University of Pennsylvania
Department of Electrical and System Engineering
Digital Integrated Circuits and VLSI Fundamentals
ESE5700, Fall 2024

Final project: Configurable Logic Block

Anh Tran
anhthh@seas.upenn.edu

Linh Chu
linhchu@seas.upenn.edu

March 7, 2025

Contents

1 Introduction **2**

2 16-input Look Up Table (LUT) **2**
2.1 2:1 Multiplexer (MUX) 2
2.2 16 LUT 3
2.3 Delay measurement 5

3 SRAM cell design **6**
3.1 Schematic design 6
3.2 Design verification 8
3.3 SRAM block for Configurable Logic Block (CLB) 13

4 Clock signal generator **15**
4.1 Design 15
4.2 Verification 15

5 D Flip Flop **16**
5.1 Schematic design 16
5.2 Verification 17

6 Serial In Parallel Out (SIPO) Register **18**
6.1 Schematic design 18
6.2 Verification 18

7 Configurable Logic Block (CLB) **20**
7.1 Schematic design 20
7.2 Design verification 21
7.3 Design with symbol 23

8 Metrics measurement **24**
8.1 Frequency 24
8.2 Energy 26
8.3 Area 28
8.4 Figure of Merit (FOM) 29

9 Summary **29**

10 Attempt to optimize the CLB **29**

11 Conclusion **31**

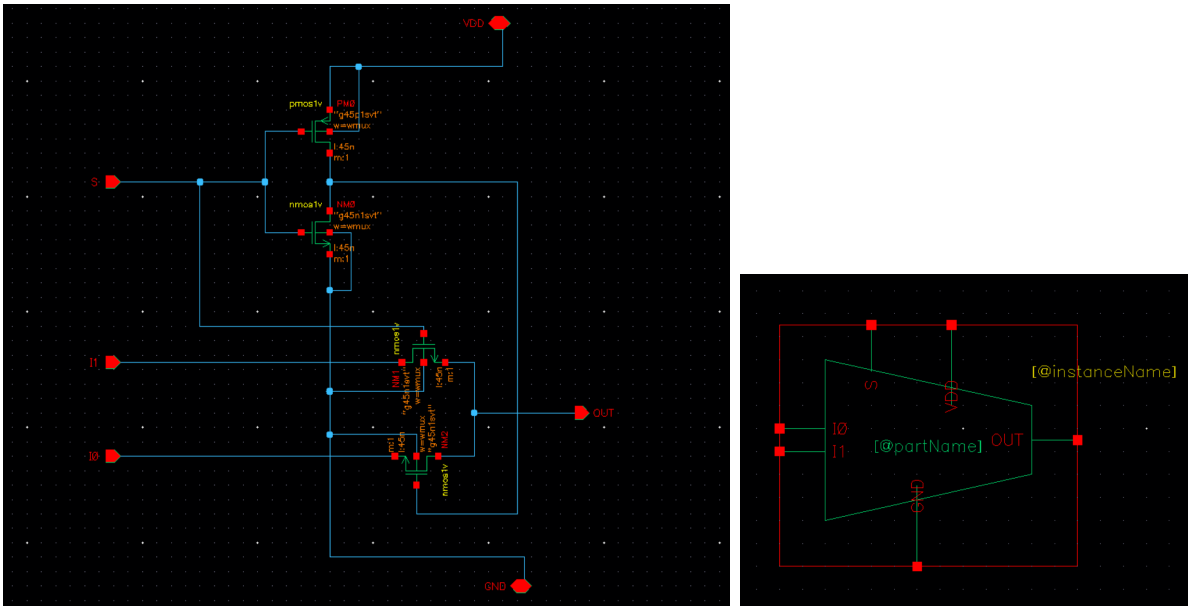
12 Contribution **32**

1 Introduction

Configurable Logic Blocks (CLBs) are fundamental components of Field-Programmable Gate Arrays (FPGAs). CLBs enable rapid prototyping and reconfigurable computing, allowing the hardware's functionality to be dynamically updated without the need for fabricating a new chip. Moreover, CLBs are renowned for their ability to perform high-speed parallel computations. These advantages make CLBs a critical building block that finds widespread applications in various domains, including digital signal processing, communication systems, cryptography, real-time control, and Artificial Intelligence. Designing a CLB involves tackling several challenges, such as managing timing hazards between components like SRAM cells, Look-Up Tables (LUTs), and clock generators to ensure reliable operation. The complexity of the system with its numerous interconnected elements makes optimizing the layout and wiring particularly challenging. Additionally, balancing power efficiency with computational performance is crucial, as increasing flexibility and functionality often result in higher power consumption and reduced efficiency. In this project, we will present our approach to designing a CLB and outline the procedures employed to minimize the Figure of Merit.

2 16-input Look Up Table (LUT)

2.1 2:1 Multiplexer (MUX)



(a) MUX 2:1 schematic (b) MUX 2:1 symbol

Figure 1: MUX 2:1 design

Verification To verify the MUX design, we use the circuit like below

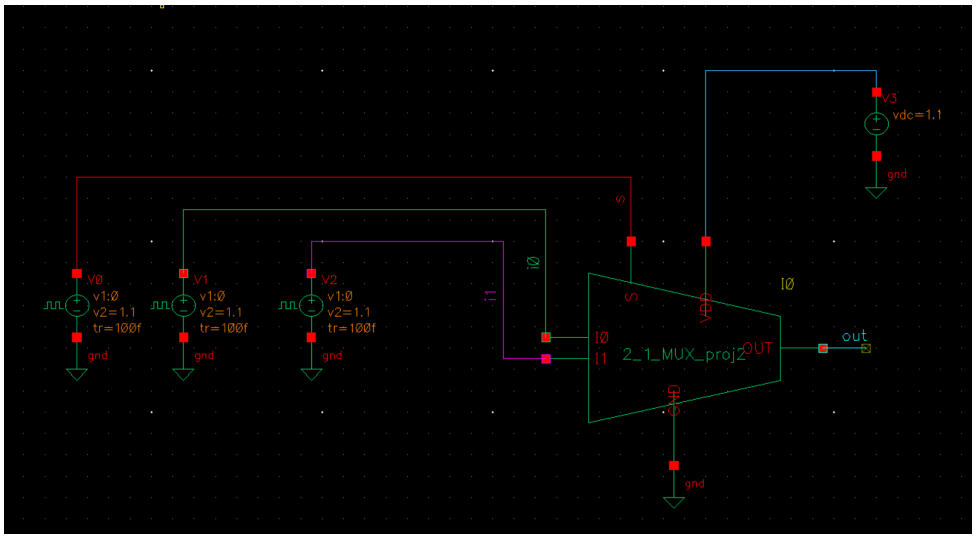


Figure 2: Test circuit for 2:1 MUX

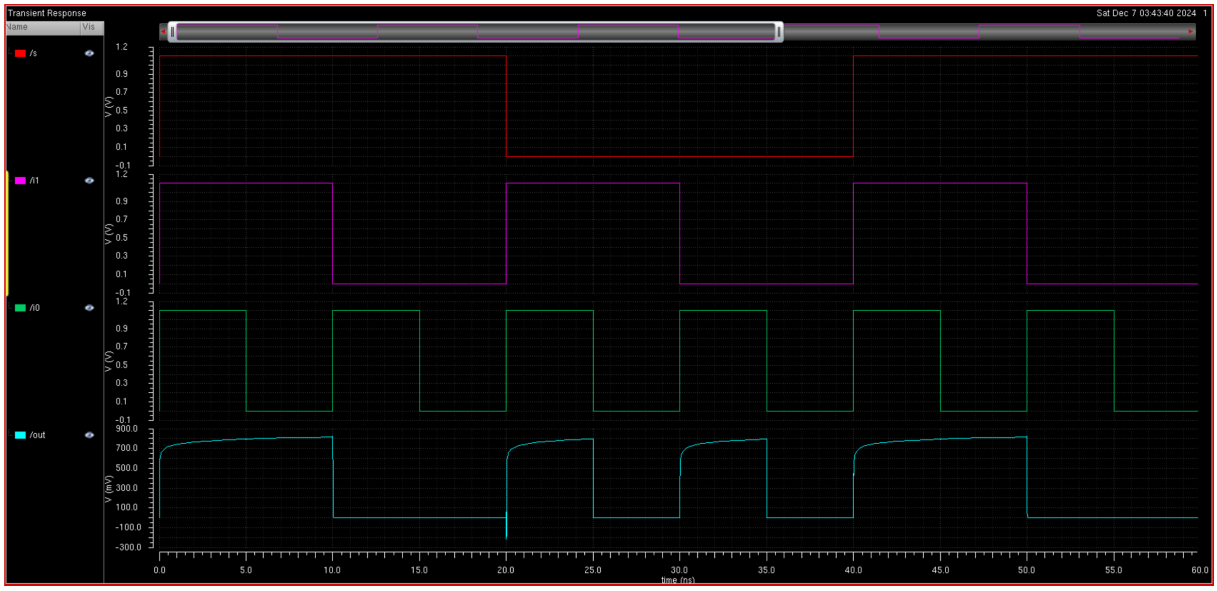
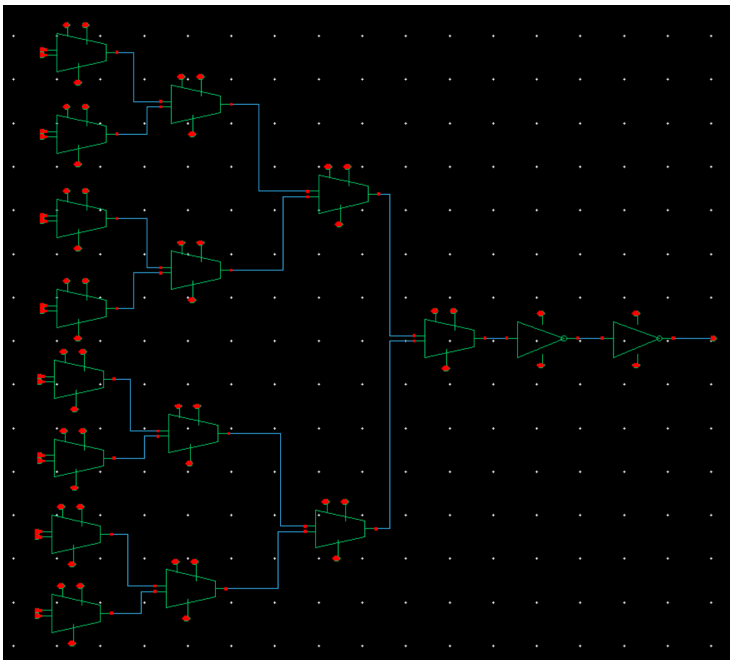


Figure 3: Verification result for 2:1 MUX

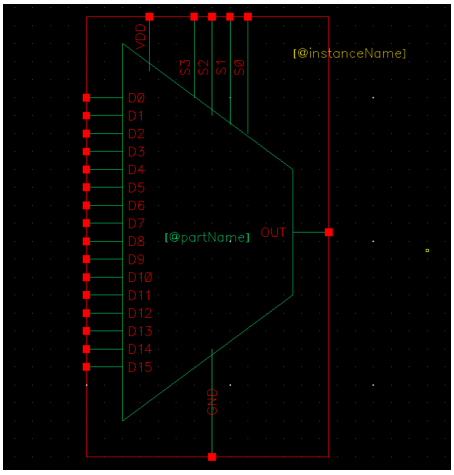
As can be seen from Figure 3, our MUX operates correctly as output follow I_1 when $S = 1$ and I_0 when $S = 0$.

2.2 16 LUT

To create a 16-input LUT (16 LUT), we combine multiple 2:1 multiplexers (MUX) as shown in Figure 4. Additionally, at the output, we use two minimum-sized inverters connected in series. This is done to restore the signal degradation that occurs in each MUX (due to their design using pass transistor logic) without altering the logic of the circuit.



(a) 16 LUT schematic



(b) 16 LUT symbol

Figure 4: 16 LUT design

Verification:
To test the functionality of our LUT circuit, we use the setup shown in Figure 5. The inputs I0, I1, I2, ... I15 are alternately connected to 0 and 1. The S0, S1, S2, S3 ports are connected to pulse sources with different frequencies. By observing the changes in the output corresponding to the S ports, we can verify the operation of the LUT.

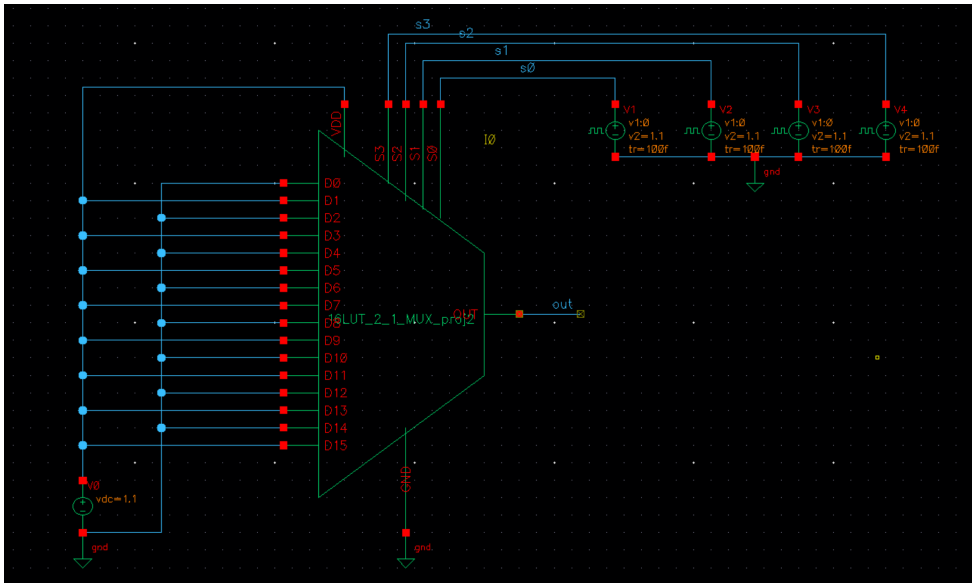


Figure 5: Test circuit for 16 LUT

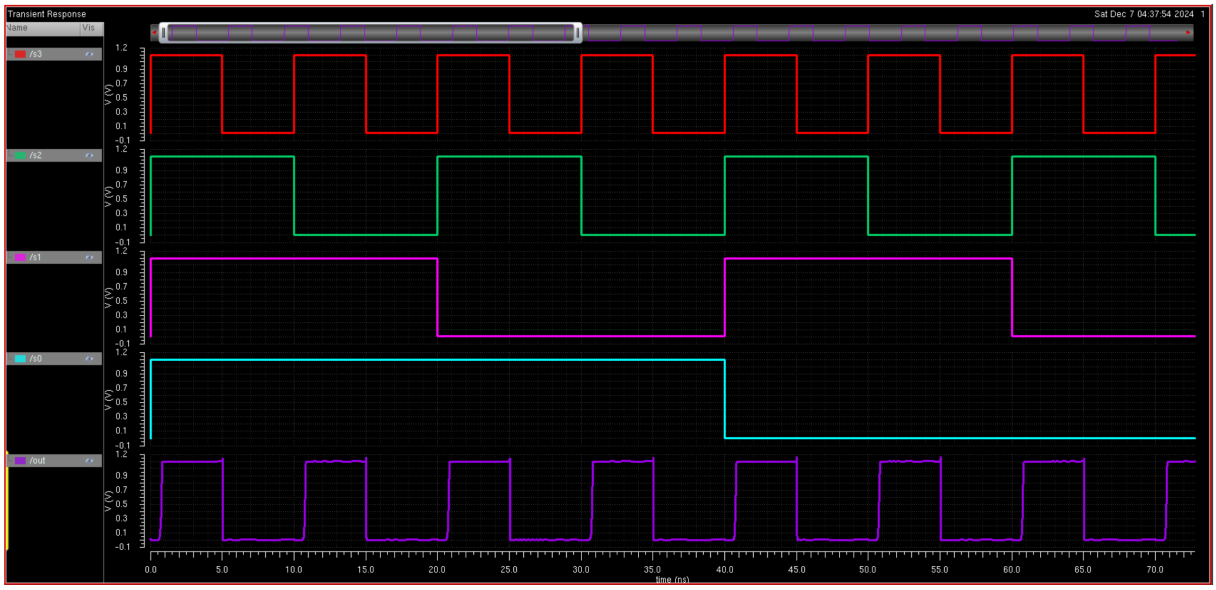


Figure 6: Verification result of 16 LUT

As can be seen from Figure 6, the signals from the S ports have frequencies that double progressively. With this signal pattern, we can test all 16 combinations of S0, S1, S2, and S3. As expected, our LUT correctly selects the appropriate signal from the corresponding I port for output.

2.3 Delay measurement

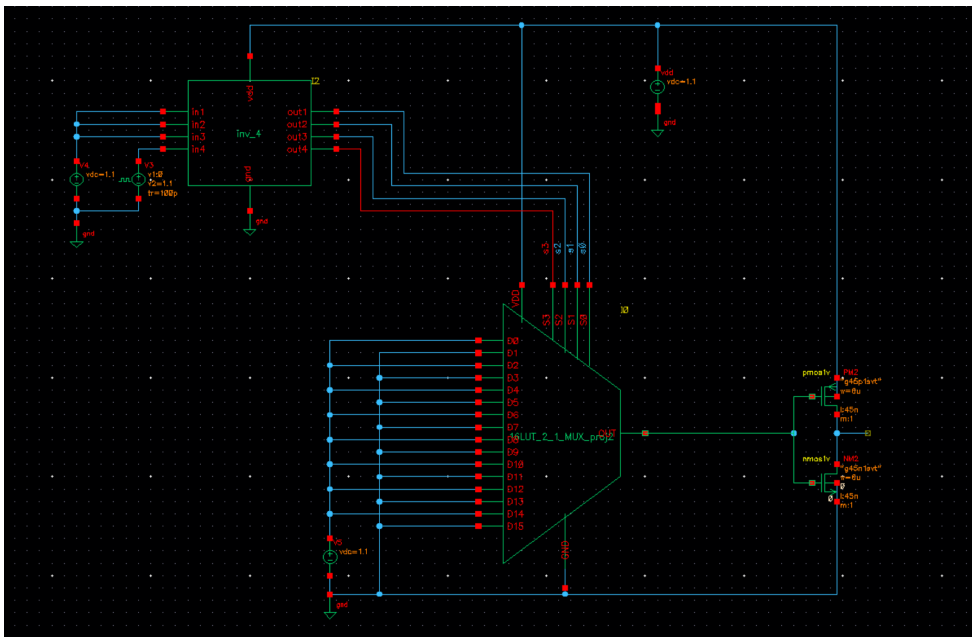


Figure 7: Delay measurement setup for 16 LUT

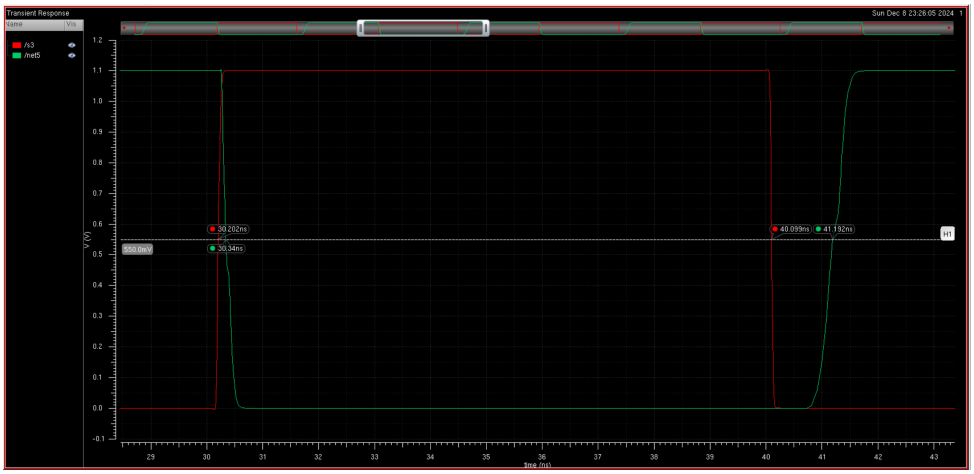


Figure 8: Measured delay of 16 LUT (output is denoted by green color)

From Figure 8, the worst case delay is

$$\tau_{worst-case} = \tau_{pLH} = 41.192 - 40.099 = 1.093(\text{ns}) \tag{1}$$

3 SRAM cell design

For the SRAM cell design, we are using a 6T SRAM cell with proper sizing so that the cell will work reliably. We are also designing the SRAM cell periphery circuit that will work with the cell design.

3.1 Schematic design

The SRAM cell schematic is:

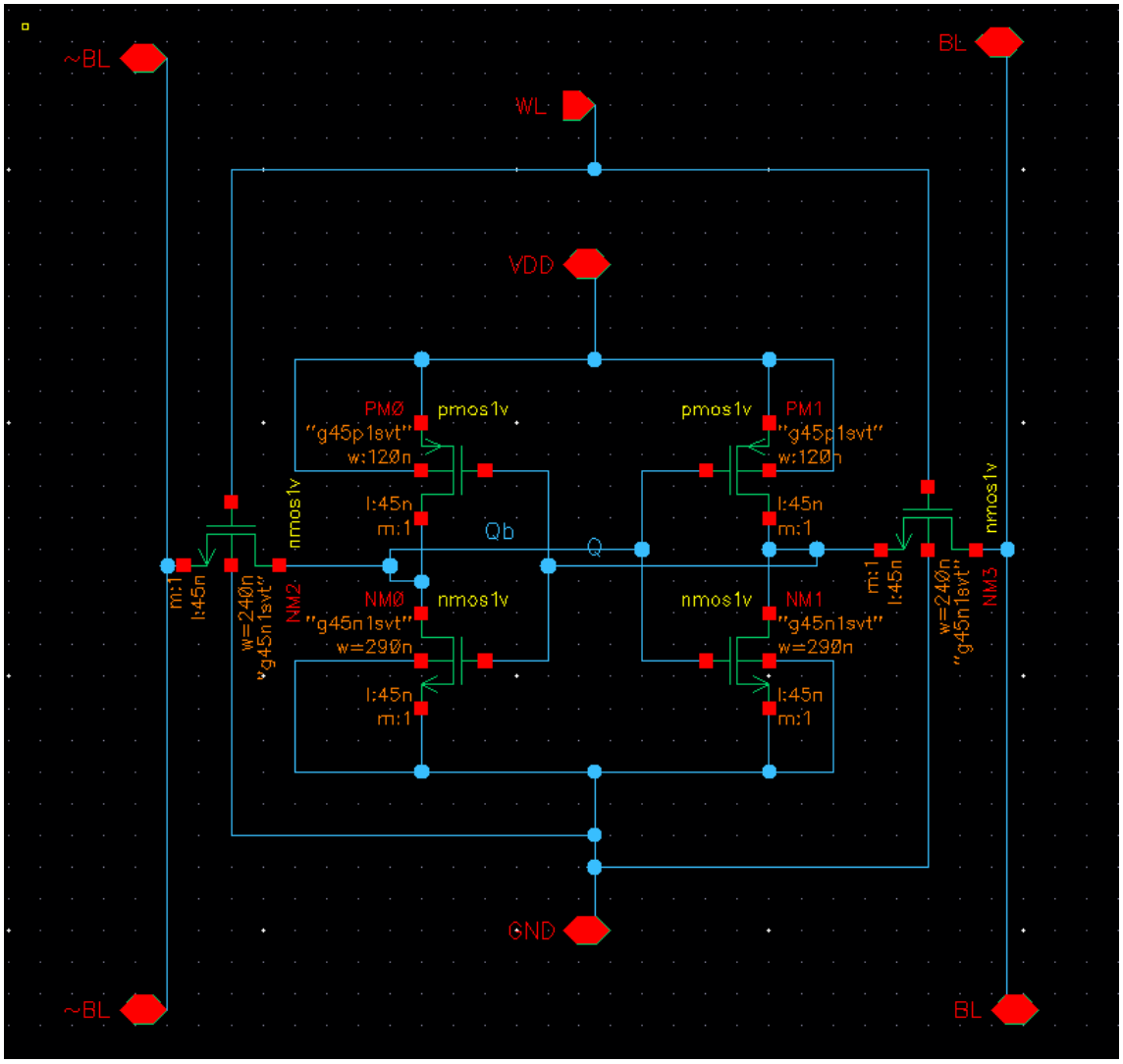


Figure 9: The design of an SRAM cell.

For the 6T SRAM design, we are letting 2 PMOS (M2 and M4) be minimum-sized (l =45nm; w = 120nm). The NMOS M5 and M6 have a length of 45nm and a width of 240nm. The NMOS M1 and M3 have a length of 45nm and a width of 290nm.

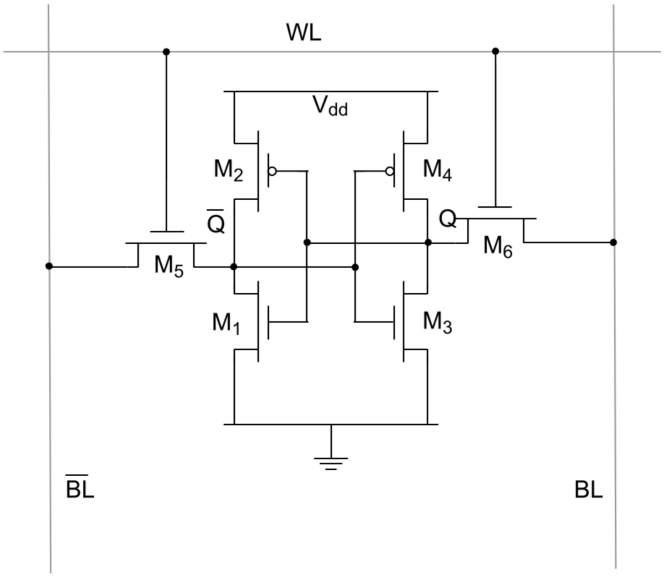


Figure 10: The schematic of SRAM cell (Taken from ESE 5700 Fall 2024 lecture 19).

The symbol of 1 SRAM cell is shown below:

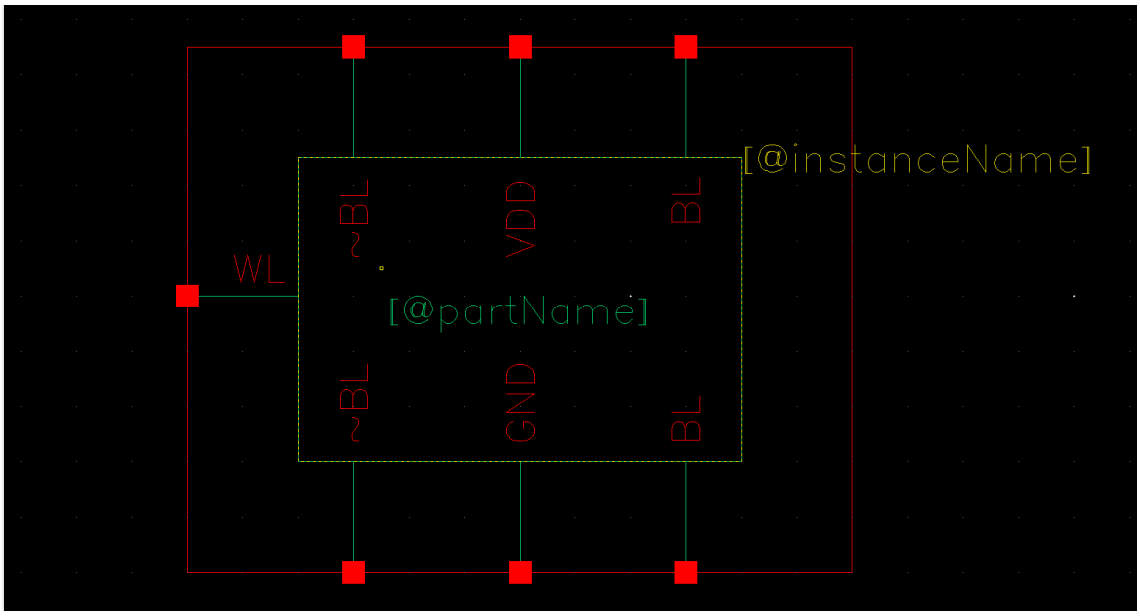
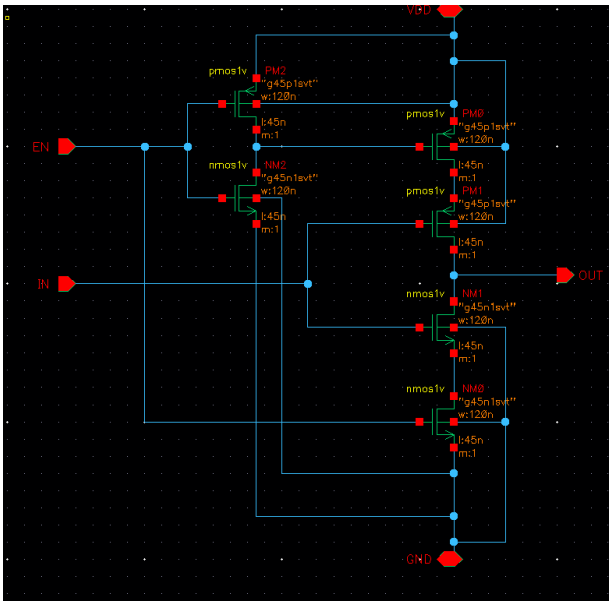


Figure 11: The symbol of SRAM cell.

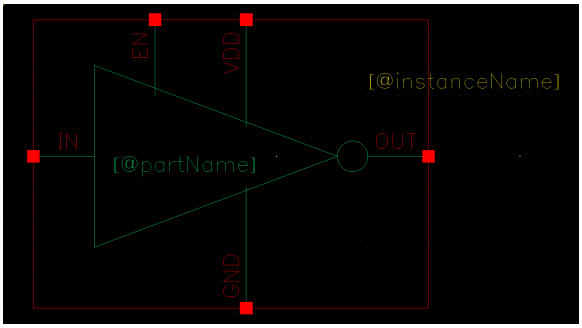
3.2 Design verification

For the verification, we build the SRAM periphery circuit: For read operation, we use the design with precharge to VDD; For the write operation, we use the tristate buffer and tristate inverter.

- The schematic and the symbol of the tristate inverter are shown below:



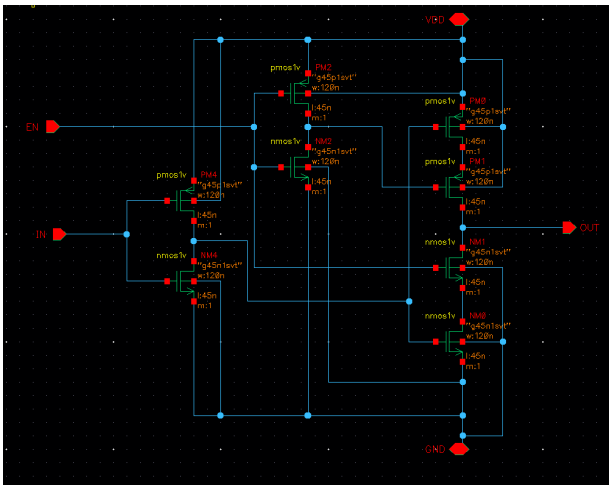
(a) Tristate inverter schematic.



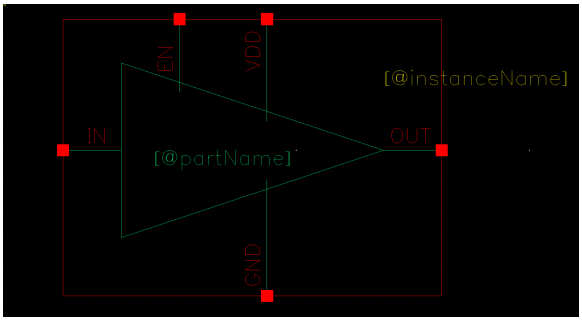
(b) Tristate inverter symbol

Figure 12: Tristate inverter design

- The schematic and the symbol of the tristate buffer are shown below:



(a) Tristate buffer schematic.



(b) Tristate buffer symbol

Figure 13: Tristate buffer design

- The control signal for the SRAM is shown below:

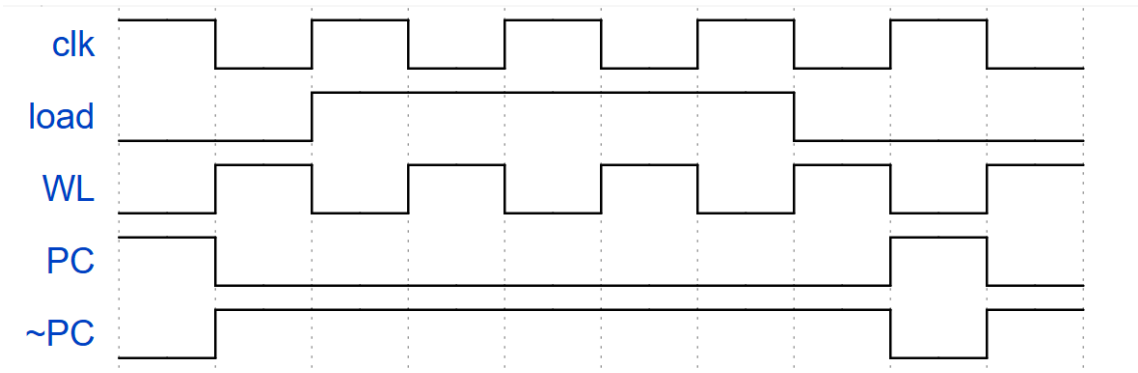


Figure 14: The control signal for the SRAM cell.

With this, we can see that the wl $=\sim$ clk, and $\sim$ PC $=$ wl $+$ load. The generation of the control signal is:

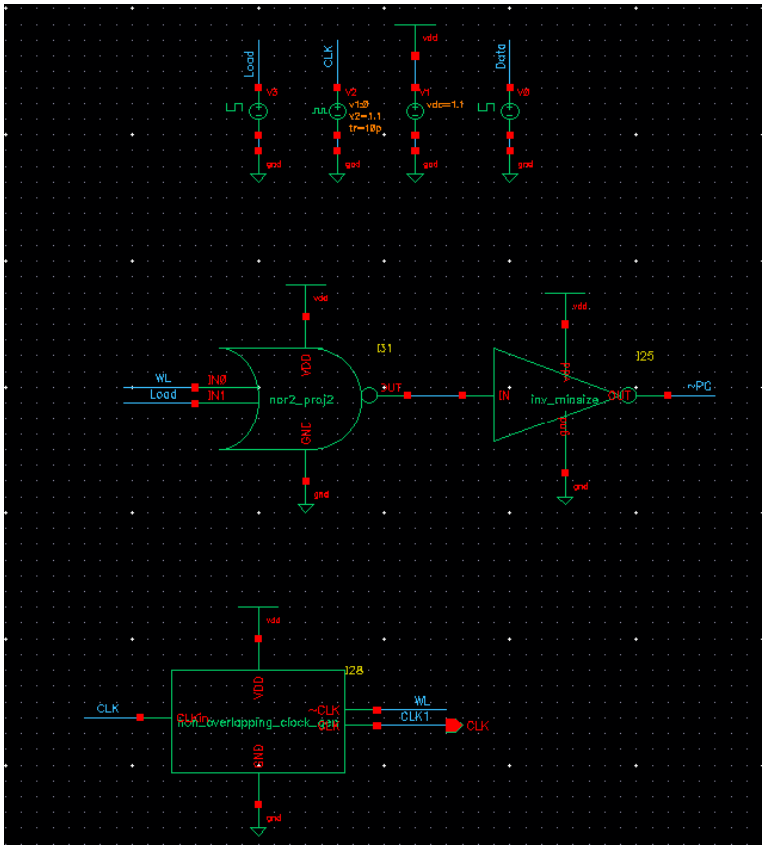
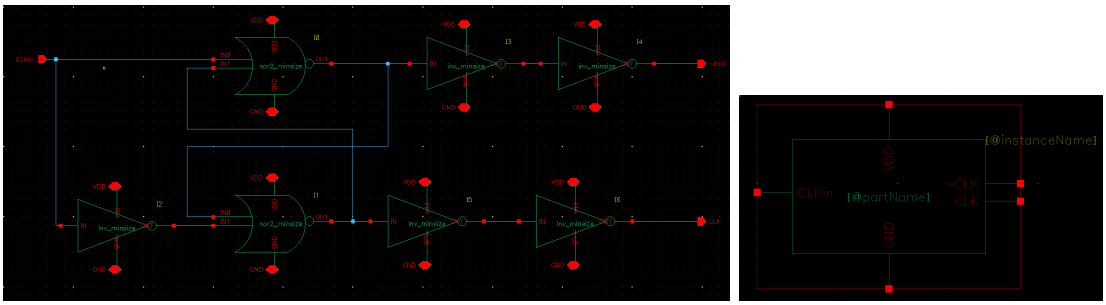


Figure 15: The generator of the control signal for the SRAM cell.

As we have our wl =∼ clk, we include a non-overlapping clock generator. Its schematic and symbol are:



(a) Non-overlapping clock generator schematic. (b) The symbol

Figure 16: Non-overlapping clock design

The whole test schematic of the SRAM cell with the SRAM peripheral is:

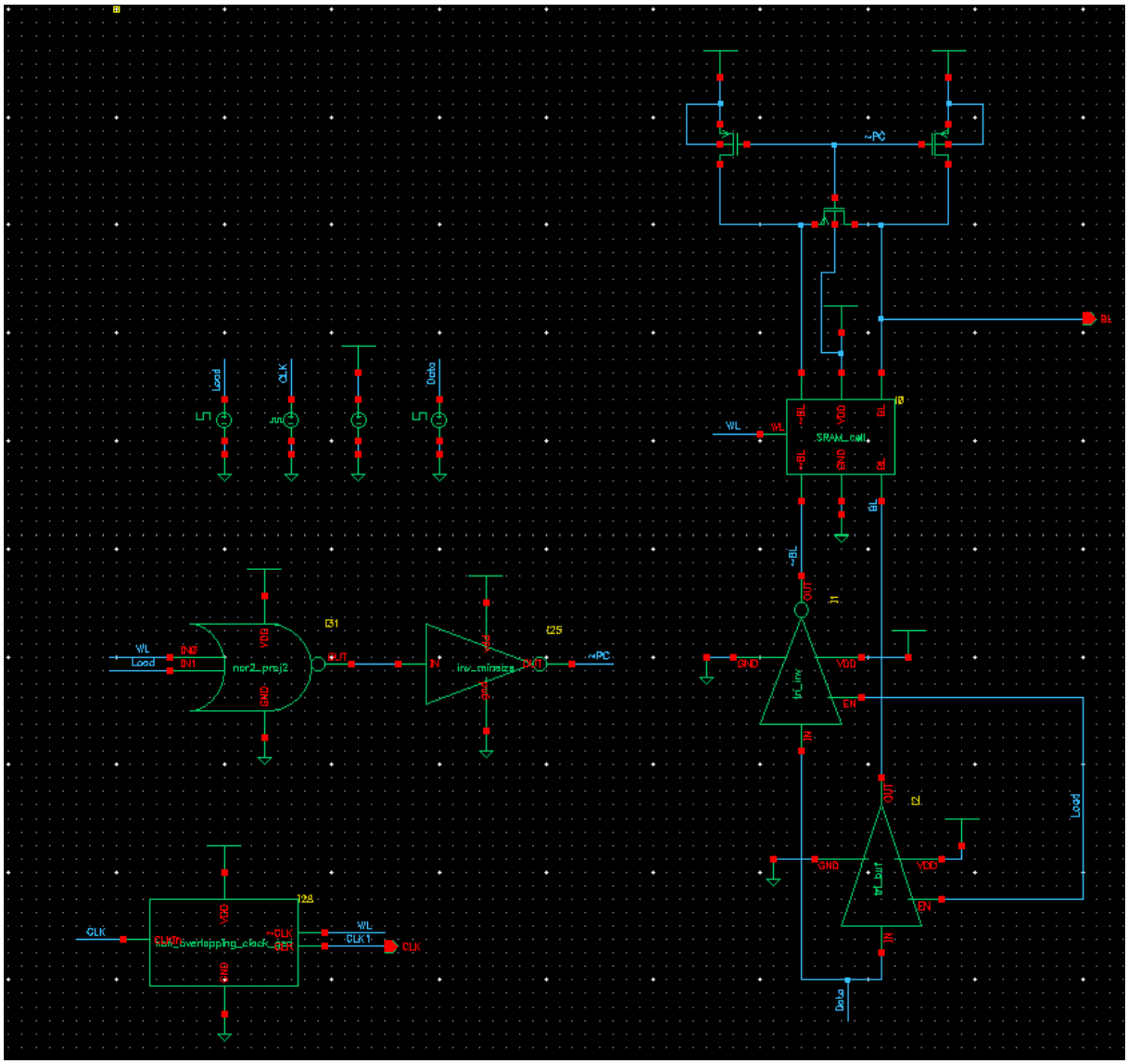


Figure 17: The test schematic of the SRAM cell.

For our design, we use a precharge to Vdd peripheral circuit for read operation, and we use tristate buffer and tristate inverter for write operation.
The verification result of the SRAM cell is shown below:

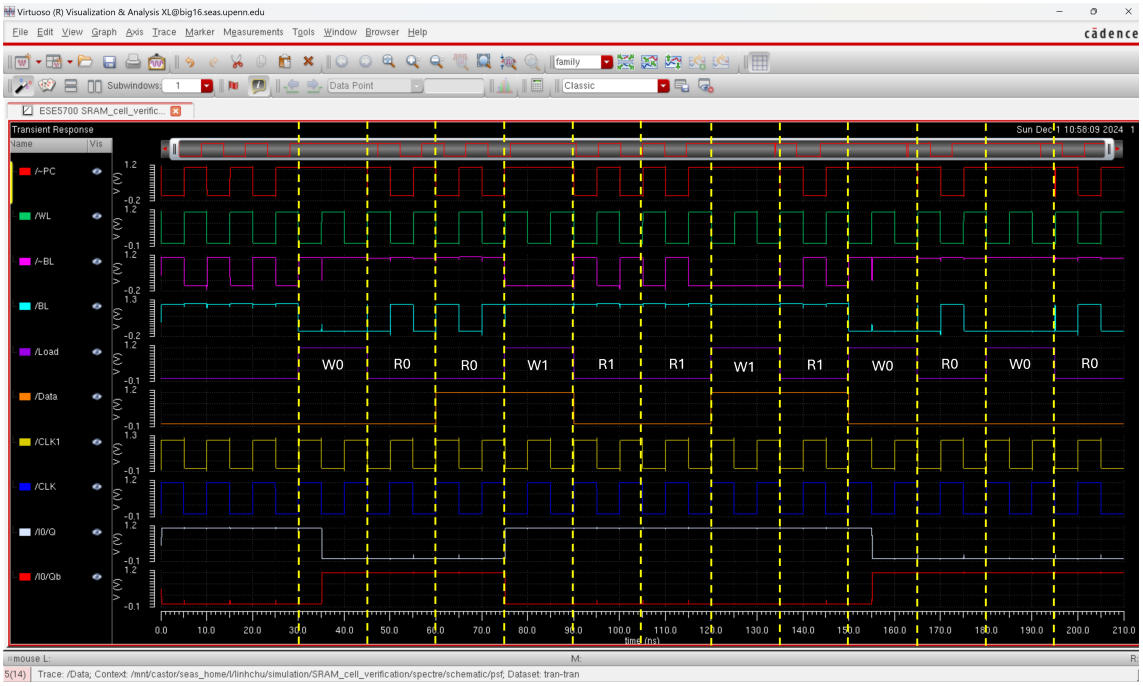
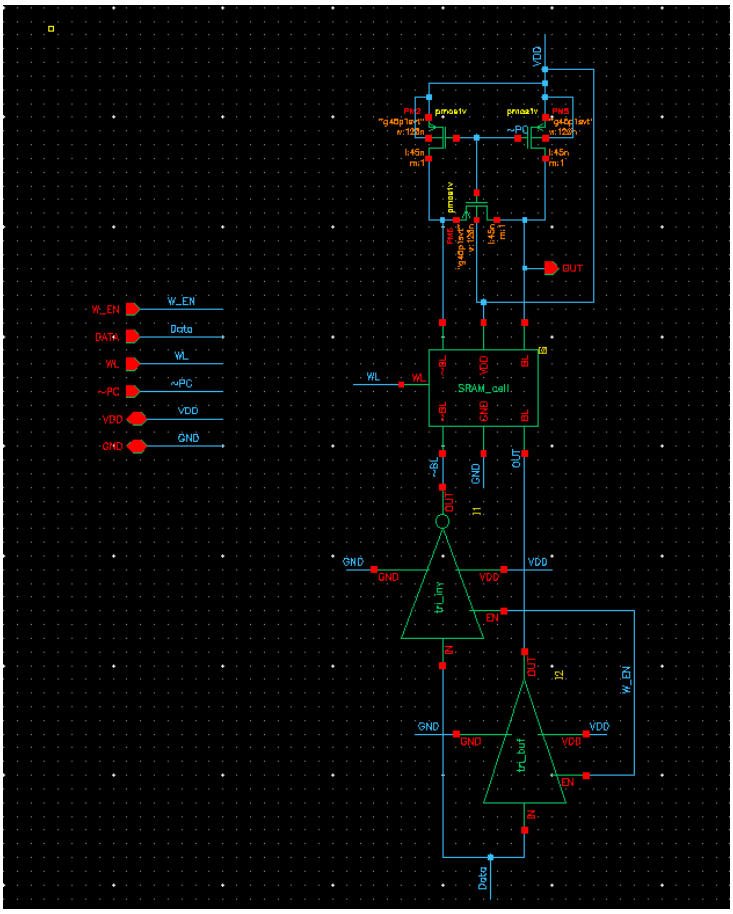
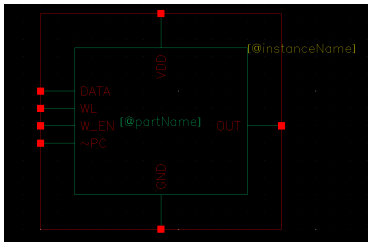


Figure 18: The verification result of the SRAM cell.

For this verification process, we are using the test case of write 0, read (0), read (0), write 1, read (1), read (1), write 1, read (1), write 0, read (0), write 0, read (0). From the verification simulation result, we can see that the SRAM is working correctly.
Thus, we create a symbol for the SRAM cell which includes the peripheral of the SRAM:



(a) SRAM cell with peripheral schematic.

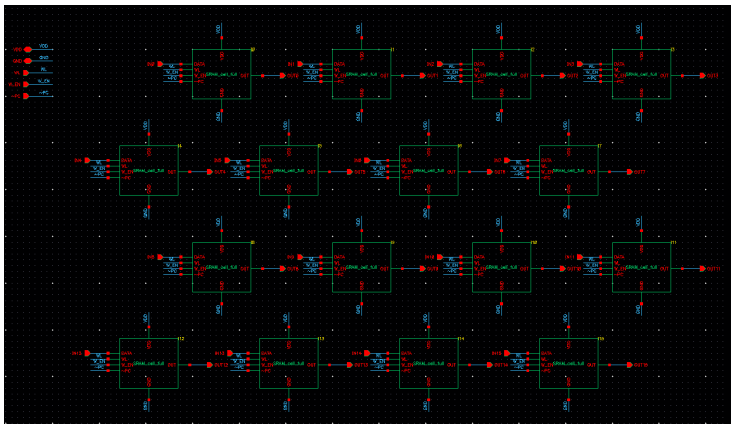


(b) The symbol of the SRAM cell with peripheral

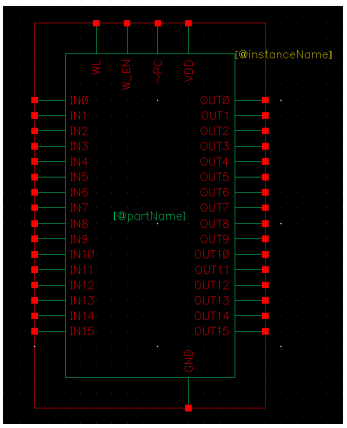
Figure 19: SRAM cell full design

3.3 SRAM block for Configurable Logic Block (CLB)

For the CLB, we design a block of memory that includes a single word of 16 bits. This means our SRAM block will have only 1 word line for all 16 bits and 16 bitlines, one for each SRAM cell. The design in Cadence is shown below:



(a) SRAM block schematic.



(b) The symbol

Figure 20: SRAM block design

We also create a test schematic to ensure that the operation works correctly. As we

already tested the functionality of the single SRAM cell and it works correctly, here, we only test if the SRAM block can read and write each bit synchronously.

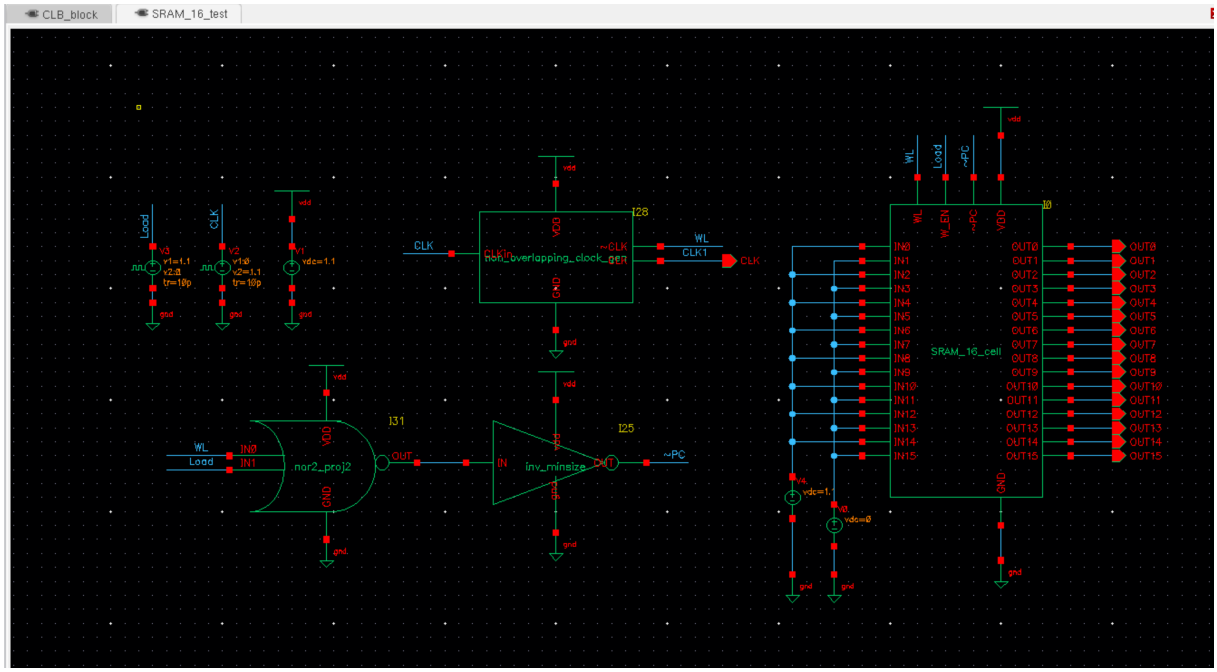


Figure 21: The test schematic of the SRAM block.

The result of the simulation is:

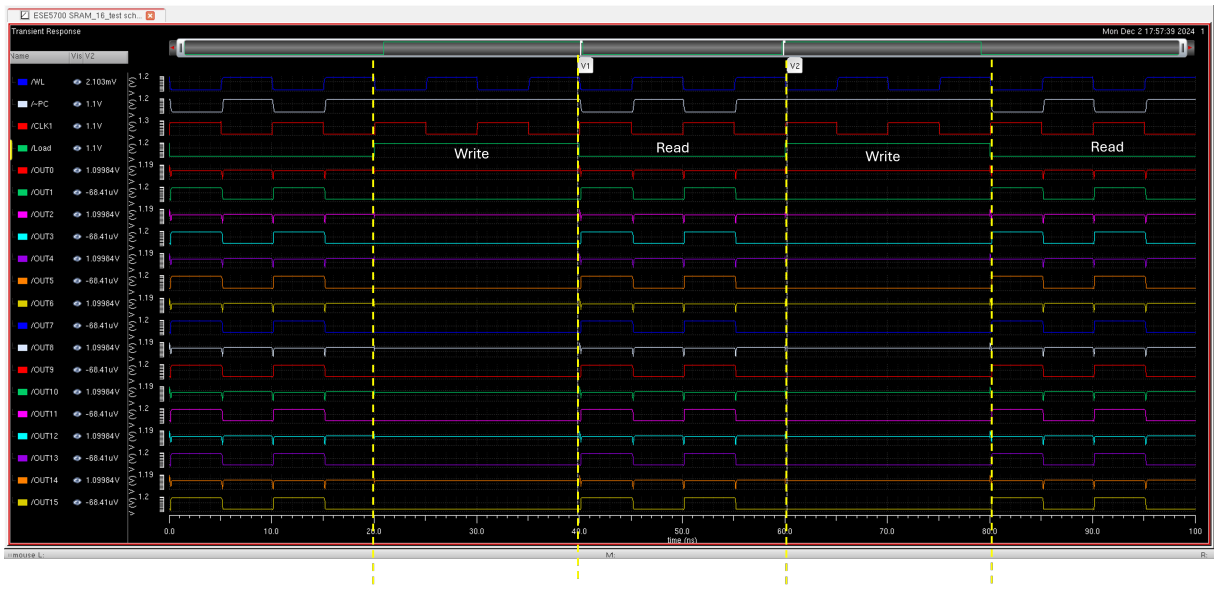


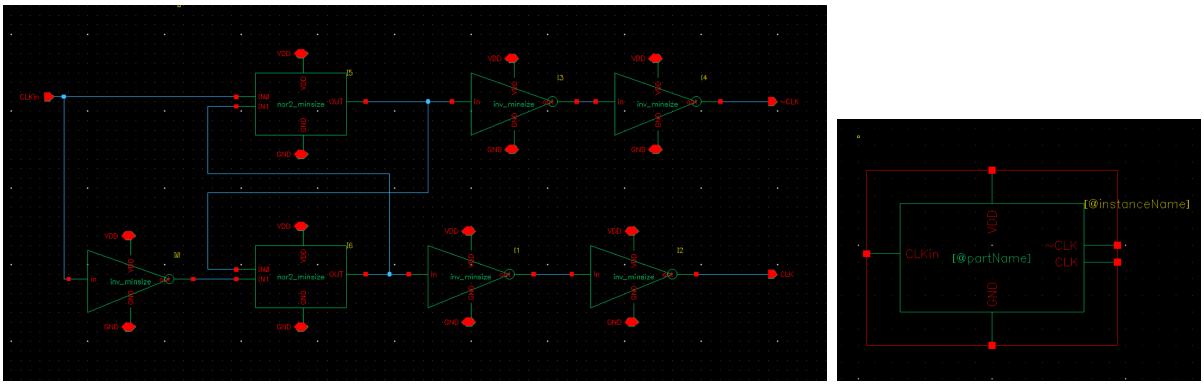
Figure 22: The simulation result of the SRAM block.

From the simulation, we can see that the SRAM block working correctly.

4 Clock signal generator

4.1 Design

In this section, we design a circuit that generates two non-overlapping clock signals (CLK and $\tilde{\text{C}}$LK) from one single input clock (Figure 23).



(a) Clock generator circuit (b) Clock generator symbol

Figure 23: Non-overlapping clocks generator

4.2 Verification

The clock generator can be verified using the circuit below

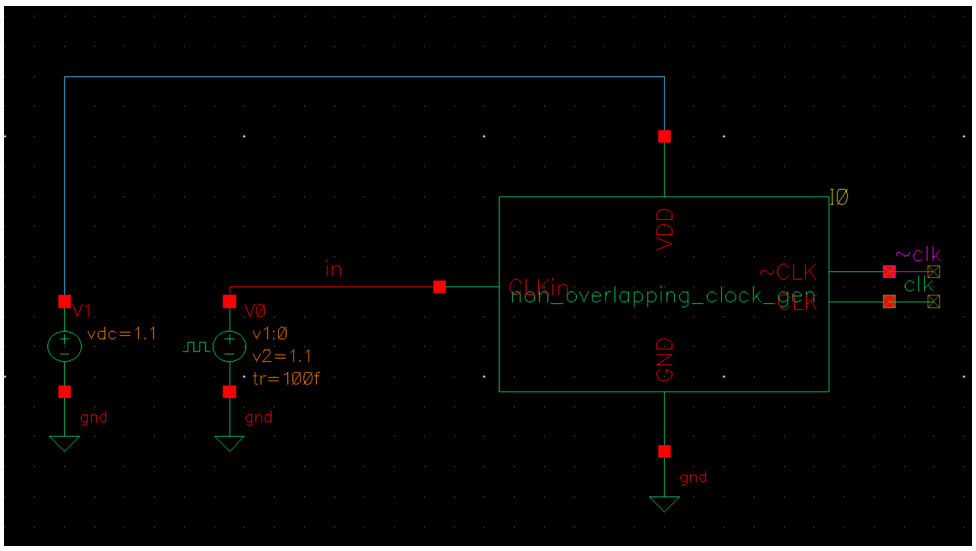


Figure 24: Clock generator test circuit

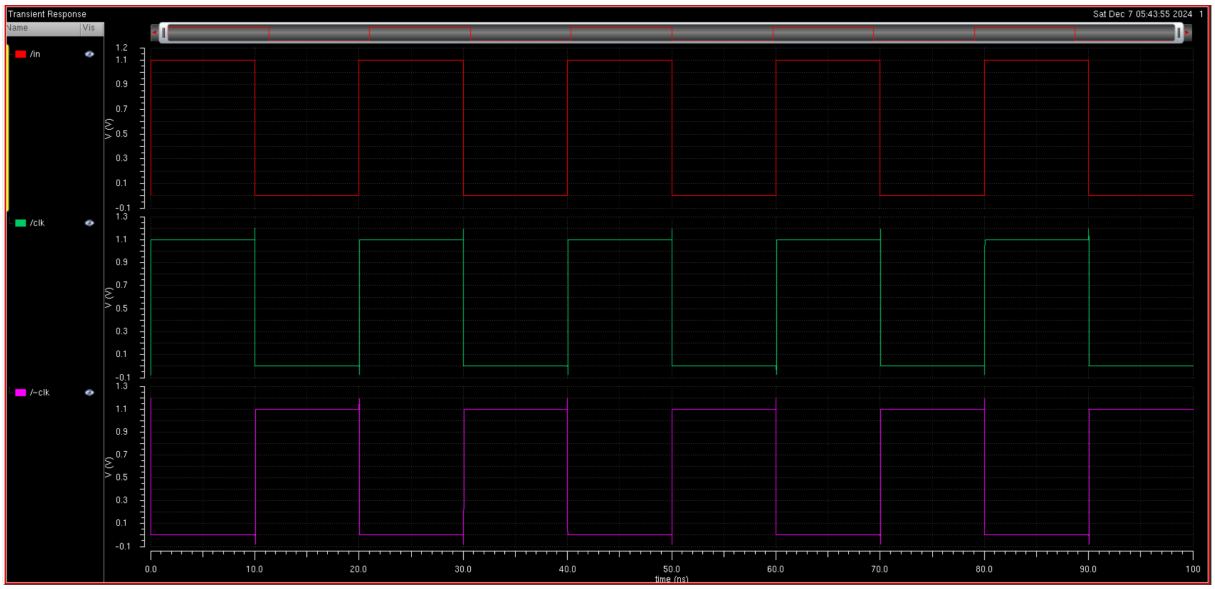


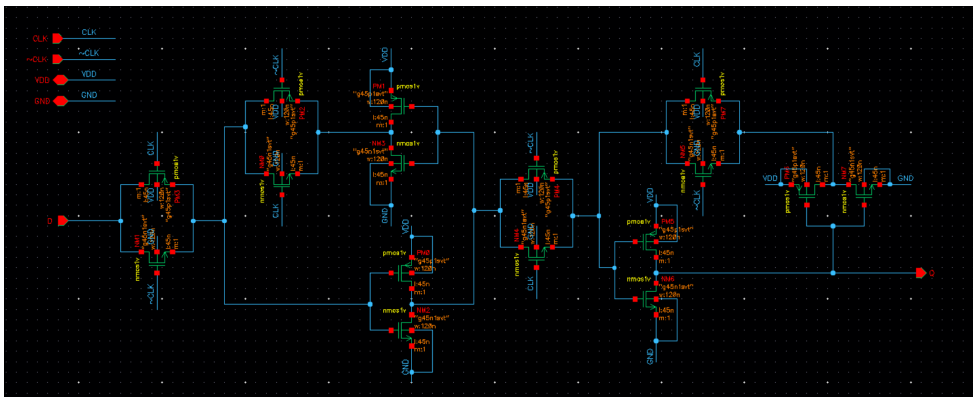
Figure 25: Clock generator verification result

Figure 25 demonstrates the correctness of the clock generator as it produces two clock signals that are out of phase, non-overlapping, and without any loss of magnitude.

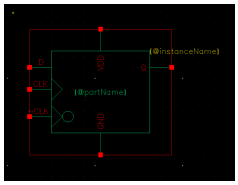
5 D Flip Flop

5.1 Schematic design

A D flip-flop is constructed using inverters and pass transistor logic as shown in Figure 26.



(a) D Flip FLop schematic



(b) Symbol

Figure 26: D Flip FLop design

5.2 Verification

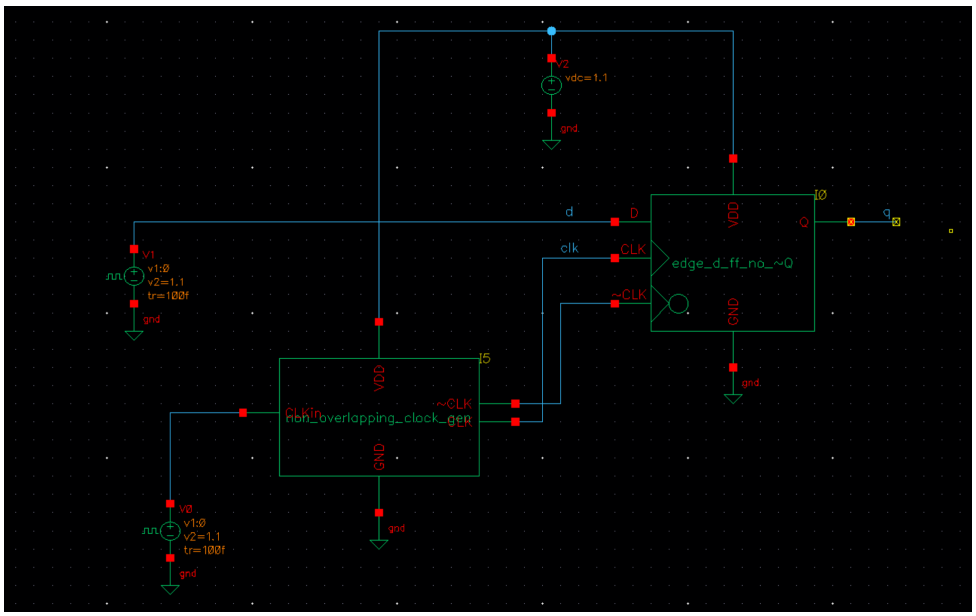


Figure 27: Test circuit for D flip flop

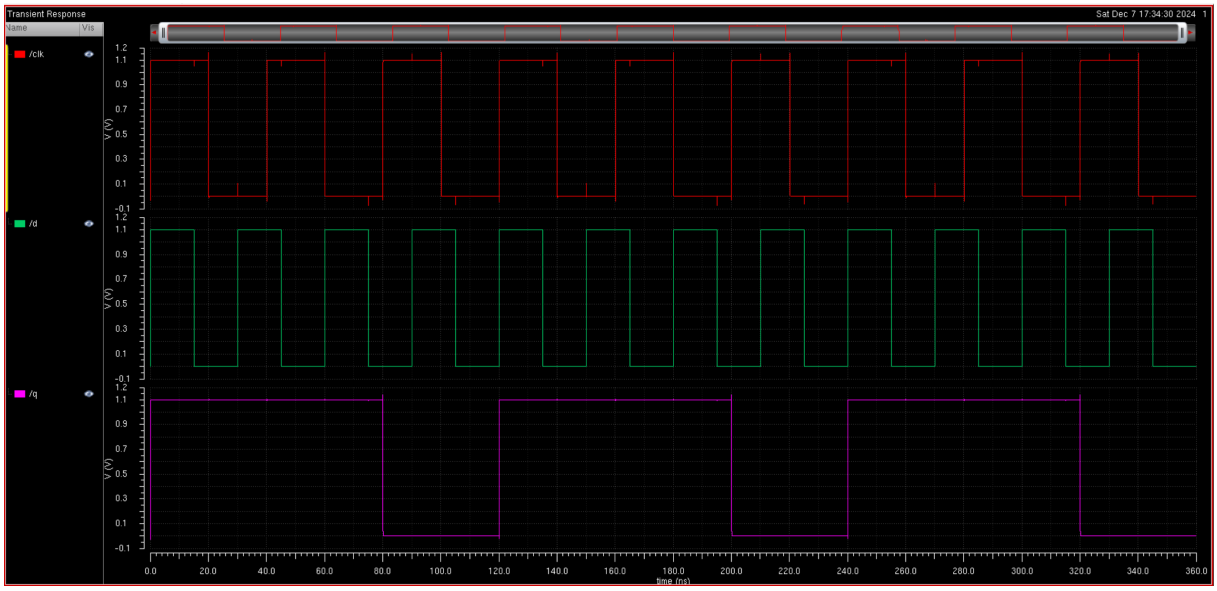


Figure 28: Verification result of D flip flop

As can be seen from Figure 25, output Q is updated by input D at the positive edges; while it stays the same elsewhere. Additionally, output $\tilde{Q}$ is inversion of Q, as expected. We also tried to see what is the setup time of the D flip-flop. As the D flip-flop is already working, we try to get the data as close as possible to the rising edge of the clock. Below is the simulation that we got.

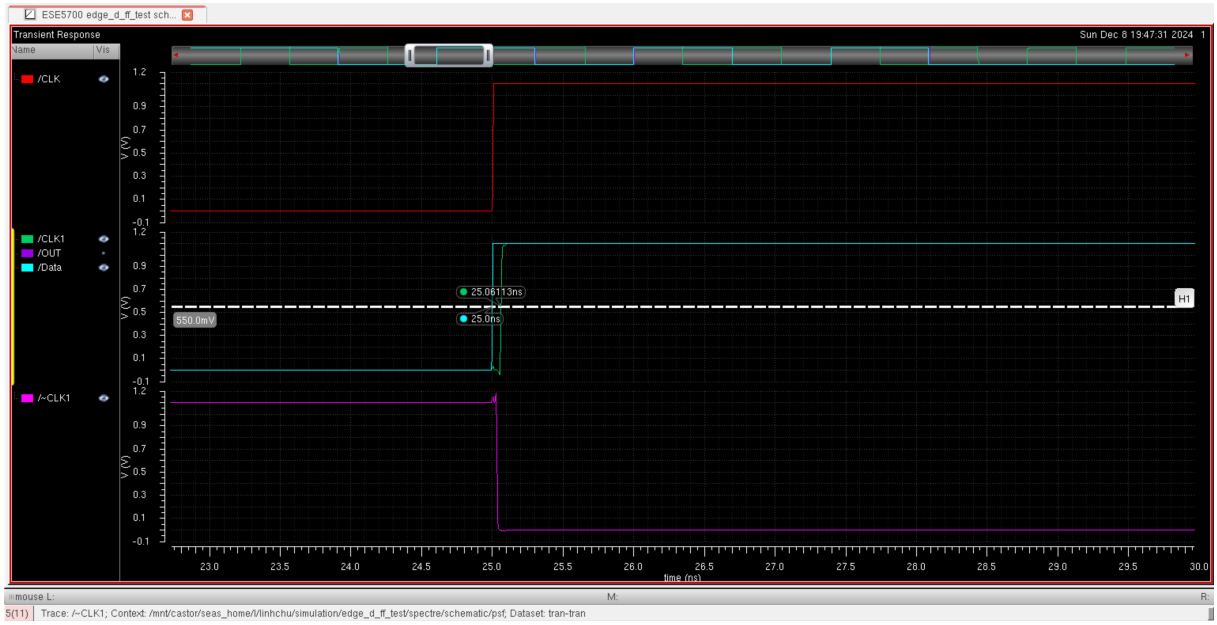


Figure 29: The setup time of D flip flop

From the figure, the setup time of D flip-flop is:

$$t_{setup} = 25.06113 - 25 = 0.06113ns = 61.13ps$$

6 Serial In Parallel Out (SIPO) Register

6.1 Schematic design

We utilize a Serial-In Parallel-Out (SIPO) shift register to input a serial bitstream as we are using a design of SRAM with 1 word of 16 bits. The parallel output from the shift register is directed to an SRAM array. This can be implemented by cascading 16 D flip-flops in series, where the parallel outputs at each stage are connected to 16 cascaded SRAM units (as Figure 30).

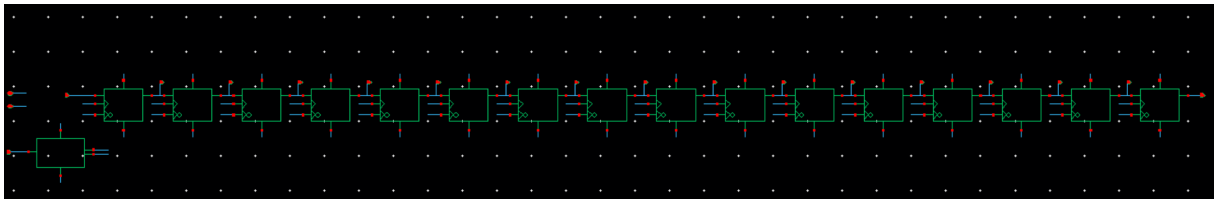


Figure 30: Serial-In Parallel-Out (SIPO) design

6.2 Verification

To verify the SIPO, we let our data input be 1100 1010 0101 1010. We would expect a parallel output from D0 to D15 in the same order as the serial input. Below is the schematic to verify the function of the SIPO.

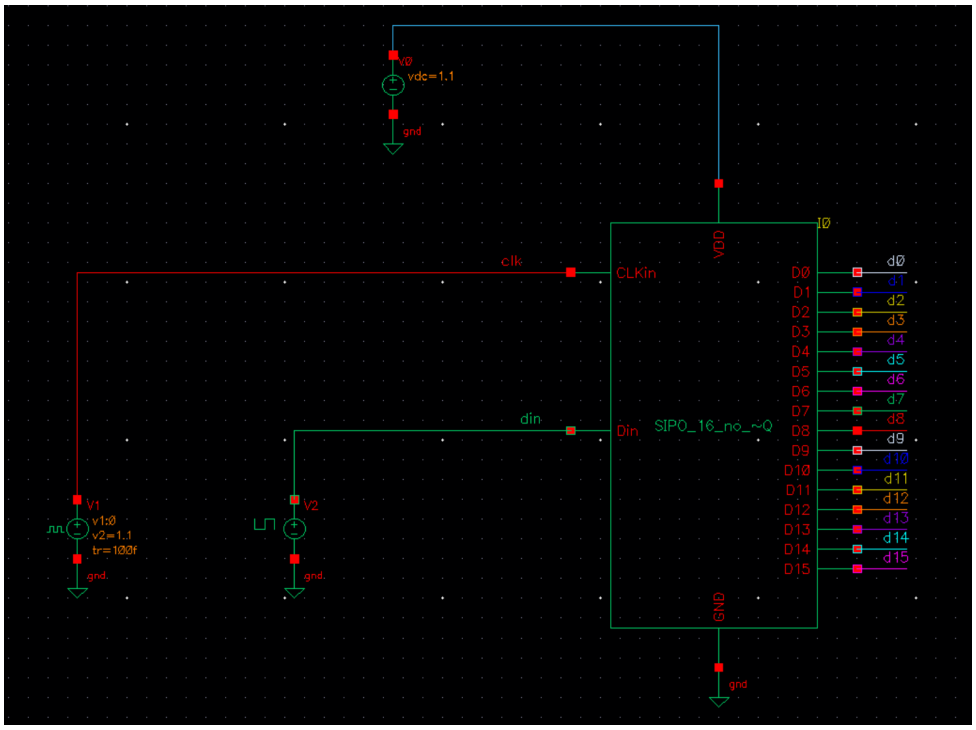


Figure 31: Test circuit for Serial-In Parallel-Out register

The simulation result is shown below:

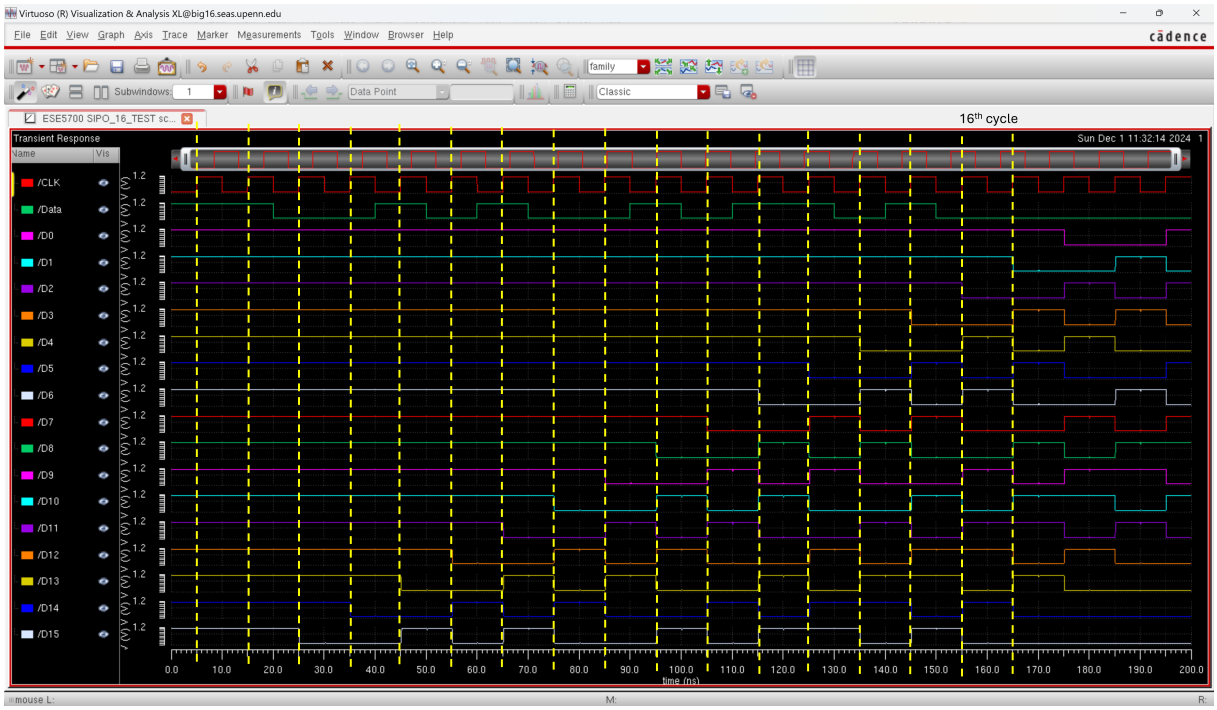


Figure 32: Verification result for Serial-In Parallel-Out register

From the verification result, we can see that after the 16th cycle, all the data has been set correctly as the serial data input.

7 Configurable Logic Block (CLB)

7.1 Schematic design

The CLB consists of an SRAM block, a Look-up Table, and a D flip-flop. As our input data is serial data, we also need a SIPO to create a parallel input for the designed SRAM. The design of the control signal is the same as 14.

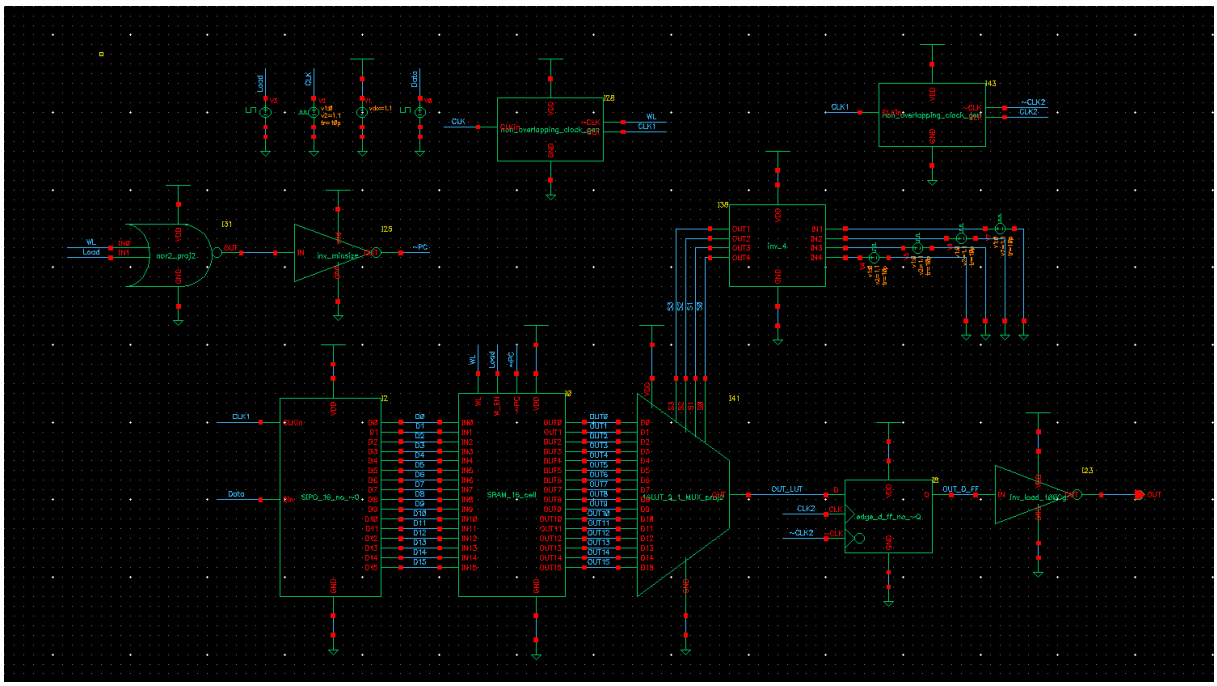


Figure 33: CLB schematic design

As we are using a minimum size LUT, we let the input drive of the address bit be 4 minsize inverters.

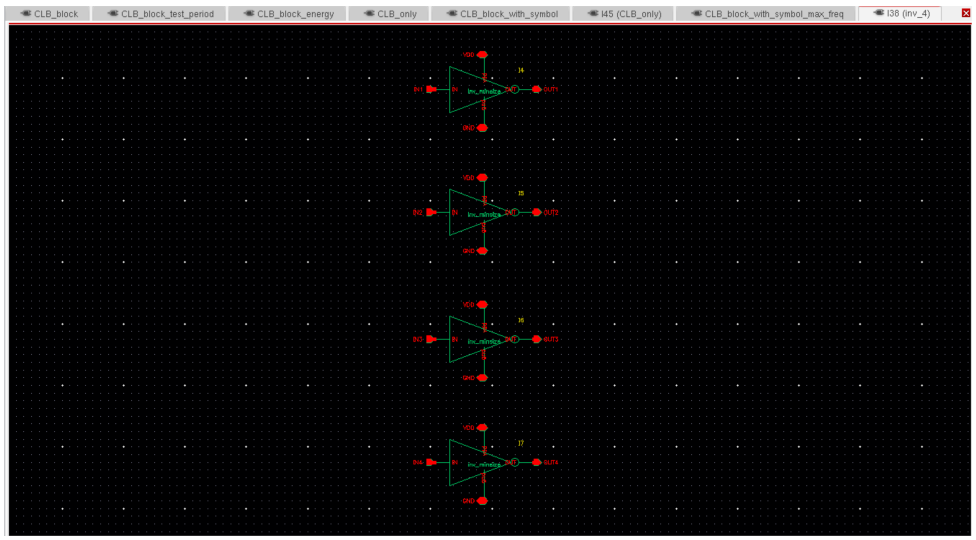


Figure 34: Address input drive of the LUT.

7.2 Design verification

To make sure the CLB works correctly, first, we set the clock's period to a high value. Here, we choose the clock period to be 10ns.
To verify the design, we let the clock signal be 10ns and the load signal will be high for 16 clock cycles to finish writing in all 16 bits of SRAM cell. The address bit will be cycled through all 16 possible combinations. For the address bit, to ensure that the output data of LUT does not violate the timing of the D flip-flop, we let S3 be a pulse twice the period of the clock, which is 20ns. For the LUT to cycle through all of the possible address bits, we let the period of S2 be 40ns, S1 be 80ns, and S0 be 160ns. For the CLB to not abruptly write at the beginning, we let the first 2 periods be a dummy and the first 2 data also be a dummy (the first 2 data bits are 01).
In our schematic, we will write all the data into the SRAM in the duration of 16 clock cycles. To simplify, we let the Load signal be on for 16 cycles to write the data in, and then off to read the data out. The same design is applied when measuring the metric.
Here, we have 2 different test cases: the first is with the data input of 1010 1010 1010 1010, and the second case is with the data input of 1010 1100 0101 0011. Each data bit will be a pulse of 10ns.
For the first case, the output of the verification process is:

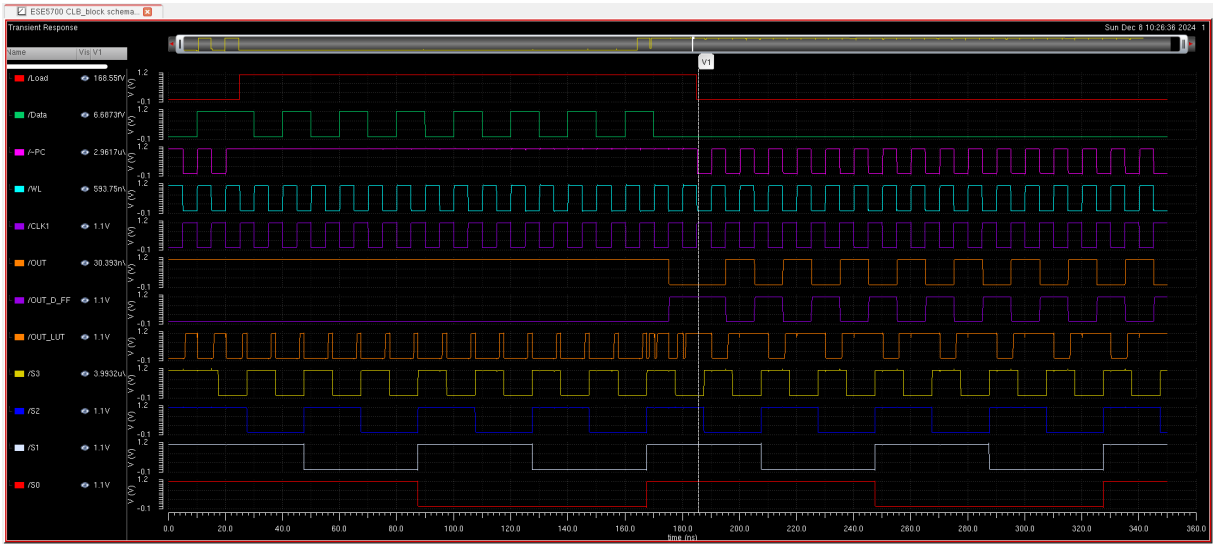


Figure 35: CLB output verification when the input data is 1010 1010 1010 1010

For this case, we are expecting the output to be 0101 0101 0101 0101 when the address bit goes from 1111 to 0000 (cycling through all 16 possible combinations). We can see that the output is what we expected.
For the second case, the output of the verification process is:

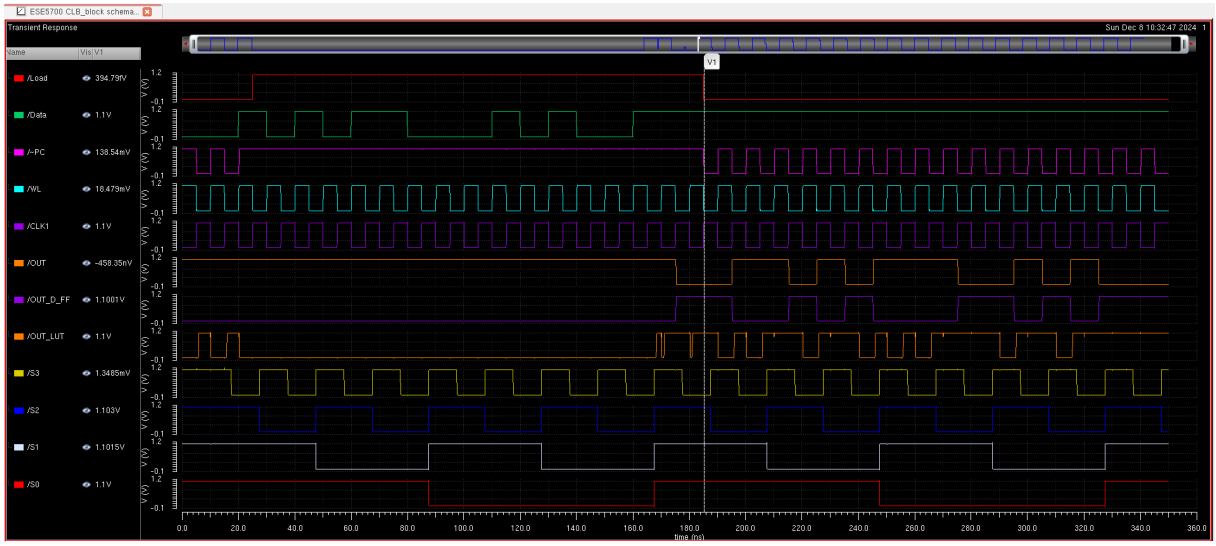


Figure 36: CLB output verification when the input data is 1010 1100 0101 0011

For this case, we are expecting the output to be 1100 1010 0011 0101 when the address bit goes from 1111 to 0000 (cycling through all 16 possible combinations). We can see that the output is what we expected.
From 35 and 36, we can see that the CLB is working correctly.
We also want to see the behavior of the D flip-flop in the CLB.

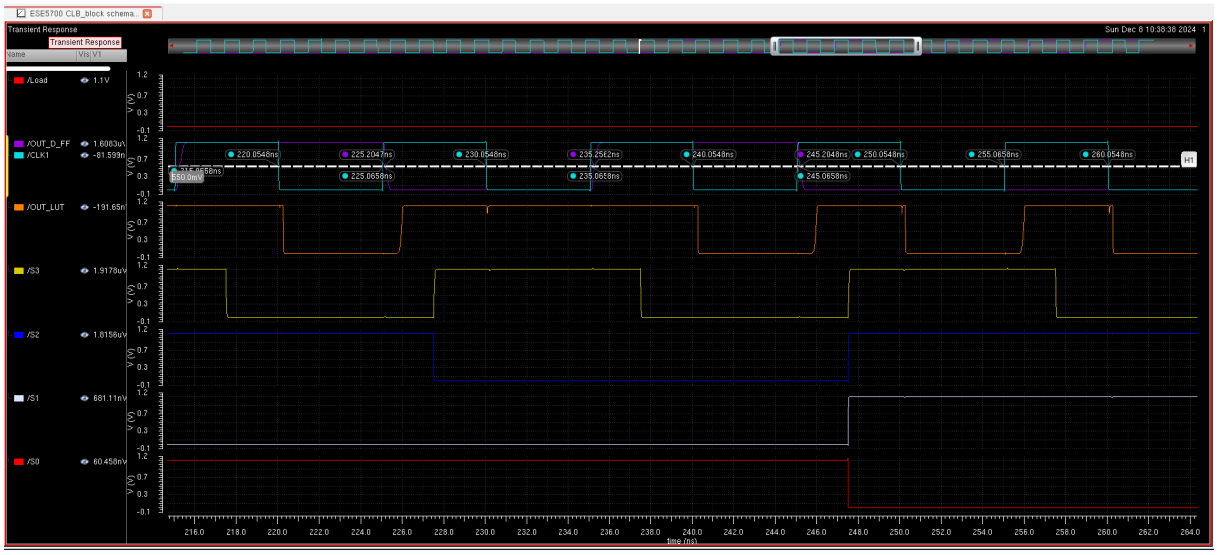


Figure 37: $t_{c\to q}$ of D flip-flop in this CLB

The $t_{c\to q}$ when the output of the flipflop change from low to high is:

$$t_{c\to q1} = 235.2562 - 235.0658 = 0.1904ns$$

The $t_{c\to q}$ when the output of the flipflop change from high to low is:

$$t_{c\to q1} = 245.0658 - 245.2048 = 0.139ns$$

Let $t_{c\to q} = 0.1904ns$.
In the same schematic, we also measure the delay of the read operation of the SRAM.

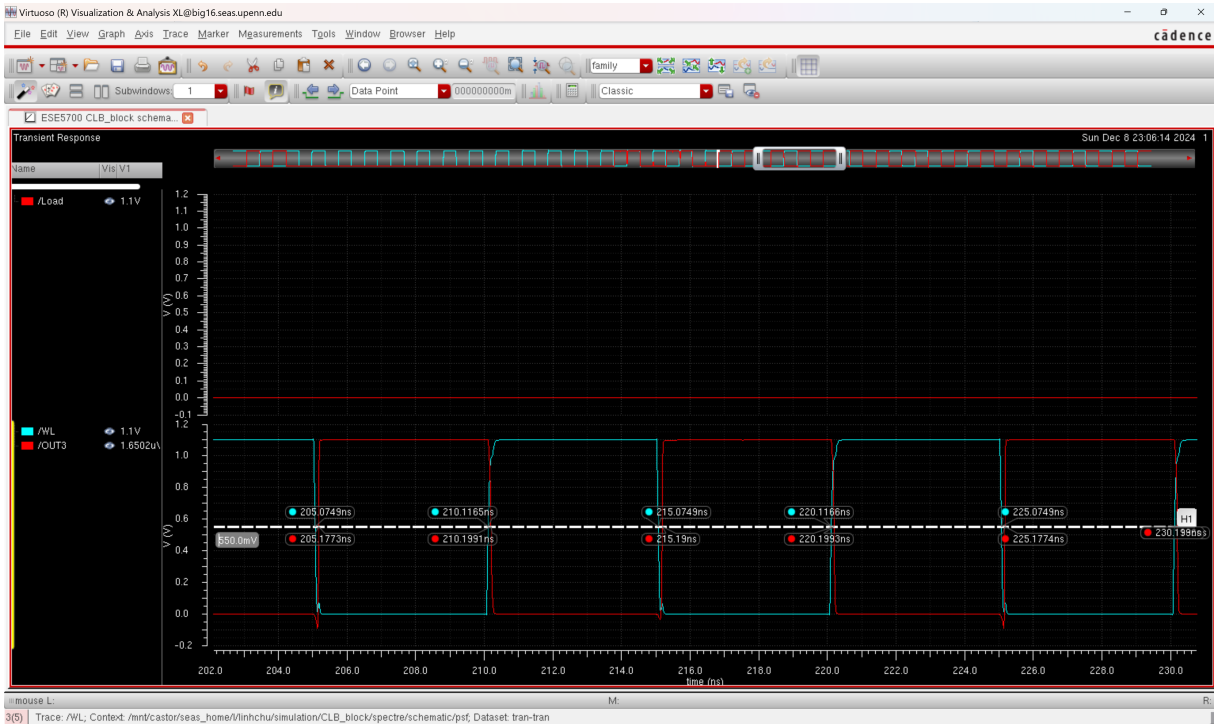


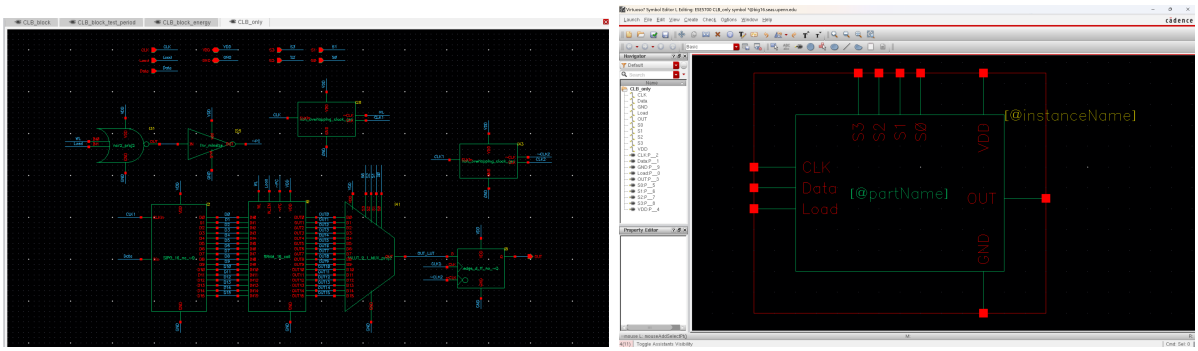
Figure 38: t_{SRAM} of the read operation of the SRAM in this CLB

From the figure, the delay of the read operation is:

$$t_{SRAM} = 210.1991 - 210.1165 = 0.0826ns = 82.6ps$$

7.3 Design with symbol

After having verified the functionality of the CLB block, we design a symbol for it:



(a) CLB block design (b) CLB symbol

Figure 39: CLB design with symbol

To make sure that the symbol works the same as the schematic 33, we also test the symbol once with the input data of 1010 1010 1010 1010:

Figure 40: CLB output verification when input is 1010 1010 1010 1010.

We see that the output is the same as 35, thus it is still working correctly.

8 Metrics measurement

8.1 Frequency

The test schematic to find the maximum frequency is shown below

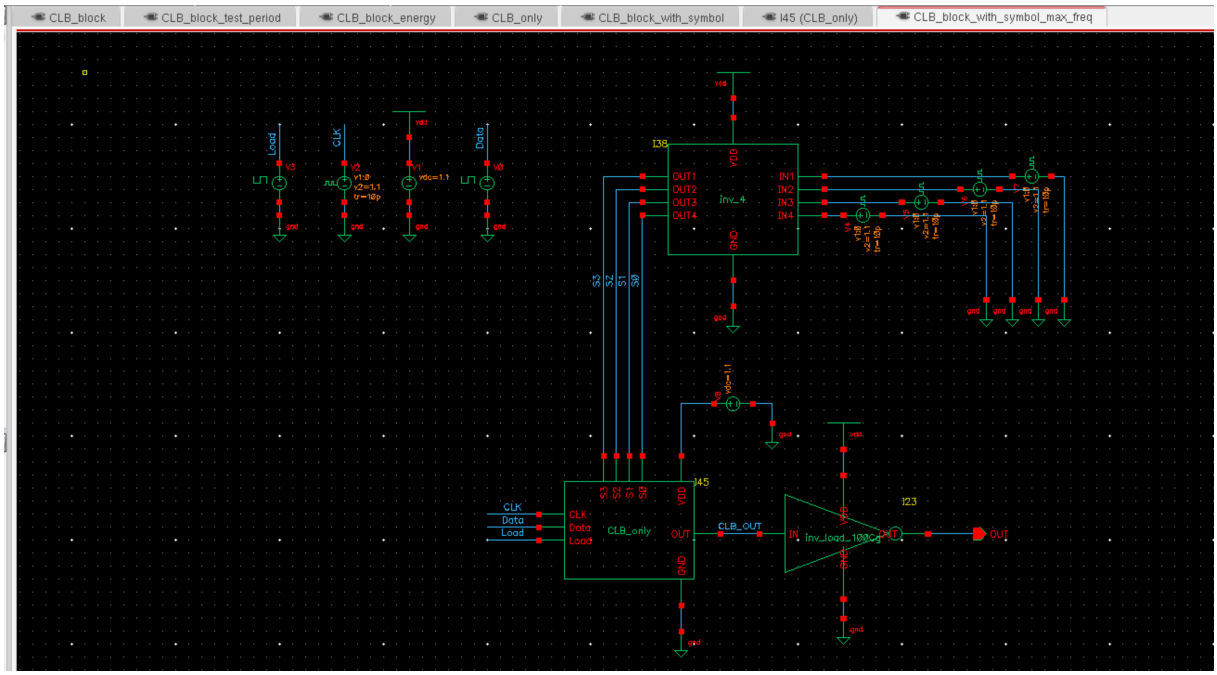


Figure 41: Schematic to measure the maximum frequency.

We let the clock signal period be a variable and the load signal will be high for 16 clock

cycles to finish writing in all 16 bits of SRAM cell. The address bit will be cycled through all 16 possible combinations. For the address bit, to ensure that the output data of LUT does not violate the timing of the D flip-flop, we let S3 be a pulse twice the period of the clock. For the LUT to cycle through all of the possible address bits, we let the period of S2 be 4 times the period of the clock, S1 be 8 times the period of the clock, and S0 be 16 times the period of the clock. For the CLB to not abruptly write at the beginning, we let the first 2 periods be a dummy and the first 2 data also be a dummy (the first 2 data bits are 10).
Here, to verify the functionality of the CLB, we let the input data to be 1010 1010 1010 1010. We would expect an output of 0101 0101 0101 0101 when the address cycling from 15 to 0.
In our schematic, we will write all the data into the SRAM in the duration of 16 clock cycles. To simplify, we let the Load signal be on for 16 cycles to write the data in, and then off to read the data out.
With the setup time found in section 5 ($t_{setup} = 61.13ps$), the $t_{c\to q} = 0.1904ns$ and $t_{SRAM} = 82.6ps$ from section 7, and the maximum delay of the LUT from section 1 ($t_{LUT} = 1.093ns$), the clock period of the CLB must not violate the timing constraint of the D flip-flop, which mean the period must be:

$$\begin{aligned} & T_{period} \geq t_{setup} + t_{c\to q} + t_{SRAM} + t_{LUT} \\ \leftrightarrow \quad & T_{period} \geq 61.13ps + 190.4ps + 82.8ps + 1093ps \\ \leftrightarrow \quad & T_{period} \geq 1427.33ps \end{aligned}$$

However, the CLB still works correctly when we let the clock period be 1ns. The simulation result is:

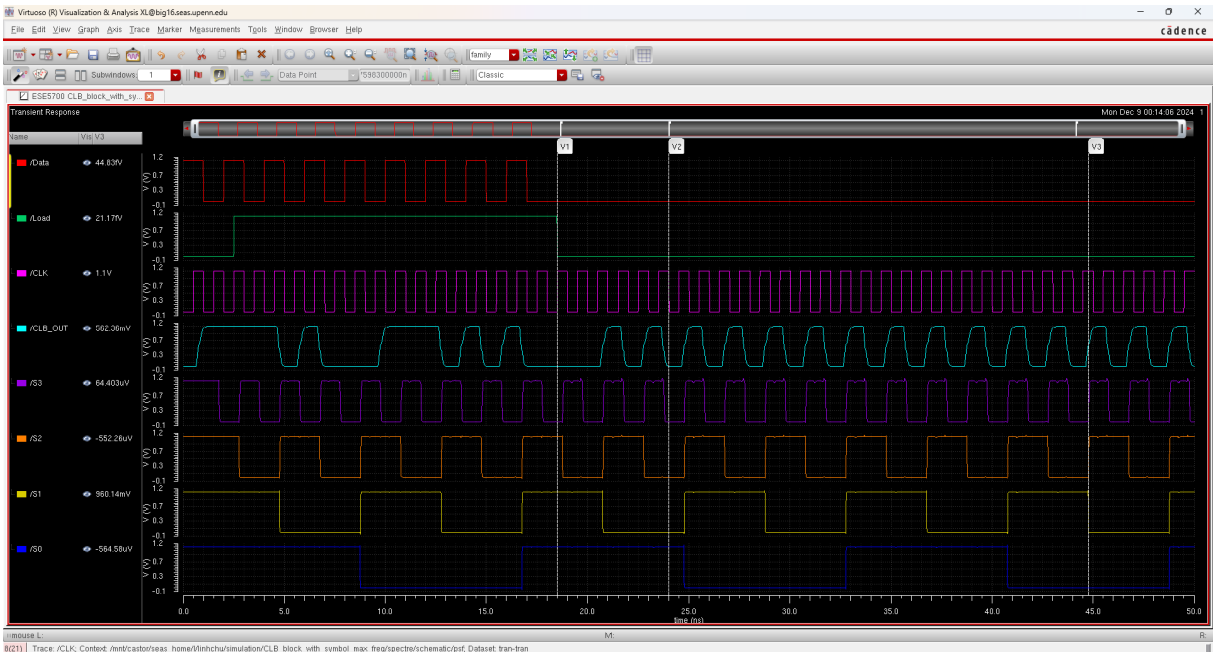


Figure 42: The CLB at the maximum frequency of 1ns.

We also test the design with the input 1010 1100 0101 0011. We would expect the output of 1100 1010 0011 0101 when the address bit cycles from 15 to 0.

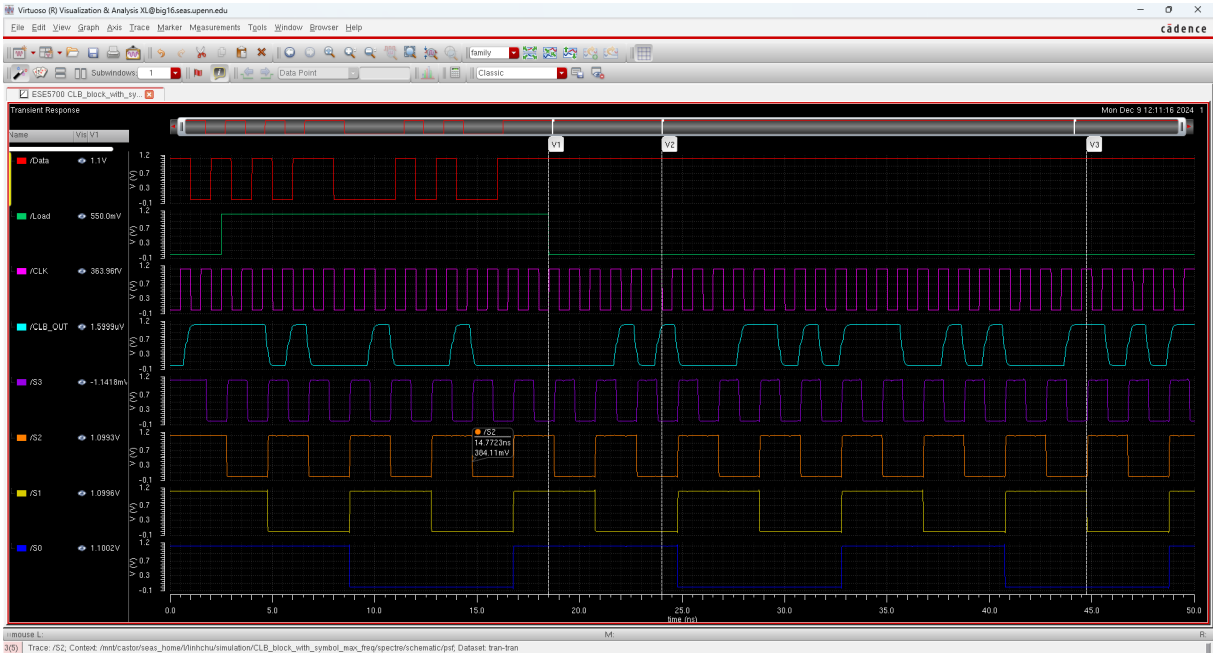


Figure 43: The CLB at the maximum frequency of 1ns when the input is 1010 1100 0101 0011.

The simulation result shows that at the frequency of 1ns, the CLB still working correctly. Thus, the maximum frequency of the CLB is 1GHz.

8.2 Energy

The schematic for calculating energy is:

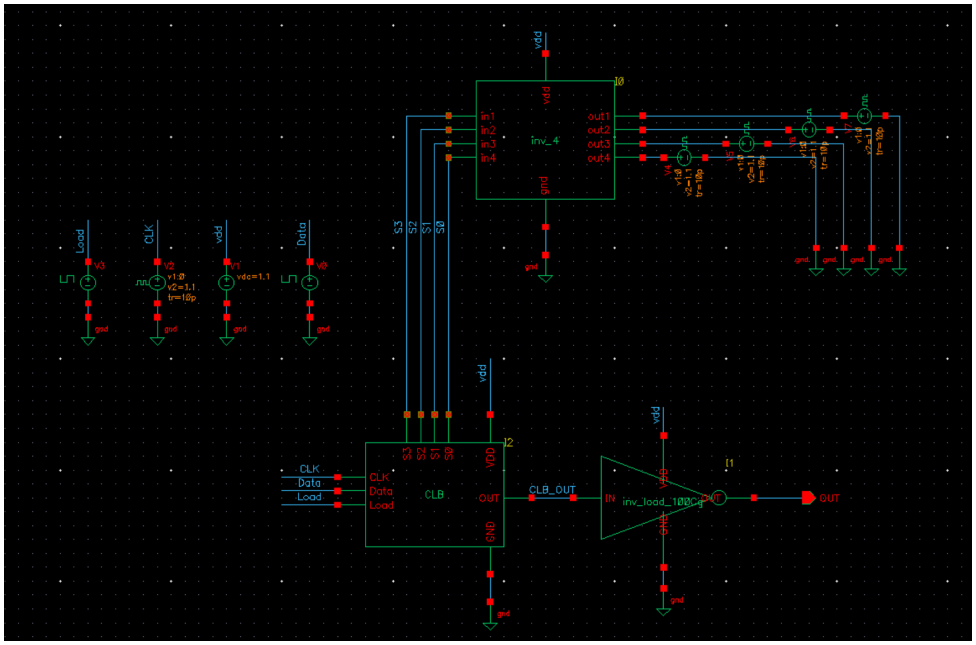


Figure 44: Circuit to measure energy at maximum frequency

We let the data input be 1010 1010 1010 1010 as in this case, the LUT when cycling through all 16 possible combinations will have to switch the most, which can make this a worst-case possible energy consumption.

The waveform result for energy measurement

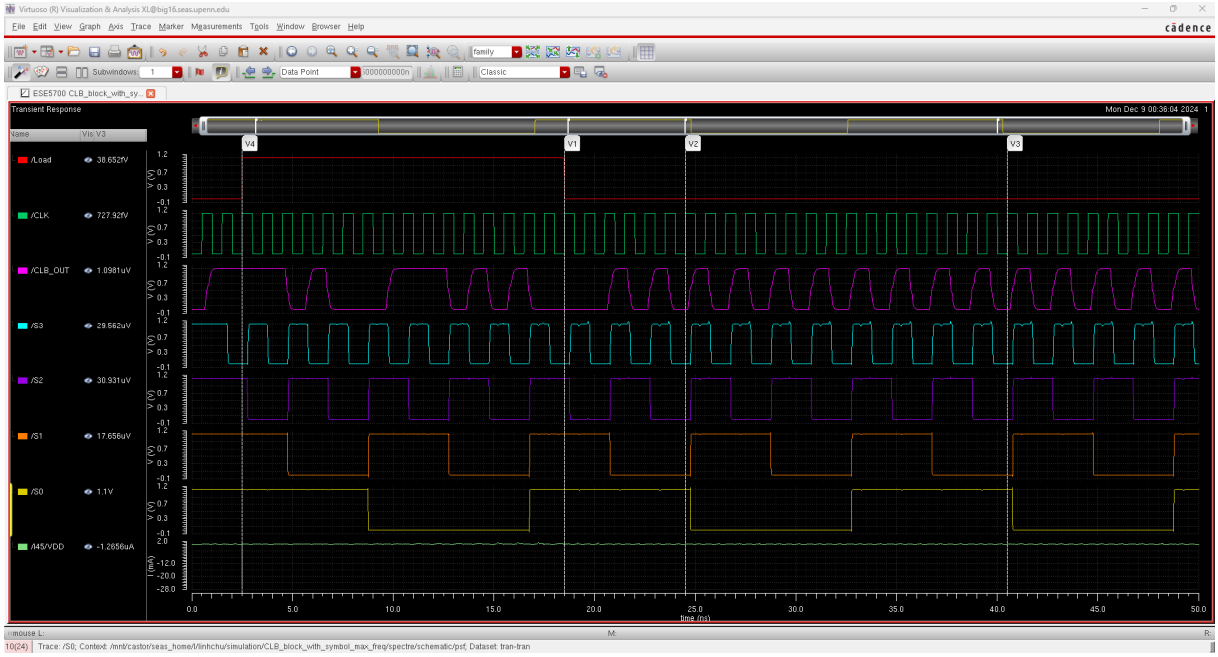


Figure 45: Total current through source V_{dd}

Loading energy: Loading energy is defined as the energy consumed by the entire circuit to load all the data into the SRAM. This energy can be calculated by integrating the product of the current and the voltage V_{dd} over 16 cycles of the clock signal as the loading process needs 16 clock cycles. This corresponds to the time required for 16 data inputs to travel from the source vbits, through the SIPO (Serial-In Parallel-Out) register, and get written to the SRAM (as the data is loaded serially, with two consecutive inputs being offset by one clock cycle).

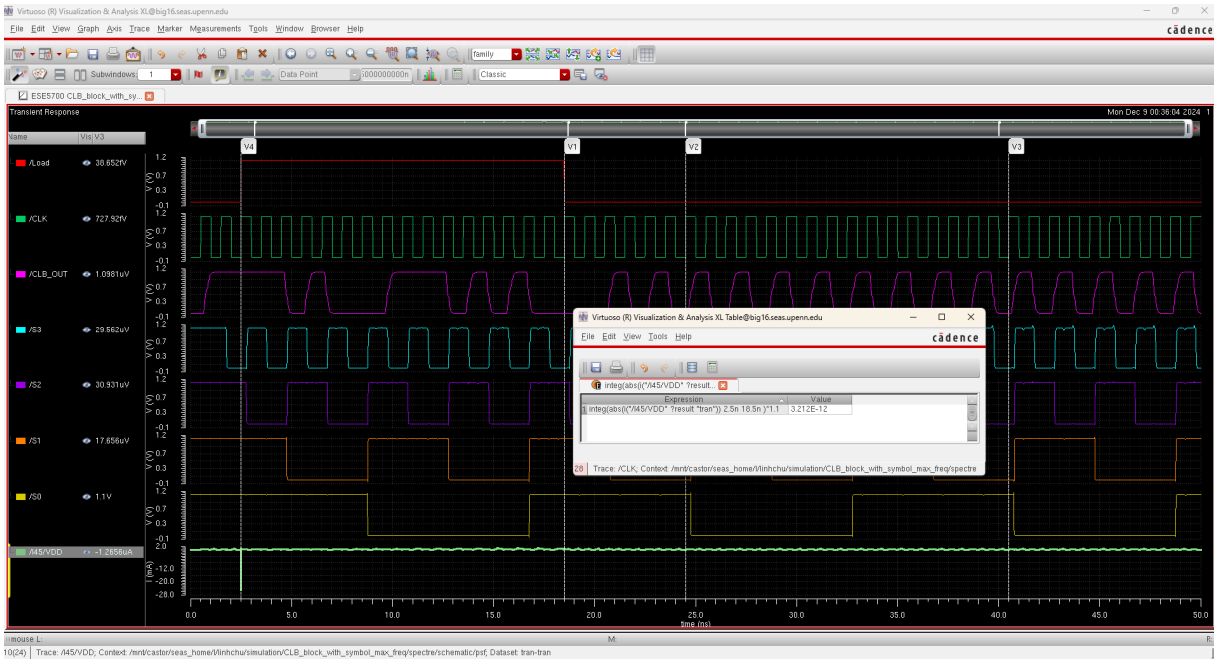


Figure 46: Loading energy calculation

As can be seen from Figure 46, the loading energy is $3.212 \times 10^{-12}\ J$.
Active energy: Active energy is the energy consumed to cycle through all address bits from 0 to 15 at the maximum operating frequency. This can be calculated by integrating the product of the current and the supply voltage V_{dd} over a time period of 16 clock cycles for the address bits to cycle through all possible outputs.

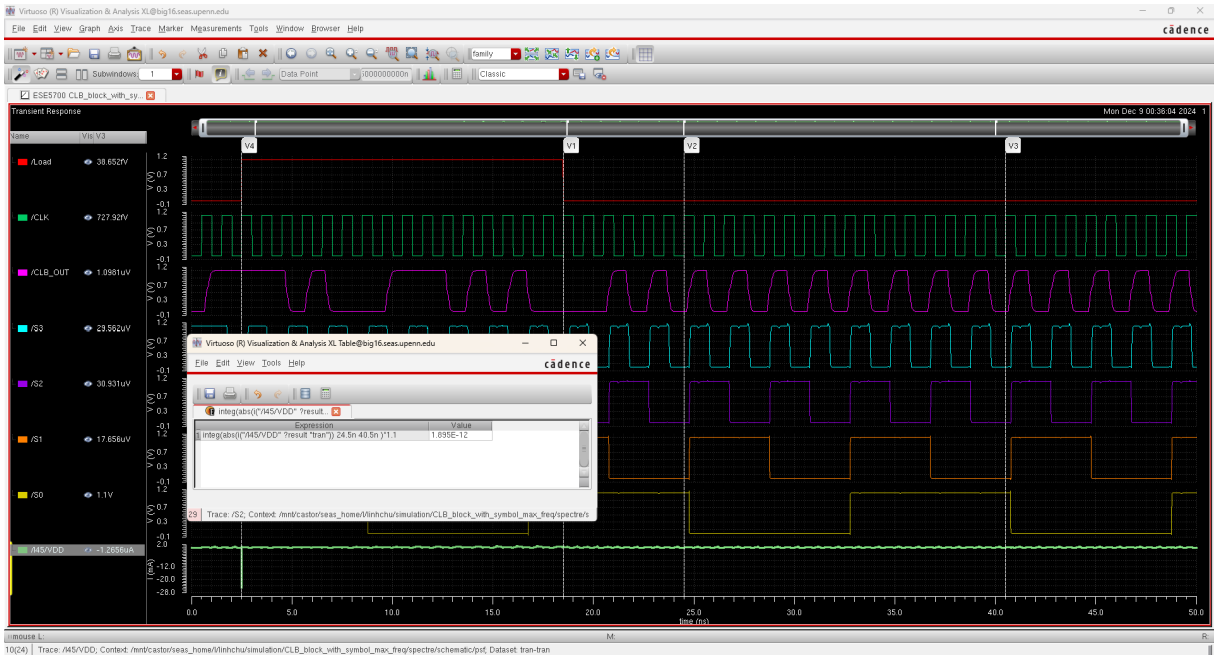


Figure 47: Active energy calculation

From Figure 47, the active energy is $1.895 \times 10^{-12}\ J$.

As we assume that the average energy is 20% loading energy and 80% active energy, the average energy is:

$$\begin{aligned}\text{Average Energy} &= 20\%(\text{Loading energy}) + 80\%(\text{Active energy}) \\ &= 0.2 \times 3.212 \times 10^{-12} + 0.8 \times 1.895 \times 10^{-12} \\ &= 2.1584 \times 10^{-12}\ J\end{aligned}$$

8.3 Area

With the design and sizing of the SRAM cell shown in 9, the area of one SRAM cell is:

$$area_{SRAM} = 120 \times 2 + 240 \times 2 + 290 \times 2 = 1300(nm)$$

As we are using the minimum size LUT, the area of 1 2:1 MUX is:

$$area_{2:1mux} = 120 \times 4 = 480(nm)$$

Thus, the area metric is:

$$area_{total} = 1300 \times 16 + 480 \times 15 = 28000(nm) = 28\mu m$$

8.4 Figure of Merit (FOM)

The FOM of this design is:

$$\begin{aligned}\text{FOM} &= area_{total} \times \text{Average Energy} \times \frac{1}{maxFrequency} \\ &= 28\mu m \times 2.1584 \times 10^{-12}\ J \times \frac{1}{1GHz} \\ &= 6.0435 \times 10^{-26}\ (m \cdot J \cdot s)\end{aligned}$$

9 Summary

Metric	Value
Frequency	1 GHz
Average Energy	$2.1584 \times 10^{-12}\ J$
Area	28 μm
FOM	$6.0435 \times 10^{-26}\ m \cdot J \cdot s$

Table 1: Metric summary

10 Attempt to optimize the CLB

We also have some attempts to try for a better FOM, however, it has not been achieved. As the LUT is the bottleneck of the design (has the highest delay), we try to size the LUT up to see if we can reduce the delay of the LUT, which would mean that increase our frequency. As our maximum frequency is 1GHz (or a period of 1ns), we tried to lower our clock period to 900ps.
To do this, first, we need to make sure that the write operation of the SRAM and the LUT is correct. We size up the tristate buffer and tristate inverter to have a width of 240nm. Then, we tried to size up the 2:1 MUX width and the restoration inverters at the end of the LUT so that they have the same size of wmux.

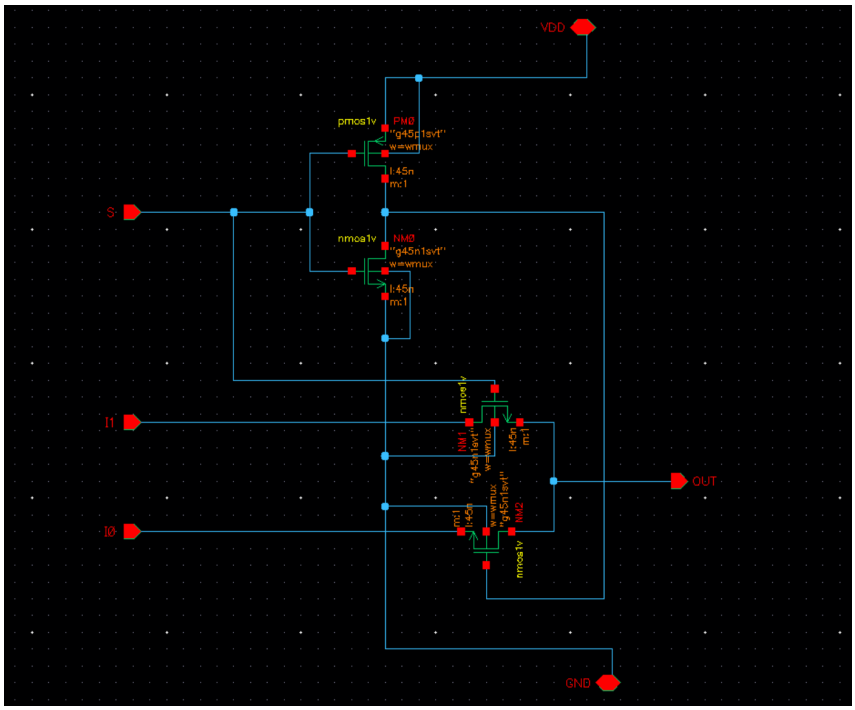


Figure 48: MUX 2:1 schematic

First, we let the width of all the transistors wmux = 240ns. The simulation result is shown below:

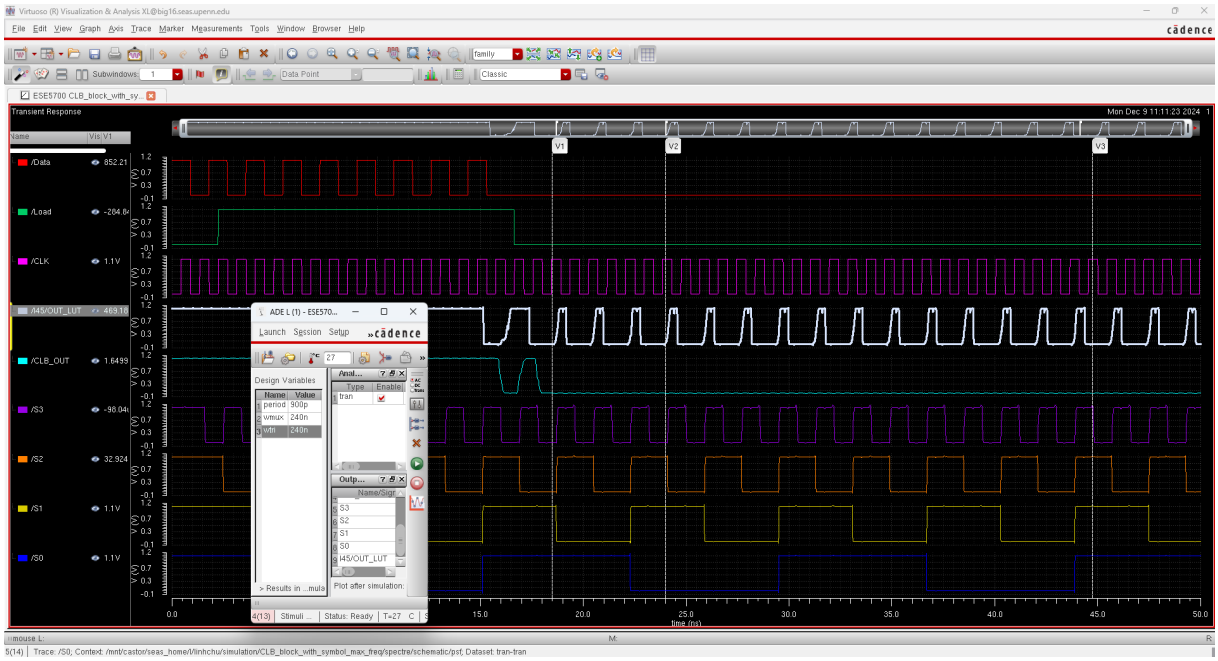


Figure 49: CLB output when the clock period is 900ps, wmux = 240ns.

From the figure, we can see that although the output of the LUT is what we expected, the D flip-flop cannot capture the data value 1 of the LUT. We believe this happens due to the timing violation of the D flip-flop. Then, we try to increase the wmux to 720nm to see what happens to the CLB.

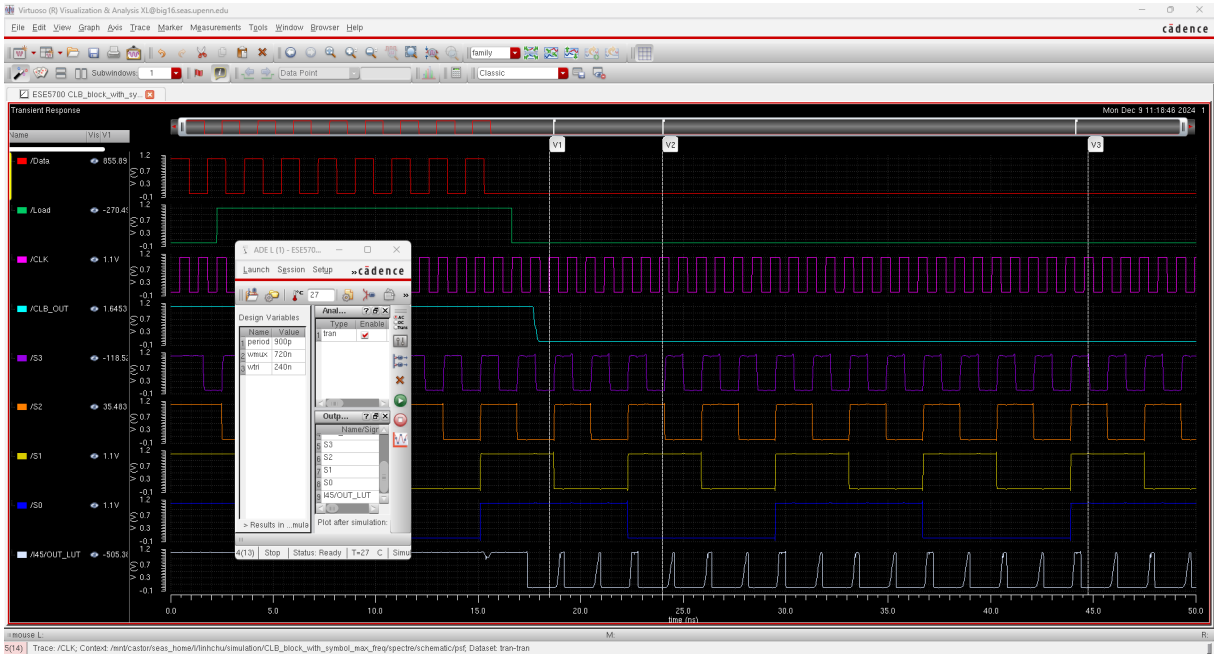


Figure 50: CLB output when the clock period is 900ps, wmux = 720ns.

From the figure, we can see that although the output of the LUT is what we expected, the D flip-flop cannot capture the data value 1 of the LUT. Here, we decided to stop with the sizing of the LUT as the area has now increased by 12% and we haven't achieved a 10% reduce in frequency.

11 Conclusion

In this report, we have presented our approach to constructing a Configurable Logic Block (CLB), including our comprehensive design for the individual components and the setups we developed to verify the circuit's functionality. We benchmarked our design using metrics such as maximum frequency, energy consumption, area, and FOM. Finally, we experimented with various circuit sizes to optimize the FOM, achieving the best result of $6.0435 \times 10^{-26}\ m \cdot J \cdot s$.

For future work, we aim to further investigate different sizing schemes and explore alternative logic styles for the components to enhance the design, minimize delay, and reduce energy consumption.

12 Contribution

Student	Contribution
Linh Chu	Design, and verify the SRAM cell and peripheral, D flip-flop, SIPO, CLB Measure the max frequency of the CLB Try to optimize the design
Anh Tran	Design circuit for components verification Measure active and loading energy of the CLB Try to optimize the design

Table 2: Contribution of each team member.