North South University

Department of Electrical and Computer Engineering

LAB REPORT

Course name: Digital Logic Design

Course Code: 231

Experiment Number: 01

Experiment name: Digital Logic Gates and Boolean Functions

Experiment Date: 2nd March, 2021

Report Submission Date: 8th March, 2021

Section: 06

Group no: 04

Student Name	ID
Khalid Bin Shafiq	1911342642
Rafidul Islam	1912152642
Rashiqur Rahman Rifat	1911445642
Towsif Muhtadi Khan	1911576642
Remarks:	**Score:**

Objectives

- Study the basic logic gates - AND, OR, NOT, NAND, NOR, XOR.
- Get acquainted with the representation of Boolean functions using truth tables, logic diagrams and Boolean Algebra.
- Prove the extension of inputs of AND and OR gates using the associate law.
- Become familiarized with combinational logic circuits.

Theory

Logic gates

Logic gates are the components which are used to design a logical circuit or digital circuit. These logic gates are used to perform logical operations from logical inputs to get a single logical output. Logic gates only consider two discrete values of voltage level to determine the inputs. Those values also can called binary inputs. The binary value 1 is to represent logical high or True and the value 0 is to represent the logical low or false. Logic gates are named as AND, OR, NOT, NAND, NOR, XOR to perform logical operations and design digital circuit.

Truth Table

A truth table is a binary input-output table, used to represent the corresponding outputs by a function using logic gates. It describes the relationship between the input and output of a logic circuit. For each and every possible combination of inputs, a truth table show all possible outputs.

Boolean Algebra

Boolean algebra is used to analyze and simplify the digital (logic) circuits. It uses only the binary numbers 0 and 1. It is also called as Binary Algebra or logical Algebra. Digital circuits are expressed with Boolean algebra. A set of axioms and theorems are used to simplify Boolean equations.

Combinational Logic

It is a kind of digital logic which is implemented by Boolean circuits. Here the output is dependent on the present inputs and is not affected by previous state. For analyzing, it requires writing the Boolean functions for each element of the circuit, producing their truth tables, and subsequently combining each function for the final output and truth table.

Integrated Circuit (IC)

This is an electronic circuit formed by small piece of semiconducting material. In Digital logic design, we use Integrated circuits of 7400 series. These IC's contains various types of gates. For example, 7408 IC contains 4 AND gates where, 7404 IC contains 6 NOT gates. All of these gates have 14 pins. Among these pins the 7^{th} pin is designed to connect to the ground and the 14^{th} pin is designed to connect to +5V as V_{cc}

Equipment List

- IC 7400 Quadruple 2-input NAND gates
- IC 7402 Quadruple 2-input NOR gates
- IC 7404 Hex Inverters (NOT gates)
- IC 7408 Quadruple 2-input AND gates
- IC 7432 Quadruple 2-input OR gates
- IC 7486 Quadruple 2-input XOR gates
- Trainer Board
- Wires

Circuit Diagram

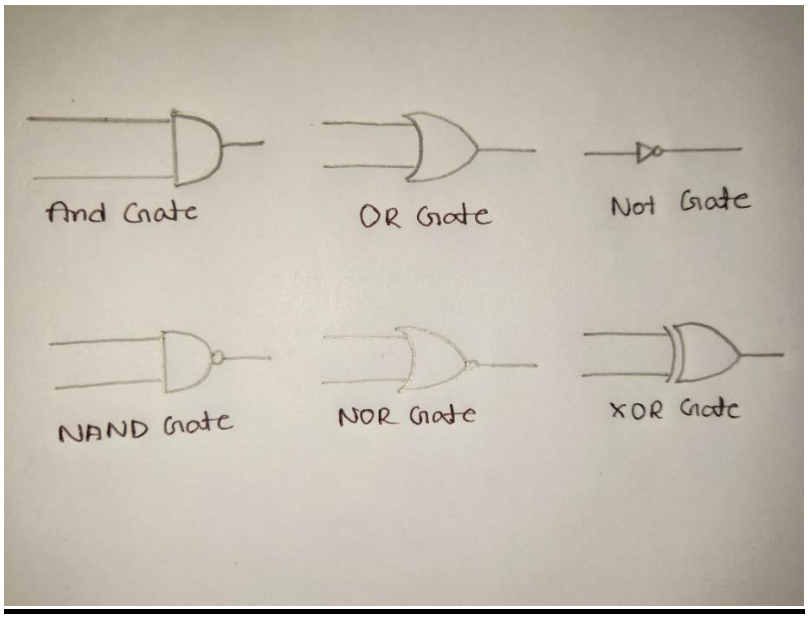


Figure F.1.1: Pin configurations of gates in ICs

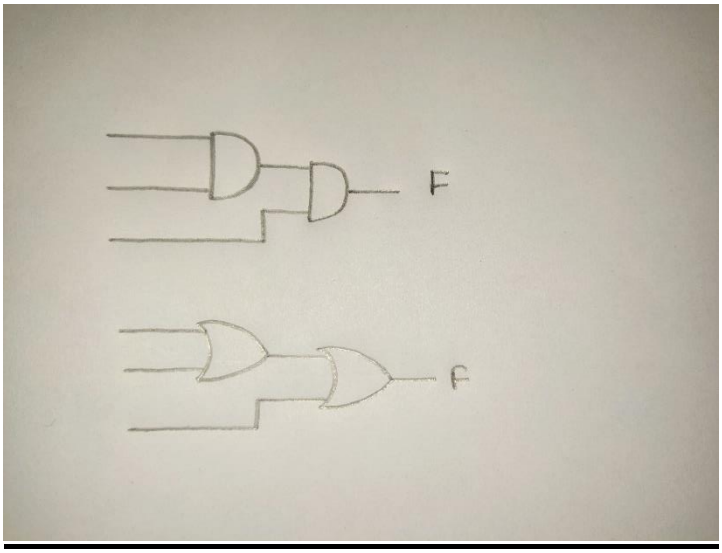


Figure F.2.1: Extension of inputs of AND and OR gates

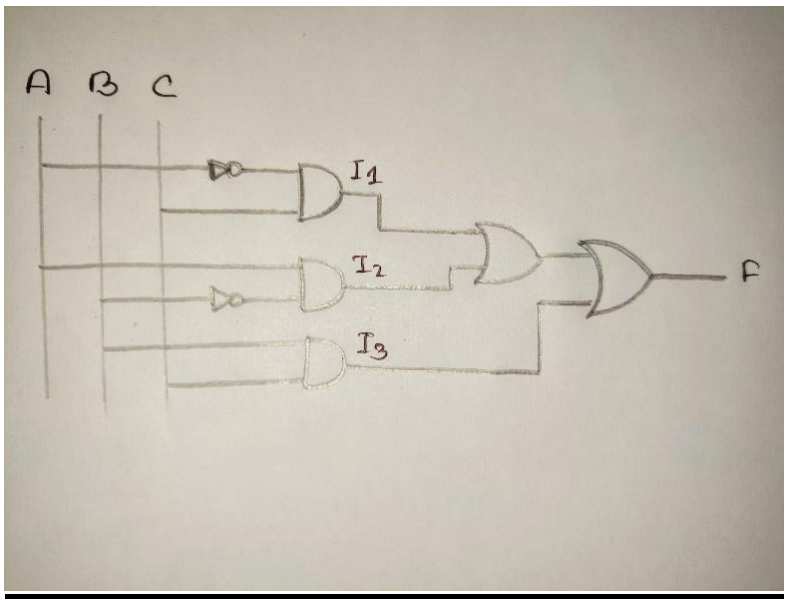


Figure F.3.1: Logic Diagram for the given Boolean Function

Data and Table

Input A	Input B	And *F=A.B*	OR *F=A+B*	NAND *F=(A.B)'*	XOR	NOR *F=(A+B)'*
0	0	0	0	1	0	1
0	1	0	1	1	1	0
1	0	0	1	1	1	0
1	1	1	1	0	0	0

Input A	NOT F=A'
0	1
1	0

Table F.1.1: Truth Table of Logic Gates

Input A	Input B	Input C	$F=ABC$	$F=A+B+C$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Table F.2.1: Truth Tables for 3-input AND and OR

$F=ABC$=A(BC)=(AB)C
$F=A+B+C=A+(B+C)=(A+B)+C$

Table F.2.2: Expressing 3-input gates as 2-input gates using associative law.

Input A	Input B	Input C	$I_1=A'C$	$I_2=AB'$	$I_3=BC$	$F=I_1+I_2+I_3$
0	0	0	0	0	0	0
0	0	1	1	0	0	1
0	1	0	0	0	0	0
0	1	1	1	0	1	1
1	0	0	0	1	0	1
1	0	1	0	1	0	1
1	1	0	0	0	0	0
1	1	1	0	0	1	1

Table F.3.1: Truth Table for the given Boolean Function

Result Analysis and Discussion

In Lab 01, we did 2 experiments to understand digital logic gates and Boolean functions. In the First Experiment, our goal was to learn about the basic characteristics of the AND, OR, XOR, NAND, NOT & NOR GATES and how to build the gates using Logisim. Then we got to know how to use to gates to simulate a circuit diagram. In case of OR gate, the output is 1 if one or both the inputs are 1. On the other hand, the output of an AND gate will be 1 only when both the inputs are 1. If one of the inputs is 0, then the output will be 0. And a NOT gate performs logical negation on its input. In other words, if the input is 1, then the output will be 0 and vice versa. Following these basic operations, we did our first experiment.

And then we did our second experiment in which our objective was to proof associative law by constructing a 3-input AND gate or OR gate from 2-input AND or OR gates. So, we drew our desire circuit for experiment 2. The circuit output truth table matched with our truth table. So, after analyzing this truth table we got to know that the associative law is same for both practical simulation and theoretical proof. Then we simulated 6-input AND gate in Logisim using only 2-input AND gates.

After doing these two experiments, now we know how to use Logisim and its functions and have a basic idea of logic gates and truth table. Also know about Boolean function and associative law.

Questions

E.1.2 Questions:

1) What are the names of the ICs that you would need if you wanted to use 13 AND gates, 12 NOT gates and 15 NOR gates in a circuit? How many of each IC would you need?

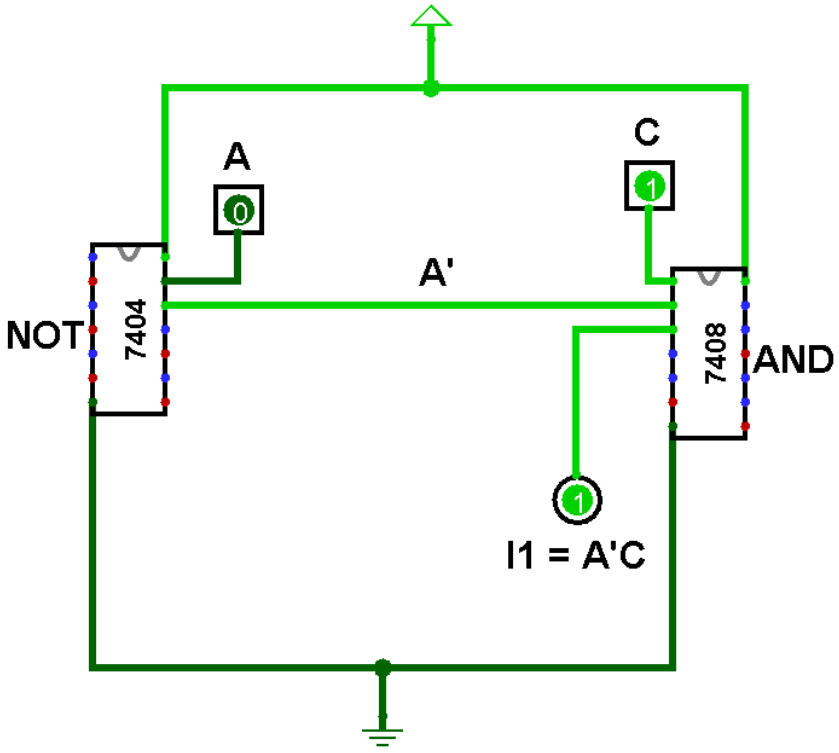
Ans: For 13 AND gates, I would need 4 7408 ICs. Each 7408 IC contains 4 AND gates. Therefore, we need $\lceil 13/4 \rceil = 4$ ICs.
For 12 NOT gates, I would need 2 7404 ICs. Each 7404 IC contains 6 NOT gates. Therefore, we need $12/6 = 2$ ICs.
For 15 NOR gates, I would need 4 7402 ICs. Each 7402 IC contains 4 NOR gates. Therefore, we need $\lceil 15/4 \rceil = 4$ ICs.

2) How can you power your logic ICs if the +5V port of your trainer board stops working?

Ans: logic IC can be powered directly by connecting its +5V port to DC power supply in case the +5V port of the trainer board stops working.

E.3.2 Questions:

1) Draw the IC diagram for the first implicant I1. In place of the logic gates, draw the ICs and all the connections required to make the circuit work.



Contributions:

1) Khalid Bin Shafiq - Question Answer
2) Rafidul Islam - Circuit Diagram
3) Rashiqur Rahman Rifat - Theory
4) Towsif Muhtadi Khan - Data and Table

Lab 1: Digital Logic Gates and Boolean Functions

F. Data Sheet

F.1 Introduction to Basic Logic Gates

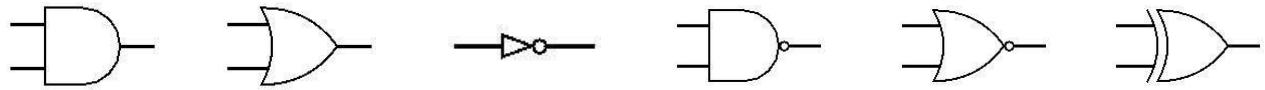


Figure F.1.1: Pin configurations of gates in ICs

Input	AND	OR	NAND	XOR	NOR
0 0	0	0	1	0	1
0 1	0	1	1	1	0
1 0	0	1	1	1	0
1 1	1	1	0	0	0

Input	NOT
0	1
1	0

Table F.1.1: Truth Table of Logic Gates

F.2 Constructing 3-input AND & OR gates from 2-input AND & OR gates

A B C	F = ABC	F = A+B+C
0 0 0	0	0
0 0 1	0	1
0 1 0	0	1
0 1 1	0	1
1 0 0	0	1
1 0 1	0	1
1 1 0	0	1
1 1 1	1	1

Table F.2.1: Truth Tables for 3-input AND and OR

F = ABC = A(BC)=(AB)C
F = A+B+C = A+(B+C)=(A+B)+C

Table F.2.2: Expressing 3-input gates as 2-input gates using associative law.

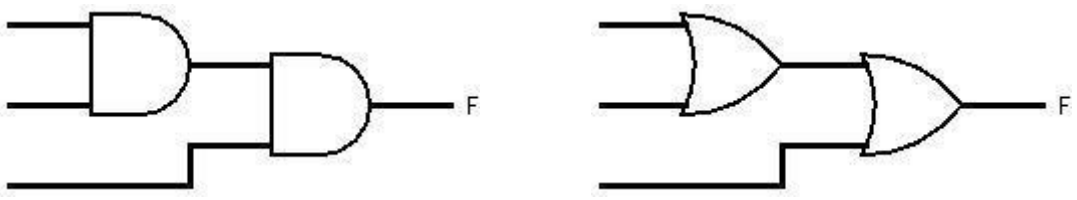


Figure F.2.1: Extension of inputs of AND and OR gates

F.3 Implementation of Boolean Functions

A B C	$I_1 = A'C$	$I_2 = AB'$	$I_3 = BC$	$F = I_1 + I_2 + I_3$
0 0 0	0	0	0	0
0 0 1	1	0	0	1
0 1 0	0	0	0	0
0 1 1	1	0	1	1
1 0 0	0	1	0	1
1 0 1	0	1	0	1
1 1 0	0	0	0	0
1 1 1	0	0	1	1

Figure F.3.1: Truth Table for the given Boolean Function

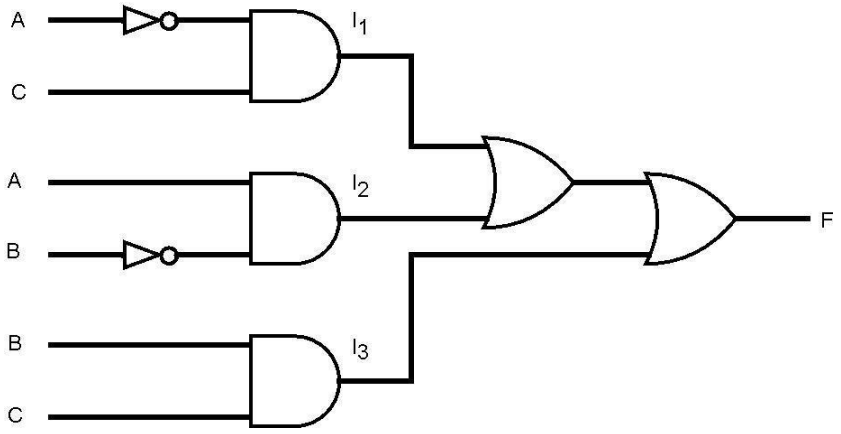
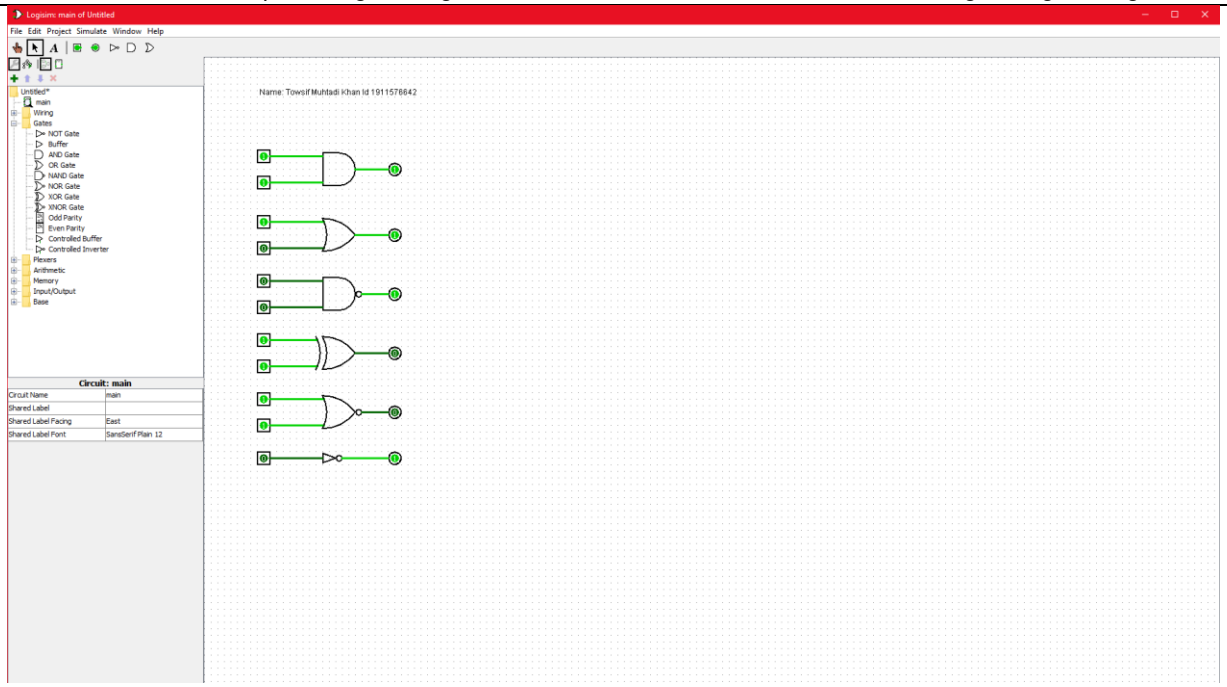


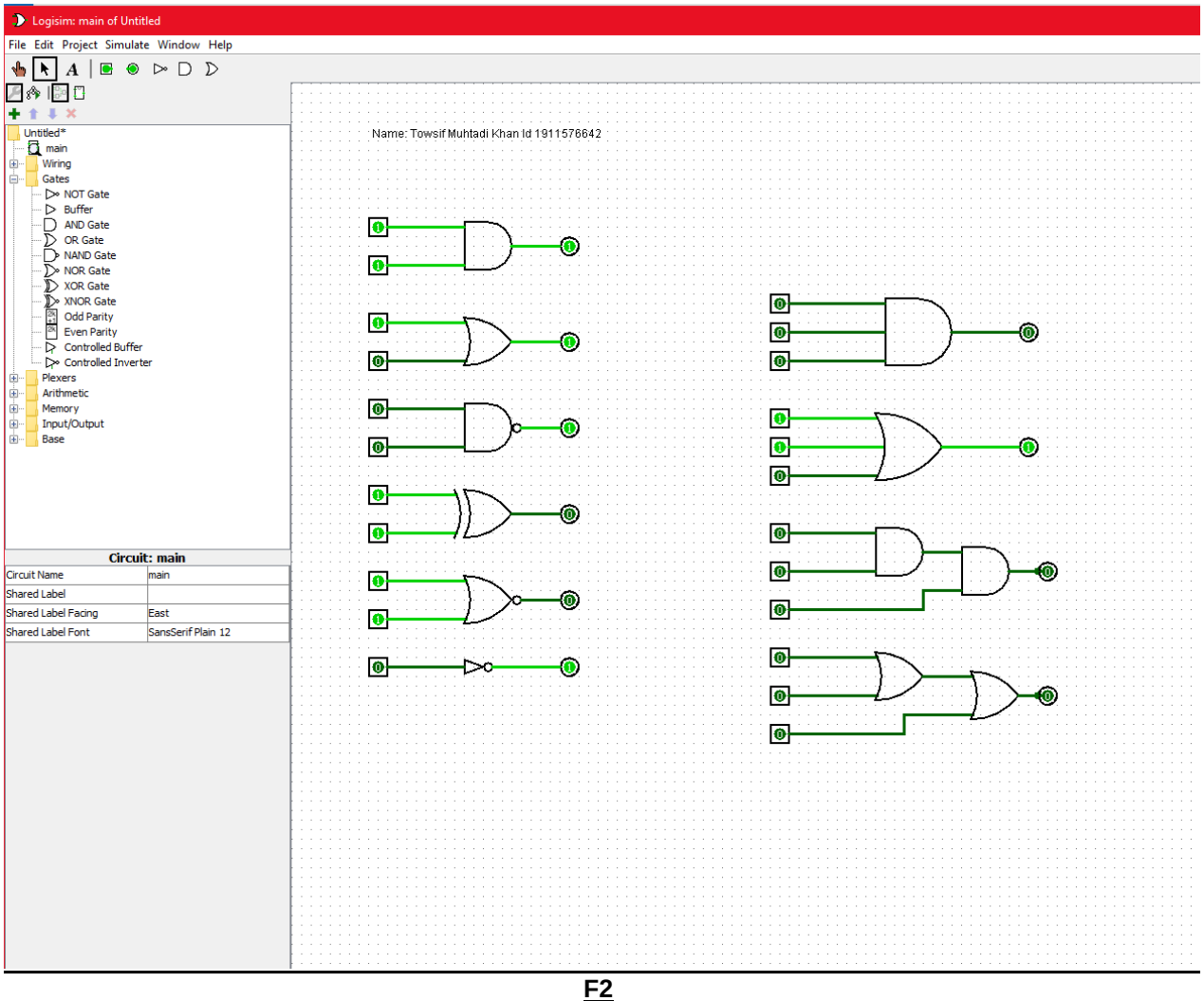
Figure F.3.1: Logic Diagram for the given Boolean Function

Attachment:

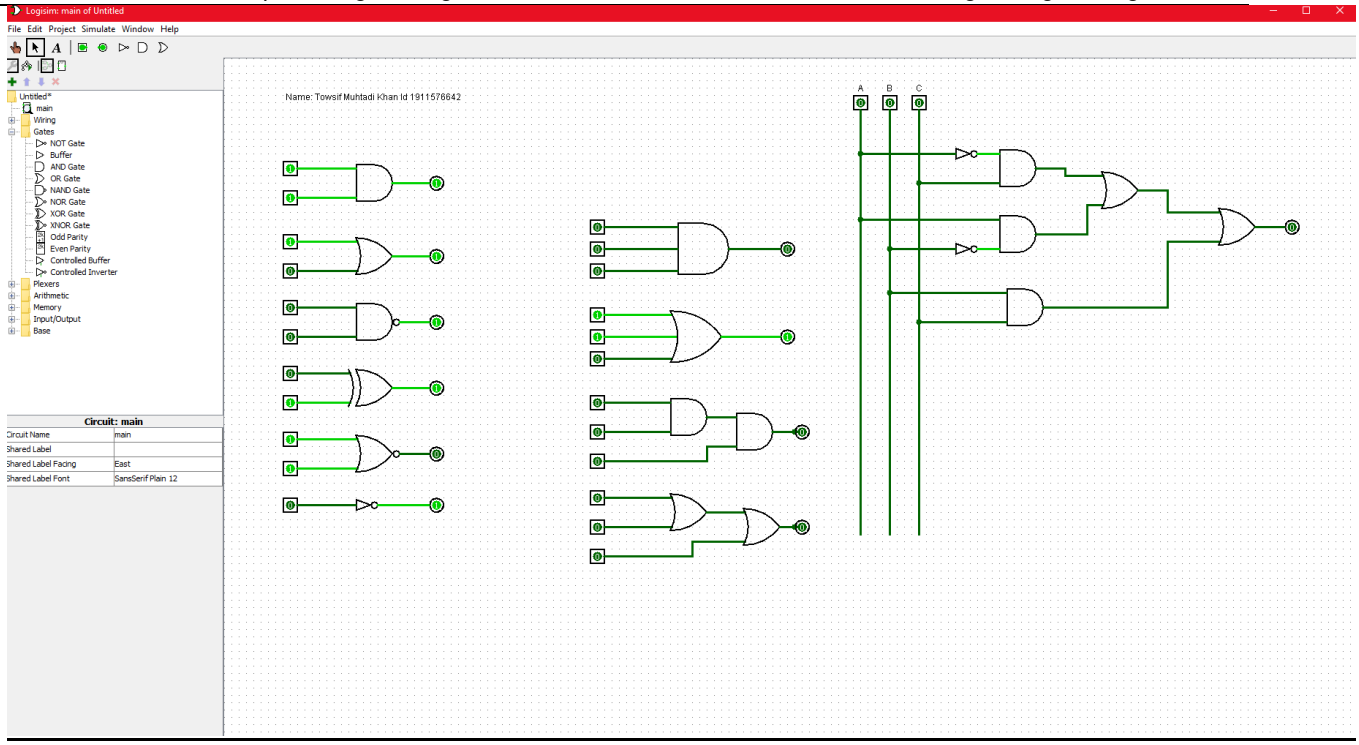
1. Attach the screenshots /image of the simulations below
2. Attach the required logic Diagrams with proper figure name and labeling below.


Logisim: main of Untitled
File Edit Project Simulate Window Help
Name: Towsif Muhtadi Khan Id 1911576642
Untitled*
main
Wiring
Gates
NOT Gate
Buffer
AND Gate
OR Gate
NAND Gate
NOR Gate
XOR Gate
XNOR Gate
Odd Parity
Even Parity
Controlled Buffer
Controlled Inverter
Plexers
Arithmetic
Memory
Input/Output
Base
Circuit: main
Circuit Name main
Shared Label
Shared Label Facing East
Shared Label Font SansSerif Plain 12

F1


Logisim: main of Untitled
File Edit Project Simulate Window Help
Name: Towsif Muhtadi Khan Id 1911576642
Untitled*
main
Wiring
Gates
NOT Gate
Buffer
AND Gate
OR Gate
NAND Gate
NOR Gate
XOR Gate
XNOR Gate
Odd Parity
Even Parity
Controlled Buffer
Controlled Inverter
Plexers
Arithmetic
Memory
Input/Output
Base
Circuit: main
Circuit Name main
Shared Label
Shared Label Facing East
Shared Label Font SansSerif Plain 12

F2



F3

Towsif Muhtadi Khan-1911576642