ELEC-451&540 HOMEWORK-3

Group Name: DreamTeam-VLSI

Members:

Ömer Faruk Yurtseven- 244204004032

Efe Bayrakçeken- 210102002026

Erhan Gök- 200102002027

Beyzanur Cam- 210102002037

Question 1

Design the gates and the circuits at schematic level to satisfy the defined delay constraints. During the design phase, estimate the propagation delay and the logic effort for each design (Hint: You may use the INV results that you obtained for the other circuits.) Assume that the sizing should be performed to equalize the fall and rise times of each gate.

The requirements that we were asked is shown in the table below.

LOGIC GATE/CIRCUIT	Delay for FO4 INV load
INV	<3ps
NAND2-NAND3-NAND4	<6ps
NOR2-NOR3-NOR4	<7ps
XOR2	<12ps
XNOR2	<12ps
2X1 MUX	<30ps

Schematic Design

Schematic designs were created using XSchem, focusing on estimating propagation delays based on logical effort.

The sizing of these transistors was calculated in the later parts of the report.

Inverter

The inverter just consists of 2 transistors. The sizings were made to be the same resistance for the pmos and the nmos.

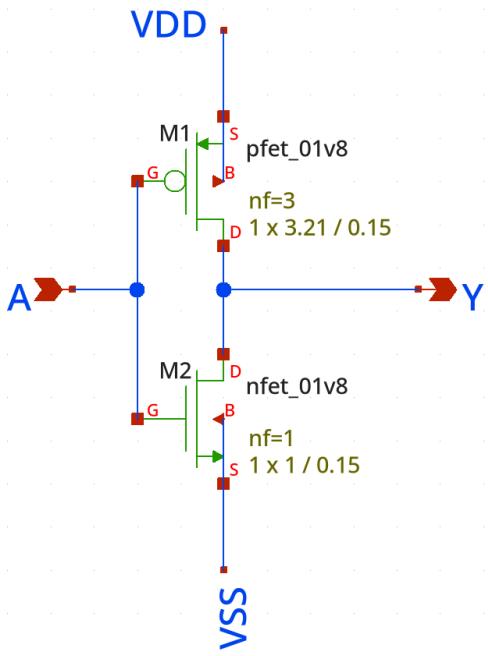


Figure 1Inverter Schematic

NAND

Nand gates have series N-Fets for the pull-down network and pararel P-Fets for its complement. This results in the increase in the sizing of the N-fets proportional to the number of series mosfets.

The figure below shows the schematic for NAND2

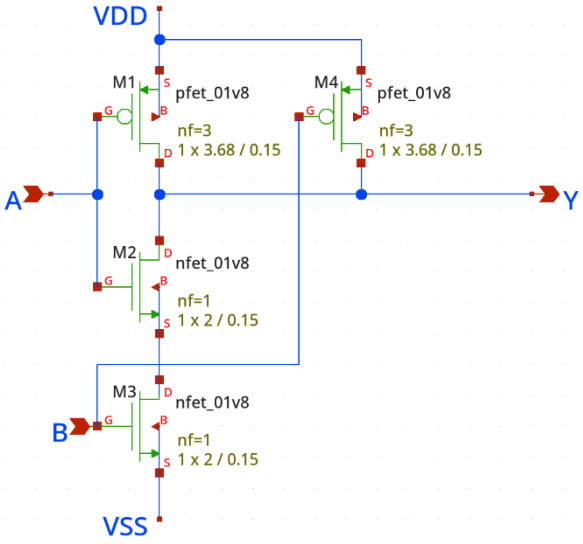


Figure 2 NAND2 Schematic

As can be seen, there are 4 mosfets, 2 for each input. The input capacitance is 5.21C according to this schematic.

The schematic is basically the same for NAND3.

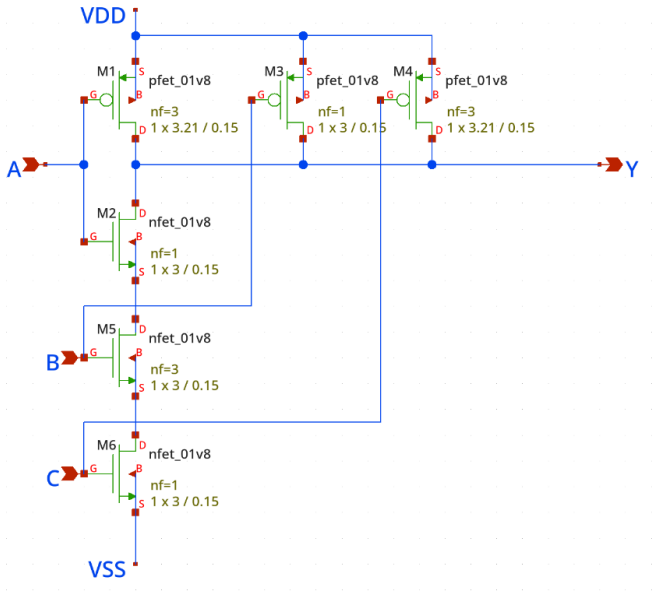


Figure 3 NAND3 Schematic

Nand4 has very large sizes for the nfets.

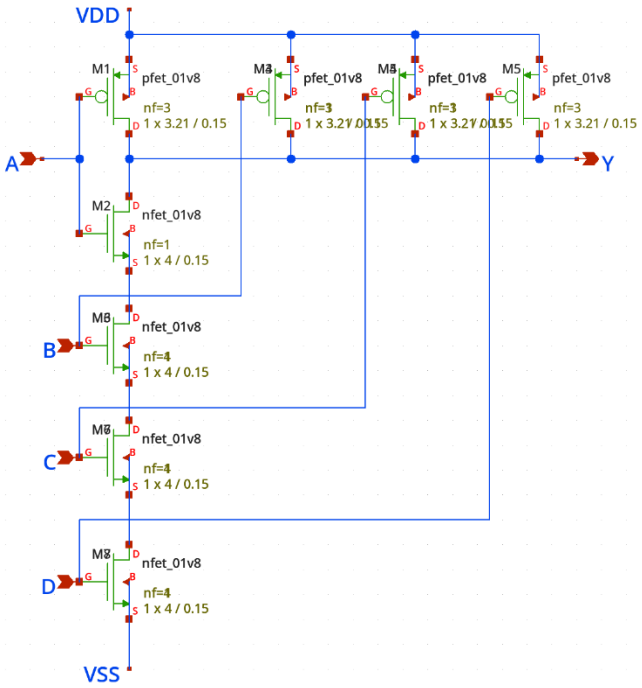


Figure 4 NAND4 Schematic

NOR

NOR gates are built using parallel N-FETs in the pull-down network and series P-FETs in the pull-up network to create the complementary function. Because the P-FETs are in series, the resistance in the pull-up network increases as more P-FETs are added. This makes it harder for the gate to drive the output high quickly. To fix this, the P-FETs need to be made larger to reduce their resistance, and the sizing has to increase based on how many P-FETs are in series. While this helps keep the gate's performance balanced, it also means using more area and power since

larger P-FETs take up more space and add extra capacitance. These are important trade-offs to think about when designing NOR gates, especially when speed and efficiency are a priority.

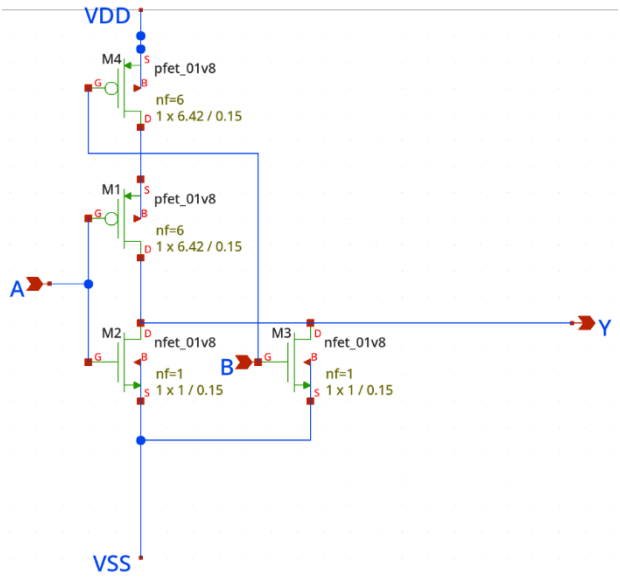


Figure 5 NOR2 Schematic

The NOR2 gate consists of two parallel N-FETs in the pull-down network and two series P-FETs in the pull-up network. This configuration allows the gate to output a low signal if either or both inputs are high. To maintain performance, the P-FETs need to be sized larger to counter the increased resistance from their series connection. The NOR2 gate is commonly used in digital designs for its simplicity and functionality.

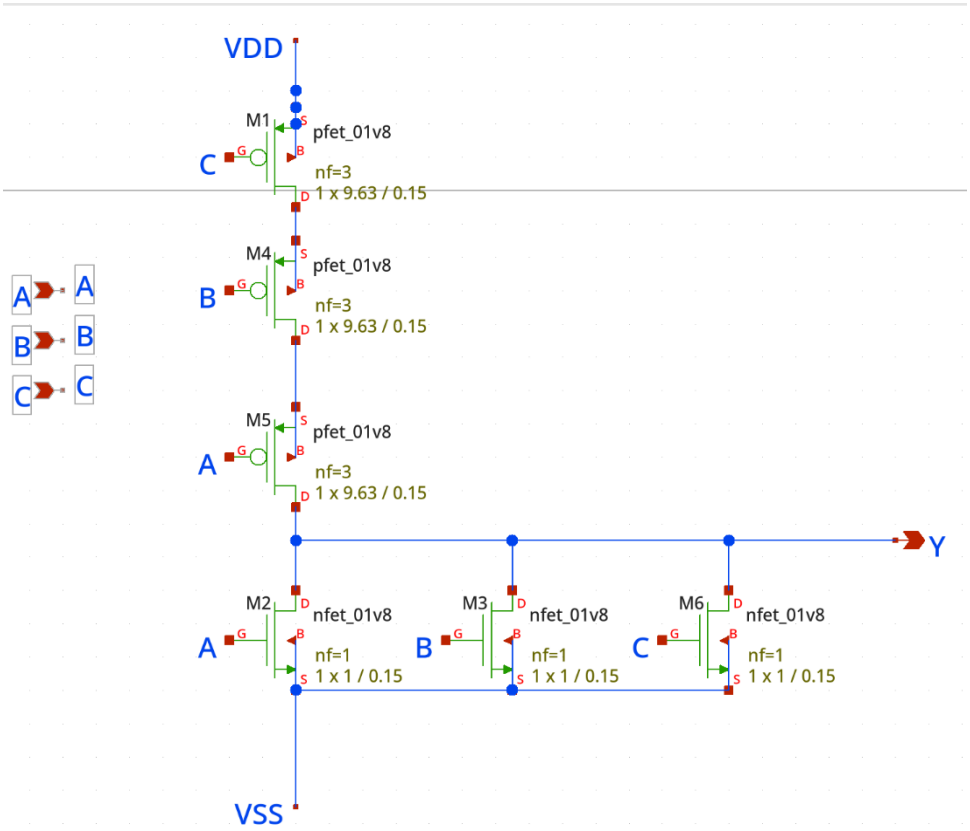


Figure 6 NOR3 Schematic

In the NOR3 gate, the pull-down network includes three parallel N-FETs, and the pull-up network consists of three series P-FETs. As the number of series P-FETs increases, the resistance in the pull-up network also increases, making it slower to drive the output high. To address this, the P-FETs are sized even larger compared to those in the NOR2 gate. However, this comes at the cost of higher area and power usage, which is a consideration in designs where NOR3 gates are required.

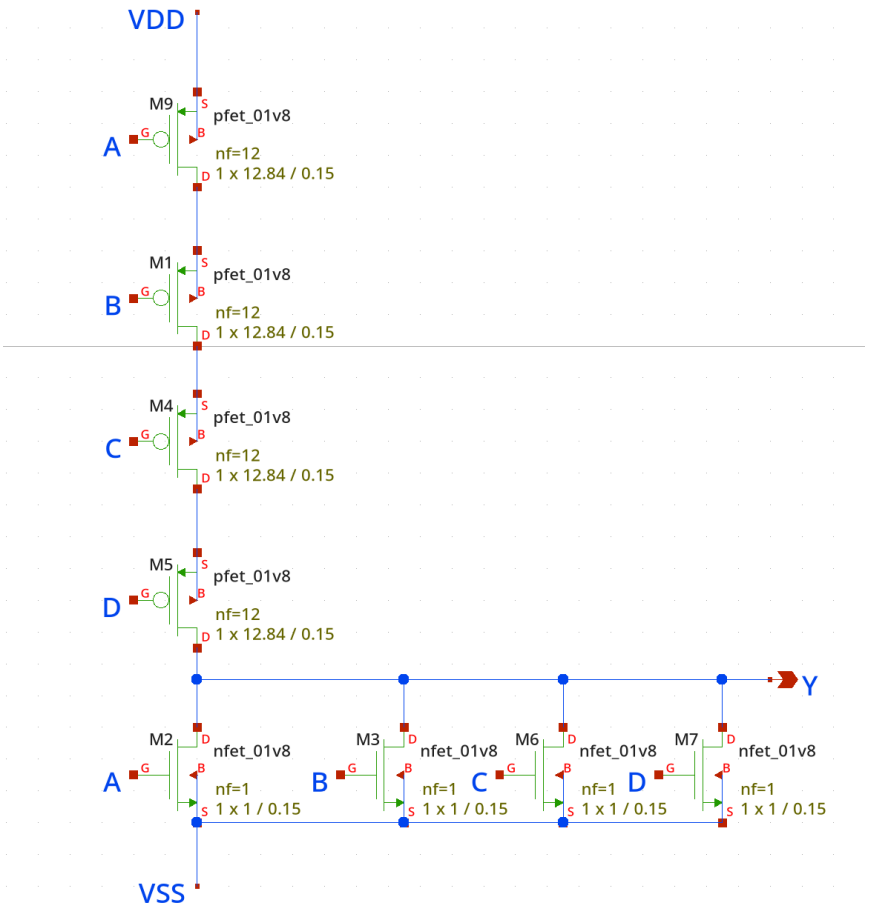


Figure 7 NOR4 Schematic

The NOR4 gate expands the configuration further with four parallel N-FETs in the pull-down network and four series P-FETs in the pull-up network. This increases the challenge of driving the output high due to the significantly higher resistance in the pull-up network. To ensure the gate functions properly, the P-FETs need to be scaled up even more than in NOR2 and NOR3 gates. This makes the cell enormous.

XOR2, XNOR2, 2x1 Mux

It is possible to realize XOR2, XNOR2, and MUX2 circuits using a single design. For this purpose, a subcircuit named CG4 has been created, which serves as the foundation for implementing these circuits.

When designing the CG4 circuit, PMOS and NMOS transistors were intended to share diffusion regions. Diffusion sharing significantly reduces parasitic capacitances by almost half, and this effect was taken into account during the delay calculations using the Elmore delay formula. Rise and fall delays were calculated separately, and new transistor dimensions were derived by attempting to equalize these delays. While determining the dimensions of the PMOS transistors, proportional scaling relative to the NMOS transistor dimensions was applied. For the NMOS transistors, the design ensured that the pull-down network had the same resistance as the pull-down network of a previously designed unit inverter circuit. Additionally, since the design plans to utilize multi-finger structures, this was also considered during the sizing process.

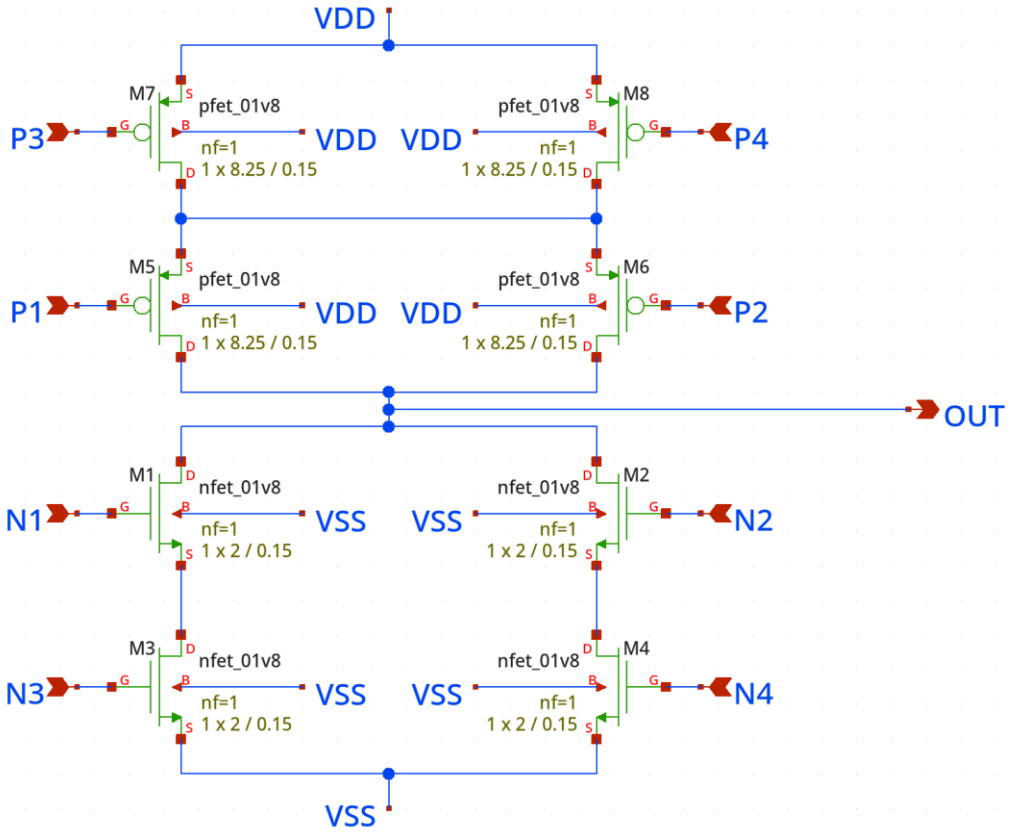
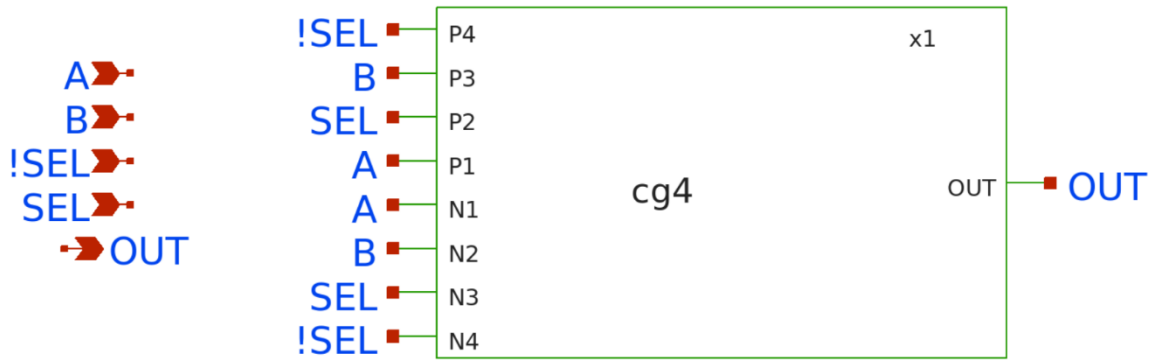
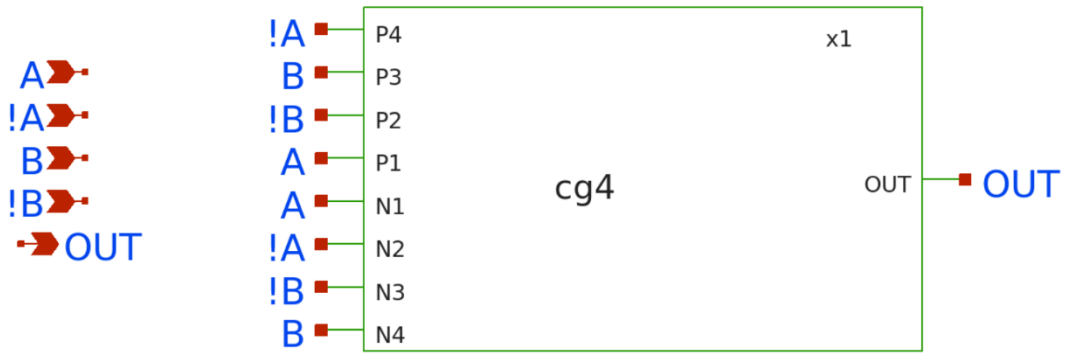


Figure 8 Schematic of XOR2, XNOR2 and 2x1 Mux.

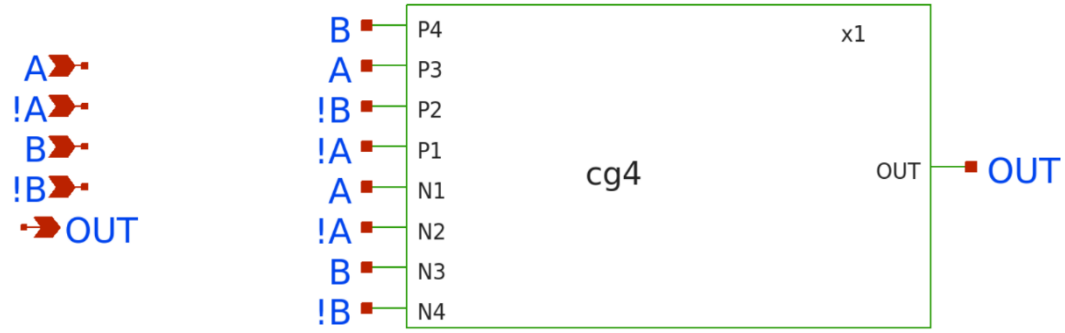
In the symbol they are connected as such to make different gates.



MUX Connections



XNOR connections



XOR connectons

Delay Calculations

The included figures illustrate the rise and fall propagation delay results for the INV, NAND (NAND2, NAND3, NAND4), NOR (NOR2, NOR3, NOR4), XOR2, XNOR2, and 2x1 MUX gates designed using the SKYWATER130nm technology. These delays were calculated based on the Elmore Delay Model, which utilizes the resistance and capacitance values of circuit elements to analytically determine propagation delays. In the designs, resistances were equalized to optimize the performance of each gate. The propagation delays were analyzed for both rising and falling signal transitions, ensuring that all gates meet the specified delay constraints.

ESTIMATED PROPAGATION DELAY CALCULATIONS:

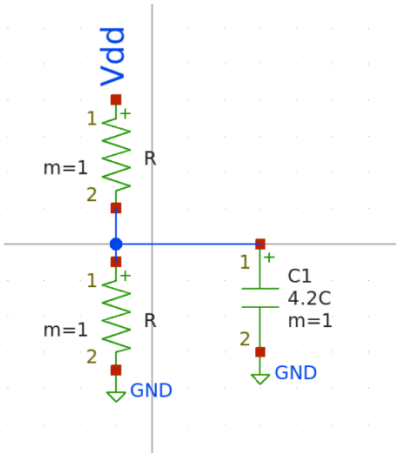


Figure 9 Inverter delay

$$t_p = \frac{4.2RC + 4.2RC}{2} = 4.2RC$$

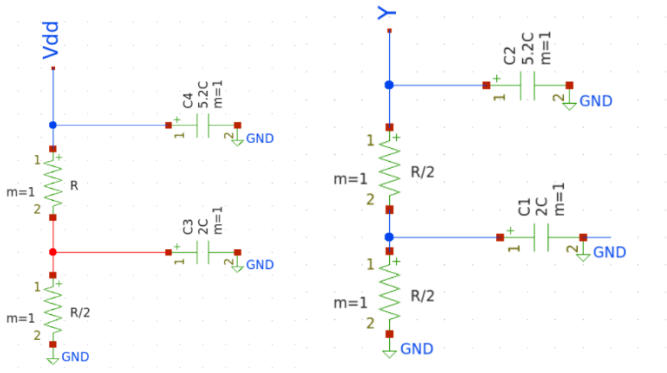


Figure 10 NAND2 rise and fall delay

$$t_{pdf} = \frac{R}{2} \cdot 2C + R \cdot 5.2C = 6.2\,RC$$

$$t_p = \frac{6.2RC + 5.2RC}{2} = \frac{11.4RC}{2} = 5.7RC$$

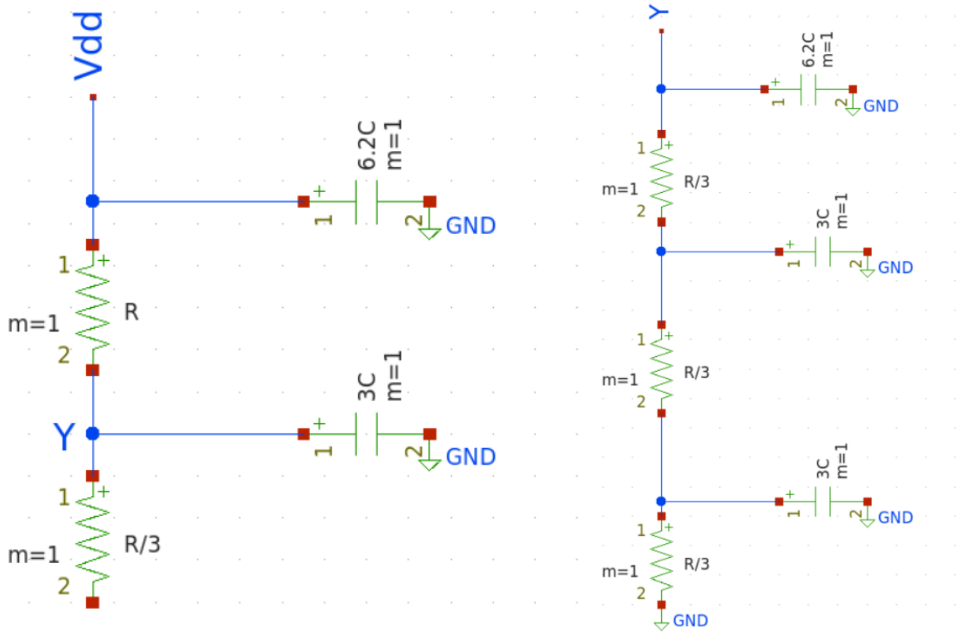


Figure 11 NAND3 rise and fall delay

$$t_{pdf} = \frac{R \cdot 3C}{3} + \frac{2R \cdot 3C}{3} + R \cdot 6 \cdot 2C = 9.2\,RC \qquad t_{pd} = \frac{6.2RC + 9.2RC}{2} = 7.7RC$$

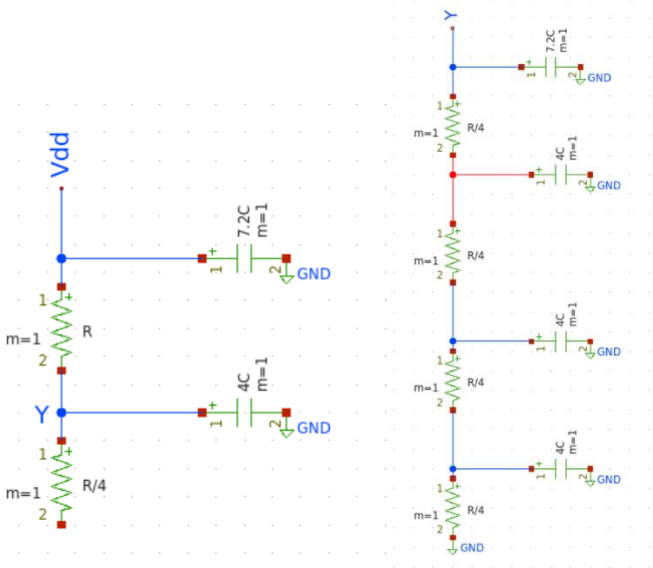


Figure 12 NAND4 rise and fall delay

$$t_{\text{pdf}} = \frac{R}{2} \cdot 4C + \frac{R}{2} \cdot 4C + \frac{3R}{4} \cdot 4C + R \cdot 7.2C = 13.2\,RC$$

$$t_{pd} = \frac{13.2RC + 7.2RC}{2} = \frac{20.4RC}{2} = 10.2RC$$

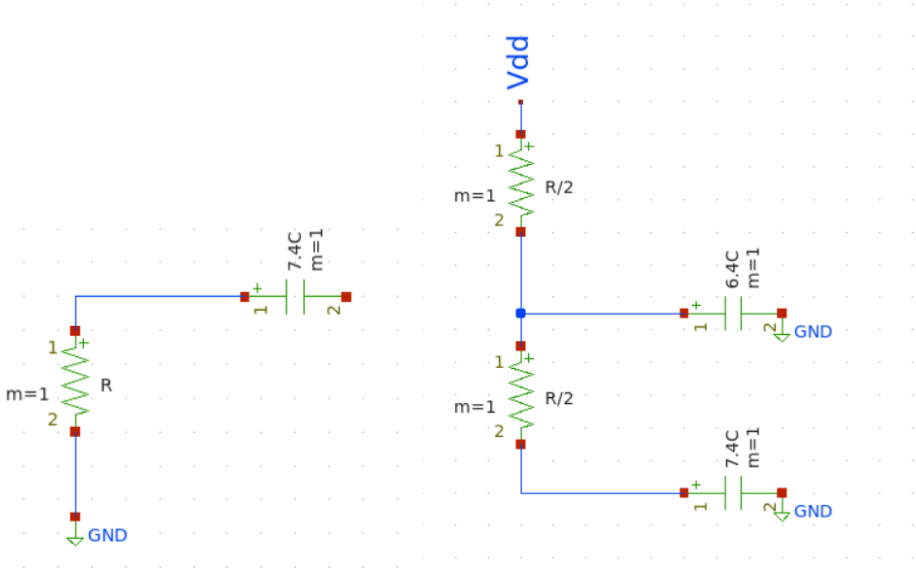


Figure 13 NOR2 rise and fall delay

$$t_{pd} = \frac{t_{pdf} + t_{pdr}}{2} = \frac{7.4RC + 10.6RC}{2} = \frac{18RC}{2} = 9RC$$

$$t_{pd} = 9RC$$

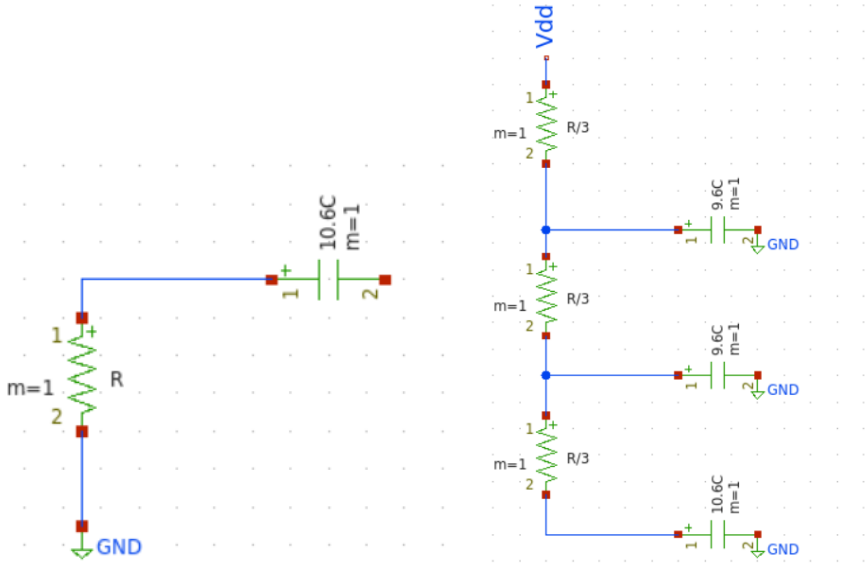


Figure 14 NOR3 rise and fall delay

$$t_{\text{pdf}} = 10.6\,RC$$

$$t_{pdr} = \frac{R}{3} \cdot 9.6c + \frac{2R}{3} \cdot 9.6c + R \cdot 10.6c$$

$$t_{pdr} = 20.2Rc$$

$$t_{pd} = \frac{10.6RC + 20.2RC}{2} = 15.4RC$$

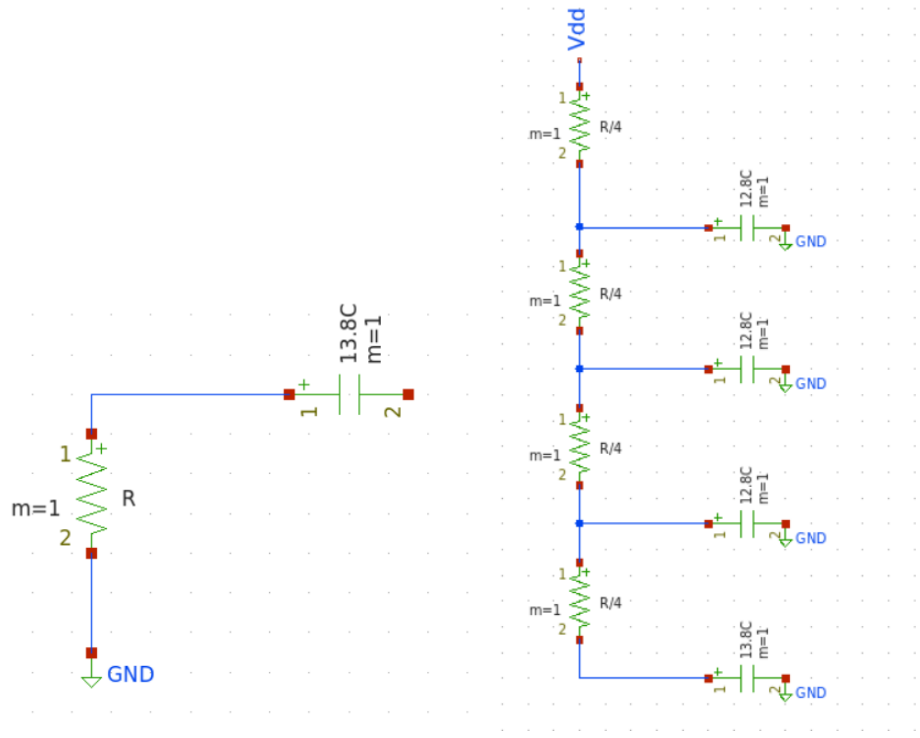


Figure 15 NOR4 rise and fall delay

$$tpdr = \frac{R}{4}.(12.8).C + \frac{R}{2},(12.8).C + \frac{3R}{4}.(12.8).C + R.(13.8).C = 33\,RC$$

$$tpdf = 13.8\,RC$$

$$tpd = 46.8\,/2\,RC$$

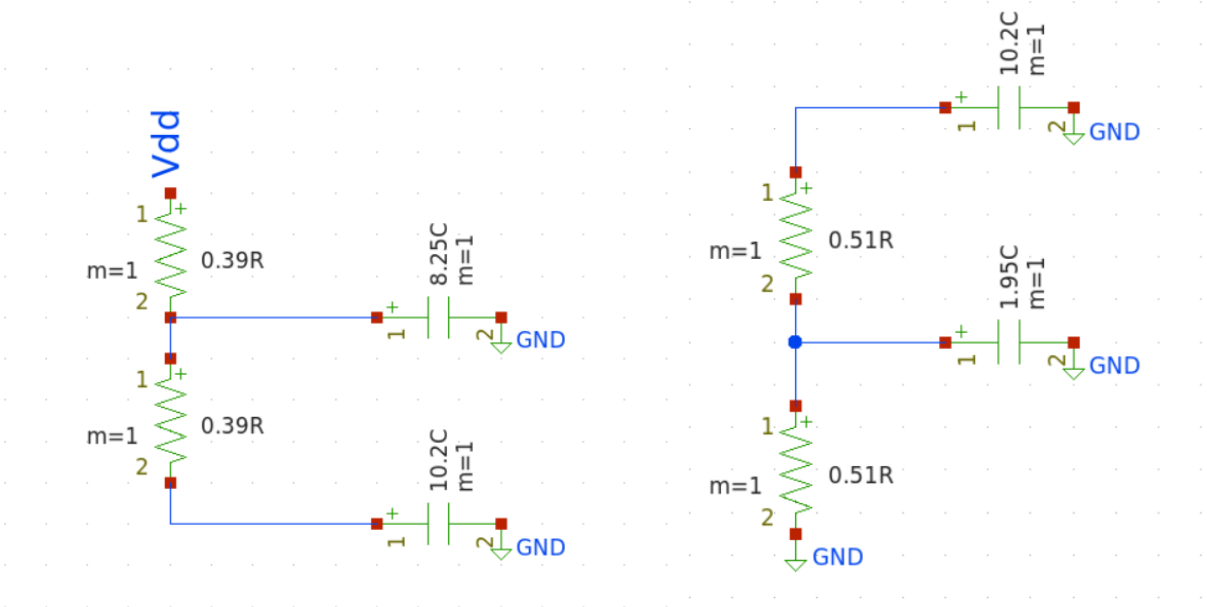


Figure 16 XOR2 XNOR2 MUX2 rise and fall delay

$$T_{pdr} = \left(\frac{3.21}{8.25}\right) \cdot 8.25 \cdot RC + \left(\frac{2 \cdot 3.21}{8.25}\right) \cdot 10.2 \cdot RC \qquad T_{pdf} = \left(\frac{1}{1.95}\right) \cdot 1.95 \cdot RC + \left(\frac{2 \cdot 1}{1.95}\right) \cdot 10.2 \cdot RC$$

$$T_{pdr} = 11.1475 \cdot RC \qquad T_{pdf} = 11.4615 \cdot RC$$

$$T_{pd} = \frac{T_{pdr} + T_{pdf}}{2}$$

$$T_{pd} = 11.3045 \cdot RC$$

LOGICAL EFFORT CALCULATIONS:

The logical effort (g) of a gate is a measure of its delay relative to an inverter. It quantifies how effectively the gate can drive a load compared to a standard inverter, considering the gate's topology and transistor sizing. The calculation involves analyzing the gate's electrical properties, including its input capacitance and output drive strength.

For this analysis, the logical effort for various gates, including NAND, NOR, and XOR, was determined based on their respective configurations and the number of inputs. Each gate's logical effort was computed relative to an inverter to provide a standardized comparison. These values are critical in understanding the gates' performance, optimizing circuit design, and minimizing overall propagation delay in digital circuits.

Logic Gate	**Calculated Logical Effort**
Inverter	1
NAND2-3-4	1.26, 1.48, 1.71
NOR2-3-4	1.76, 2.52, 3.29
XOR2-NOR2-2X1MUX	2, 2, 2

Table 1 Logical Efforts

Input capacitance values for each of gate:

NAND2	NAND3	NAND4	NOR2	NOR3	NOR4
$C_{in} = 5.4$	$C_{in} = 6.2$	$C_{in} = 7.2$	$C_{in} = 7.4$	$C_{in} = 10.6$	$C_{in} = 13.8$
INVERTER	XOR2				
$C_{in} = 4.2$	$C_{in} = 8.4$				

In order to calculate logical effor input capacitances are divided to inverters input capacitance value.And values in Table1. are obtained.

Question 2

Is the inverter propagation delay minimum? Please remember the design for average delay concept discussed in the class and re-size the inverter (Hint: only consider the average delay).

The inverter propagation delay is not minimum. Equalizing the resistances of MOSFETs ensures that the rise and fall delays are balanced. However, this does not necessarily result in the minimum propagation delay. By taking the square root of ratio of sizing we can obtain the best delay.

$$tpd\_best = \sqrt{\frac{p}{n}}$$

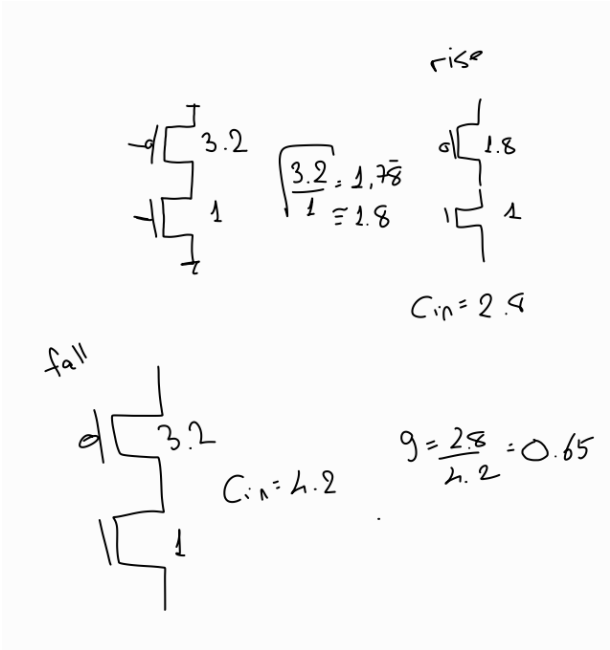


Figure 17 Best delay calculations

For calculating both rise and fall delays, while the fall delay worsened significantly, the rise delay improved. Additionally, when logical efforts were considered, they decreased significantly, dropping from 1 to 0.65. This method gave the best propagation delay.

Question 3

Design the NAND2 gates using Psedo- and Dynamic- Logic approaches and compares the results in terms of area, power, and delay.

To determine the size of the PMOS, Elmore delay equations for both rise and fall delays were equalized, yielding a size ratio of x=3.67 and x = 2.63. When comparing the areas, it can be observed that the overall area becomes smaller. Consequently, the logical effort is reduced, leading to a smaller propagation delay.

Pseudo logic, on the other hand, typically consumes more power due to dynamic power dissipation and leakage through partially conducting paths. Although the design is simpler than static CMOS, it suffers from higher static power dissipation because the PMOS transistor continuously draws current when the output is low.

In conclusion, pseudo-logic is more power-efficient in static conditions but suffers from higher static power dissipation, whereas dynamic-logic provides faster operation at the cost of higher dynamic power consumption. The comparision between these two approaches depends on the the acceptable power consumption, speed, and area constraints.

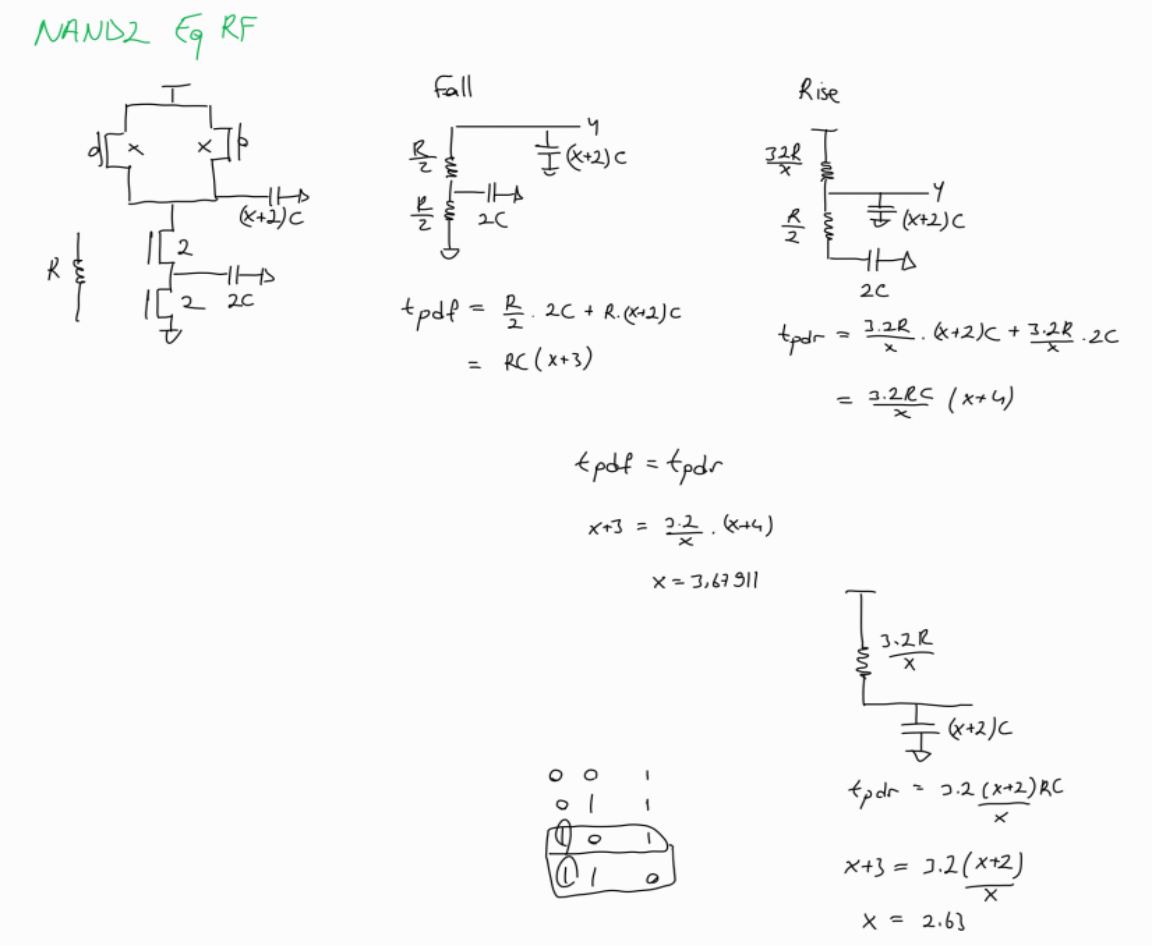


Figure 18 NAND2 Question 3 Calculations

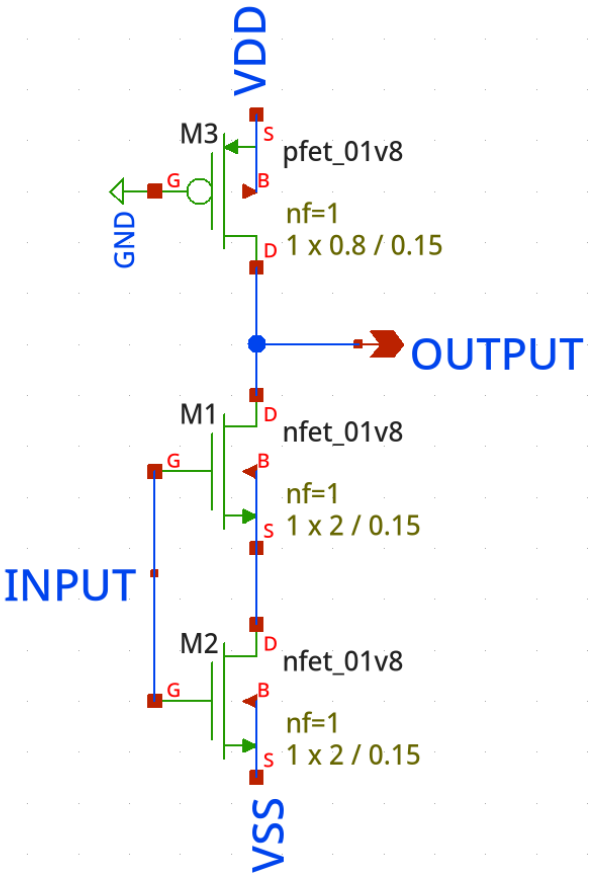
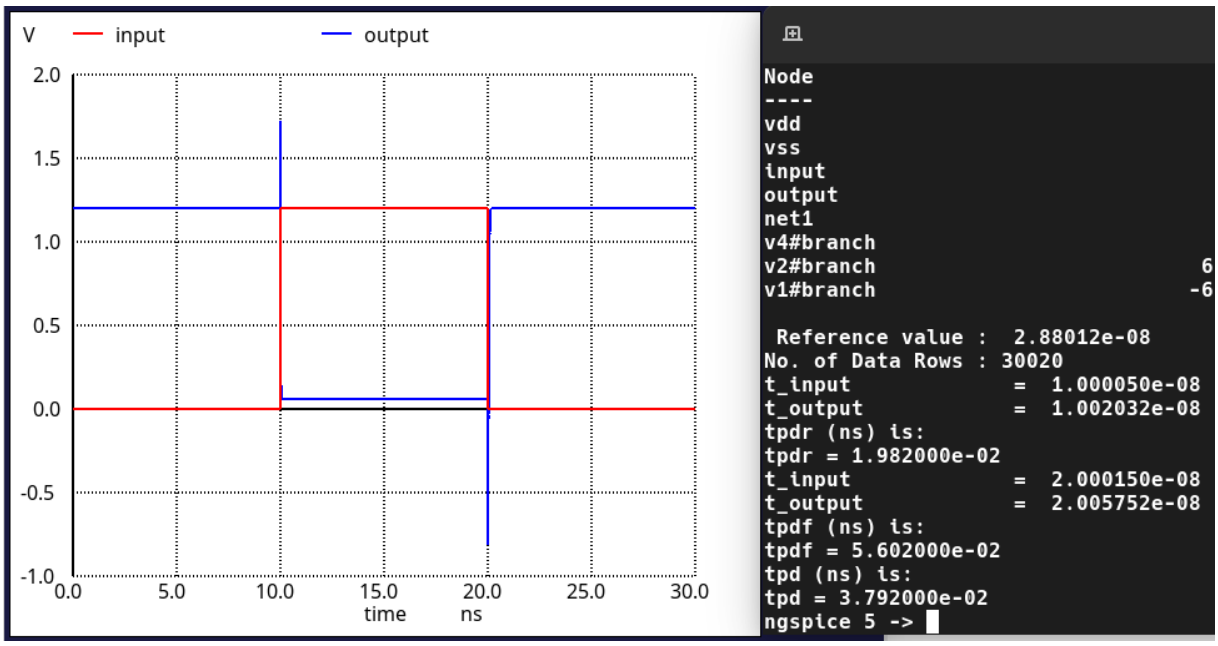


Figure 19 Pseudo Pmos Schematic

These calculatios were also validated through xschem. As can be seen, the output never reaches zero.



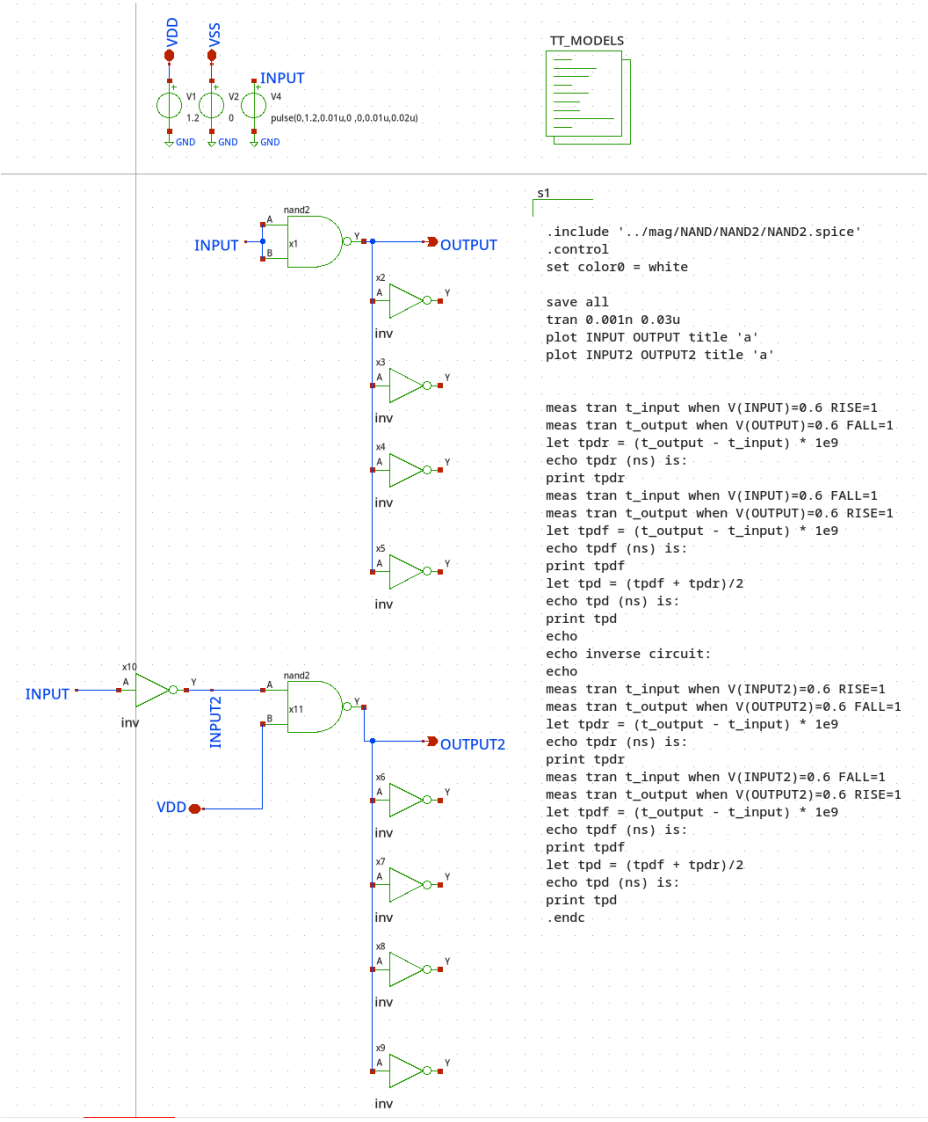
Question 4

Draw the layout of the circuits in 2 and tabulate the pre-and post-layout simulation results within a table. Also, give the variations in delay due to parasitic effects. Which gate has the maximum relative variation in delay? Is this expected?

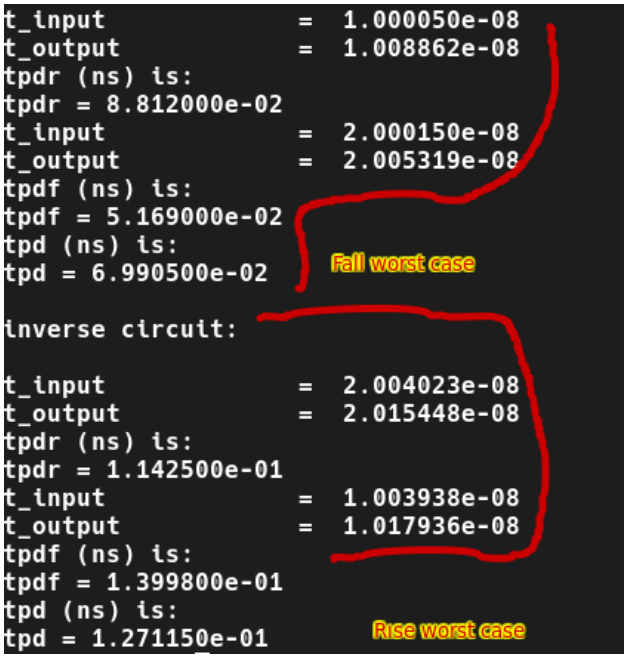
The layouts were drawn in magic VLSI. To test these circuits, a standard test bench was used. This allowed us to compare our results more easily between the gates. This is Shown in the image.

Since we were asked to calculate the FO4 delay of these cells, 4 parallel inverters were connected to the output of the Cell-Under-Test. This load is a standard way of testing logic cells.

The worst-case scenario for each cell was simulated. The worst cases are going to be explained in detail in the relevant sections.



How to read the terminal output & Information about the testbench



The output of the terminal window for the simulations are shown in the figure above. In the test benches, 2 types of inputs are tested. In one scenario, all the inputs are given the same signal, in the other case, the signal is fed through just one input pin. This simulates the different ways that the input can be received.

Example: Nor2 Spice File.

Due to the incredible number of circuits in this report, not every spice file can be shown. Here is an example of the nor2 spice file extracted from the layout.

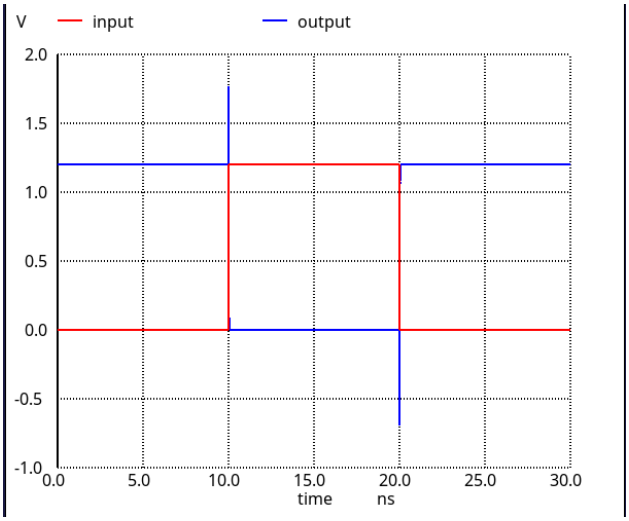
```
* NGSPICE file created from nor2_pex.ext - technology: sky130B

.subckt nor2 A B VSS VDD Y
X0 a_154_100.t1 B.t0 VDD.t9 VDD.t8 sky130_fd_pr__pfet_01v8 ad=0.3103 pd=2.43 as=0.6206 ps=4.86 w=2.14 l=0.3
X1 a_1098_100.t1 A.t0 Y.t3 VDD.t13 sky130_fd_pr__pfet_01v8 ad=0.3103 pd=2.43 as=0.3103 ps=2.43 w=2.14 l=0.3
X2 Y.t1 A.t1 a_862_100.t1 VDD.t14 sky130_fd_pr__pfet_01v8 ad=0.3103 pd=2.43 as=0.3103 ps=2.43 w=2.14 l=0.3
X3 a_272_100.t1 B.t1 a_626_100.t1 VDD.t7 sky130_fd_pr__pfet_01v8 ad=0.3103 pd=2.43 as=0.3103 ps=2.43 w=2.14 l=0.3
X4 a_862_100.t0 A.t2 a_272_100.t5 VDD.t12 sky130_fd_pr__pfet_01v8 ad=0.3103 pd=2.43 as=0.3103 ps=2.43 w=2.14 l=0.3
X5 a_626_100.t0 B.t2 VDD.t6 VDD.t5 sky130_fd_pr__pfet_01v8 ad=0.3103 pd=2.43 as=0.3103 ps=2.43 w=2.14 l=0.3
X6 VDD.t4 B.t3 a_390_100.t0 VDD.t3 sky130_fd_pr__pfet_01v8 ad=0.3103 pd=2.43 as=0.3103 ps=2.43 w=2.14 l=0.3
X7 Y.t4 A.t3 a_1334_100.t1 VDD.t11 sky130_fd_pr__pfet_01v8 ad=0.6206 pd=4.86 as=0.3103 ps=2.43 w=2.14 l=0.3
X8 VSS.t3 A.t4 Y.t2 VSS.t2 sky130_fd_pr__nfet_01v8 ad=0.58 pd=4.58 as=0.29 ps=2.29 w=2 l=0.3
X9 a_390_100.t1 B.t4 a_272_100.t2 VDD.t2 sky130_fd_pr__pfet_01v8 ad=0.3103 pd=2.43 as=0.3103 ps=2.43 w=2.14 l=0.3
X10 Y.t0 B.t5 VSS.t1 VSS.t0 sky130_fd_pr__nfet_01v8 ad=0.29 pd=2.29 as=0.58 ps=4.58 w=2 l=0.3
X11 a_1334_100.t0 A.t5 a_272_100.t4 VDD.t0 sky130_fd_pr__pfet_01v8 ad=0.3103 pd=2.43 as=0.3103 ps=2.43 w=2.14 l=0.3
X12 a_272_100.t0 B.t6 a_154_100.t0 VDD.t1 sky130_fd_pr__pfet_01v8 ad=0.3103 pd=2.43 as=0.3103 ps=2.43 w=2.14 l=0.3
X13 a_272_100.t3 A.t6 a_1098_100.t0 VDD.t10 sky130_fd_pr__pfet_01v8 ad=0.3103 pd=2.43 as=0.3103 ps=2.43 w=2.14 l=0.3
R0 B.n5 B.t5 535.899
R1 B.n2 B.t1 417.341
R2 B.n0 B.t0 417.341
R3 B.n3 B.t3 372.918
R4 B.n2 B.t2 372.918
R5 B.n0 B.t6 372.918
R6 B.n1 B.t4 372.918
R7 B B.n4 113.737
R8 B.n1 B.n0 44.424
R9 B.n3 B.n2 44.424
R10 B.n4 B.n3 16.1887
R11 B.n4 B.n1 15.0593
R12 B.n5 B 6.02403
R13 B B.n5 5.48621
R14 B.n5 B 4.75479
R15 VDD.n2 VDD.t9 158.858
R16 VDD.n2 VDD.n1 145.48
R17 VDD.t0 VDD.t11 110.01
R18 VDD.t10 VDD.t0 110.01
```

```
R19 VDD.t13 VDD.t10 110.01
R20 VDD.t14 VDD.t13 110.01
R21 VDD.t12 VDD.t14 110.01
R22 VDD.t7 VDD.t12 110.01
R23 VDD.t5 VDD.t7 110.01
R24 VDD.t3 VDD.t5 110.01
R25 VDD.t2 VDD.t3 110.01
R26 VDD.t1 VDD.t8 110.01
R27 VDD.n0 VDD.t1 78.3126
R28 VDD VDD.n0 47.4658
R29 VDD.n0 VDD.t2 31.6982
R30 VDD.n1 VDD.t6 13.3486
R31 VDD.n1 VDD.t4 13.3486
R32 VDD VDD.n2 0.0655794
R33 a_154_100.t0 a_154_100.t1 26.6968
R34 A.n0 A.t3 417.341
R35 A A.t4 384.558
R36 A.n4 A.t2 372.918
R37 A.n3 A.t1 372.918
R38 A.n2 A.t0 372.918
R39 A.n1 A.t6 372.918
R40 A.n0 A.t5 372.918
R41 A.n5 A.n4 65.5064
R42 A.n1 A.n0 44.424
R43 A.n2 A.n1 44.424
R44 A.n3 A.n2 44.424
R45 A.n4 A.n3 44.424
R46 A.n5 A 4.89462
R47 A A.n5 1.88285
R48 Y.n1 Y.t4 158.811
R49 Y.n1 Y.n0 145.726
R50 Y Y.n2 83.3758
R51 Y.n0 Y.t3 13.3486
R52 Y.n0 Y.t1 13.3486
R53 Y.n2 Y.t2 8.7005
R54 Y.n2 Y.t0 8.7005
R55 Y Y.n1 0.43199
R56 a_1098_100.t0 a_1098_100.t1 26.6968
R57 a_862_100.t0 a_862_100.t1 26.6968
R58 a_626_100.t0 a_626_100.t1 26.6968
R59 a_272_100.n3 a_272_100.n2 146.124
R60 a_272_100.n2 a_272_100.n0 146.123
R61 a_272_100.n2 a_272_100.n1 145.821
R62 a_272_100.n1 a_272_100.t5 13.3486
R63 a_272_100.n1 a_272_100.t1 13.3486
R64 a_272_100.n0 a_272_100.t4 13.3486
R65 a_272_100.n0 a_272_100.t3 13.3486
R66 a_272_100.n3 a_272_100.t2 13.3486
R67 a_272_100.t0 a_272_100.n3 13.3486
R68 a_390_100.t0 a_390_100.t1 26.6968
R69 a_1334_100.t0 a_1334_100.t1 26.6968
R70 VSS VSS.t0 974.856
R71 VSS.t0 VSS.t2 323.49
R72 VSS.n0 VSS.t3 41.6266
R73 VSS.n0 VSS.t1 41.3938
R74 VSS VSS.n0 0.023
C0 Y A 0.204858f
C1 B A 0.159179f
C2 VDD A 0.049461f
C3 Y B 0.019228f
C4 Y VDD 0.113724f
C5 B VDD 0.249656f
C6 Y VSS 1.05355f
C7 A VSS 0.788605f
C8 B VSS 0.907937f
C9 VDD VSS 5.18777f
.ends
```

Inverter

Pre-Layout

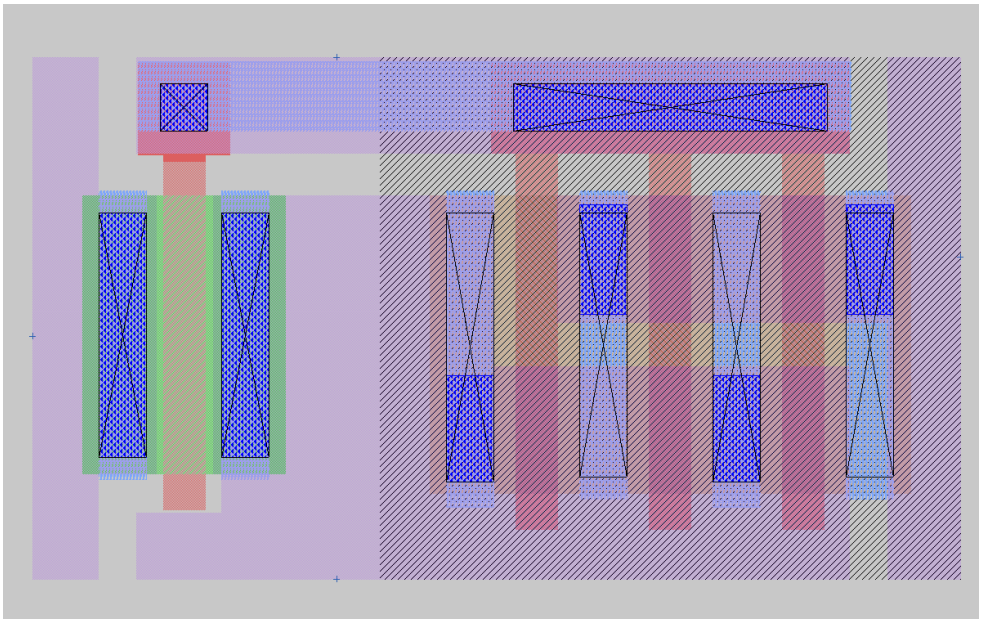
The prelayout simulations on the circuit are shown below. The delay of the inverters is found to be 15ps.




```
t_input              =  1.000050e-08
t_output             =  1.001923e-08
tpdr (ns) is:
tpdr = 1.873000e-02
t_input              =  2.000150e-08
t_output             =  2.001664e-08
tpdf (ns) is:
tpdf = 1.514000e-02
tpd (ns) is:
tpd = 1.693500e-02
```

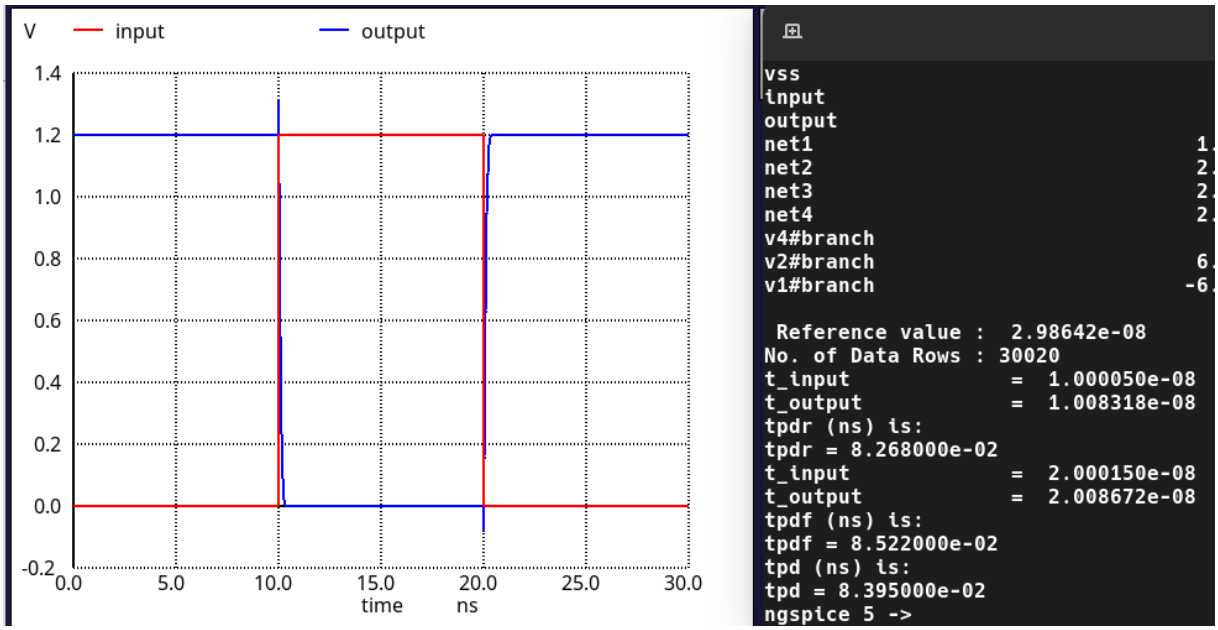
Layout & Post-Layout

The layout of the inverter was drawn in MAGIC VLSI as shown below.



In this layout, there was a focus on compactness and a major focus on standardization. The widths of the pmos and nmos are almost identical. To achieve this, fingers to the pmos were added to reduce its width without changing its effective width.

The simulation of this inverter is shown below.

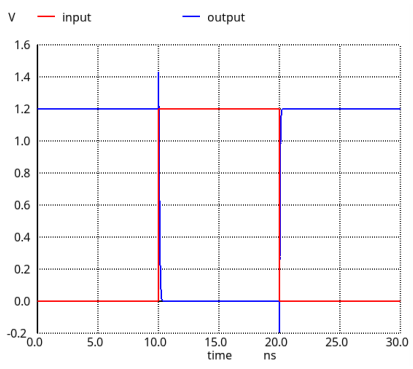


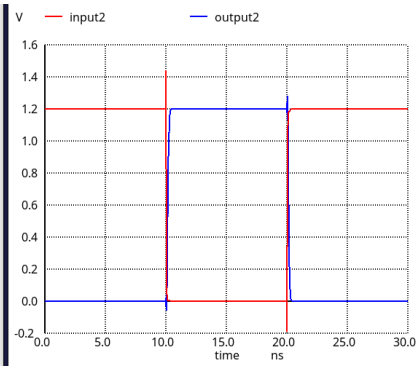
NAND

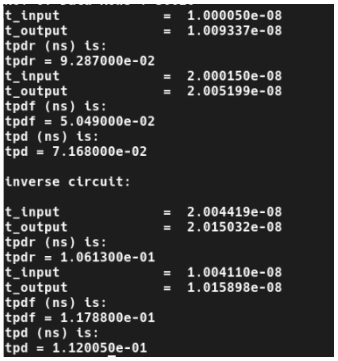
Nand2

Pre-Layout

The prelayout simulations for the Nand2 are shown below. The rise time worst case was found to be 70ps and fall time worst case was found to be ~110ps. These fall below the target values are not near the design goals, but they are sufficient for this homework.

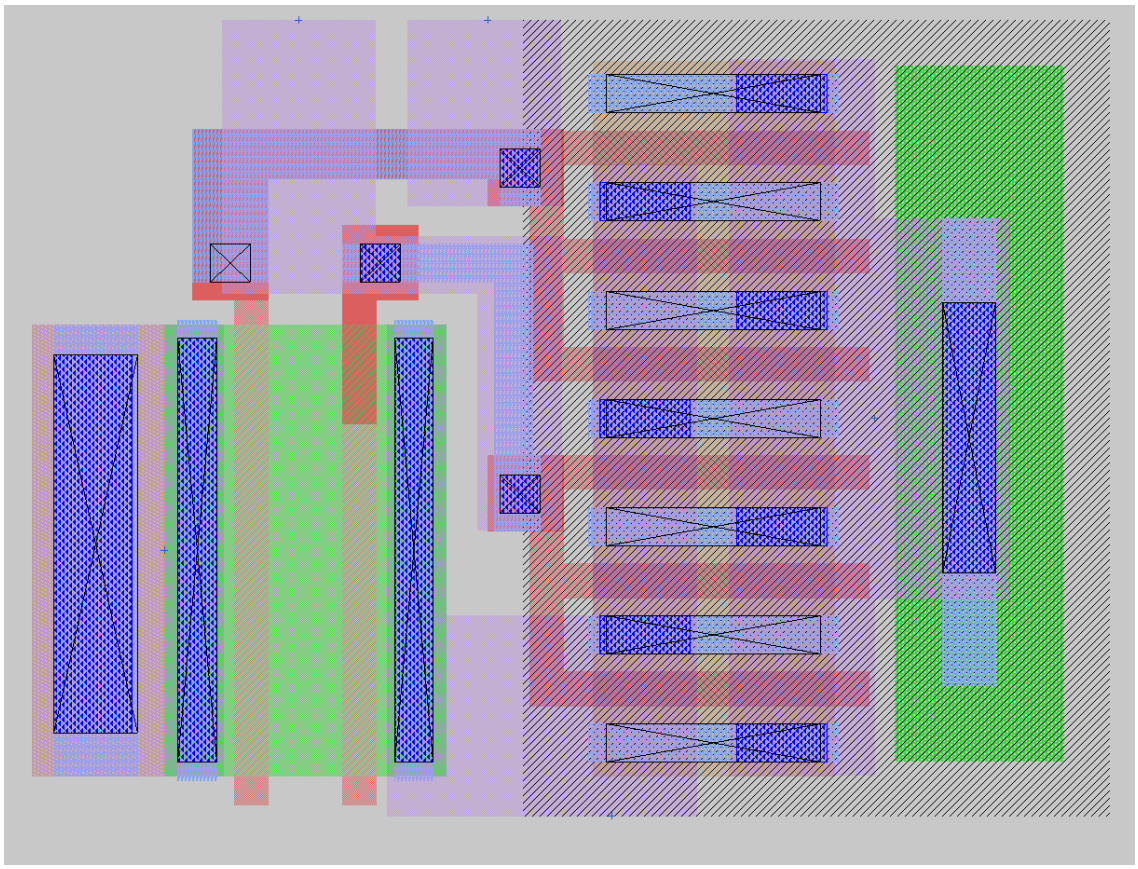


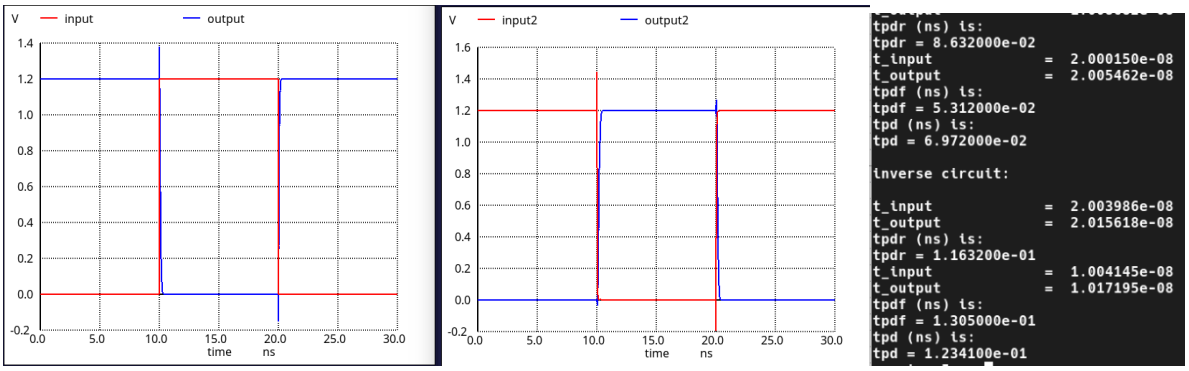




Layout & Post-Layout

The layout of the NAND2 is shown below. In this layout, the size of the same considerations was made for the inverter.

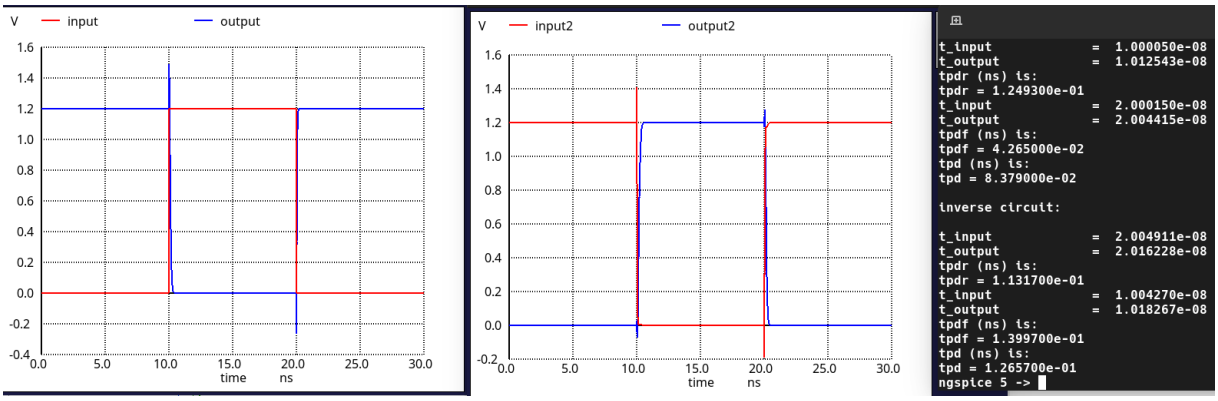




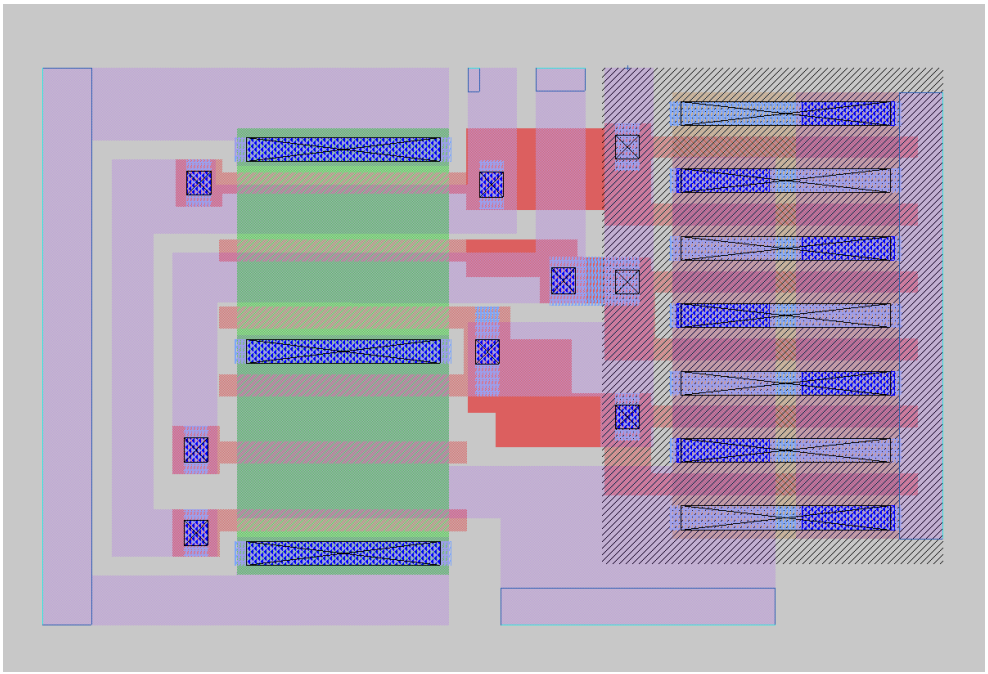
In the layout, the worst cases were a bit better. This is achieved with better input ordering.

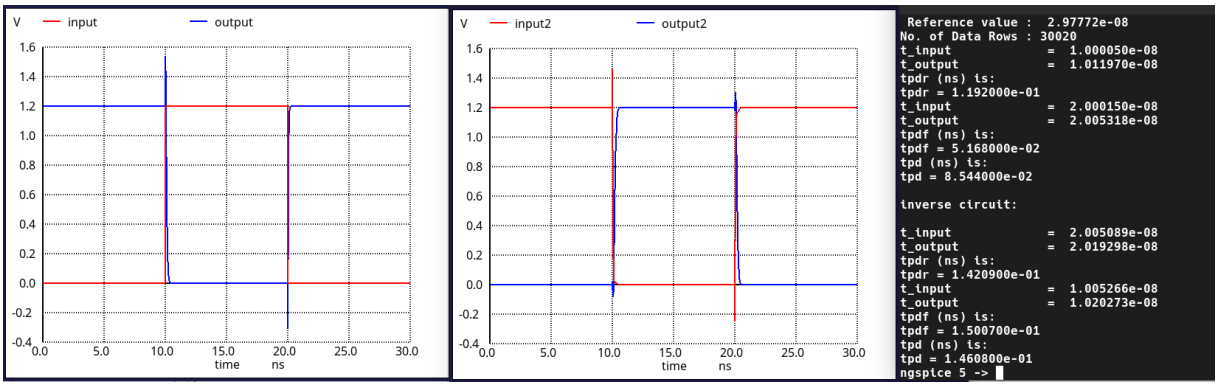
Nand3

Pre-Layout



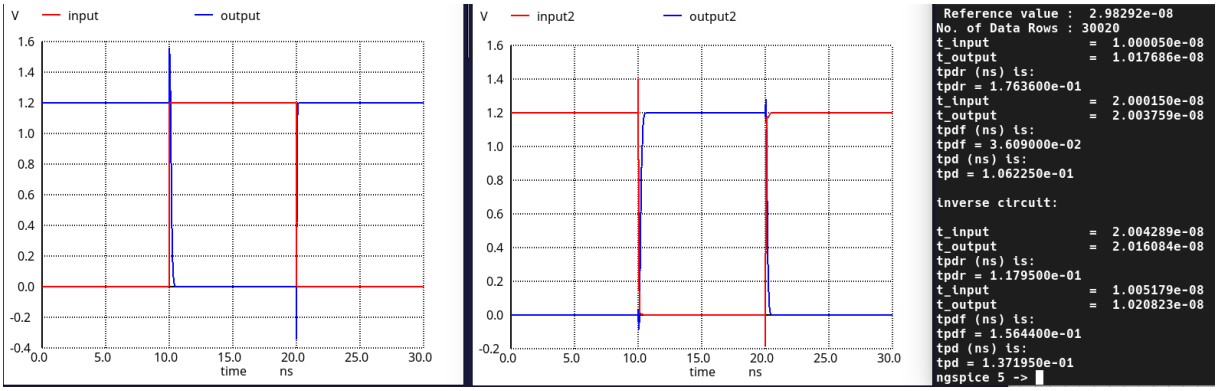
Layout & Post-Layout



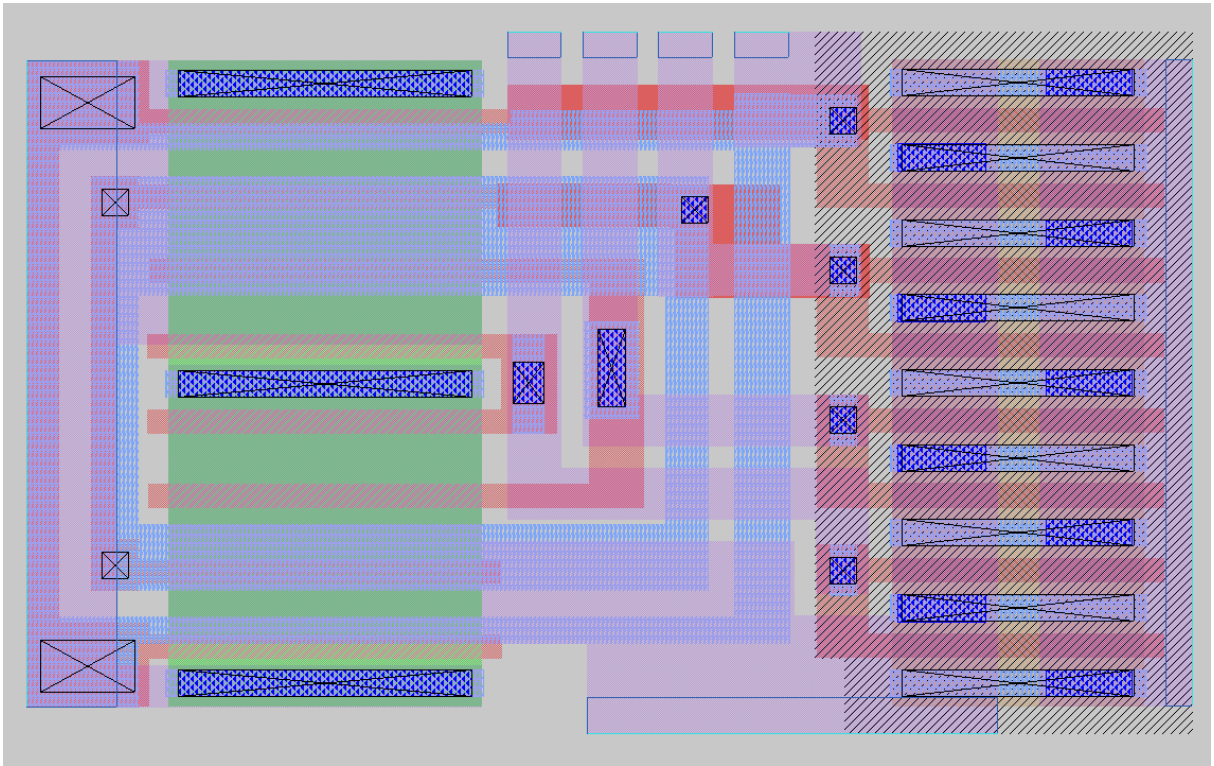


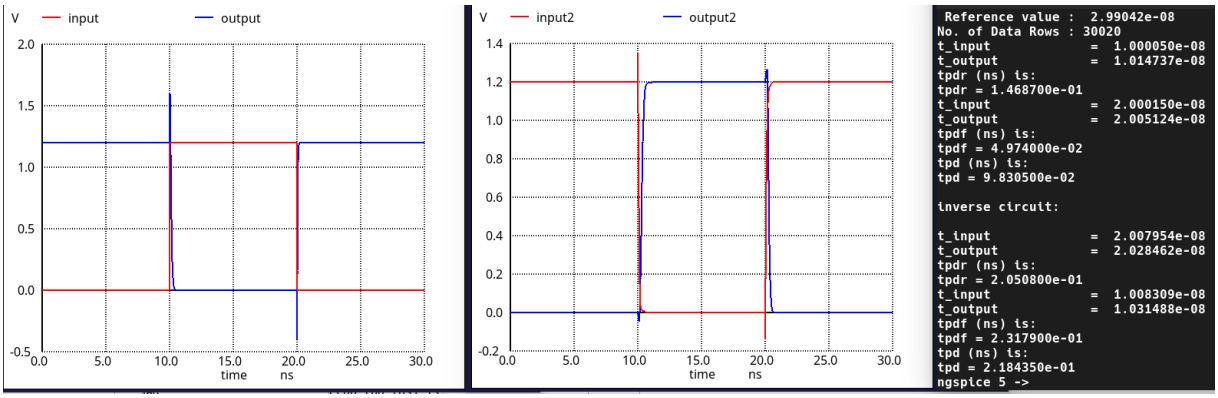
Nand4

Pre-Layout



Layout & Post-Layout

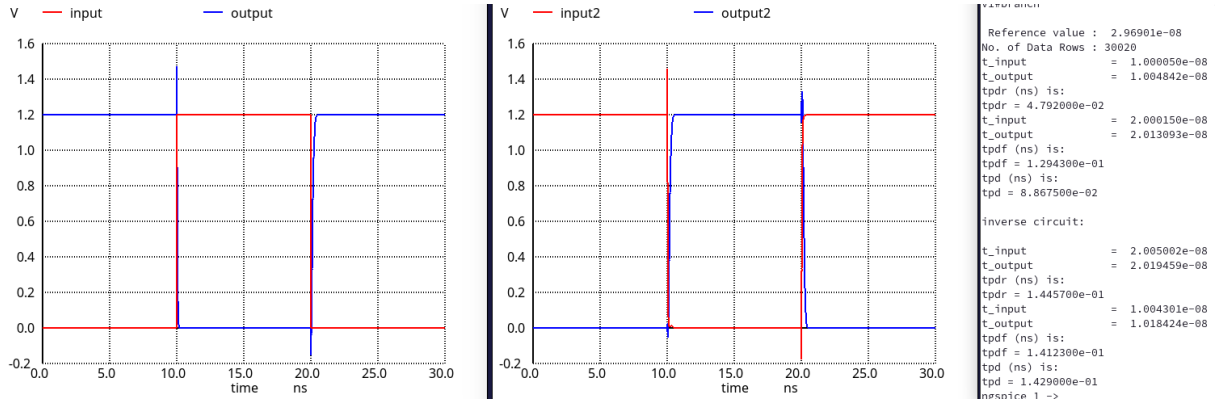




NOR2

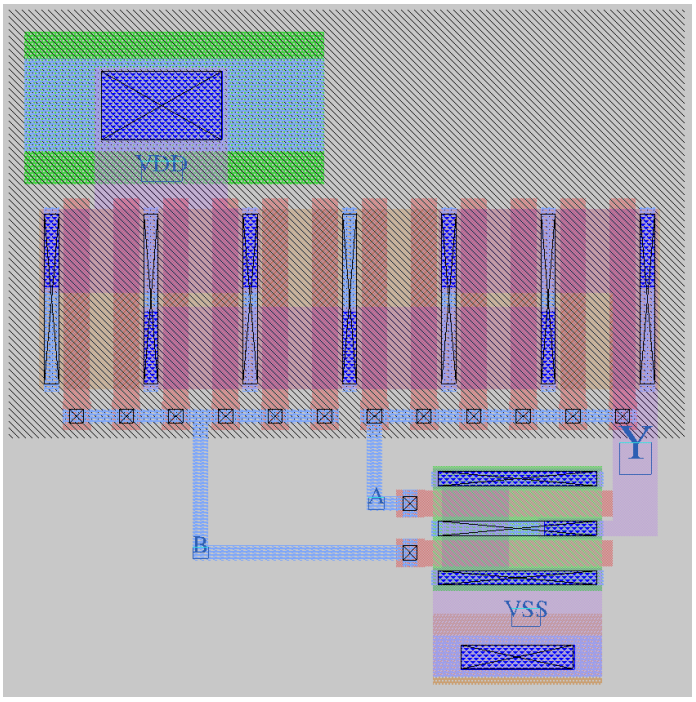
Pre-Layout

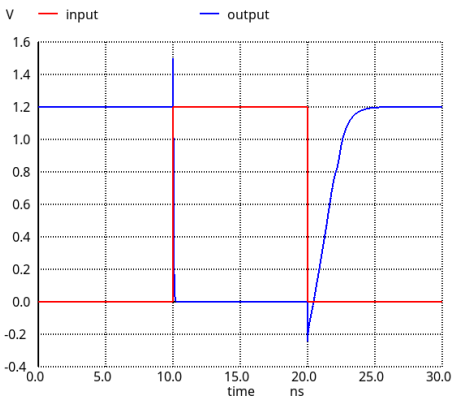
The prelayout simulations for the NOR2 are shown below.

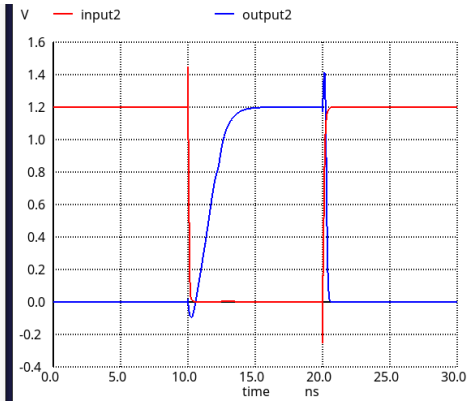


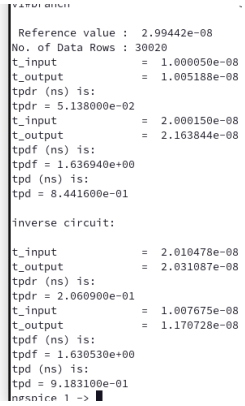
Layout & Post-Layout

The layout of the NOR2 is shown below. In this layout, finger values of the PMOS transistors are adjusted as it was in the schematic.





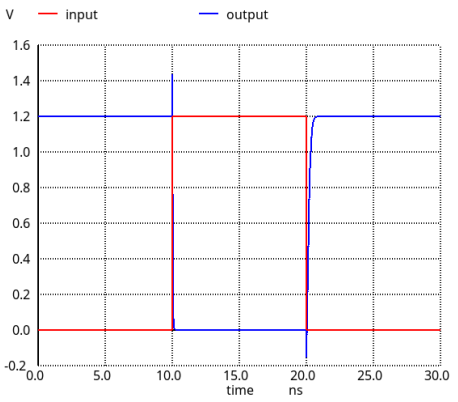


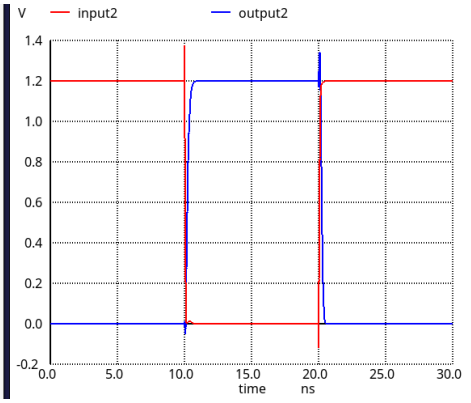


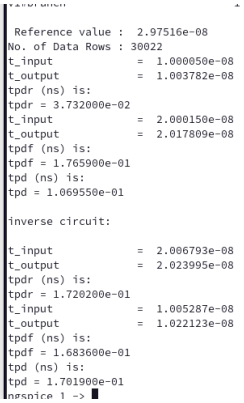
```
 Reference value :  2.99442e-08
No. of Data Rows : 30020
t_input              =  1.000050e-08
t_output             =  1.005188e-08
tpdr (ns) is:
tpdr = 5.138000e-02
t_input              =  2.000150e-08
t_output             =  2.163844e-08
tpdf (ns) is:
tpdf = 1.636940e+00
tpd (ns) is:
tpd = 8.441600e-01

inverse circuit:

t_input              =  2.010478e-08
t_output             =  2.031087e-08
tpdr (ns) is:
tpdr = 2.060900e-01
t_input              =  1.007675e-08
t_output             =  1.170728e-08
tpdf (ns) is:
tpdf = 1.630530e+00
tpd (ns) is:
tpd = 9.183100e-01
ngspice 1 ->
```

NOR3

Pre-Layout

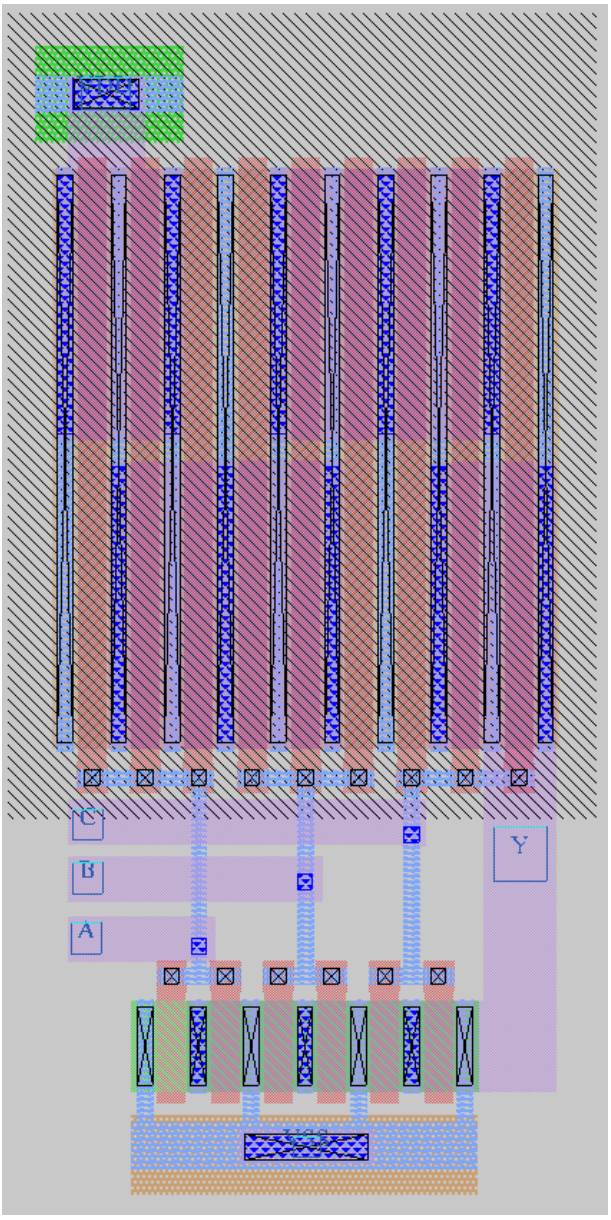


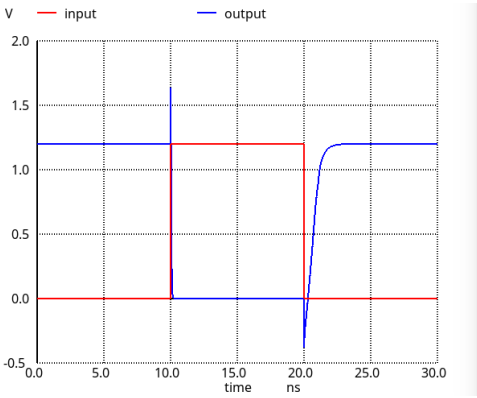


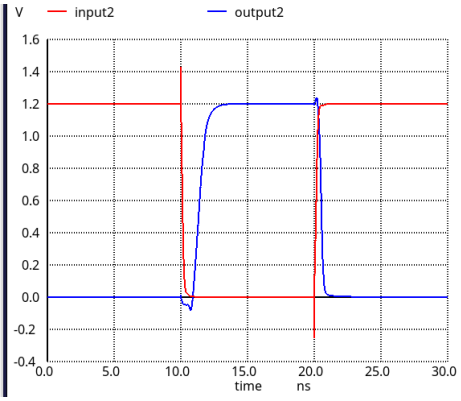


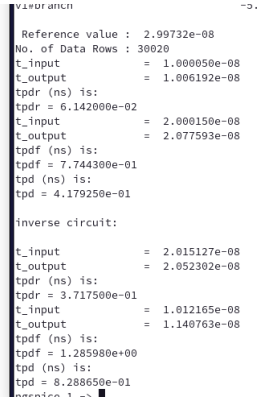
```
 Reference value :  2.97516e-08
No. of Data Rows : 30022
t_input              =  1.000050e-08
t_output             =  1.003782e-08
tpdr (ns) is:
tpdr = 3.732000e-02
t_input              =  2.000150e-08
t_output             =  2.017809e-08
tpdf (ns) is:
tpdf = 1.765900e-01
tpd (ns) is:
tpd = 1.069550e-01

inverse circuit:

t_input              =  2.006793e-08
t_output             =  2.023995e-08
tpdr (ns) is:
tpdr = 1.720200e-01
t_input              =  1.005287e-08
t_output             =  1.022123e-08
tpdf (ns) is:
tpdf = 1.683600e-01
tpd (ns) is:
tpd = 1.701900e-01
ngspice 1 ->
```

Layout & Post-Layout





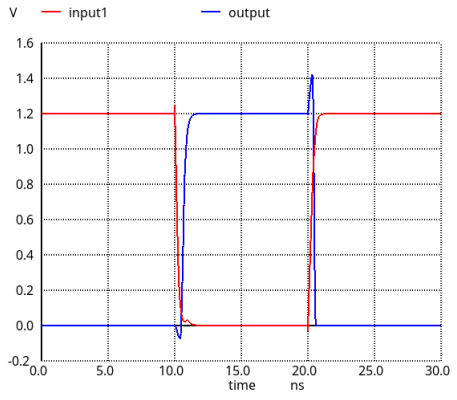


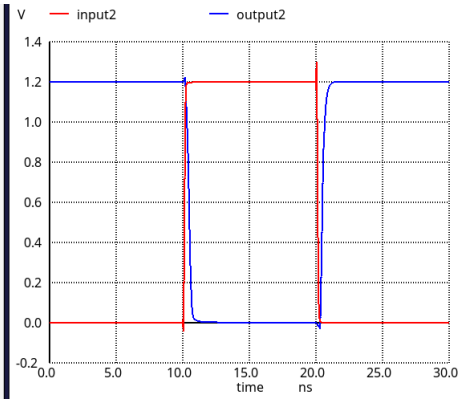


```
 Reference value :  2.99732e-08
No. of Data Rows : 30020
t_input            =  1.000050e-08
t_output           =  1.006192e-08
tpdr (ns) is:
tpdr = 6.142000e-02
t_input            =  2.000150e-08
t_output           =  2.077593e-08
tpdf (ns) is:
tpdf = 7.744300e-01
tpd (ns) is:
tpd = 4.179250e-01

inverse circuit:

t_input            =  2.015127e-08
t_output           =  2.052302e-08
tpdr (ns) is:
tpdr = 3.717500e-01
t_input            =  1.012165e-08
t_output           =  1.140763e-08
tpdf (ns) is:
tpdf = 1.285980e+00
tpd (ns) is:
tpd = 8.288650e-01
```

NOR4

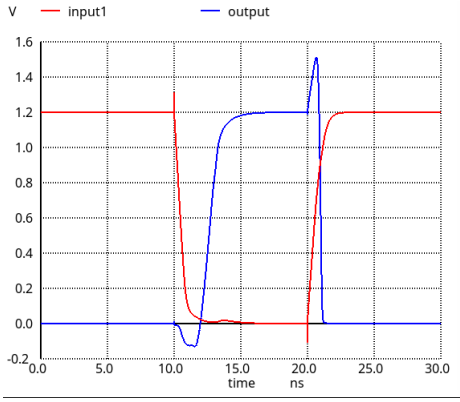
Pre-Layout

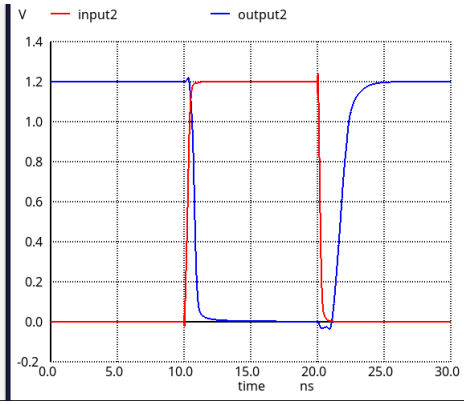




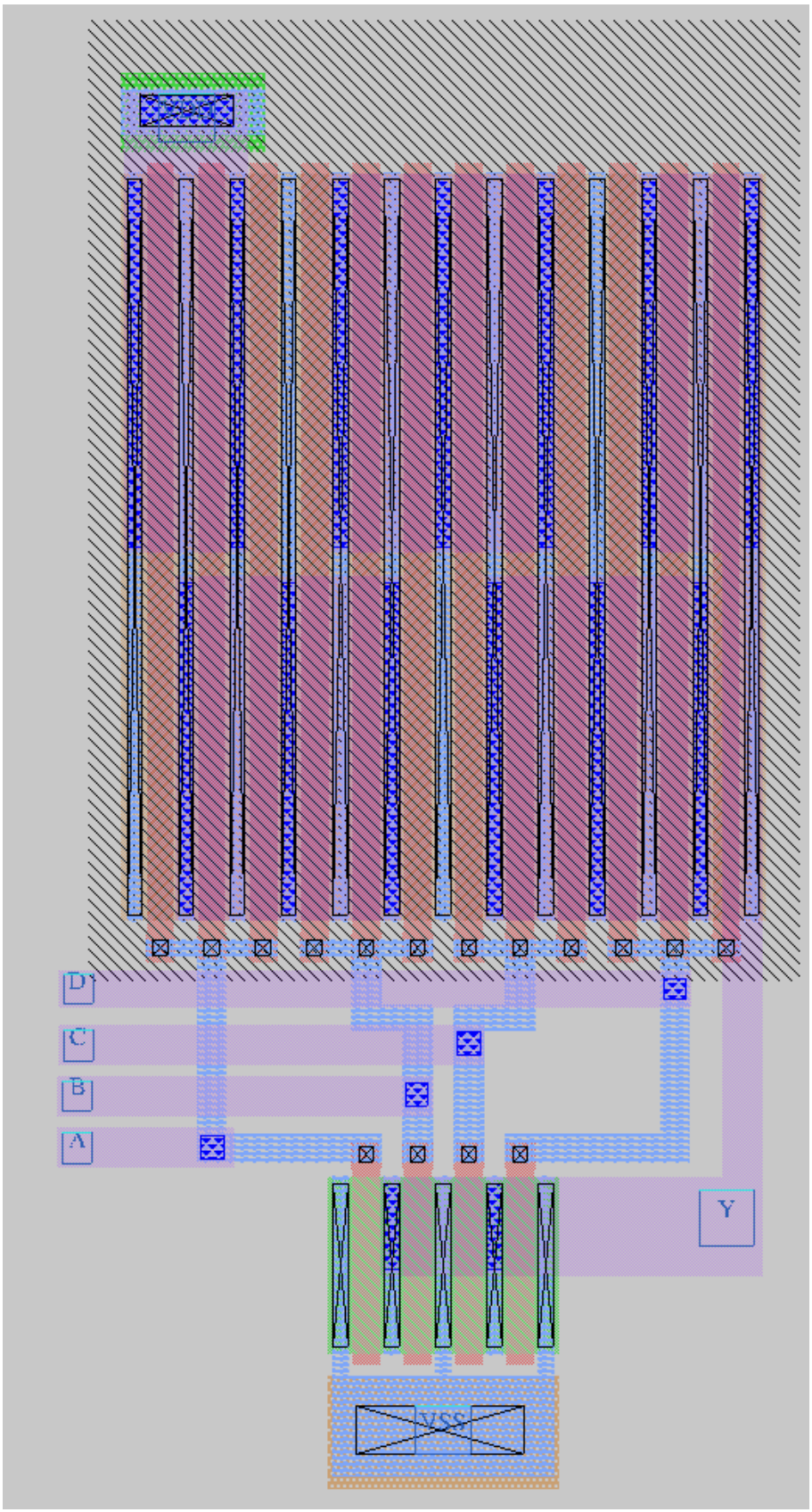

```
v1#branch                      -8.(

 Reference value :  2.95872e-08
No. of Data Rows : 30020
t_input              =  1.000050e-08
t_output             =  2.047937e-08
tpdr (ns) is:
tpdr = 1.047887e+01
t_input              =  2.000150e-08
t_output             =  1.067122e-08
tpdf (ns) is:
tpdf = -9.33028e+00
tpd (ns) is:
tpd = 5.742950e-01

inverse circuit:

t_input              =  1.013687e-08
t_output             =  1.047093e-08
tpdr (ns) is:
tpdr = 3.340600e-01
t_input              =  2.012955e-08
t_output             =  2.048707e-08
tpdf (ns) is:
tpdf = 3.575200e-01
tpd (ns) is:
tpd = 3.457900e-01
```

Layout & Post-Layout



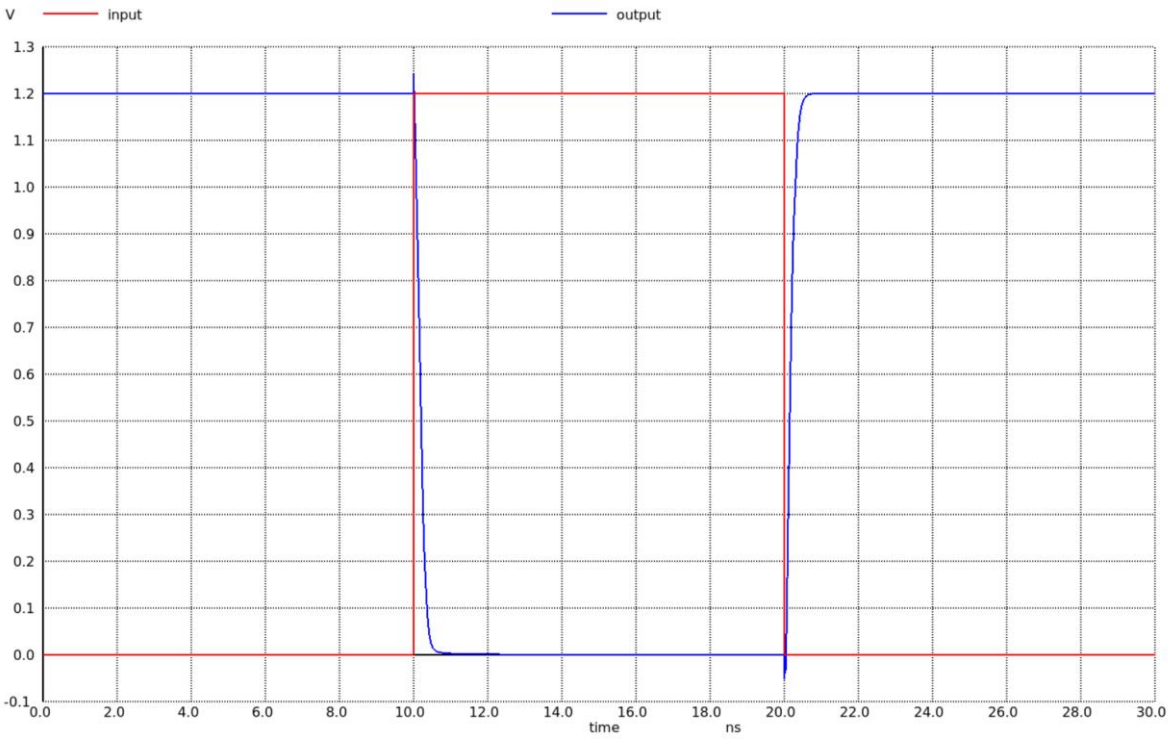



```
v1#branch                      -8.(

 Reference value :  2.99642e-08
No. of Data Rows : 30020
t_input              =  1.000050e-08
t_output             =  2.099597e-08
tpdr (ns) is:
tpdr = 1.099547e+01
t_input              =  2.000150e-08
t_output             =  1.272970e-08
tpdf (ns) is:
tpdf = -7.27180e+00
tpd (ns) is:
tpd = 1.861835e+00

inverse circuit:

t_input              =  1.027990e-08
t_output             =  1.078130e-08
tpdr (ns) is:
tpdr = 5.014000e-01
t_input              =  2.019758e-08
t_output             =  2.182319e-08
tpdf (ns) is:
tpdf = 1.625610e+00
tpd (ns) is:
tpd = 1.063505e+00
ngspice 1 ->
```


D
C
B
A
Y
VSS

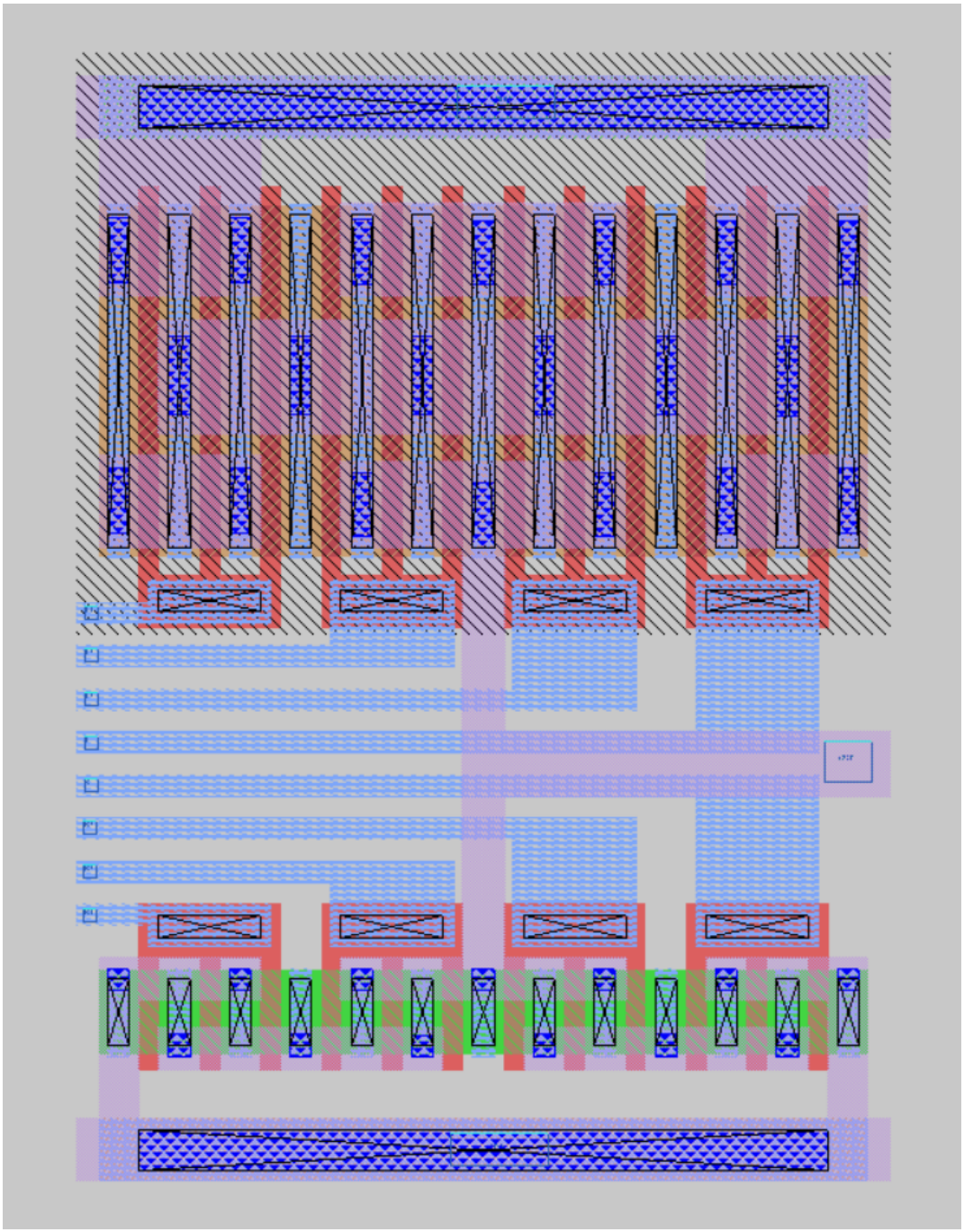
Combined Gate (XOR, XNOR, MUX)

Pre-layout

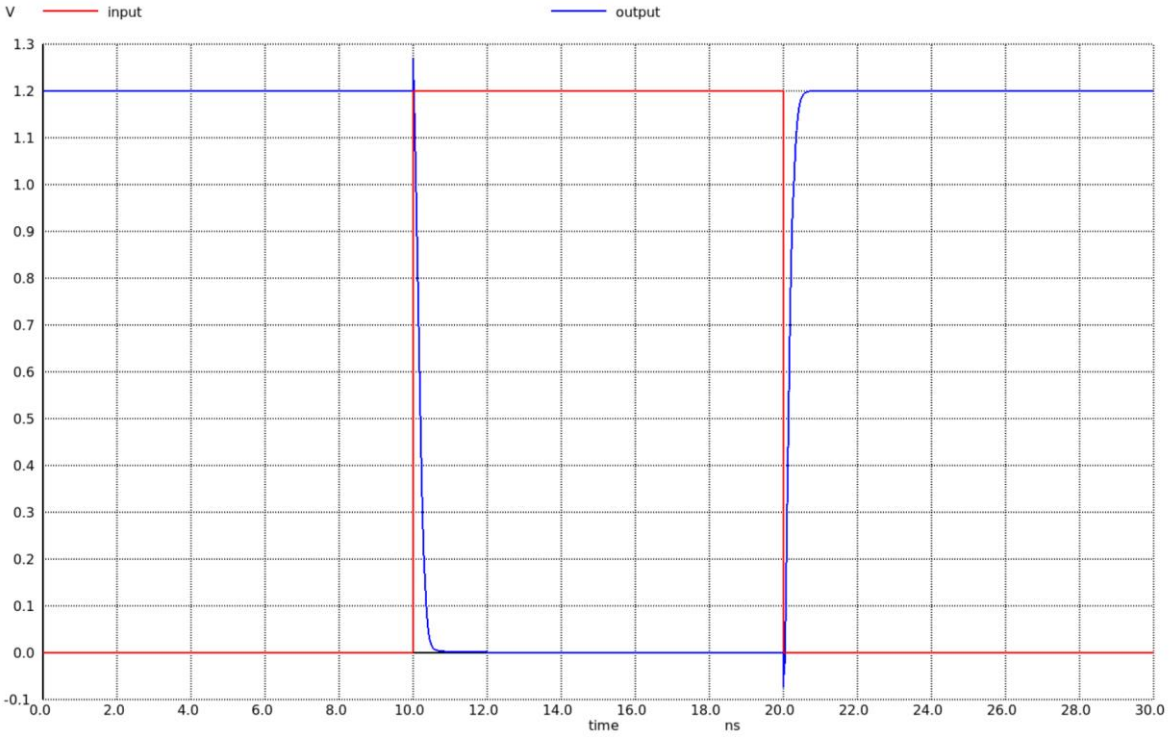



```
 Reference value :  2.99210e-08
No. of Data Rows : 300020
t_input            =  1.000005e-08
t_output           =  1.017853e-08
tpdr (ns) is:
tpdr = 1.784800e-01
t_input            =  2.000015e-08
t_output           =  2.016655e-08
tpdf (ns) is:
tpdf = 1.664000e-01
tpd (ns) is:
tpd = 1.724400e-01
```

Layout & Post-Layout



```
 Reference value :  2.99803e-08
No. of Data Rows : 300020
t_input             =  1.000005e-08
t_output            =  1.017231e-08
tpdr (ns) is:
tpdr = 1.722600e-01
t_input             =  2.000015e-08
t_output            =  2.015707e-08
tpdf (ns) is:
tpdf = 1.569200e-01
tpd (ns) is:
tpd = 1.645900e-01
```



During the layout phase, NMOS and PMOS transistors were horizontally aligned side by side to share diffusion regions, with each having three fingers to achieve a compact design. This arrangement also facilitated easier distribution of the supply and ground voltages and enabled more efficient routing of input and output signal paths to the relevant sections.

After the layout was completed, parasitic extraction was performed, and the flattened design including all parasitic components was simulated on a testbench. The simulation results revealed some differences between the schematic and the layout-based designs. However, these discrepancies were not significant enough to deviate substantially from the desired performance. The newly introduced parasitic capacitances played a major role in these differences, but their impact was not severe enough to critically impair the circuit's operation, which is reassuring.

Conclusion

In this project, various logic gates and circuits were designed and analyzed to meet specific delay constraints and performance metrics. The propagation delay and logic effort for each gate were carefully estimated, and sizing was performed to balance the rise and fall times. The inverter design was revisited to optimize for average delay, demonstrating the importance of trade-offs in circuit design for achieving optimal performance.

NAND2 gates were implemented using both pseudo-logic and dynamic logic approaches, and the results were compared in terms of area, power, and delay, highlighting the advantages and limitations of each design methodology. Layouts for all circuits were created with attention to design rules and parasitic effects. Post-layout simulations revealed variations in delays due to parasitics, and the gate with the maximum relative variation was identified and analyzed.

This project emphasized the importance of layout considerations, including proper metallization and routing practices, to minimize parasitic effects and maintain performance. The results showcase the interplay between theoretical design parameters and practical implementation challenges, providing valuable insights into the design of efficient and reliable digital circuits.

The layouts were drawn without using m2 for routing. The length of polysilicon was kept at minimum. The gates were all placed on the left side of the transistor, power rails ath the top bottom and the output on the right. The logic cells were kept as close to the square as possible. This allows us to use these cells for further projects more easily.