UCI
EECS 119
Project # 3
4-bit Full Adder/Subtractor Prepared by

1. Noah Mathew (mathewn2@uci.edu)
2. Artin Tamaddon (atamaddo@uci.edu)

Introduction:

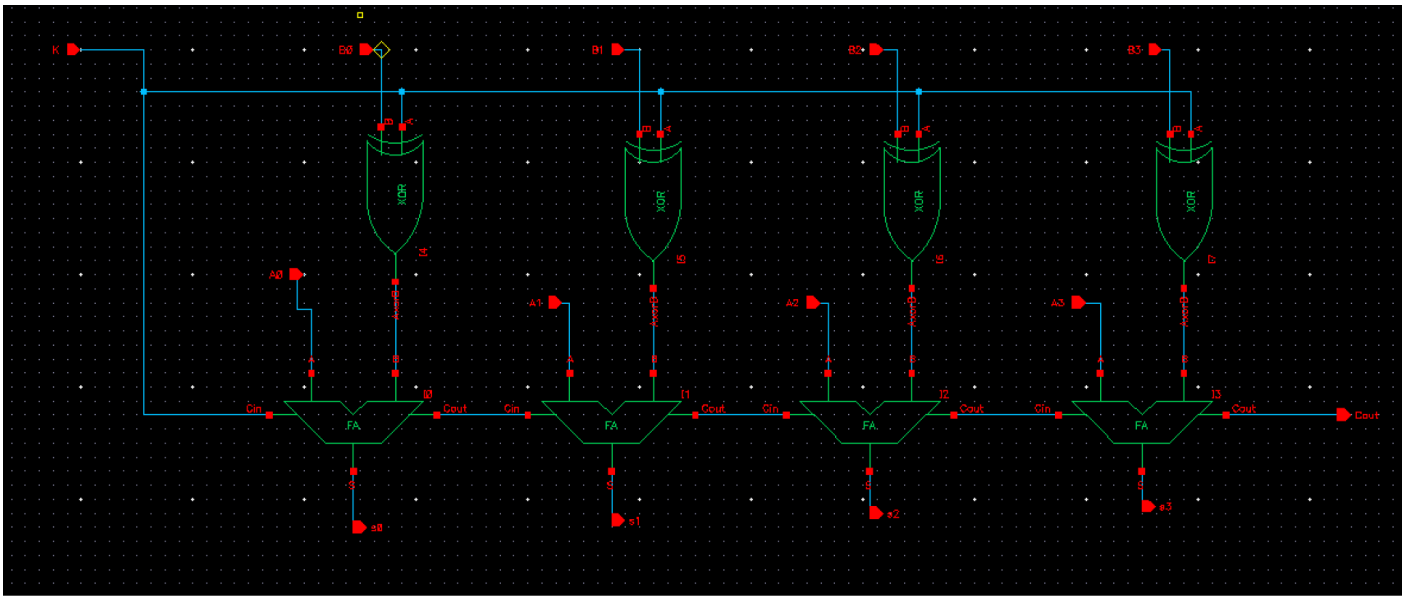
The objective of this project is to design, implement, and verify a hierarchical 4-bit full adder using CMOS logic. This includes creating transistor-level schematics for a 1-bit full adder, generating corresponding layouts using proper design rules, and instantiating four copies to construct a complete 4-bit ripple-carry adder. The project reinforces skills in schematic capture, layout design, hierarchical cell creation, LVS/DRC verification, and multi-bit digital circuit integration.

Theory:

A full adder is a fundamental combinational logic block that computes the sum of two input bits and an incoming carry using complementary CMOS logic, where PMOS devices form the pull-up network and NMOS devices form the pull-down network to provide full rail-to-rail voltage swing and minimal static power consumption. A one-bit full adder is typically implemented using XOR and NAND-based structures: the XOR network generates the partial sum A XOR B, while additional gates combine this value with the carry in to produce the final sum and carry out. Because XOR gates require more stacked transistors than simple inverters or NAND gates, they introduce greater propagation delay, and in a ripple-carry architecture these delays accumulate linearly across all stages, making the worst-case delay proportional to the number of bits. Noise margins and switching thresholds remain stable due to the complementary nature of the CMOS logic, preserving signal integrity as each stage processes its inputs. At the layout level, each full-adder cell is constructed using matched PMOS and NMOS transistors with proper well spacing, diffusion sizing, and routing to meet design rules, while parasitic capacitances from interconnects and vias introduce small deviations from ideal theoretical delay. By designing and verifying a single full-adder cell and then instantiating it hierarchically four times, the resulting four-bit ripple-carry adder achieves consistent device geometry, predictable timing behavior, and correct functionality as confirmed through DRC and LVS verification.

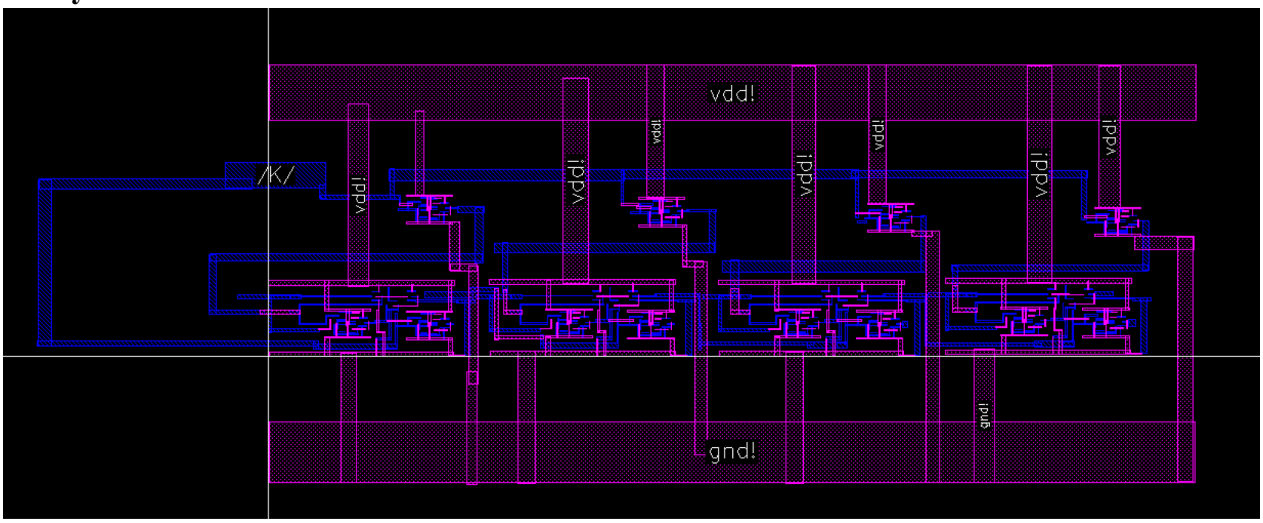
Design: (Simulation Layout Questions)

1. Circuit Schematic



Circuit Schematic of 4-bit Full Adder

2. Layout



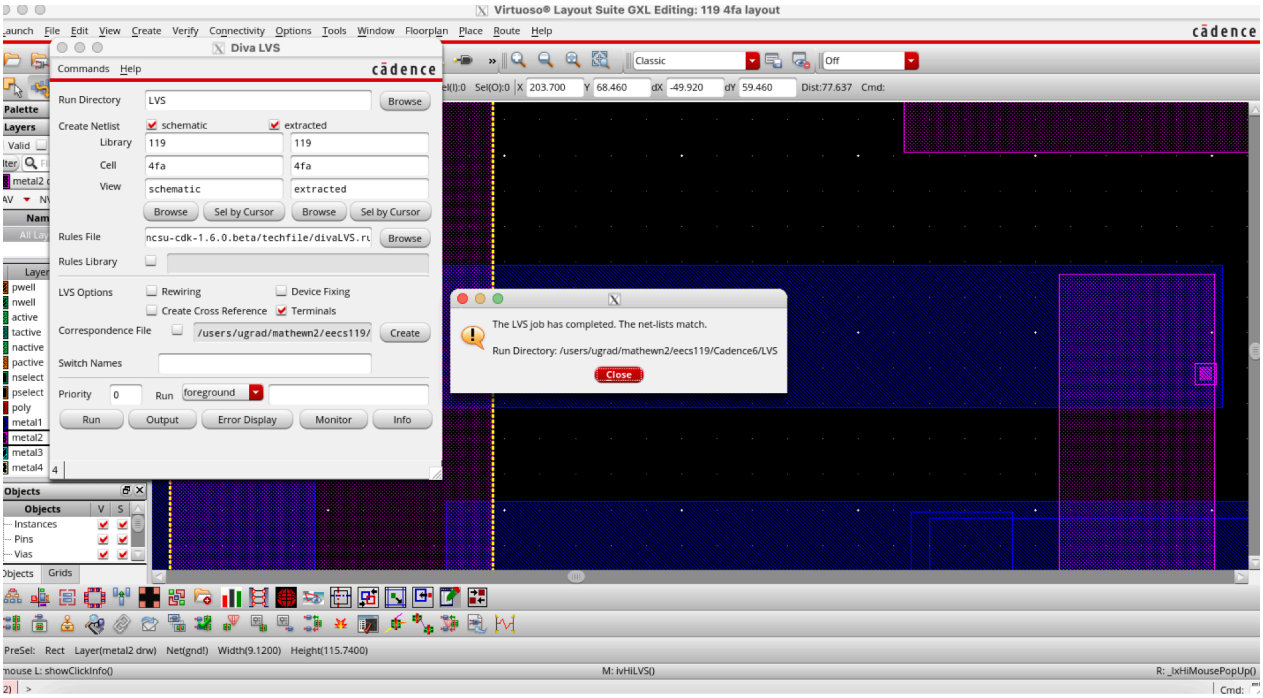
4-bit Full Adder Layout

3. Layout DRC & LVS match

```
executing: drc(metalcapCapEdge via3Edge (sep < (lambda * 5.0)) errMesg)
executing: saveDerived(geomAnd(metalcap via3) errMesg)
executing: drc(metalcapBottomEdge via4Edge (enc < (lambda * 5.0)) errMesg)
executing: drc(metalcapBottomEdge via3Edge (enc < (lambda * 5.0)) errMesg)
DRC started.......Wed Nov 19 03:33:49 2025
    completed ....Wed Nov 19 03:33:50 2025
    CPU TIME = 00:00:00  TOTAL TIME = 00:00:01
*********   Summary of rule violations for cell "4fa layout"   *********
   Total errors found: 0
```

mouse L: showClickInfo() M: ivHiDRC() R: _lxHiMousePopUp()

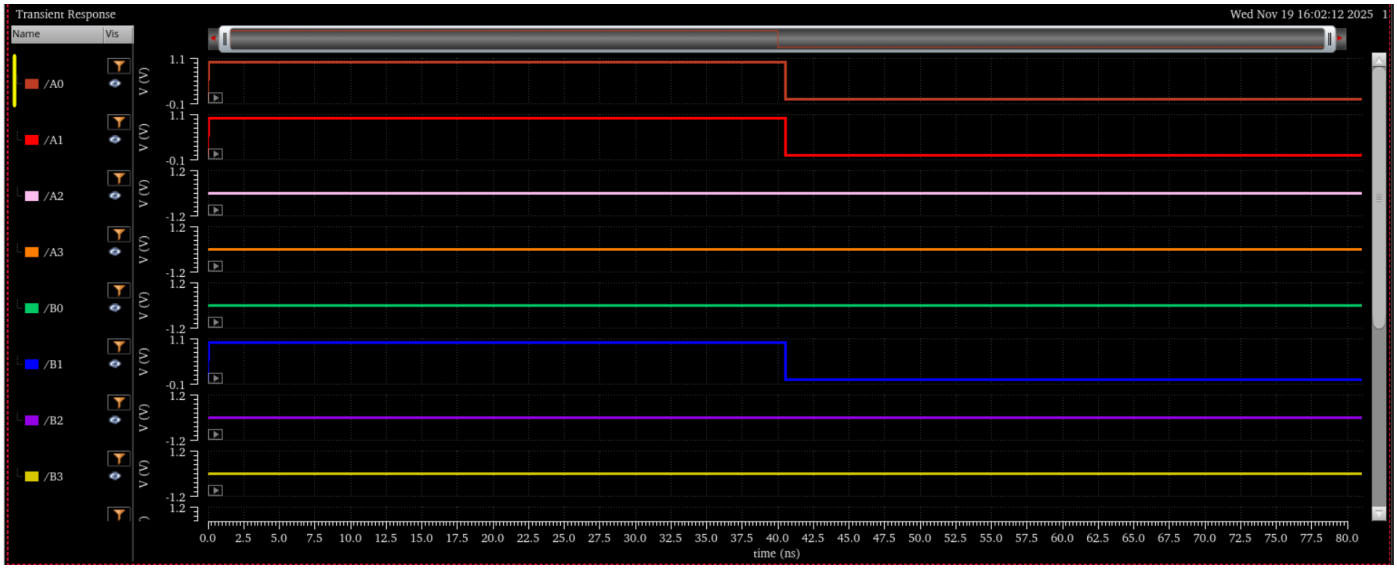
No DRC Errors



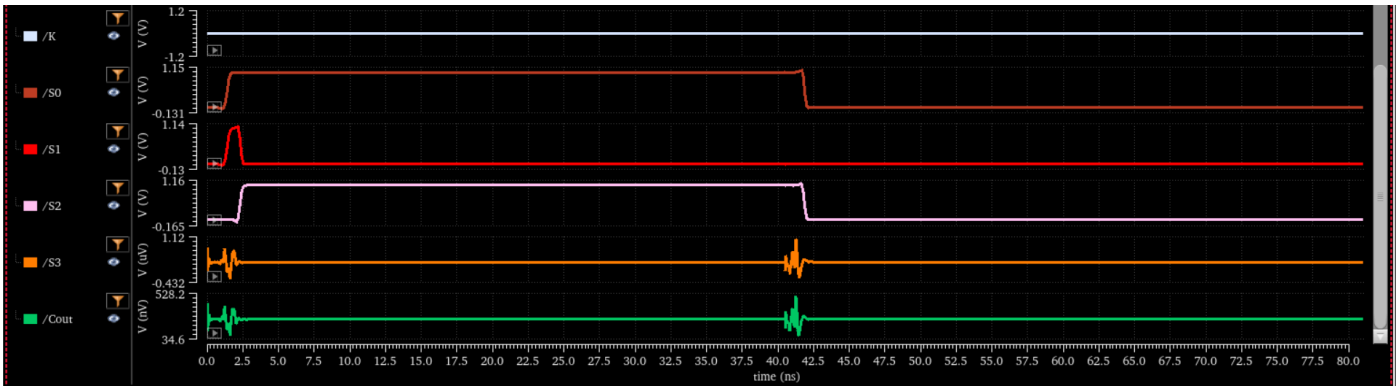
LVS Screenshot

4 & 5.Simulation

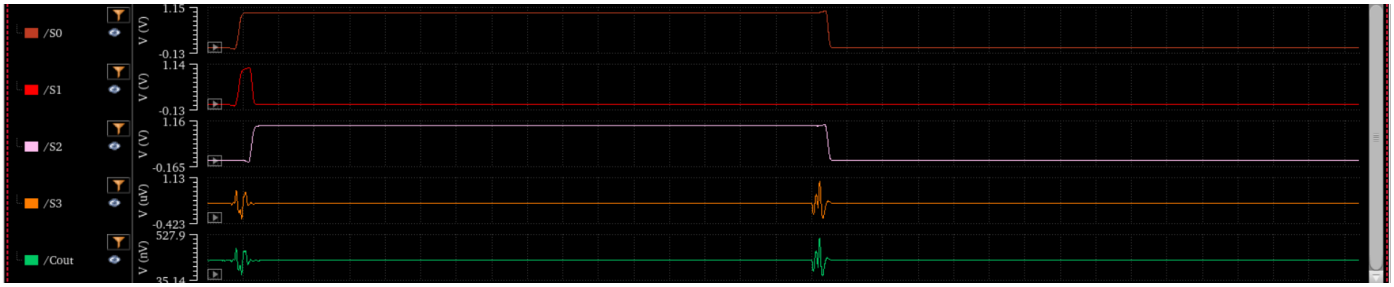
1.3+2=5
A0A1A2A3
3=1100
B0B1B2B3
2=0100
S0S1S2S3
5=1010



Inputs (A&B)

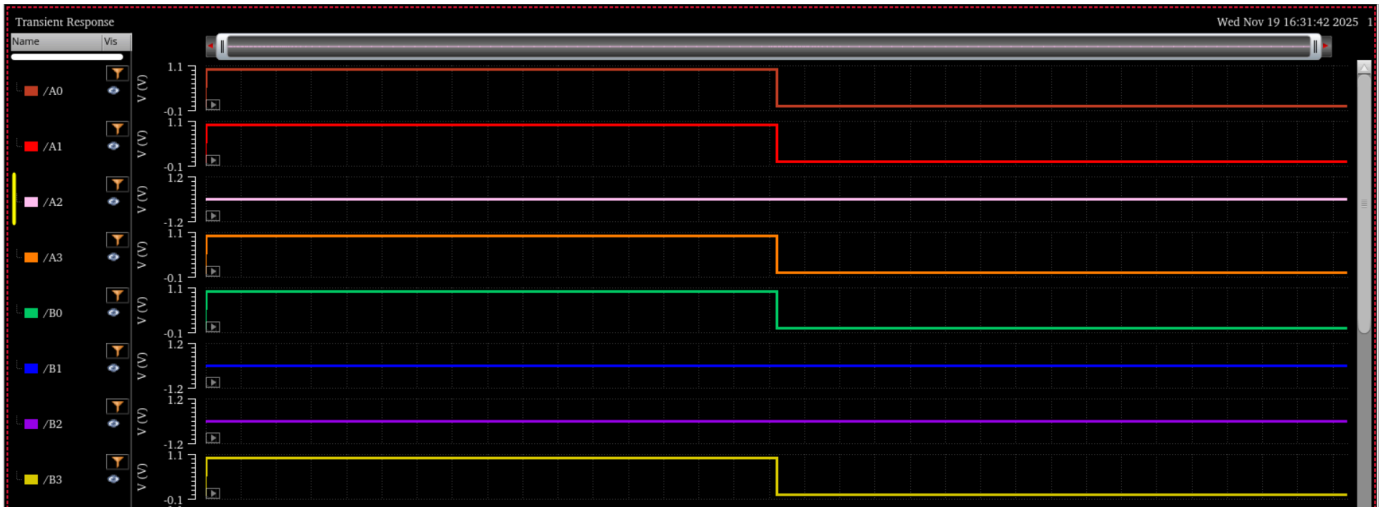


Simulation (Schematic Output)

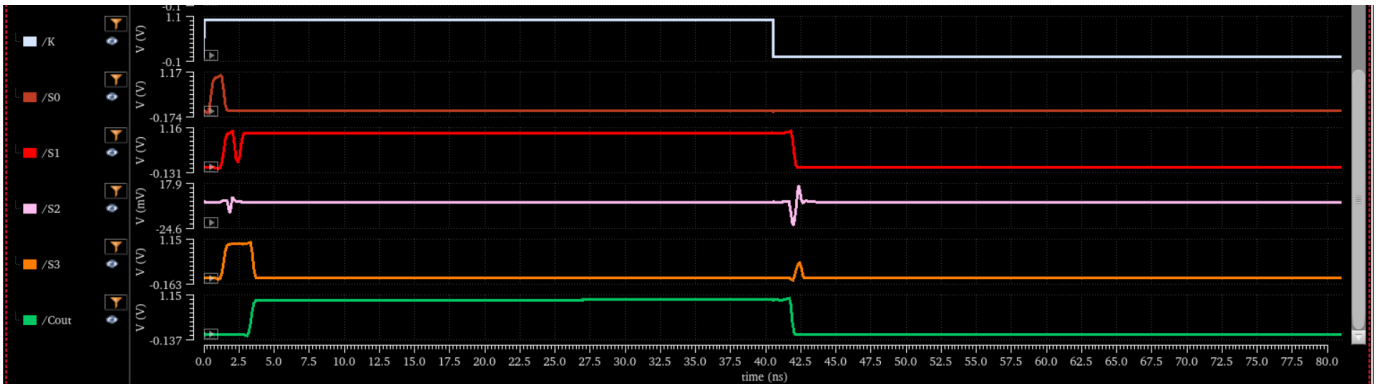


Simulation (Post Layout Output)

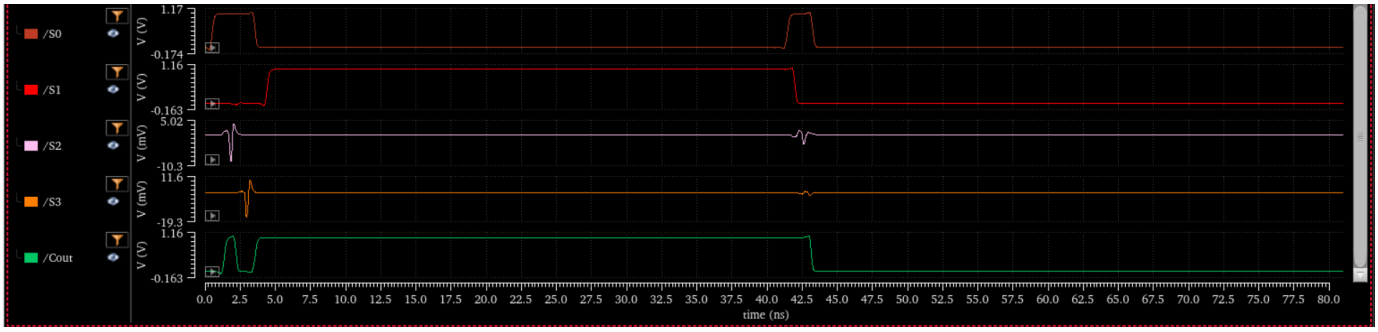
2.11-9=2
A0A1A2A3
11=1101
B0B1B2B3
9=1001
S0S1S2S3
2=0100



Inputs (A&B)

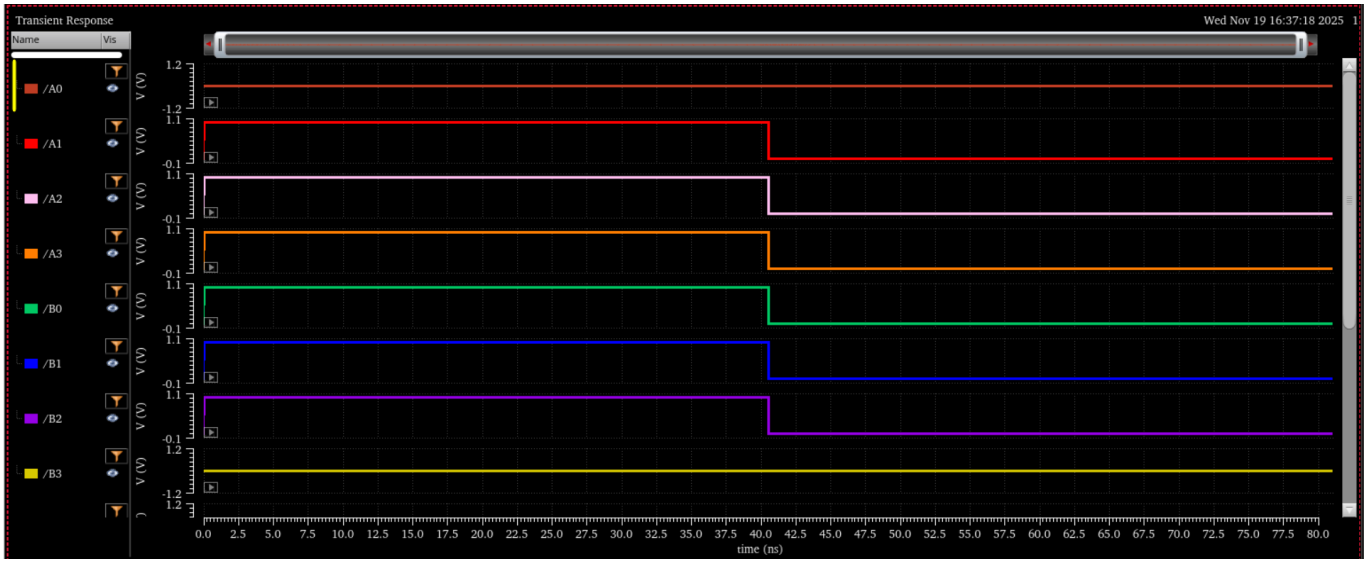


Simulation (Schematic Output)

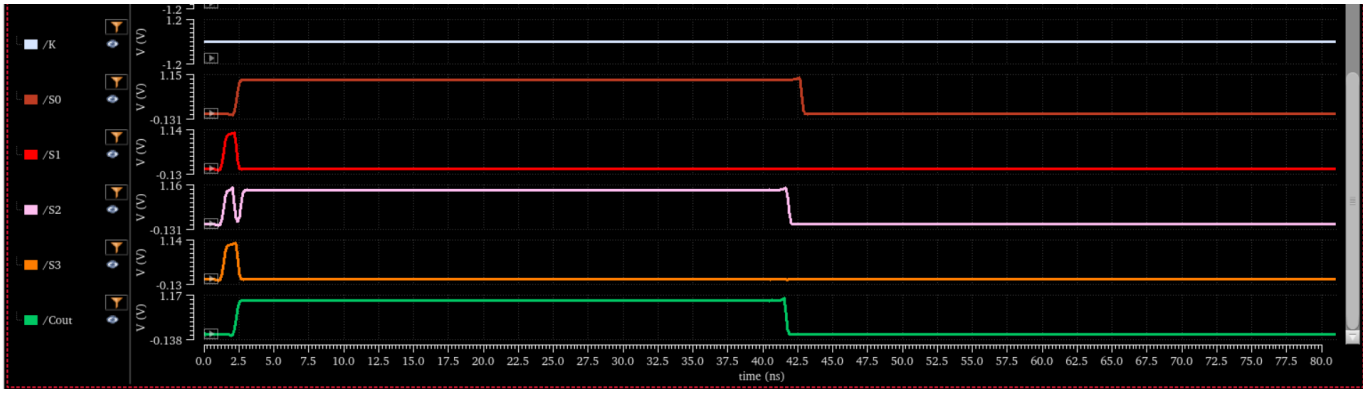


Simulation (Post Layout Output)

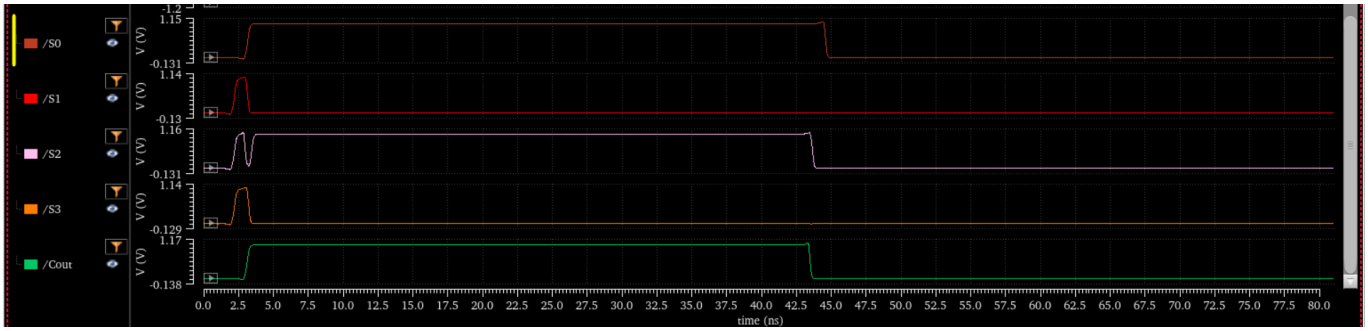
3. 14+7=21
A0A1A2A3
14=0111
B0B1B2B3
7=1110
S0S1S2S3Cout
21=10101



Inputs (A&B)

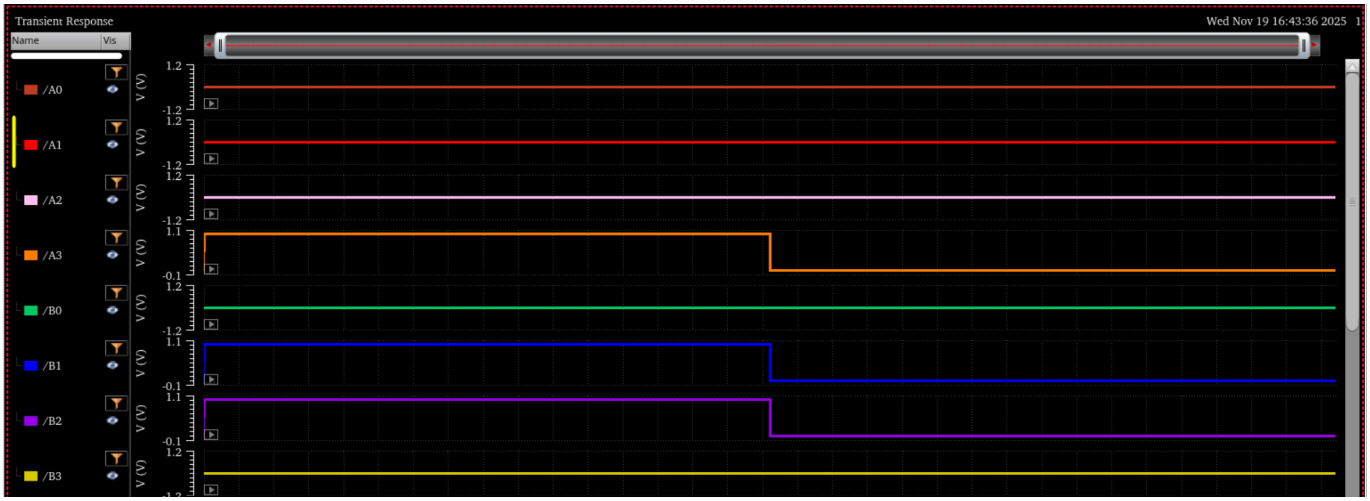


Simulation (Schematic Output)

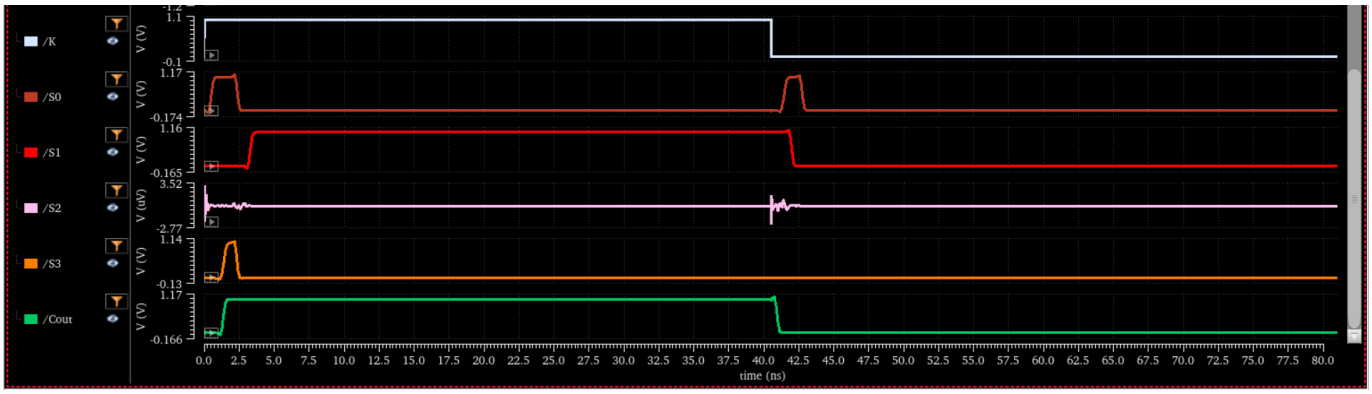


Simulation (Post Layout Output)

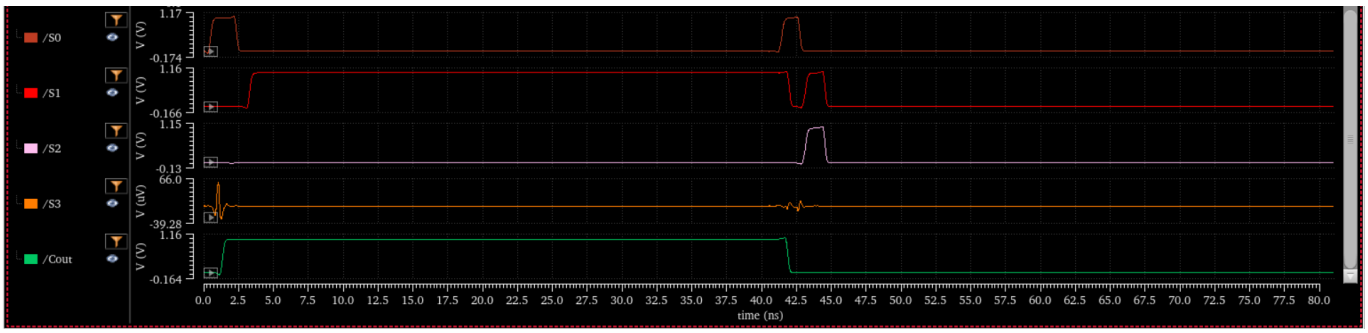
4. 8-6=2
A0A1A2A3
8=0001
B0B1B2B3
6=0110
S0S1S2S3Cout
2=0100



Inputs (A&B)

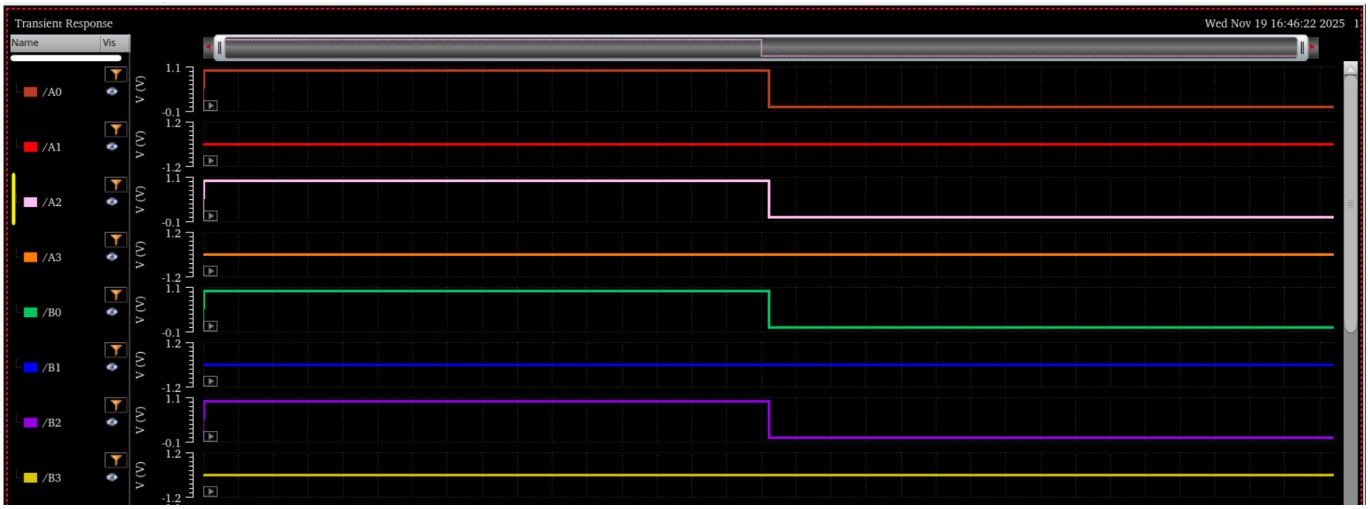


Simulation (Schematic Output)

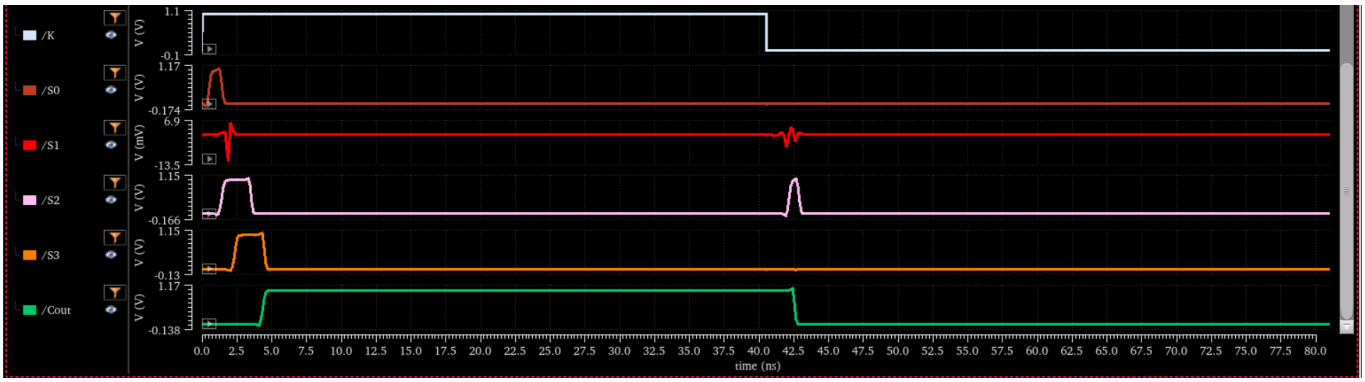


Simulation (Post Layout Output)

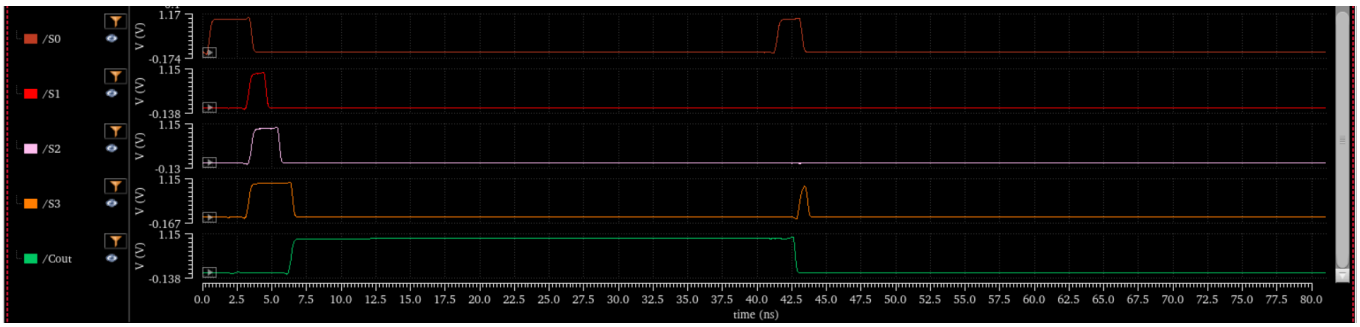
5. 5-5=0
A0A1A2A3
5=1010
B0B1B2B3
5=1010
S0S1S2S3Cout
0=0000



Inputs (A&B)



Simulation (Schematic Output)



Simulation (Post Layout Output)

Conclusion:

In this project, we extended our CMOS design workflow to construct, simulate, and verify a one-bit full adder and a hierarchical four-bit ripple-carry adder using Cadence. By first developing and LVS-verifying the required building blocks—including inverters, NAND gates, and XOR-based logic—we established a reliable foundation for assembling larger combinational circuits. The hierarchical design of the four-bit adder demonstrated how propagation delay accumulates across stages as the carry signal ripples from the least significant to the most significant bit, reflecting the theoretical timing limitations of ripple-carry architectures. Throughout the layout process, we gained practical experience managing transistor placement, well taps, routing congestion, and global power distribution, while observing how parasitic capacitances from interconnects affect signal behavior compared to schematic simulations. LVS verification reinforced the importance of consistent naming, proper pin definitions, and careful net connectivity when integrating multiple subcells. Overall, this lab strengthened our understanding of how elementary CMOS logic gates scale into multi-bit arithmetic units, highlighting the critical relationship between circuit theory, device-level implementation, and physical layout considerations.

Bibliography:

[1] R. Jacob Baker (2019), *CMOS: Circuit Design, Layout, and Simulation*, 4th Edition, Wiley-IEEE Press.

[2] Adel S. Sedra and Kenneth C. Smith (2020), *Microelectronic Circuits*, 8th Edition, Oxford University Press.