Digital Logic Design (EL-1005)

LABORATORY MANUAL

Spring-2024

LAB 04

Advance Logic Gate and Boolean Algebra

STUDENT NAME ______ ROLL NO ______ SEC ______

INSTRUCTOR SIGNATURE & DATE ______

MARKS AWARDED: /10

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES (NUCES), KARACHI

Date: 12th Feb 2024

Prepared By: Muhammad Nouman Hanif

Lab Session 04: # Advance Logic Gate and Boolean Algebra

OBJECTIVES:

The objectives of this lab are:

- Investigate the logical properties of exclusive-OR (XOR), exclusive-NOR (XNOR) function and implement it using basic and universal gates.
- To utilize the fundamental operations of Boolean algebra in logic circuit measurements.

APPARATUS:

- Logic Works
- Logic Trainer

COMPONENTS:

ICs 74LS86 (XOR), 74LS266 (XNOR), 74LS02 (NOR), 74LS00 (NAND), 74LS08 (AND), 74LS32 (OR), 74LS04 (NOT)

Introduction:

Secondary gates can be made by the combinations of primary and universal gates. There are two types of secondary gates which may be termed as advanced gates,

1. The XOR Gate.
2. The XNOR Gate.

1. The Exclusive-OR Gate (XOR Gate):

The exclusive OR function is an interesting and useful logical operation. As the name implies, it is similar to the previously studied OR function, but it's a new and distinct operation. **"It is a device whose output is 1 only when the two inputs are different, but 0 if the inputs are the same."** This is useful for **comparator circuits**; if the inputs are different, then the output will be true, otherwise it is false. **The symbol for exclusive-OR function is $\oplus$ and the logical expression is shown in fig below.**

Symbol:

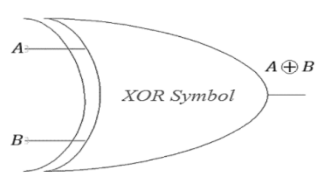


Figure 2 Exclusive-OR Gate Symbol

Function Table:

Inputs		Output
A	**B**	**Y**
L	L	L
L	H	H
H	L	H
H	H	L

Table: 1 XOR Gate Truth Table

H= Logic High, L= Logic Low

Connection Diagram:

74LS86 IC contains four 2-input XOR gates. The connection diagram for this IC is shown below:

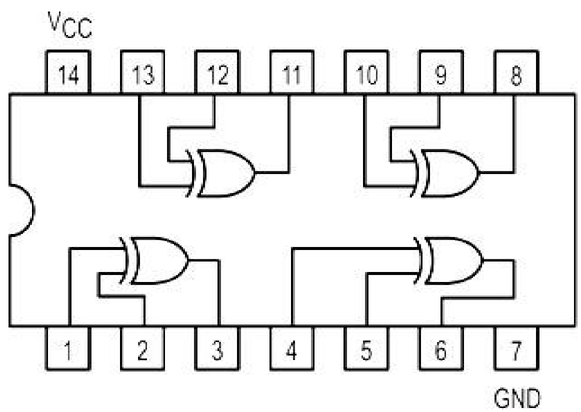
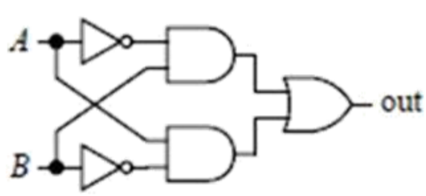


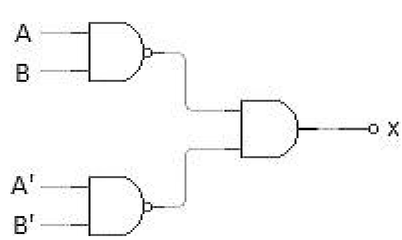
Figure 2 XOR Gate Connection diagram

The XOR gate can be implemented by using primary and universal gates as follows



A⊕ B = A'B+AB'

Figure 3 XOR Gate using primary Gate



A⊕ B = (AB)' **.** (A'B')' = (AB)' **.** A+B = A'B+AB'

Figure 4 XOR Gate using universal Gates

2. Exclusive-NOR Gate (XNOR)

An XNOR gate (sometimes referred to as Exclusive NOR gate) is a digital logic gate with two or more inputs and one output that performs logical equality. **The output of an XNOR gate is 1 when all of its inputs are same. If some of its inputs are 1 and others are 0, then the output of the XNOR gate is 0.**

Symbol:

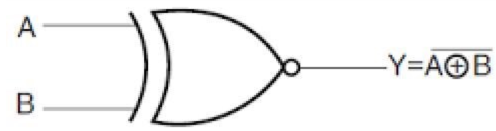


Figure 5 Exclusive-NOR Gate Symbol

Function Table:

Inputs		Output
A	**B**	**Y**
L	L	H
L	H	L
H	L	L
H	H	H

Table: 2 XNOR Gate Truth Table

H= Logic High, L= Logic Low

Connection Diagram:

74LS266 IC contains four 2-input XOR gates. The connection diagram for this IC is shown below:

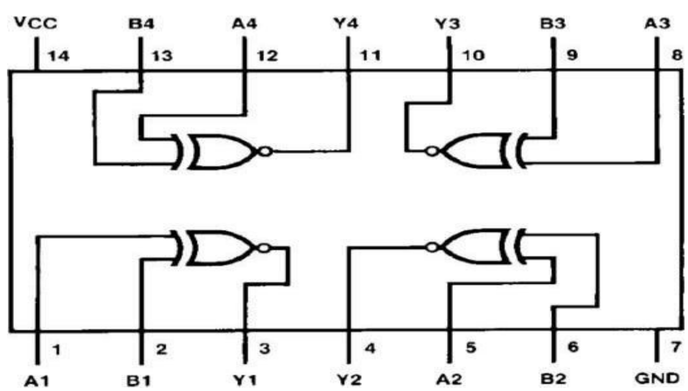


Figure 6 XNOR Gate Connection diagram

XNOR Gate can also be implemented by using primary gates as follows.

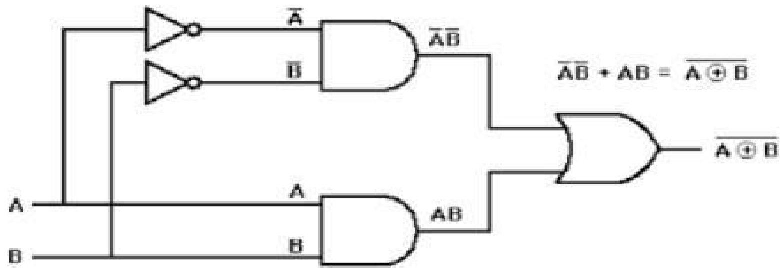


Figure 7 XNOR Gate implementation using primary gate

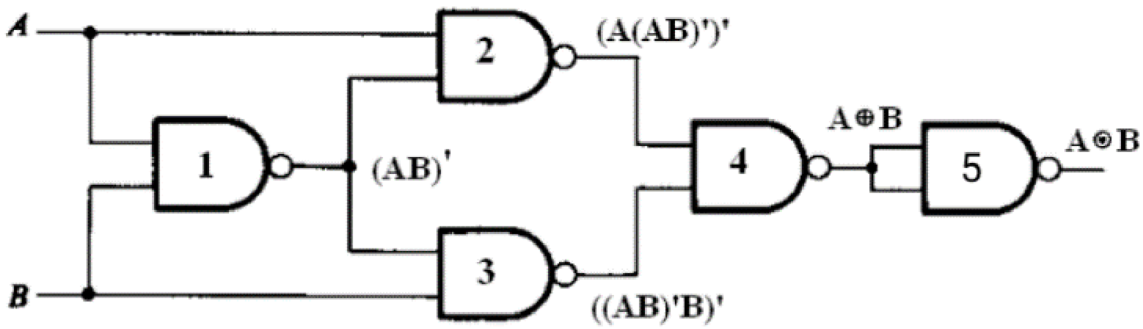


Figure 8 XNOR Gate implementation using universal gate

Boolean Algebra:

When a Boolean expression is implemented with logic gates, each term requires a gate, and each variable within the term designates an input to the gate. **Boolean algebra is applied to reduce an expression for obtaining a simpler circuit.** A Boolean function can be written in a variety of ways when expressed algebraically. There are, however, a few ways of writing algebraic expressions that are considered to be standard forms.

The standard forms contain product terms and sum terms. An example of a product term is XYZ. This is a logical product consisting of an AND operation among three literals. An example of a sum term is X+Y+Z. This is a logical sum consisting of OR operation among the literals.

Rules and Law of Boolean Algebra:

i. **Commutative law**

   Commutative law states that the inter-changing of the order of operands in a Boolean equation does not change its result.

   a. Using OR operator → A + B = B + A
   b. Using AND operator → A * B = B * A

Figure 9 Commutative law in AND Gate

ii. **<u>Associative Law</u>**

a) **Associate Law of Addition:**

Associative law of addition states that OR more than two variables i.e., mathematical addition operation performed on variables will return the same value irrespective of the grouping of variables in an equation. It involves in swapping of variables in groups. The Associative law using OR operator can be written as

$$A+(B+C) = (A+B)+C$$

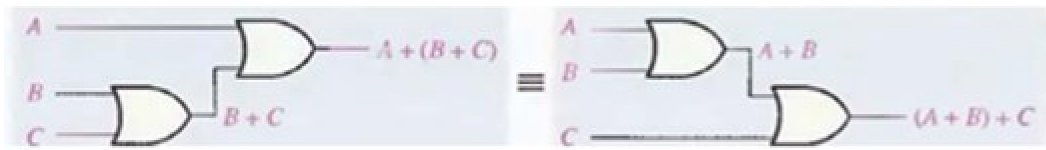


Figure 10 Application of Associative law of addition

b) **Associate Law of Multiplication**

Associative law of multiplication states that AND more than two variables i.e. mathematical multiplication operation performed on variables will return the same value irrespective of the grouping of variables in an equation. The Associative law using AND operator can be written as

$$A * (B * C) = (A * B) * C$$

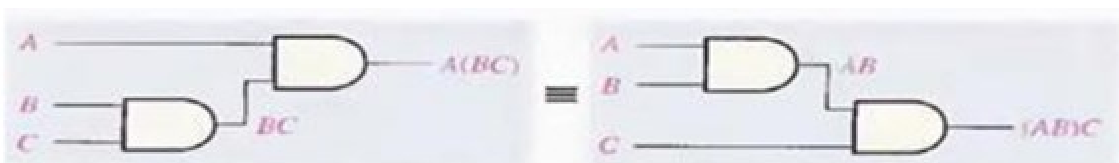


Figure 11 Application of Associative law of Multiplication

iii. **Distributive law**

This is the most used and most important law in Boolean algebra, which involves in 2 operators: AND, OR. The multiplication of two variables and adding the result with a variable will result in same value as multiplication of addition of the variable with individual variables. Distributive law can be written as

$$A + BC = (A + B) (A + C)$$

This is called OR distributes over AND.

The addition of two variables and multiplying the result with a variable will result in same value as addition of multiplication of the variable with individual variables. Distributive law can be written as

$$A (B+C) = (A B) + (A C)$$

This is called AND distributes over OR.

Figure 12 Application of Distributive law of Multiplication over addition and vice-versa

Lab Tasks

Lab Task#1:

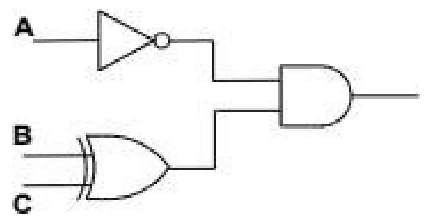
a) Design XOR Logic Gate on Logic Works using Primary Gates only.

b) Design XNOR Logic Gate on Logic Works using Primary Gates only.

c) Design XNOR Logic Gate on Logic Works using NAND Gates only.

Lab Task#2:

Write the Boolean expression for the logic circuits in the Figure. Also implement the given circuits on Logic Works and draw Truth table:



Lab Task#3:

For the Boolean function $F = \bar{A}\bar{B} + A\bar{D} + C$, draw logic circuit diagram using primary gates on logic works and draw its truth table.

Lab Task#4:

Implement the following scenario/ Logic on Logic Works

A device is needed to indicate when two LOW levels occur simultaneously on its inputs and to produce a HIGH output as an indication. Specify the device.

Lab Task#5:

Implement the following scenario/ Logic on Logic Works

As part of an aircraft's functional monitoring system, a circuit is required to indicate the status of the landing gears prior to landing. A green LED display turns on if all three gears are properly extended when the "gear down" switch has been activated in preparation for landing.

A red LED display turns on if any of the gears fail to extend properly prior to landing. When a landing gear is extended, its sensor produces a LOW voltage. When a landing gear is retracted, its sensor produces a HIGH voltage. Implement a circuit to meet this requirement.

Lab Task#6:

Implement the following scenario/ Logic on Logic Works

A certain system contains two identical circuits operating in parallel. As long as both are operating properly, the outputs of both circuits are always the same. If one of the circuits fails, the outputs will be at opposite levels at some time. Devise a way to monitor and detect that a failure has occurred in one of the circuits.