North South University

CSE231L

Experiment # 2

Name of Experiment: **Digital Logic Gates and Boolean Functions**

Date of Performance: 16 October, 2019

Date of Submission: 23 October, 2019

Section: 13

Group: 3

Submitted To: Farhana Saleh

Submitted By:

ID	Name
1530486042	Md. Abdul Zabbar
1711038042	MD. ASHRAFUL KABIR
1712747042	Ashik Iqbal
1731046042	Nahian -Al Sabri
1530187042	Md.Ahasun kamal

Lab2: Universal Gates

A. objectives:

* understand the concept of universal gates (NAND & NOR)

* Implement the basic logic gates using universal gates.

* Implement Boolean functions using universal gates.

* understand the gate level optimization.

B. Apparatus:

* Trainer Board.

* IC 7400 Quandruple 2 input NAND gates.

* IC 7402 Quandruple 2 input NOR gates.

C. Theory: A universal gate is a gate which can implement any boolean function without need to use any other gate type. The NAND

and NOR gates are universal gates. In practice, this is advantageous since NAND and NOR gates are economical and easier to fabricate and are the basic gate used in all IC digital logic families.

Figure C1 shows the implementation of NOT, AND & OR gates using only NAND gates:

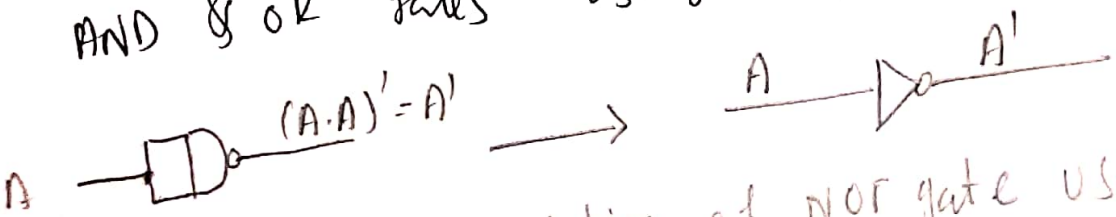


Fig : implementation of NOT gate using NAND gate

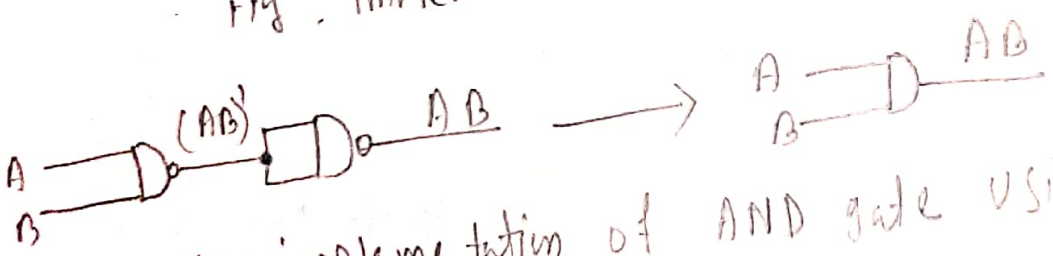


Fig: implementation of AND gate using NAND gate

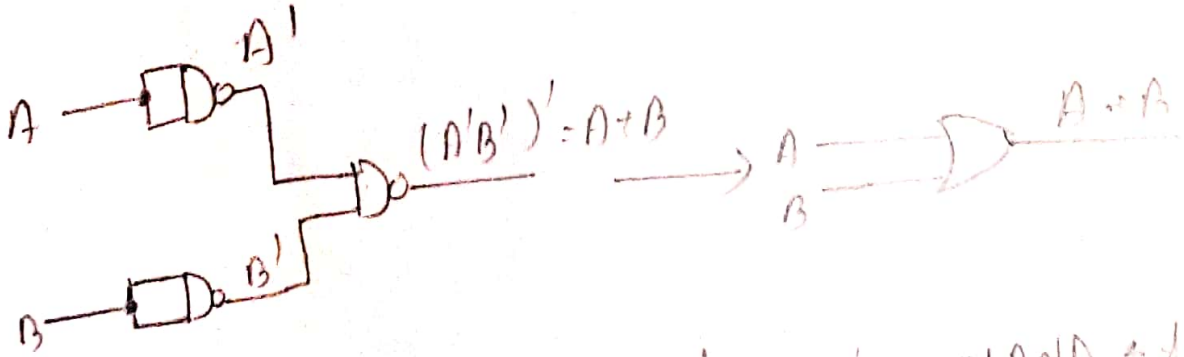


fig : implementation of OR gate using NAND gate.

Figure C1 : NAND as a universal gate.

D. Procedure: 1. Verify each of the NAND gate equivalent circuits in Figure C1 to perform the same operations of the basic gates.

2. Design, Construct and test the implementations of XOR and XNOR gates using NAND gates only. Show the circuits in Figure F1, clearly labeling the pin numbers.

3. Design, Construct and test the implementations of NOT, AND, OR, XOR and XNOR gates using NOR gates only.

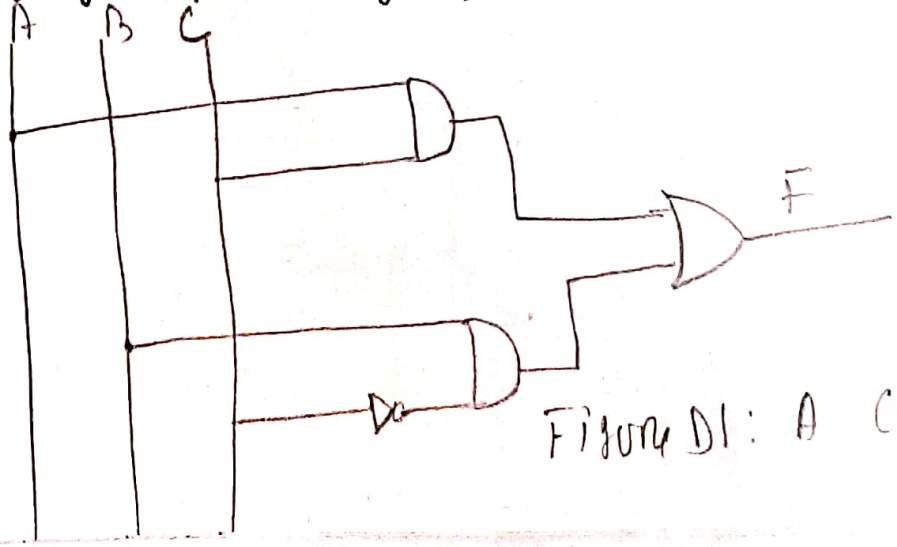


Figure D1: A Combinational Circuit.

4. Then, we did complete the truth table for the circuit in figure D1 in table F1 (section F)

5. And then we converted the circuit in figure D1 to NAND gate equivalent circuit, showing the steps involved and clearly labeled the pin numbers in the final circuit design. Then we showed our work in figure F3 (section F)

(i) Part 1 - we replaced each of the gates with its NAND gate equivalent.

(ii) Part 2 - Identified inversions that are compensated in part 1 and redrawn the final circuit in part 2.

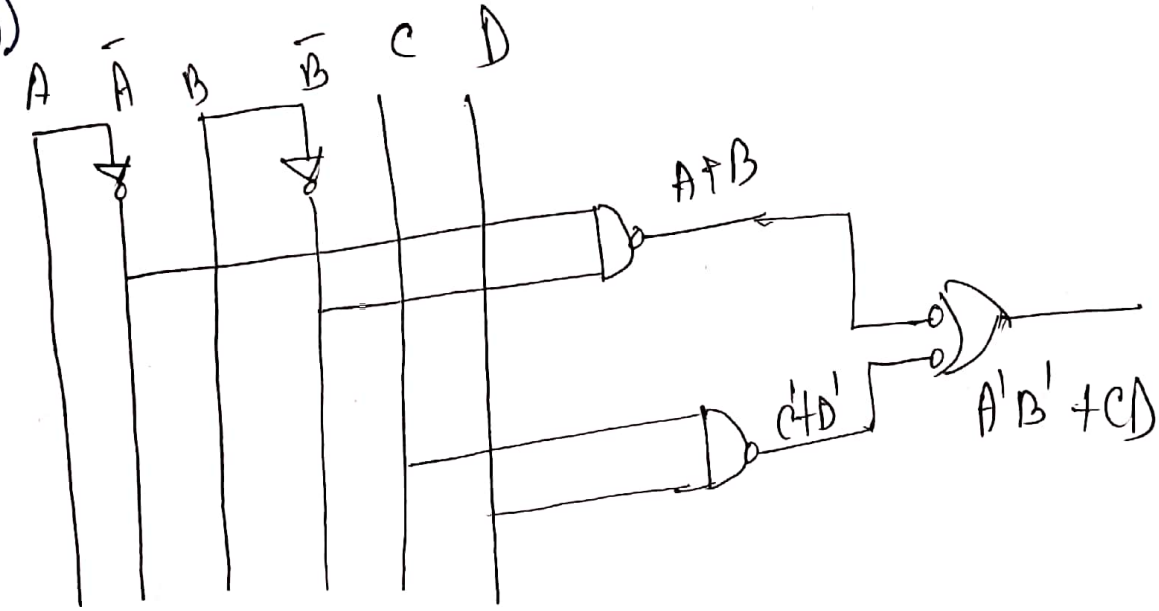
5. validated the operation of the universal gate circuit from the truth table.

Question Answers:

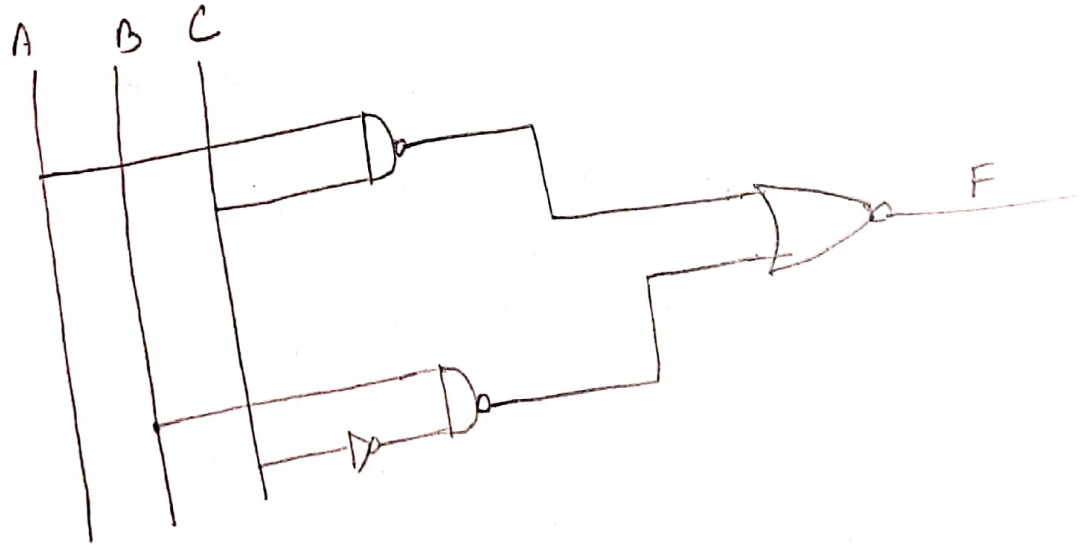
1) These gates are called universal gates, because all other gates like AND, OR, NOT, NOR and XNOR can be derived from it.

And, it is economical because multiple gate packages all contain multiples of the same gate. Ex: the 7400 contains 4 x 2-input NAND gates. So, it makes sense to use a unused gate in the 7400 as inverter, rather than add a 7404 inverter IC. So, rather than increase the IC count, take advantage of unused gates for other purposes.

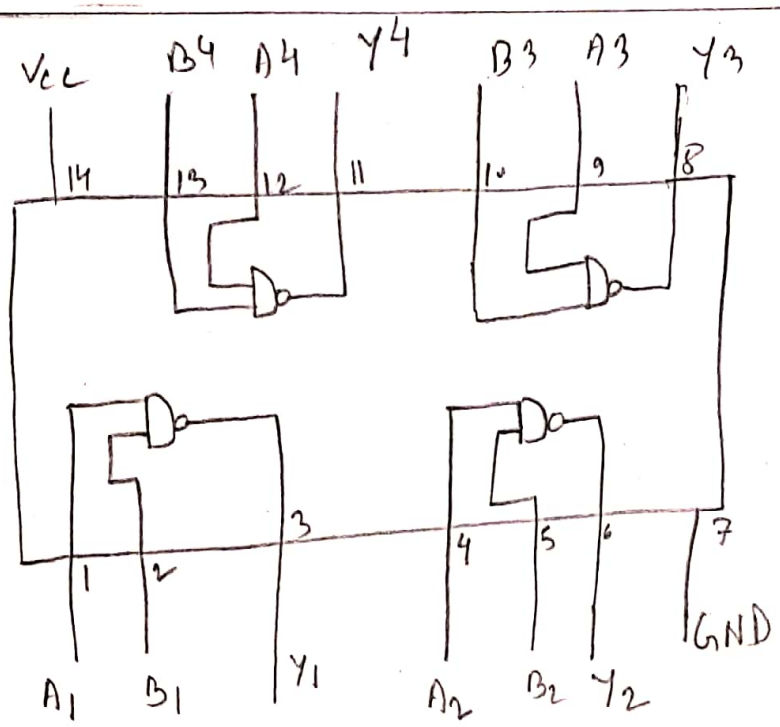
2)



3)



A. Obi



Discussion:

(i) We have faced some IC Problems.

(ii) We have problems in voltage source. So, we plugged two wires into switch and switches into 1 to make voltage and 0 to make ~~gnd~~ GND.

Lab 2: Universal Gates

A. Objectives

- Understand the concept of Universal Gates (NAND & NOR)
- Implement the basic logic gates using universal gates
- Implement Boolean functions using universal gates
- Understand gate level minimization

B. Apparatus

- Trainer Board
- IC 7400 Quadruple 2-input NAND gates
- IC 7402 Quadruple 2-input NOR gates

C. Theory

A universal gate is a gate which can implement any Boolean function without need to use any other gate type. The NAND and NOR gates are universal gates. In practice, this is advantageous since NAND and NOR gates are economical and easier to fabricate and are the basic gates used in all IC digital logic families.

Figure C1 shows the implementation of NOT, AND & OR gates using only NAND gates.

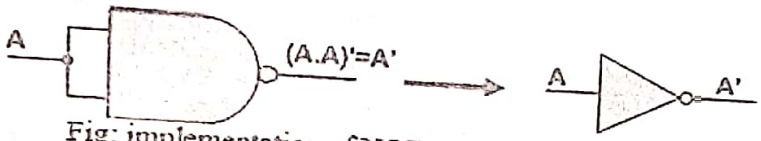


Fig: implementation of NOT gate using NAND gate

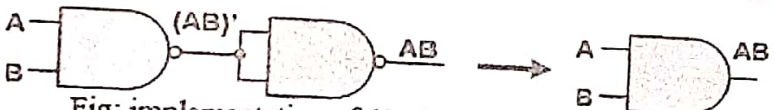


Fig: implementation of AND gate using NAND gate

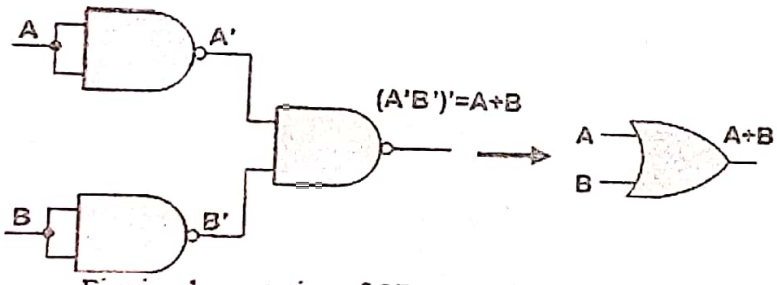


Fig: implementation of OR gate using NAND gate

Figure C1: NAND as a universal gate

D. Procedure

1. Verify each of the NAND gate equivalent circuits in **Figure C1** to perform the same operations of the basic gates.
2. Design, construct and test the implementations of XOR and XNOR gates using NAND gates only. Show the circuits in **Figure F1** (Section F), clearly labeling the pin numbers.

3. Design, construct and test the implementations of NOT, AND, OR, XOR and XNOR gates using NOR gates only. Show the circuits in **Figure F2** (Section F), clearly labeling the pin numbers.

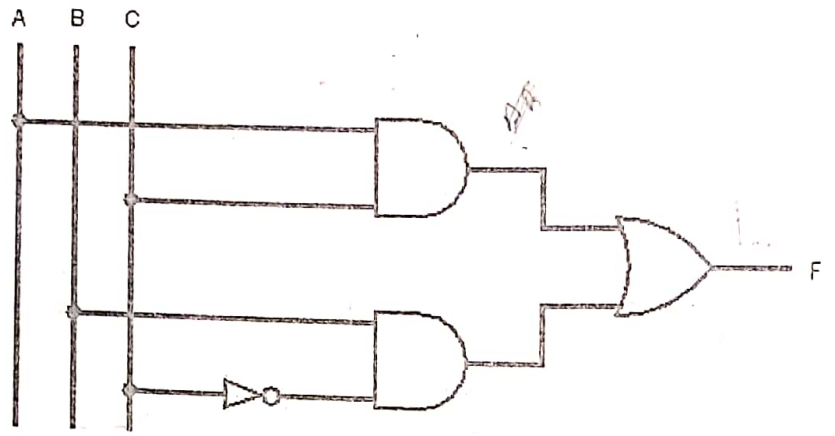


Figure D1: A combinational circuit

4. Complete the truth table for the circuit in **Figure D1** in Table F1 (Section F).
5. Convert the circuit in **Figure D1** to a NAND gate equivalent circuit, showing the steps involved and clearly labeling the pin numbers in the final circuit design. Show your work in **Figure F3** (Section F).
   (i) **Part 1** - Replace each of the gates with its NAND gate equivalent.
   (ii) **Part 2** - Identify any inversions that are compensated (i.e. one inverter followed by another) in part 1 and redraw the final circuit in part 2.
6. Validate the operation of the universal gate circuit from the truth table.

Questions:

1) Why are NAND and NOR gates called 'universal gates'? Why is it economical to use only one type of gate to produce digital logic ICs?
2) Draw the logic diagram for the given Boolean equation using only NOR gates. Use as few gates as possible.
   - **F = A'B' + CD**
3) Convert the circuit in Figure D1 to a minimized NOR gate equivalent circuit.
4) Use Logisim to simulate the circuit you drew for question 3. Attach a printout of the Logisim circuit screenshot and truth table screenshot (on a single page) to this assignment sheet
5) Draw the IC diagram for the circuit in **Figure F3 – Part 2.**

CSE231L – Lab 2 – Universal Logic Gates

Data Sheet: Instructor's Signature: .....

Section: 13	Group No.: 3	Date: 16 Oct, 2019

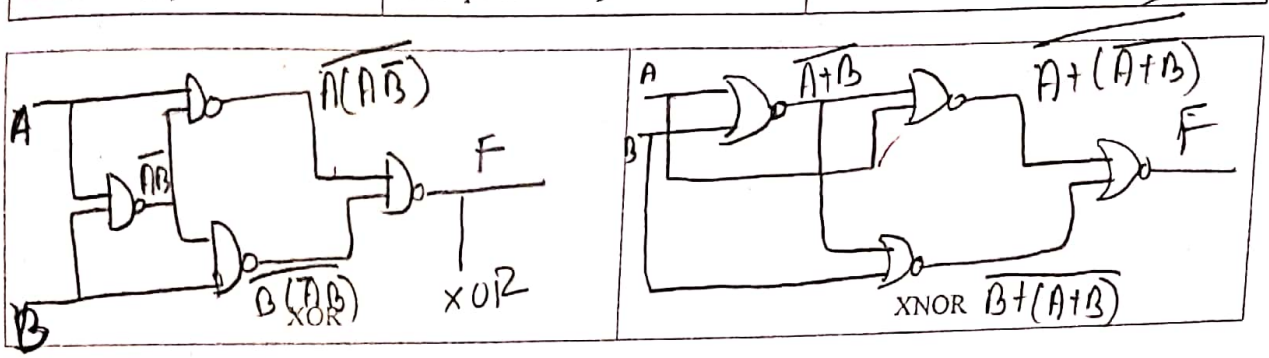


Figure F1: Implementation of XOR and XNOR using NAND gates

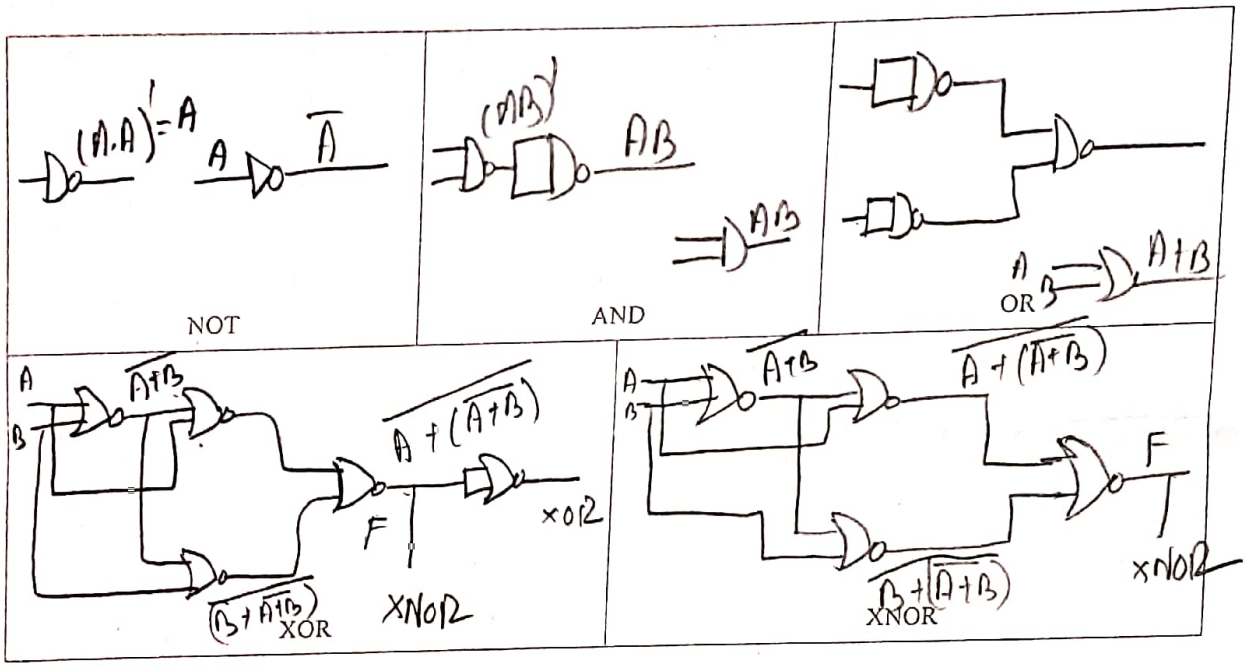
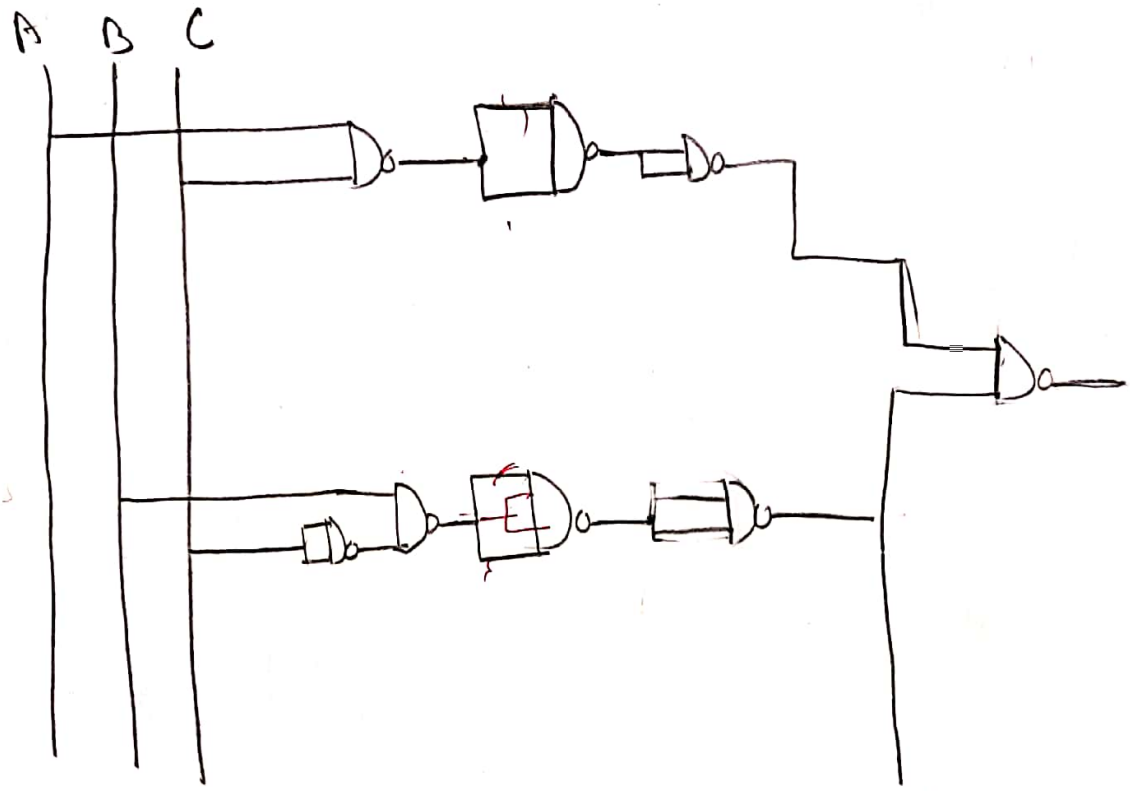


Figure F2: Implementation of NOT, AND, OR, XOR and XNOR using NOR gates

A B C	$I_1 = AC$	$I_2 = BC'$	$F = I_1 + I_2$
0 0 0	0	0	0
0 0 1	0	0	0
0 1 0	0	1	1
0 1 1	0	0	0
1 0 0	0	0	0
1 0 1	1	0	1
1 1 0	0	1	1
1 1 1	1	0	1

Table F1: Truth table of combinational circuit in Figure F2

Part 1



Part 2

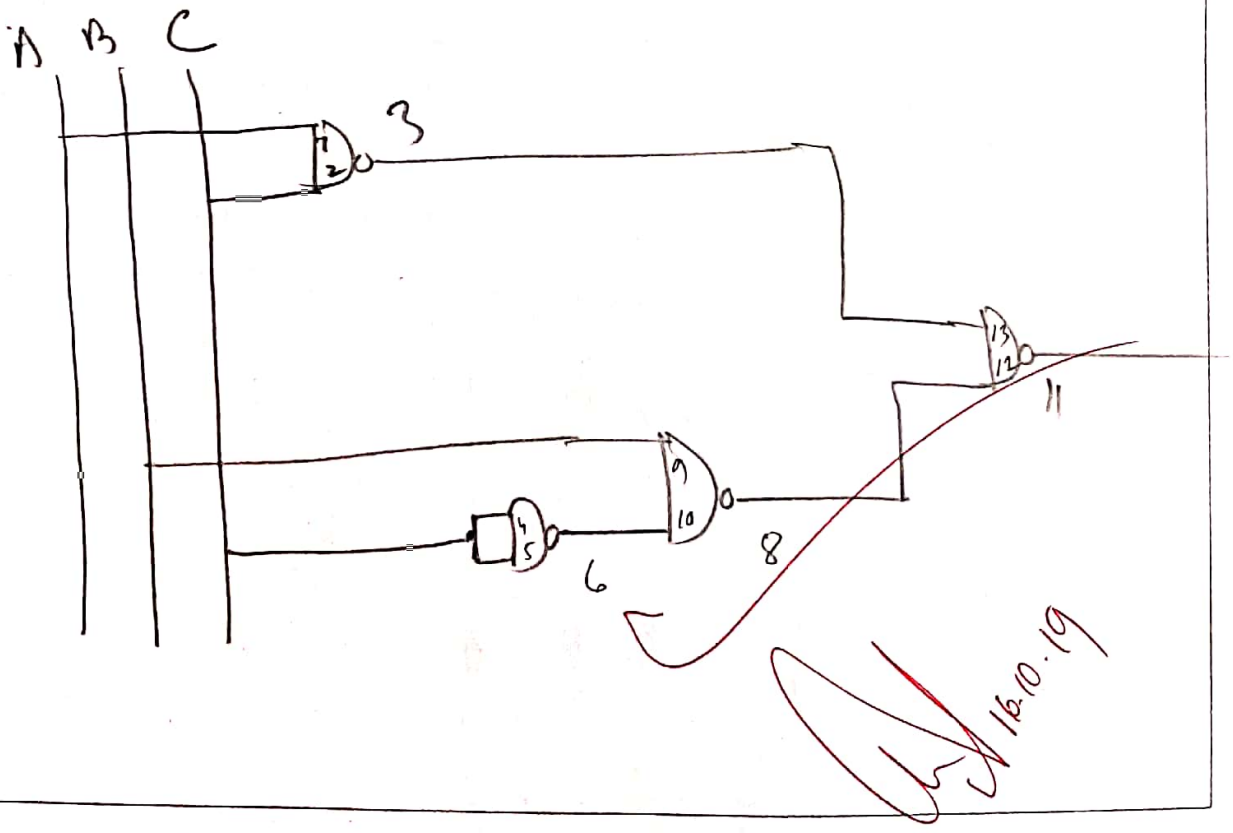


Figure F3: Universal (NAND) gate implementation of the circuit of Figure D1