Digital Logic Design
(EL-1005)
LABORATORY MANUAL
Spring-2022

LAB 04
Advance Logic Gate and Boolean Algebra

_____ _____ ___

STUDENT NAME ROLL NO SEC

_____

INSTRUCTOR SIGNATURE & DATE

MARKS AWARDED: /2

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

Date: 21st Feb 2022

Lab Session 04: Advance Logic Gate and Boolean algebra

OBJECTIVES:

The objectives of this lab is:

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

APPARATUS:

- Logic trainer
- Logic probe

COMPONENTS:

ICs 74LS02, 74LS00, 74LS08, 74LS32, 74LS04, 74LS86, Jumper Wire

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:

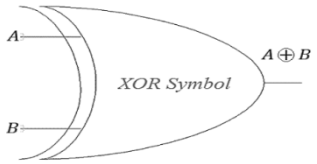


Figure 2 Exclusive-OR Gate Symbol

Function Table:

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

Table: 1 XOR Gate Truth Table

H= Logic High, L= Logic Low

Connection Diagram:

7486 IC contains four 2-input XOR gates. The connection diagram for this IC are shown below:

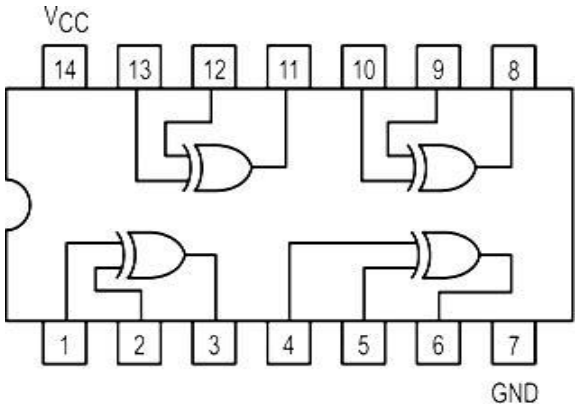
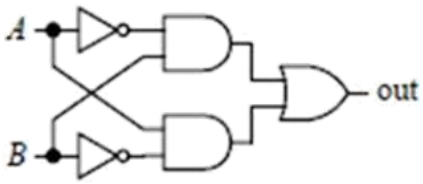


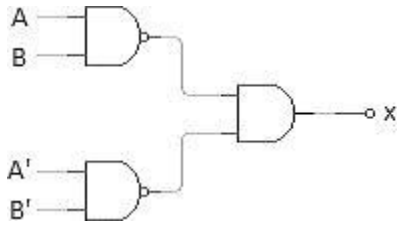
Figure 2 XOR Gate Connection diagram

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



$A \oplus B = A'B+AB'$

Figure 3 XOR Gate using basic Gate



$A \oplus B = (AB)'.(A'B')'=(AB)'.A+B$

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:

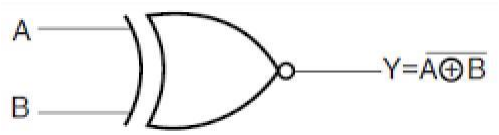


Figure 5 Exclusive-NOR Gate Symbol

Function Table:

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

Table: 2 XNOR Gate Truth Table

H= Logic High, L= Logic Low

Connection Diagram:

74266 IC contains four 2-input XOR gates. The connection diagram for this IC are shown below:

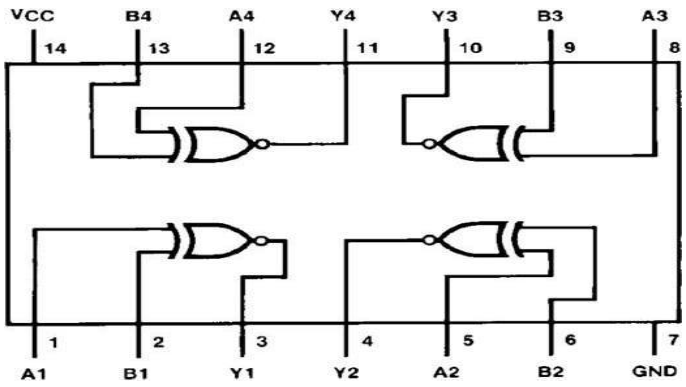


Figure 6 XNOR Gate Connection diagram

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

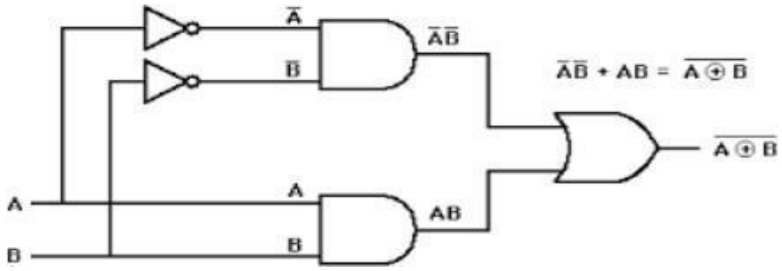


Figure 7 XNOR Gate implementation using primary gate

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 8 Commutative law in AND Gate

ii. Associative Law

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

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

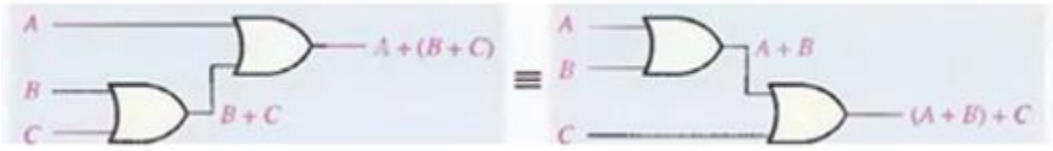


Figure 9 Application of Associative law of addition

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

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

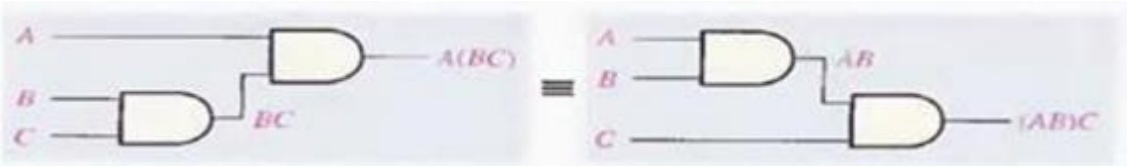


Figure 10 Application of Associative law of Multiplication

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 11 Application of Distributive law of Multiplication over addition and vice-versa

Report for Experiment 04

Name_____**StudentID**_____**Section**_____**Date**_____

Exercise#01 Implement the circuit for XOR Logic circuit using 74LS86 on Trainer

Exercise#02 Implement the circuit for XNOR Logic circuit using Basic Gates (AND-OR-NOT) on Trainer

Exercise#03 Implement the circuit for AND – OR and NOT Logic circuit using universal Gates (NAND or NOR Gates only) on Trainer.

Exercise#04 Implement Even Parity and Odd Parity on Trainer.

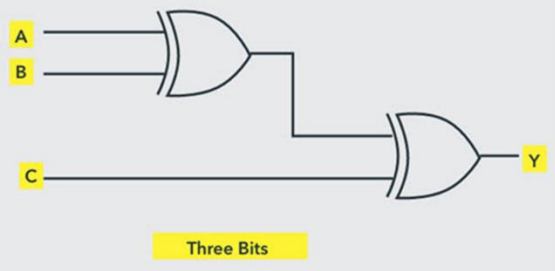
Parity Generator and Checker

A Parity Generator is a combinational logic circuit that generates the parity bit in the transmitter. On the other hand, a circuit that checks the parity in the receiver is called Parity Checker. A combined circuit or device of parity generators and parity checkers are commonly used in digital systems to detect the single bit errors in the transmitted data.

Even Parity and Odd Parity

The sum of the data bits and parity bits can be even or odd. In even parity, the added parity bit will make the total number of 1s an even number, whereas in odd parity, the added parity bit will make the total number of 1s an odd number.

The basic principle involved in the implementation of parity circuits is that sum of odd number of 1s is always 1 and sum of even number of 1s is always 0. Such error detecting and correction can be implemented by using Ex-OR gates (since Ex-OR gate produce zero output when there are even number of inputs).



3-bit message			Even parity bit generator (P)
A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Even Parity Generator

Let us assume that a 3-bit message is to be transmitted with an even parity bit. Let the three inputs A, B and C are applied to the circuit and output bit is the parity bit P. The total number of 1s must be even, to generate the even parity bit P.

The figure shows the truth table of even parity generator in which 1 is placed as parity bit in order to make all 1s as even when the number of 1s in the truth table is odd.