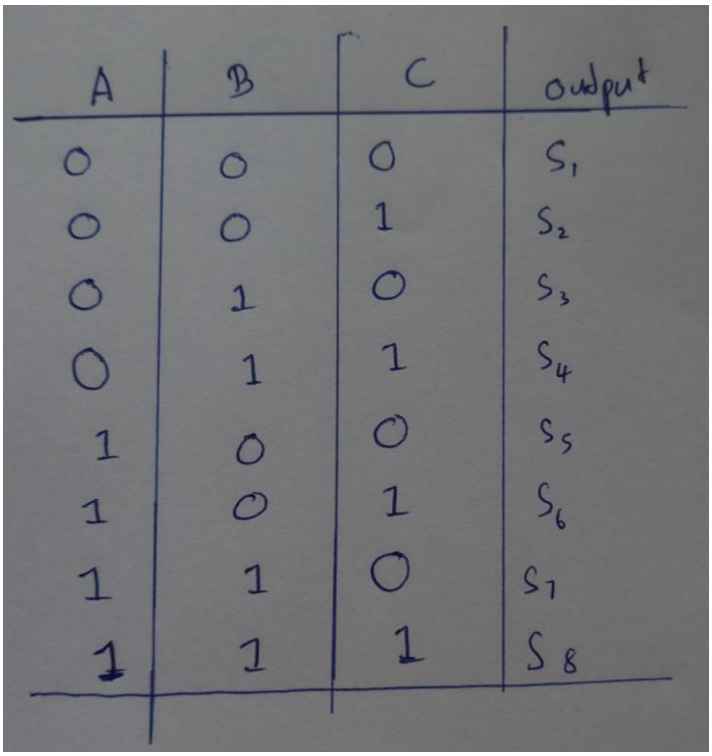
Circuit_2_Part_3

The task was to design a PLD circuit capable of implementing any three input combinational



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

circuit. The truth-table of any three input combinational circuits will be as below.

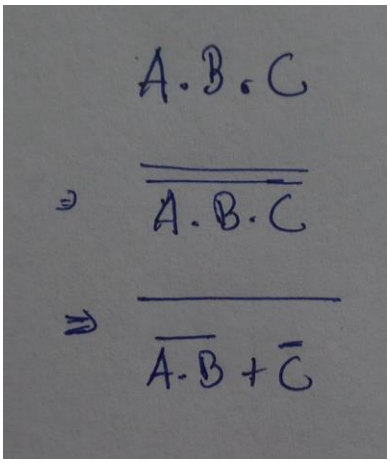
Figure 3.3.1

Truth-table of any three input combinational circuit

Outputs S1, S2,...., S8 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 Si 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, A', B, B', C, or 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.



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

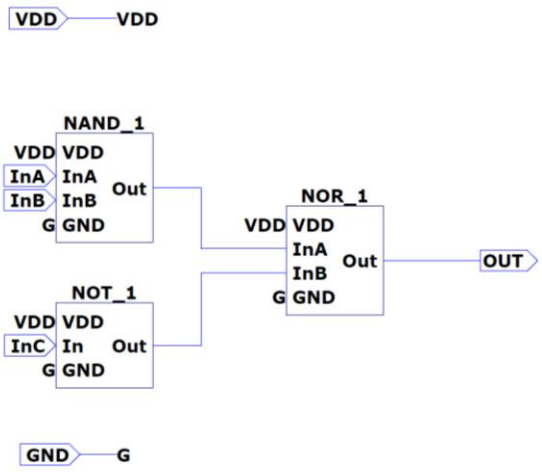
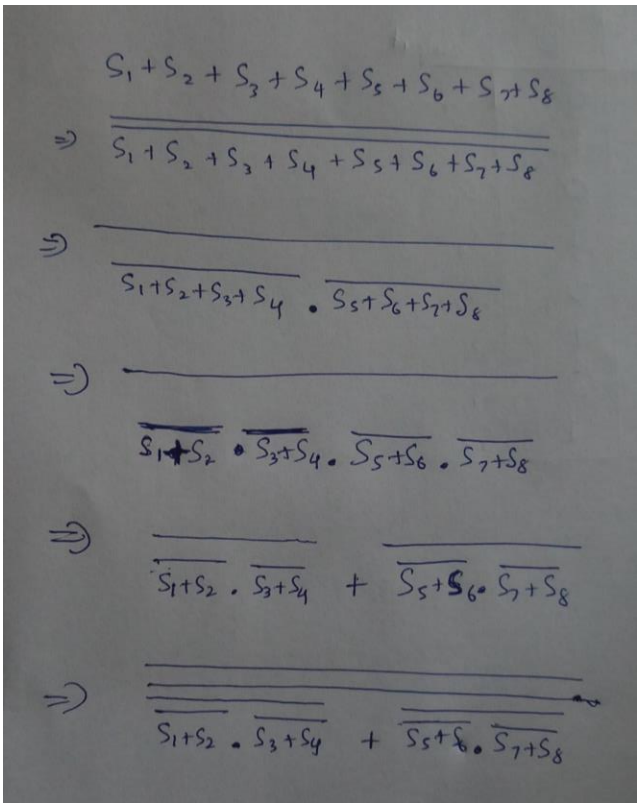


Figure 3.3.2

Implementing the three-input AND gate using NOR, AND, 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.



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

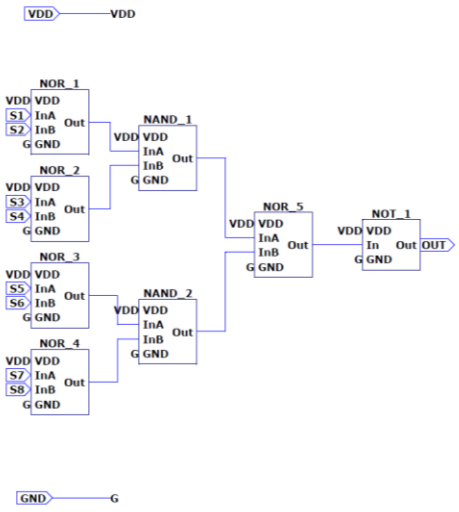


Figure 3.3.3

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

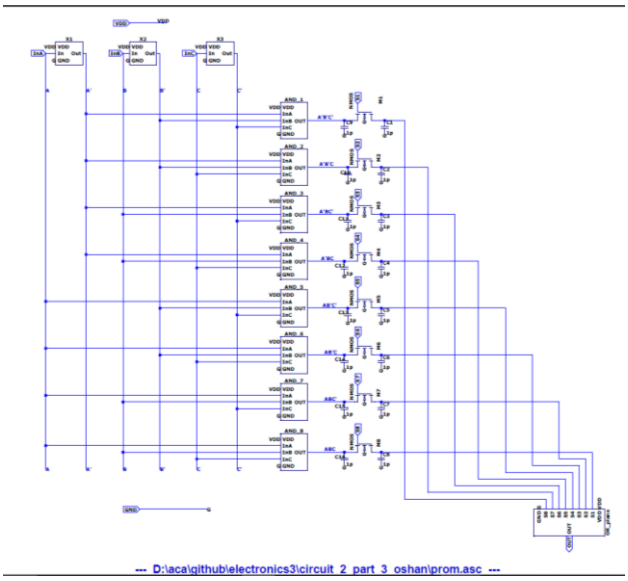


Figure 3.3.4

PROM circuit

We have used NMOS transistors as switches which choose, which min-terms are taken into the sum of products.

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

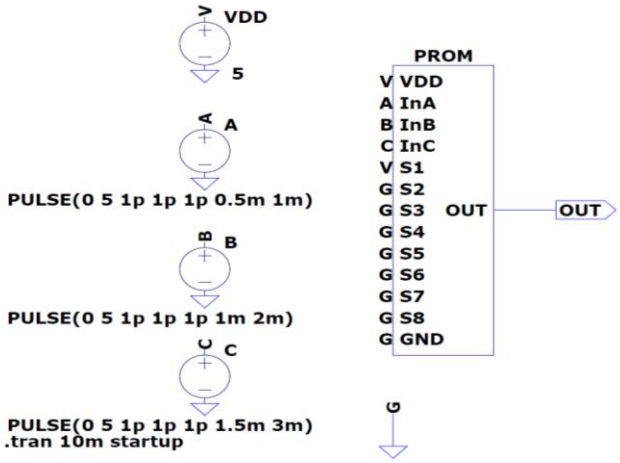


Figure 3.3.5

PROM configured as a NOR gate

Results were as below,

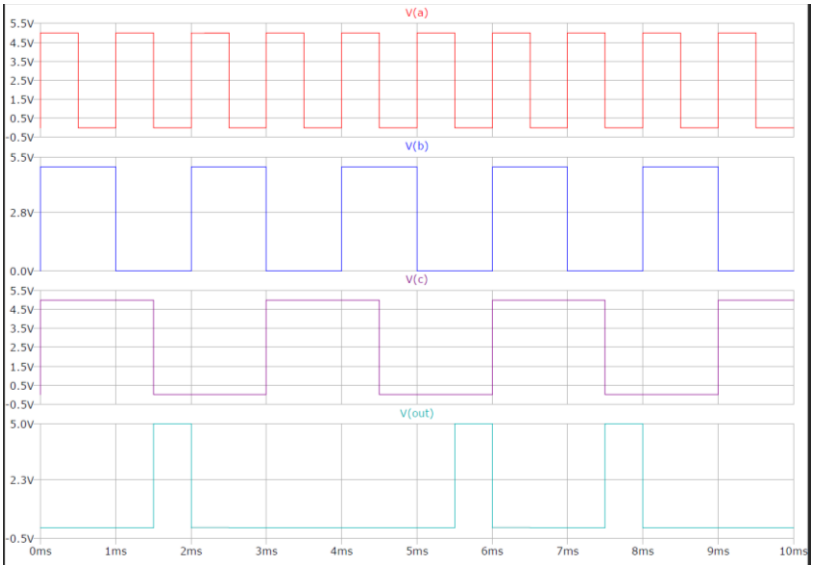


Figure 3.3.6
Results of PROM configured as a NOR gate

We can observe that the PROM is functioning correctly.

Discussion

One major problem we encountered when designing the PLD was, configuring the programmability of the OR plane. For the programmability we need configurable switches. Each switch decides the corresponding min-term get added to the final expression or not. Expected characteristics of the switches are,

1. Giving the corresponding signal when the switch is on
2. Giving binary 0 at the output, when the switch is off.

There are two possible ways for achieving this.

1. Using a 2 input AND gate with corresponding signal and switch bit as two inputs.
2. NMOS transistor used as a pass transistor with, switch bit given to the gate terminal and corresponding signal given to the source terminal.

Considering the power efficiency and latencies we concluded that using NMOS as a pass transistor is the best way to achieve our goal.
To achieve the second characteristic we expected, giving binary 0 at the output, when the switch is off, we have connected a large load at the output (drain terminal) of the switch as pull

down resistor. So, when transistor is at the high impedance state it will be grounded through the resistor achieving binary 0.