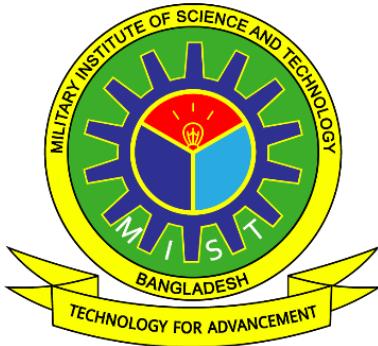


# MILITARY INSTITUTE OF SCIENCE & TECHNOLOGY

## Department of Electrical Electronic and Communication Engineering

Subject: VLSI II Laboratory (EECE – 458)



### Assignment

Name	MD. Mahidul Haque
ID	202216043
Course Code	EECE – 458
Batch	EECE-20
Section	A
Date of submission	August 9, 2025

# Design and Analysis of Basic Digital Blocks Using Cadence Virtuoso

---

## Introduction to Cadence Virtuoso

Cadence Virtuoso is a leading Electronic Design Automation (EDA) toolset providing a comprehensive platform for designing, placing out, simulating, and verifying analog, digital, and mixed-signal integrated circuits (ICs). Virtuoso environment provides circuit designers the ability to build their concepts from initial schematics to physical layouts for fabrication readiness, with accuracy and efficiency at all points in the design flow. With strong schematic capture, parameterized cell layout, interactive editing of layout, and simulation, Cadence Virtuoso allows engineers to maximize performance, area, and power consumption, all drivers of VLSI design today. Its in-place verification and analysis allow designers to spot problems early, before silicon is created, and save time to market by minimizing rework and expensive re-spins. Virtuoso's flexibility makes it an extremely desirable tool for both research and industry, offering unparalleled control and accuracy in designing sound and manufacturable ICs.

## Introduction to Configurable Logic Block (CLB)

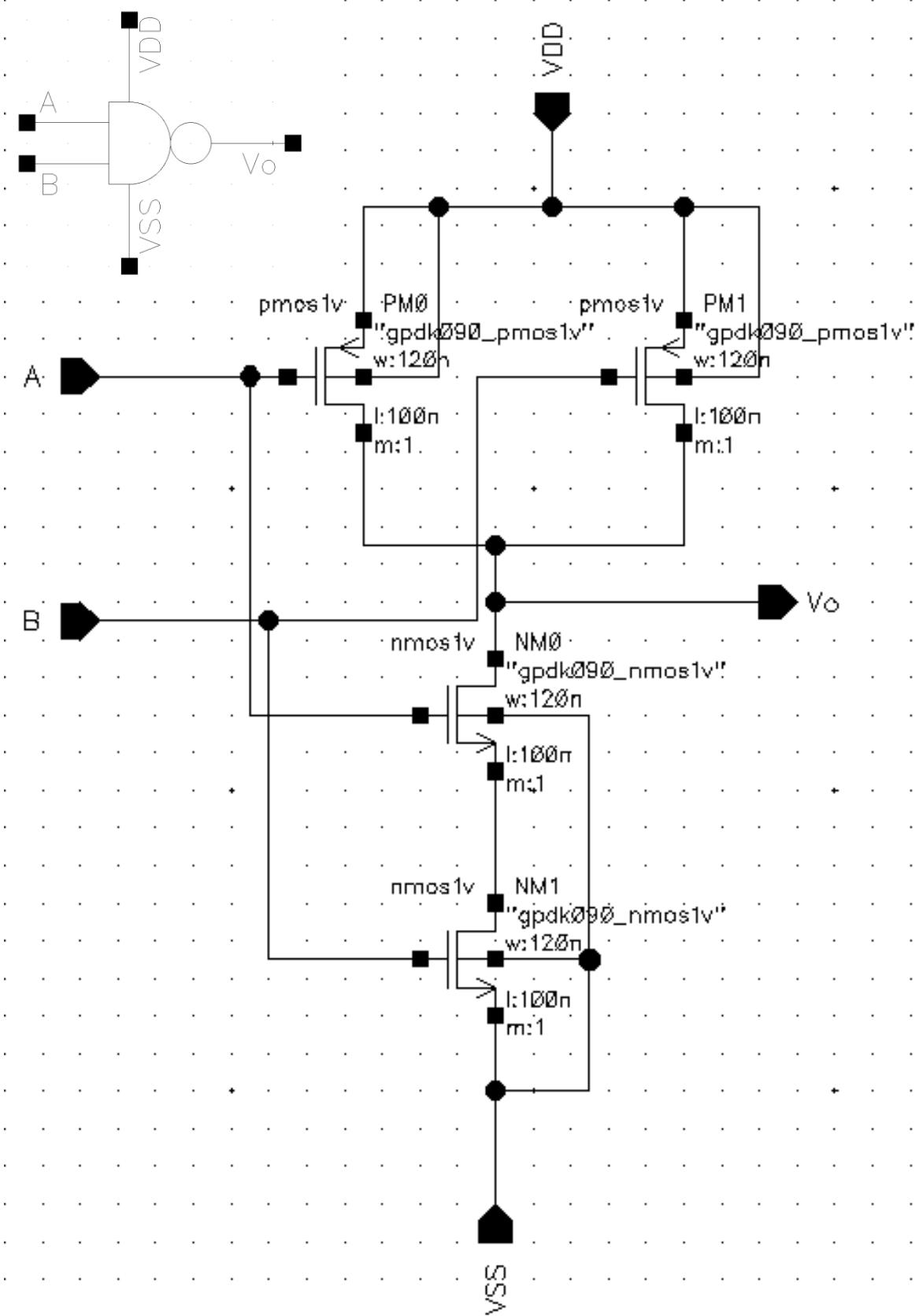
Configurable Logic Block (CLB) is a core component in digital systems, most notably in Field Programmable Gate Arrays (FPGAs) and some custom digital integrated circuits. CLBs provide designers with the ability to realize a variety of logical functions by defining the internal organization of logic gates, multiplexers, flip-flops, and occasionally memory cells.

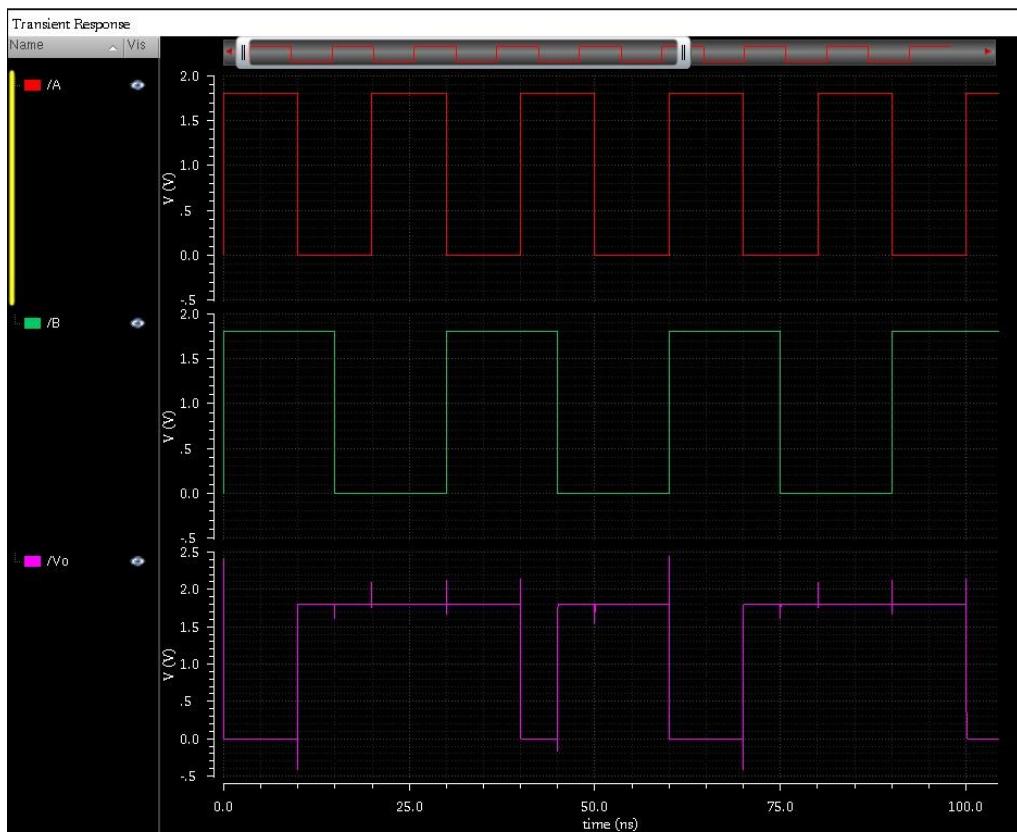
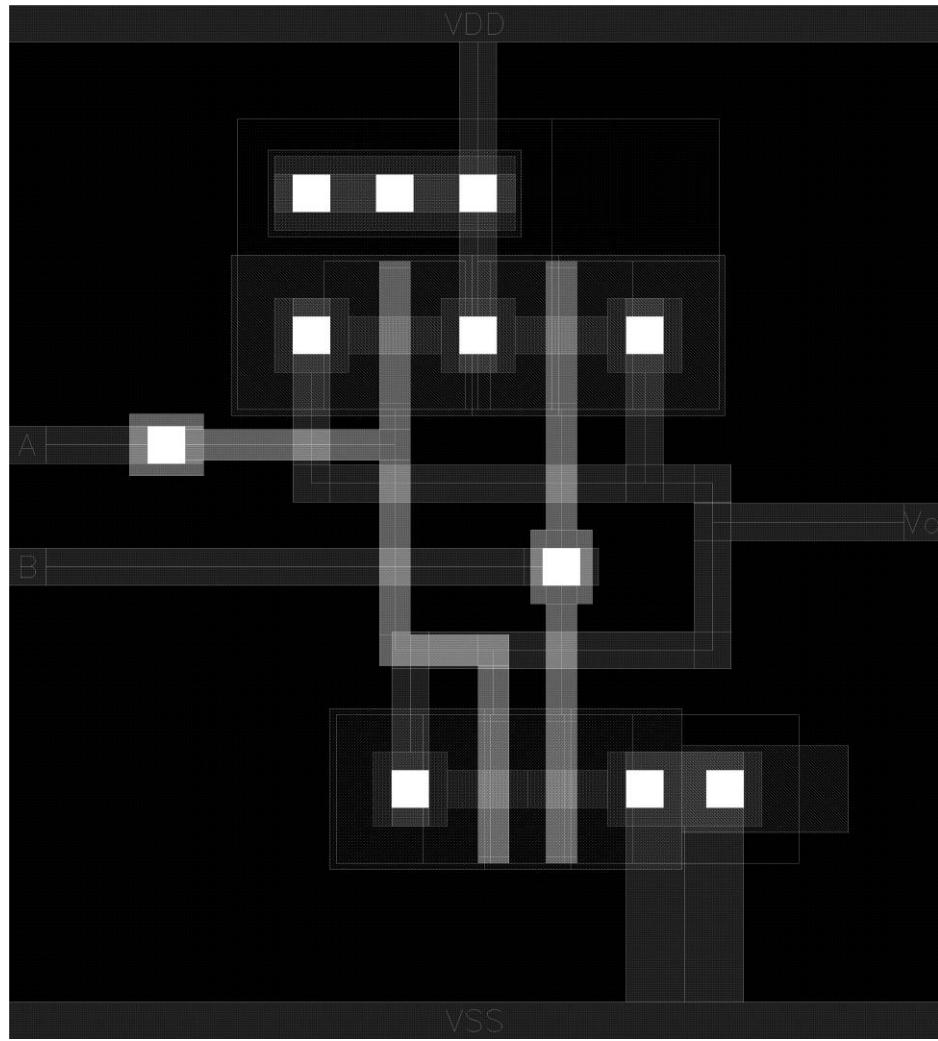
Reconfigurability results in unparalleled design flexibility to implement complex digital functions without hardware redesign. CLBs are typically made up of Look-Up Tables (LUTs) to apply combinational logic, flip-flops to apply sequential logic, and routing resources to interconnect them with other CLBs and system blocks. Because they are reprogrammable, CLBs allow digital systems to be rapidly designed and verified and are used for prototyping and production-level implementation of digital circuits. Understanding CLB function and structure is at the core of knowing digital VLSI systems and FPGA design.

## Schematic Symbols, Layouts, and Test Outputs

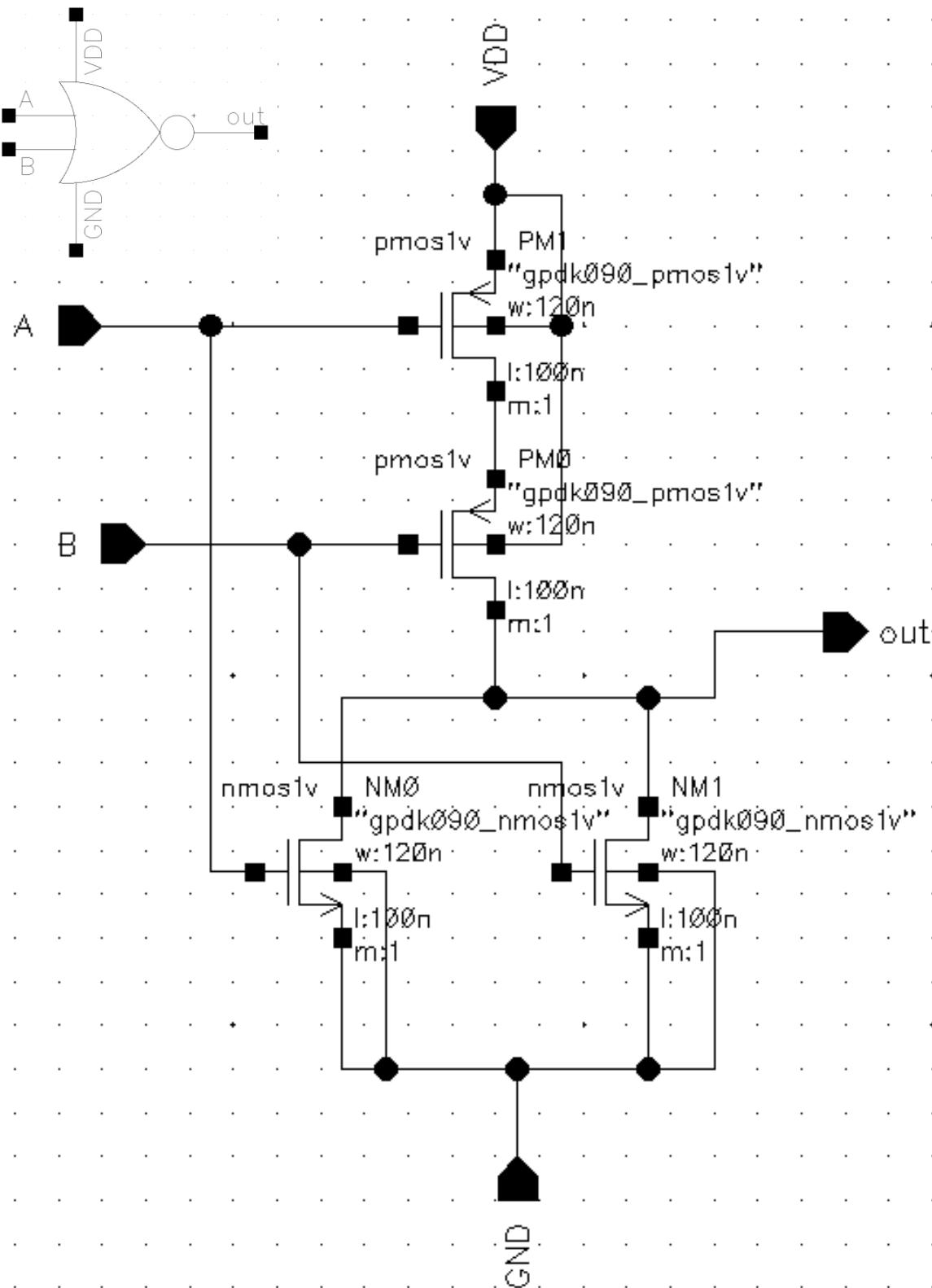
In this section, all of the created digital blocks are displayed with their schematic symbol, layout design, and test run output. These outputs and images give a complete overview of each component's design and functions. From the observation of these outputs, designers can verify proper functioning, find fault in the design, and optimize the circuits to increase performance. The following components are discussed extensively:

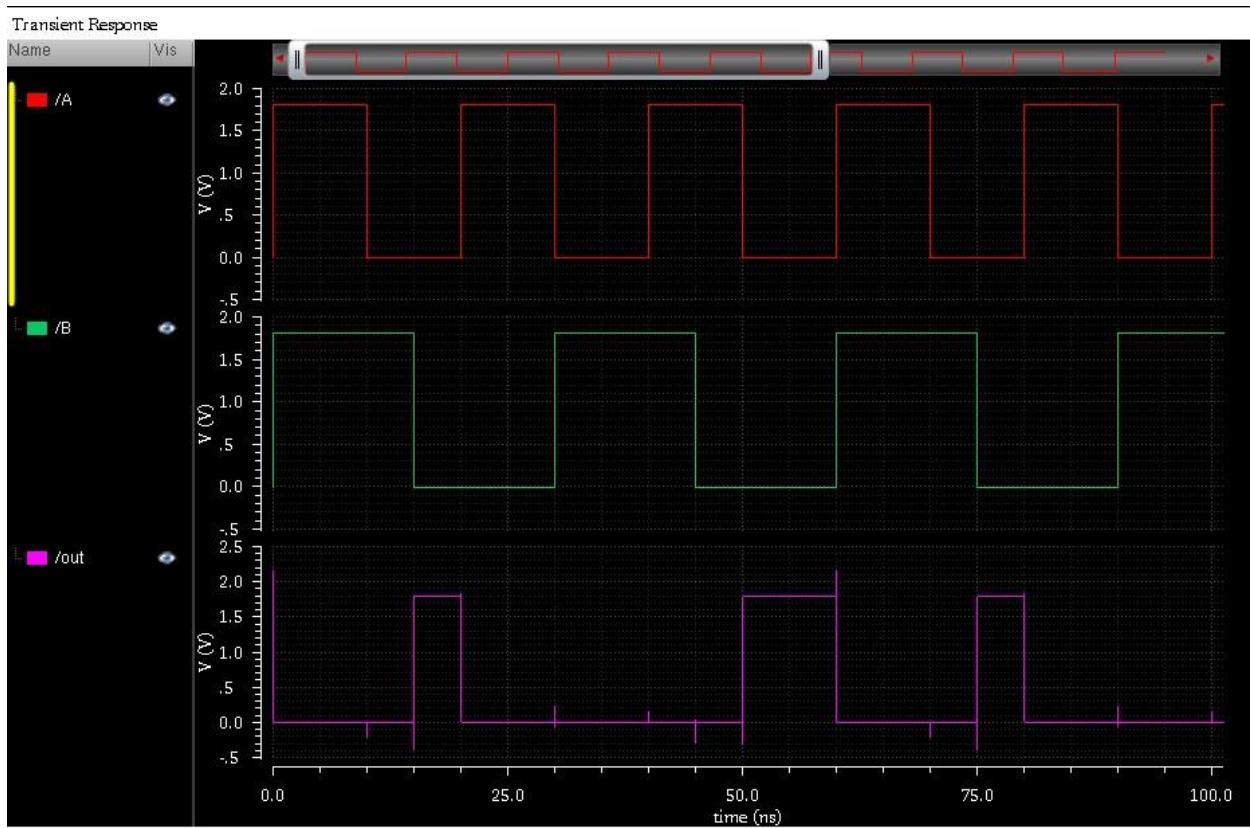
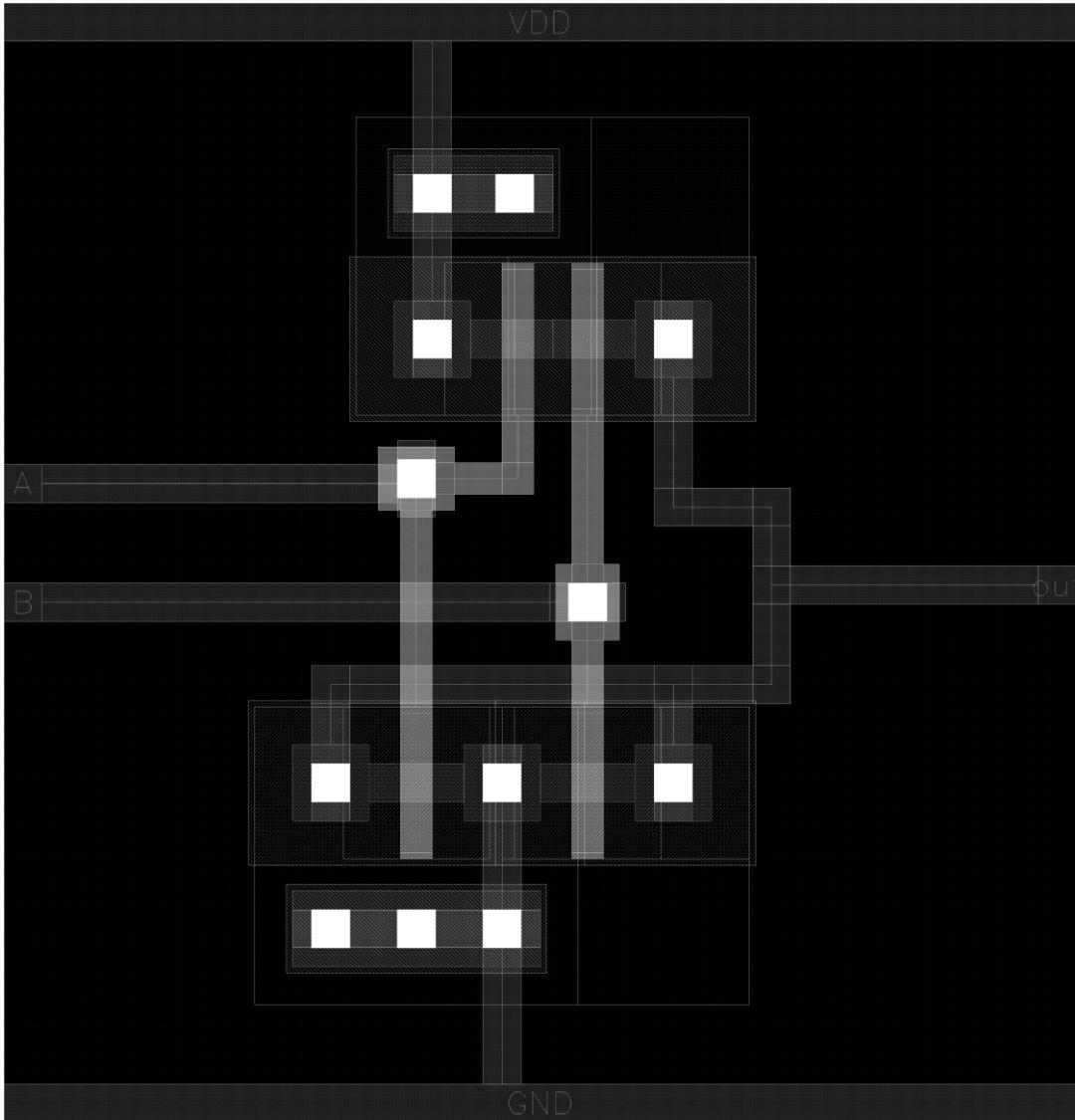
## 1.2IPNAND



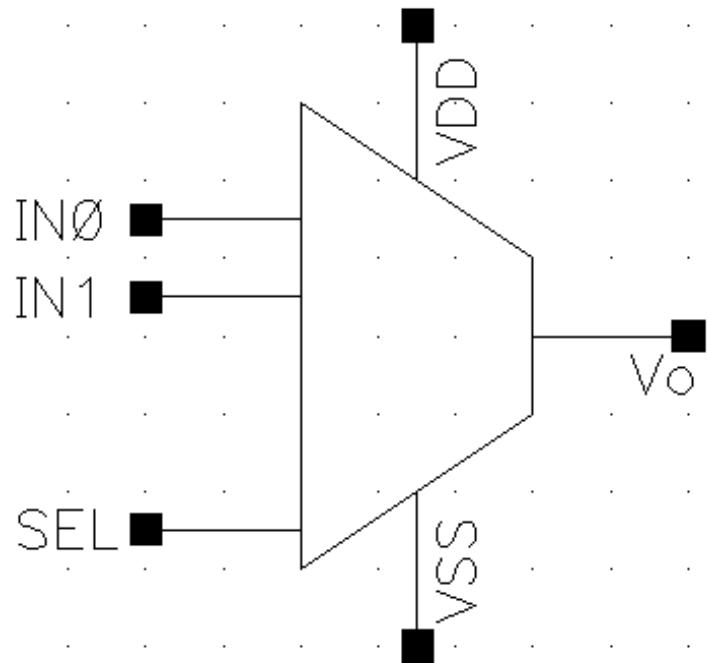
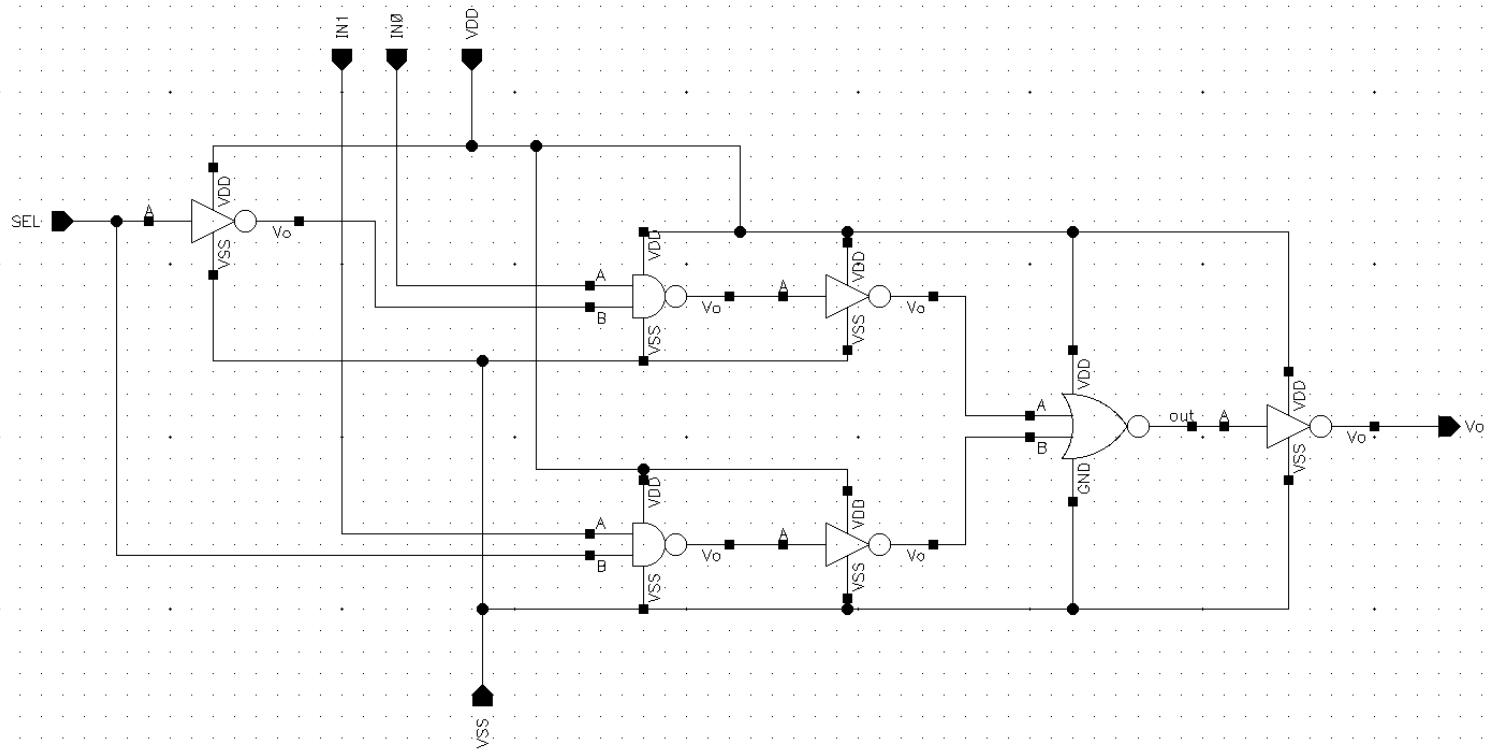


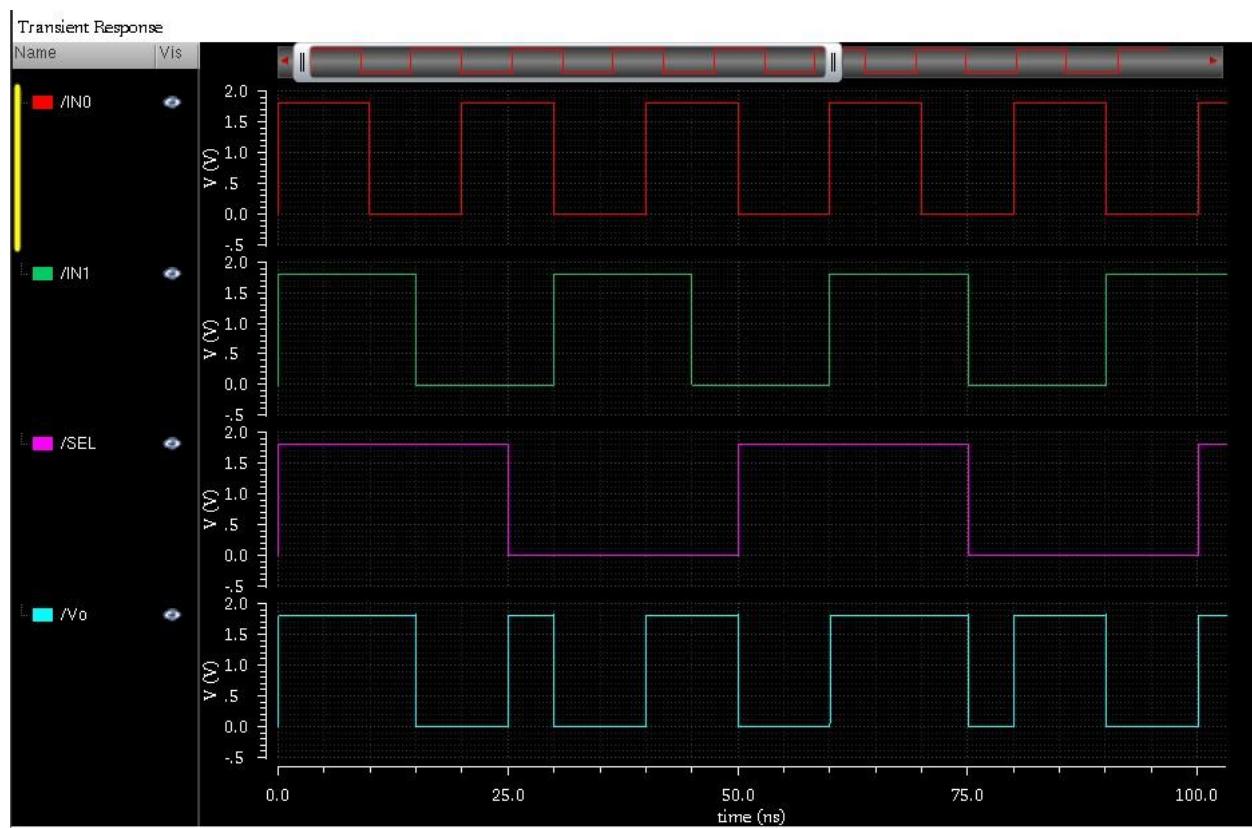
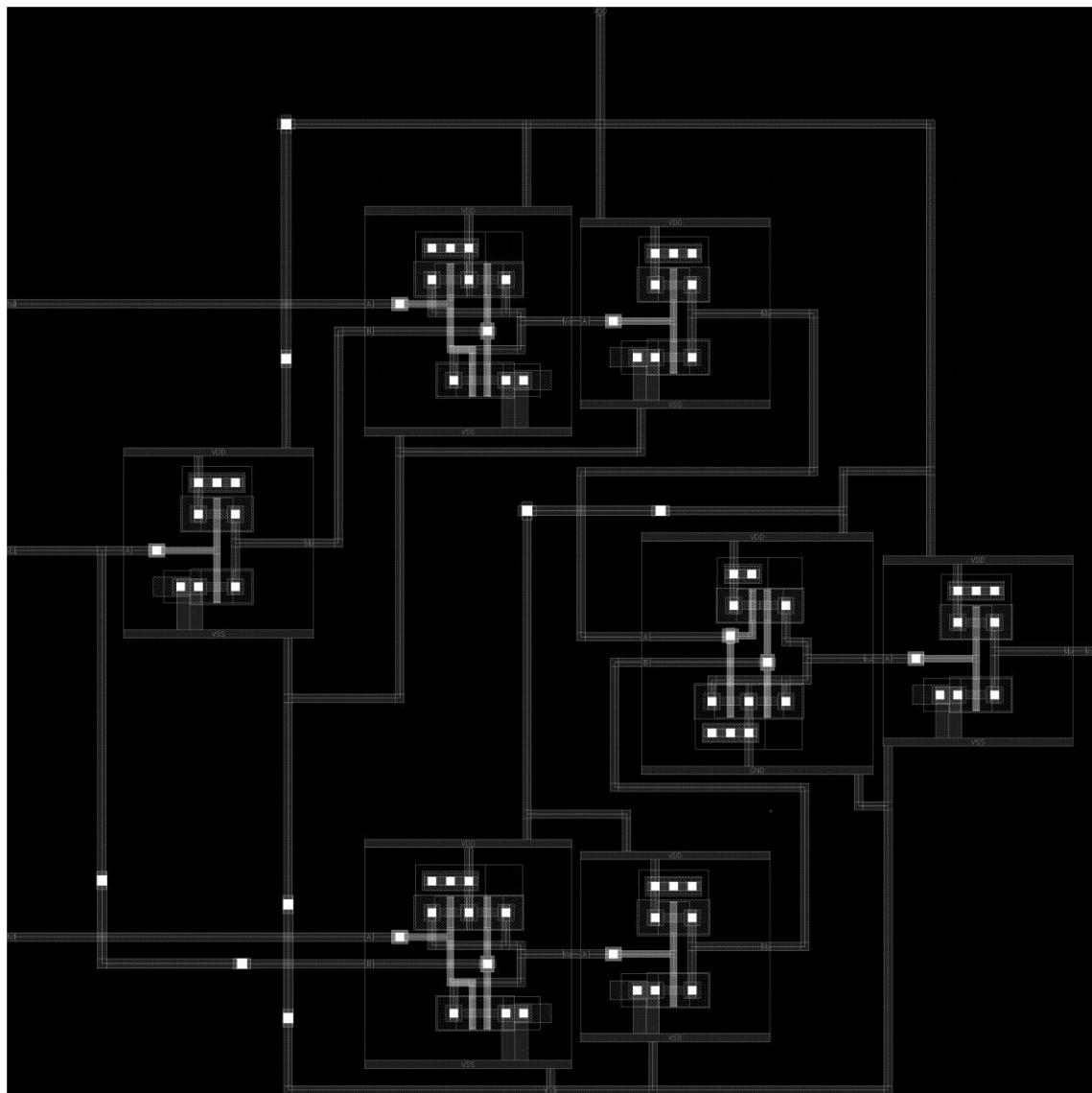
## 2. 2IPOR



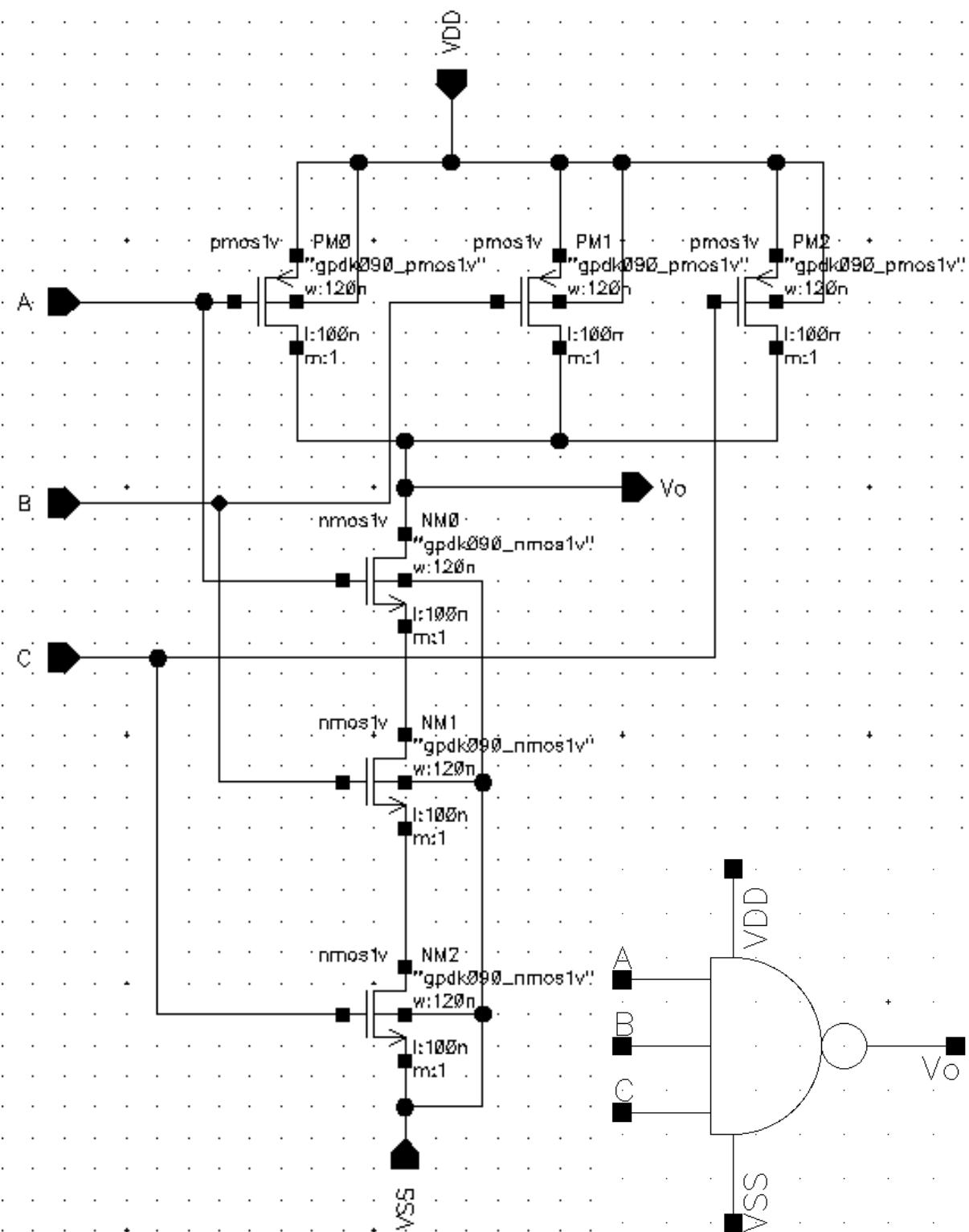


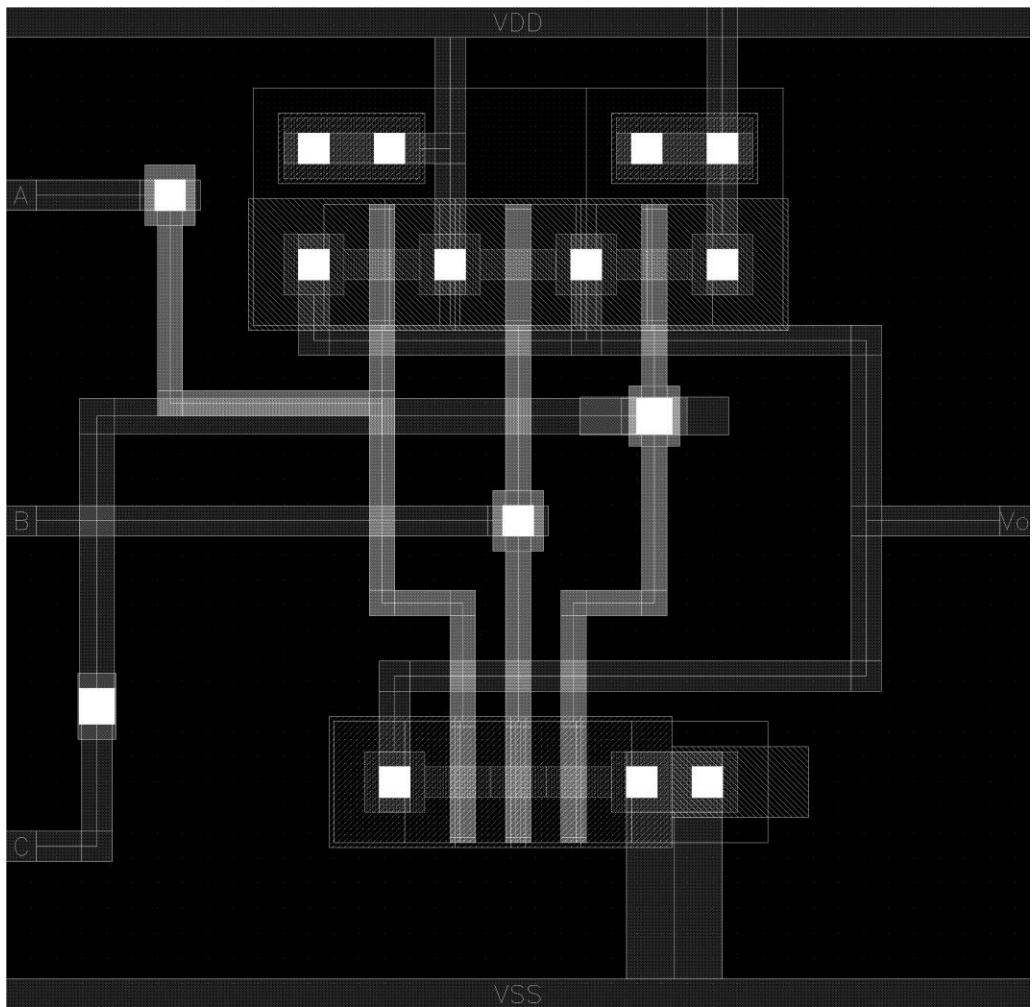
### 3. 2x1MUX



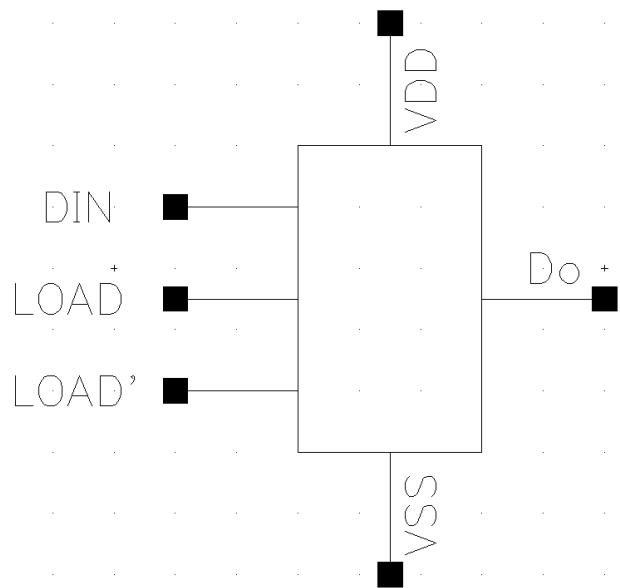
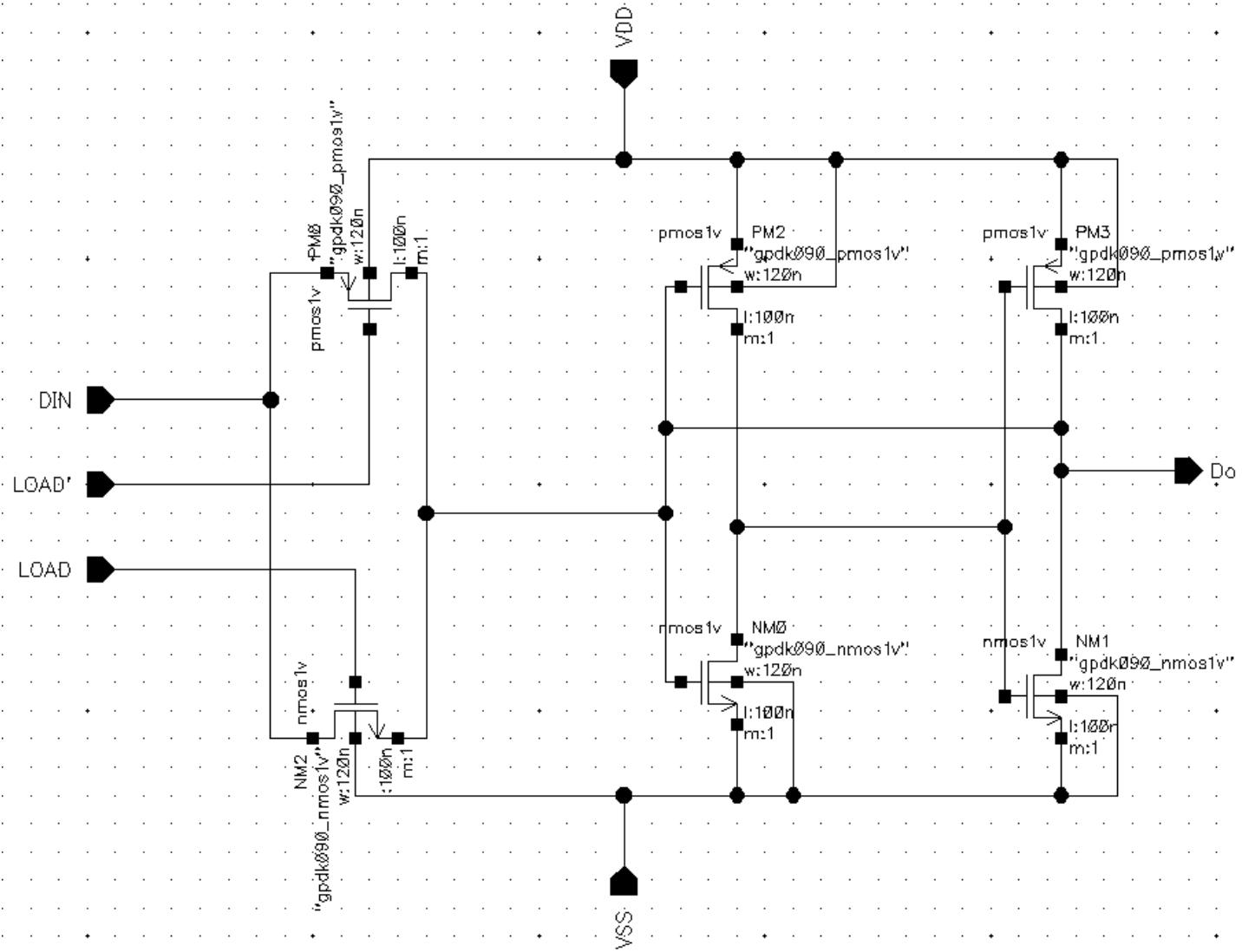


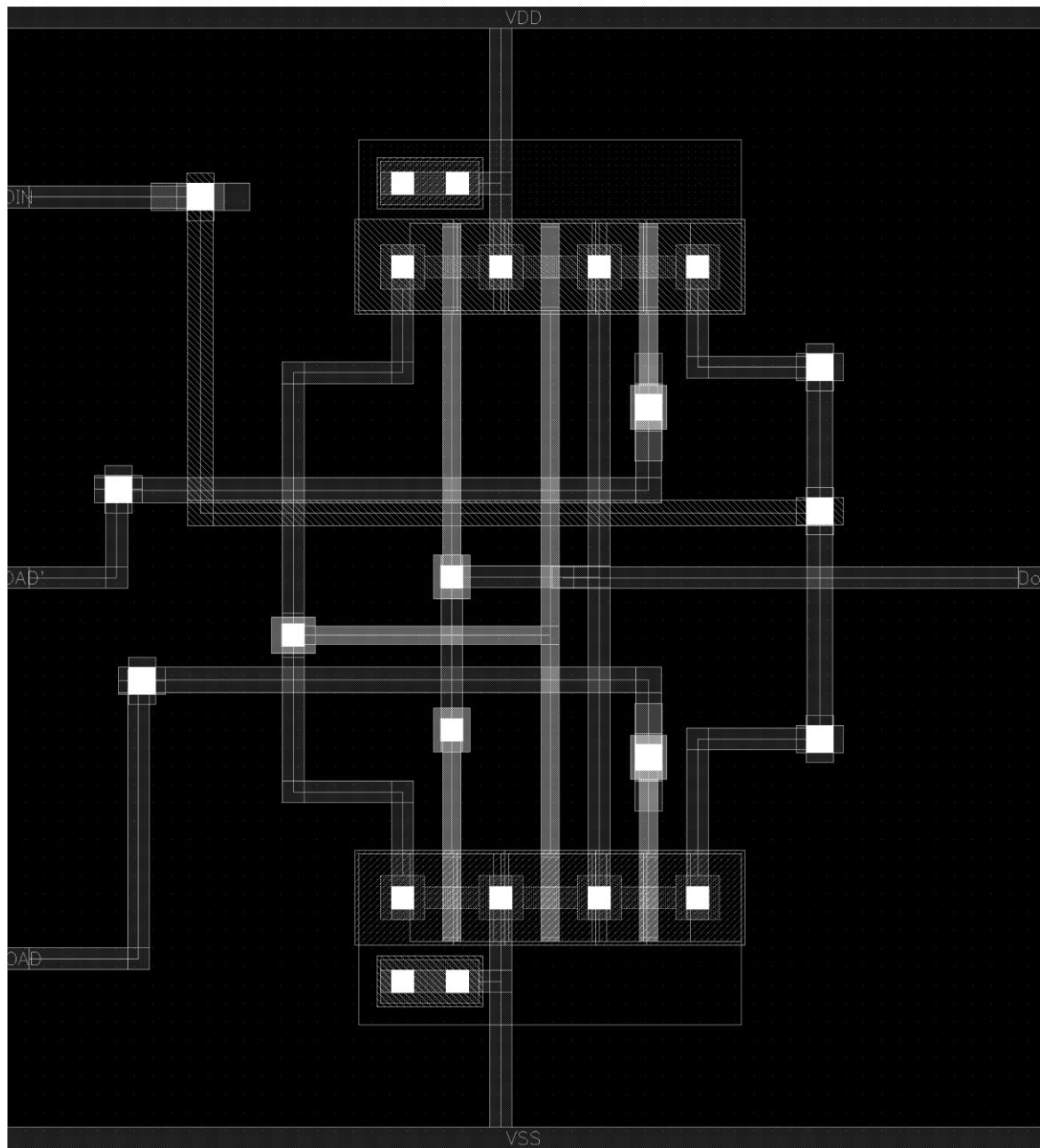
#### 4.3IPNAND



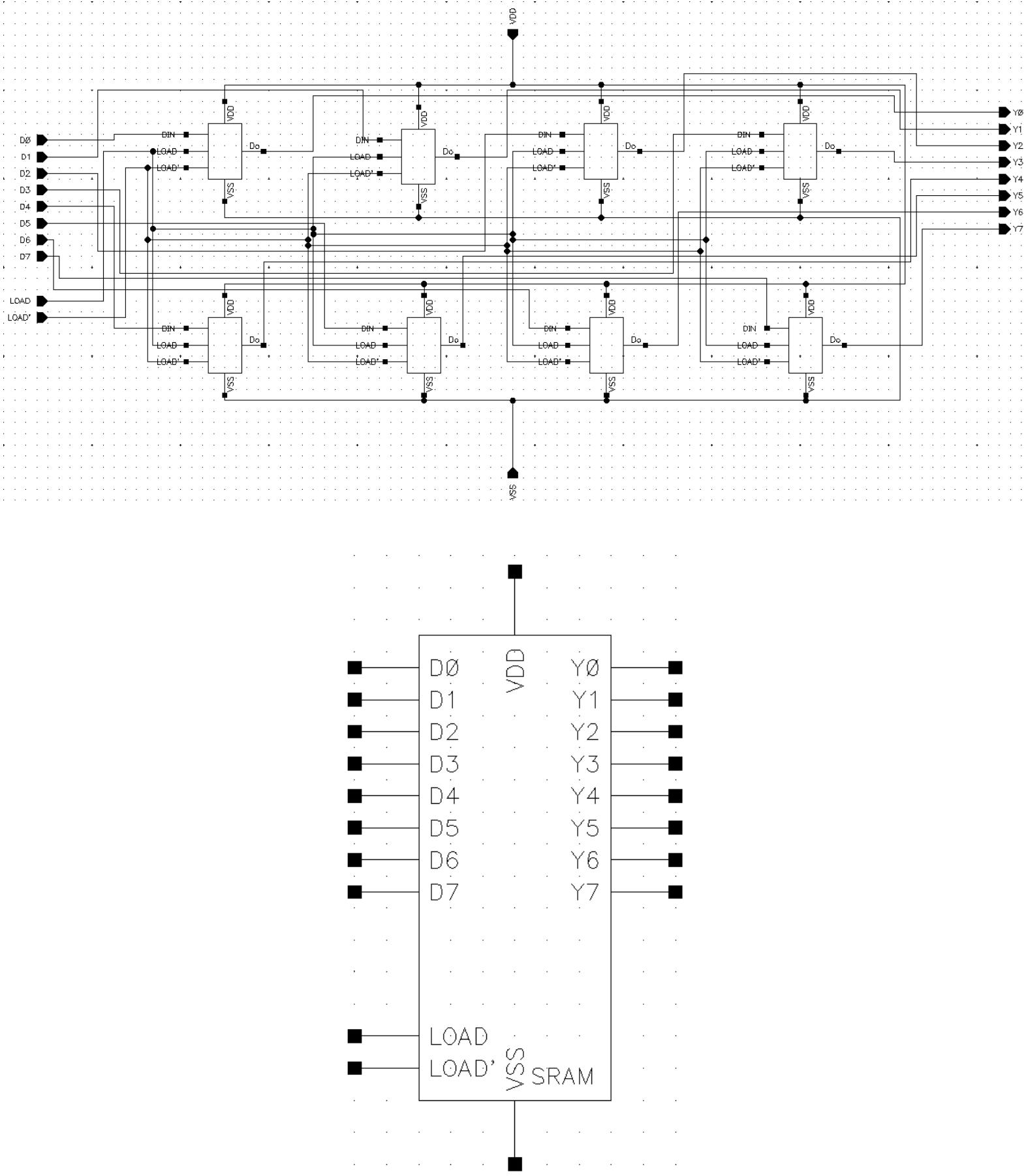


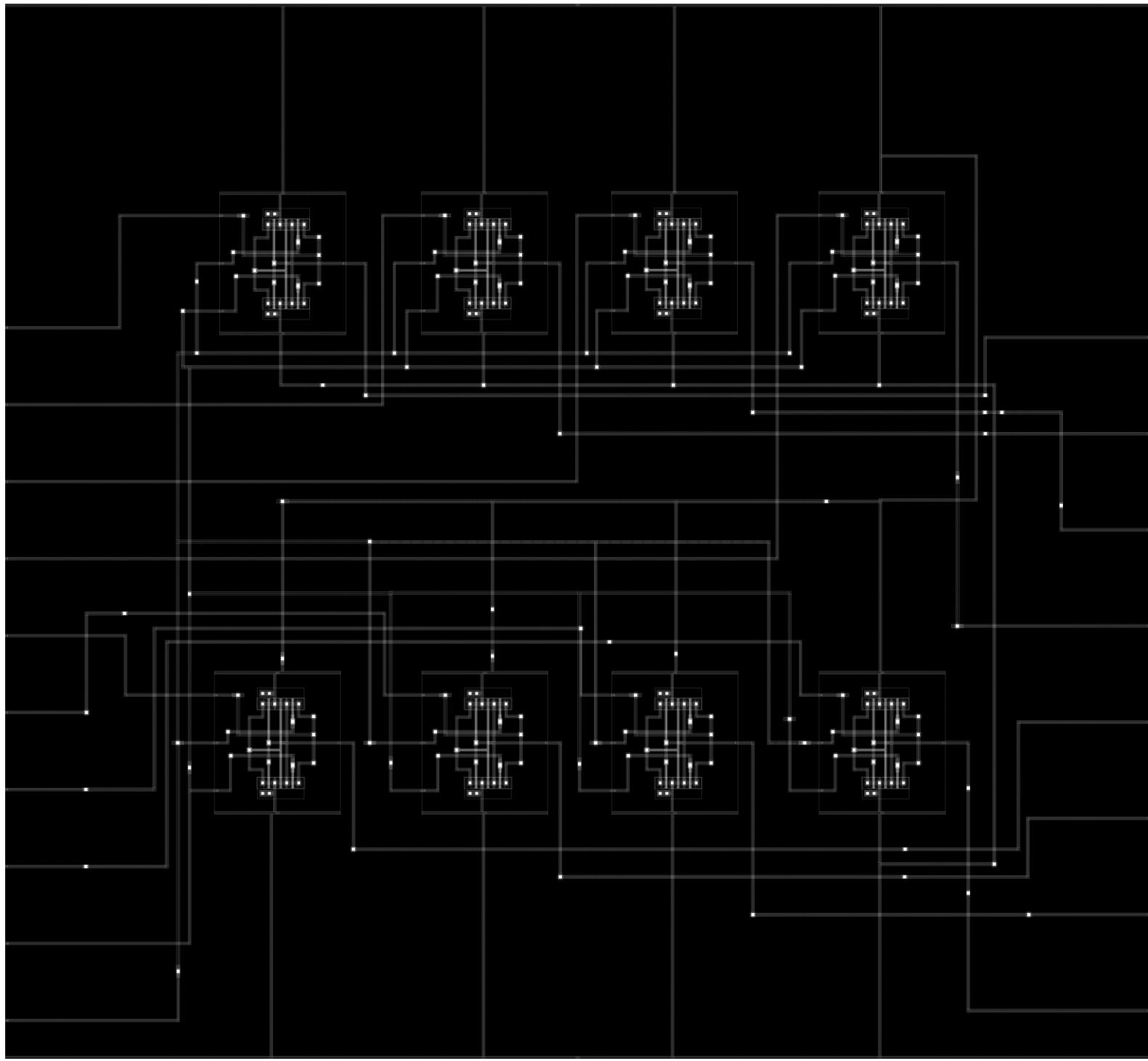
## 5. 6TSRAMCell



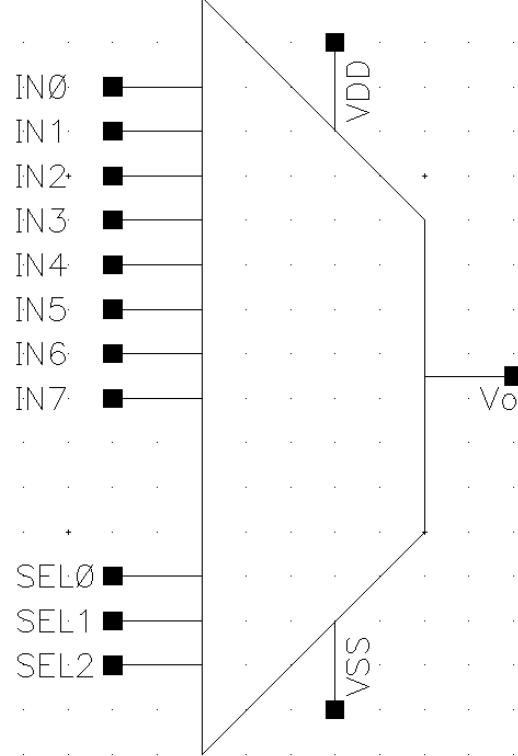
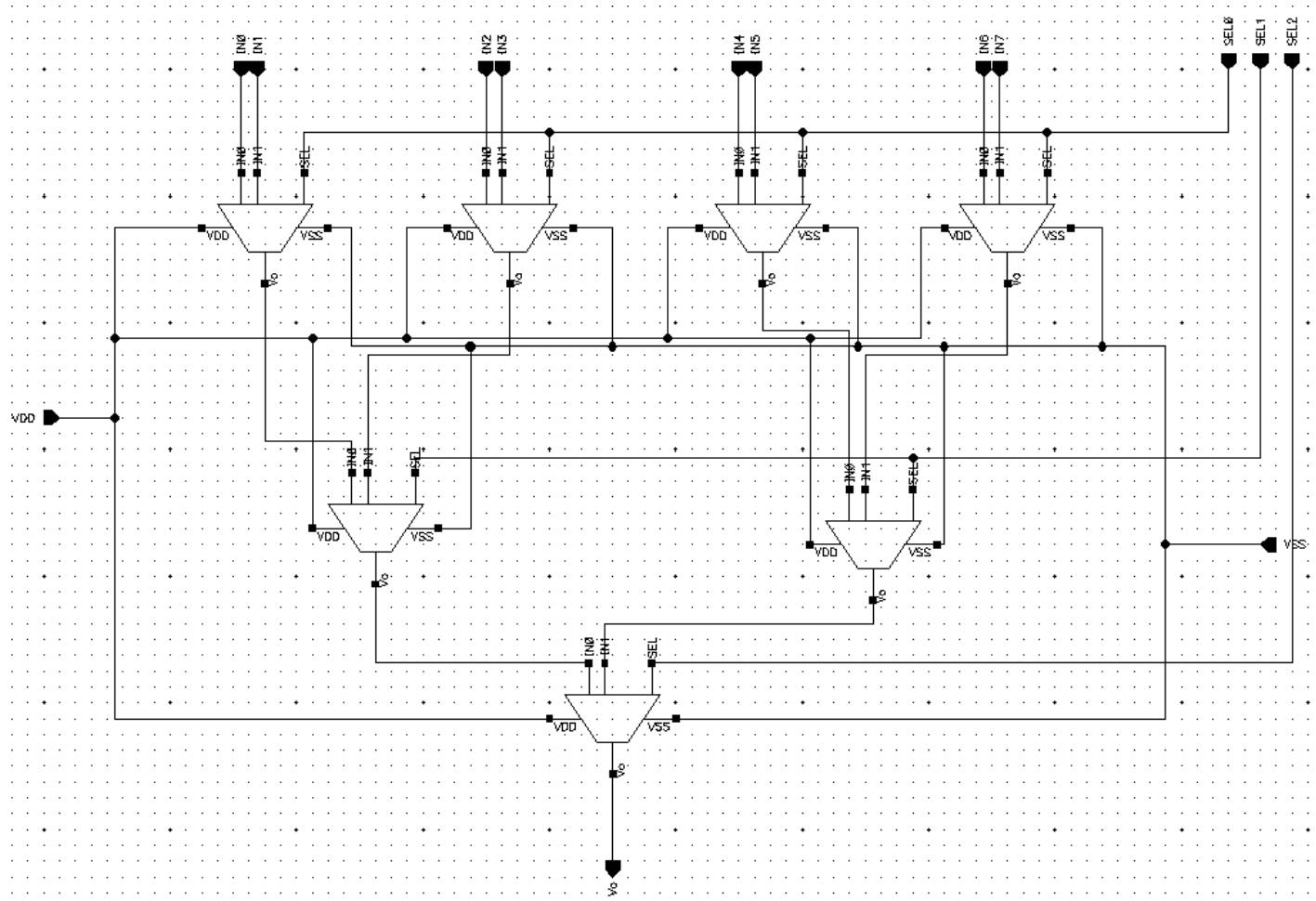


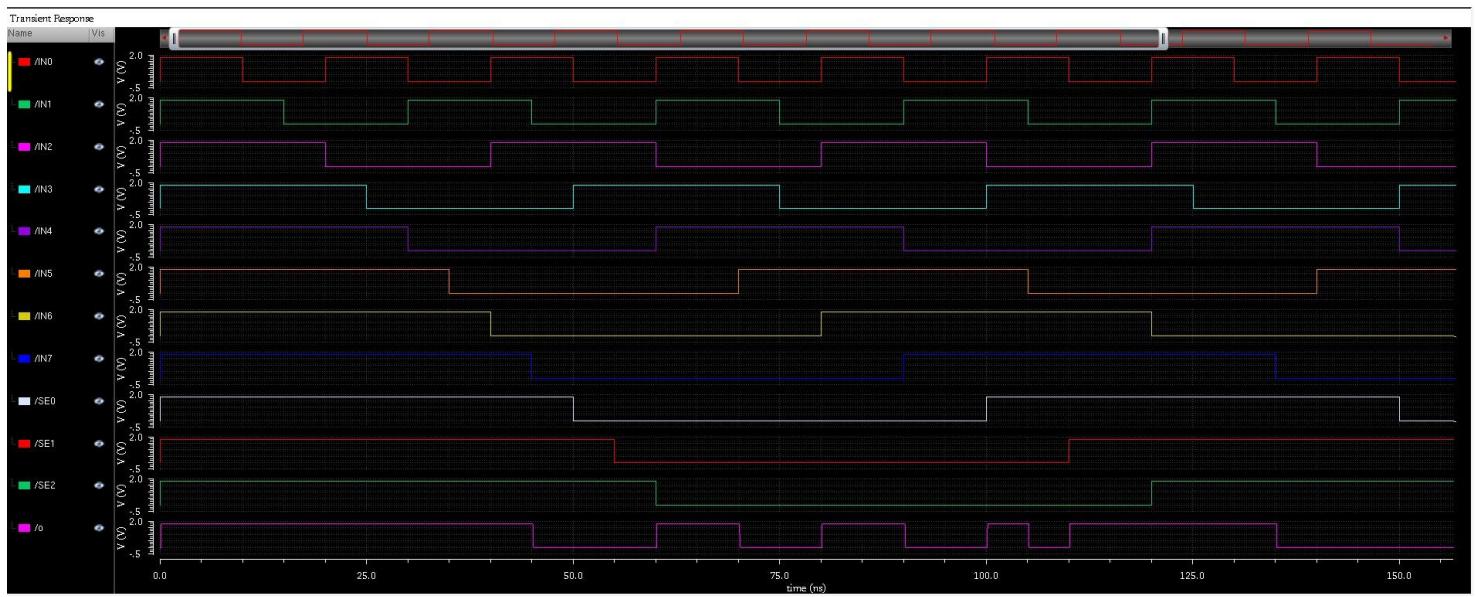
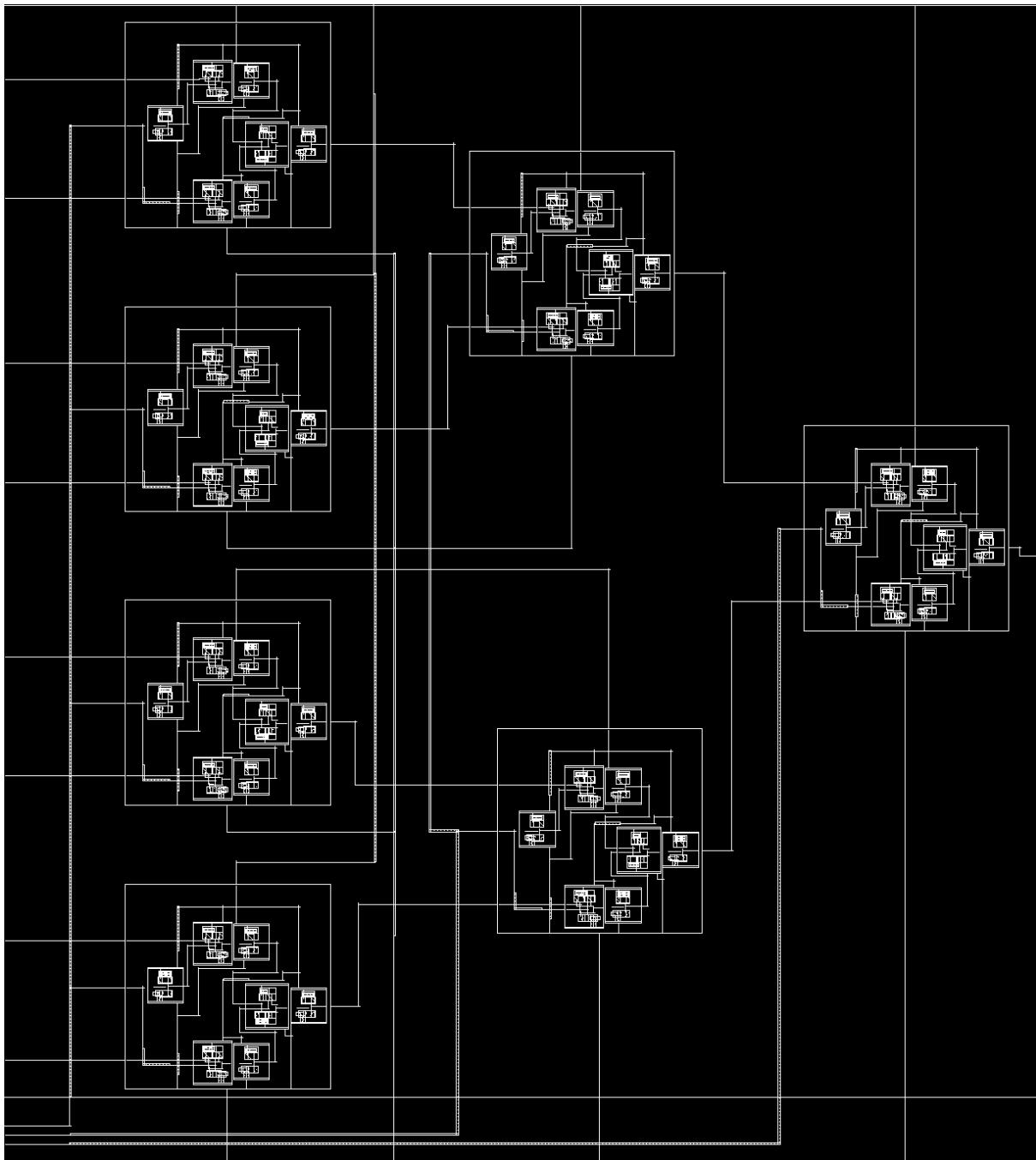
## 6. 8bit SRAM



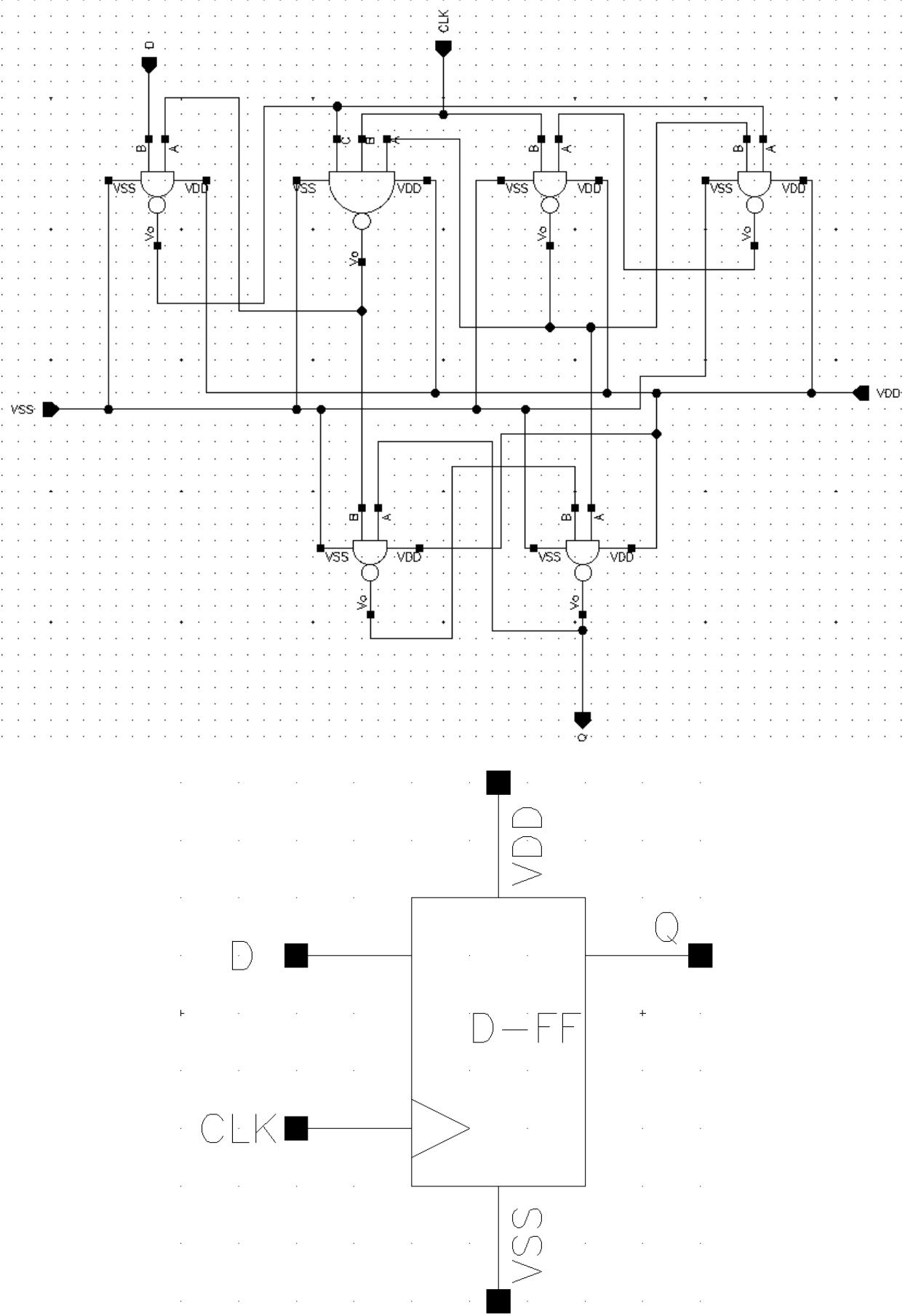


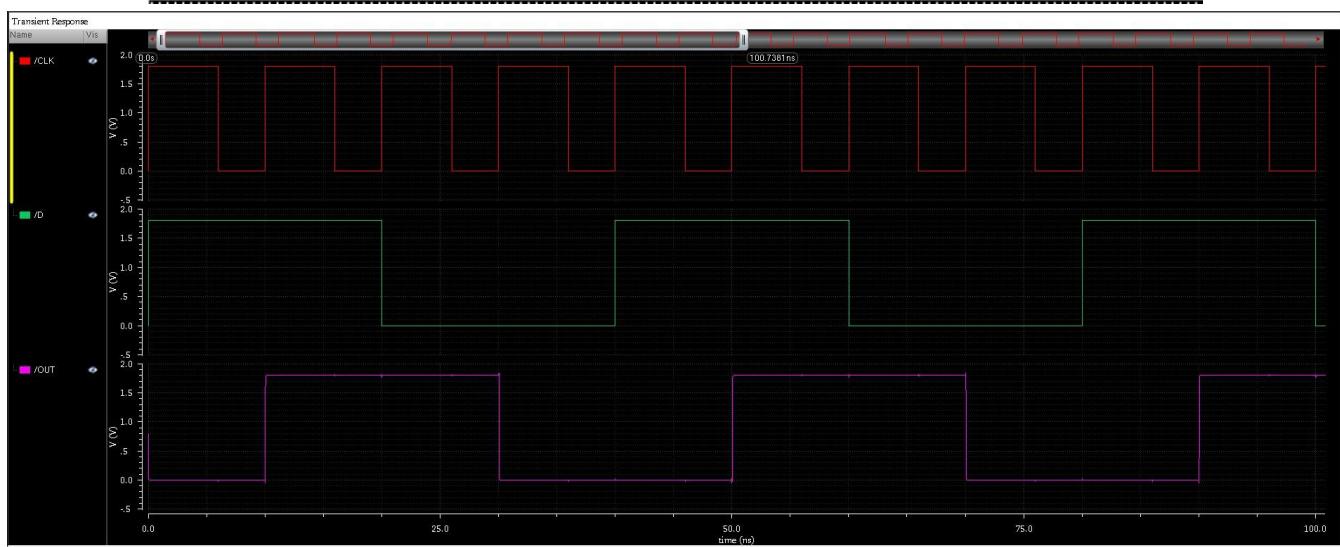
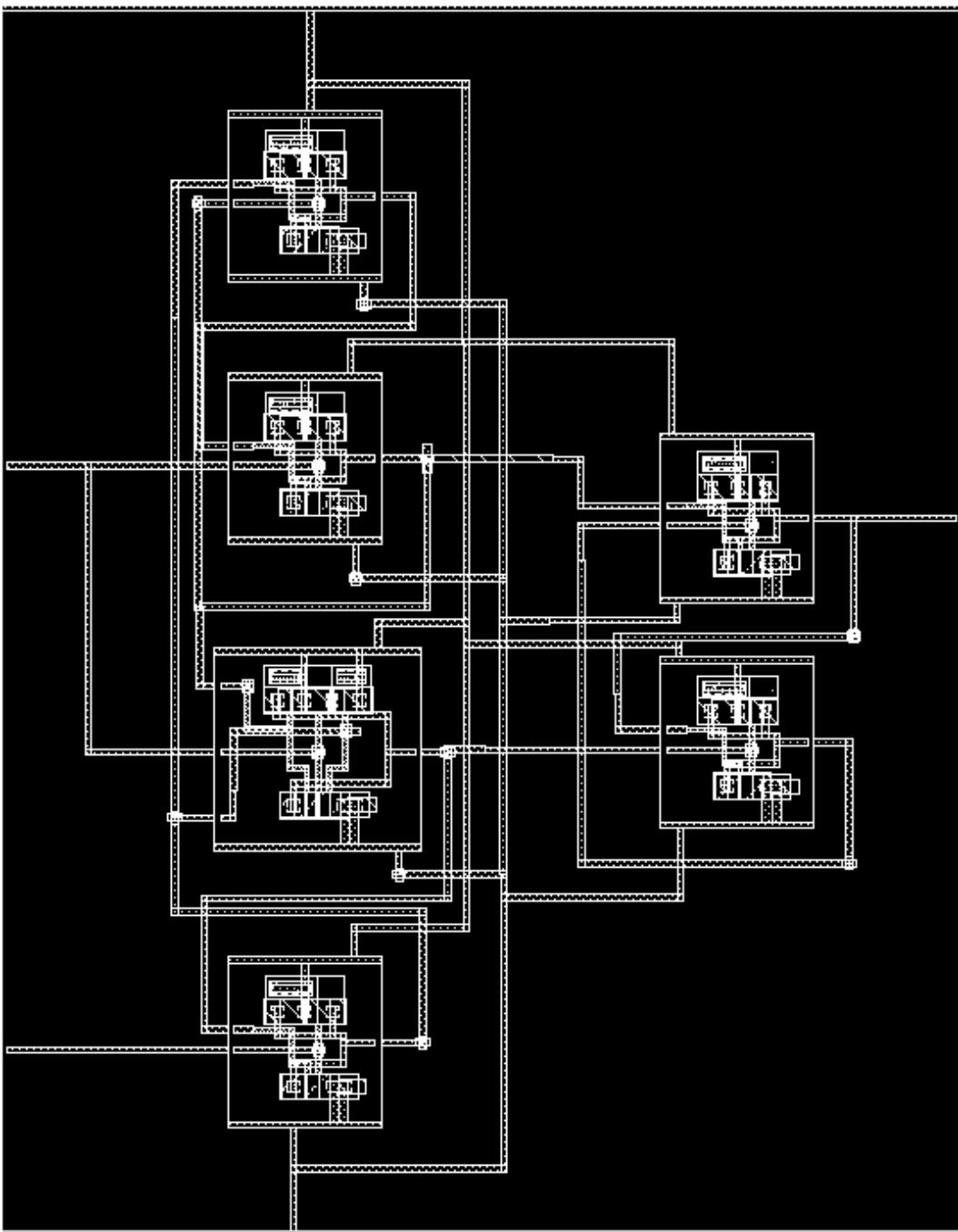
## 7. 8x1MUX



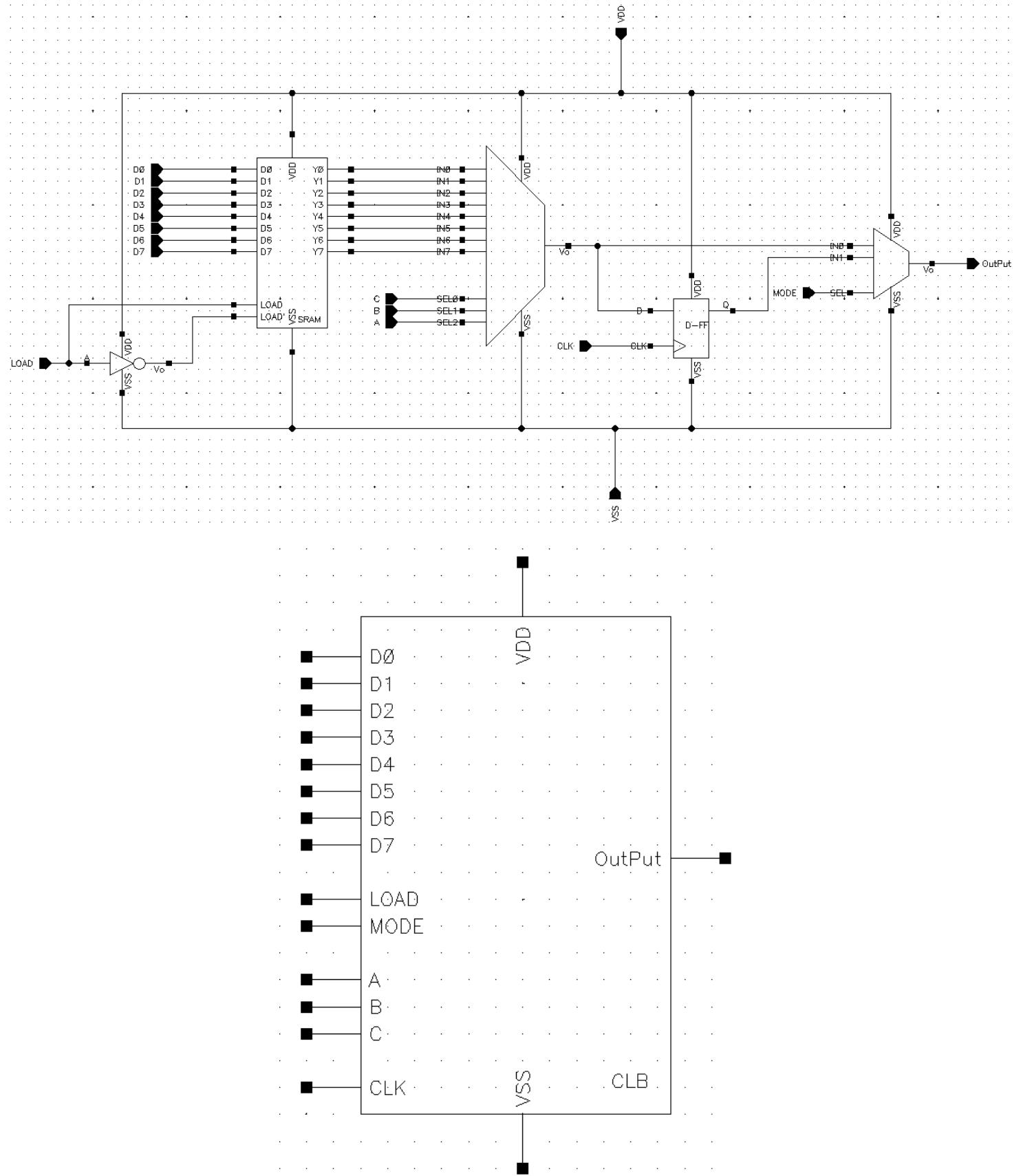


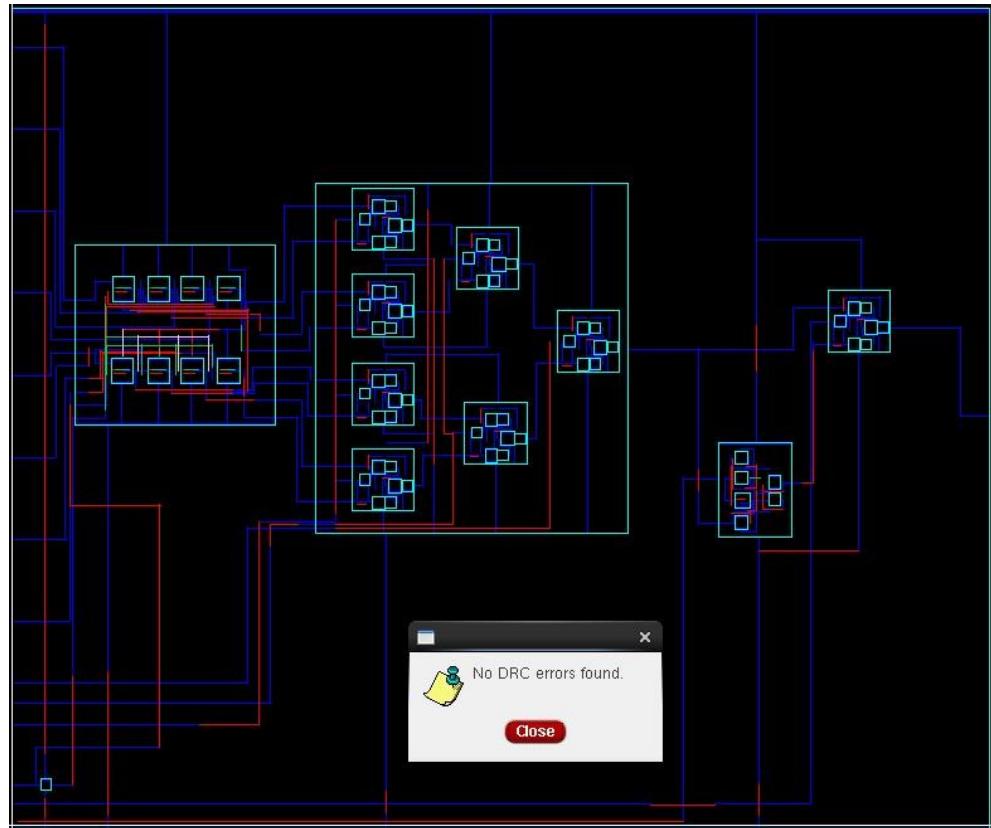
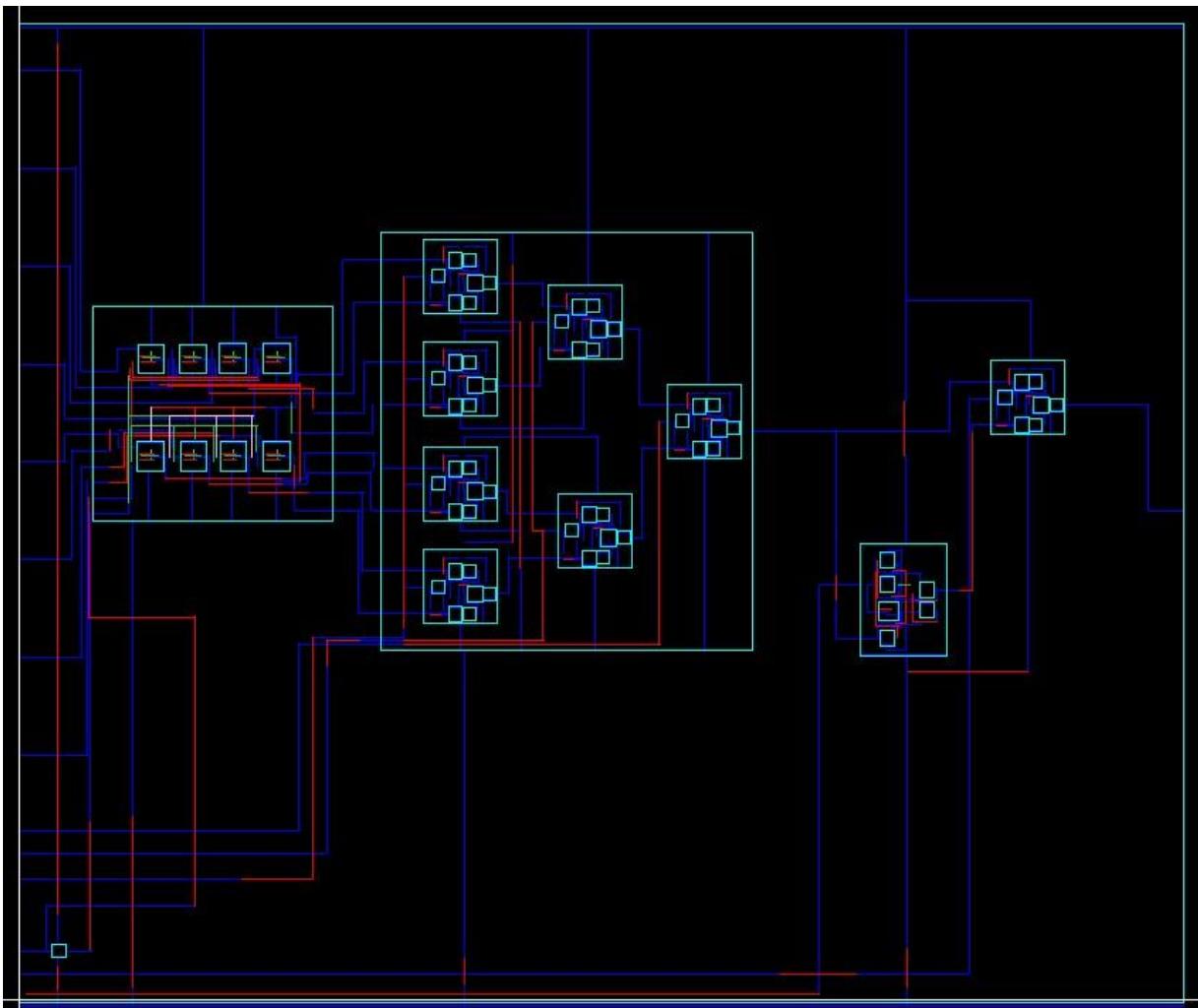
## 8. D-FF (D Flip-Flop)

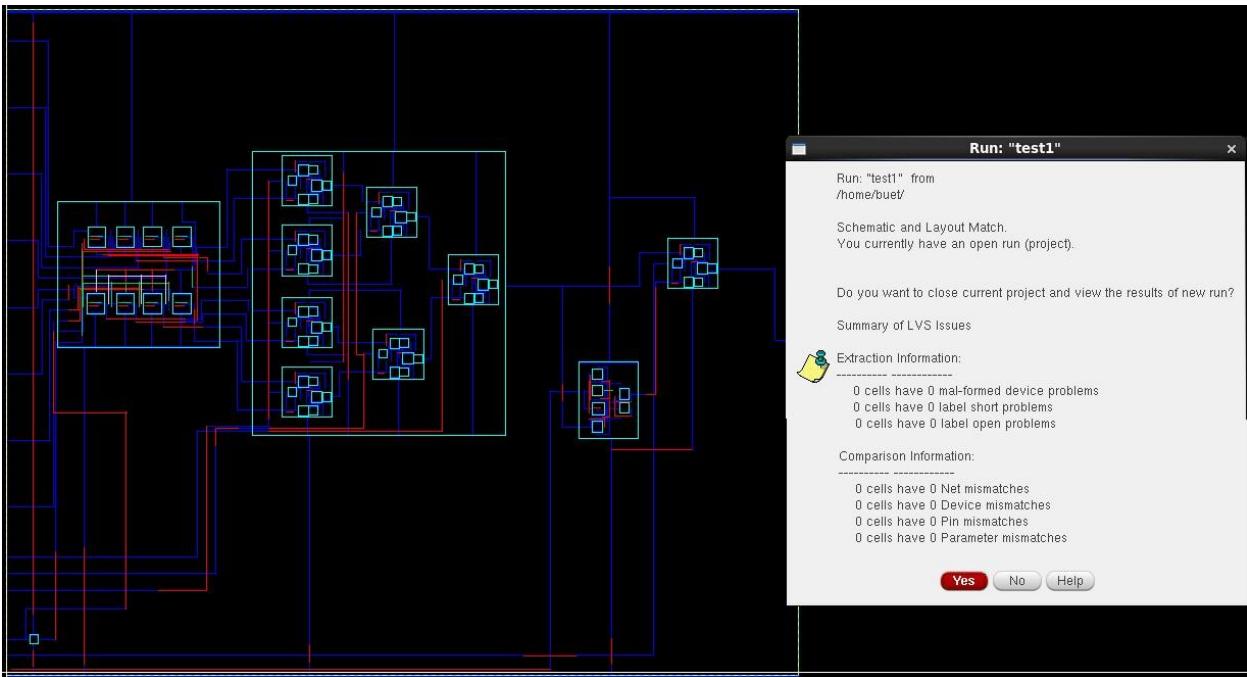




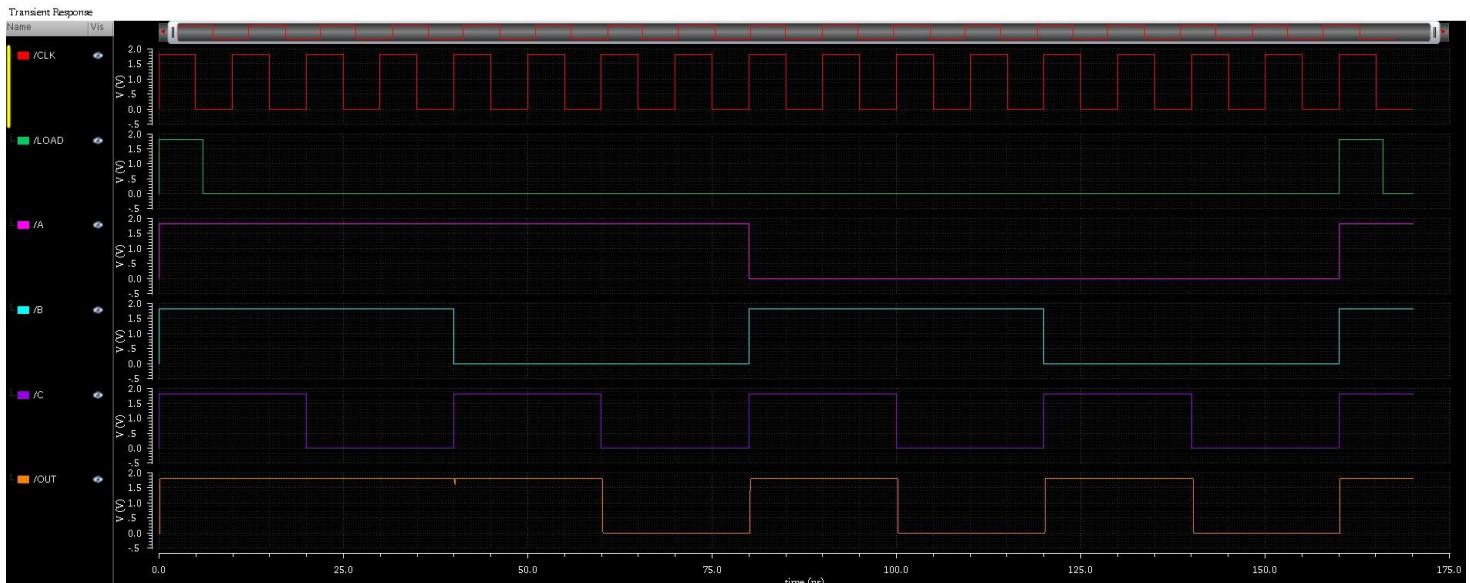
## 9. CLB



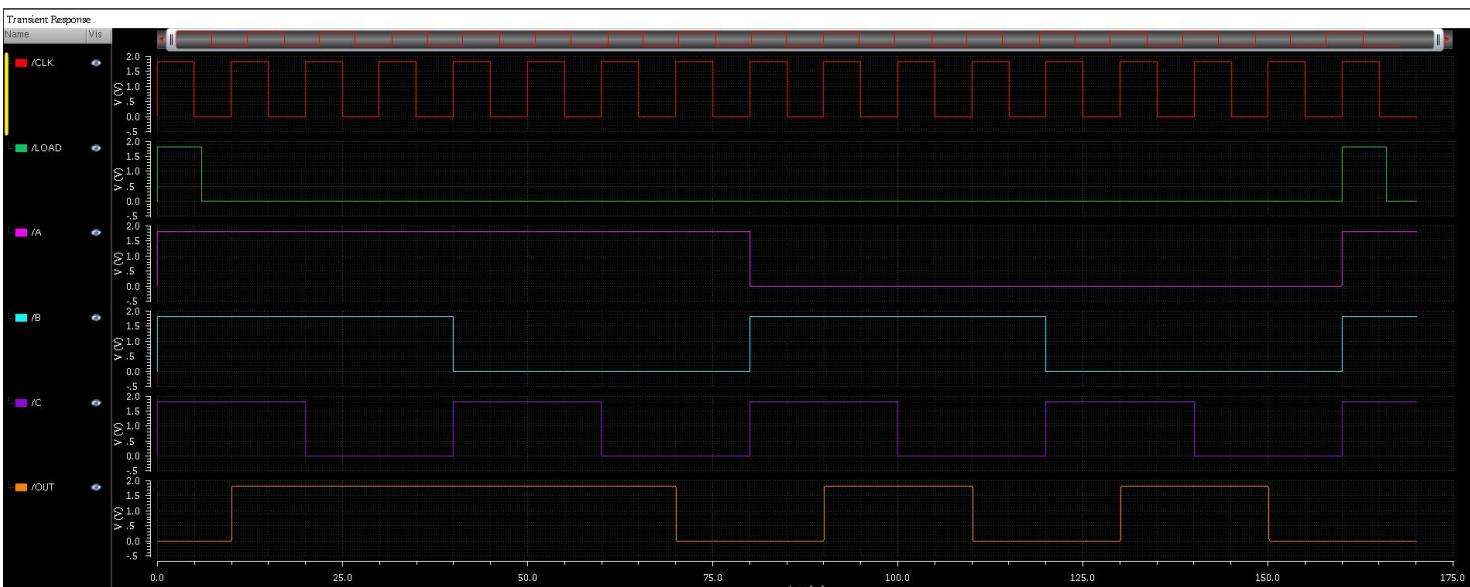




## CLB Combinational Analysis



## CLB Sequential Analysis



# Circuit Analysis

## D-FF Setup Fixing by adding delay



Failed Setup when clock lags D for 10ps

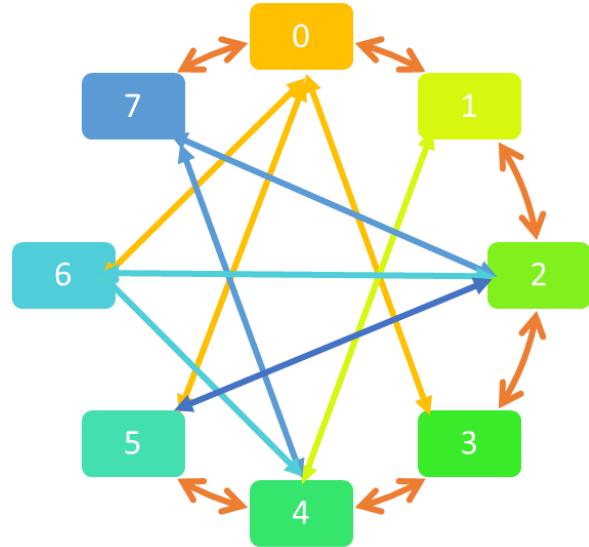


Setup Success when clock lags D for 11ps

## Worst Case Delay Finding and calculation

We had to find the state delay of every combination when the output changes when input changes simultaneously. Therefore, the state diagram looks like this according to the custom-made truth table:

States	A	B	C	AB + C
state0	0	0	0	0
state1	0	0	1	1
state2	0	1	0	0
state3	0	1	1	1
state4	1	0	0	0
state5	1	0	1	1
state6	1	1	0	1
state7	1	1	1	1

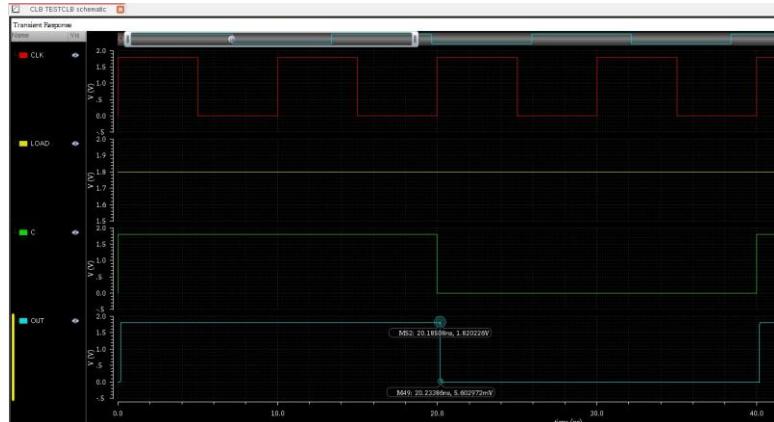
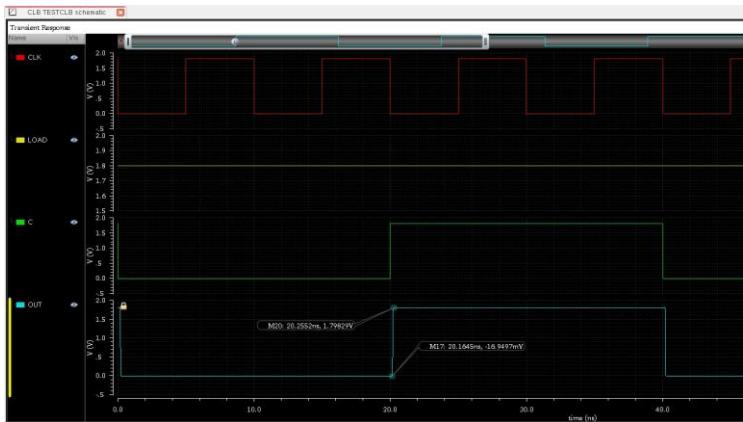


After that we create circuits that change states between them and subtract the upper level and lower-level time to find the rise time and fall time delay. We taken the 1<sup>st</sup> cycle where the output changes with respect to the input or else the delay add after every cycle making it difficult to find the exact delay of rise and fall. And thus, it leads us to this table where we found our worst-case delay is in state transition of state4 and state3

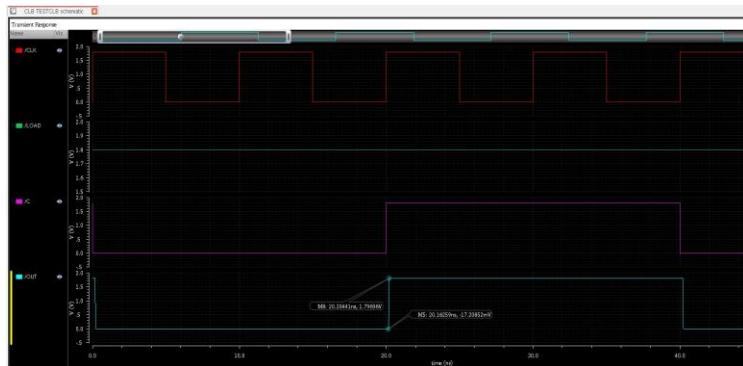
State transition	Rise Time (ps)	Fall Time (ps)
0 to 1	36.75	48.78
0 to 3	45.8	38.33
0 to 5	39.89	42.48
0 to 6	36.64	35.45
0 to 7	41.08	42.6
2 to 1	35.6	42.8
2 to 3	41.82	45.52
2 to 5	25.15	46.68
2 to 6	26.82	37.21
2 to 7	25.34	44.68
4 to 1	42.8	51.59
4 to 3	44.2	55.49
4 to 5	40.97	45.9
4 to 6	49.26	45.16
4 to 7	48.06	43.92

Table of State Transition delays

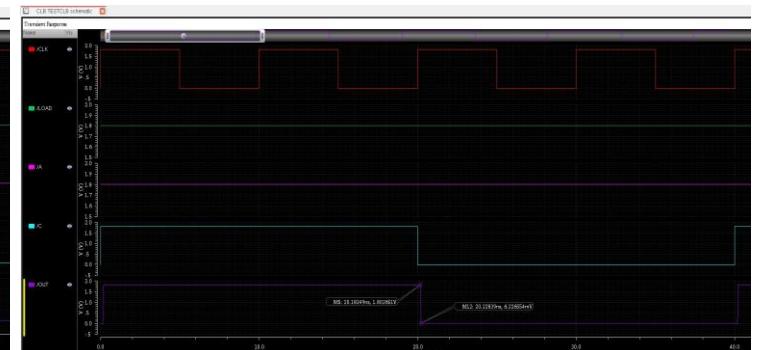
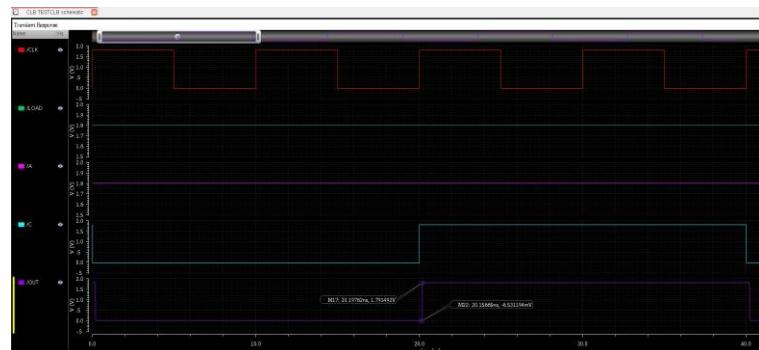
Some examples of the curves that show how we got the values:



Rise and Fall time delay when transition between state0 to state1

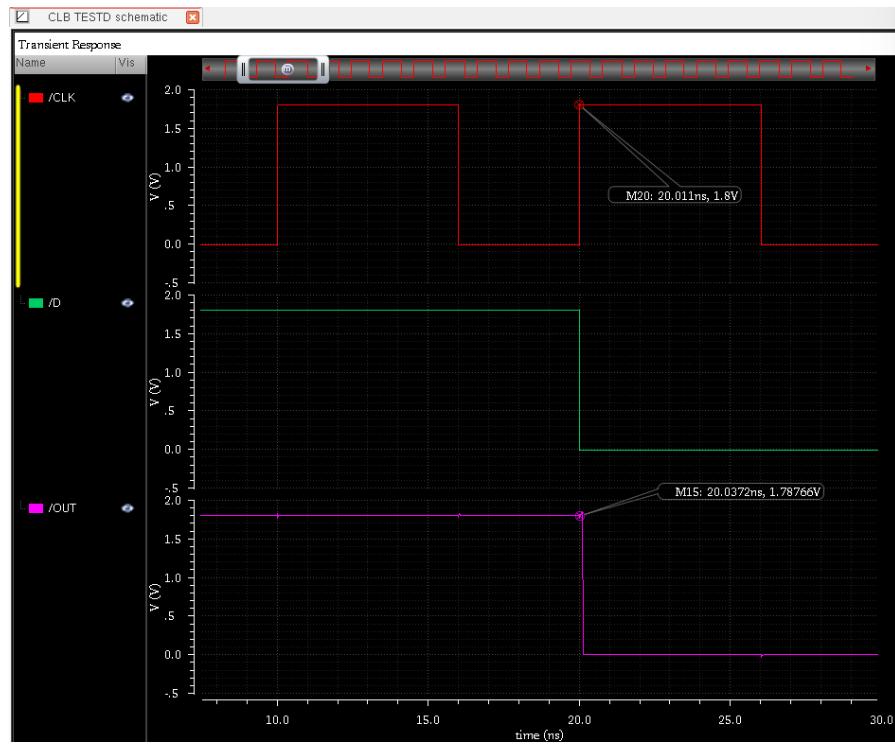


Rise and Fall time delay when transition between state2 to state3



Rise and Fall time delay when transition between state4 to state5

## Clock to Q delay



From the Plot the Q changes at 26.2ps after the active clock edge.

Therefore we get our optimal period,

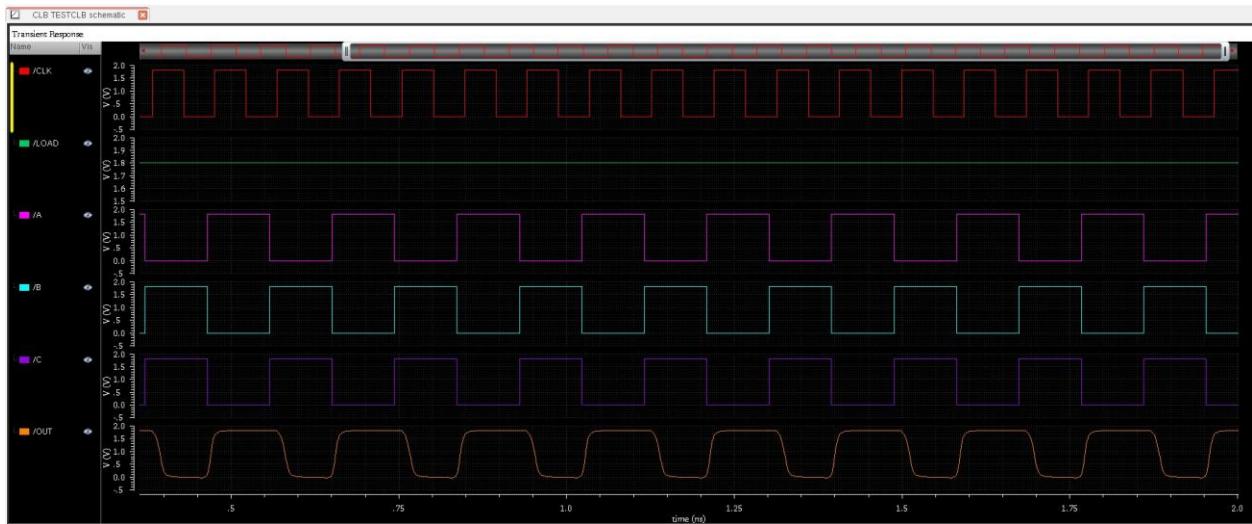
$$\text{Minimum clock period} = \text{Total combinatorial delay} + \text{Clock to Q delay} + \text{Setup time}$$

$$= (55.49 + 26.2 + 11)\text{ps}$$

$$= 92.69\text{ps}$$

$$\approx 93 \text{ ps}$$

## Critical Frequency



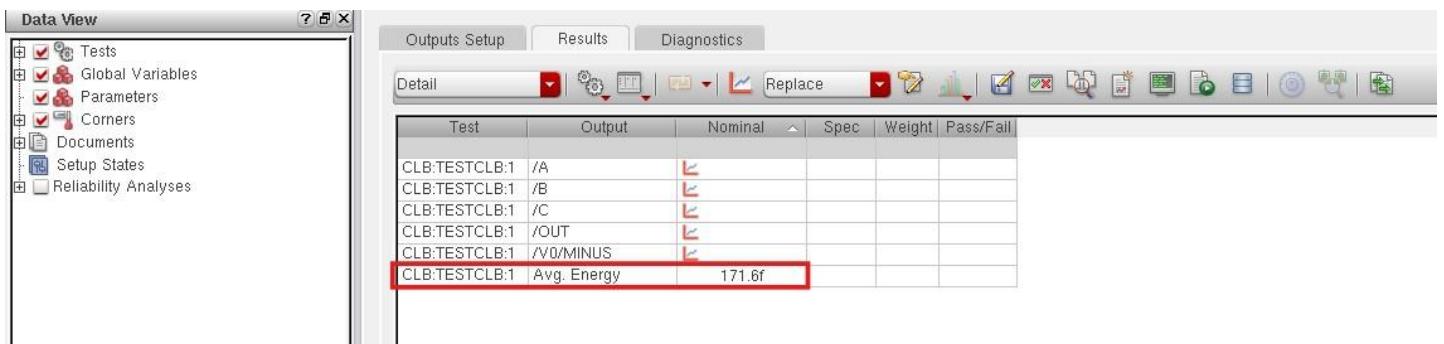
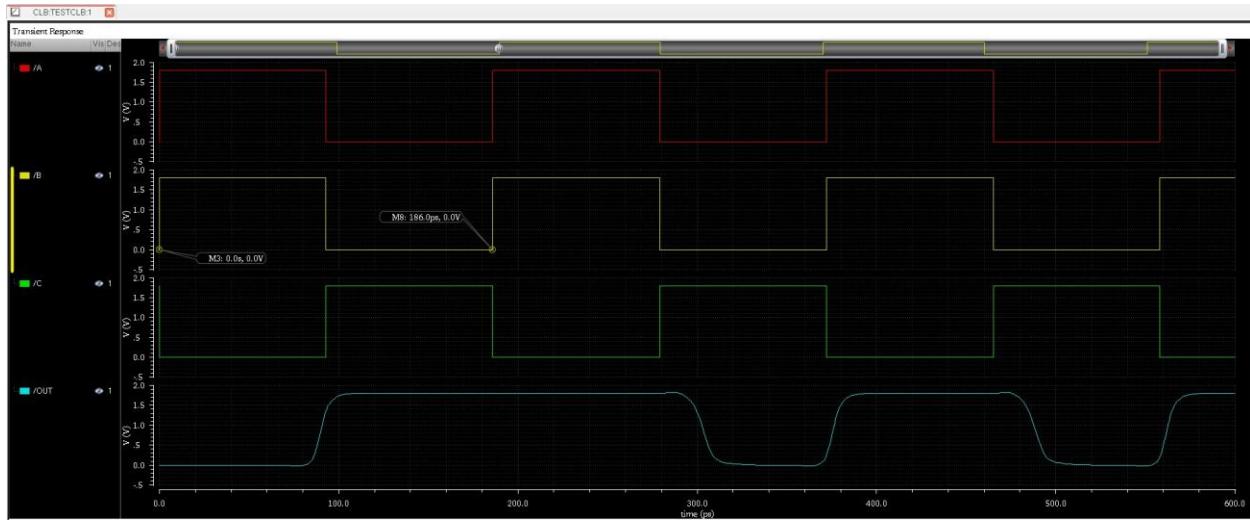
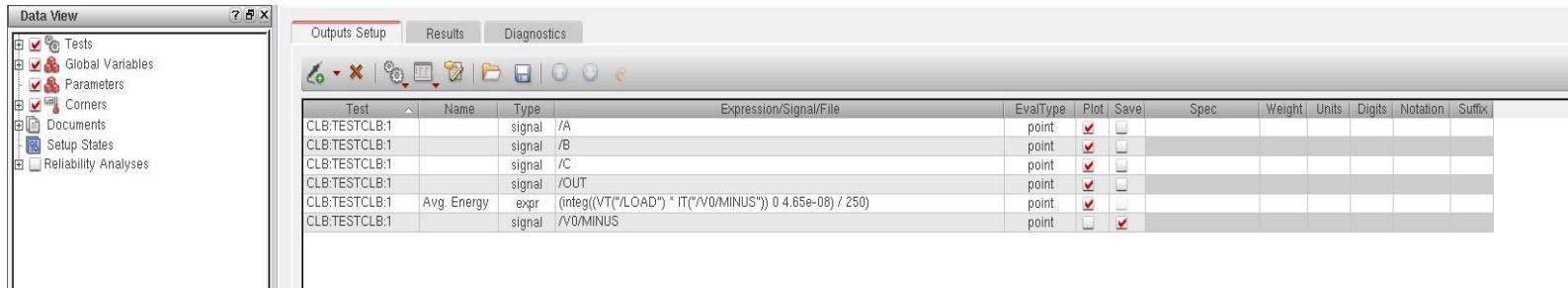
When we apply **86ps** to **93ps** (Our Minimum clock period) the output barely follows the input. But when we apply **85ps** as clock period we get this:



Which is distorted output. Therefore, after multiple simulation we get **86ps** as our critical frequency for this CLB in sequential mode.

## Avg. Energy Calculation

For the Average Energy Calculation, we used ADE XL and taken the current & voltage that our CLB is drawing out. We calculated how much energy is consuming at 500 cycles of the input choosing our optimal clock period for this calculation. The outputs are given below:



Therefore, we can see our CLB consumes **171.6fJ** of average energy.

## Layout Area Calculation

To determine this, we use SKILL IDE and measured the CLB layout in  $m^2$ . After creating the .il file we open the CIW window to load that file and call the function. But before all this we must run the layout so that CIW can fetch it from the cache memory. The code and the outputs are given below:

Cadence® SKILL IDE

File Edit Tools Options Debug Window Help

cadence

Steps: 1 Levels: 1 Search

AREAINMETER2.il

```
procedure( calculateLayoutAreaInMeters()
    let( (cv bbox x_min y_min x_max y_max width height area_m2)
        cv = geGetEditCellView() ; Get current layout cell view
        bbox = cv~>bBox           ; Get bounding box in microns

        if( bbox then
            ; Extract coordinates
            x_min = car(car(bbox))
            y_min = cadr(car(bbox))
            x_max = car(cadr(bbox))
            y_max = cadr(cadr(bbox))

            ; Calculate width & height in microns
            width = abs(x_max - x_min)
            height = abs(y_max - y_min)

            ; Area in  $\mu\text{m}^2$ 
            area_um2 = width * height

            ; Convert to  $\text{m}^2$  ( $1 \mu\text{m}^2 = 1e-12 \text{ m}^2$ )
            area_m2 = area_um2 * 1e-12

            printf("Layout Area: %g  $\text{m}^2\n$ " area_m2)
        else
            printf("Bounding box not found. Is the layout empty?\n")
    )
)
```

Line: 30 Col: 1 Top-Level # 1

```
-->-----  
load("/home/buet/CLB/AREAINMETER2.il")  
t  
hiCaughtControlC  
calculateLayoutAreaInMeters()  
Layout Area: 5.28976e-08  $\text{m}^2$   
t
```

```
load("/home/buet/CLB/AREAINMETER2.il")  
calculateLayoutAreaInMeters()
```

Therefore, the area of the CLB layout is **5.28976 x 10<sup>-8</sup> m<sup>2</sup>**.

### **FOM Calculation**

We know the equation of FOM is:

$$\mathbf{FOM} = E_{avg} \times t_{delay} \times Area$$

From our previous calculation we have,

$$E_{avg} = 171.6 \times 10^{-15} \text{ J}$$

$$t_{delay} = 93 \times 10^{-12} \text{ s}$$

$$Area = 5.28976 \times 10^{-8} \text{ m}^2$$

Therefore, our FOM is,

$$\begin{aligned}\mathbf{FOM} &= E_{avg} \times t_{delay} \times Area \\ &= 171.6 \times 10^{-15} \times 93 \times 10^{-12} \times 5.28976 \times 10^{-8} \\ &= \mathbf{8.44182 \times 10^{-31} \text{ J}\cdot\text{s}\cdot\text{m}^2}\end{aligned}$$

### **Conclusion**

Such a design, layout, and verification process is crucial to becoming proficient in digital IC design using Cadence Virtuoso. Design and validation of real components such as logic gates, multiplexers, flip-flops, and memory cells furnish a designer with fantastic knowledge on real-world VLSI workflows. It solidifies the theoretical foundations of this concept but is complemented further by providing the engineer with practical experience in sound and efficient production of digital circuits for inclusion in systems. These activities contribute towards building a solid foundation in digital design procedures that are core to success in the semiconductor industry and research.