Challenge Problem 1

AVVARU BHARAT

Download all latex-tikz codes from

https://github.com/Bharat437/EE5803−FPGA−LAB/tree/main/Challenge−1

1 Problem

Obtain and implement an algorithm to convert any truth table to NAND logic.

2 Algorithm

1. From Truth table we will have inputs array and ouput array. Implement a for loop along the length of the ouput array.
2. Inside for loop, when output array will be equal to 1 for that index read the inputs from input array.
3. If input is 0 then take Complement of that input and if input is 1 take Normal input. Like this form all min terms.
4. Take complement of each min term which is nothing but a NAND gate operation. These are the first level of NAND gates.
5. Now pass all the outputs of first level of NAND gates through another NAND gate and obtain required output.

3 Explanation

For example the given truth table is as below

X	Y	Z	G(X,Y,Z)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Table 1: Given Truth table

From truth table, the min terms for which $G = 1$ are $2\left(\overline{X}.Y.\overline{Z}\right)$, $4\left(X.\overline{Y}.\overline{Z}\right)$, $5\left(X.\overline{Y}.Z\right)$, $7(X.Y.Z)$.
Now we can draw the logic circuit using NAND gates as below.

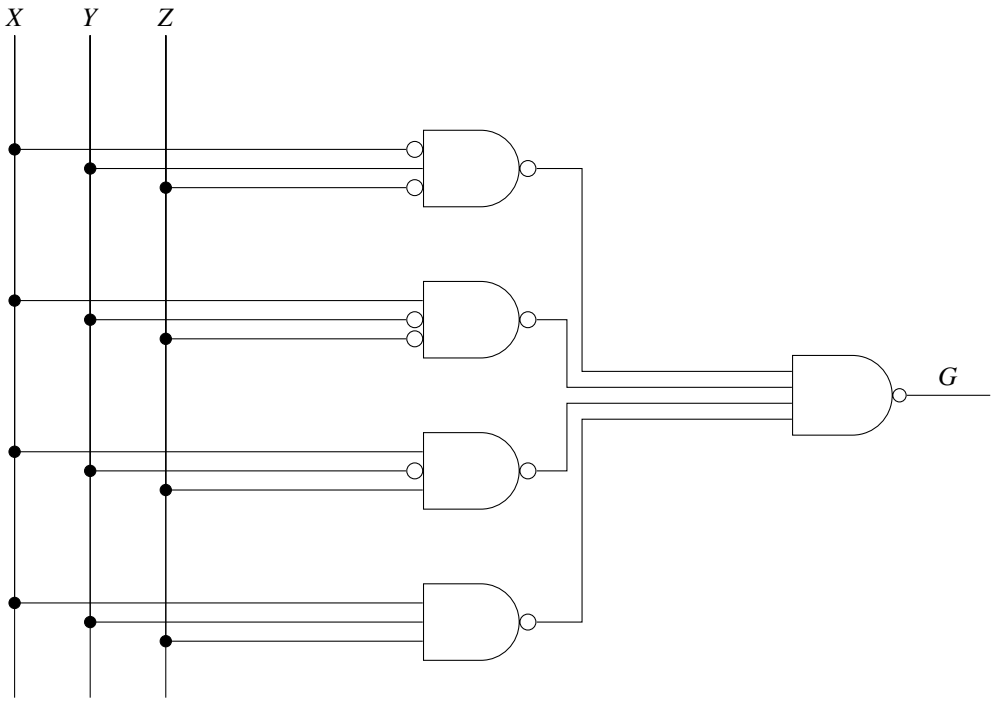

Figure 1: Logic Circuit using NAND gates

The same using AND-OR logic i.e. SOP form is drawn as shown below.

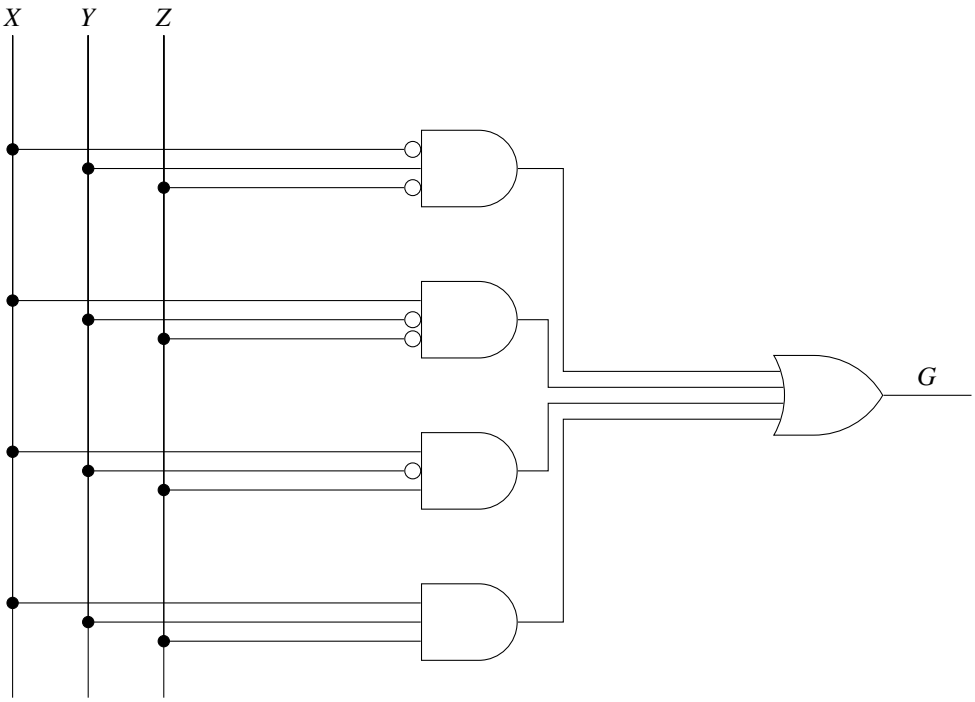

Figure 2: Logic Circuit in SOP form

4 Conclusion

From the above circuit figures 1 and 2, we can say that SOP form i.e. AND-OR logic is equivalent to NAND-NAND logic. The Verification is also done using a c code.