Department of Electronic & Telecommunication Engineering

University of Moratuwa

EN 2110 – Electronics III

Circuit Designs and Simulations

Group no. 13

Group Member	**Index number**
Premathilaka H. D. M.	180497C
Samarasinghe P.	180554B
Sauranga H. W. C.	180574K

This report is submitted in partial fulfillment of the requirements
for the module EN 2110 – Electronics III.

Date of submission: 14/05/2021

Individual Contributions

Circuit 01 (Parasitic effect in timing analysis)	3-Stage ring oscillator	180497C
Circuit 02 (PLD)	Programmable logic block for NAND and NOR	180554B, 180497C
	Single switch matrix using six pass transistors	180497C, 180554B
	PLD for any 3-input combinational logic circuit	180574K

- The circuits were designed and simulated using LTspice XVII.
- The graphs were plotted using Excel 2016.
- The diagrams were drawn using draw.io

Table of Contents

Circuit 01 – Parasitic Effect in Timing Analysis 3

1. 3-Stage ring oscillator 3

Circuit 02 – PLD 7

1. Programmable logic block for NAND and NOR 7
2. Single switch matrix using six pass transistors 11
3. Designing a 3-input PLD 15

Circuit 01 – Parasitic Effect in Timing Analysis

1. 3-Stage ring oscillator

- A ring oscillator is a device that consists of an odd number of cascaded NOT gates (inverters) and the output of the circuit is provided as a feedback to the input of the circuit.
- Consequently, the output of the circuit continuously oscillates between two voltage levels (high and low).
- The following diagrams represent the PMOS-NMOS implementation of a NOT gate which is the basic building block of a ring oscillator.

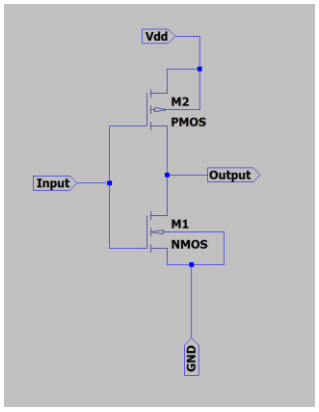

Figure 1 - PMOS/NMOS implementation of a NOT gate

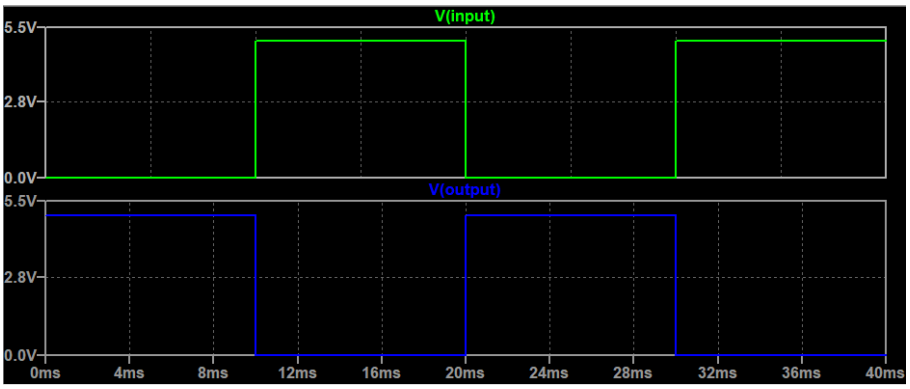

Figure 2 - Output of a NOT gate

- Thus, by cascading 3 such blocks, we can design a three-stage ring oscillator.

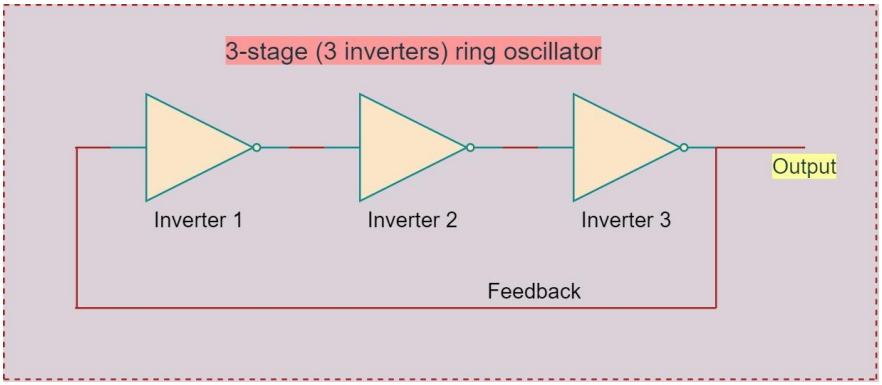

Figure 3 - Ring oscillator diagram

- In physical devices, it is impossible for the outputs to switch instantaneously due to various factors.
- For example, fabrication of a NOT gate using NMOS and PMOS produces a parasitic capacitance at the output node affecting the propagation.
- Therefore, the output of every inverter in a ring oscillator changes within a finite amount of time after the input has changed.
- Owing to this fact, the ring oscillator falls into the class of time-delay oscillators.
- Moreover, the gate delay is considered to be the time between when the input and its output cross the toggle point.

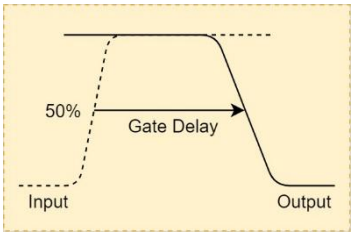

Figure 4 - Gate delay of a NOT gate

- The schematic of the 3 stage ring oscillator is as follows.

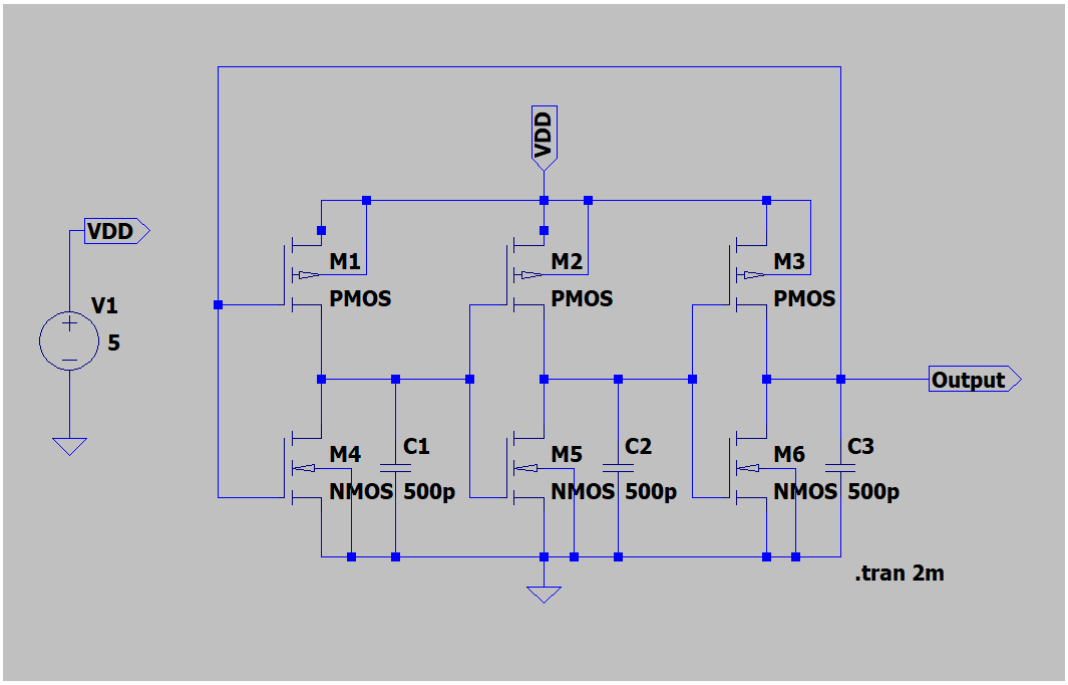

Figure 5 - Ring oscillator schematic

- In the above schematic, we have used **ideal PMOS/NMOS transistors. Thus, to replicate the effect of parasitic capacitances on the propagation of an input, we have connected a capacitor at the output node of each gate.**
- **For simulations, we used C1, C2 and C3 values that are in range with gate capacitances found in commercially available PMOS/ NMOS transistors.**

- For the above setup, the output waveform varies as given below.

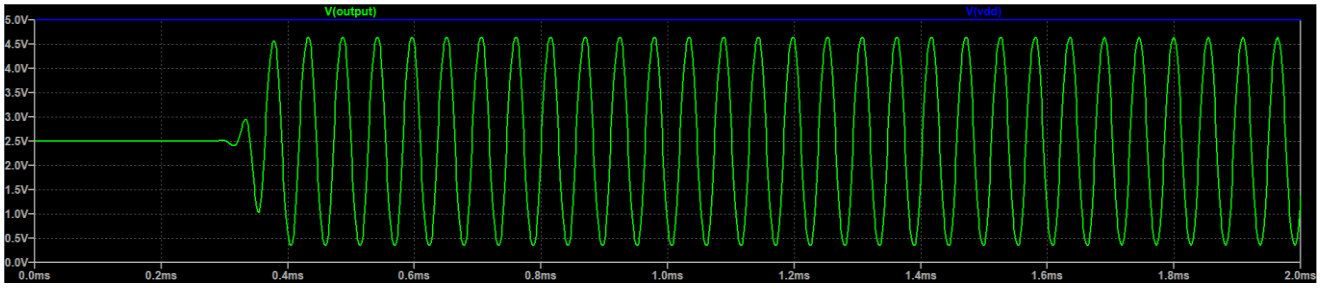

Figure 6 - Output waveform of the ring oscillator

- At each gate, the input waveform will be subjected to a phase shift due to the gate delay. Consider the following diagram. For a given instance, we can clearly see the differences in amplitudes of the outputs at each gate (on the black line).

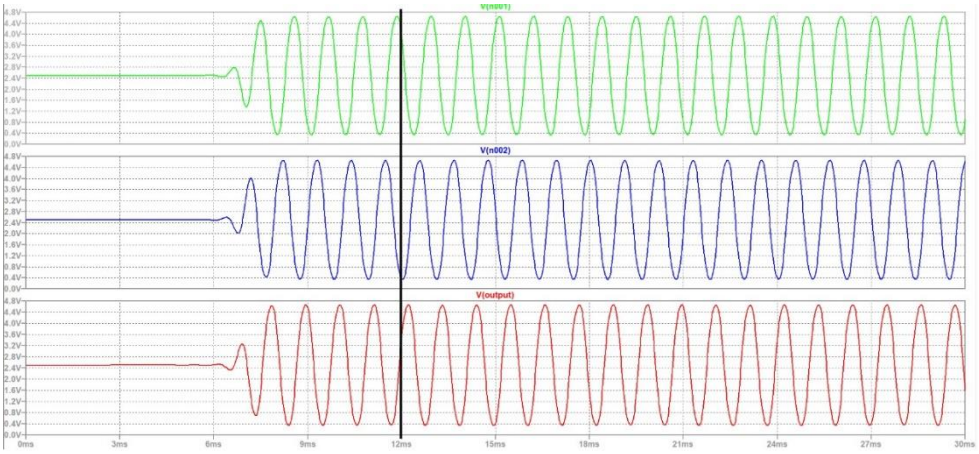

Figure 7 - Phase shifts due to gate delays

- Accordingly, the period of oscillation for a n-stage ring oscillator would be equal to;

$$Period\ of\ oscillation = \sum (t_{PLH} + t_{PHL})_i \ \ where\ i = 1,2,\ldots.n\ \ (n\ is\ odd)$$

- To maintain simplicity, let's assume that the PMOS and the NMOS transistor of a given stage have the same drive capability. Thus, it is safe to assume that for a given stage, the delays of NMOS and PMOS transistors are equal.
- Moreover, let's take $t_{PLH} \approx t_{PHL} = t_d$.
- As shown in figure 5, we have considered the parasitic capacitances of all the stages to be equal. Therefore, all stages would have the same propagation delay.
- As a result, the aforementioned equation can be reduced to;

$$Period\ of\ oscillation\ = 2\ \times no.of\ stages\ \times propagation\ delay\ of\ one\ stage$$

$$T_{period} = 2\ \times n \times t_d$$

- Propagation delay (t_d) directly depends on the parasitic capacitances. Therefore, let's try to figure out a relation between the parasitic capacitances and the period of oscillation.
- The following table depicts the period of oscillation observed for different parasitic capacitances.

Table 1 - Period of oscillation for different parasitic capacitances

Parasitic Capacitance (pF)	Period of Oscillation (µs)	Frequency of Oscillation (kHz)
50	5.5112	181.4486863
100	11.0389	90.5887362
200	22.0779	45.29416294
500	54.9509	18.19806409
1000	109.8543	9.102966384

- We can observe a linear relation between the parasitic capacitances and the period of oscillation.

- In summary,

$$t_d \propto C_p \rightarrow t_d = k \times C_p \text{ where } k \text{ is a constant}$$

$$T_{period} = 2 \times 3 \times k \times C_p \quad (n = 3)$$

$$\boxed{T_{period} = 6k \times C_p}$$

- The following plot concludes that the correlation between the parasitic capacitances and period of oscillation is nearly linear (ideally, this would be perfectly linear).

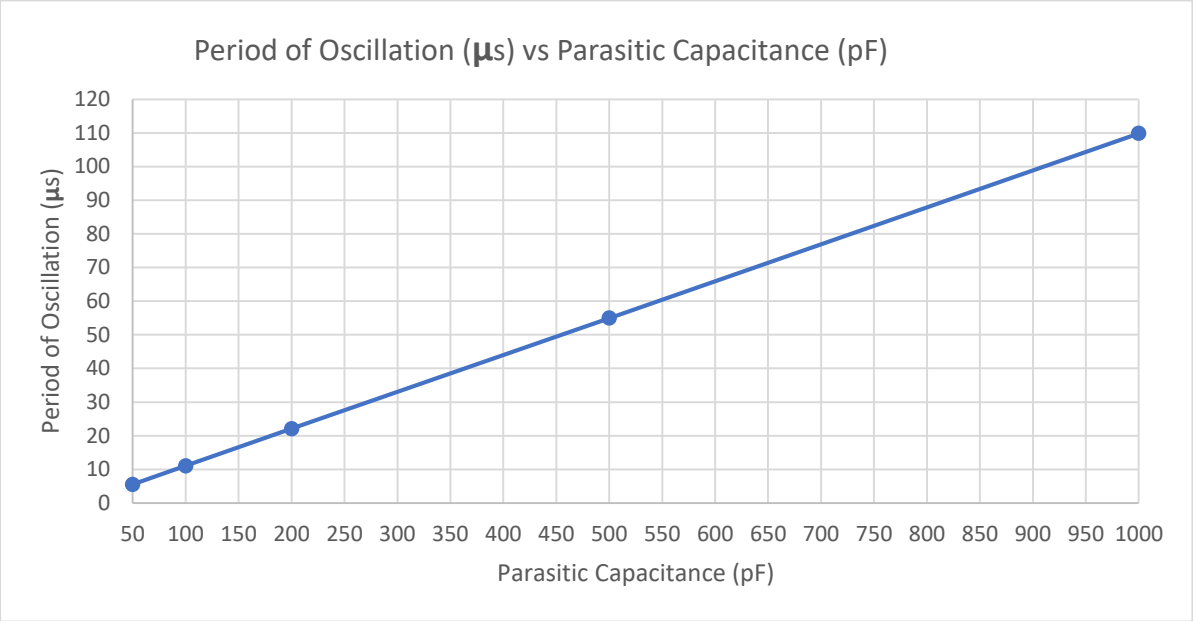

Figure 8 - Period of oscillation vs parasitic capacitance

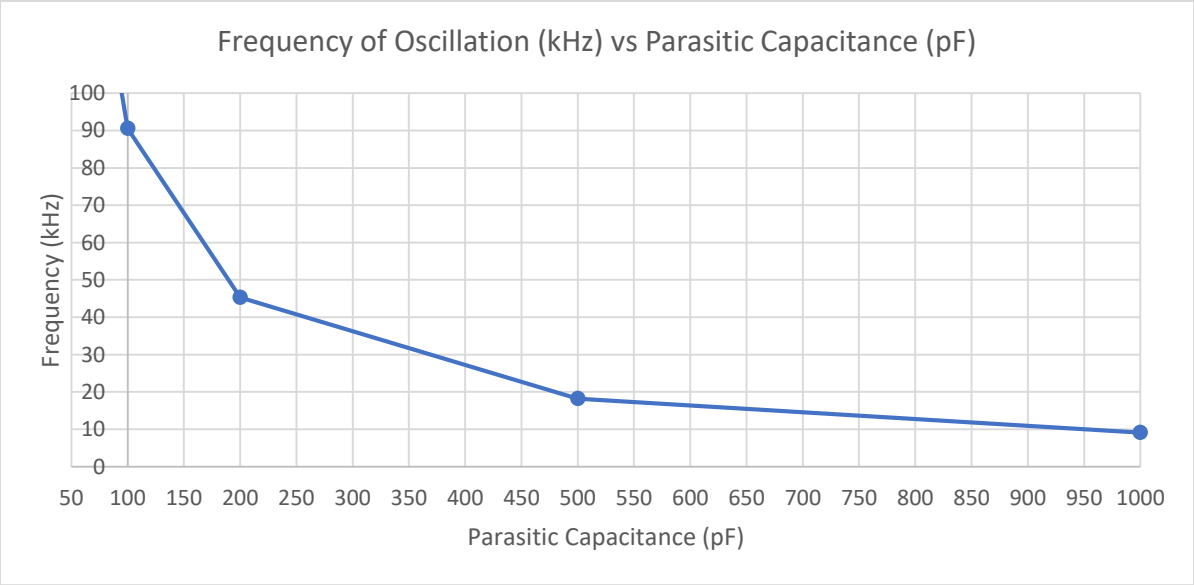

Figure 9 - Frequency of oscillation vs parasitic capacitance

- Other factors that affect the period of oscillation are the supply voltage and the temperature.
- The period of oscillation is inversely proportional to the supply voltage. Thus for a certain application, if we want to reduce the period of oscillation, we can either decrease the no. of stages in the ring oscillator or increase the supply voltage.
- Some common applications of ring oscillators are mentioned below.
    - Used during wafer testing to measure the effects of manufacture process variations.
    - Jitter in ring oscillators are used in hardware random number generators.
    - Used to measure the effect of voltage and temperature on an IC.

Circuit 02 – PLD

1. Programmable logic block for NAND and NOR

- Let's configure the logic block in a way that it acts as a NOR gate when the selection bit is 0 (low) and as a NAND gate when the selection bit is 1 (high).
- The corresponding truth table is given below.

Table 2 - Truth table for NAND and NOR

	Selection bit (S)	Input 1 (A)	Input 2 (B)	Output (Y)
NOR	0	0	0	1
	0	0	1	0
	0	1	0	0
	0	1	1	0
NAND	1	0	0	1
	1	0	1	1
	1	1	0	1
	1	1	1	0

- We can use a Karnaugh map to derive a simplified expression for the output.

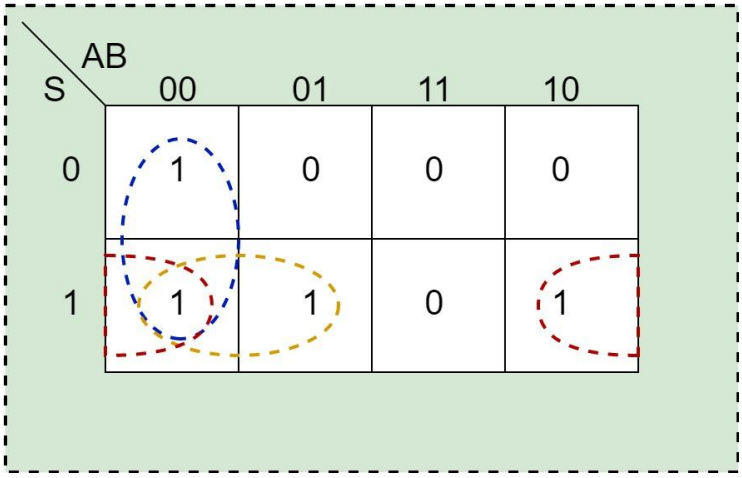

Figure 10 - Karnaugh map

- Thus, the output can be expressed using the following function.

$$\boxed{Y = \overline{A}.\overline{B} + \overline{A}.S + \overline{B}.S}$$

- First, we will implement this function using NOT, AND and OR gates. But, in physical devices, we usually encounter NOT, NAND and NOR gates implemented using PMOS and NMOS transistors. Therefore, we will implement the function entirely using NAND and NOT gates (using PMOS and NMOS) subsequently.
- The schematic for the basic implementation is given below.

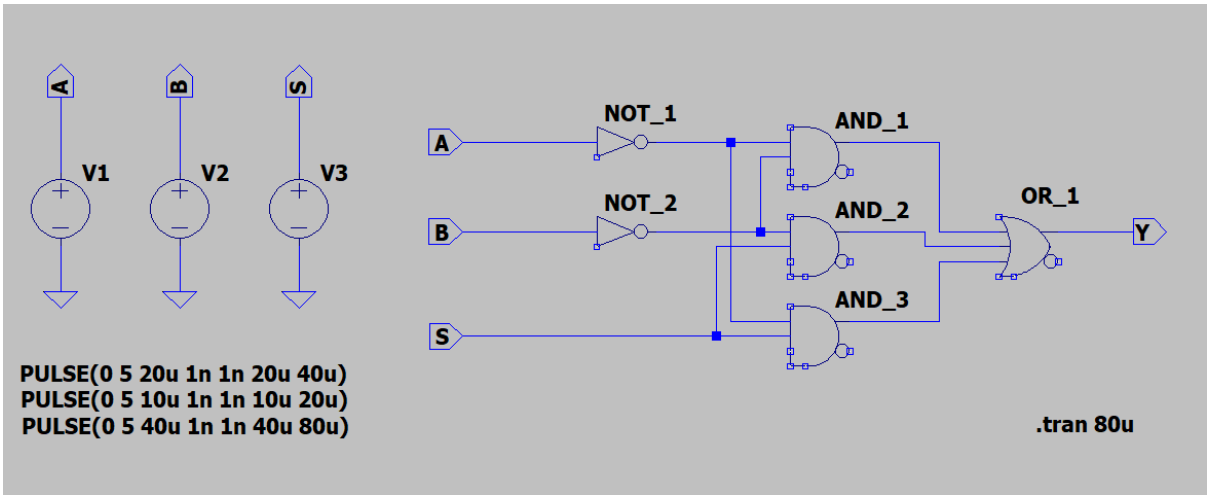

Figure 11 - Basic implementation of the combinational circuit

- We can implement the AND gates and OR gates using NAND and NOT gates as shown below.

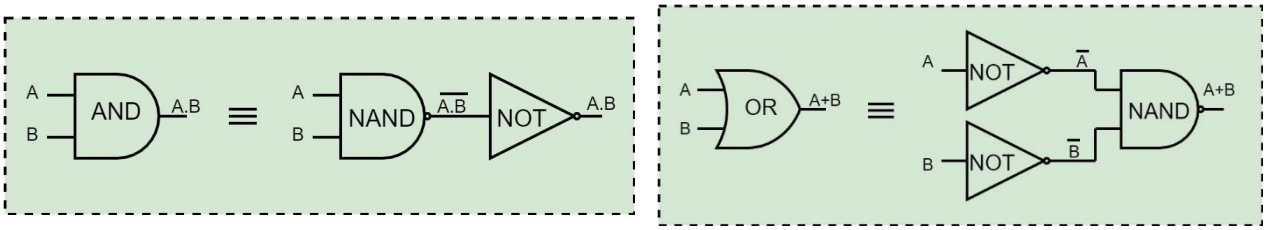

Figure 12 - NAND and NOT gate implementations of AND and OR gates

- PMOS/NMOS implementation of the NOT gate was mentioned under circuit 01. Given below are the PMOS/NMOS implementations of the 2-input NAND gate and the 3-input NAND gate.

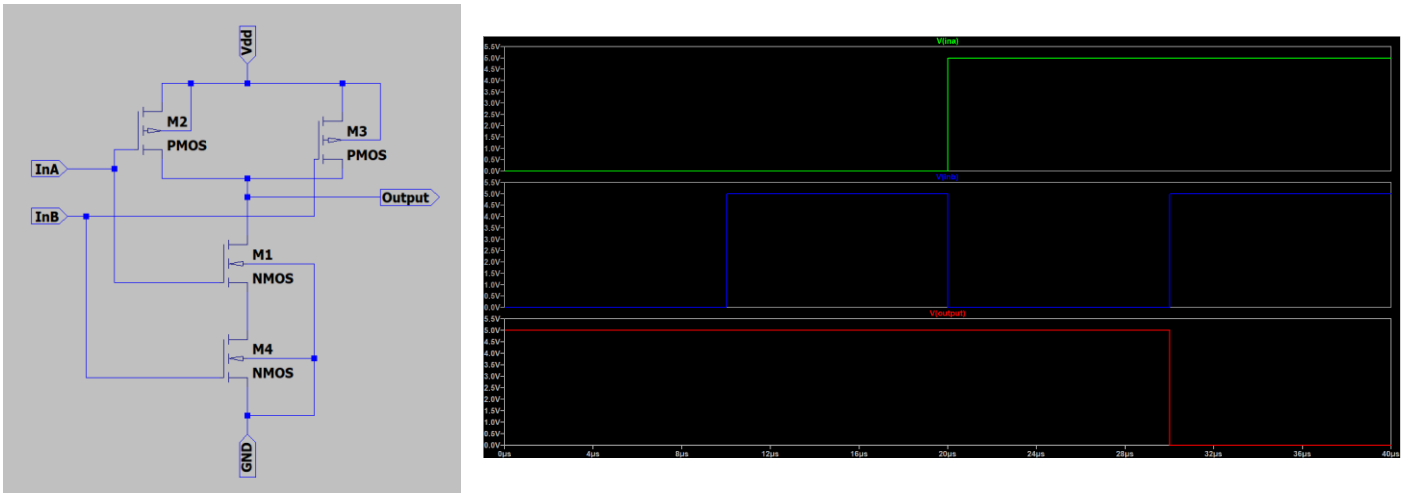

Figure 13 - PMOS/NMOS implementation of a 2-input NAND gate

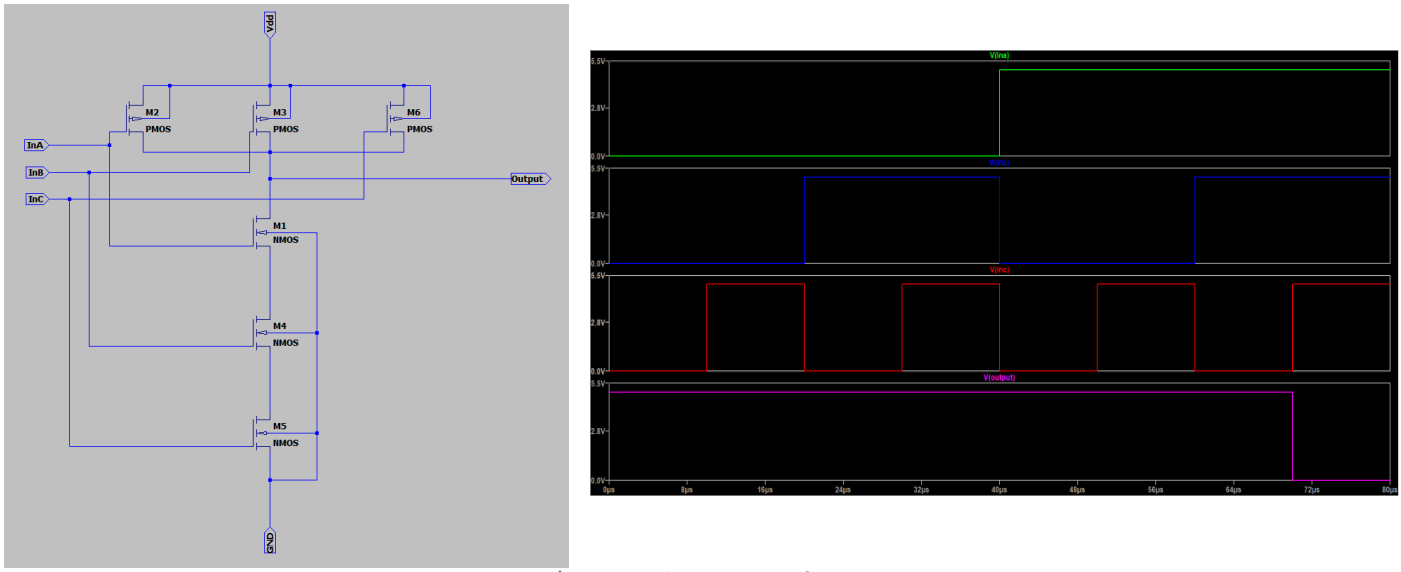

Figure 14 - PMOS/NMOS implementation of a 3-input NAND gate

- Using the above schematics for NOT and NAND gates, we can design the programmable logic block that can be configured as a NAND or a NOR gate, as shown below.

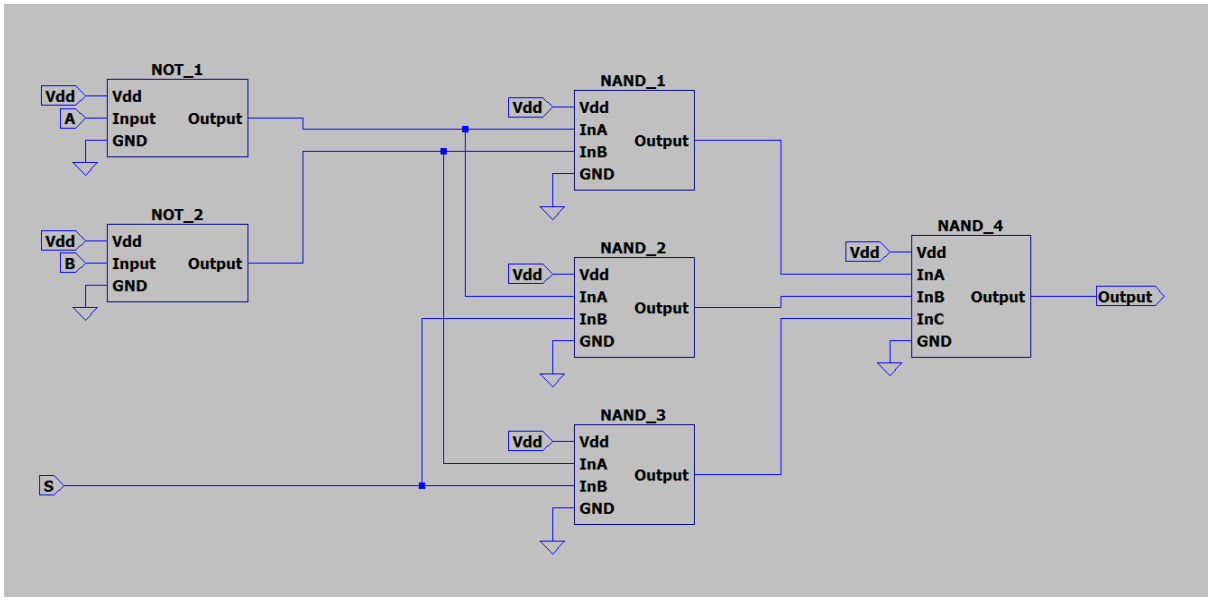

Figure 15 - Schematic for the programmable logic block

- We can verify the above schematic by passing three pulses as inputs to the above circuit.
- From the waveforms given below, we can observe that the programmable logic block gives out desired outputs for all possible combinations.

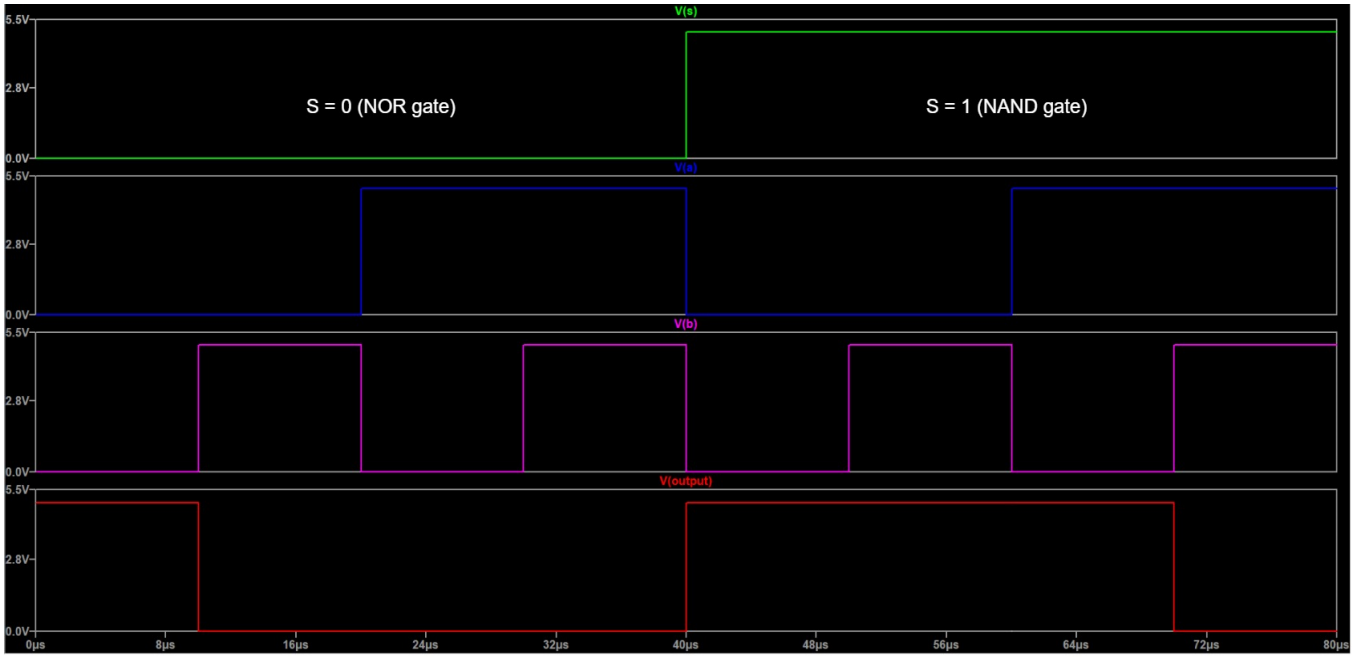

Figure 16 - Output waveform (red wave) of the programmable logic block

- The same setup can be used to observe hazards such as static-one and static-zero by adjusting t_{rise} and t_{fall} values of the input pulses A and B.
- Static-zero hazard can be observed when S = 0 (NOR gate) and static-one hazard can be observed when S = 1 (NAND gate).

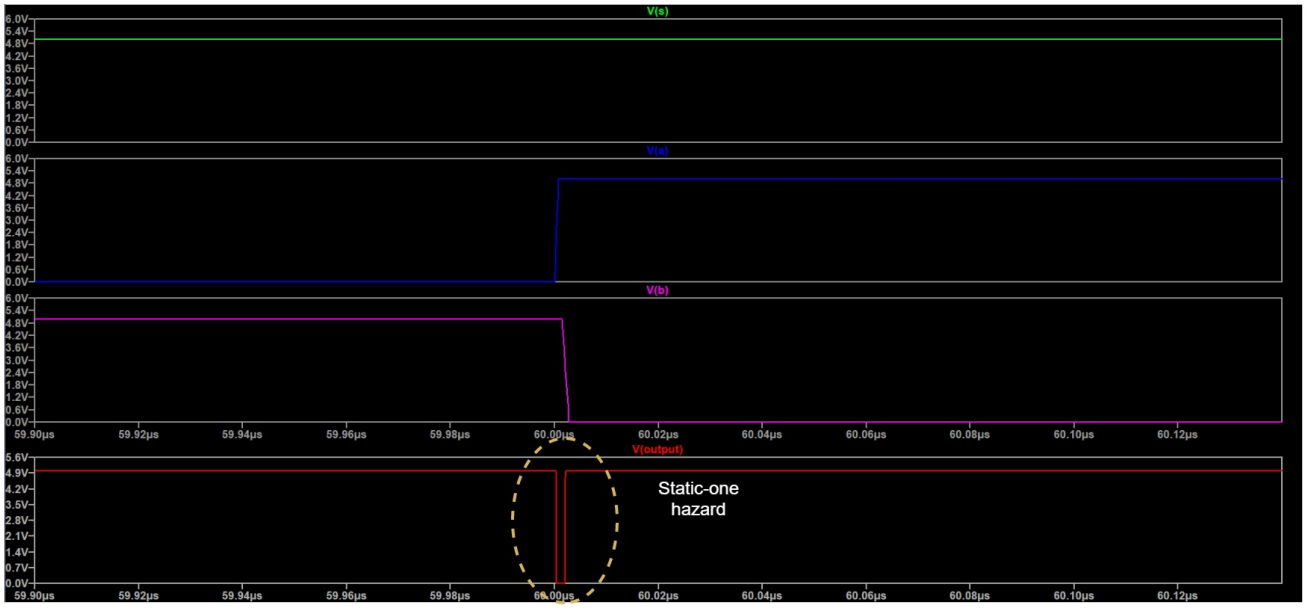

Figure 17 - Static-one hazard

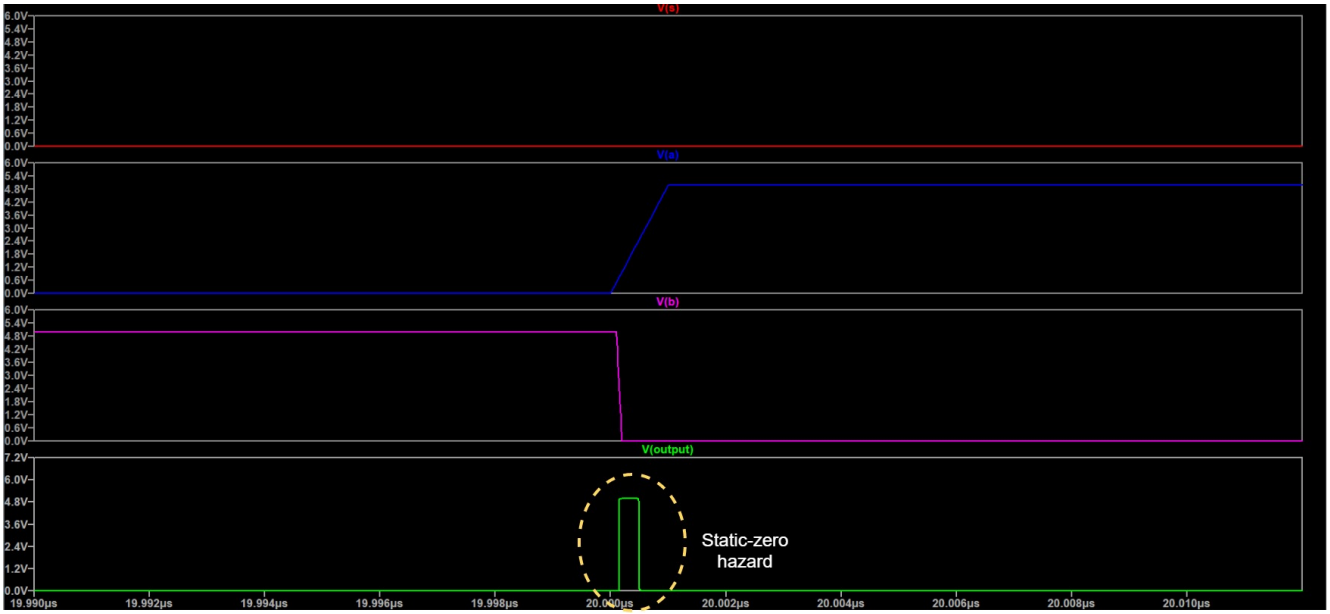

Figure 18 - Static-zero hazard

2. Single switch matrix using six pass transistors

- A FPGA consists of 3 building blocks.
  - Configurable logic blocks (CLBs)
  - Input/output blocks (IOBs)
  - Interconnections
- Programmable interconnections control the routing between CLBs allowing the signals to flow from one block to another, appropriately.
- The connections between CLBs, IOBs are regulated by the programmable switches and these switches are grouped together as the programmable switch matrix.
- The programmable switches inside the PSM are also called programmable interconnect points and these are pass transistors.
- It is possible to use NMOS transistors as PTL (pass-transistor logic) switches.

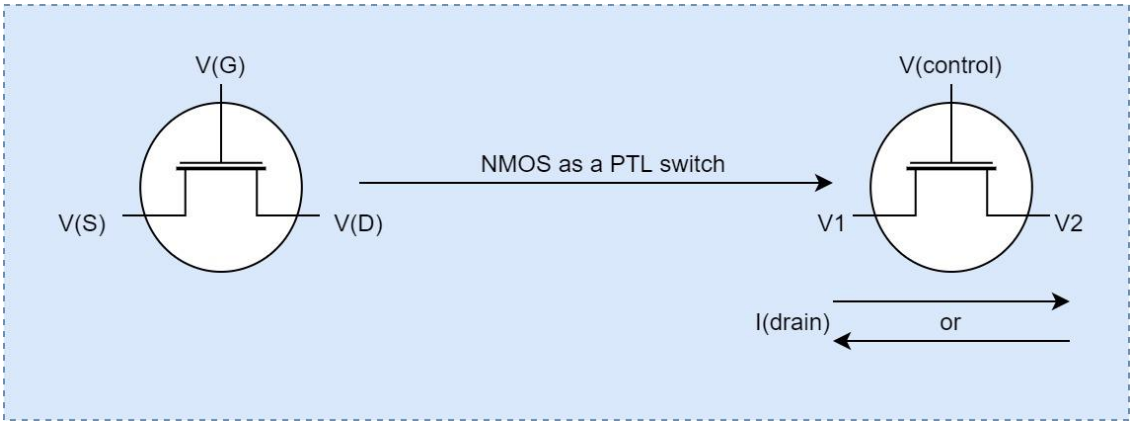

Figure 19 - NMOS as a PTL switch

- The NMOS will act as a closed switch when voltage applied to the gate is high and vice versa. Sufficiently large V_{GS} will allow a current to flow between S and D terminals and in this sense, NMOS acts as a voltage controlled switch.

- When the body or the substrate of the NMOS is grounded, the NMOS will be symmetric and hence, the source and drain terminals can be used interchangeably depending on the voltages (potentials) at those two terminals.
- Suppose $V_G = V_{DD}$. If $V_S = 0$, then $V_{GS} = V_{DD} > V_{th}$. Thus the MOSFET is turned on and as a result, the voltage at the source terminal would be passed to the drain terminal.
- Suppose $V_G = V_{DD}$. If $V_S = V_{DD}$, then $V_{GS} = 0 < V_{th}$. Thus the MOSFET is turned off. At this instance, the voltage at the source terminal wouldn't be passed to the drain terminal.
- The maximum voltage that can be passed through the NMOS would be $V_S = V_{DD} - V_{th}$.
- Hence, NMOS is considered to be a strong passer of logic 0 and a weak passer of logic 1.
- **Nevertheless, in a nearly ideal NMOS, $V_{th} \approx 0$. Therefore, such an NMOS will be able to pass signals (high and low) through it, as long as the gate voltage is high.**
- Using these principles, we have designed a single switch matrix consisting of six pass transistors (ideal transistors), as given below.

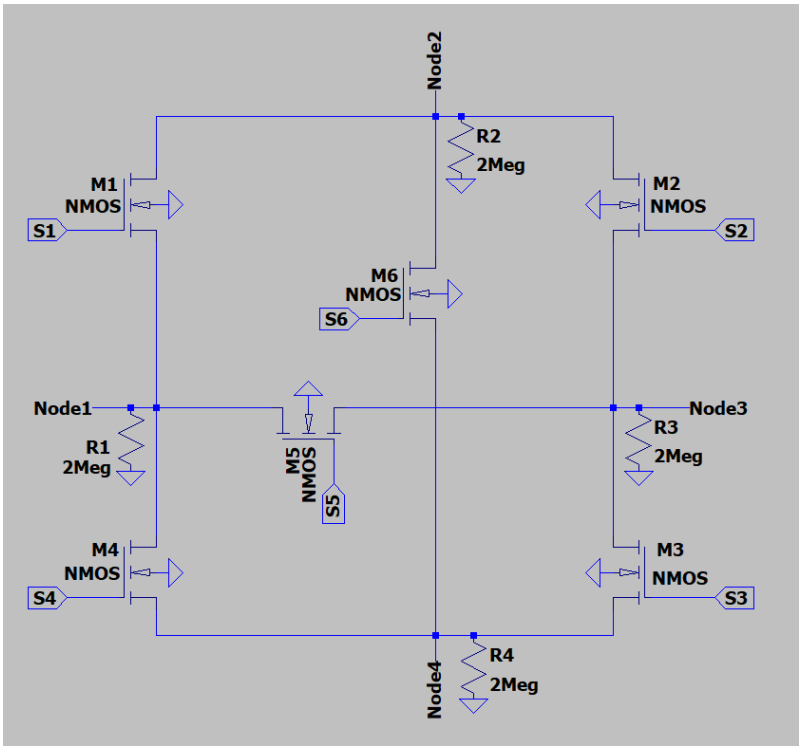

Figure 20 - Single switch matrix schematic

- Signals can arrive at any of the four nodes. But at a given instance, only one NMOS would be turned ON. i.e. only one path is active.
- For example, let's assume that a signal has reached node 1 of the above schematic. From there onwards, the signal could reach node2, node3 or node4 depending on the transistor that's switched on.
- Accordingly, if the signal is to be passed onto node2, then S1 should be logic 1 (high) while S2, S3, S4, S5 and S6 must be logic 0 (low).

Table 3 - Control bits for node-to-node connections

Connection	S1	S2	S3	S4	S5	S6
Node 1 and Node 2	1	0	0	0	0	0
Node 2 and Node 3	0	1	0	0	0	0
Node 3 and Node 4	0	0	1	0	0	0
Node 1 and Node 4	0	0	0	1	0	0
Node 1 and Node 3	0	0	0	0	1	0
Node 2 and Node 4	0	0	0	0	0	1

- Let's modify the schematic of the single switch matrix to simulate the transmission of a signal from node1 to any other node. (Note that we can use the same schematic to simulate the transmission of signals between any two nodes.)

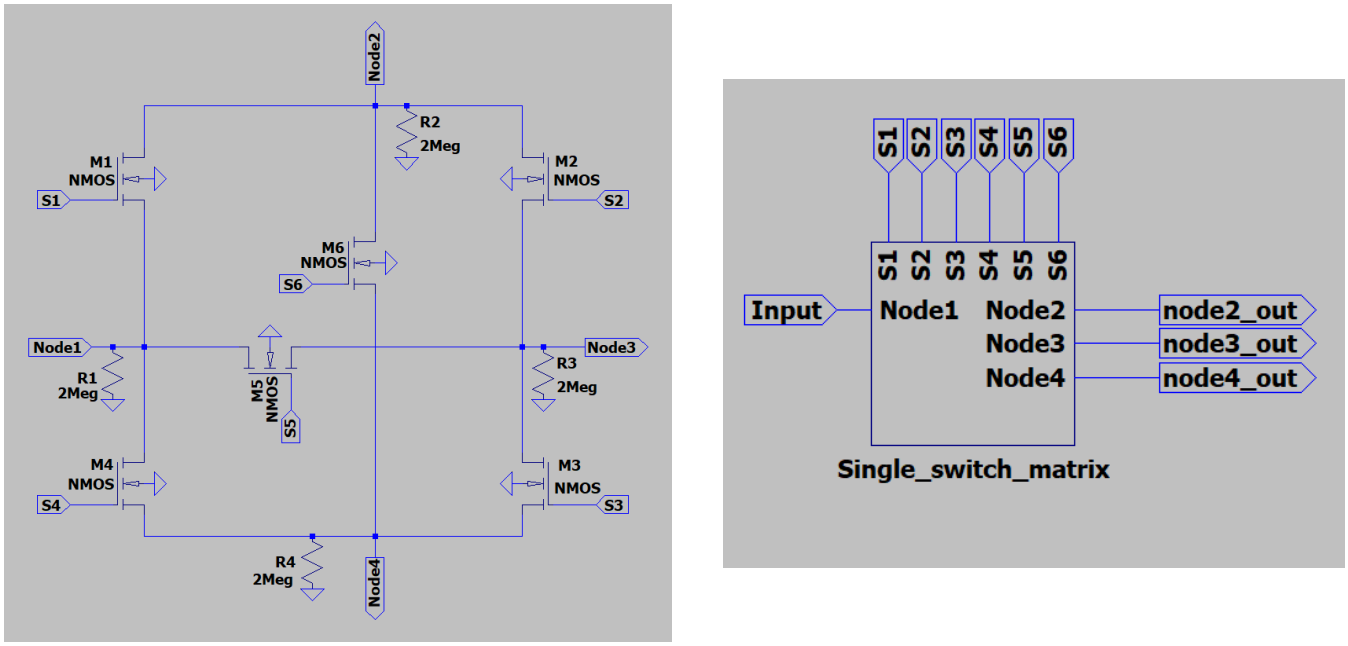

Figure 21 - Single switch matrix – modified

- We will use a constant DC voltage as the input at node 1. Then, 3 pulses will be given to S1, S4 and S5. Note that S2, S3 and S6 are maintained at logic 0 throughout the simulation.

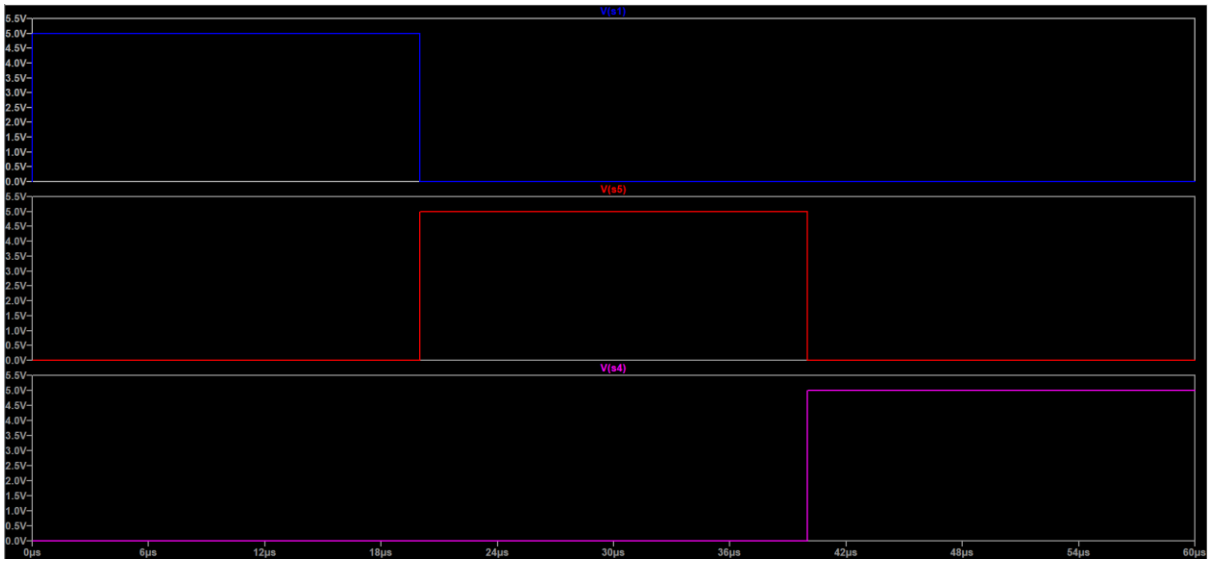

Figure 22 - Control pulses to S1, S5 and S4, respectively

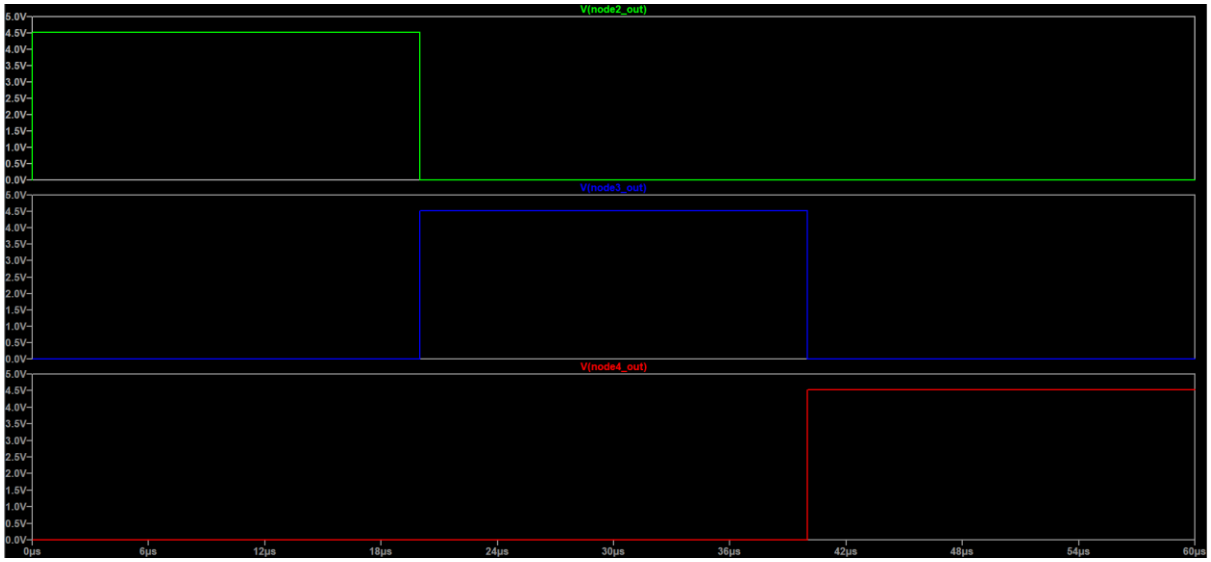

Figure 23 - Output waveforms at the nodes 2,3 and 4

- From the output waveforms shown above, we can clearly observe that,
  - the input is passed to node 2 when S1 is high
  - the input is passed to node 4 when S4 is high
  - the input is passed to node 3 when S5 is high

3. Designing a 3-input PLD

- In designing a 3-input PLD that can implement a combinational logic circuit, there are many approaches that can be used.
- For example, **P**rogrammable **R**ead **O**nly **M**emory (PROM), **P**rogrammable **L**ogic **A**rray (PLA) and **P**rogrammable **A**rray **L**ogic (PAL) are all viable options.
- Out of these, we decided to implement a PROM structure considering design feasibility.

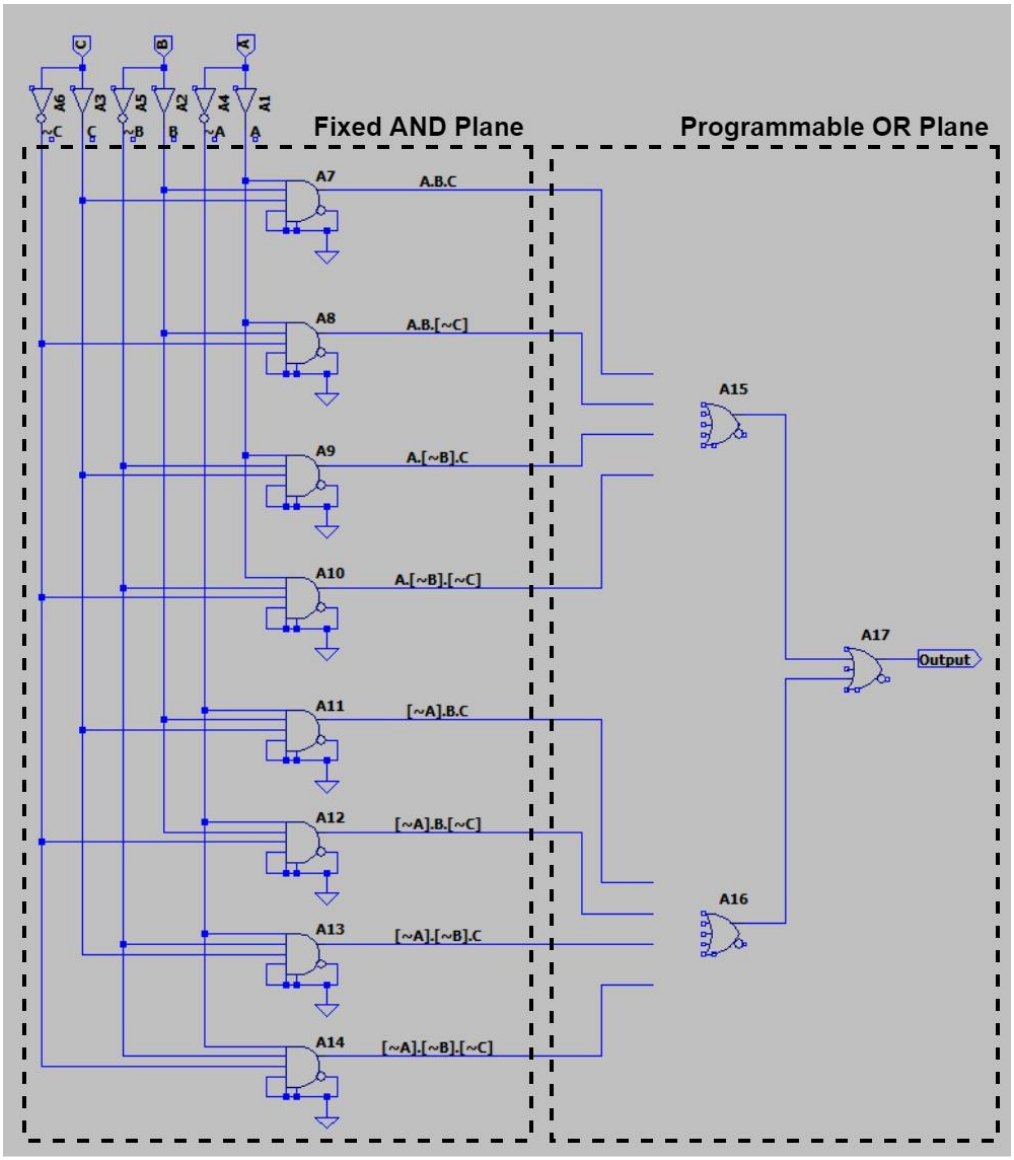

Figure 24 - SPLD – PROM

- In this report, we will be selecting the '**Full Adder Circuit**' as an example circuit to show the implementation of the design, but this PROM structure can be reprogrammed into implementing any 3-input combinational logic circuit and this reprogram ability will be detailed in a later section of the report.

- Full-adder circuit logic:

Table 4 - Truth table for a full-adder

A	B	C_{in}	Sum (S)	C_{out}
1	1	1	1	1
1	1	0	0	1
1	0	1	0	1
1	0	0	1	0
0	1	1	0	1
0	1	0	1	0
0	0	1	1	0
0	0	0	0	0

3-input 2-output

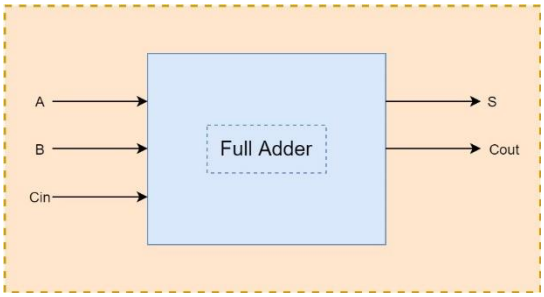

Figure 25 - Full-adder block diagram

❖ For convenience, let C_{in} = C.

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

$$C_{out} = A.B.C + A.B.\overline{C} + A.\overline{B}.C + \overline{A}.B.C \quad or \quad C_{out} = A.B + C.(A.\overline{B} + \overline{A}.B)$$

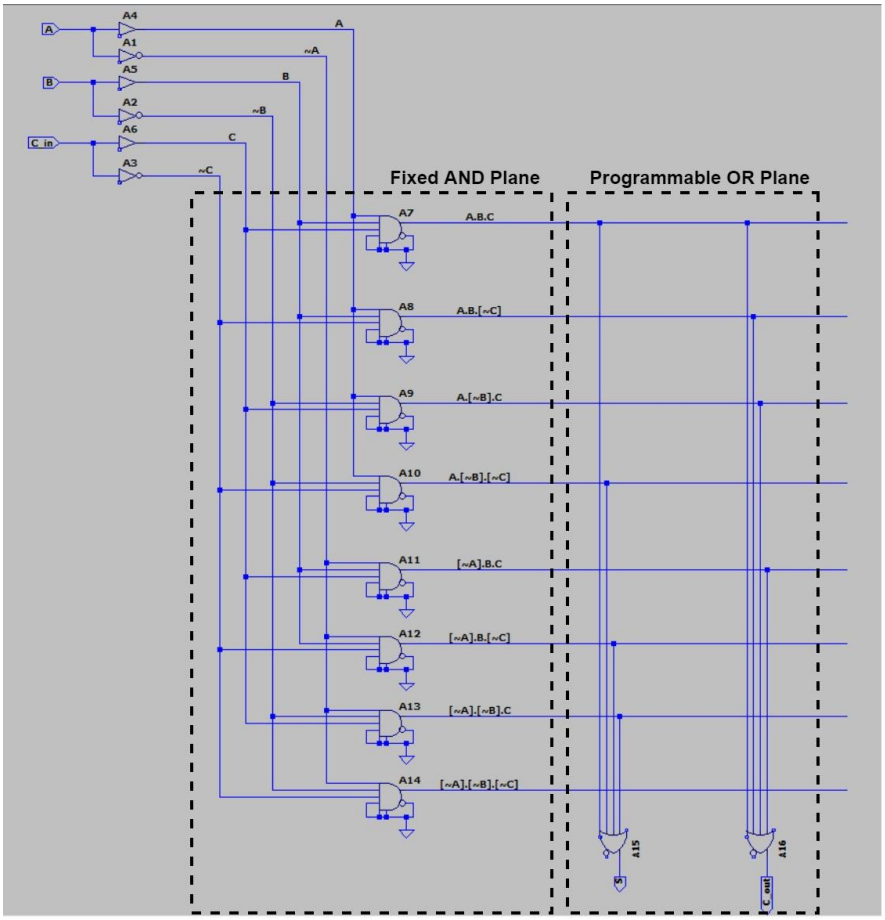

Figure 26 - Full-adder circuit schematic

- Implementation with NMOS transistors:
- We will be using a **decoder** and **NMOS switches** coupled with **pull-down resistors** in implementing this design.
- The 3 inputs A, B and C_{in} will be given as the inputs to the decoder. To maintain simplicity, we have omitted the enable pin in the decoder.
- Only one of the outputs from the decoder will be Logic HIGH at a given time, depending on the input combination.
- The circuit schematic showing the gate level implementation of the decoder completed with the NMOS switches and pull-down resistors is shown below.
- In the schematic shown below, we have used NMOS switches in the programmable OR plane to ease the simulation of a PROM.

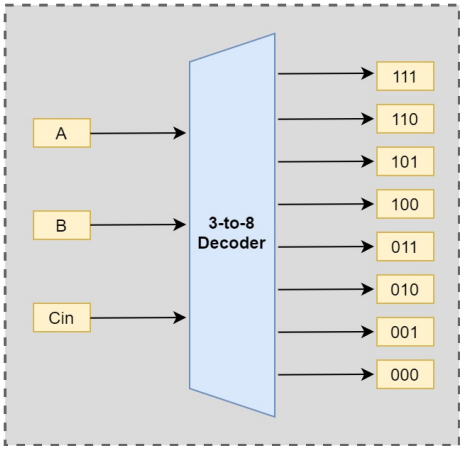

Figure 27 - 3-to-8 Decoder

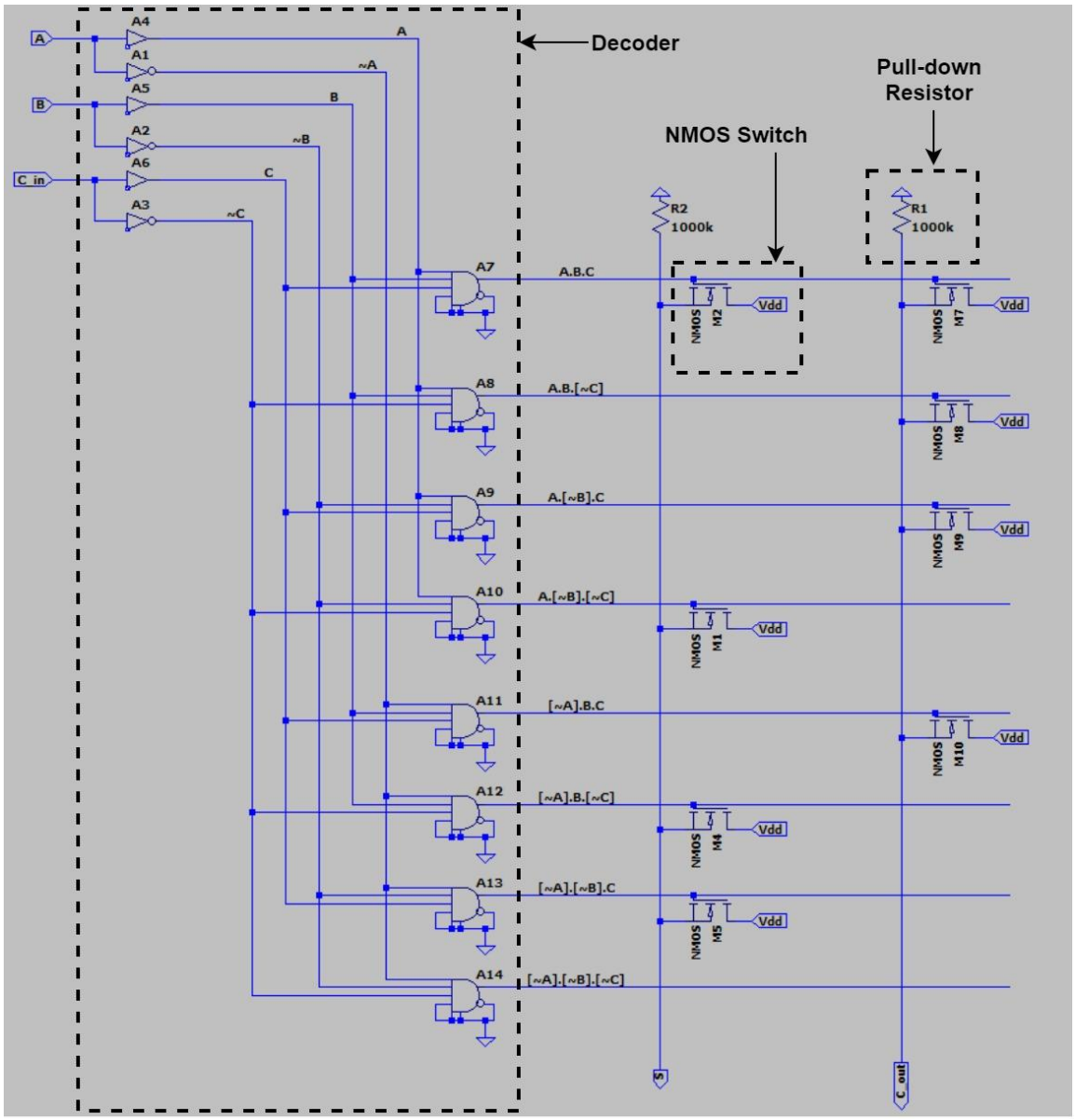

Figure 28 – Implementing the programmable OR plane using NMOS transistors

- Functionality:
- The decoder takes in 3 inputs as mentioned and decodes them into the 8 possible combinations.
- **Note that the NMOS switch setup is connected only to the combinations for which the respective output is to be made Logic HIGH.**
- Let's consider an example case.
- Assume the inputs are A = HIGH, B = LOW, C = LOW. Then the $A.\overline{B}.\overline{C}$ output will be asserted HIGH by the decoder.
- The NMOS transistor connected to the output terminal of A10 in the above schematic, is now behaving as a closed switch as V_{GS} is HIGH.
- Therefore, 'Sum' output is now connected to V_{dd} through the switch. Hence, the 'Sum' output is HIGH.
- There is no NMOS transistor connected to this decoder output and the C_{out} line. Therefore, C_{out} will not be connected to V_{dd}.
- To prevent the C_{out} from being a dangling output, we have used a pull-down resistor to ground this output. This avoids any possible complications.
- Similarly, the 'Sum' output will be asserted logic HIGH when the input combinations are $(A.B.C), (A.\overline{B}.\overline{C}), (\overline{A}.B.\overline{C})$ and $(\overline{A}.\overline{B}.C)$.
- The C_{out} output will be asserted logic HIGH when the input combinations are $(A.B.C), (A.B.\overline{C}), (A.\overline{B}.C)$ and $(\overline{A}.B.C)$.

- Re-programmability:
- We have demonstrated how to use PROM to implement the 3-input 2-output combinational circuit of a Full Adder.
- But this PROM can be used to implement any 03-input logic function by connecting NMOS switches to be driven by each decoder output as shown in the figure below.
- Depending on the function to be implemented, connections can be made between the relevant NMOS switches and the decoder outputs. This enables the implementation of any 3-input combinational logic circuit. (If a connection is to be made, the corresponding decoder output should be connected to the gate terminal of the NMOS).

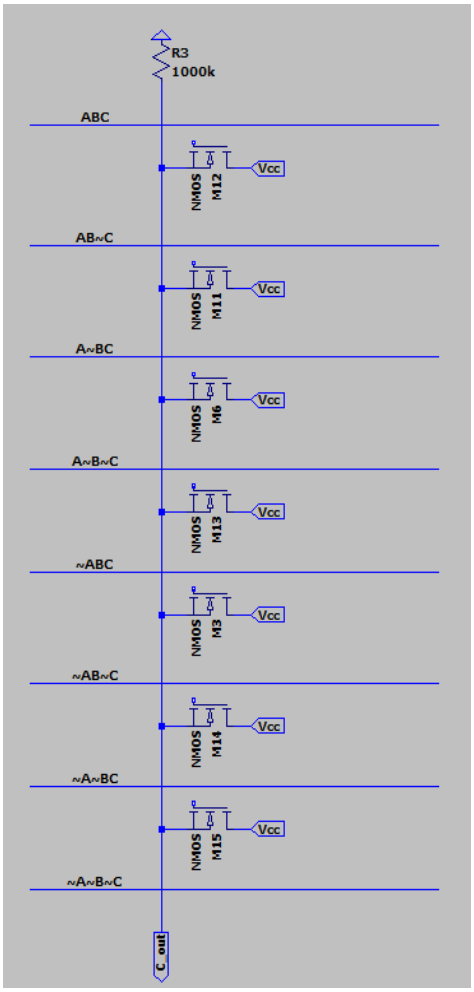

Figure 29 - Programmable output block

- Simulation:
- The following figure shows the variation of the **'Sum'** output for different input combinations.

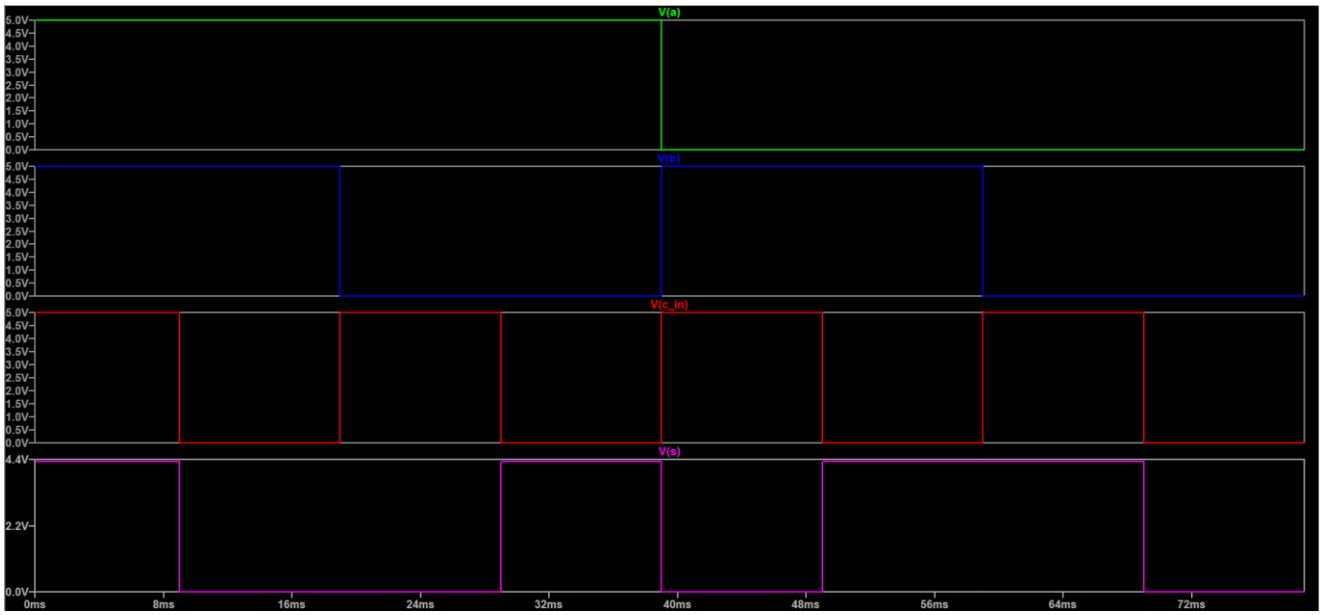

Figure 30 - 'Sum' output variation

- The variation of the '$\mathbf{C_{out}}$' output is as follows.

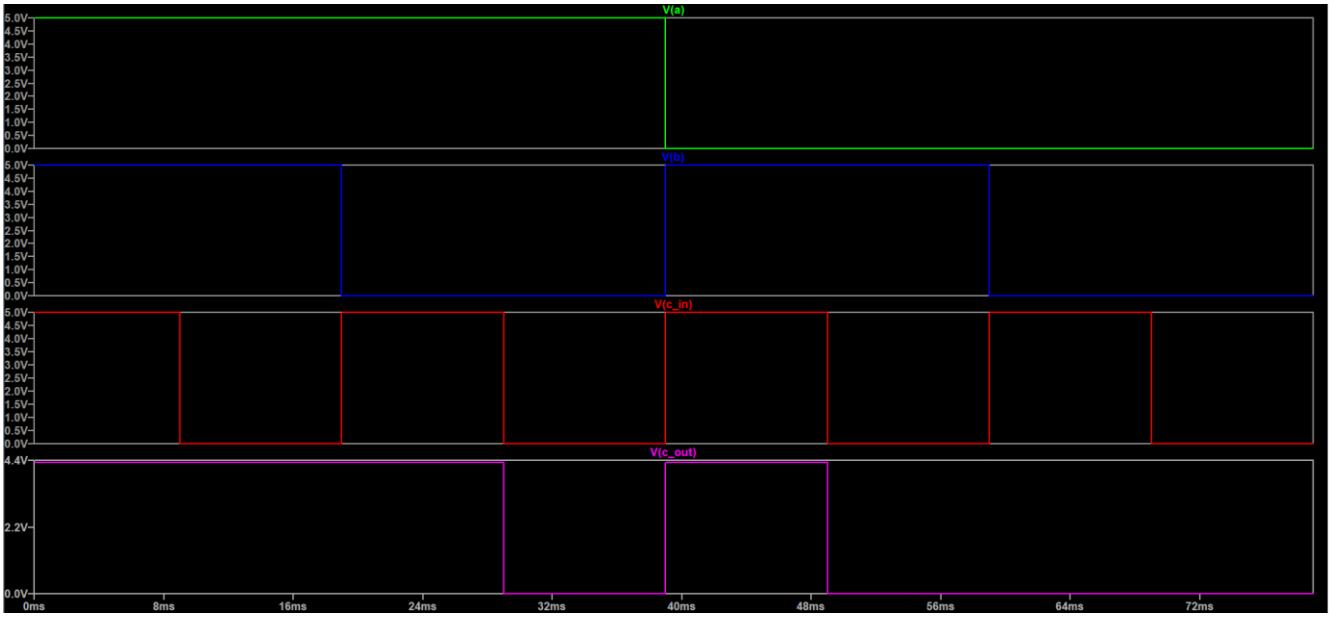

Figure 31 – C_{out} output variation

- From the above waveforms, we can conclude that the desired outputs are obtained for all possible input combinations.

Bibliography

(n.d.). Retrieved from https://www.electronics-tutorials.ws/transistor/tran_7.html

(n.d.). Retrieved from https://www.elprocus.com/mosfet-as-a-switch-circuit-diagram-free-circuits/#:~:text=P%2DChannel%20MOSFET&text=It%20is%20a%20four%2Dterminal,direction%20of%20positively%20charged%20holes.

Keim, R. (2016, August 12). Retrieved from https://www.allaboutcircuits.com/technical-articles/the-cmos-transmission-gate/

Mercer, D. (2019, September 25). Retrieved from https://wiki.analog.com/university/courses/alm1k/alm-lab-ring-osc