Department of Electronic and Telecommunication Engineering

University of Moratuwa, Sri Lanka

EN 2110 - Electronics - III

Group Project

Report

Submitted by

Caldera H. D. J.	180079X
Oshan J. W. P.	180437U
Thalagala B.P.	180631J

Submitted on

May 13, 2021

Contents

1 Parasitic effect in Timing analysis 2

1.1 System Design 2

1.2 Simulation Results and Discussion 3

2 PLD 5

2.1 Part 1 5

2.2 Part 2 7

2.3 Part 3 11

Name	**Index**	**Contribution**
Caldera H. D. J.	180079X	PLD - Part 1, Part 2
Oshan J. W. P.	180437U	PLD - Part 3
Thalagala B.P.	180631J	Parasitic effect in Timing analysis

Table 1: Contributions of each member

1 Parasitic effect in Timing analysis

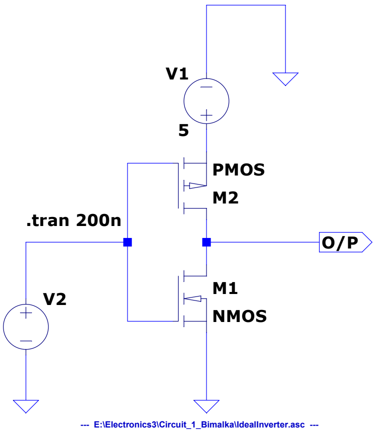
Objective: *Design a 3 stage (3 inverters) ring oscillator. Find the correlation of the parasitic effect in the oscillation period.*

1.1 System Design

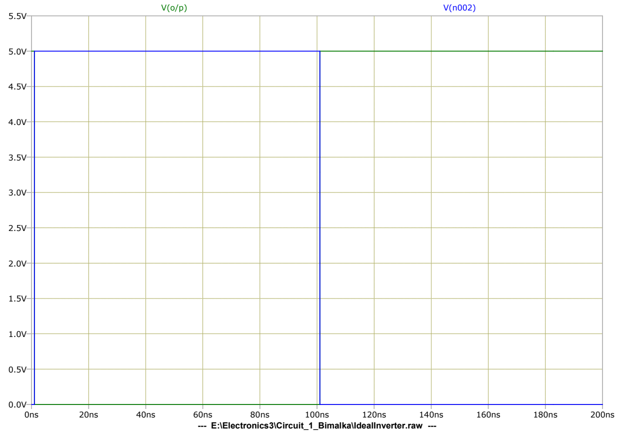
Ring oscillator is a combination of delay stages arranged in series to form a closed loop chain. It consists of an ***odd number of identical inverters (NOT gates)*** and it produces a periodically oscillating output. The period of oscillation(T) of a ring oscillator can be expressed as follows where n is the number of cascaded NOT gates and τ_d is the propagation delay of a single stage.

$$T = 2.n.\tau_d \quad \Longrightarrow \quad Oscillation\ frequency = \frac{1}{2.n.\tau_d}$$

Following figure illustrates the input output characteristic of an ideal inverter. There, output is changed as soon as the input signal changes. That is no propagation delay.



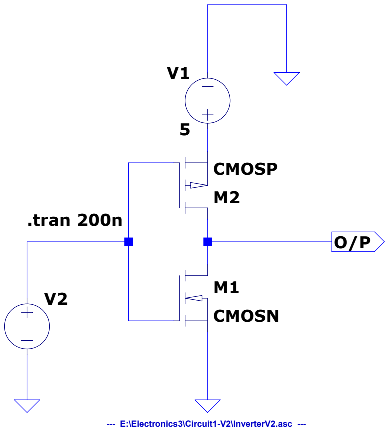
(a) Ideal Inverter Model



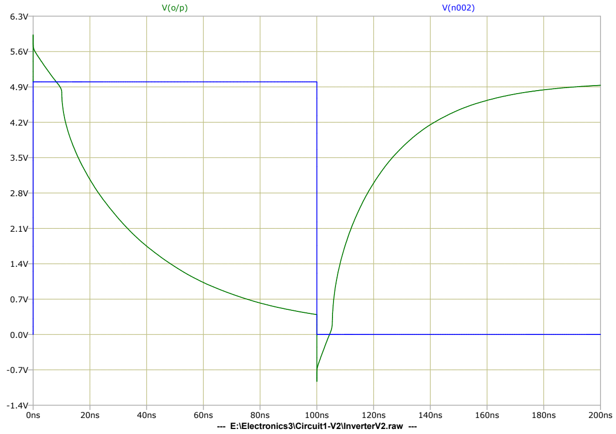
(b) Ideal inverter Output

Whereas the below figure illustrates the input output characteristic of a single stage of a general real inverter. It can be observed that finite amount of time is required for output to be valid for a given valid input due to the time taken to charge and discharge the internal parasitic capacitors. This model has been used to find the parasitic effect on the oscillation period as it is realistic. For this experiment an NMOS and PMOS models were custom designed and Parameters of those models are as follows.

```
.MODEL CMOSP PMOS KP=96u VTO=0.906 LAMBDA=0.01 TOX=21n GAMMA=0.486 CGDO=54p CGSO=54p CGBO=336p
.MODEL CMOSN NMOS KP=96u VTO=0.786 LAMBDA=0.01 TOX=21n GAMMA=0.586 CGDO=402p CGSO=402p CGBO=362p
```



(c) Inverter Model with additional delay element



(d) Effect of Parasitic Capacitance

Basic structure of a ring oscillator can be depicted as follows using logic gates.

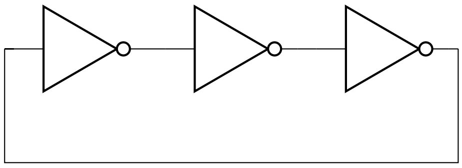


Figure 1: Basic structure of a 3 stage ring oscillator

A genral Ring Oscillator can be implemented using NMOS, PMOS transistors and additional delay elements(capacitors). But as the objective is to find the correlation between parasitic effect and oscillation period, the additional delay elements were not used and only the internal capacitance were taken in to account. Each inverter consists of one PMOS and one NMOS named as CMOSP and CMOSN.

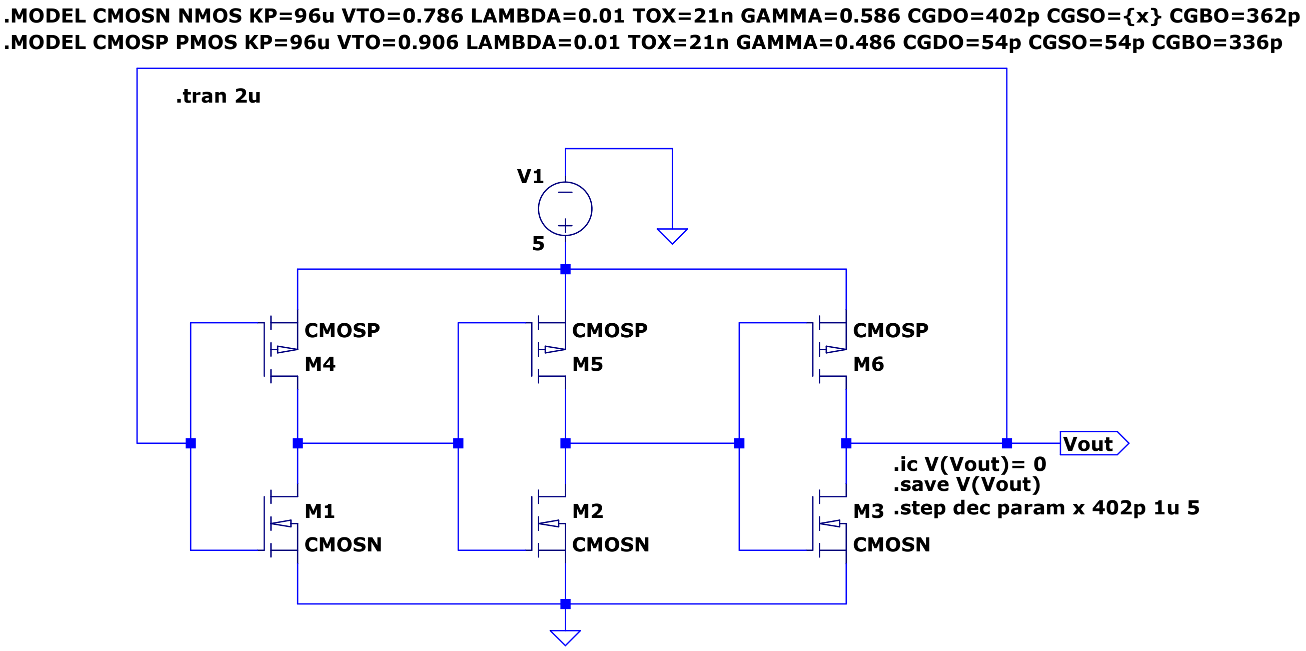
1.2 Simulation Results and Discussion

Note 1: *Only the gate source capacitance were considered for the analysis, as it is what matters the most for the propagation delay of a MOSFET. In addition to that when changing the capacitor values, maximum gate source capacitance was set to be 1 μF only to clearly illustrate the variation in the oscillation period. The real values are in pF range.*

Note 2: *Gate source capacitance of the PMOS transistors' were kept constant and only the NMOS transistors' gate source capacitance were changed as it is sufficient for identifying the effect of parasitic capacitance on the period of oscillation.*

Following ring oscillator schematic has been used to simulate the effect of parasitic capacitance on the period of the output waveform. Simulation was run for 2 μs time intervals(`.tran 2u`) for 18 different gate source capacitor values starting form 402 pF to 1 μF.

For the simulation to work an initial pulse is required. This requirement was satisfied through defining an initial condition(`.ic V(Vout) = 0`) for the `Vout` node using LTSpice XVII's *Spice Directive* feature. The same feature was used to sweep through different gate source capacitor values. (`.save V(Vout), .step dec param x 402p 1u 5`)



```
.MODEL CMOSN NMOS KP=96u VTO=0.786 LAMBDA=0.01 TOX=21n GAMMA=0.586 CGDO=402p CGSO={x} CGBO=362p
.MODEL CMOSP PMOS KP=96u VTO=0.906 LAMBDA=0.01 TOX=21n GAMMA=0.486 CGDO=54p CGSO=54p CGBO=336p
.tran 2u
.ic V(Vout)= 0
.save V(Vout)
.step dec param x 402p 1u 5
```

Figure 2: 3 stages CMOS Ring Oscillator

When the gate source capacitance is increased, as the plot depicts the period of oscillation increases and hence the frequency decreases. Because as mentioned at the beginning the period of oscillation(T) is affected by τ_d, the propagation delay of a single stage which depends on the parasitic capacitance.

$$Oscillation\ frequency = \frac{1}{2.n.\tau_d}$$

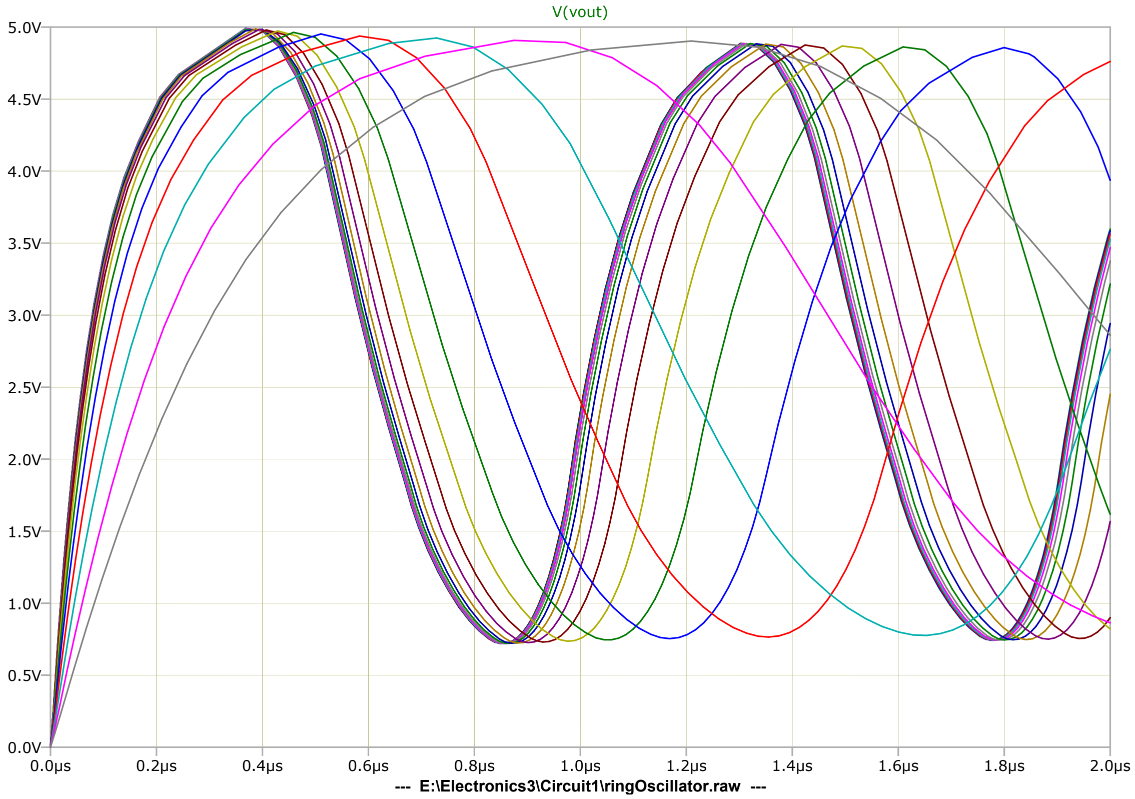


Figure 3: Waveform for Voltages of 3 stages CMOS Ring Oscillator

2 PLD

2.1 Part 1

Objective: *Design a programmable logic block to configure it as a 'NAND' or a 'NOR' gate using a single selection bit.*

First a truth table is drawn for this part considering a single selection bit (S) with two inputs (A, B) such that S=0 for 'NAND' and S=1 for 'NOR' operations respectively.

S	A	B	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Table 2: The truth table

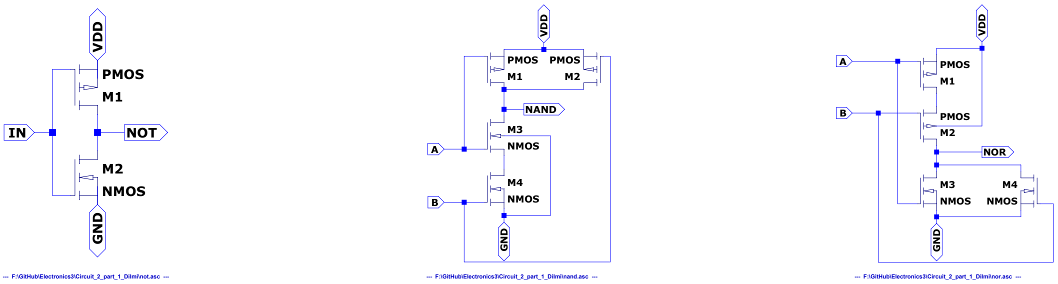
Then the relevant logic expression was obtained using a karnaugh map and it was further simplified to obtain the combination of 'NAND' and 'NOR' operations.

S\AB	00	01	11	10
0	1	1	0	1
1	1	0	0	0

Table 3: Karnough Map for the above truth table

$$\begin{aligned} F &= \overline{S}.\overline{A} + \overline{S}.\overline{B} + \overline{A}.\overline{B} \\ &= \overline{S}.(\overline{A} + \overline{B}) + \overline{A}.\overline{B} \\ &= \overline{S}.(\overline{A.B}) + \overline{A+B} \\ &= \overline{S + A.B} + \overline{A+B} \\ &= \overline{(S + A.B).(A+B)} \\ &= \overline{(S + \overline{\overline{A.B}}).(\overline{\overline{A+B}})} \end{aligned}$$

So, the resultant combinational logic circuit is as follows. (2 NANDs, 2 NORs, 3 NOTs) For the implementation of this circuit; 'NOT', 'NAND', and 'NOR' gates were designed using 'NMOS' and 'PMOS' transistors. Their schematics in LTspice are depicted below.


VDD
PMOS
M1
IN
NOT
M2
NMOS
GND
— F:\GitHub\Electronics3\Circuit_2_part_1_Dilmi\not.asc —
VDD
PMOS
PMOS
M1
M2
NAND
M3
NMOS
A
M4
NMOS
B
GND
— F:\GitHub\Electronics3\Circuit_2_part_1_Dilmi\nand.asc —
VDD
A
PMOS
M1
B
PMOS
M2
NOR
M3
M4
NMOS
NMOS
GND
— F:\GitHub\Electronics3\Circuit_2_part_1_Dilmi\nor.asc —

(a) Schematic of NOT gate (b) Schematic of NAND gate (c) Schematic of NOR gate

Figure 4: Basic Gates in the CMOS logic

Finally, the PLD block is designed using the above gates and the waveforms were obtained.

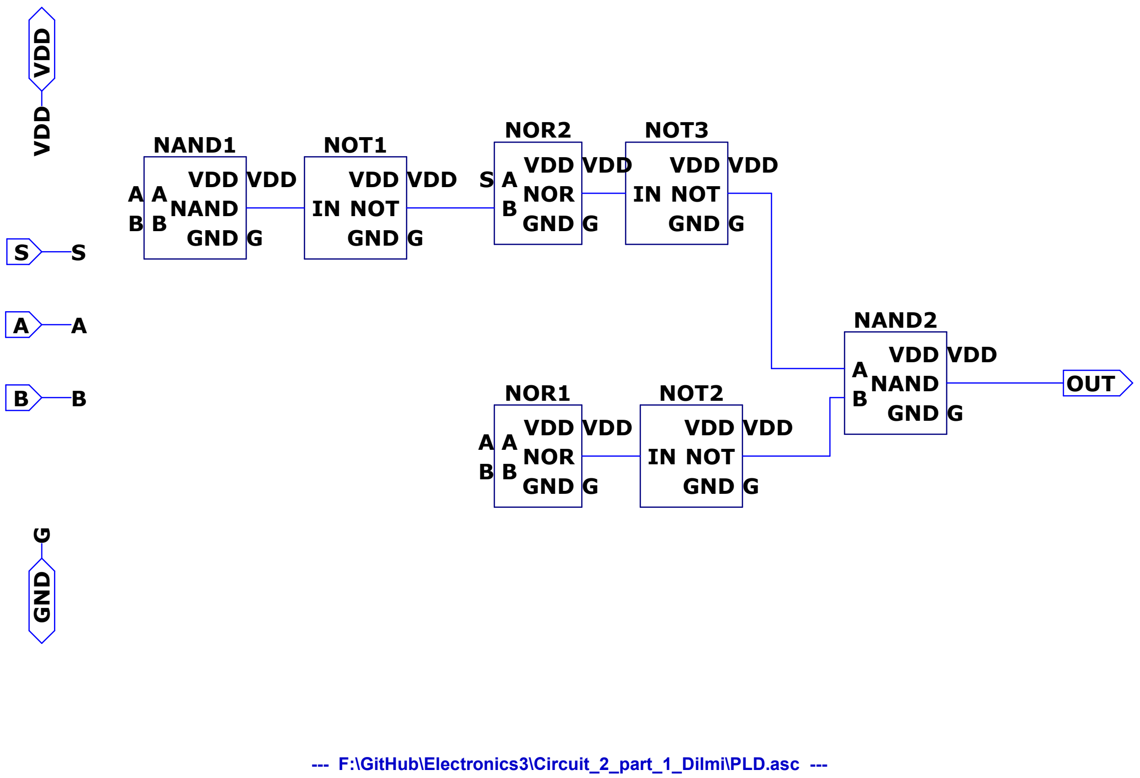


Figure 5: Circuit designed using logic blocks

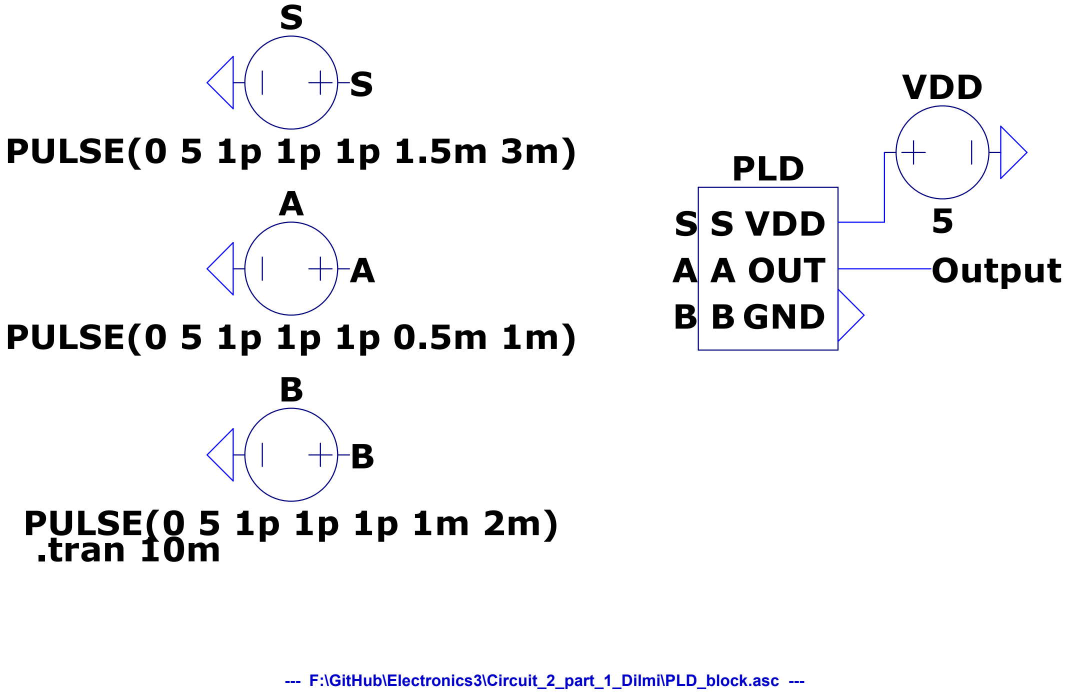

S
PULSE(0 5 1p 1p 1p 1.5m 3m)
A
PULSE(0 5 1p 1p 1p 0.5m 1m)
B
PULSE(0 5 1p 1p 1p 1m 2m)
.tran 10m
PLD
S VDD
A OUT
B GND
VDD
5
Output
--- F:\GitHub\Electronics3\Circuit_2_part_1_Dilmi\PLD_block.asc ---

Figure 6: Designed PLD block

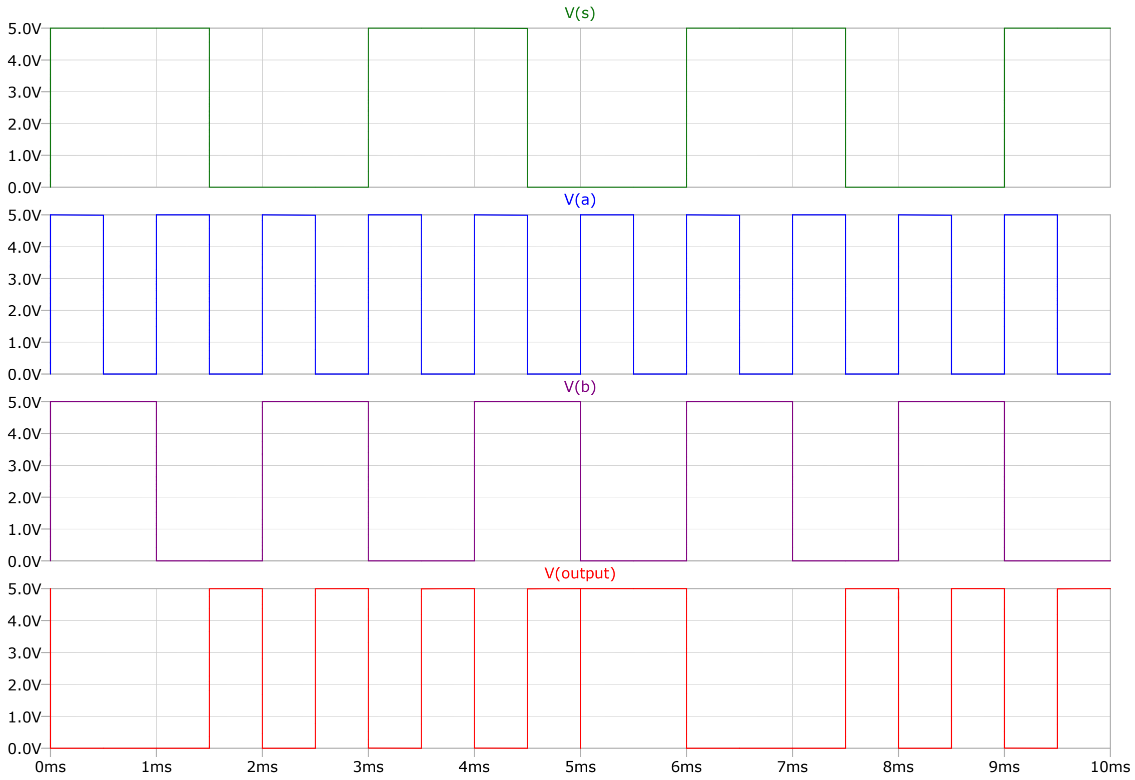
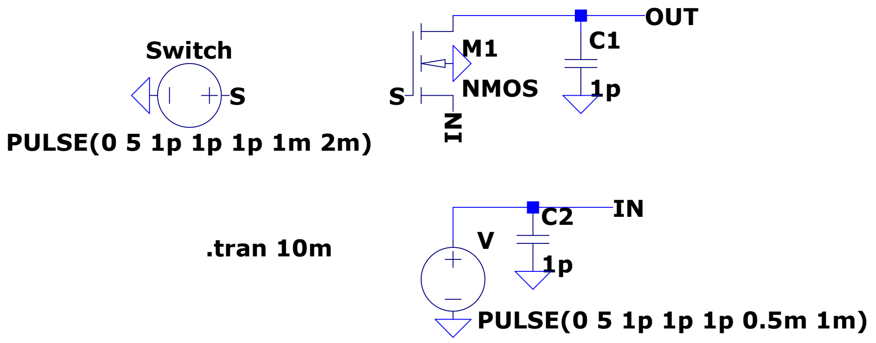


Figure 7: Waveforms for inputs and output of PLD

2.2 Part 2

Objective: *Design a single switch matrix using six pass transistors.*

In this part, the single switch matrix is needed to be designed using six pass transistors. So as the first step, a pass transistor was designed and its performance was checked. Simply an 'NMOS' transistor is fed with a switch, could be used for this task. So when the switch is on, the input signal will be received at the output (Threshold voltage is considering as zero since an ideal nmos). Also capacitors were used to ground the high impedance state which occurred when the NMOS is OFF.



--- F:\GitHub\Electronics3\Circuit_2_part_2_Dilmi\pass_ttr.asc ---

Figure 8: Schematic diagram of the pass transistor

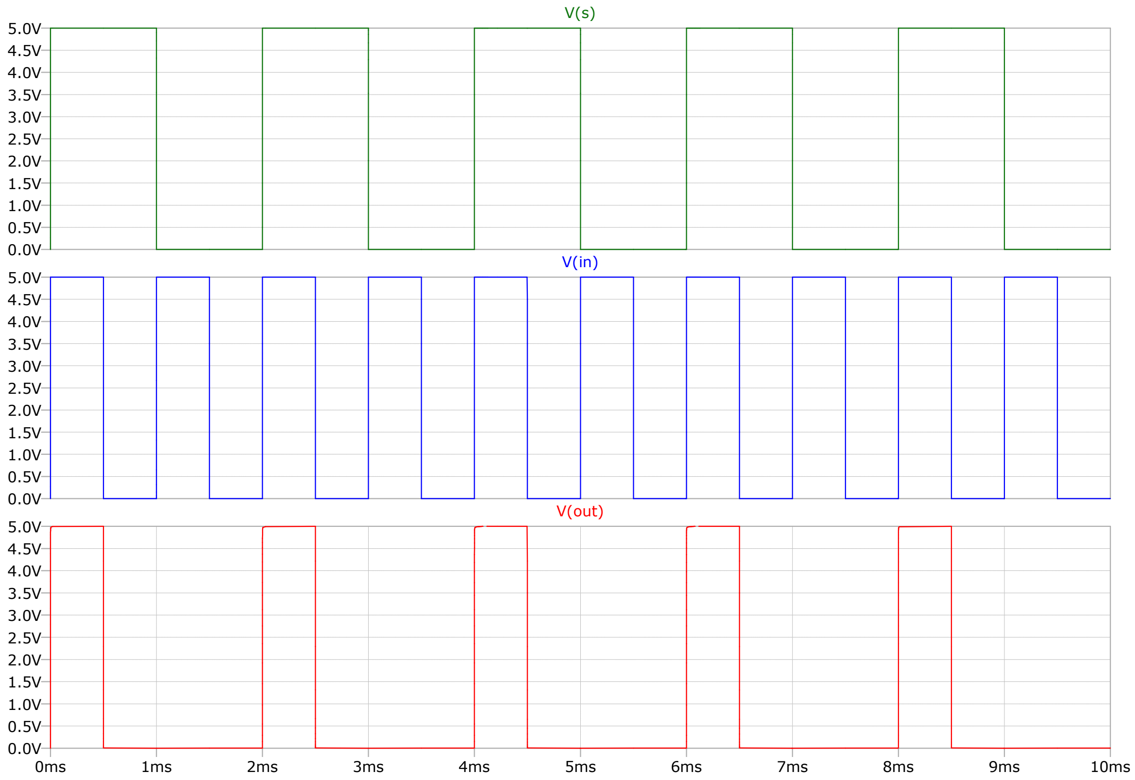


Figure 9: Waveform of the pass transistor

Then using six such transistors, the single switch matrix was designed and the schematic diagram of it is shown below.

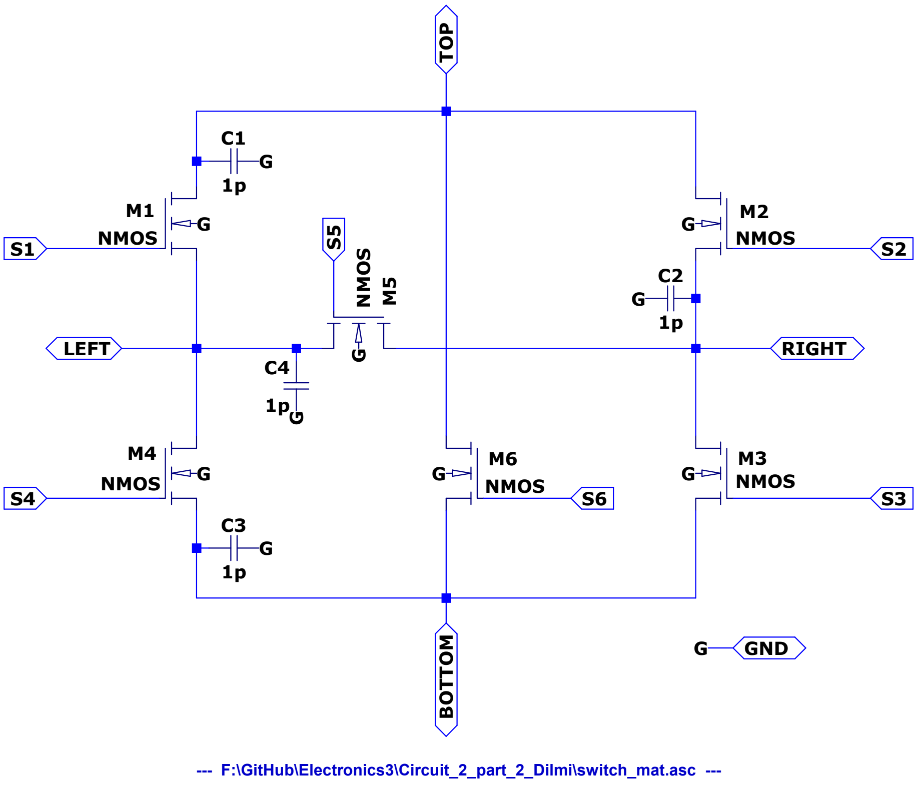


Figure 10: Schematic diagram of the single switch matrix

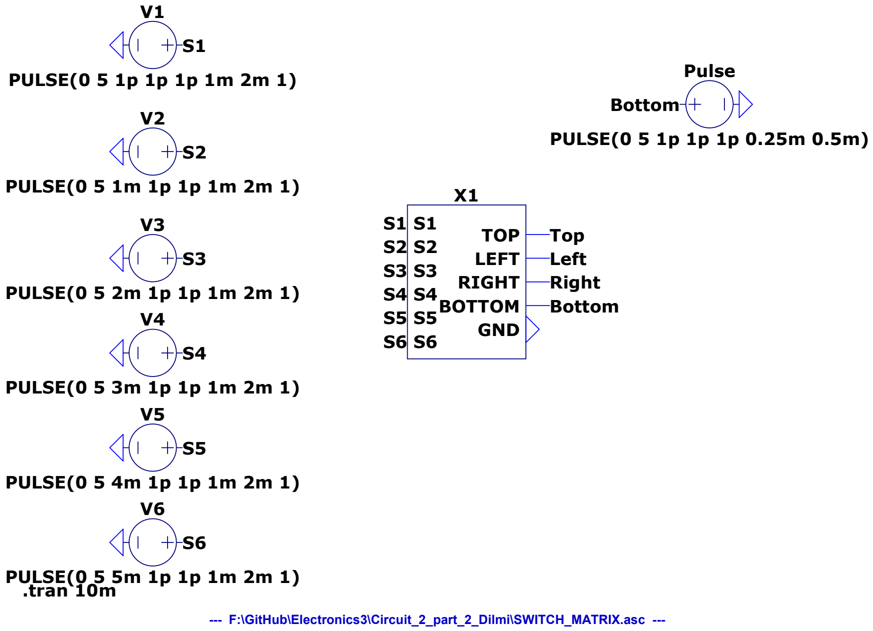

V1
S1
PULSE(0 5 1p 1p 1p 1m 2m 1)
V2
S2
PULSE(0 5 1m 1p 1p 1m 2m 1)
V3
S3
PULSE(0 5 2m 1p 1p 1m 2m 1)
V4
S4
PULSE(0 5 3m 1p 1p 1m 2m 1)
V5
S5
PULSE(0 5 4m 1p 1p 1m 2m 1)
V6
S6
PULSE(0 5 5m 1p 1p 1m 2m 1)
.tran 10m
Pulse
Bottom
PULSE(0 5 1p 1p 1p 0.25m 0.5m)
X1
S1 S1
S2 S2
S3 S3
S4 S4
S5 S5
S6 S6
TOP
LEFT
RIGHT
BOTTOM
GND
Top
Left
Right
Bottom
--- F:\GitHub\Electronics3\Circuit_2_part_2_Dilmi\SWITCH_MATRIX.asc ---

Figure 11: Designed single switch matrix block

Finally the functionality of the circuit was checked by giving pulses to left, right, top, and bottom corners separately and switching on the switches at different periods.

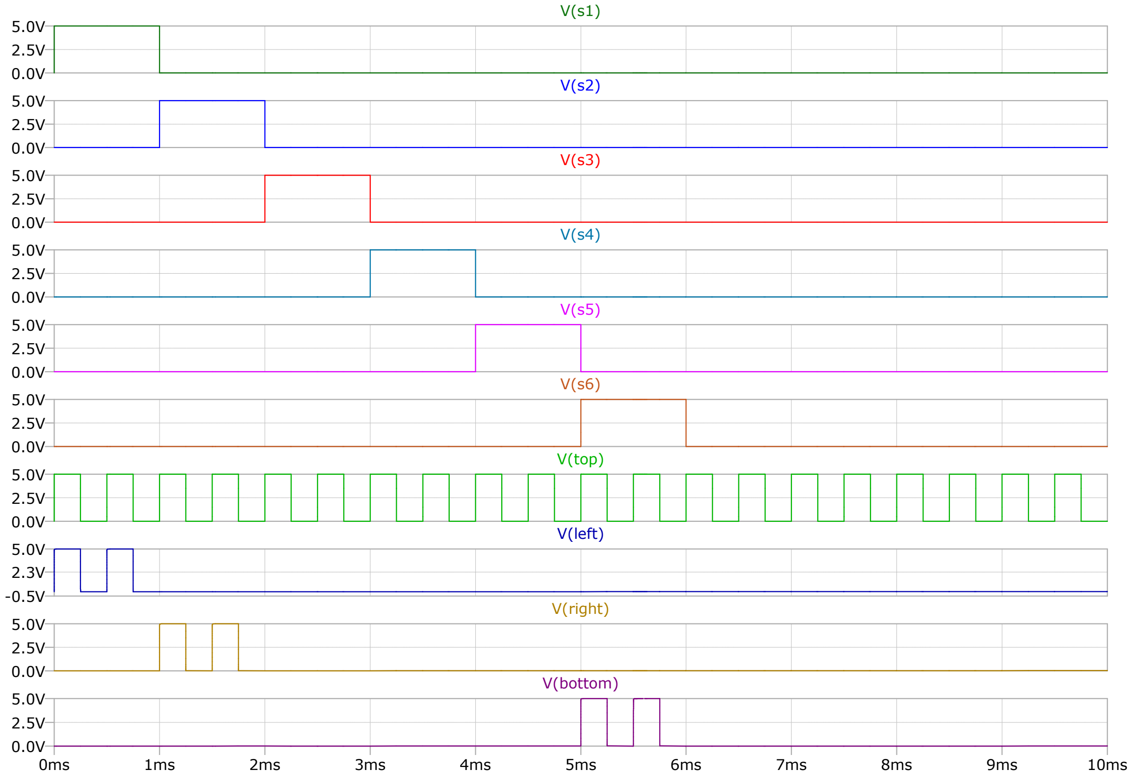


Figure 12: Waveforms when the top terminal is fed with a pulse

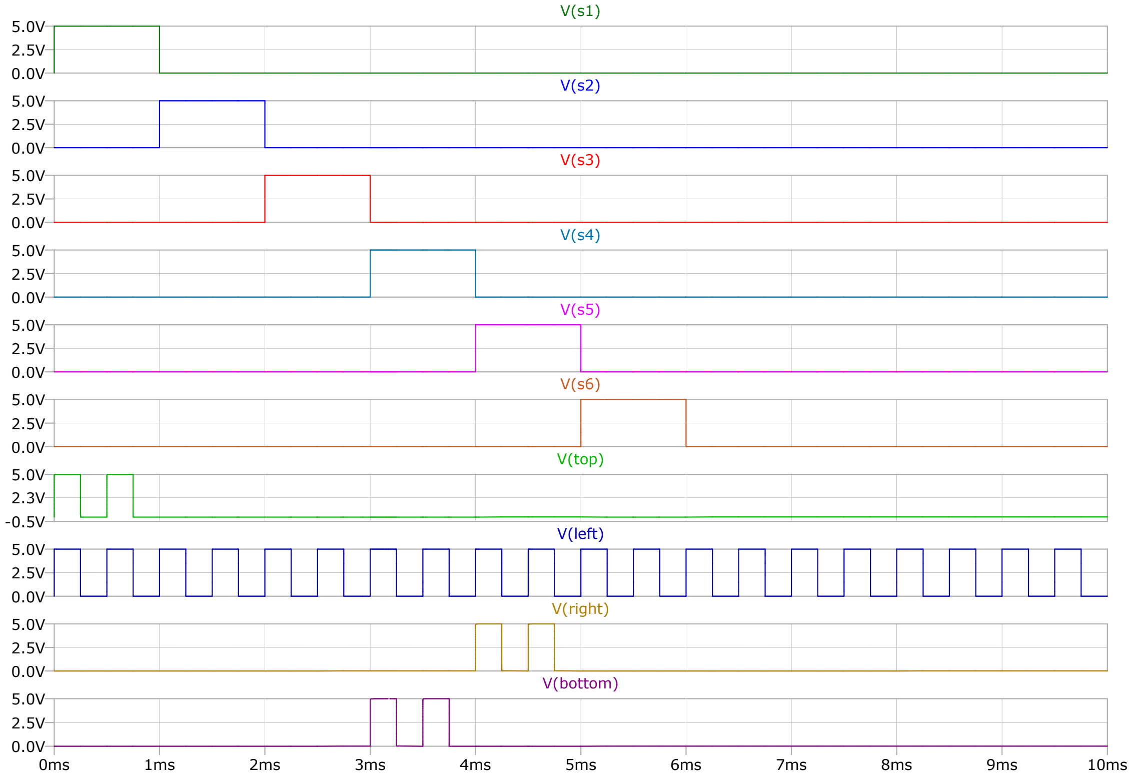


Figure 13: Waveforms when the left terminal is fed with a pulse

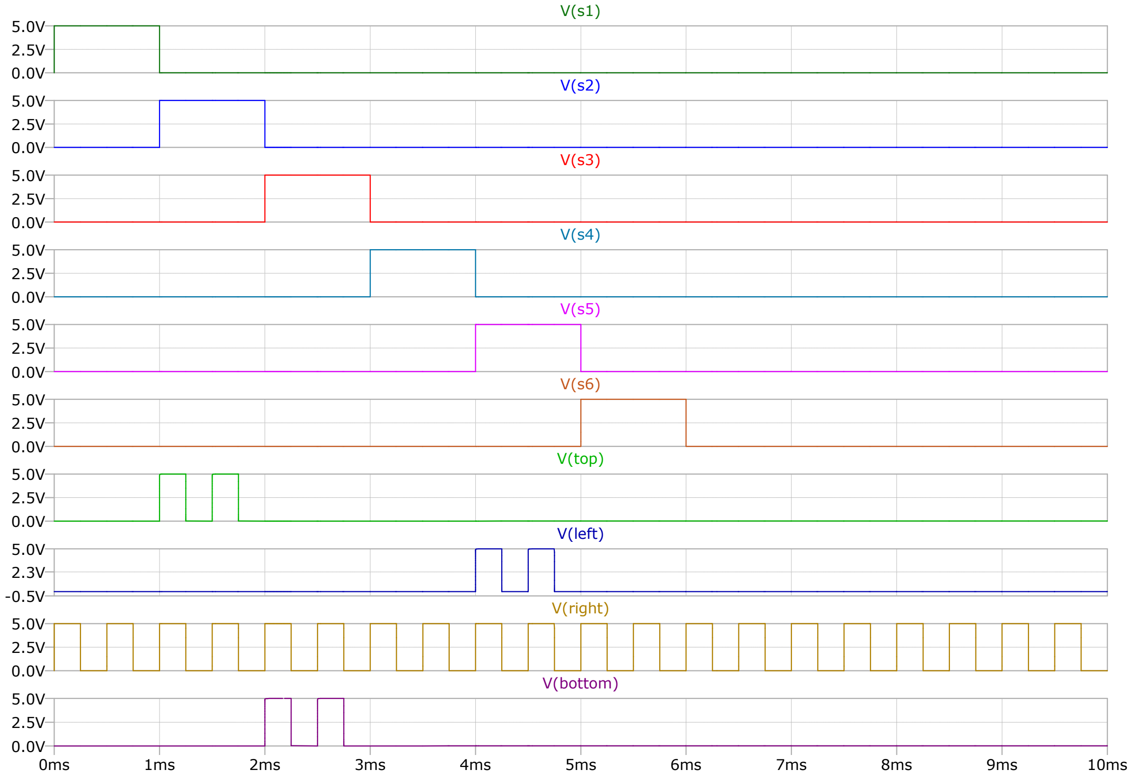


Figure 14: Waveforms when the right terminal is fed with a pulse

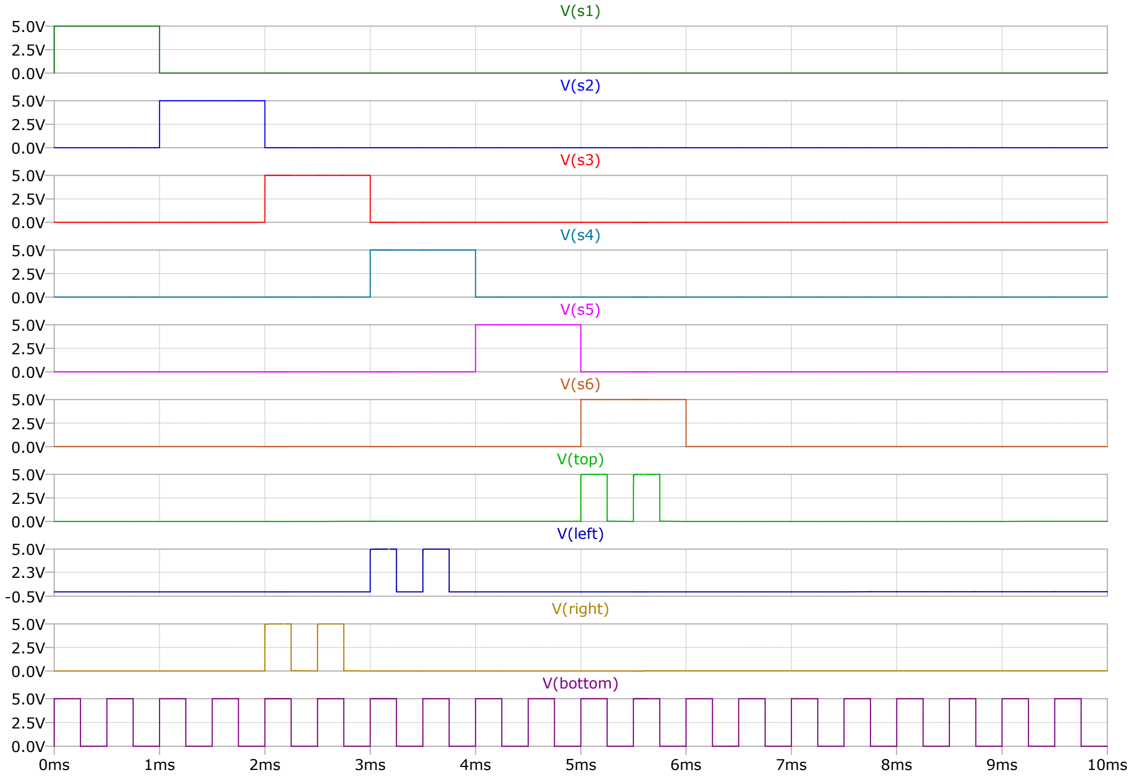


Figure 15: Waveforms when the bottom terminal is fed with a pulse

2.3 Part 3

Objective: *Design a PLD that can be used to design any 3 input combinational circuit.*

The task was to design a PLD circuit capable of implementing any three input combinational circuit. The truth-table of any three input combinational circuits will be as below.

A	B	C	Output
0	0	0	S_1
0	0	1	S_2
0	1	0	S_3
0	1	1	S_4
1	0	0	S_5
1	0	1	S_6
1	1	0	S_7
1	1	1	S_8

Table 4: The truth-table of any three input combinational circuit

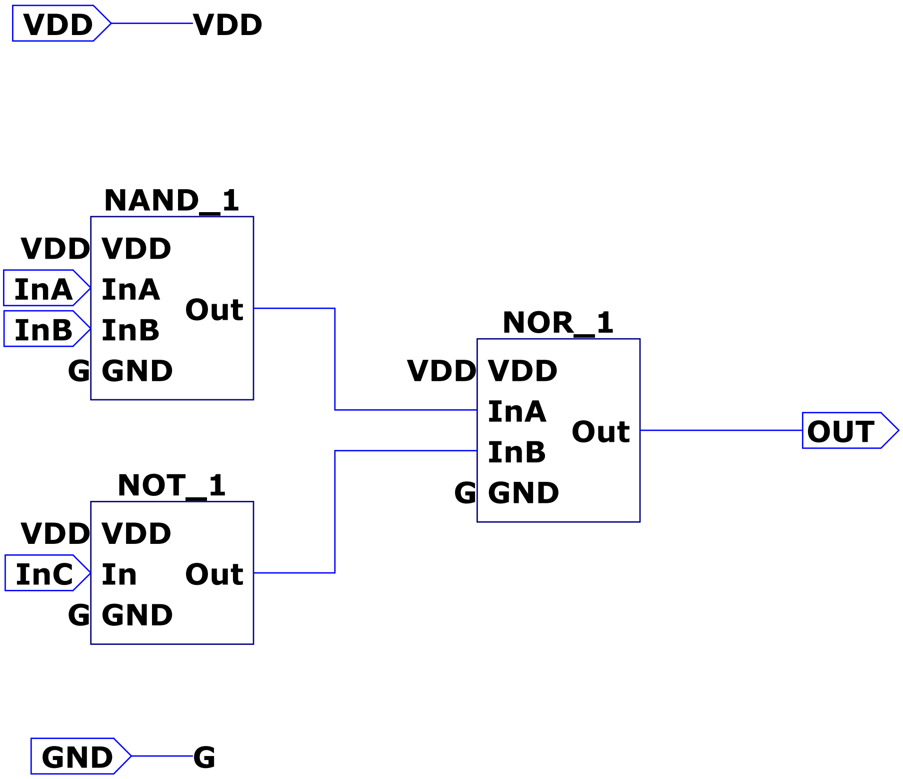
Outputs S_1, S_2,...., S_8 differ with the combinational circuit. So we can write an expression for the combinational logic circuit using the 8 minterms. Which minterms to be selected differ according to the S1, S2,...., S8. If any S_i is 1 then the corresponding minterm is taken into the sum of products expression. If Si is 0 that corresponding minterm is discarded.

So we can build the PLD with a fixed AND plane which has all eight minterms and a programmable OR plane which can be programmed using Si terms. So our PLD becomes a PROM.

Before building the PLD the AND plane and OR plane should be created. For the fixed AND plane, we need eight minterms. A minterm is a product of any three of A, $\overline{A}$, B, $\overline{B}$, C or $\overline{C}$. So we need three input and gate. We configured a three-input AND gate using NAND, NOR, and NOT gates as below for better efficiency.

$$A.B.C = \overline{\overline{A.B.C}} = \overline{\overline{A.B} + \overline{C}}$$

Using this expression we constructed the 3 input AND gates using a minimum number of logic gates.



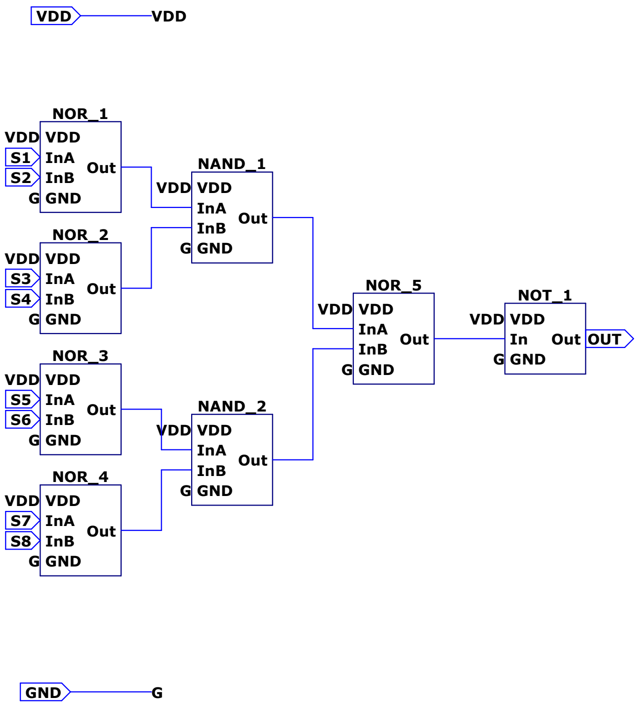
--- D:\aca\github\electronics3\circuit_2_part_3_oshan\3_and.asc ---

Figure 16: Implementing the three-input AND gate using NOR, NAND, and NOT gates.

Using seven separate OR gates(7 NOR gates + 7 NOT gates) to implement the OR plane, increases complexity and the latency of the circuit by a huge factor. Instead, we can simplify the expression and use a minimum number of gates as below.

$$\begin{aligned}
&= S_1 + S_2 + S_3 + S_4 + S_5 + S_6 + S_7 + S_8 \\
&= \overline{\overline{S_1 + S_2 + S_3 + S_4 + S_5 + S_6 + S_7 + S_8}} \\
&= \overline{\overline{(S_1 + S_2 + S_3 + S_4)}.\overline{(S_5 + S_6 + S_7 + S_8)}} \\
&= \overline{\overline{(S_1 + S_2)}.\overline{(S_3 + S_4)}.\overline{(S_5 + S_6)}.\overline{(S_7 + S_8)}} \\
&= \overline{\overline{(S_1 + S_2)}.\overline{(S_3 + S_4)}} + \overline{\overline{(S_5 + S_6)}.\overline{(S_7 + S_8)}} \\
&= \overline{\overline{\overline{\overline{(S_1 + S_2)}.\overline{(S_3 + S_4)}} + \overline{\overline{(S_5 + S_6)}.\overline{(S_7 + S_8)}}}}
\end{aligned}$$

Using this expression we were able to build an OR plane with a minimum number of components as below.

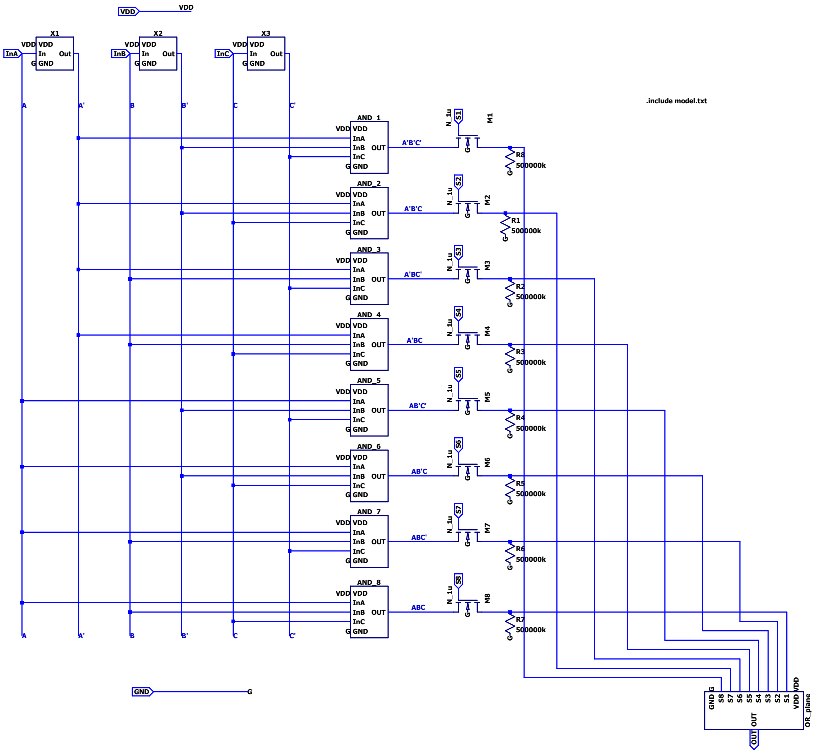


--- D:\aca\github\electronics3\circuit_2_part_3_oshan\8_or.asc ---

Figure 17: Implementing the OR plane

Instead of using a total of 14 logic gates, now we have implemented it using only 8 logic gates. This reduces the latency and complexity by a huge factor.

PROM is constructed as below



--- D:\ACA\Github\Electronics3\Circuit_2_part_3_Oshan\New folder\PROM.asc ---

Figure 18: PROM circuit

We have used nmos transistors as switches which choose, which minterms are taken into the sum of products. We didn't choose passgates as switches as it increases the complexity and the latency of the circuit.

We tested the circuit for different combinational circuits by configuring Si switches. Below we have configured the PROM as a simple NOR gate.

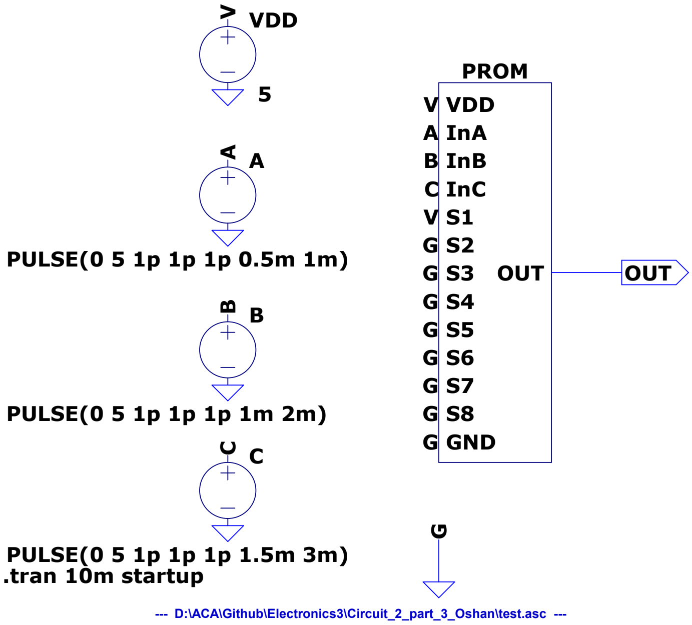

V
VDD
5
A
A
PULSE(0 5 1p 1p 1p 0.5m 1m)
B
B
PULSE(0 5 1p 1p 1p 1m 2m)
C
C
PULSE(0 5 1p 1p 1p 1.5m 3m)
.tran 10m startup
PROM
V VDD
A InA
B InB
C InC
V S1
G S2
G S3
G S4
G S5
G S6
G S7
G S8
G GND
OUT
OUT
G
--- D:\ACA\Github\Electronics3\Circuit_2_part_3_Oshan\test.asc ---

Figure 19: PROM configured as a NOR gate

Results were as below,

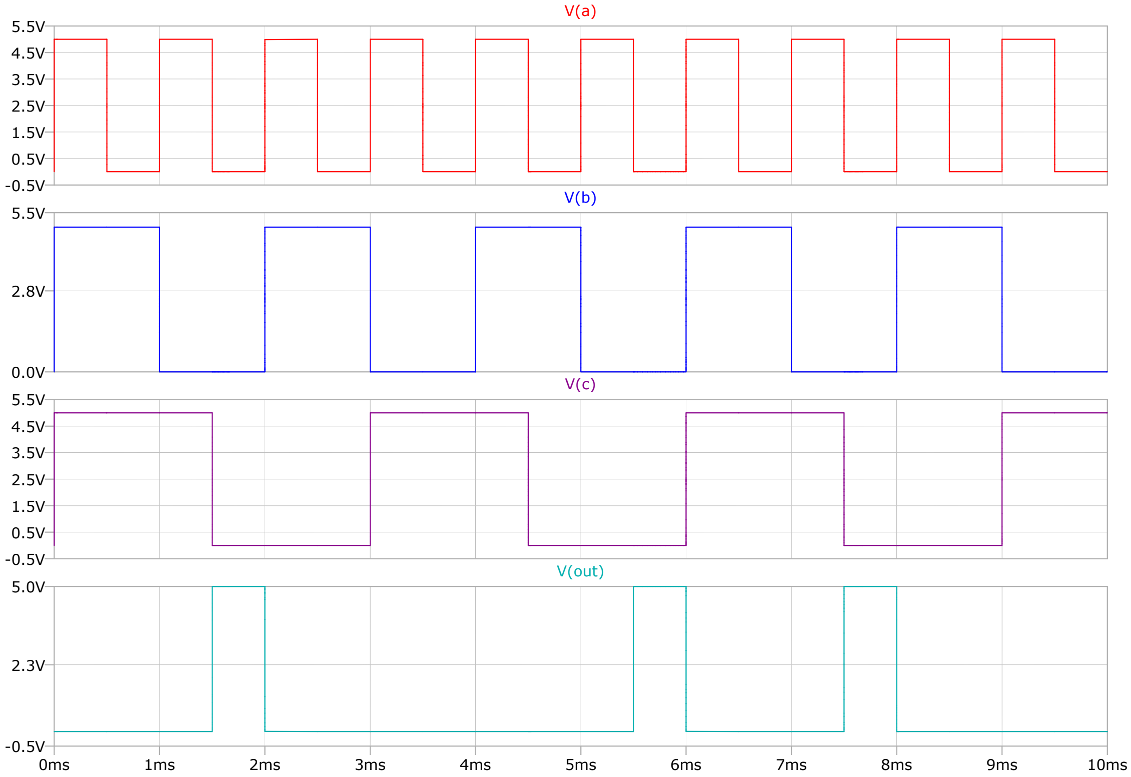


Figure 20: Results of PROM configured as a NOR gate

We can observe that the PROM is functioning correctly.