The University of Jordan

School of Engineering

Department of Computer Engineering

Majority-Controlled 3-Bit Counter Using 6T SRAM and D Flip-Flops

Digital Electronics Laboratory Project

Submitted by:

Batool Adawi 0214214

Raghad Almade 0219099

Sarah Shaheen 0207658

Instructor: Eng. Abeer Awad

Submission Date: August 25, 2025

Academic Year: 2024 / 2025

Contents

Introduction

In this project, we design a majority-controlled 3-bit counter using SRAM cells and D flip-flops. The system is built such that the counter only increments when a valid majority condition is met. Three binary values are stored in SRAM cells, then passed through D flip-flops to keep the timing synchronized. These outputs are fed into a majority logic circuit, which produces a high signal whenever at least two inputs are high. This majority signal is used as the trigger to increment the 3-bit counter by one.

The way we worked is by building simple gates, like AND, XOR, OR, NAND, NOR, to then build bigger components like the SRAM and D Flip Flop, as layouts and schematics, and test each component separately before combining it to form the whole system.

The Design

The top-level design integrates three 6T SRAM cells, D flip-flops, a majority voting circuit, and a 3-bit counter to realize a majority-controlled counting system. Each SRAM cell stores a binary value written at different intervals, representing the three system inputs. These outputs are then synchronized using D flip-flops driven by a shared clock, ensuring that all stored values are sampled at the same instant. The synchronized outputs feed into a majority logic block, which produces a high output only when at least two of the three inputs are high. This majority output is input for the 3-bit counter. If the majority condition is satisfied, the counter increments by one. Otherwise, the counter holds its current state.

We will now go through the components of the project, showing the schematic, layout and tests to prove they work as intended.

2- Input NAND Gate

Below is the schematic design, layout design, testing and propagation delay of the 2-input NAND gate.

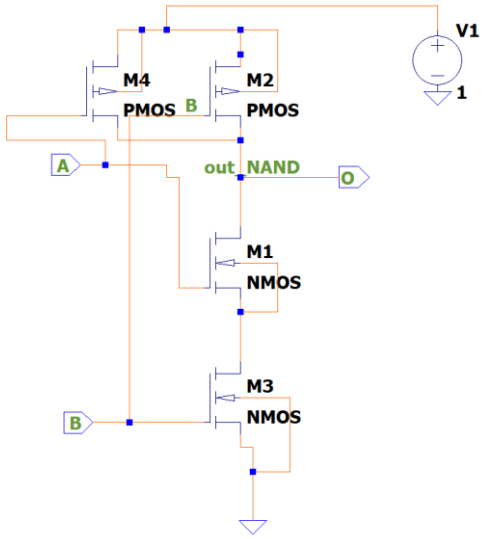

Figure 1. The schematic design of the 2 input NAND gate.

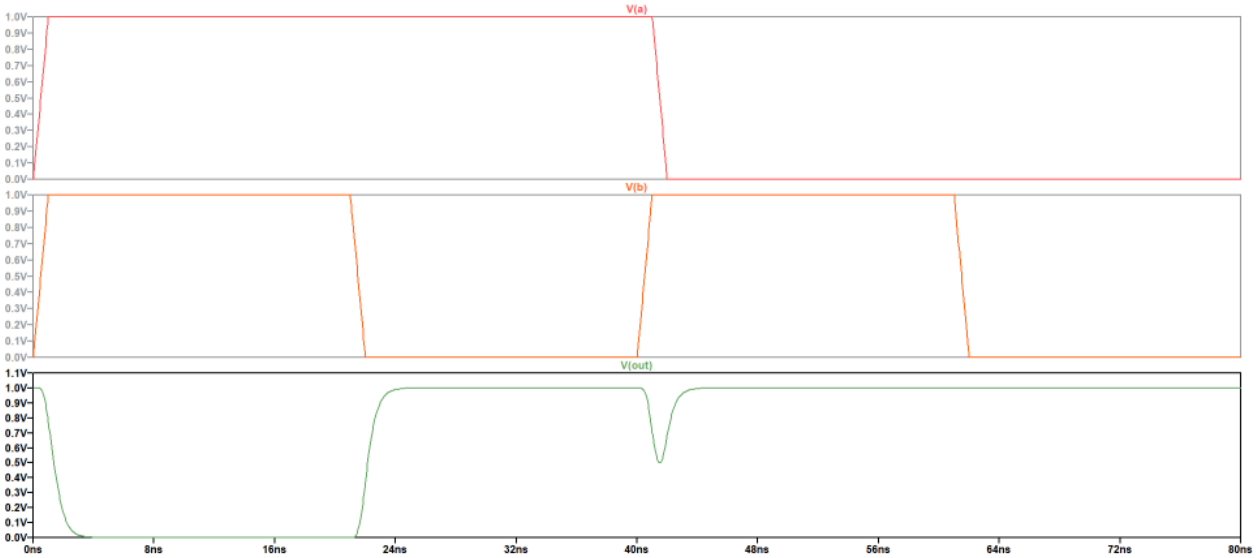

Figure 2. The functionality test of the 2-input NAND gate.

The propagation delay as shown in figure 3 is **498.38188ps.**

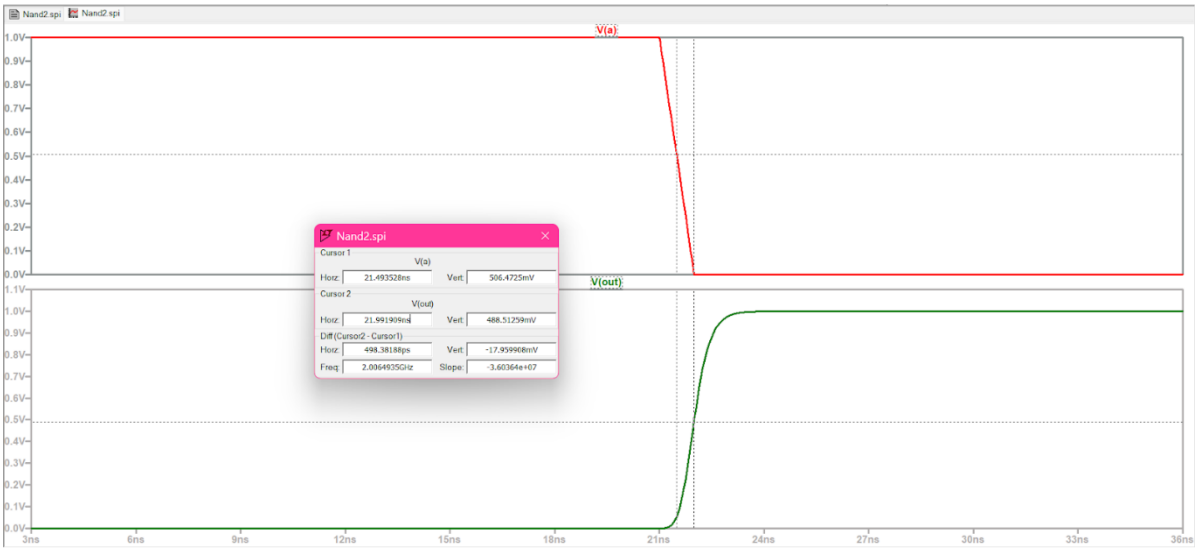

Figure 3. Propagation delay of the 2 input NAND gate.

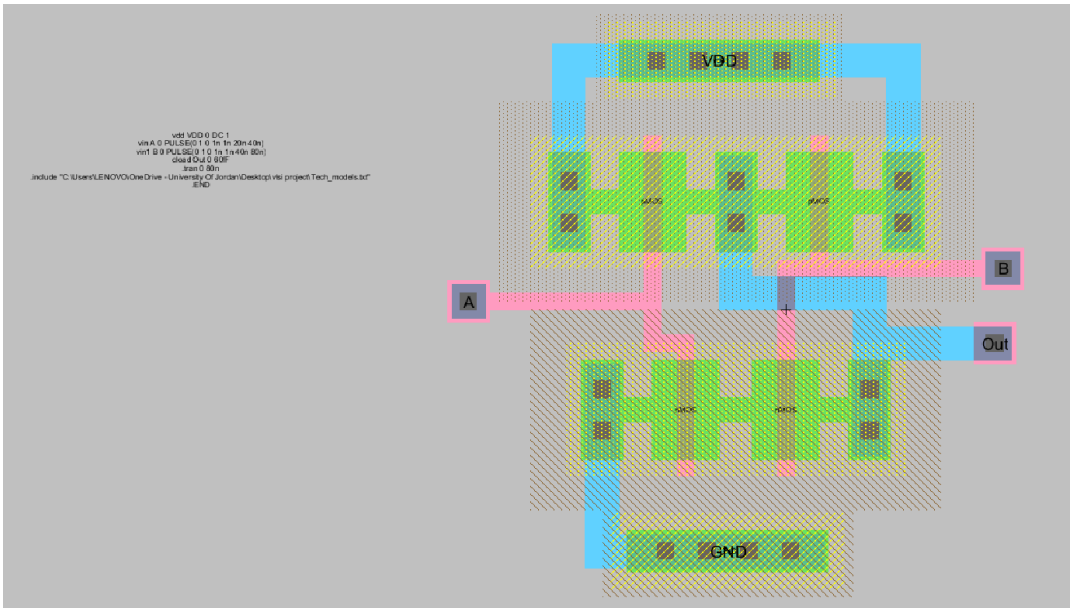

Figure 4. The layout of the 2 input NAND gate.

CMOS inverter

Below is the schematic design, layout design, testing and propagation delay of the CMOS inverter.

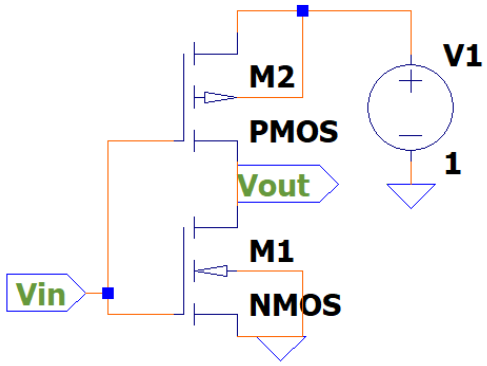

Figure 5. The schematic design of the CMOS inverter.

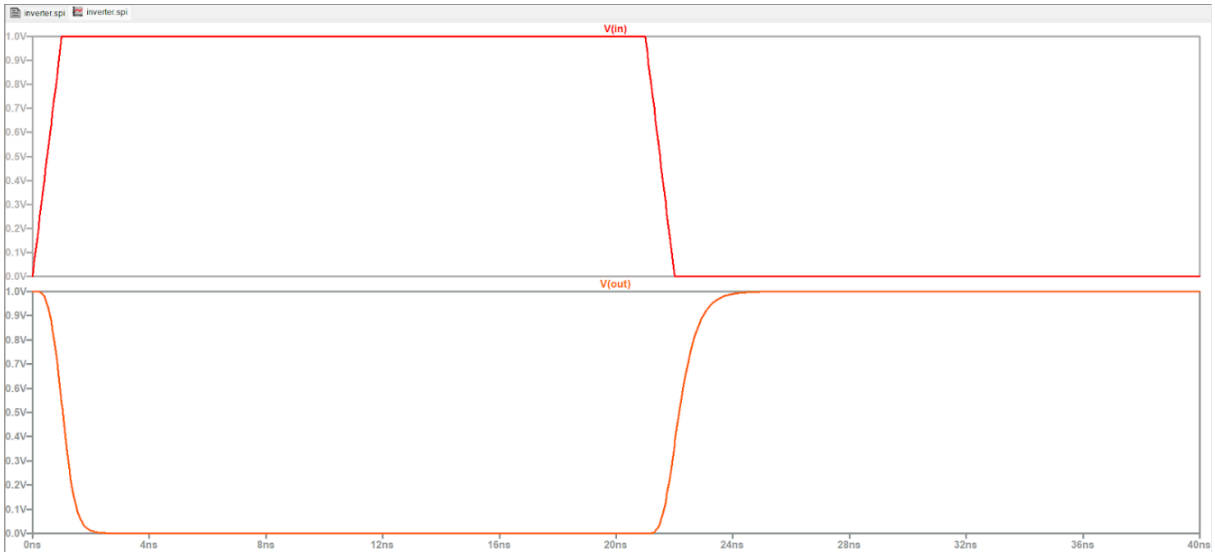

Figure 6. The functionality test of the inverter.

The inverter's propagation delay as shown in figure 7 is **647.24919ps.**

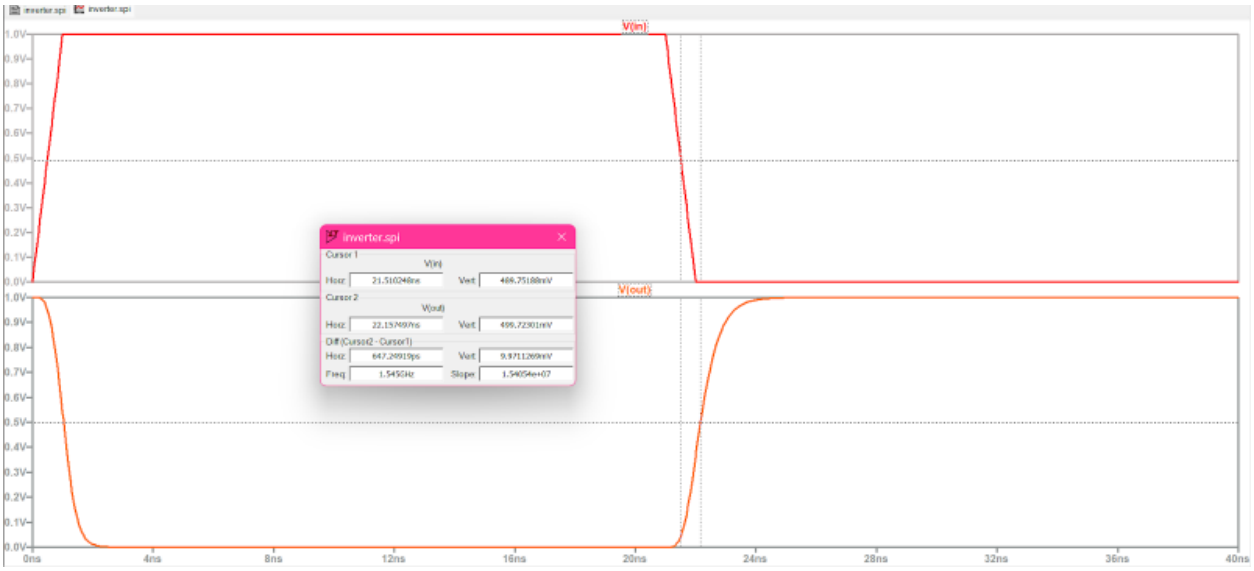

Figure 7. The propagation delay of the inverter gate

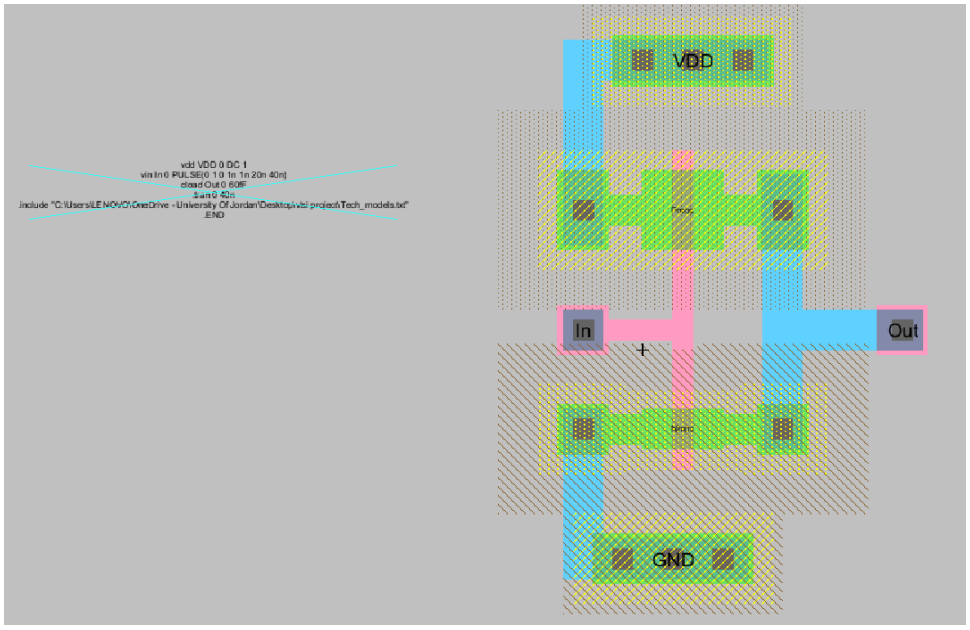

Figure 8. The layout of the inverter gate

2-Input AND Gate

Below is the schematic design, layout design, testing and propagation delay of the 2-input AND gate.

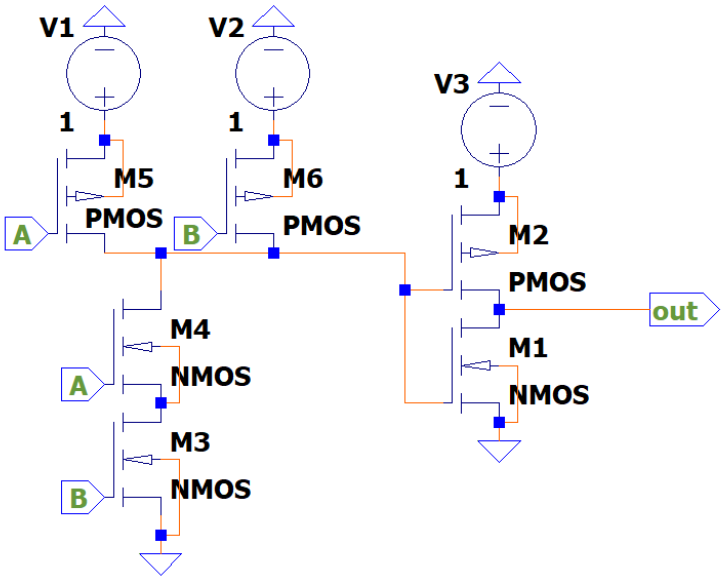

Figure 9. The schematic design of the 2 input AND gate.

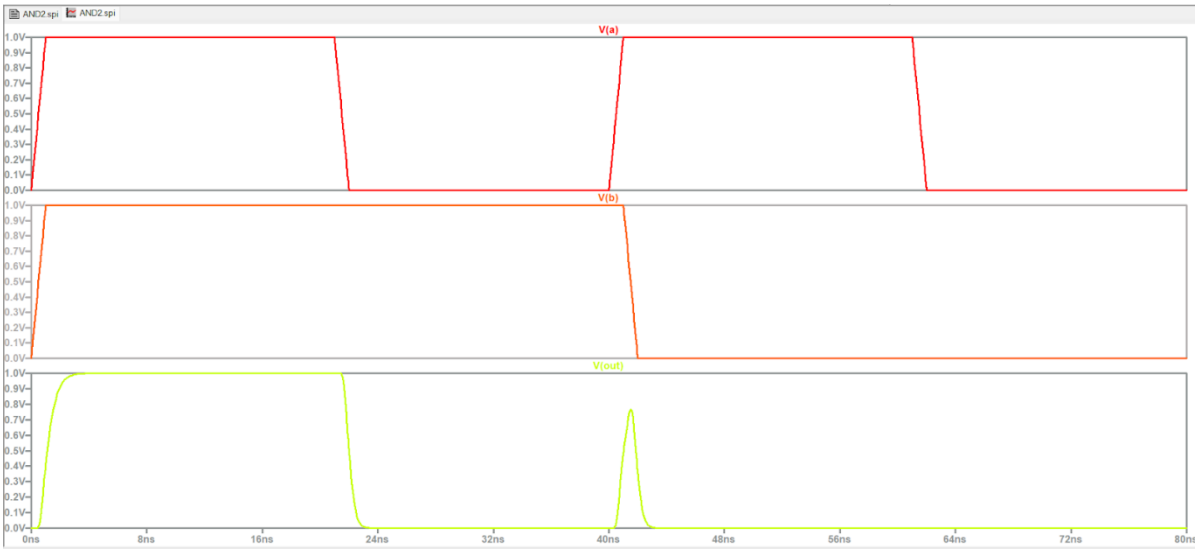

Figure 10.The functionality test of the 2 input AND gate.

The propagation delay as shown in figure 11 is = **510.24811ps.**

Figure 11. The propagation delay of the 2-input AND gate.

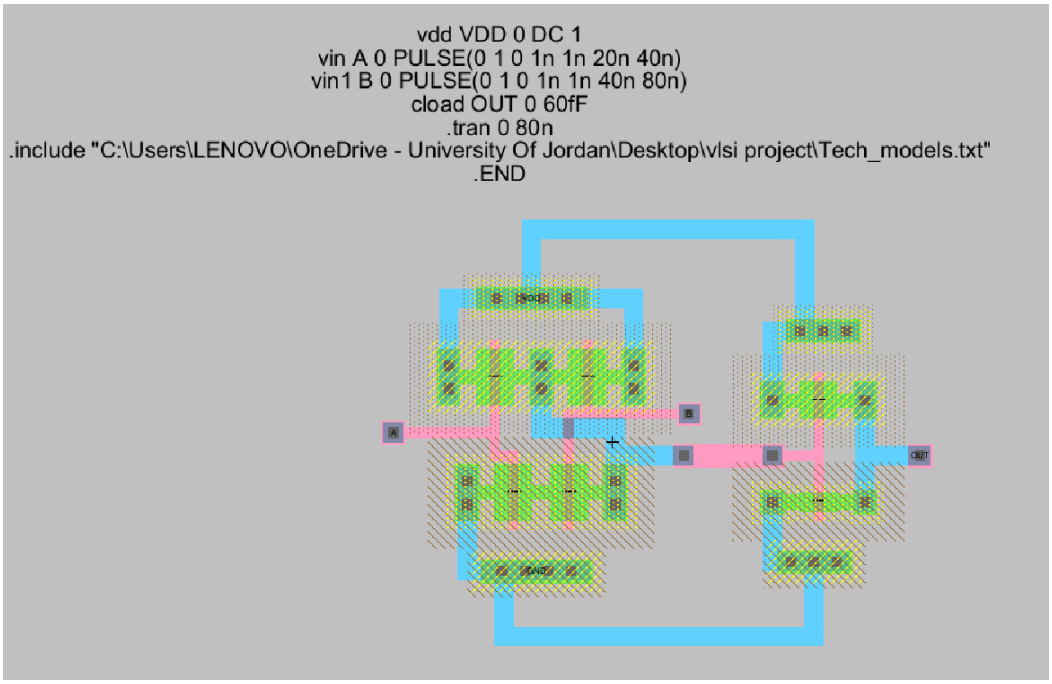

Figure 12. The layout of the 2-input AND gate.

3-Input NOR Gate

Below is the schematic design, layout design, testing, symbol and propagation delay of the 3-input NOR gate.

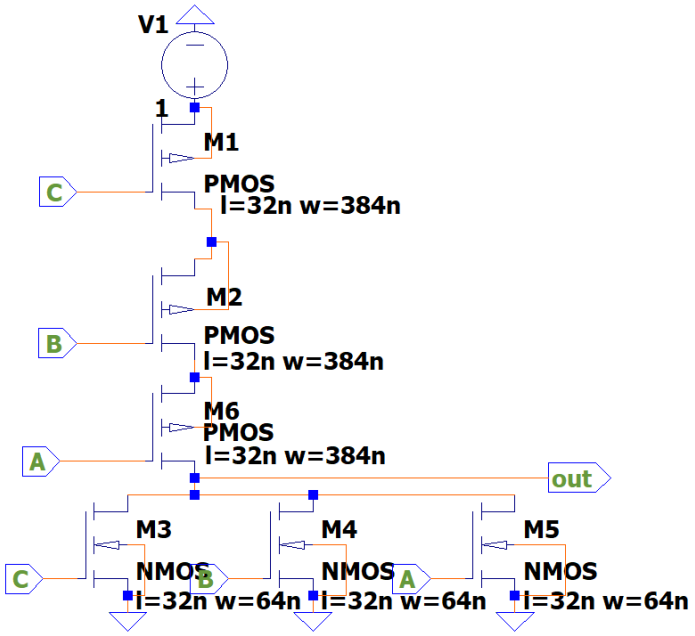

Figure 13. The schematic design of the 3-input NOR gate.

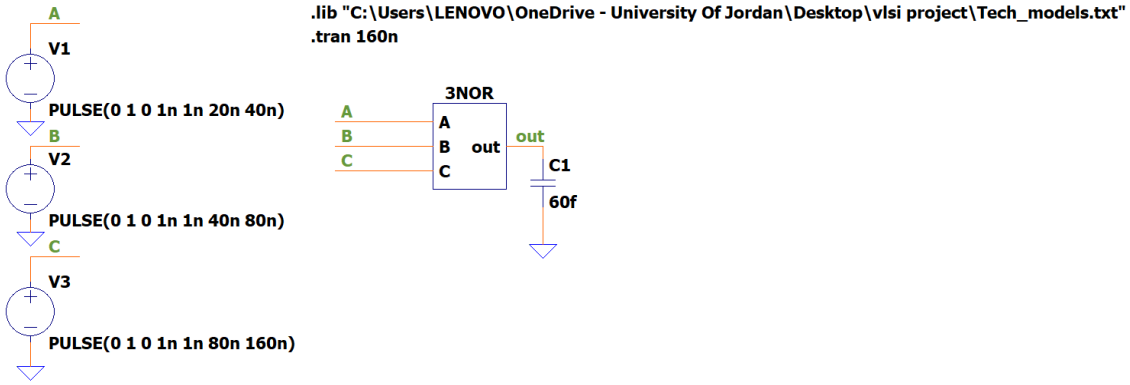

Figure 14. The symbol of the 3-input NOR gate.

Figure 15. The functionality test of the 3-input NOR gate.

The propagation delay as shown in figure 16 is = **642.08243ps.**

Figure 16. The propagation delay of the 3-input NOR gate.

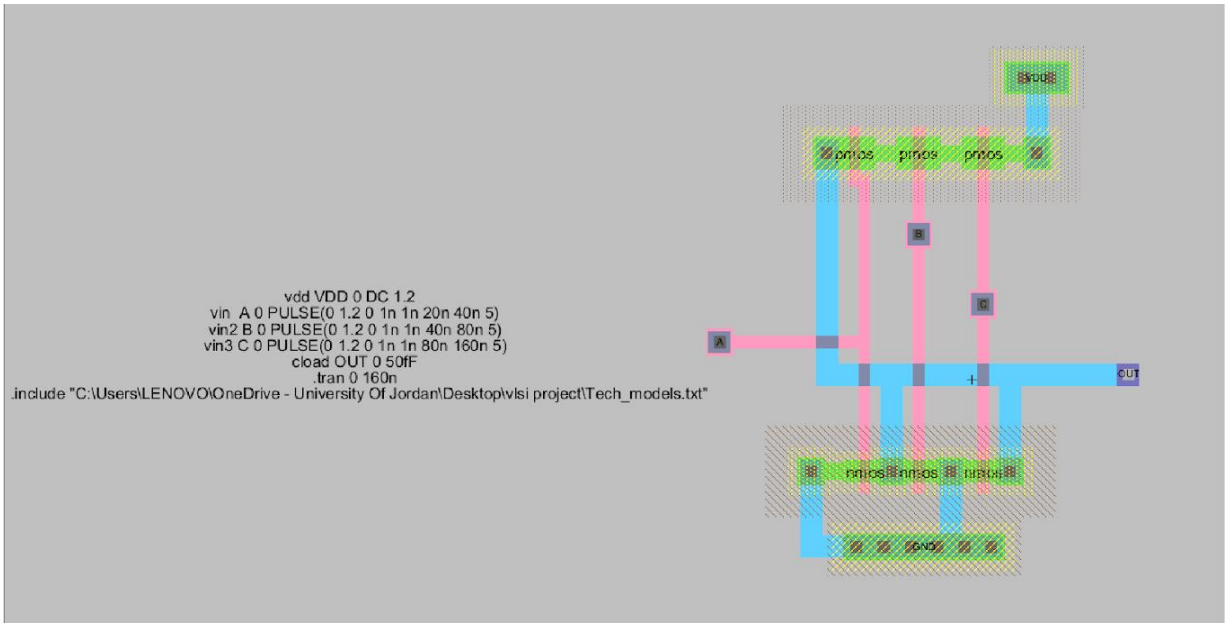

Figure 17. The layout of the 3-input NOR gate.

Majority Circuit

The majority circuit outputs a logic high (1) when at least two of the three inputs are high; otherwise, it outputs a logic low (0). To build the schematic design of this circuit, we first constructed the truth table that describes its behavior, which can be seen in table 1.

Table 1 The majority circuit truth table.

A	B	C	OUT
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

So that equation of the output is OUT = A&B + A&C + B&C

Below is the schematic design, layout design, symbol, testing and propagation delay of the majority circuit.

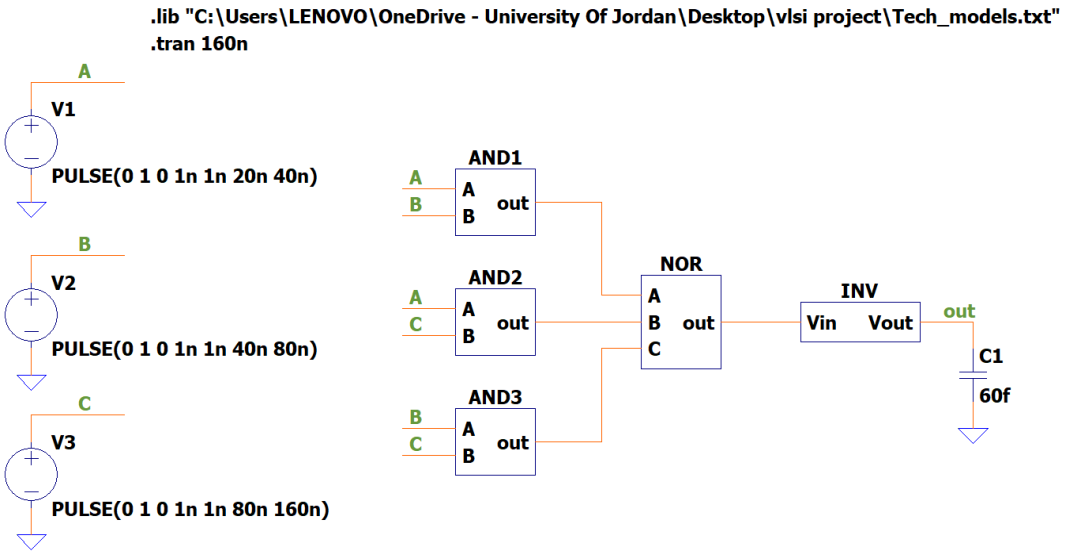

Figure 18. The schematic design of the majority circuit.

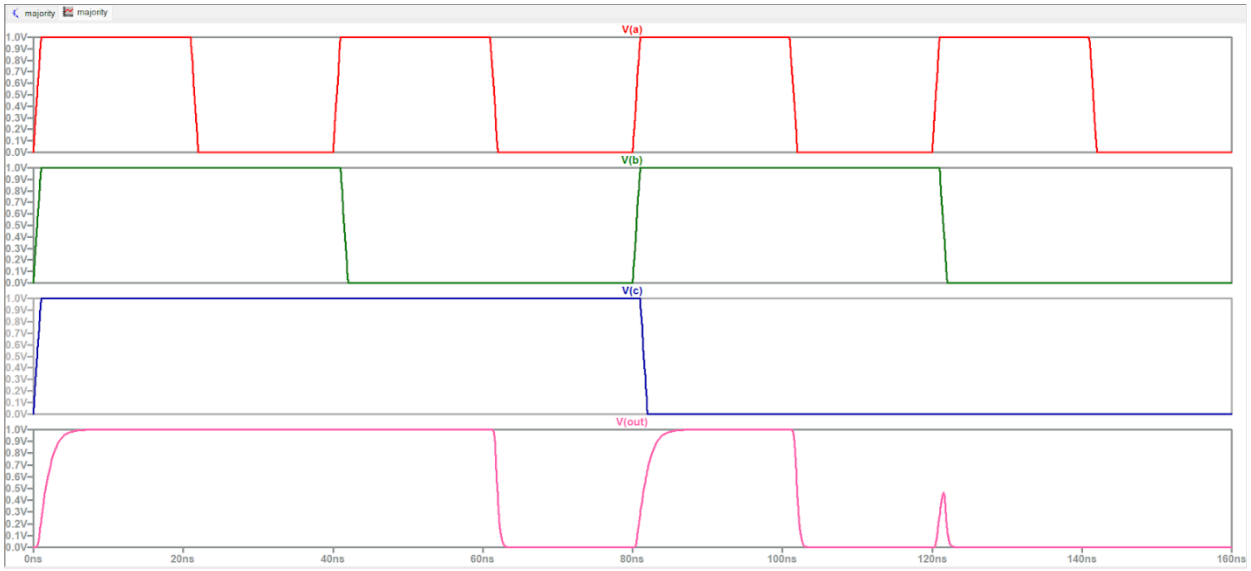

Figure 19. The functionality test of the majority circuit.

The propagation delay as shown in figure 20 is = **61.952535ns.**

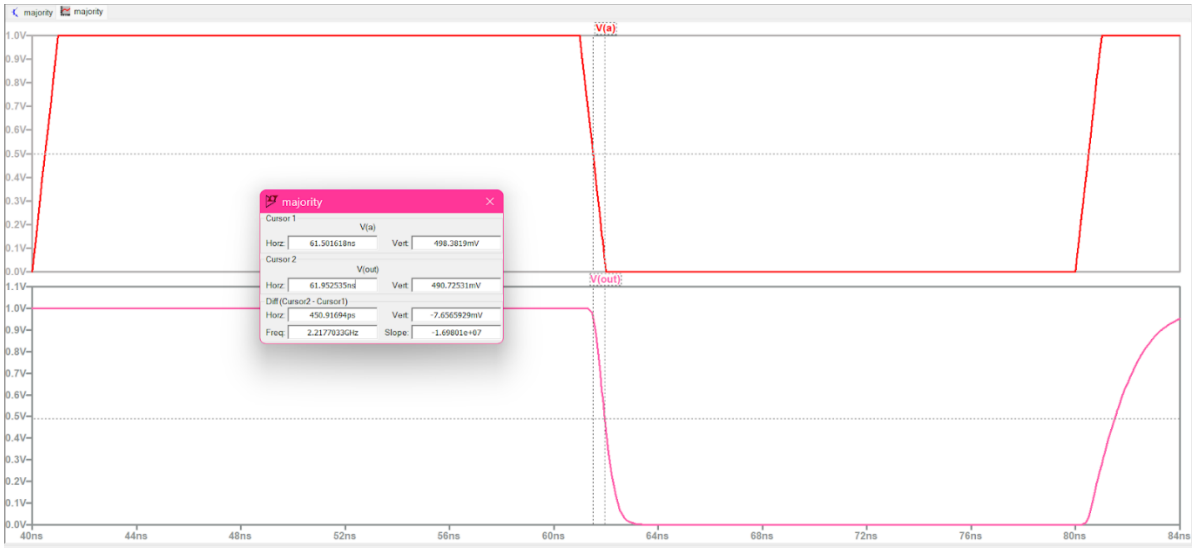

Figure 20. The propagation delay of the majority circuit.

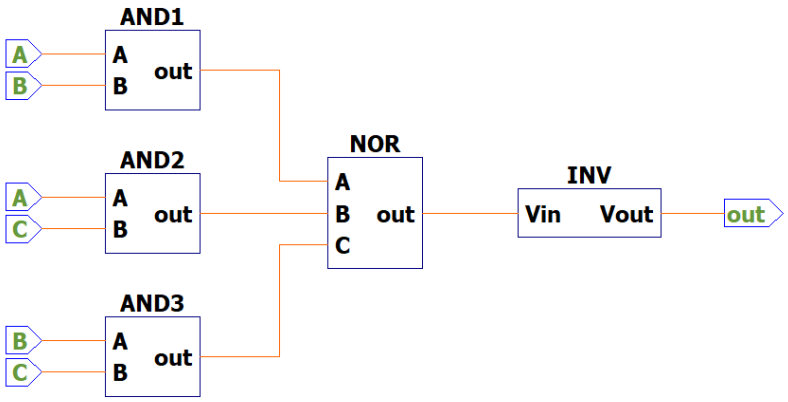

Figure 21. The symbol of the majority circuit.

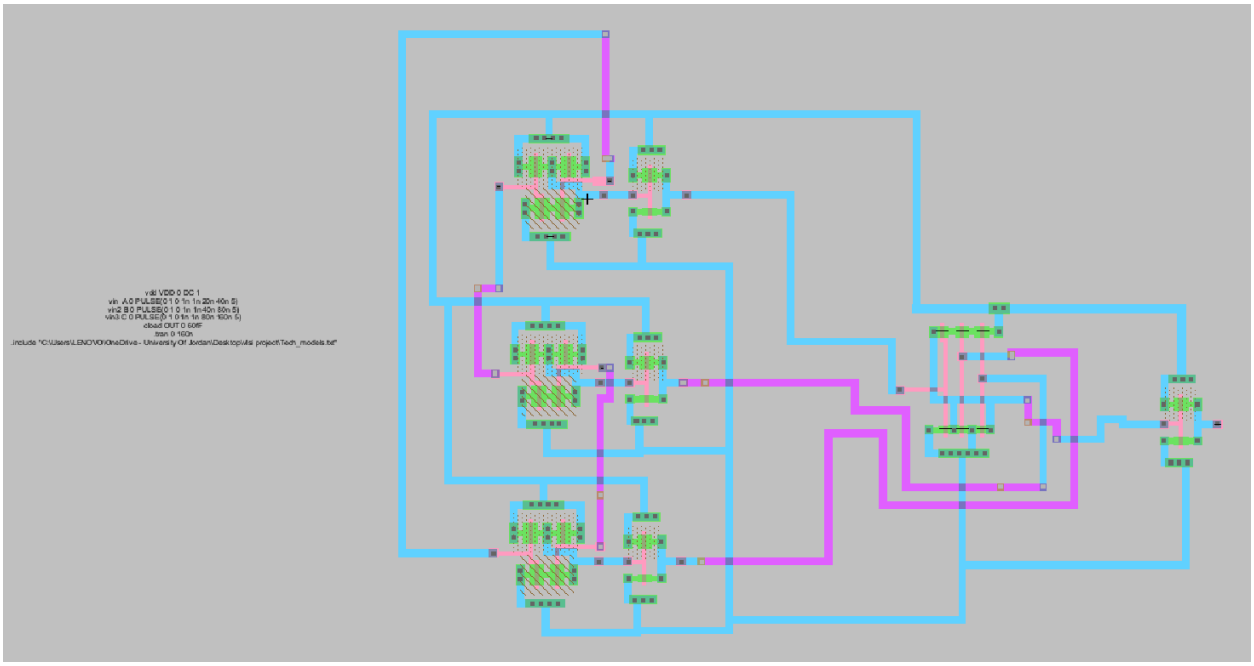

Figure 22. The layout of the Majority circuit.

Clocked SR-Latch

Below is the schematic design, layout design, testing and propagation delay of the clocked SR-latch.

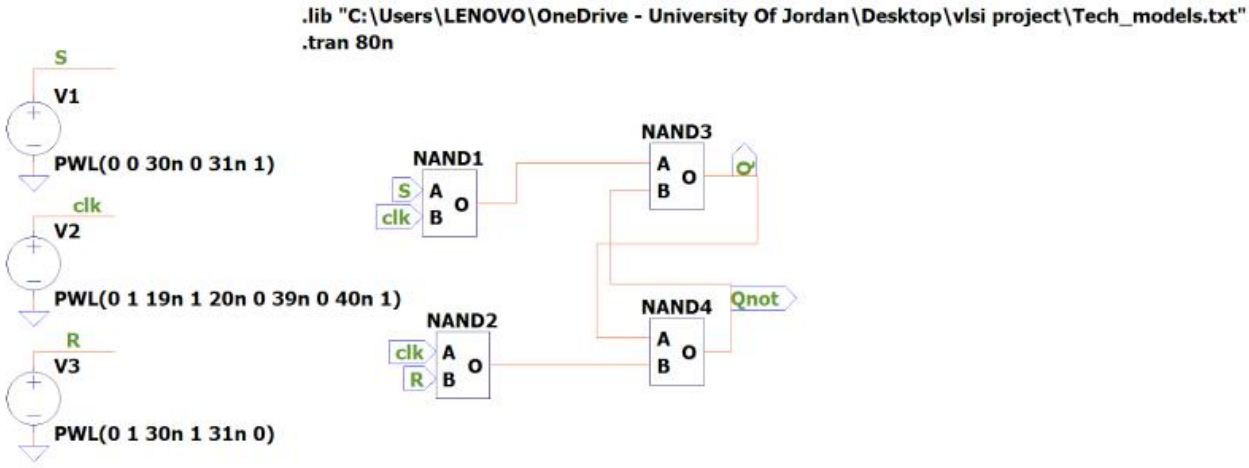

Figure 23. The schematic design of the clocked SR-latch.

Figure 24. The functionality test of the Clocked SR-Latch.

The propagation delay as shown in figure 25 is = **26.968716ps.**

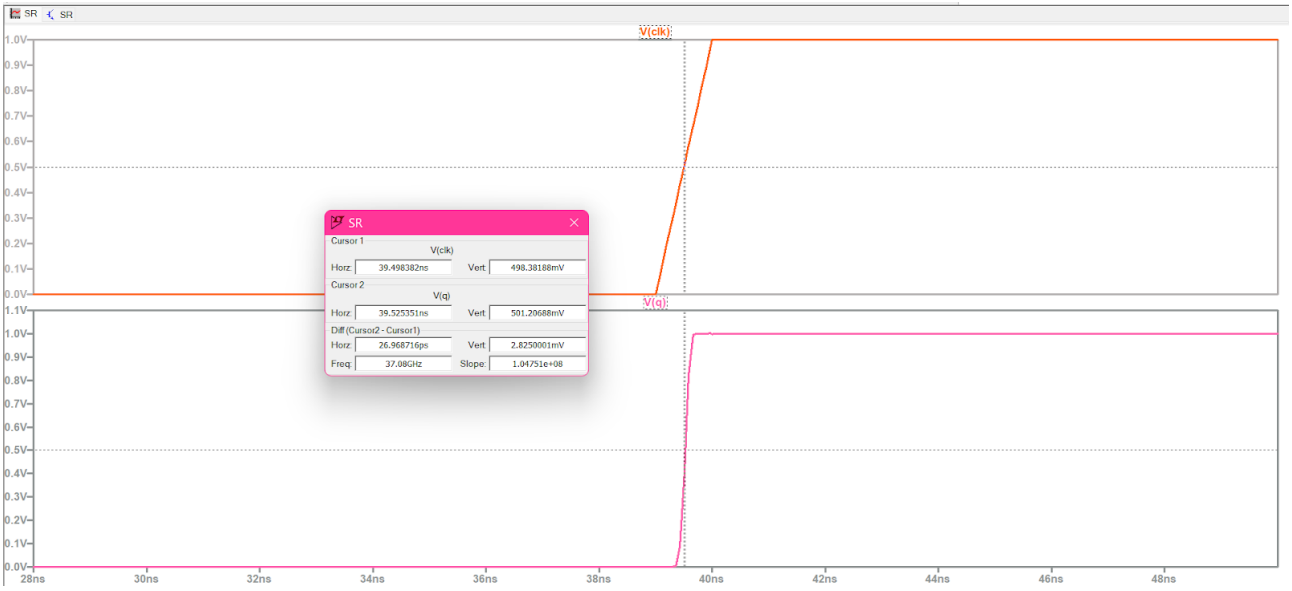

Figure 25. The propagation delay of the clocked SR-latch.

D Flip-Flop

Below is the schematic design, layout design, testing and propagation delay of the D flip-flop.

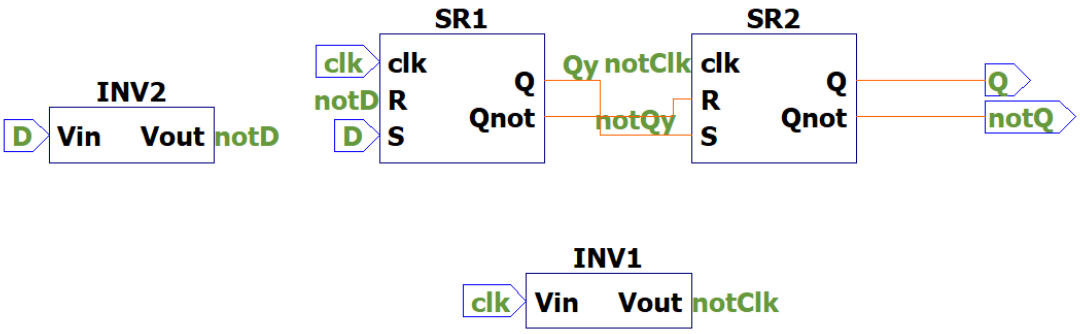

Figure 26. The schematic design of the DFF.

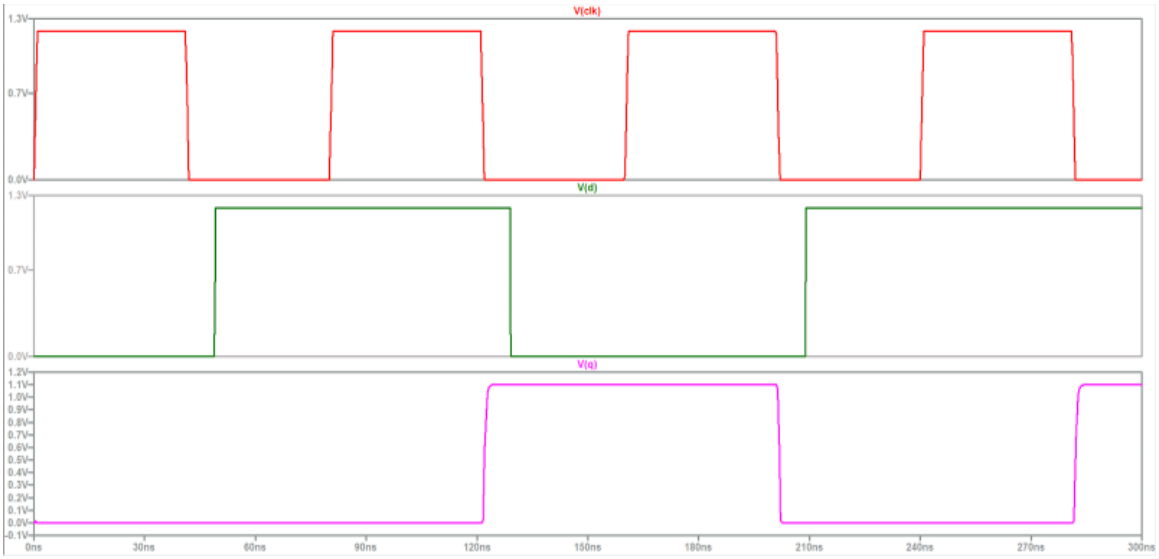

Figure 27. The functionality test of the DFF.

The propagation delay as shown in figure 28 is = **429.50108ps.**

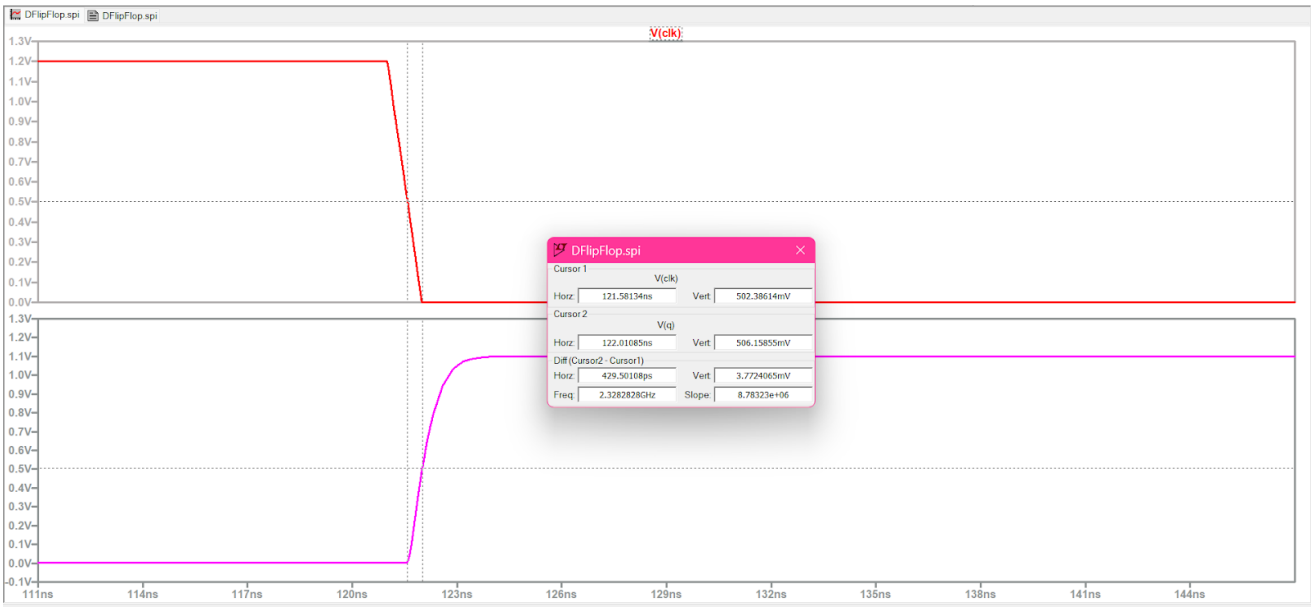

Figure 28. The propagation delay of the DFF.

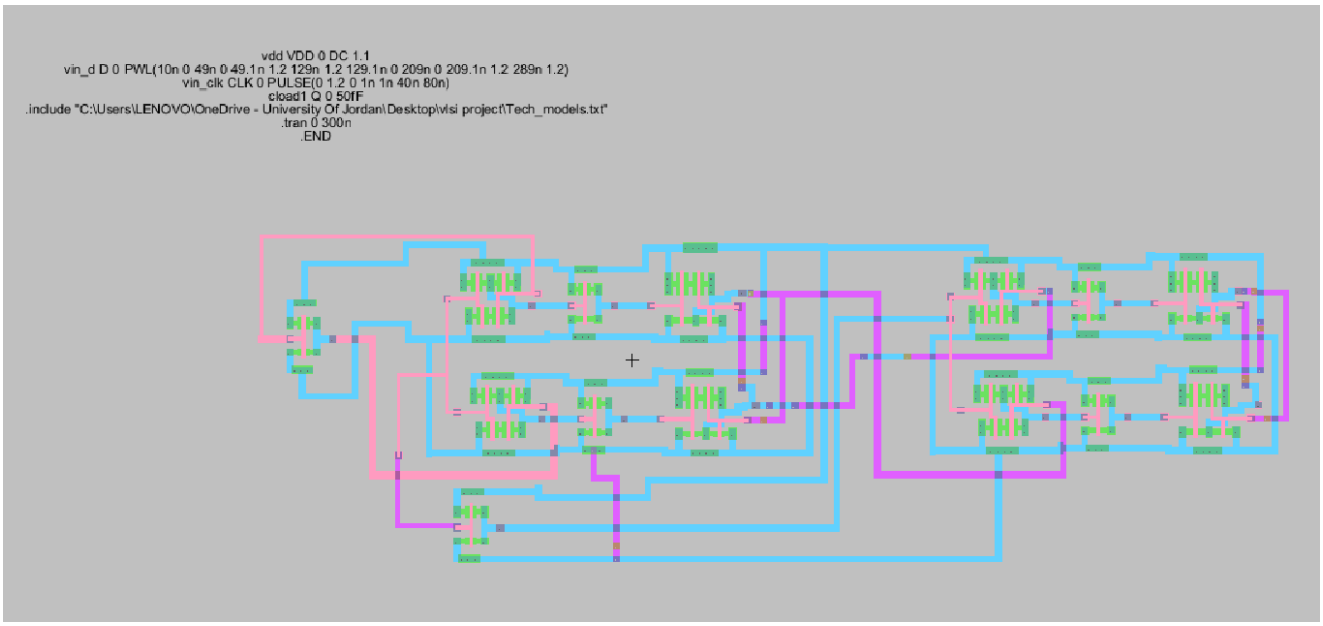

Figure 29. The layout of the DFF.

2-Input NOR Gate

Below is the schematic design, layout design, testing and propagation delay of the 2-Input nor gate.

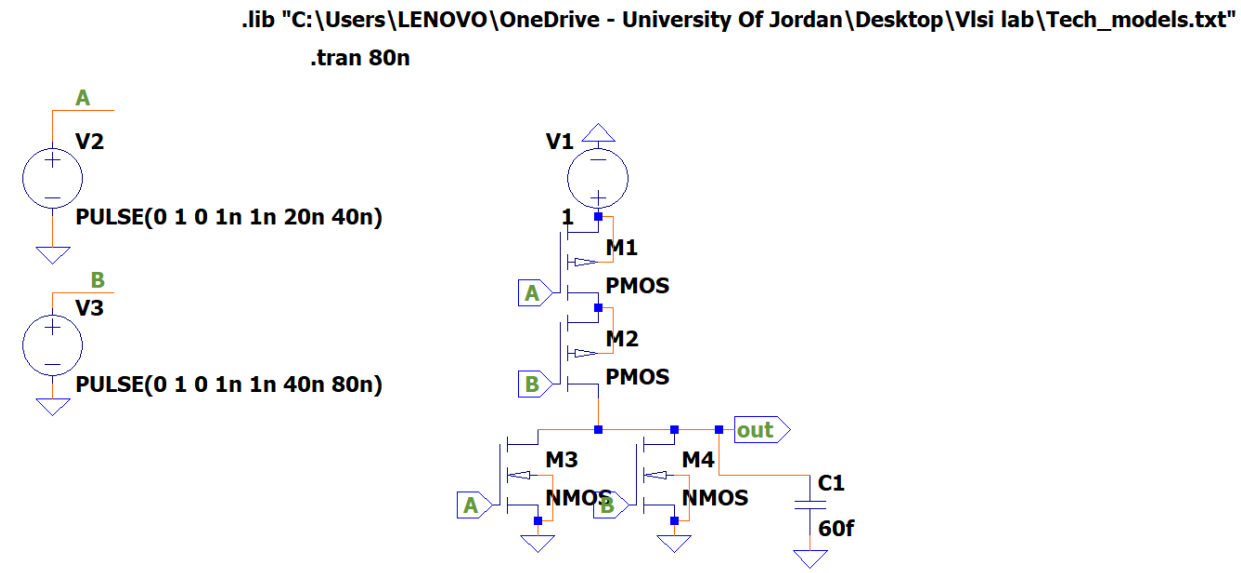

Figure 30. The schematic design of the 2-input NOR gate.

Figure 31. The functionality test of the 2-input NOR gate.

The NOR propagation delay as shown in figure 32 is = **676.40693ps.**

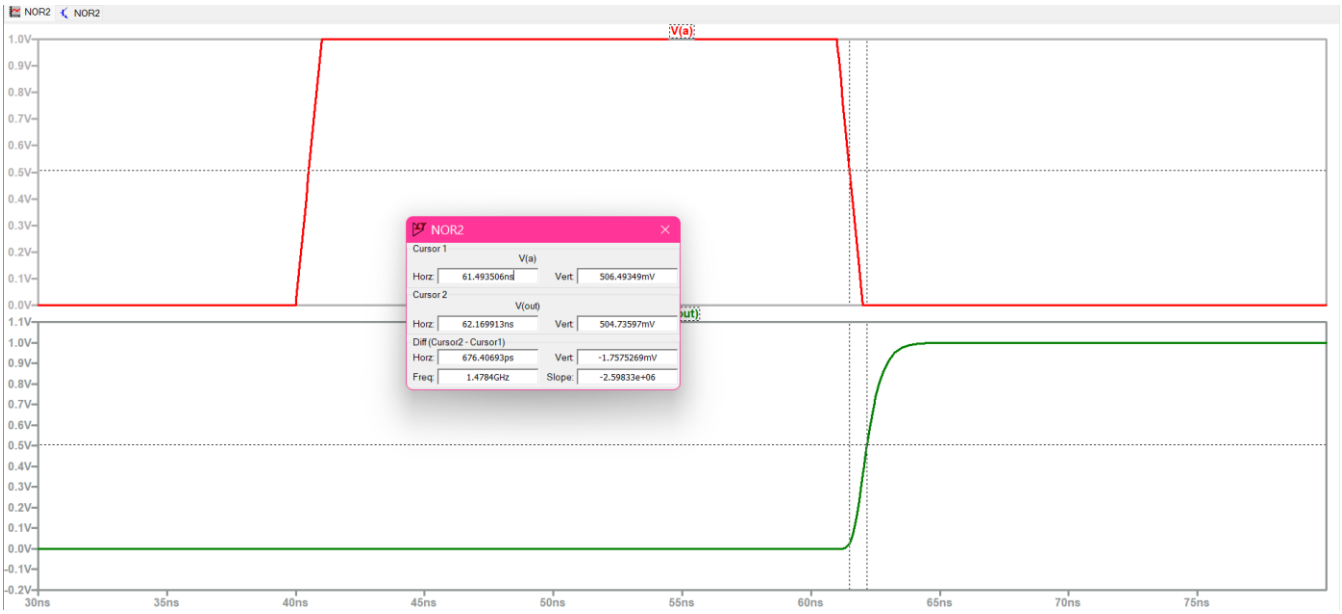

Figure 32. The propagation delay of the 2-input NOR gate.

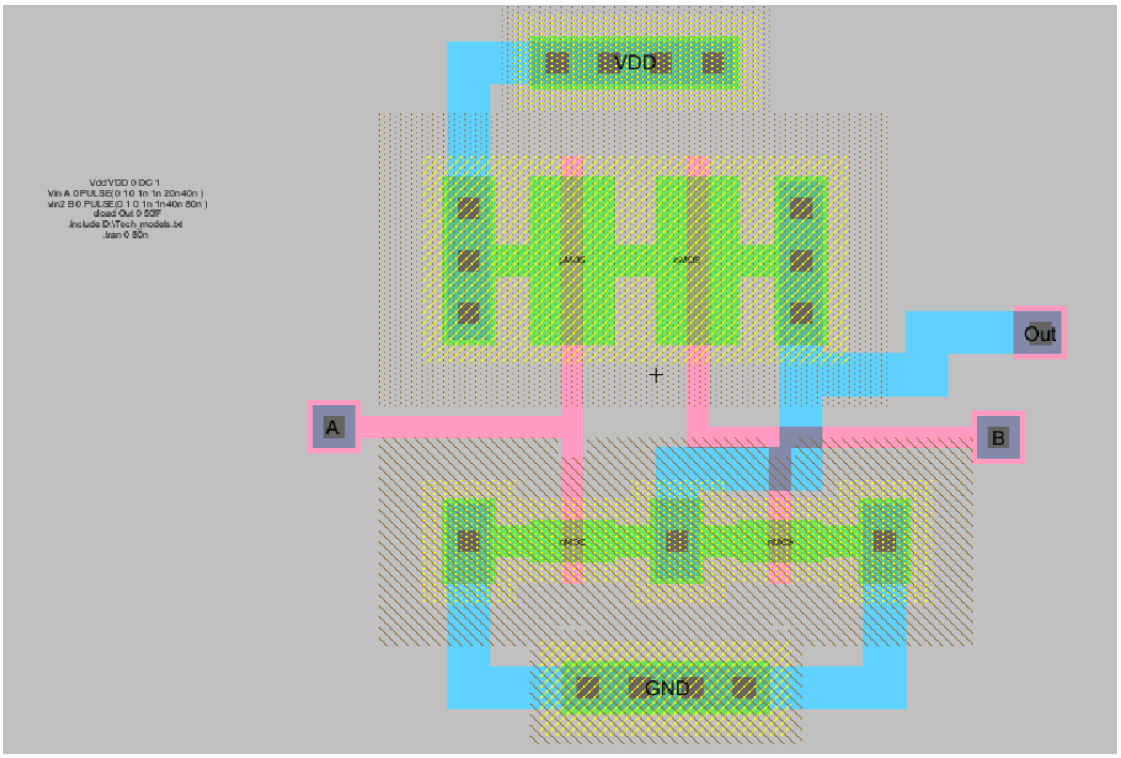

Figure 33. The layout of the 2-input NOR gate.

2-Input XOR Gate

Below is the schematic design, layout design, testing and propagation delay of the XOR gate.

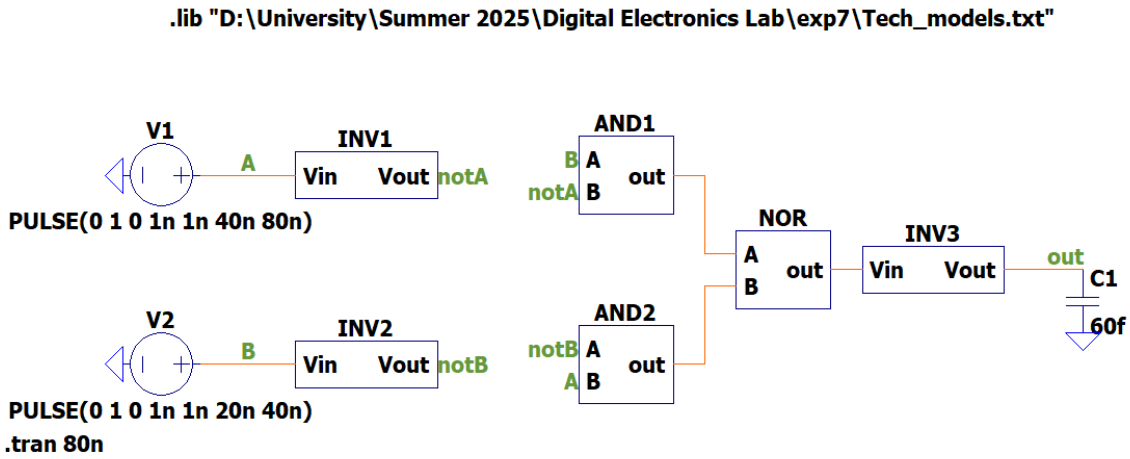

Figure 34. The schematic design of the 2-input XOR gate.

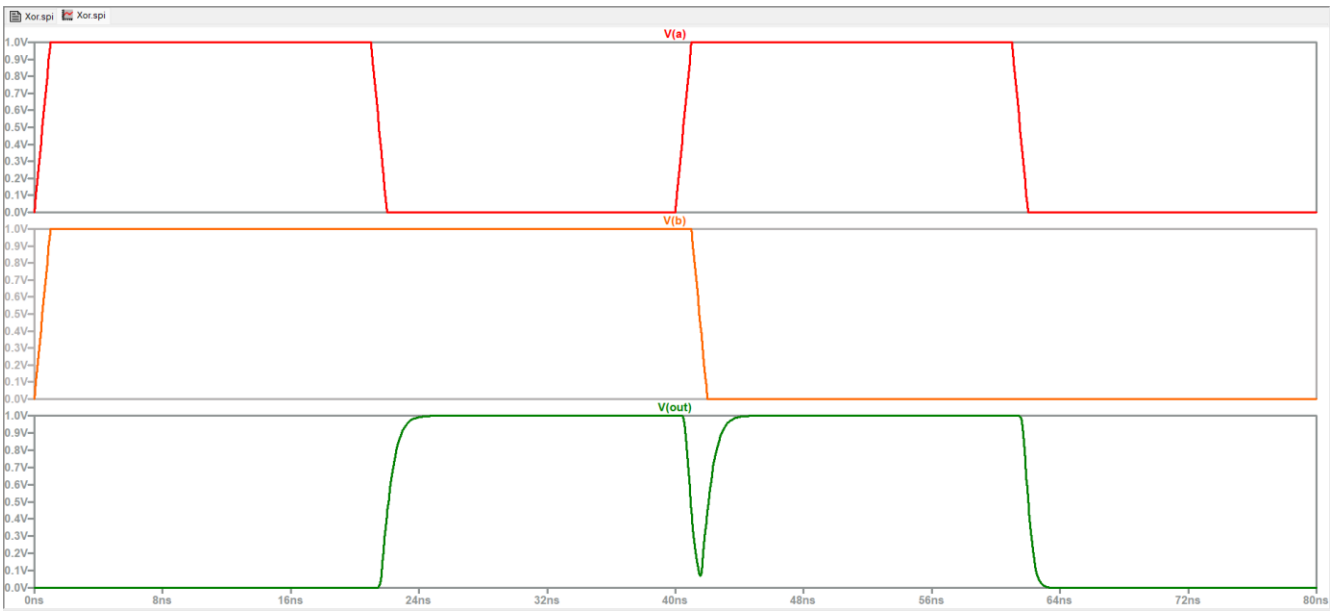

Figure 35. The functionality test of the 2-input XOR gate.

The XOR propagation delay as shown in figure 36 is = **582.52427ps.**

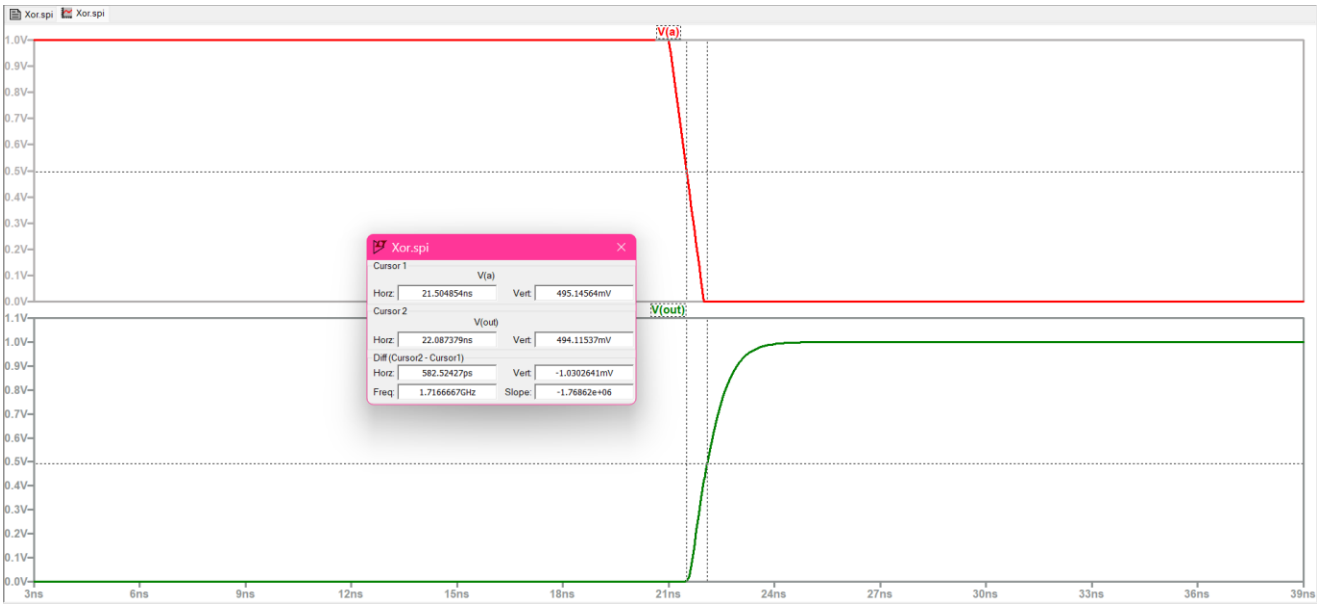

Figure 36. The propagation delay of the 2-input XOR gate.

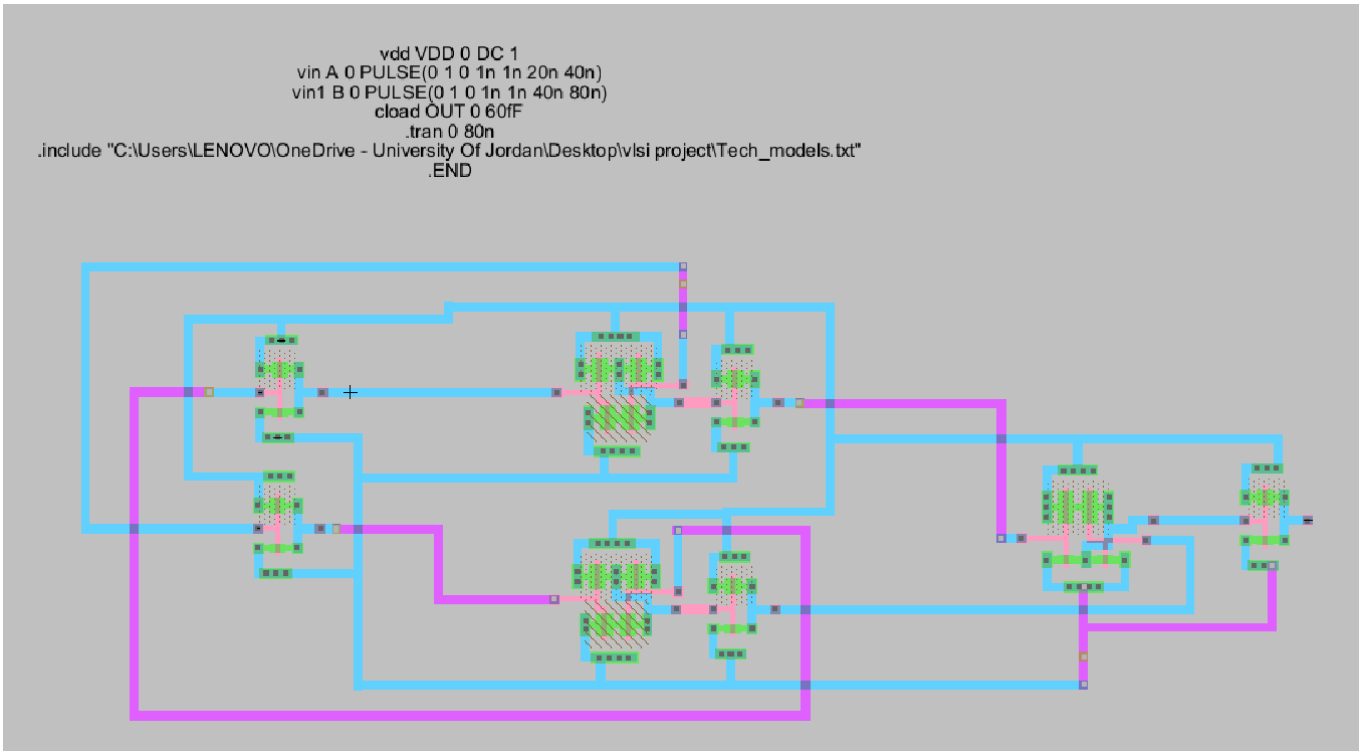

Figure 37. The layout of 2-input XOR gate.

6T SRAM Cell

Below is the schematic design, layout design, and testing of the 6T SRAM cell.

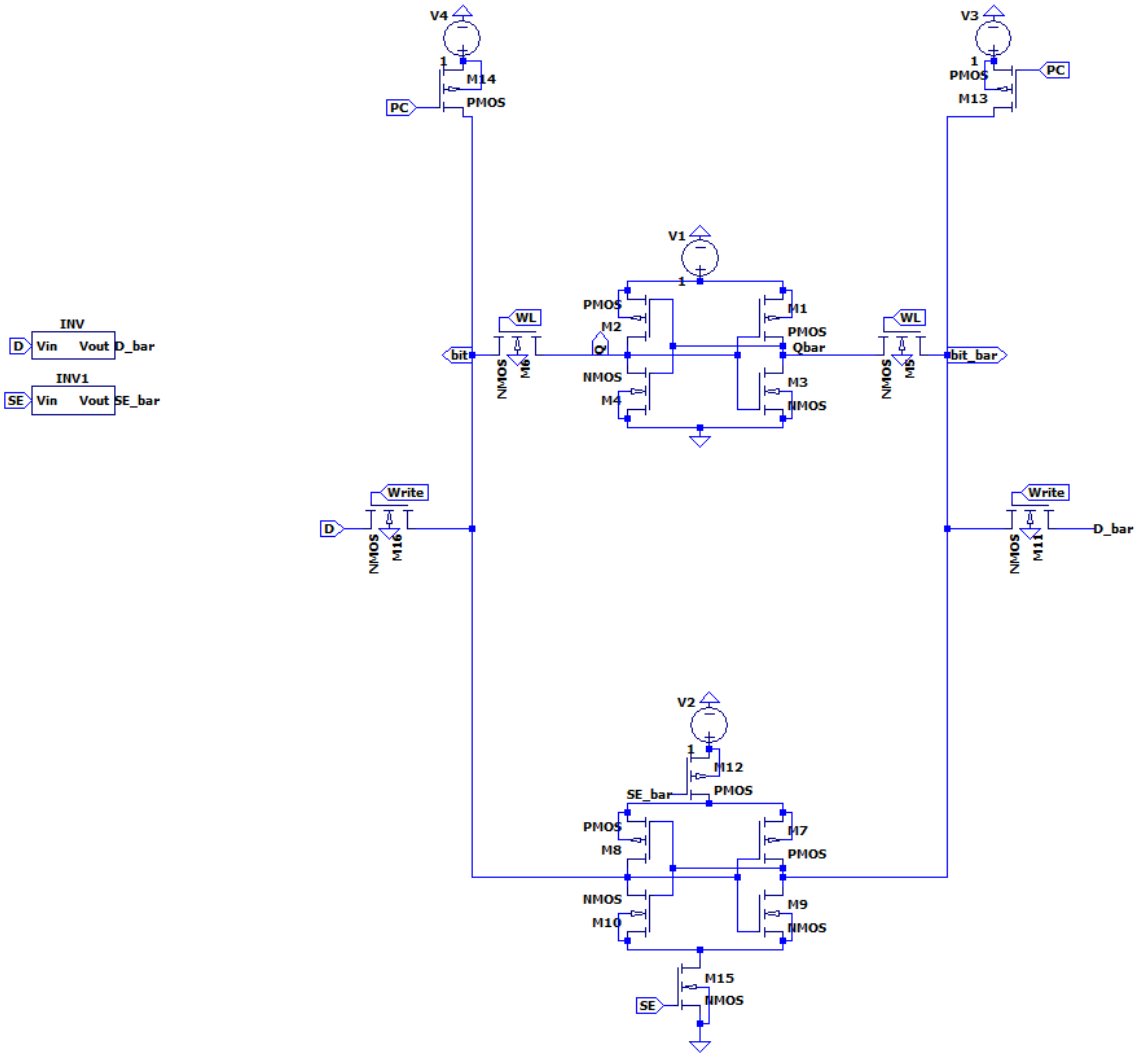

Figure 38. The SRAM cell schematic design.

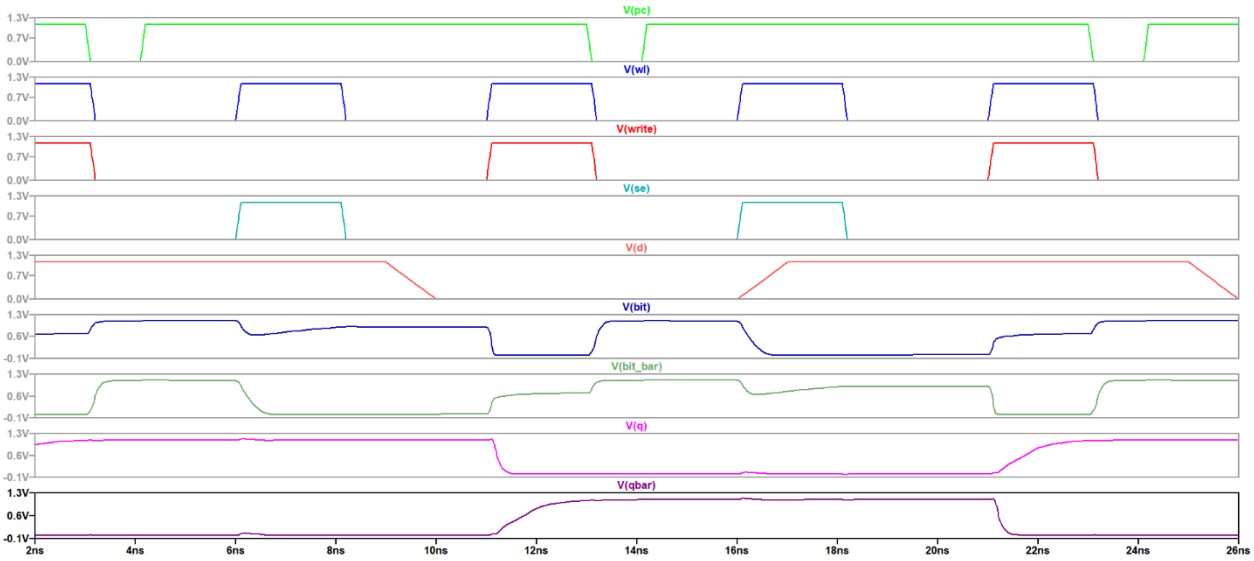

Figure 39.SRAM functionality test.

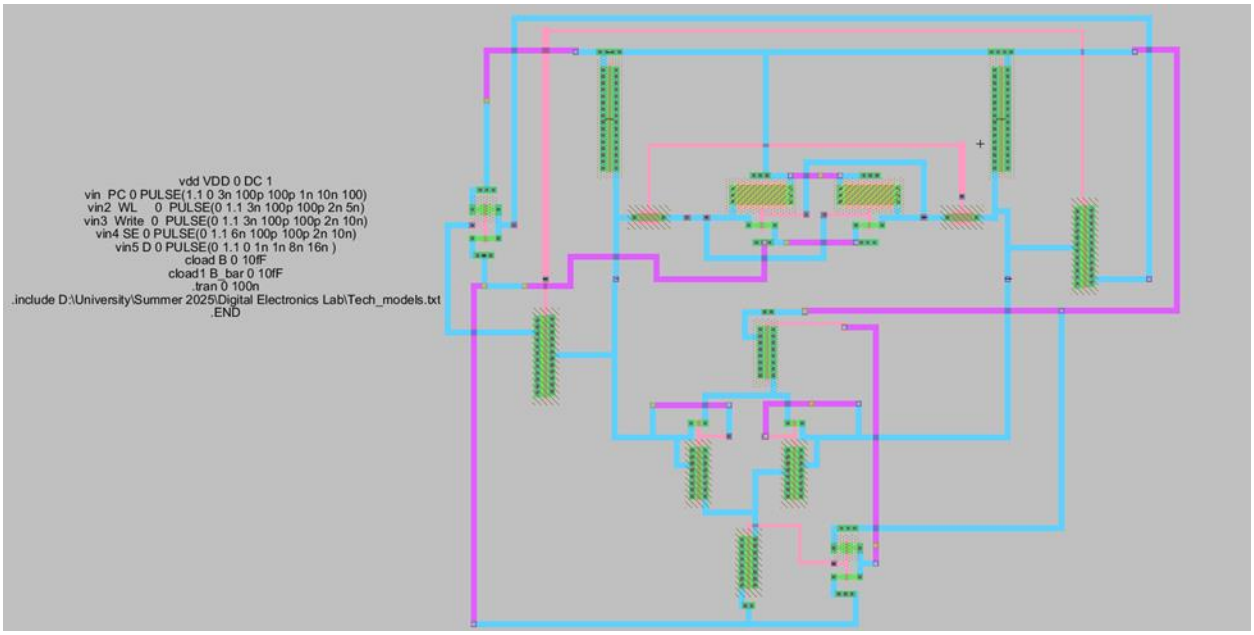

Figure 40.The layout of the SRAM cell.

3-Bit Counter

Below is the schematic design, layout design, propagation delay and testing of the 3-bit counter.

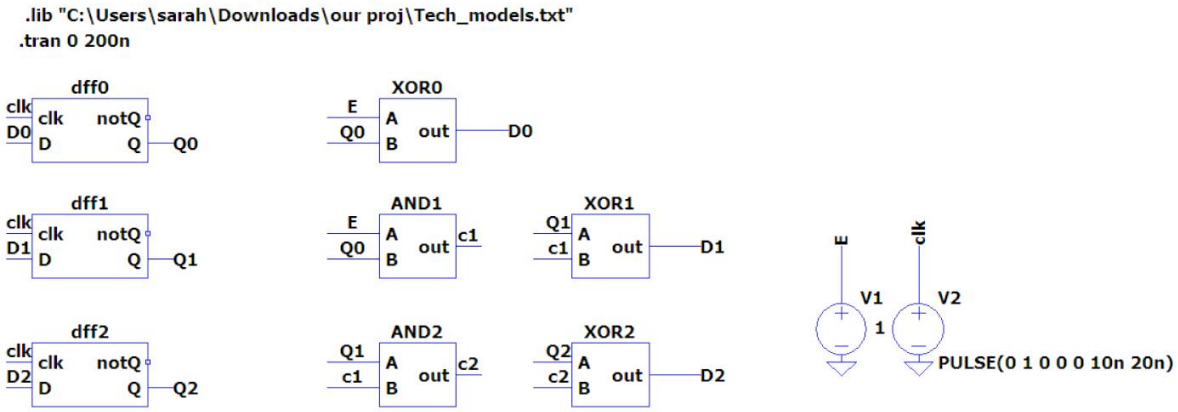

Figure 41. The 3-bit counter cell schematic design.

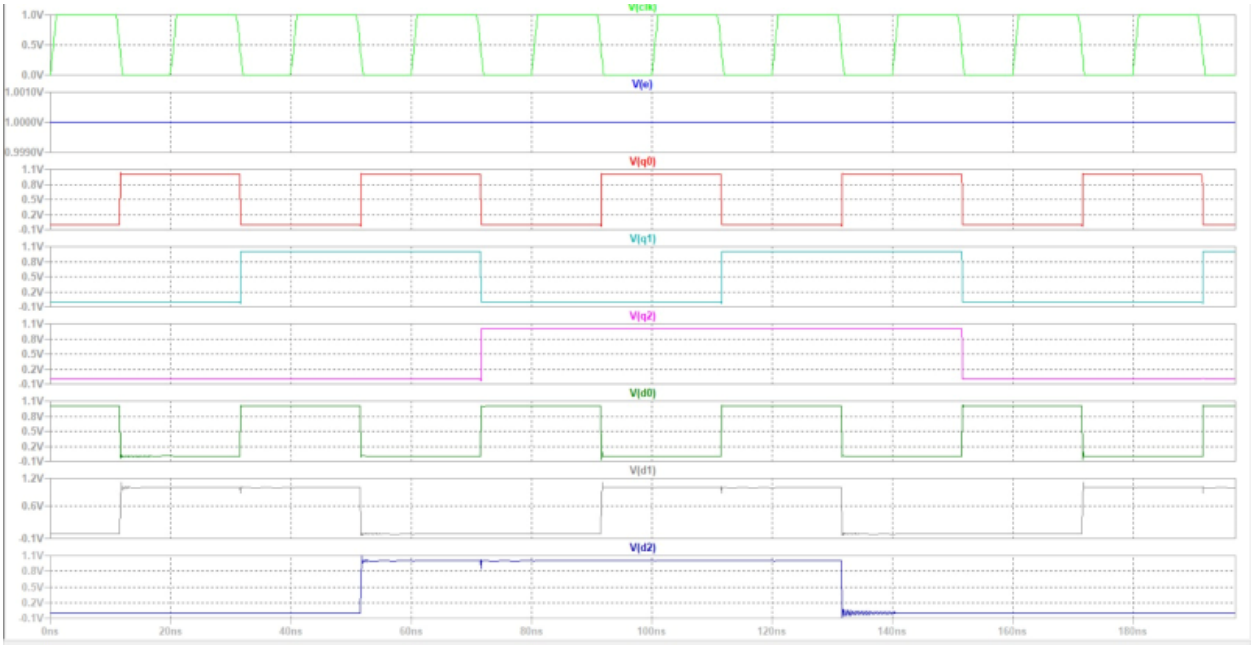

Figure 42. 3-bit counter functionality test.

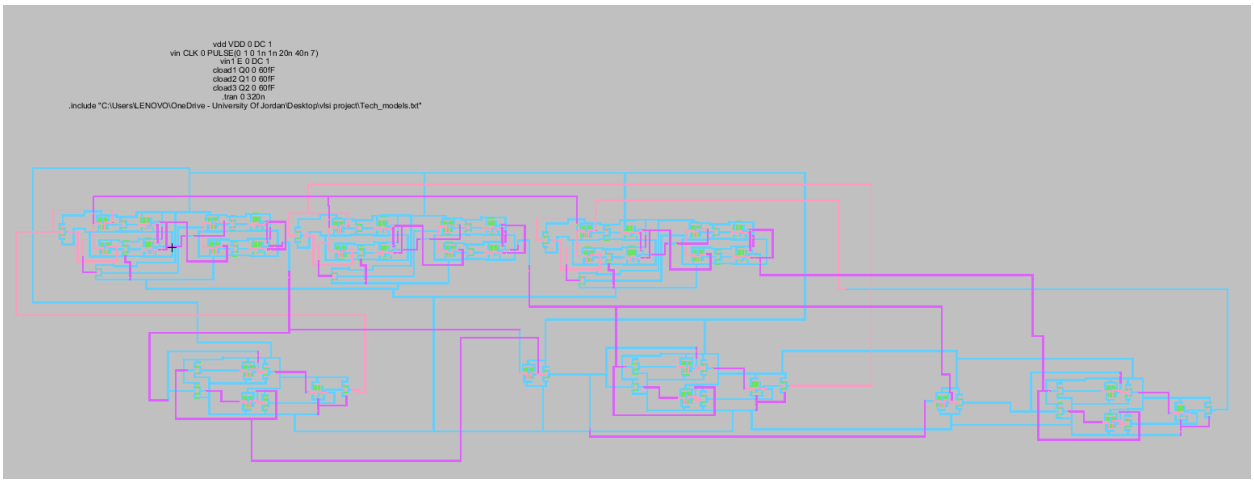

Figure 43.The layout of the 3-bit counter.

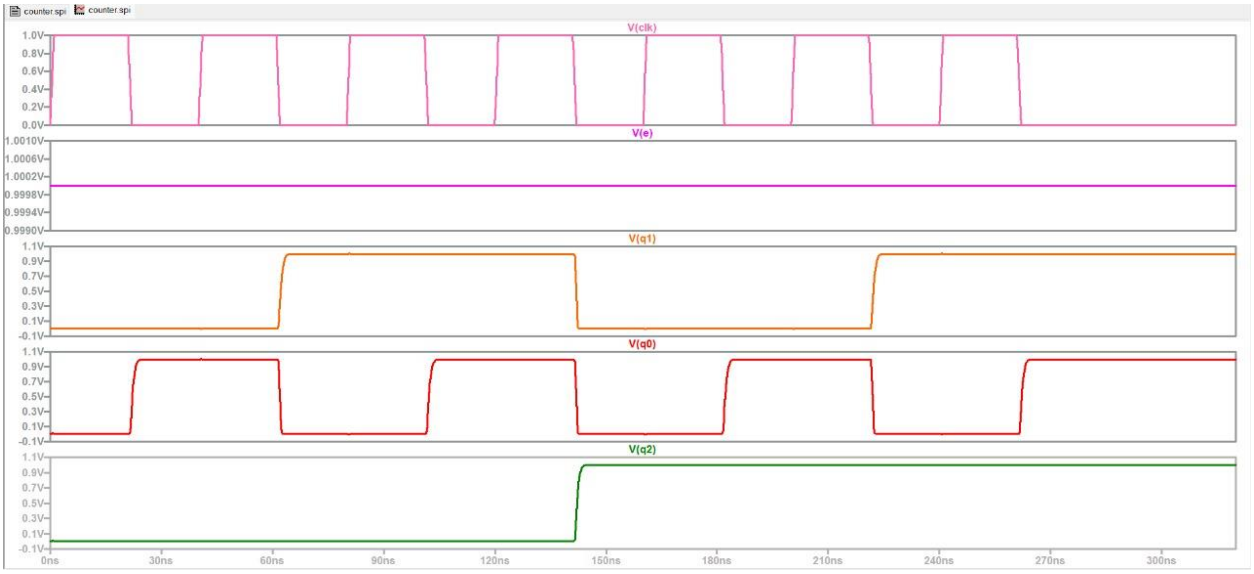

Figure 44. Layout functionality test.

Propagation delay of the 3-bit counter:

1. Q2 propagation delay = 691.97397ps, as shown below.

Figure 45.Propagation delay of the 3-bit counter, for Q2.

2. Q1 propagation delay = 485.35792ps, as shown below.

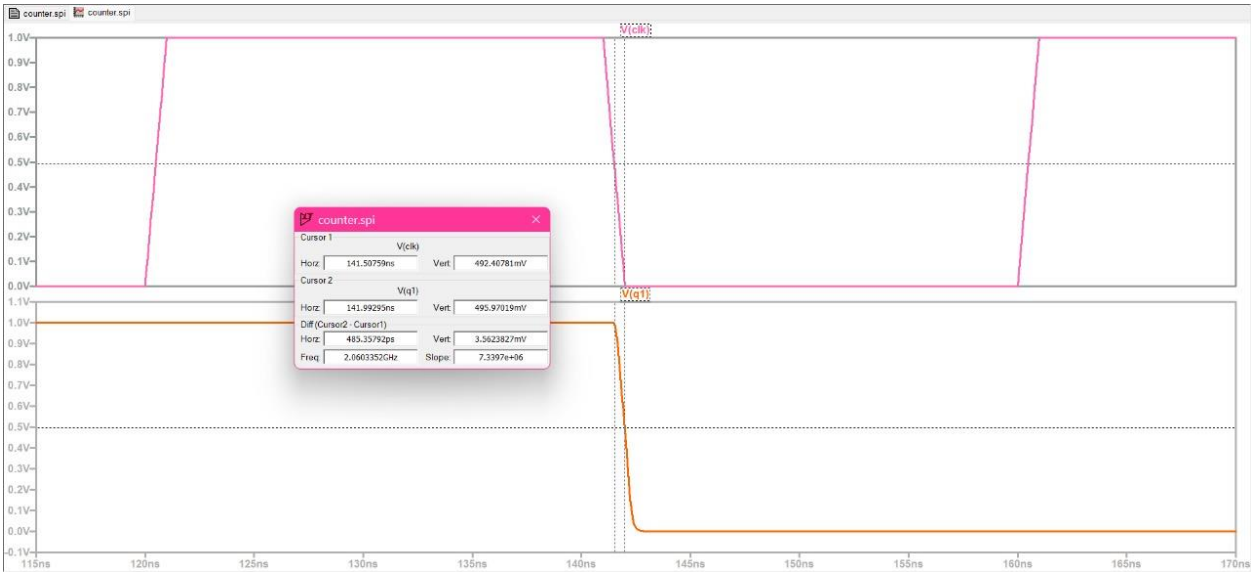

Figure 46.Propagation delay of the 3-bit counter, for Q1.

Q0 propagation delay = 493.34563ps, as shown below.

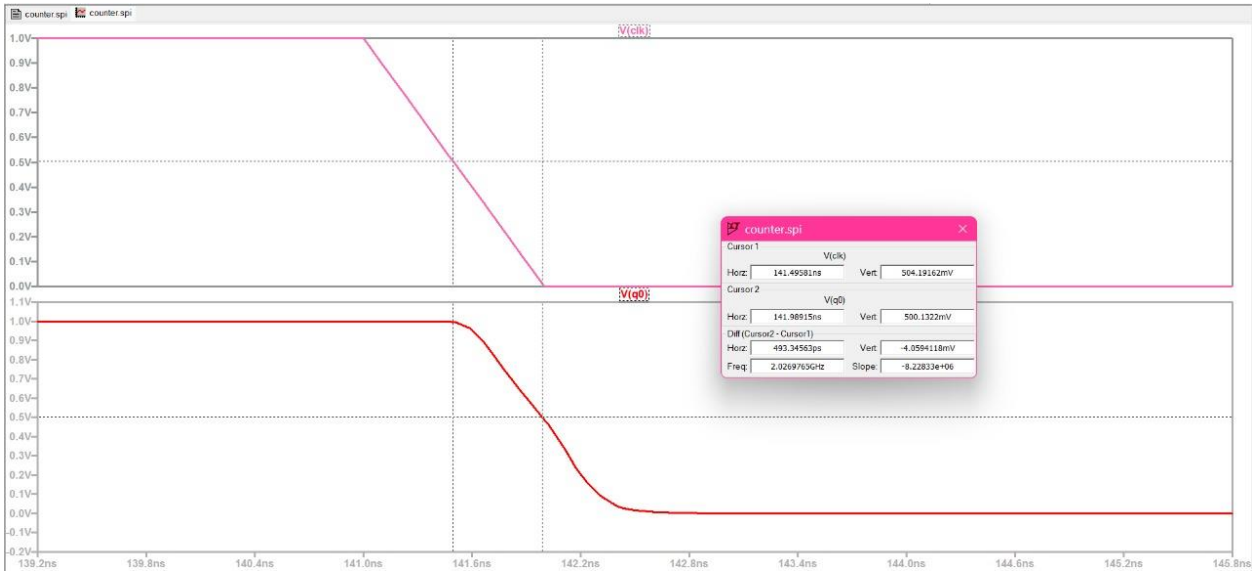

Figure 47.Propagation delay of the 3-bit counter, for Q0.

The Top-Level Circuit

After designing and testing all the individual components of the system, the next step was to integrate them into the top-level circuit. Figure 38 shows the complete top-level design.

The three SRAM cells represent the users. The D flip-flops ensure that all votes are synchronized before being processed. The majority circuit makes the final decision based on the outputs of the D flip-flops and passes the result to the counter for tracking.

Below is the schematic design of the top-level circuit.

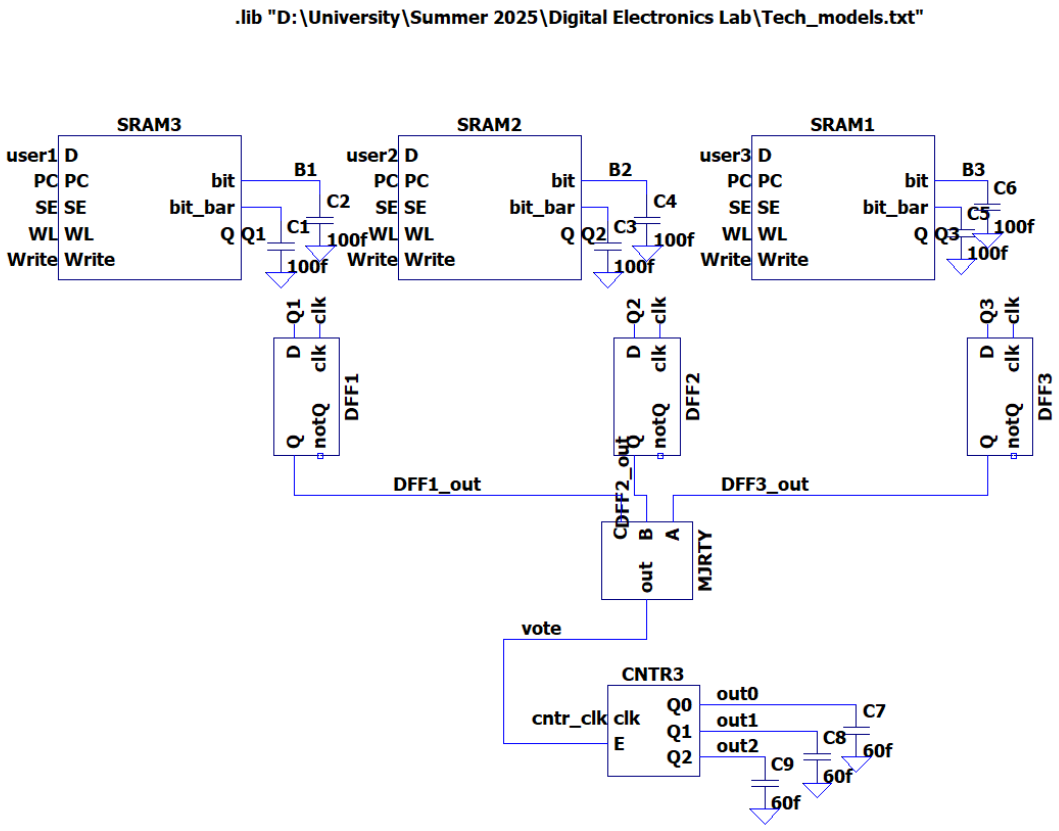

Figure 48. Top-level circuit schematic design.

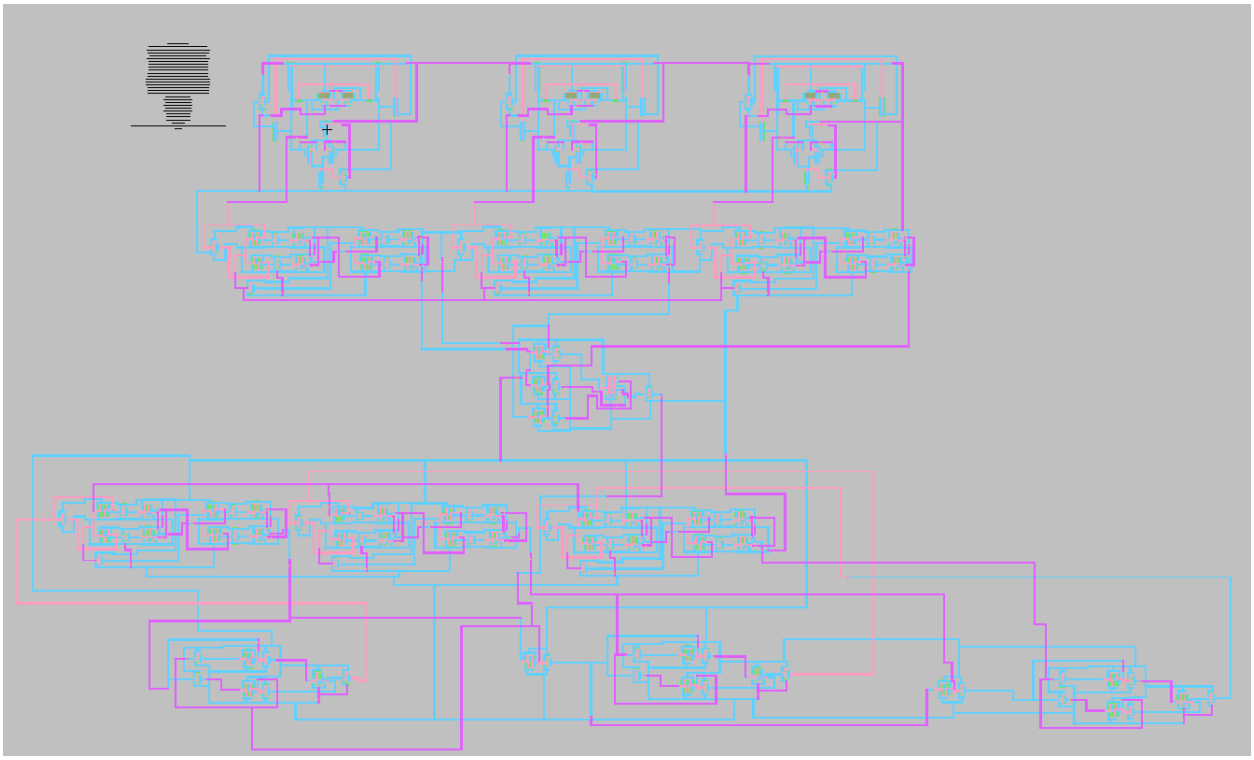

Figure 49. The layout of the top-level circuit.

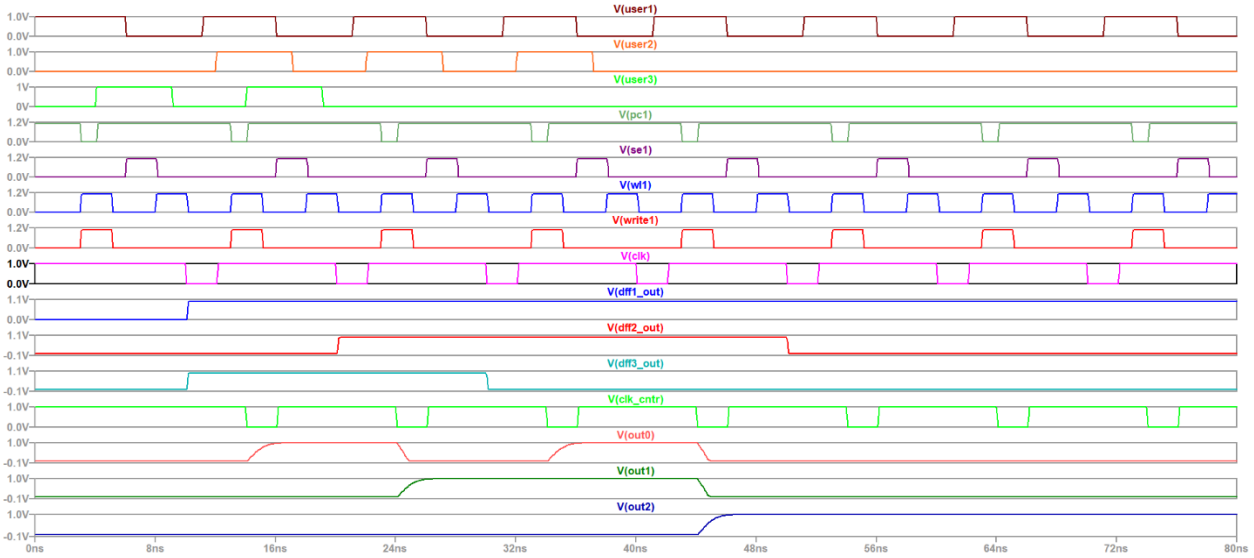

Figure 50.Functionality test of the top-level circuit (layout)

Testing

We tested various cases to ensure the functionality of our system, starting from the simplest test cases (constant inputs). With each test, we gradually increased the complexity. This step-by-step approach made the testing process easier to manage and understand.

Case 1: none of the users votes 1

The first test case examined the all-zeros condition, in which none of the users voted 1 at any given time. Figure 39 displays the signals used to generate this case, along with the counter's output, its clock, and the data flow through the circuit. As expected, the counter remained at zero throughout the simulation.

In the context of the system's analogy, this case represents all the consultants agreeing that the item is not worth bidding on, perhaps because it's a fake artwork.

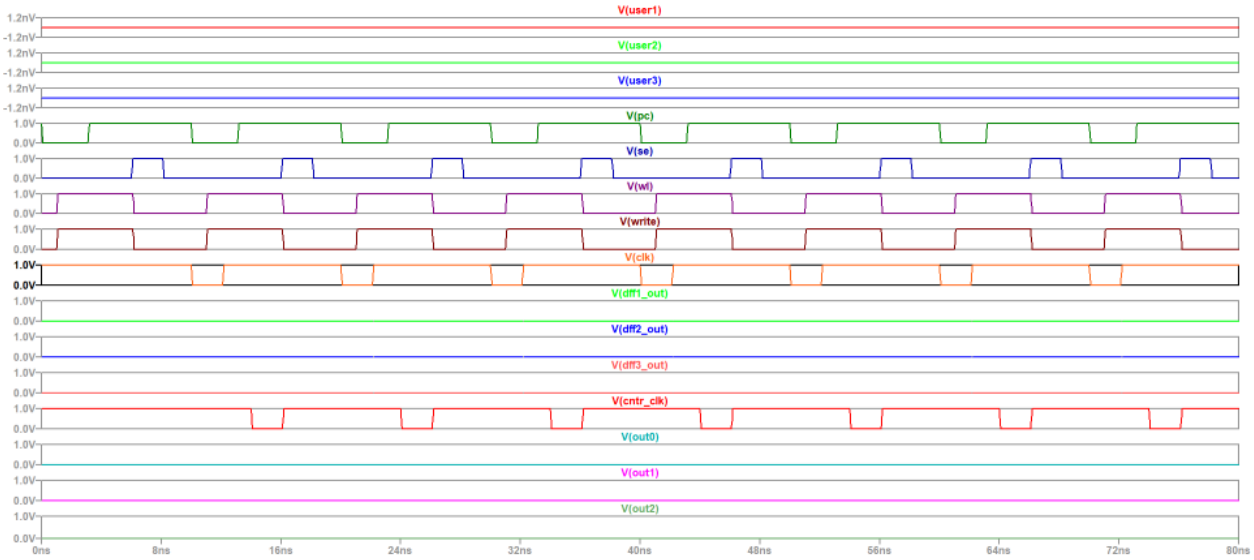

Figure 51. The first test case simulation.

Case 2: all the users vote 1

The second test case examined the all-ones condition, in which all users voted 1 at any given time. Figure 40 displays the signals used to generate this case, along with the counter's output, its clock, and the data flow through the circuit. The simulation duration was adjusted such that the maximum count of 7 could be reached and not exceeded, as we are using a 3-bit counter. As expected, the counter incremented a total of seven times, once on each falling edge of the clock, according to the majority rule.

In the context of the system's analogy, this case represents all the consultants agreeing that the item is very valuable and worth bidding on, perhaps it's a newly discovered Da Vinci painting.

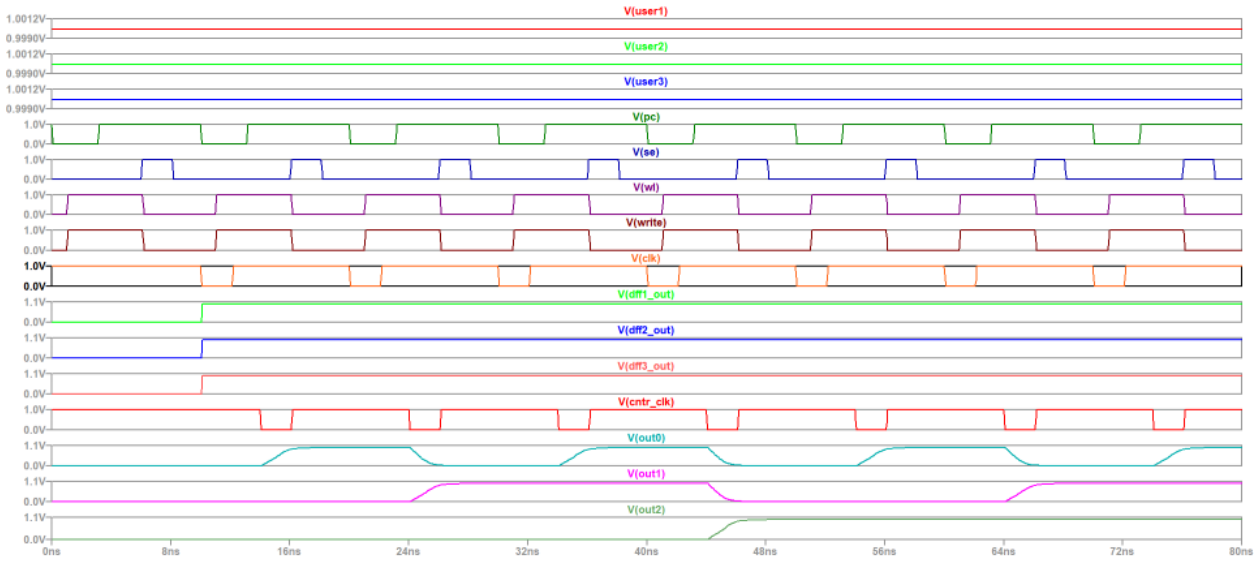

Figure 52. The second test case simulation.

Case 3: one user keeps voting 1, the others keep voting 0

The third test case examined a scnario in which only one user votes 1, while the rest vote 0s. Figure 41 displays the signals used to generate this case, along with the counter's output, its clock, and the data flow through the circuit. The counter remained zero as the majority didn't agree.

In the context of the system's analogy, this case may represent one consultant inssiting on the value of the piece, while the other's disagree. Maybe that one consultant sees something the others don't, or perhaps he has an ulterior motive. Nonetheless, the others are not easily fooled.

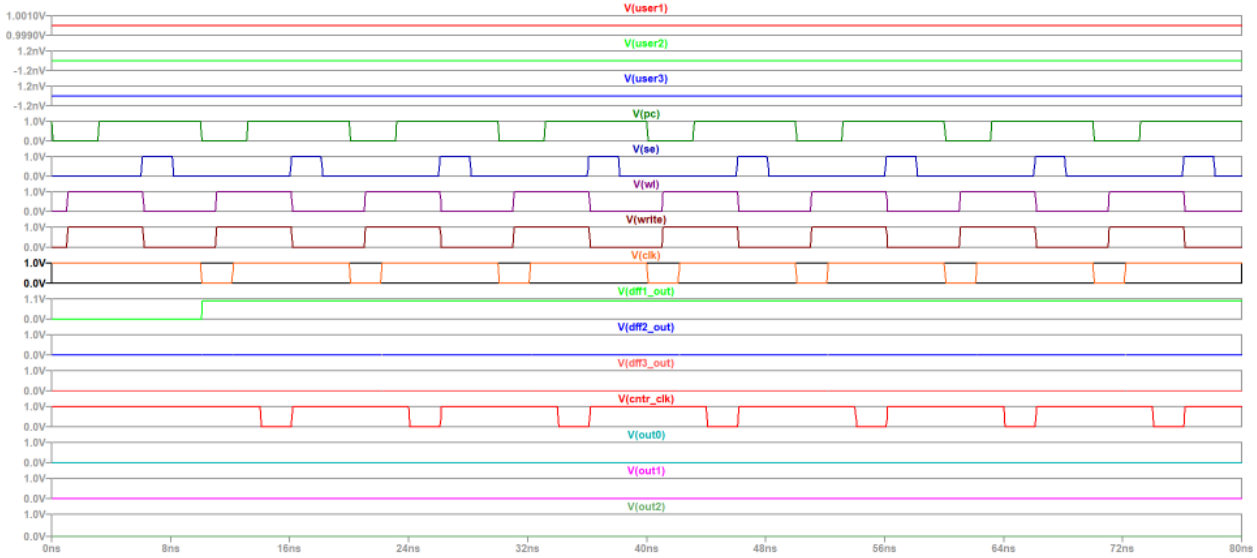

Figure 53. The third test case simulation.

Case 4: two users agree for some time, then they disagree

In this test case, we examined the scenario where 2 users have different opinions throughout the simulation. Figure 42 displays the signals used to generate this case, along with the counter's output, its clock, and the data flow through the circuit. Notice that the users 1 and 2 agree on votimg 3 times, the counter increments 3 times accordingly.

In the context of the system's analogy, the case represents a more common case where the consultants agreed to an extent. One must admit though, the thrid consultant is quite picky.

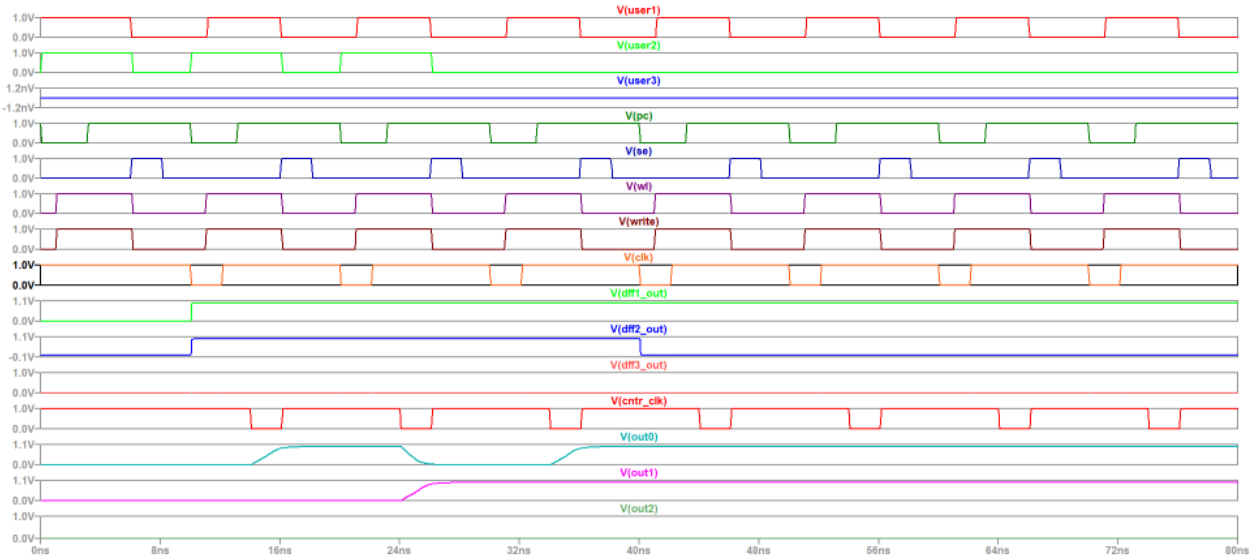

Figure 54. The fourth test case simulation.

Case 5: all three users have different votes

In this case, we examined a scenario where the different users have different votes over time. Figure 43 displays the signals used to generate this case, along with the counter's output, its clock, and the data flow through the circuit. In total, the users agreed 4 times.

In the context of the system's analogy, the case represents a common scenario where the consultants are constantly changing their opinon, as more information about the piece is gradually revealed.

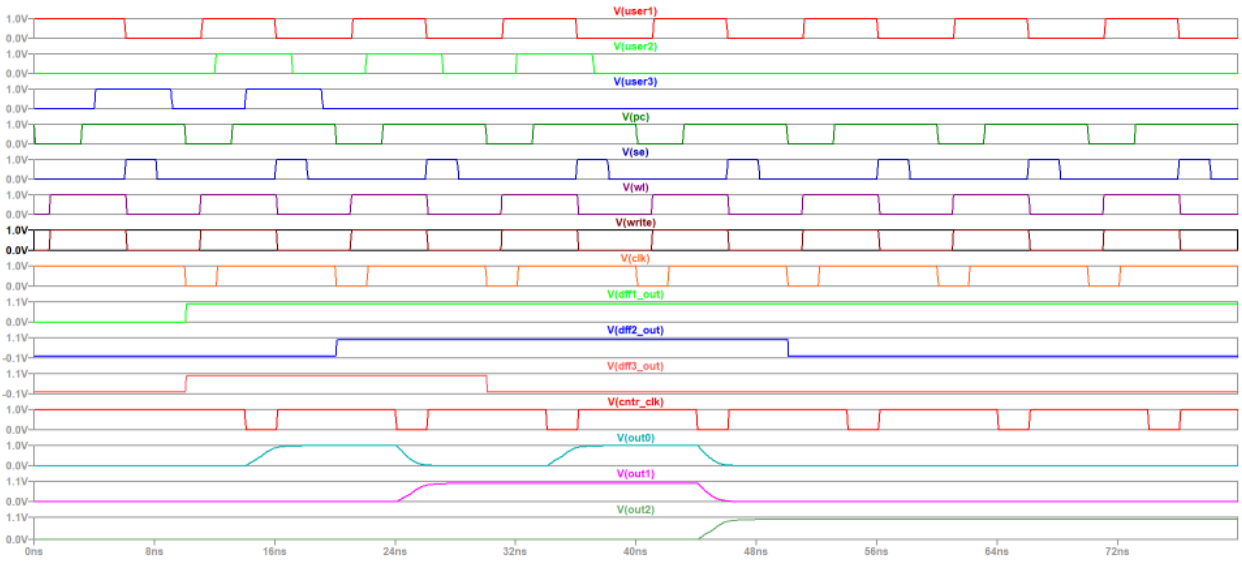

Figure 55. The fifth test case simulation.

Conclusion

Working on this project at both the schematic and layout levels helped us connect theoretical ideas, like majority logic, memory storage, and counters, to actual circuit design on both the transistors and gates levels. Measuring propagation delays and examining waveform outputs gave us a clearer view of how timing impacts the system, and how to test the system effectively.