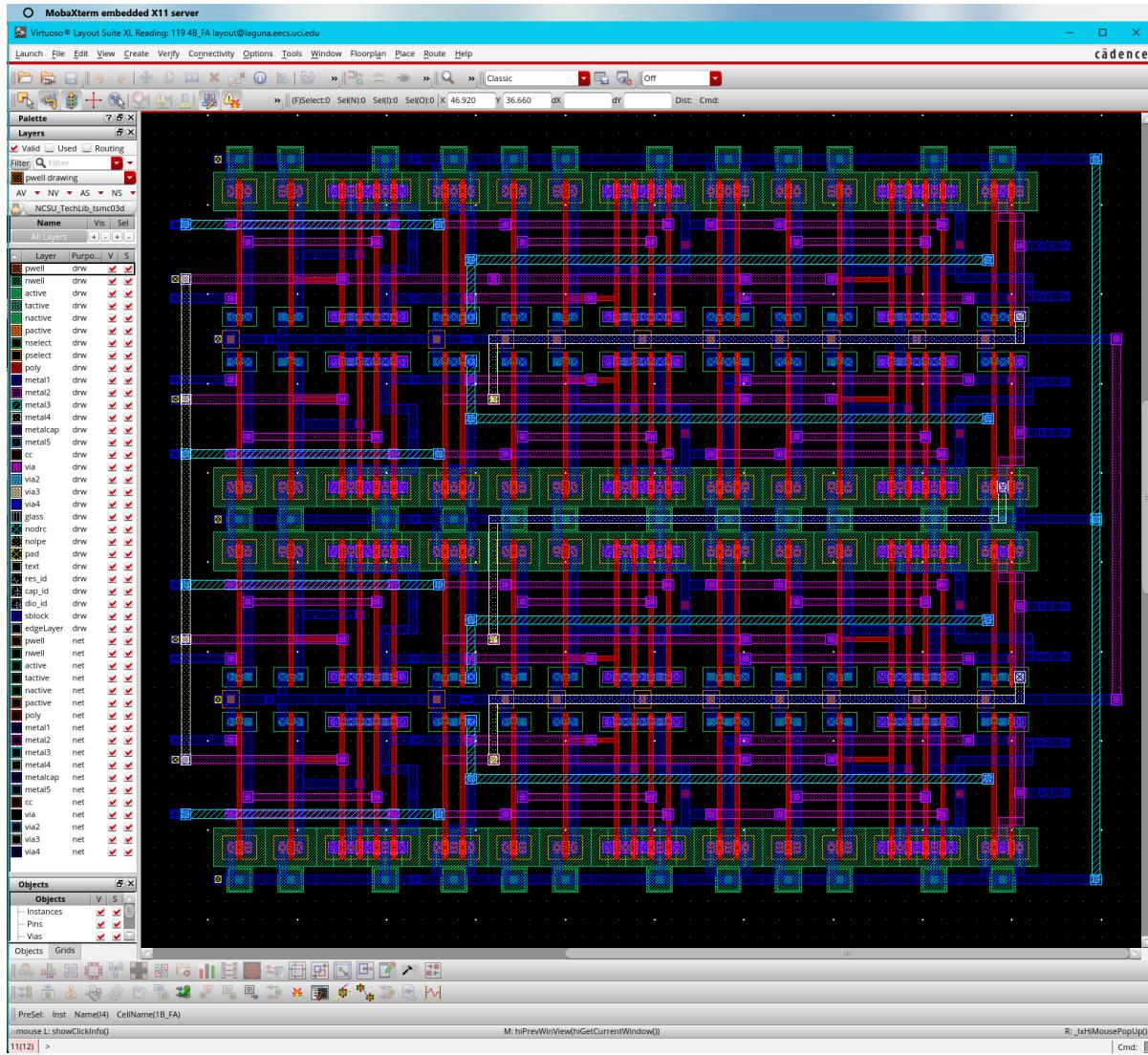


Figure 10: 4B FA Layout

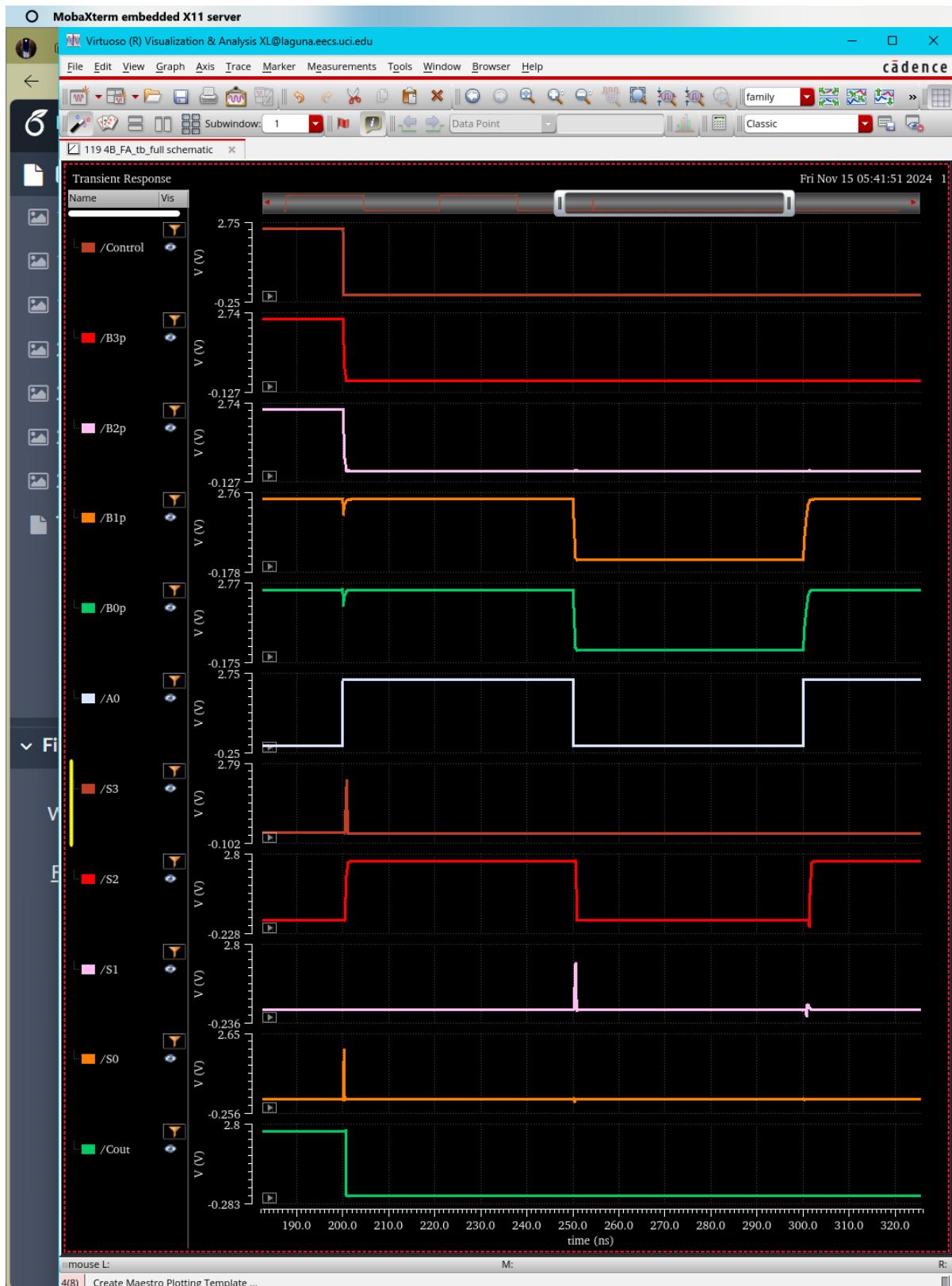


Figure 11: Schematic Simulation of $1+3=4$

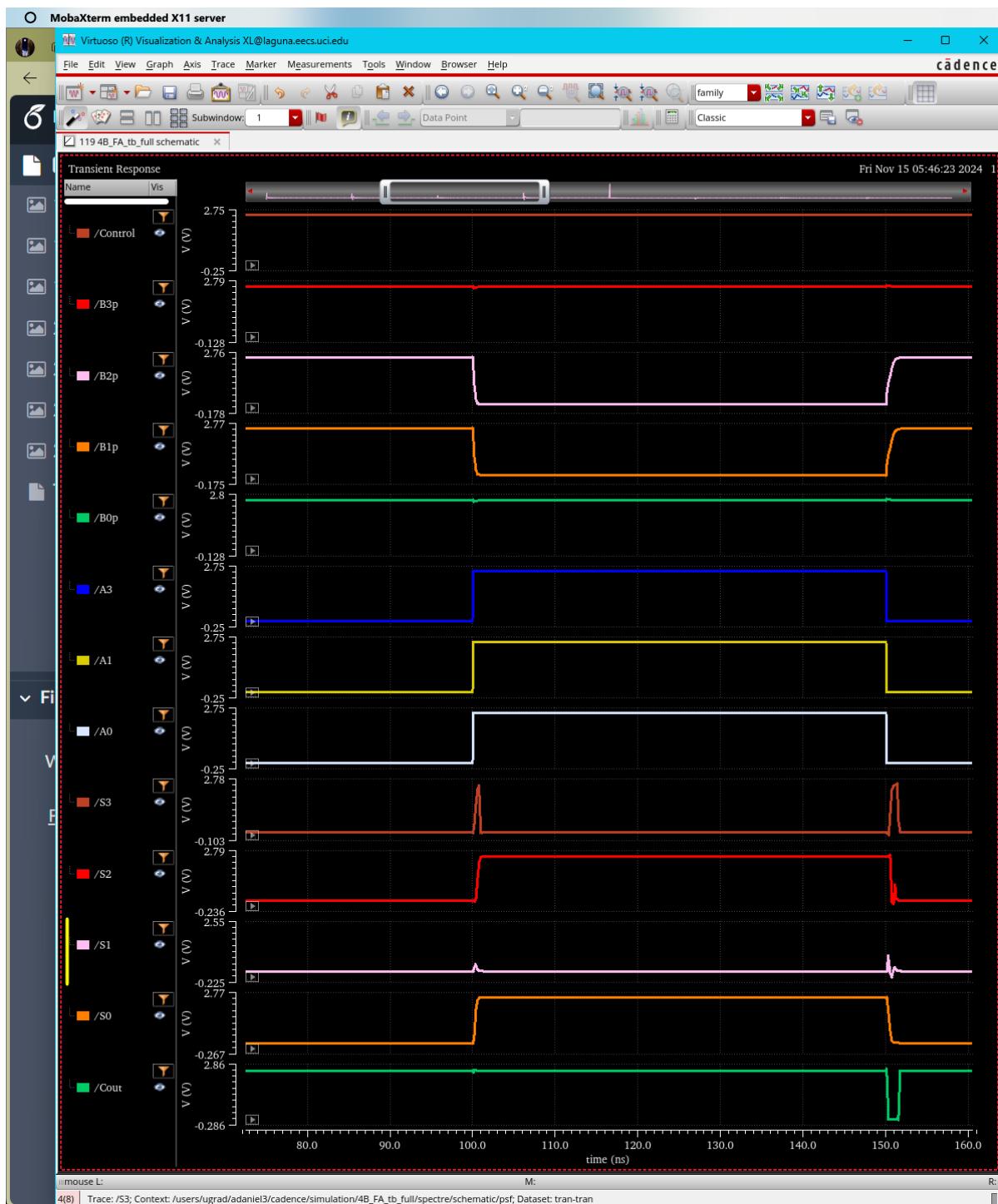


Figure 12: Schematic Simulation of 11-6=5

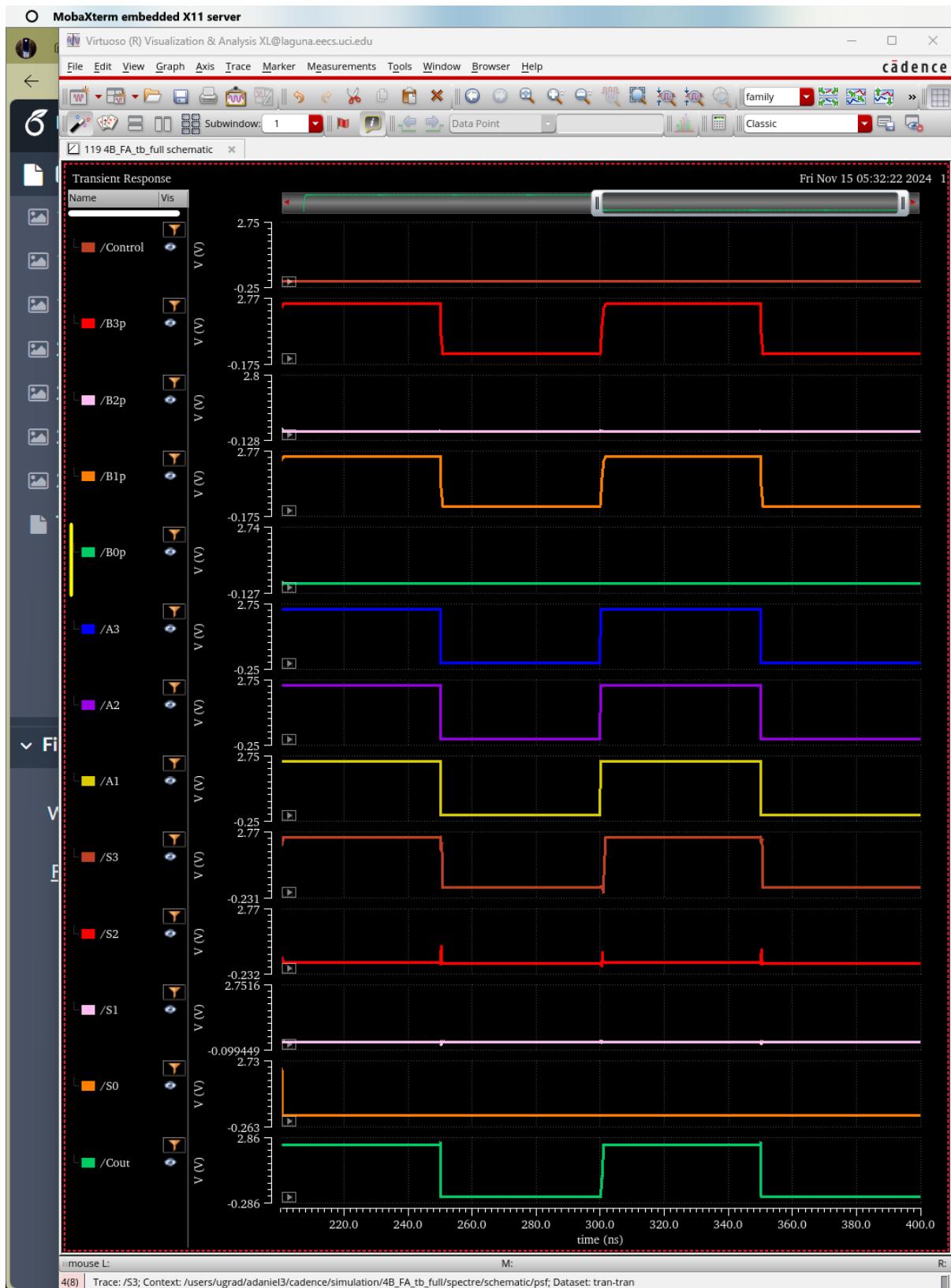


Figure 13: Schematic Simulation of $14+10=24$ with Cout HIGH

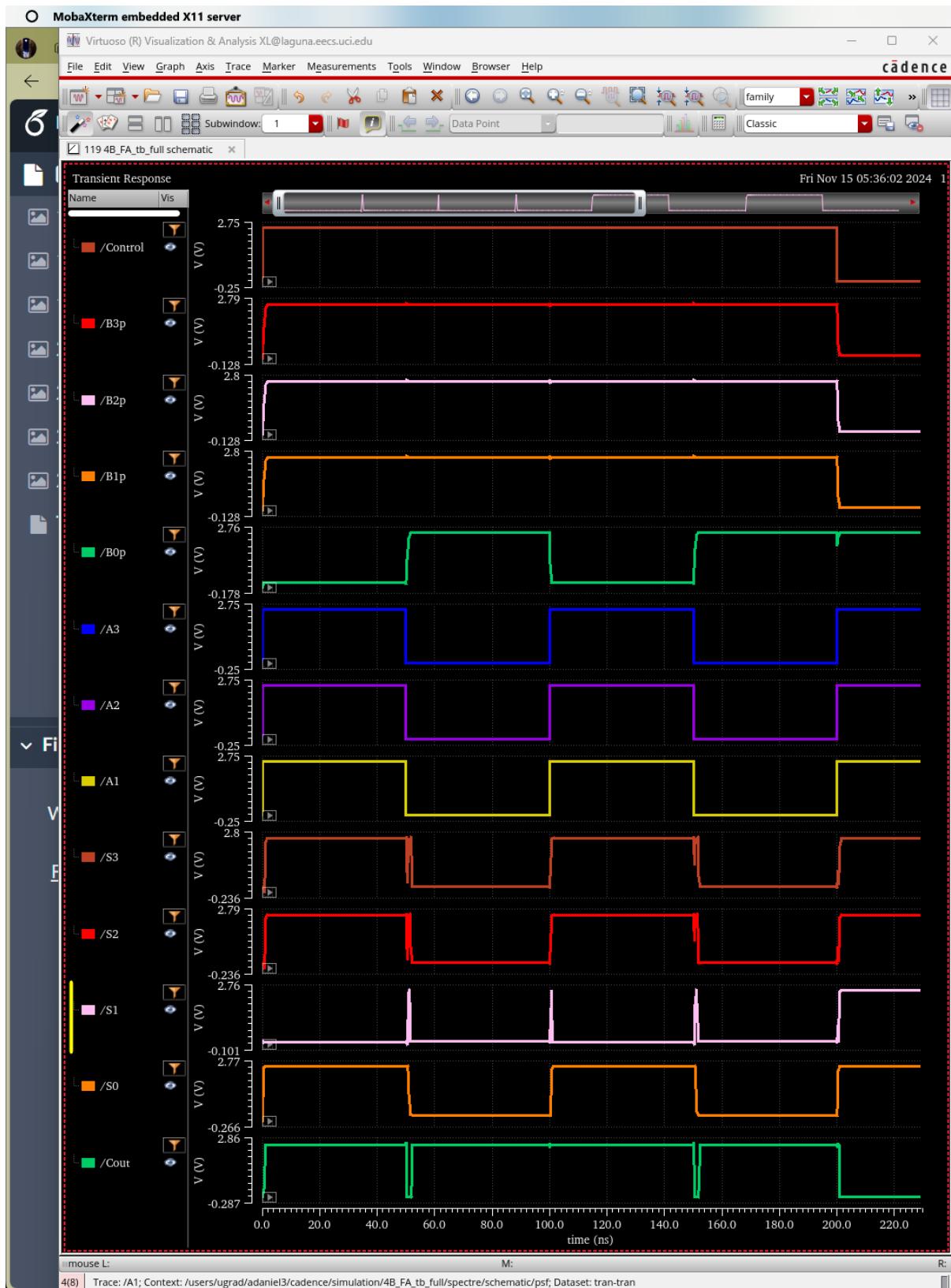


Figure 14: Schematic Simulation of 14-1=13

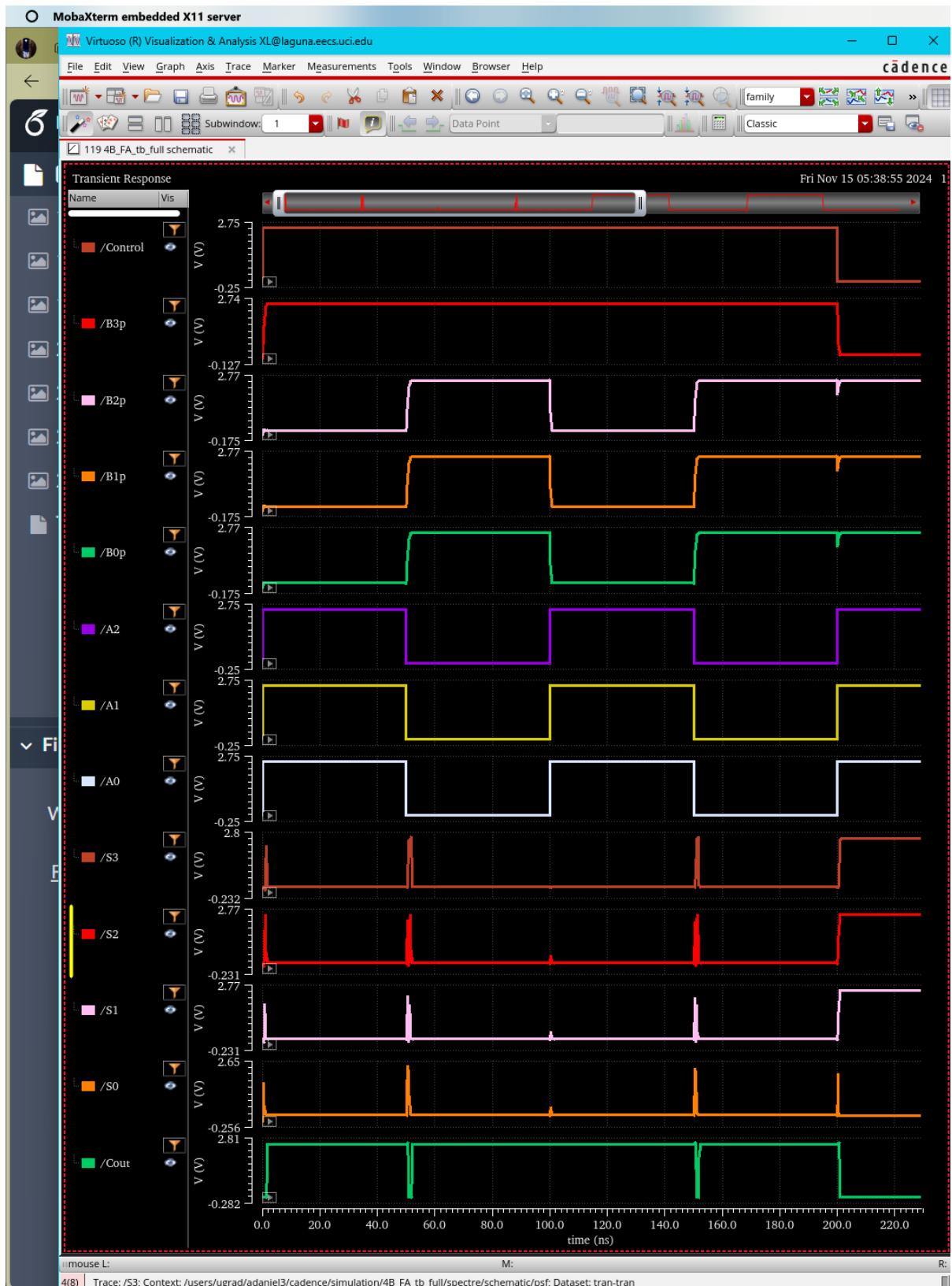


Figure 15: Schematic Simulation of 7-7=0
Page 16

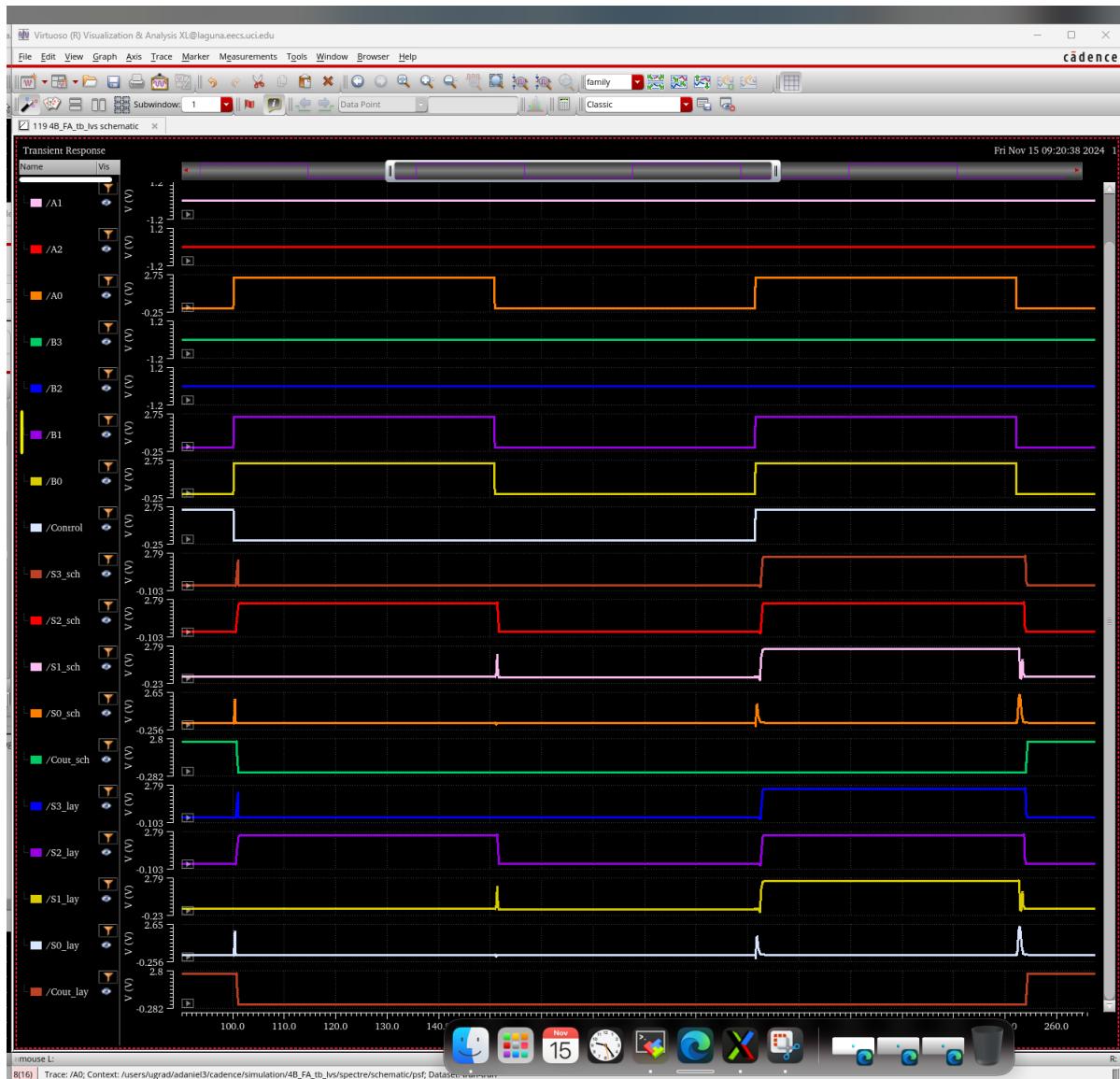


Figure 16: Layout Simulation of $1+3=4$

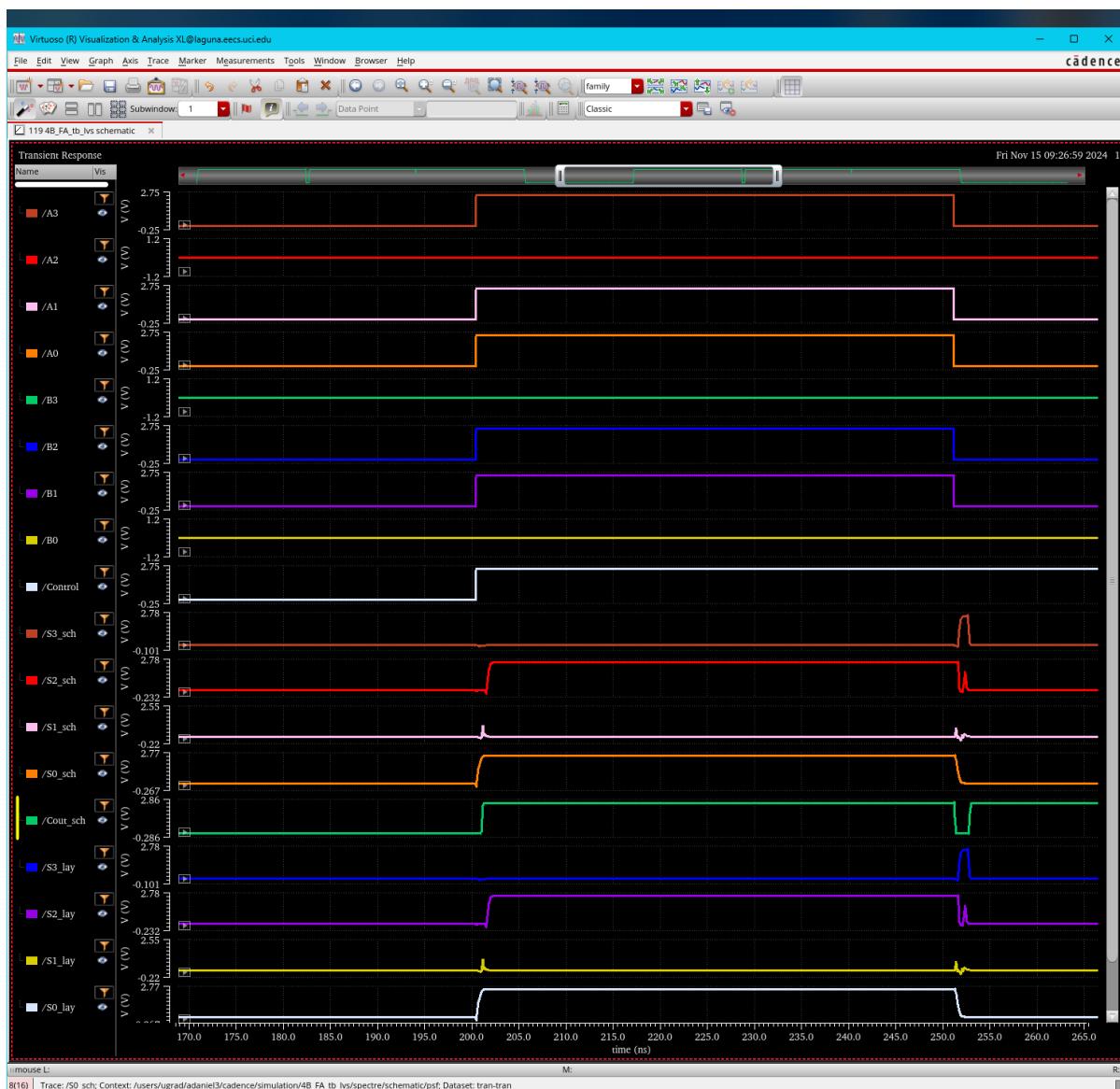


Figure 17: Layout Simulation of 11-6=5

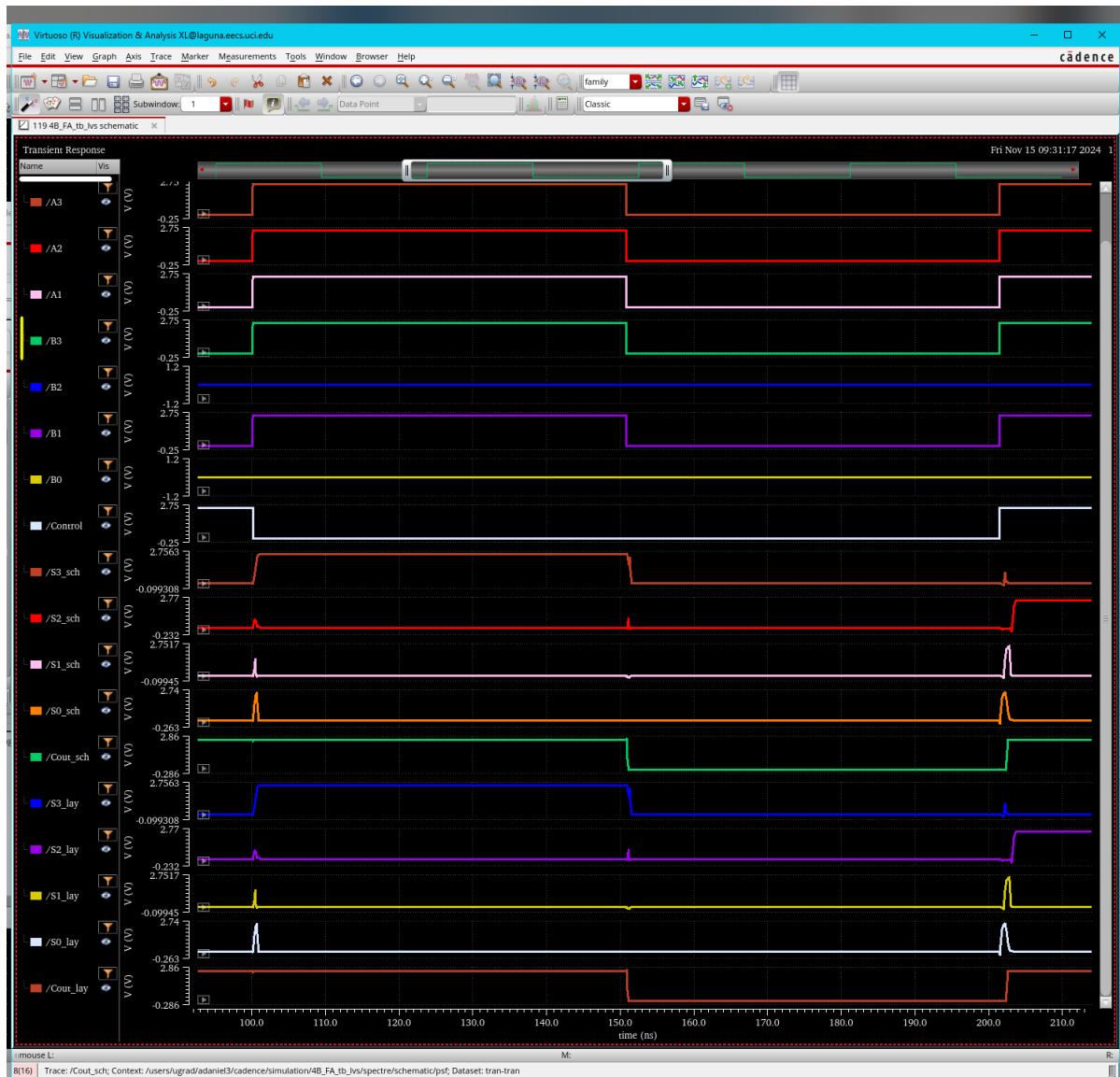


Figure 18: Layout Simulation of $14+10=24$ with Cout HIGH

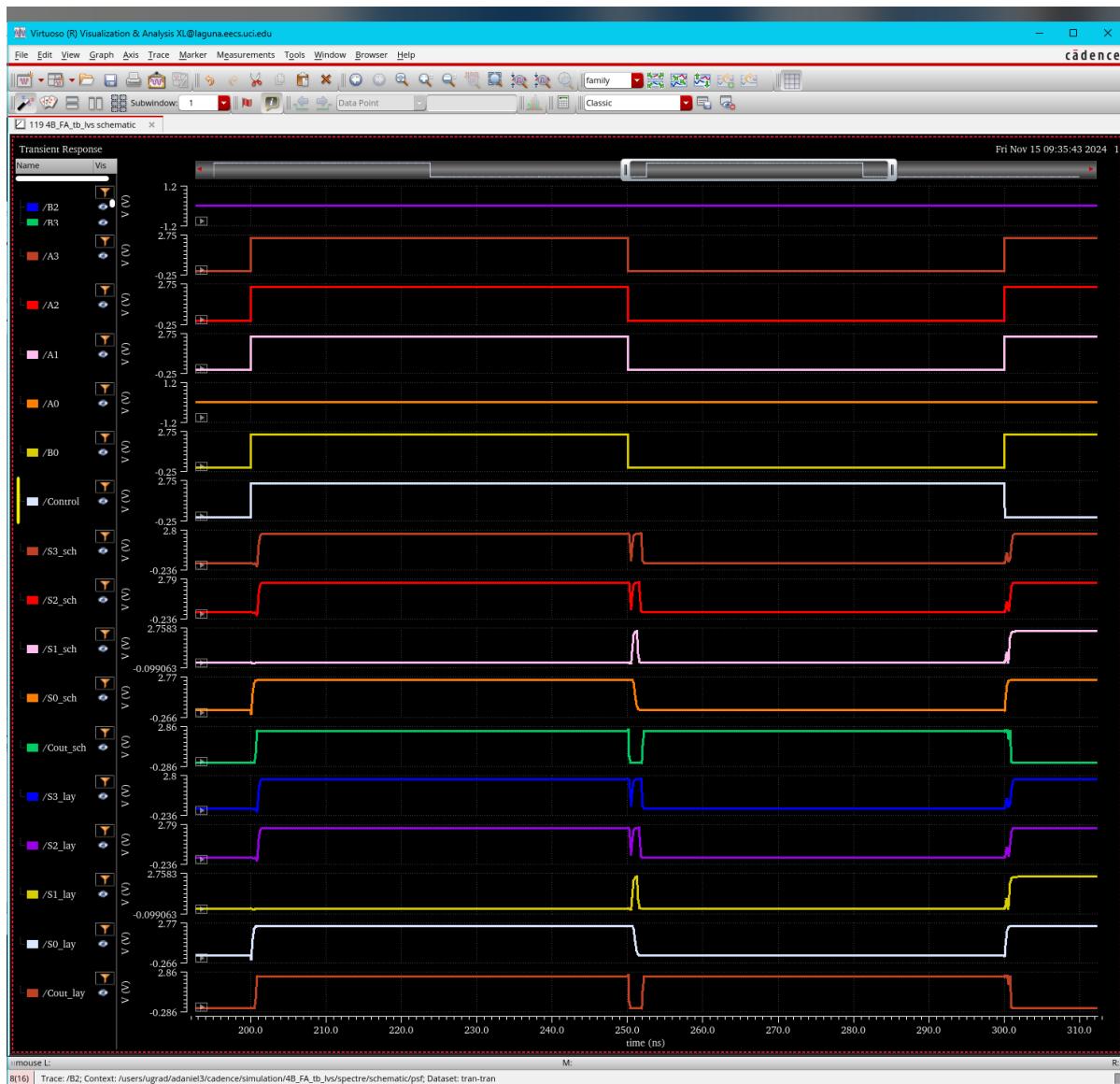


Figure 19: Layout Simulation of 14-1=13

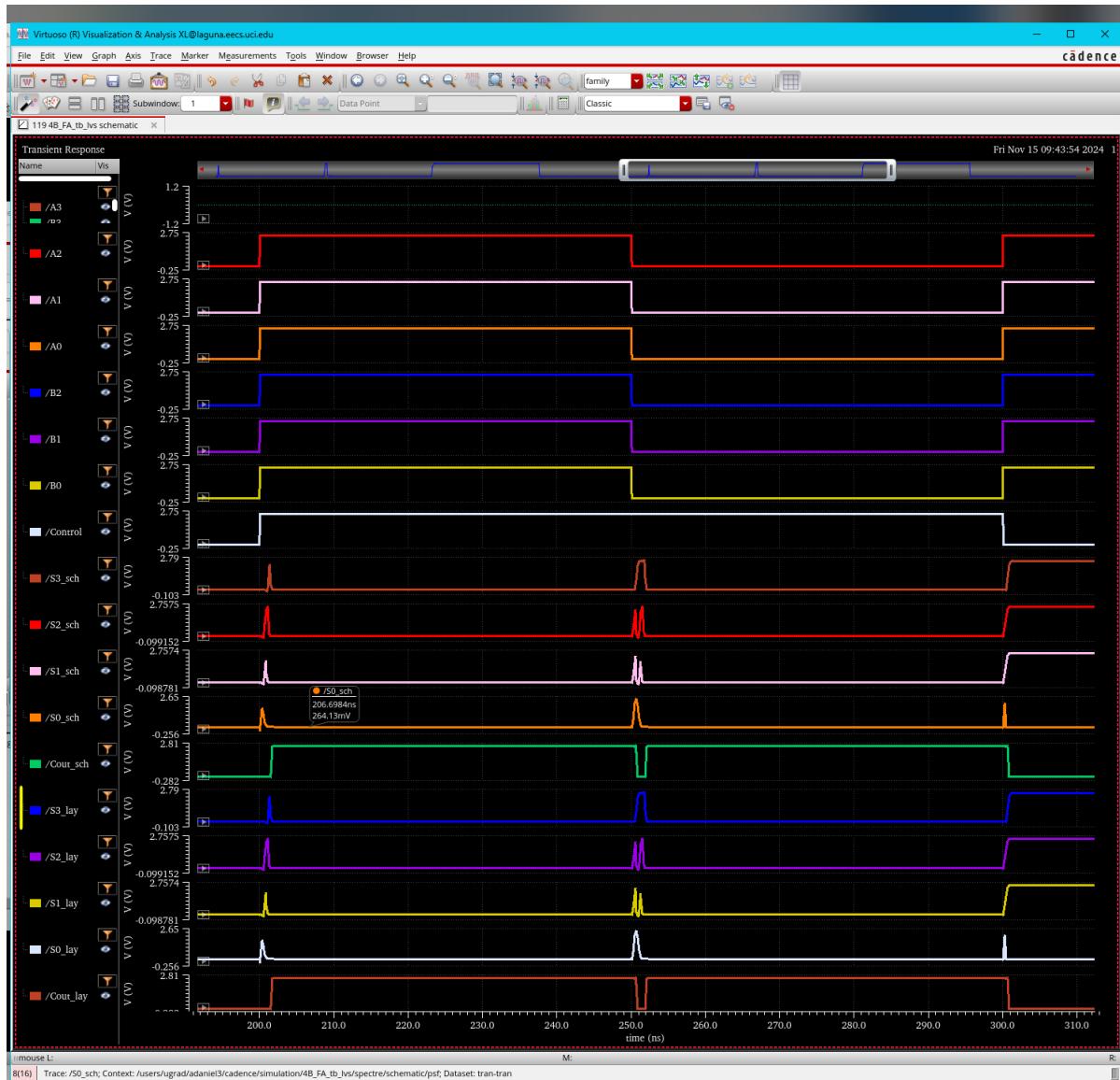


Figure 20: Layout Simulation of 7-7=0

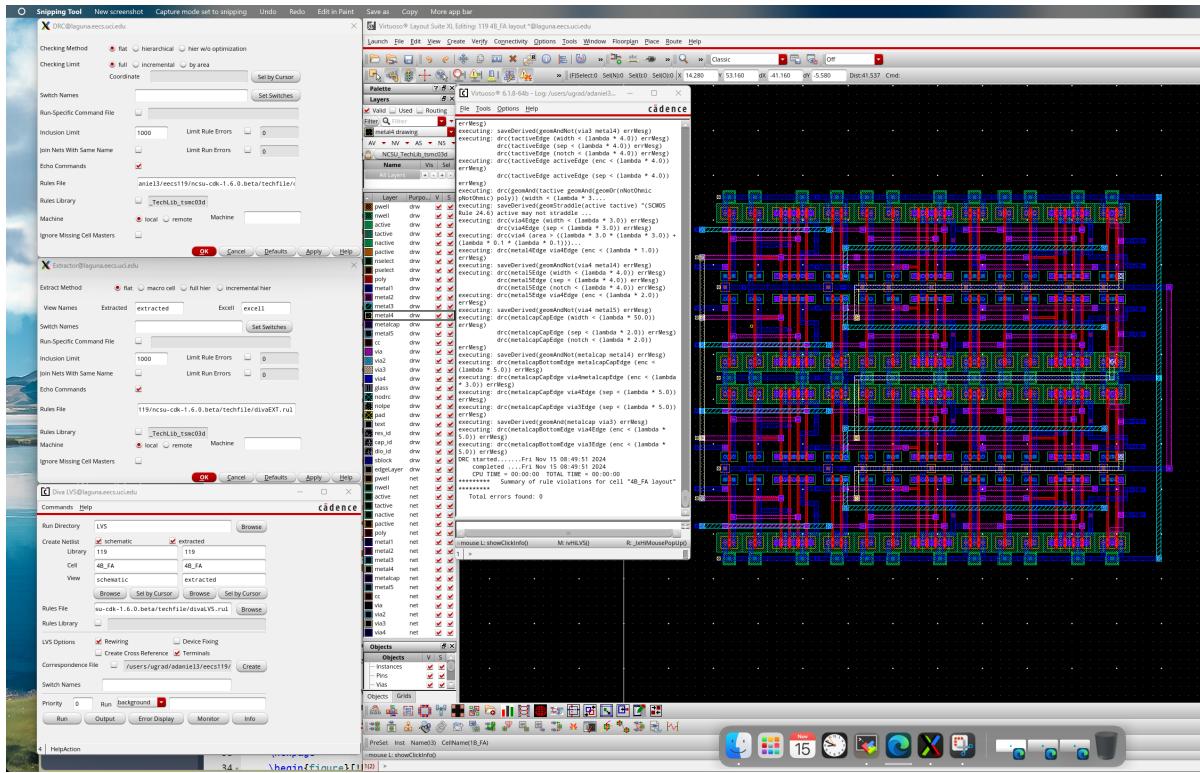


Figure 21: Picture of Successful DRC with the 4 Bit Full Adder

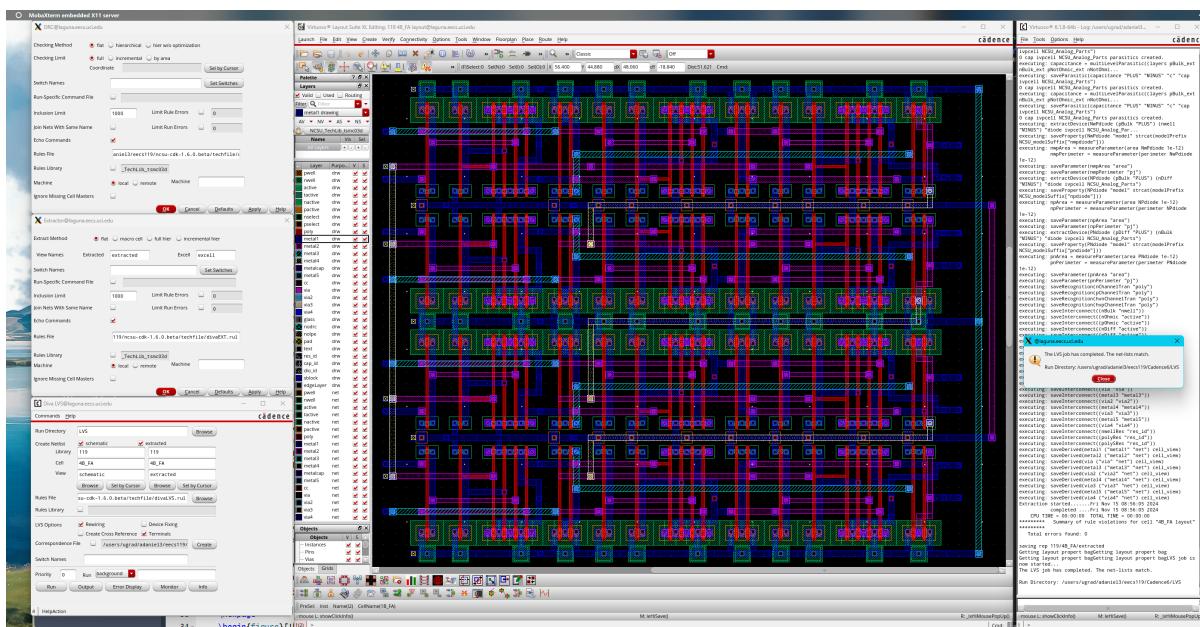


Figure 22: Picture of Successful LVS with the 4 Bit Full Adder

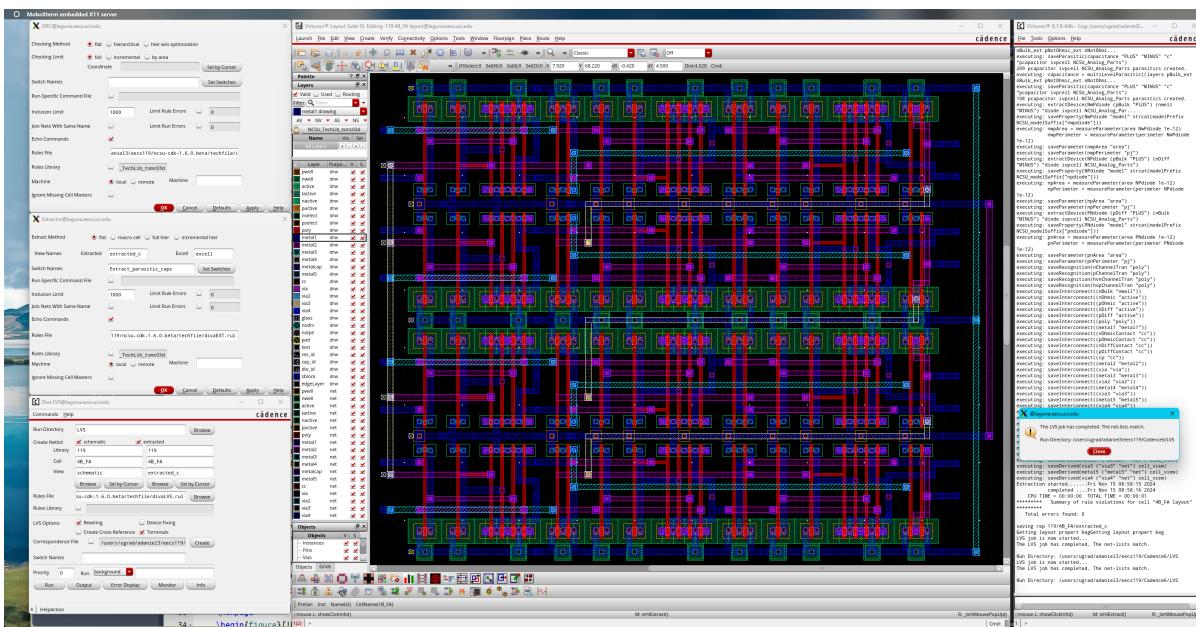


Figure 23: Picture of Successful LVS with Parasitics using the 4 Bit Full Adder

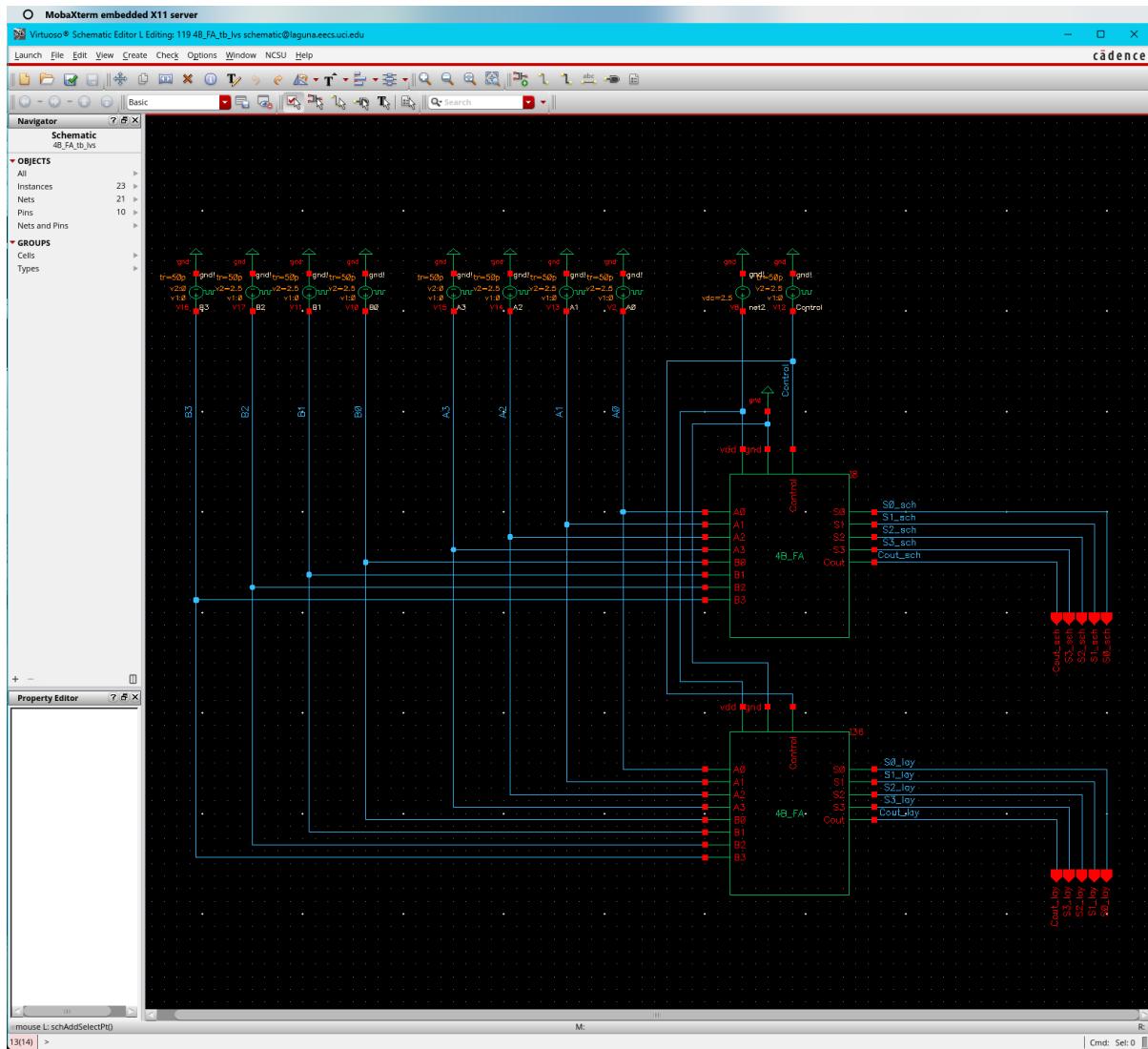


Figure 24: Picture of the testbench for the LVS Schematic for testing the layout against schematic