

Laboratory 2**Boolean Expressions using Universal Gates****1. Introduction and Purpose of Experiment**

Students will learn to simulate and implement logic circuits using only universal gates.

2. Aim and Objectives

Aim: To simulate and implement logic circuits using only NAND and NOR gates

Objectives: At the end of this lab, the student will be able to

- Use Logisim to simulate Boolean circuits using only NAND gates
- Describe the procedure to convert all the gates in a circuit to universal gates
- Draw circuit diagrams for Boolean expressions using only universal gates

3. Experimental Procedure

a. Draw truth tables and circuit diagrams for the following expressions using only NAND gates.

1. $F = AB + CD$
2. $F = X \sim Y + \sim XY + Z$
3. $F = A(CD + B) + BC$
4. $F = AB + BC + AC$

b. Draw truth tables and circuit diagrams for the following expressions using only NOR gates.

1. $F = (A + B)(C + D)$
2. $F = (A + B + C)(\sim A + \sim B + \sim C)$

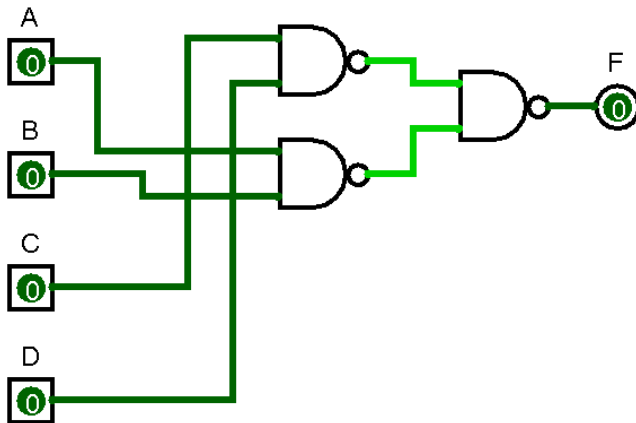
c. Use Logisim to generate truth tables and circuit diagrams for the expressions in 3(a).

d. Implement the first three expressions in the non-minimized form and verify the truth tables. Show the output to the course leader.

e. Why is it easier to draw a circuit diagram using universal gates if the Boolean expression is in standard/canonical form?

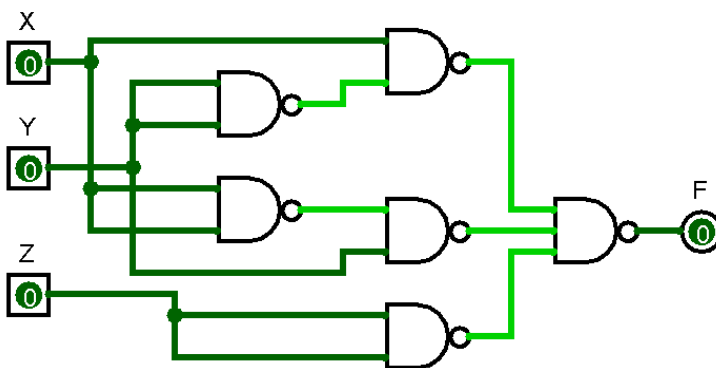
Your document should include:

- Handwritten truth tables and circuit diagrams for the expressions
- Logisim screenshots
- Answer to 3(e)

3.a.1. $F = AB + CD$


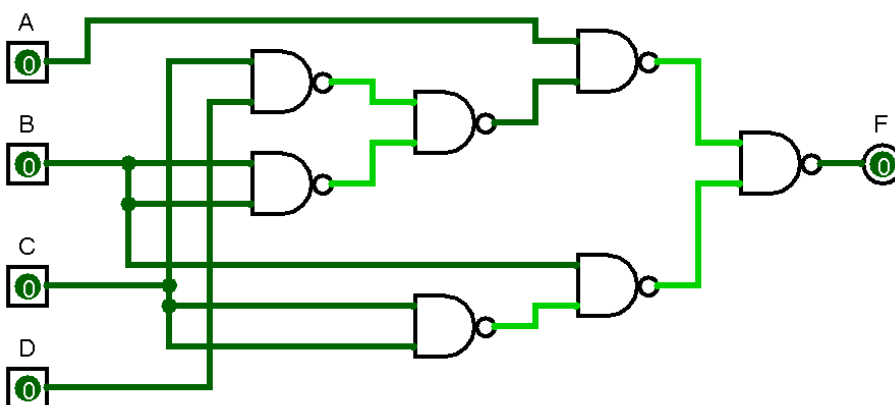
$$F = AB + CD$$

A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

 3.a.2. $F = X \sim Y + \sim XY + Z$


$$F = X \sim Y + \sim XY + Z$$

X	Y	Z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

 3.a.3. $F = A(CD + B) + BC$


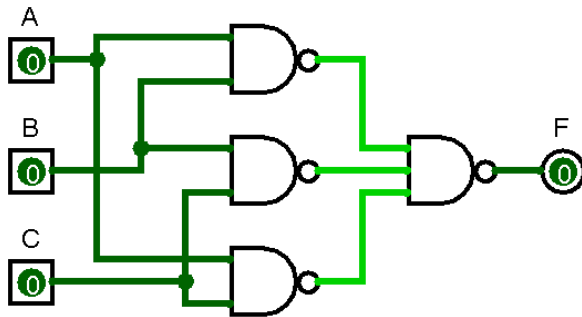
$$F = A(CD + B) + BC$$

A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Name: SATYAJIT GHANA

Reg. No: 17ETCS002159

3.a.4. $F = AB + BC + AC$



$$F = AB + BC + AC$$

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

3.e Since the Expression is already in the Canonical/Standard form, the expression can be considered to be simplified and flattened out, in the sense that its either a SOP or a POS form, the circuit using usual gates can be easily drawn, and we use the fact that NAND is the same as Invert OR, and NOR is the same as Invert AND. Hence it makes it easier to convert the canonical/standard expression into a NAND, NOR logic circuit.

