BRAC UNIVERSITY

Inspiring Excellence

Department of Electrical and Electronic Engineering

Hardware / Software(✔) Report Submission

Semester: Spring 26
Course Code: EEE283L
Course Title: Digital Logic Design
Section:03

Experiment No:01

Experiment Name: Familiarization of Fundamental Logic Gates

Group No: 02

Group members:

Sl.	***Name***	***ID***
1.	Souvik Barman Ratul (Ex01)	24121205
2.	Faiyaz Hasin Reza (Ex01)	24121374
3.	Abid Muhammad Elhan	24115001
4.	Golam Murtaza	23121017

Date of Submission: 15/02/2026

Prepared by:

Souvik Barman Ratul, Faiyaz Hasin Reza

Name of the Experiment: Familiarization of Fundamental Logic Gates.

Introduction: In this experiment, we mainly used the Proteus Design software to build and test digital circuits. Our main goal was to see how basic logic gates work by creating them on a computer rather than using real parts. Using a simulation makes it much easier to see how these circuits behave without needing physical hardware.

Objective: The main objective of this experiment is to become familiar with basic logic gates (AND,OR,NOT,NAND,NOR,XOR and XNOR) and their operations. It also aims to design and simulate logic circuits using Proteus software and observe the outputs for different combinations of input values.

Software:

- Proteus & Professional
- Digital logic library components inside Proteus

Components:

- Logic Gates (AND, OR, NOT, NAND, NOR)
- Logic toggle
- LED indicators
- Connecting wires

Simulation Diagram:

Using Fundamentel gates only

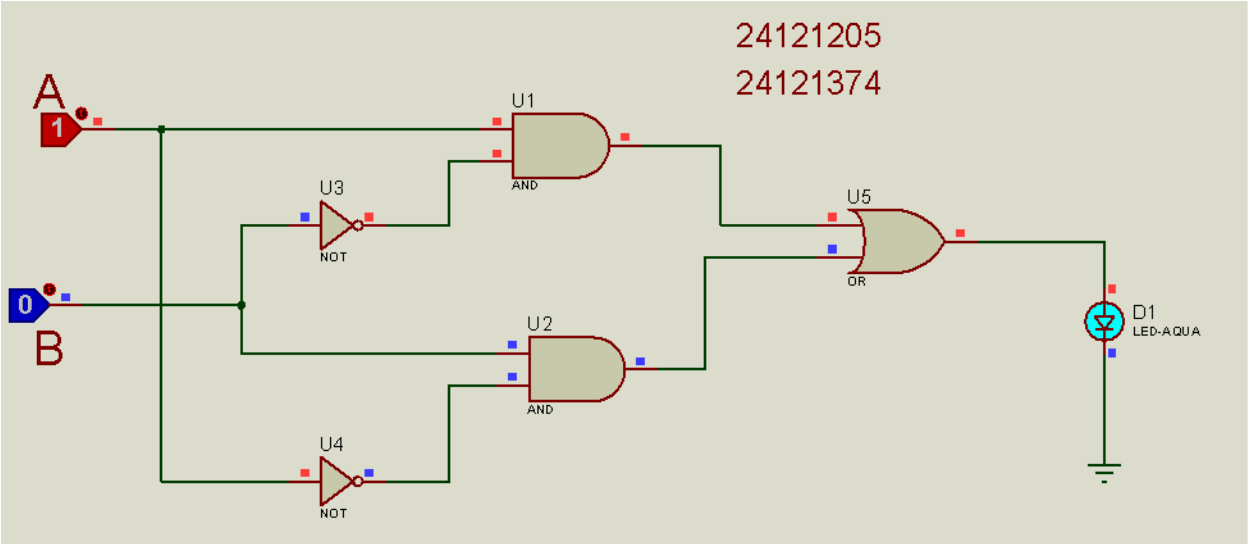


Figure: Circuit behavior for On condition [input(A=1,B=0)], (Output: High)

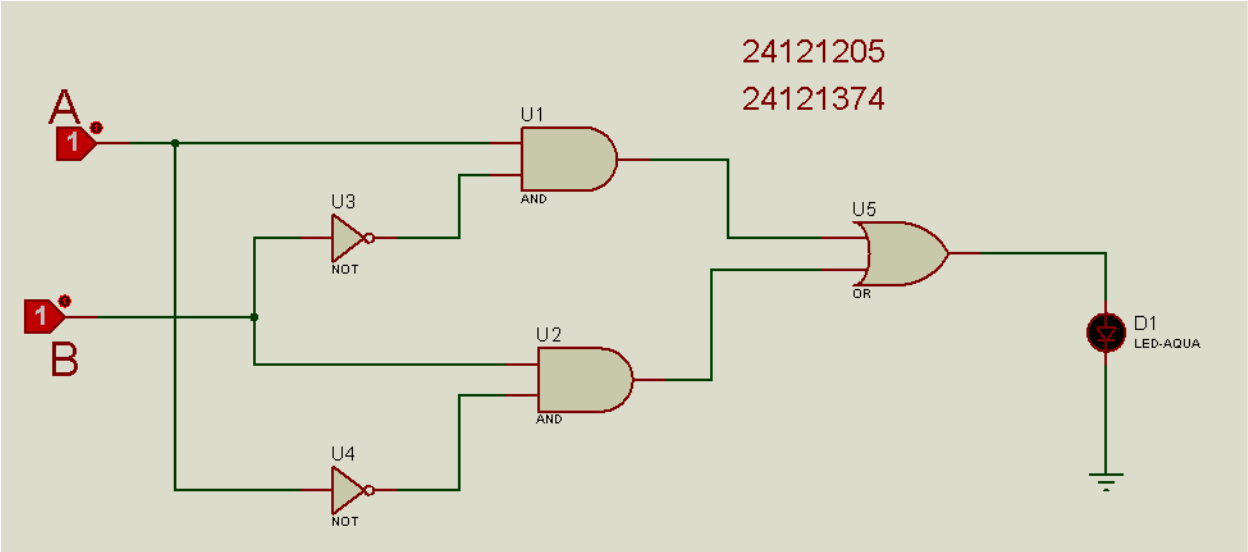


Figure: Circuit behavior for off condition [input(A=1,B=1)], (Output:Low)

Truth Table: (Using Fundamentel gates only)

A	B	$A\overline{B} + \overline{A}B$
0	0	0
0	1	1
1	0	1
1	1	0

Simulation Diagram:

Using Nor gates only

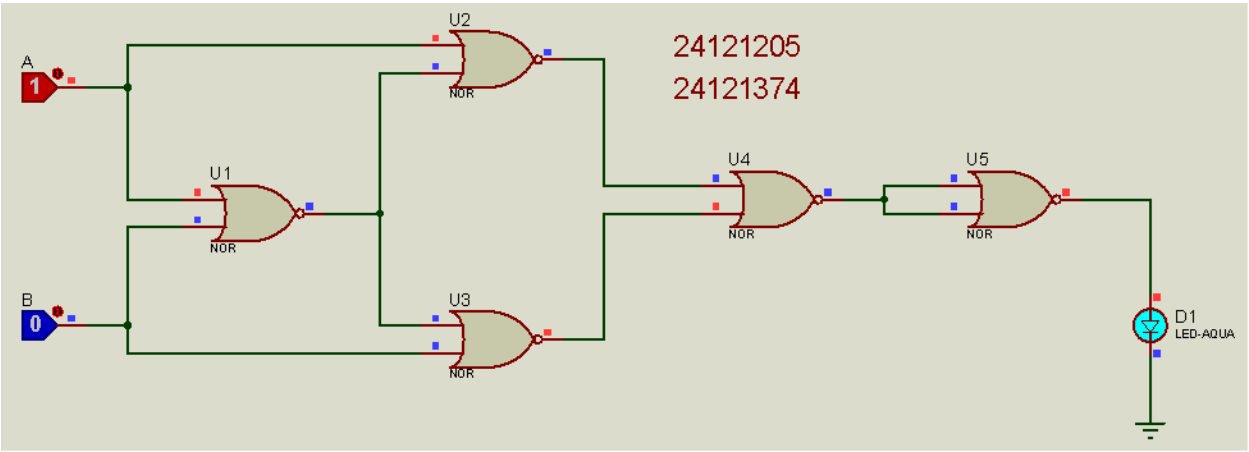


Figure: Circuit behavior for On condition [input(A=1,B=0)], (Output: High)

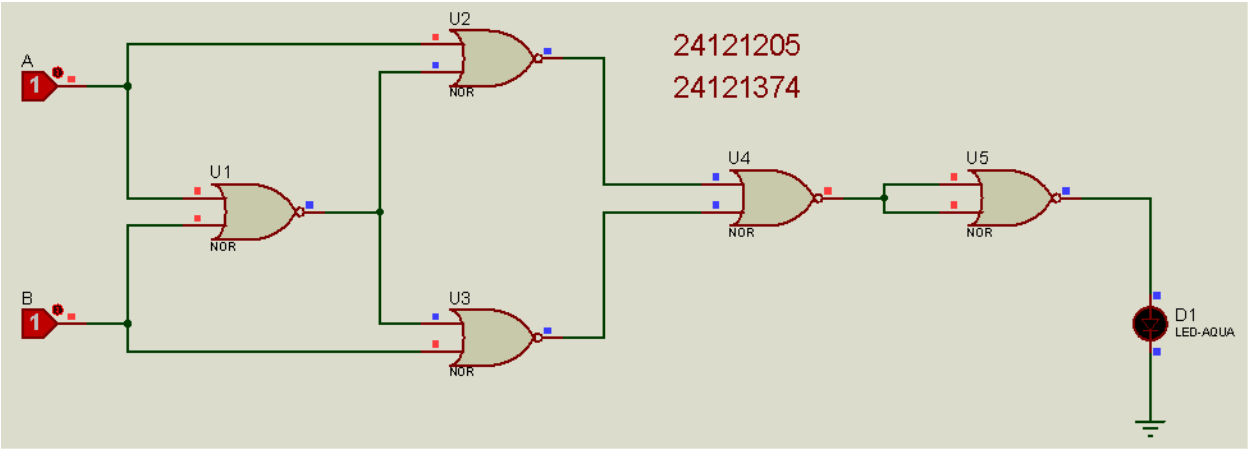


Figure: Circuit behavior for off condition [input(A=1,B=1)], (Output:Low)

Truth Table: (Using Nor gates only)

A	**B**	$A\overline{B} + \overline{A}B$
0	0	0
0	1	1
1	0	1
1	1	0

Discussion:-

In this experiment, we simulated basic logic gate circuits using Proteus professional 8 software. We tested different input combinations using logic toggles and observed the output using LED indicators. The results from the simulation matched the expected truth table values. We also implemented the same logic function using fundamental gates and then using only NOR gates. From both designs, we found that the output behavior remained the same for all input conditions. This experiment helped us clearly understand how logic gates work .

Conclusion:-

In conclusion, we successfully observed the working of basic logic gates using Proteus simulation. The output results matched the truth table for all input combinations. This experiment helped us understand logic gate operations and how the same logic function can also be implemented using only NOR gates.

Proteus File Google Drive link:

https://drive.google.com/drive/u/5/folders/162zaRLFpC-gz8e6xwn_H64u2vZf9YVOO

BRAC UNIVERSITY

Inspiring Excellence

Department of Electrical and Electronic Engineering

Hardware / Software(✔) Report Submission

Semester: Spring 26
Course Code: EEE283L
Course Title: Digital Logic Design
Section:03

Experiment No:02

Experiment Name: Study and Application of Logic Simplification Techniques

Group No: 02

Group members:

Sl.	***Name***	***ID***
1.	Souvik Barman Ratul	24121205
2.	Faiyaz Hasin Reza	24121374
3.	Abid Muhammad Elhan (Ex02)	24115001
4.	Golam Murtaza (Ex02)	23121017

Date of Submission: 15/02/2026

Prepared by: ABID MUHAMMAD ELHAN

Experiment Name: Study and Application of Logic Simplification Techniques

F(A,B,C,D) = M(1,2,4,15) + d(0,3,14)

Objective:

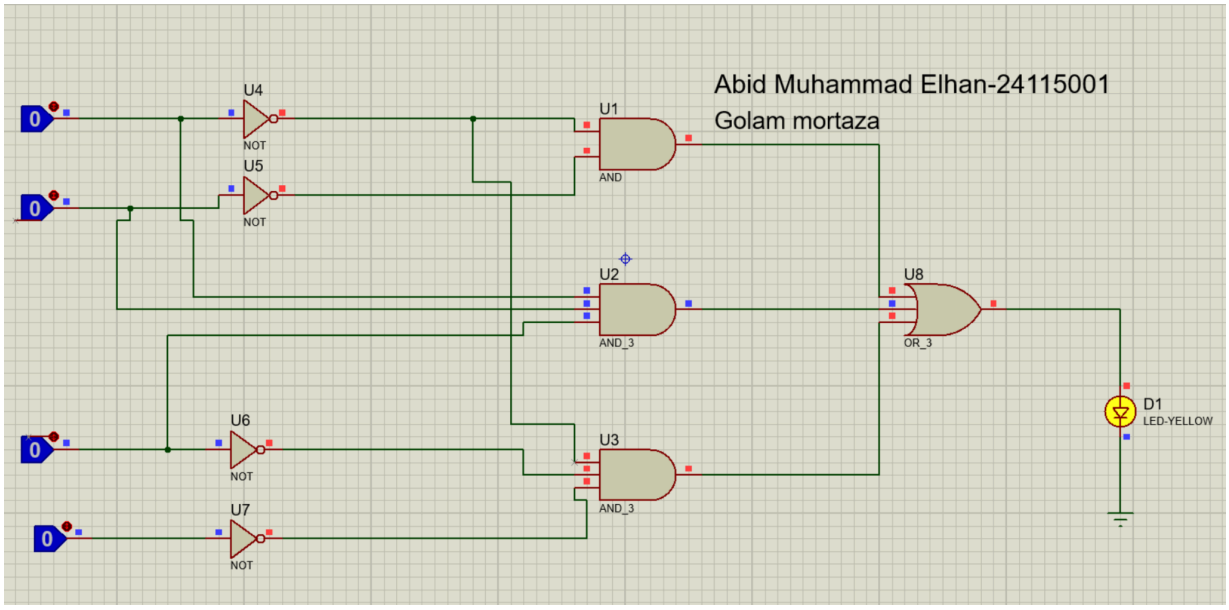
The primary objective is to utilize Proteus Design Suite to simulate and verify logic simplification techniques. By modeling minimized SOP/POS expressions, we aim to validate that theoretical K-map (not included in this as not done in class) reductions maintain functional accuracy while improving design efficiency. Key goals include performing virtual prototyping with basic logic components, verifying truth tables via Logic Probes, and analyzing the impact of "Don't Care" conditions. By using the software-driven approach we can demonstrate how reducing gate count minimizes circuit complexity and cost before physical implementation.

Apparatus:

Software–
Proteus 8 professional

Virtual Components-
- Logic gates used (NOT, OR , AND . AND_3, OR_3)
- Logictoggle
- LED-yellow as logicprobe

Circuit used:



Case:1

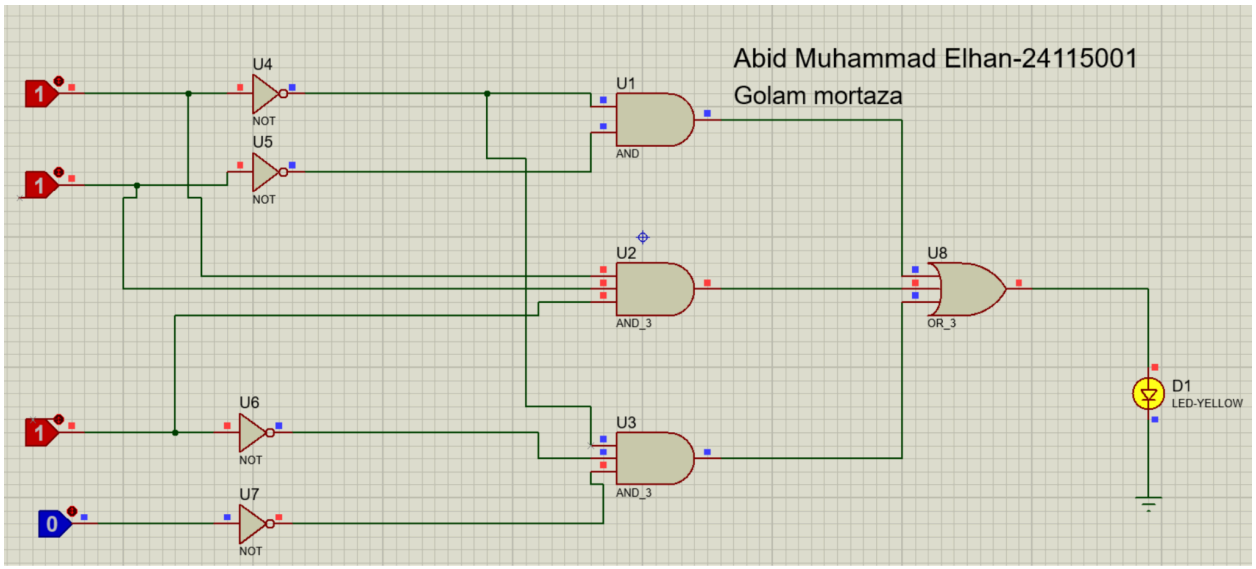


Figure: Circuit behavior for condition [input(A=1,B=1,C=1.D=0)], (Output: X)

Case-2

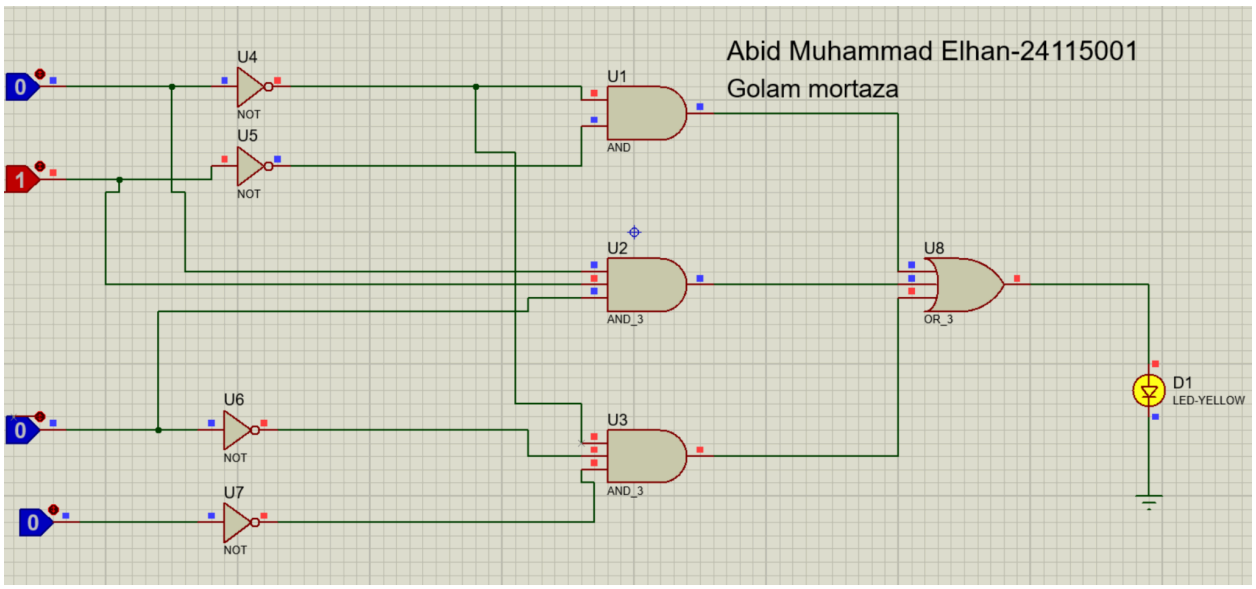


Figure: Circuit behavior for condition [input(A=0-,B=1,C=0.D=0)], (Output: 1)

Case-3

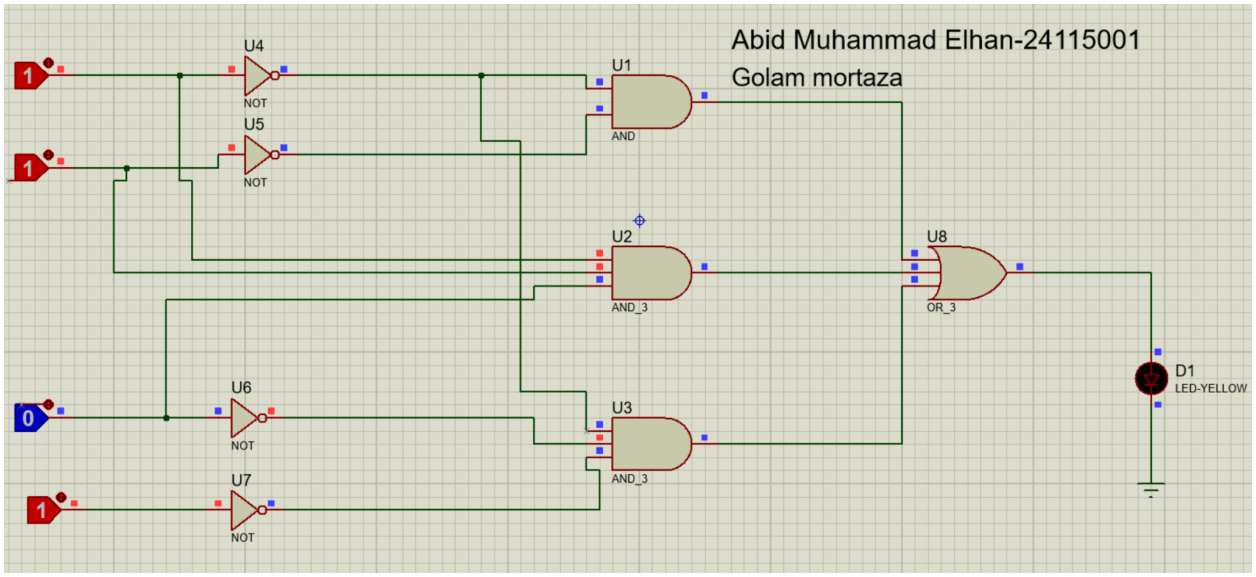


Figure: Circuit behavior for condition [input(A=1-,B=1,C=0.D=1)], (Output: 0)

Truth table:

A	B	C	D	Y
0	0	0	0	X
0	0	0	1	1
0	0	1	0	1
0	0	1	1	X
0	1	0	0	1

0	1	0	1	0
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	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	X
1	1	1	1	1

Discussion

We used Proteus 8 Professional to simulate a 4-variable logic function, focusing on reducing hardware complexity. By incorporating "Don't Care" conditions (0, 3, 14), we observed how indifferent states allow for more efficient logic minimization. The software validation confirmed that the simplified circuit correctly identifies minterms like m_4(Case 2) while maintaining low gate counts. Logic Probes and LEDs verified functional accuracy against the theoretical truth table.

Conclusion

This simulation successfully demonstrated the transition from Boolean theory to an optimized virtual prototype. We understood the "Don't Care" utilization to effectively to minimize circuit cost and complexity without compromising output integrity. The consistency between the Proteus circuit behavior and the truth table proves that software-driven verification is a reliable method for designing efficient digital systems before proceeding to physical breadboard implementation.

Proteus File Google Drive link:

https://drive.google.com/drive/u/5/folders/162zaRLFpC-gz8e6xwn_H64u2vZf9YVOO