SHAID SMARAK COLLEGE

Kirtipur, Kathmandu

Lab no: 2 of Digital logics

Submitted by :-

1st semester

Amir Maharjan

Submitted to :-

Himal Raj Gental

LAB 2: NAND & NOR IMPLEMENTATION

Objective:

- To learn how to implement NAND and NOR.

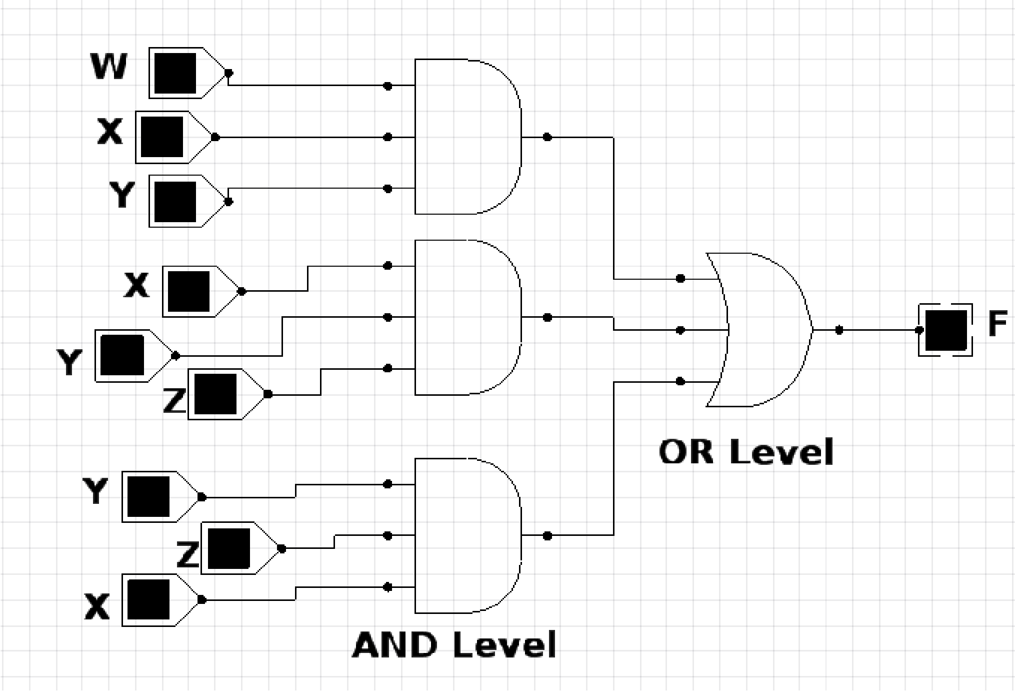
Discussion:

Digital circuits are more frequently constructed with NAND or NOR gates than AND & OR. NAND & NOR gates are easier to fabricate with electronic components and are the basic gates used in all IC digital logic families.

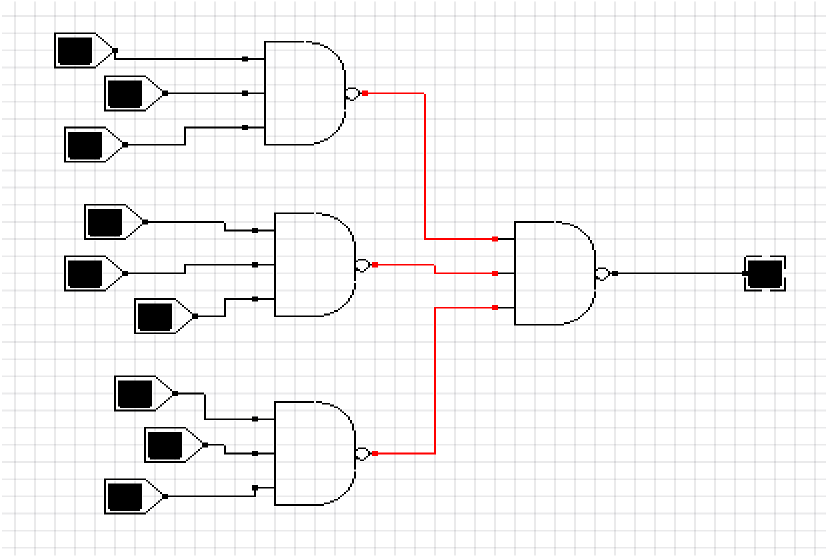
- **NAND Implementation**

Any logic function can be implemented using NAND gates. To achieve this, first the logic function has to be written in SOP form. Once the function is in SOP, then it is very easy to implement NAND gate. In other words, any logic circuit with AND gate in first level or OR in second level can be converted into a NAND-NAND gate circuit. The diagrams below are equivalent and implementation of the function:

F = AB + CD + E



AND-OR



NAND-NAND

The rule for obtaining the NAND logic diagram from a Boolean function:

First Method

1. Simplify the function and express SOP.
2. Draw NAND gate for each product term that has at least two literals.
3. Draw a single NAND gate in the second level, with inputs coming from outputs of 1st level gates.
4. A term with a single literal requires an inverter in the 1st level and applied as an input to the 2nd level NAND gate.

# Implement the following function with NAND

$$F\ (X, Y, Z) = \sum (0,6)$$

The first step is to simplify the function with a map:

X \ YZ	00	01	11	10
0	1	0	0	0
1	0	0	0	1

Fig 1: K-map

The map can't be combined.

The function we obtained from the map are as:

$$F = X'Y'Z' + XYZ$$

$$F = X'Y + XY' + Z$$

The two-level NAND implementation using the first method is given below:

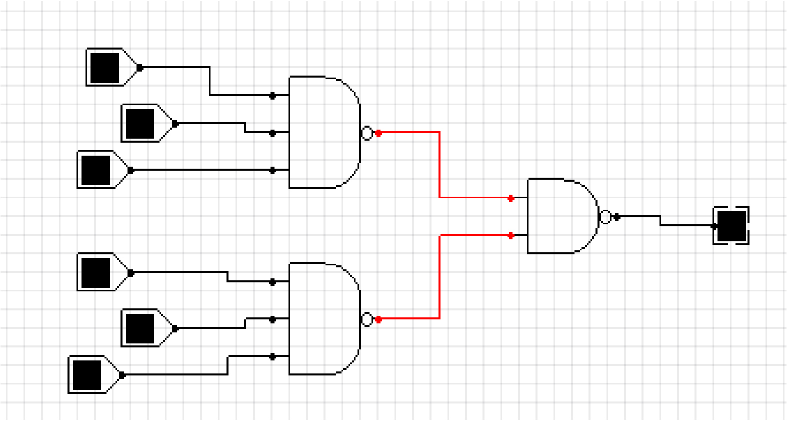


Fig: F= X'Y'Z' + XYZ'

Second Method

If we combine 0's in the map from fig 1, we obtain the simplified expression of SOP. The compliment of the function can then be implemented with two level of NAND gates using the rules stated above. If the normal output is desired, it would be necessary to insert a one-input NAND or inverter gate.

# Implement the following function with NAND

F (X, Y, Z) = ∑ (0,6)

Simplify the complement of the function in SOP. This is done by combining the o's in the map from above (Fig 1). F = X'Y + XY' + Z.

The two-level NAND gate for generating F' is: -

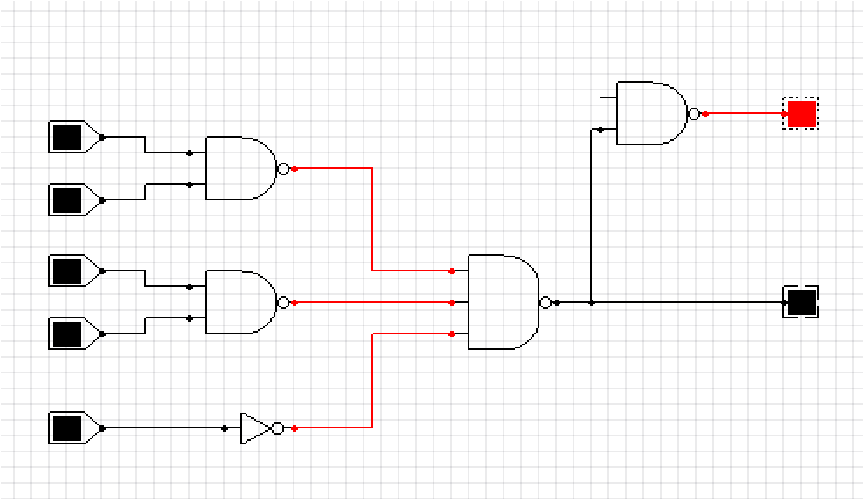
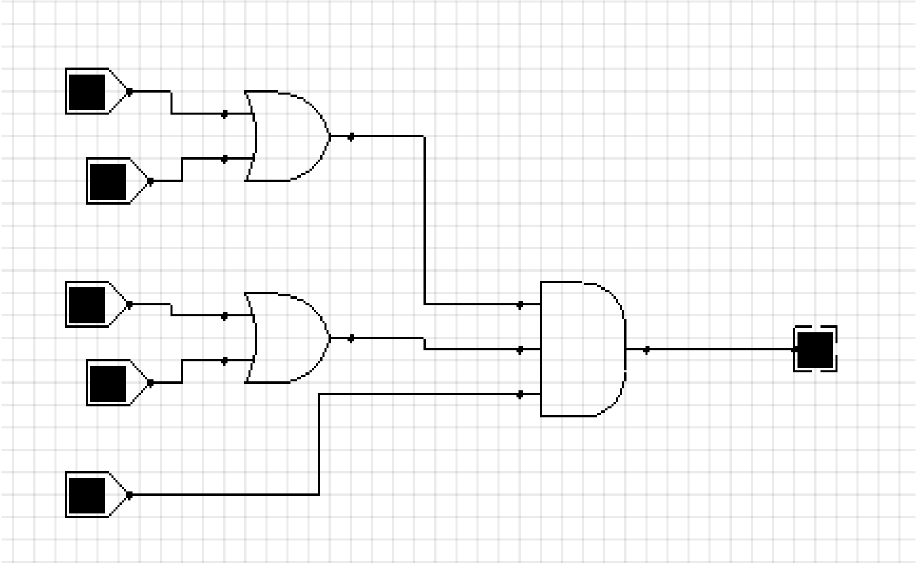


Fig: F = X'Y + XY' + Z

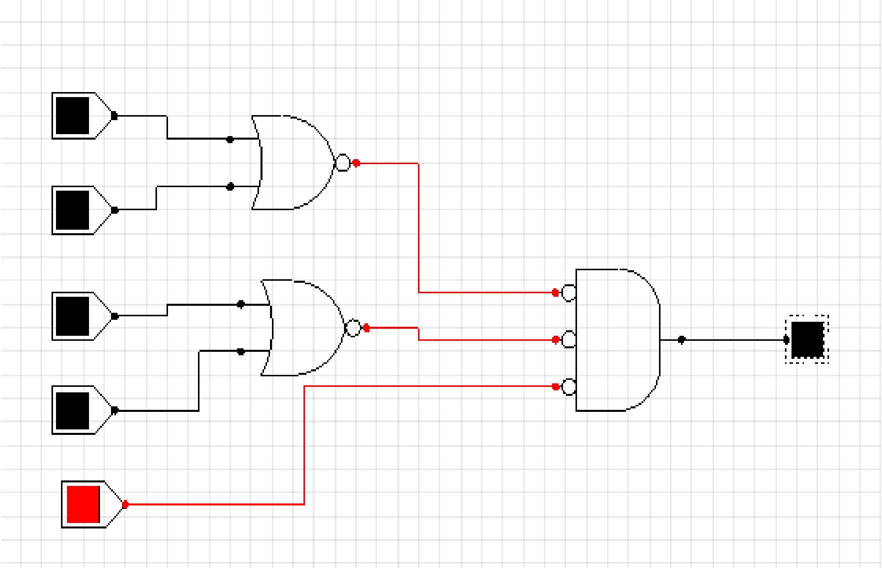
If output F is required, it is necessary to add a one input NAND gate to invert the function. This gives a three-level implementation.

- **NOR Implementation**

In order to implement the NOR gate the Boolean expression has to be in Product of Sum (POS) form. In POS form, the first level of the gate is OR gate and second level of the gate is AND gate. First, you need to have a simplified POS expression for the function you need to implement. Simplified POS expression can be made using k-map by combining the 0's and then inverting the output function.



OR-AND



NOR-NOR

All the rules for NOR implementation are similar to NAND except that these are dual.

# Implement the following function with NOR

F (X, Y, Z) = ∑ (0,6)

X YZ	00	01	11	10
0	1	0	0	0
1	0	0	0	1

Method → 1

First, combine the o’s in the map to obtain F = X’Y + XY’ + Z this is the complement of the function in SOP. Complement F’ to obtain the simplified function in POS as required for NOR implementation. The complement of the function -
F = (X + Y) (X’ + Y) Z’ is *F = (X + Y) (X’ + Y) Z’*

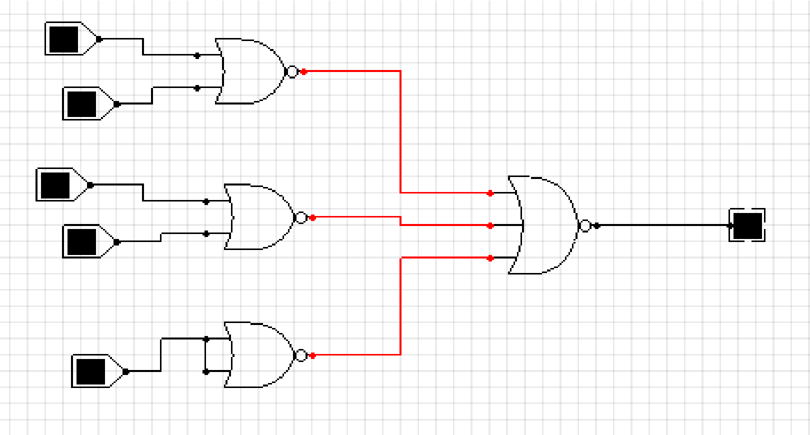


Fig: F = (X + Y) (X’ + Y) Z’

Method → 2

A second implementation is possible from the complement of the function in POS. For this case, first combine the 1’s in the k-map to obtain F = X’Y’Z’ + XYZ’. The complement of this function to obtain the complement of the function in POS as required for NOR implementation. F = (X + Y + Z) (X’ + Y’ + Z).

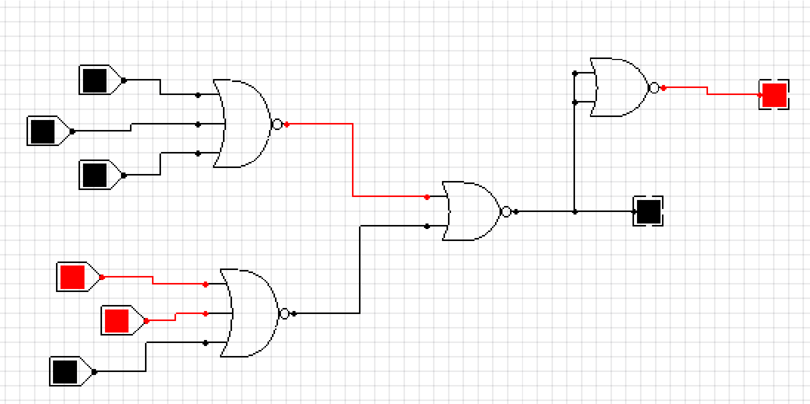


Fig: F = (X + Y + Z) (X’ + Y’ + Z)