Digital Logic Design (EL-1005)

LABORATORY MANUAL

Spring-2022

LAB 03

Universal Logic Gates - Advance Logic Gate and Boolean Algebra

STUDENT NAME ROLL NO SEC

INSTRUCTOR SIGNATURE & DATE

MARKS AWARDED: /02

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

Date: 14th Feb 2022

Lab Session 03: Universal Logic Gates - Advance Logic Gate and Boolean algebra

OBJECTIVES:

The objectives of this lab is:

- To study the realization of basic gates using universal gates (NAND gate & NOR gate)
- To learn technology mapping (NAND-NAND & NOR-NOR implementation) and its significance in order to obtain cost effective circuit for implementation

APPARATUS:

- Logic trainer
- Logic probe

COMPONENTS:

ICs 74LS02, 74LS00, Jumper Wire

Introduction:

The design of a combinational circuit starts from the specification of the problem and culminates in a logic diagram or net-list that describes a logic diagram. The procedure involves the specification, formulation, optimization, & technology mapping.

Technology mapping is actually transformation of logic diagram or net-list to a new diagram using the available implementation technology. Typically NAND and NOR gates are more desirable to use in technology mapping due to the following reasons:

1. NAND and NOR gates are said to be universal gates where universal gate is a gate which can implement any Boolean function without needing any other type of gate.
2. Using universal gate in technology mapping may further reduce cost of optimized logic diagram.
3. Universal gates are easier to fabricate with electronic components.

A convenient way to implement a Boolean function with NAND gates only (NAND-NAND implementation) is to begin with the optimized logic diagram of the circuit consisting of AND, OR and NOT gates. The function is converted to pure NAND logic by replacing each gate in logic diagram with its representation using NAND gates only as shown in figure 5-1. After that, all inverter pairs are cancelled. The same conversion procedure is applied to implement a Boolean function with NOR gates only (NOR-NOR implementation).

Universal Logic Gates:

A. **NAND Gate:**

“It is a device whose output is 1 if at least one or all of the inputs are low (0)”

Symbol:

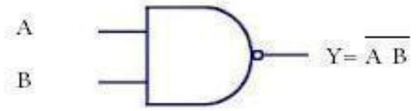


Figure 2 NAND Gate Symbol

Function Table:

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

Table: 1 NAND Gate Truth Table

H= Logic High, L= Logic Low

Connection Diagram:

74LS00 IC contains four 2-input NAND gates. The connection diagram for this IC are shown below:

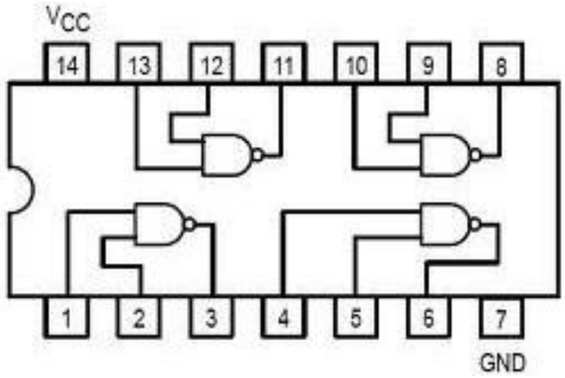


Figure 2 NAND Gate Connection diagram

B. **NOR Gate:**

"It is a device whose output is 1 if all the given inputs are low (0)".

Symbol:

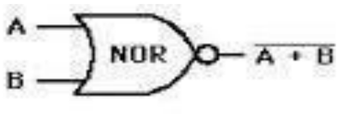


Figure 3 NOR Gate Symbol

Function Table:

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

Table: 2 NOR Gate Truth Table

H= Logic High, L= Logic Low

Connection Diagram:

74LS02 IC contains four 2-input NOR gates. The function table and connection diagram for this IC are shown below:

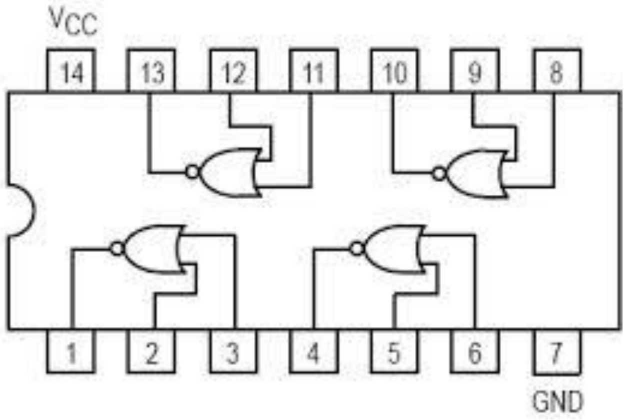


Figure 4 NOR Gate Connection diagram

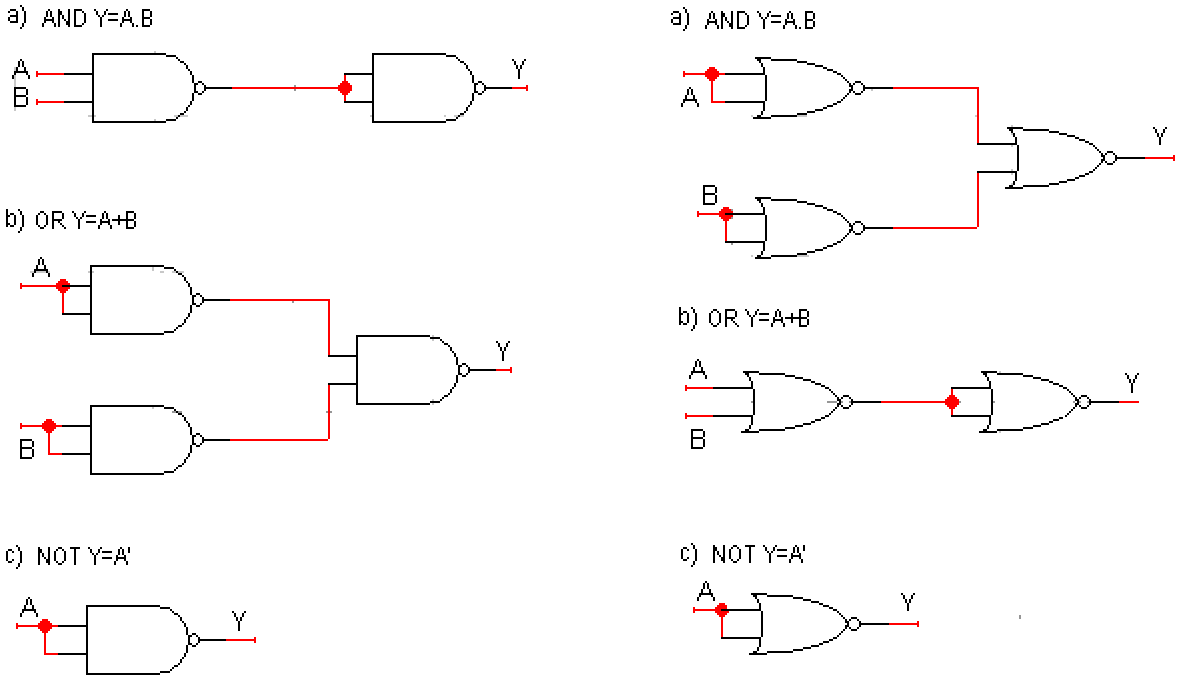


Figure 5 NAND-NAND and NOR-NOR representation of basic logic gates

Application of Universal Gates:

1. Burglar alarm

When the switch is closed one input of the NAND gate is LOW. When the LDR is in the light the other input is LOW. This means that if either of these things happen, i.e. the switch is closed or the light is on one of the inputs is LOW, the output is HIGH and the buzzer sounds.

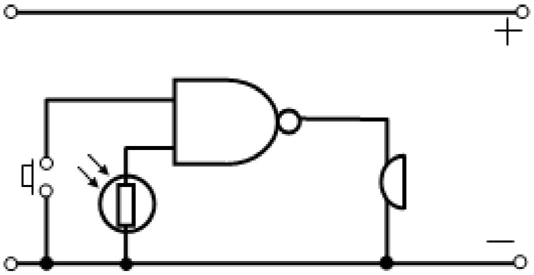


Figure 6 NAND- Gate Based Burglar alarm

2. Freezer warning buzzer

When the thermistor is COLD its resistance is LARGE and the input to the NAND gate is high. Since the NAND gate is connected as an INVERTER the output is LOW. As the thermostat warms up its resistance decreases, the voltage across it falls and the input to the NAND gate falls.

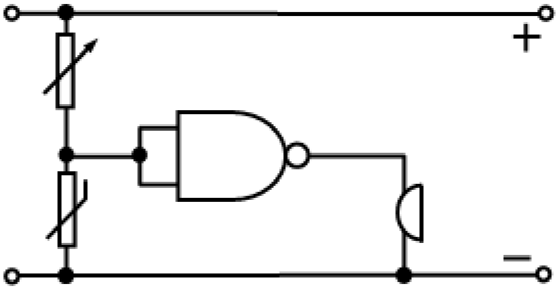


Figure 7 NAND- Gate Based Freezer warning buzzer

3. Car Door Open System of an Automobile

A car needs to be designed that the driver gets a visual indication if any of the doors of the car is open so that it helps to avoid accident and injury to the passengers. Assuming there are two doors (just for simplicity, it works for more doors as well) where this system is fitted, the circuit can be designed using a NAND gate as follows You can see from the figure that when any of the switches is open due to the door position, the NAND gate energies the lamp inside the car, hence warning the driver.

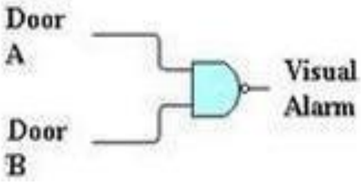


Figure 8 NAND- Gate Based Car Open System of an Automobile

Advance Logic Gate and Boolean algebra

OBJECTIVES:

- 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, 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:

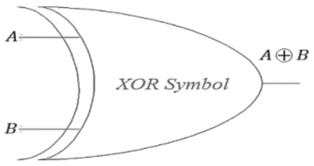


Figure 2 Exclusive-OR Gate Symbol

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:

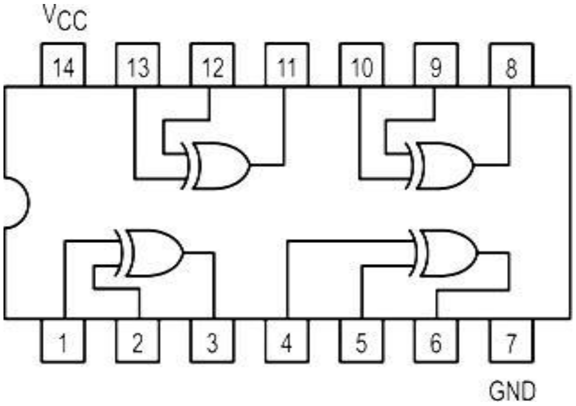
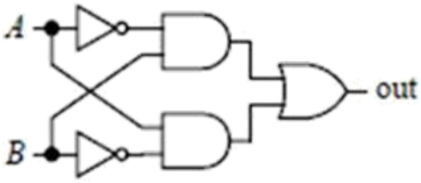


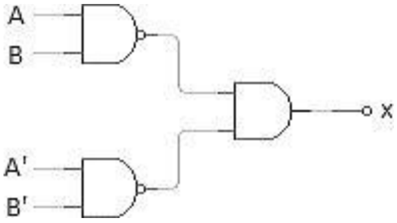
Figure 2 XOR Gate Connection diagram

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



A⊕ B = A'B+AB'

Figure 3 XOR Gate using basic Gate



A⊕ 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:

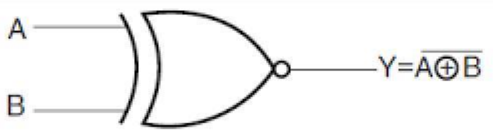


Figure 5 Exclusive-NOR Gate Symbol

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:

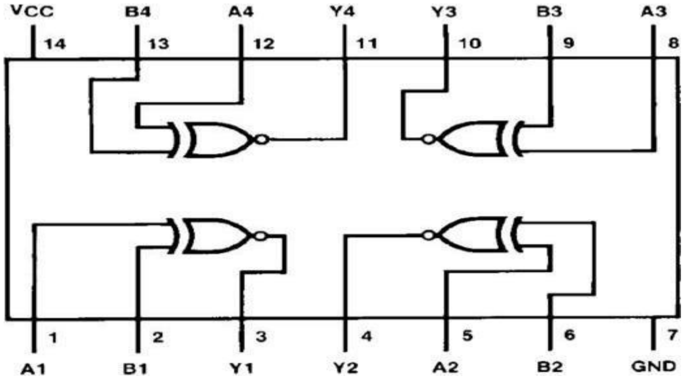


Figure 6 XNOR Gate Connection diagram

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

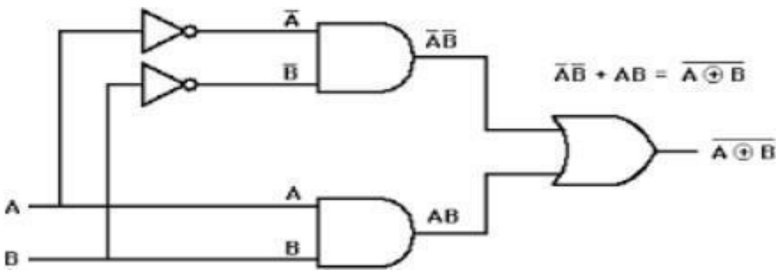

$\overline{A}$
$\overline{B}$
$\overline{A}\overline{B}$
$\overline{A}\overline{B} + AB = \overline{A \oplus B}$
$\overline{A \oplus B}$
A
B
A
B
AB

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 → $A * B = B * A$

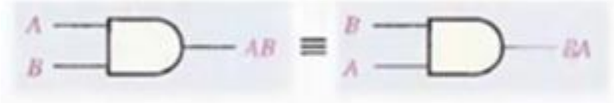


Figure 8 Commutative law in AND Gate

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

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

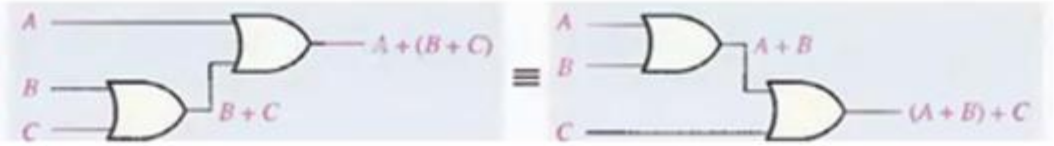


Figure 9 Application of Associative law of addition

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

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

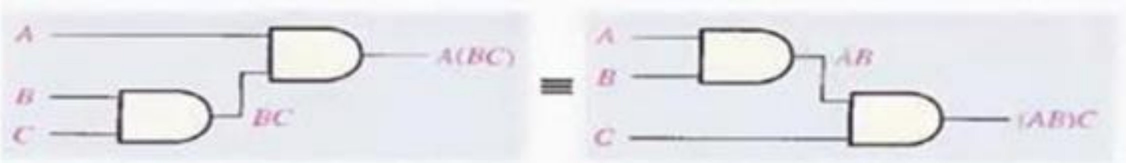


Figure 10 Application of Associative law of Multiplication

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.

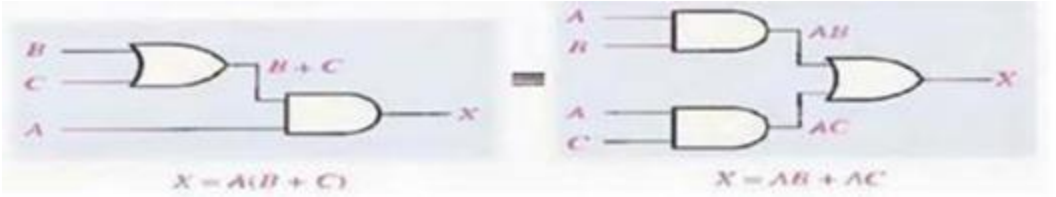


Figure 11 Application of Distributive law of Multiplication over addition and vice-versa

Report for Experiment 03

Name______________________StudentID__________________Section________Date__________

Exercise#01 Implement the Secondary/Advanced Logic (2-INPUT) Gates on Logic Works.

Exercise#02 Implement the Secondary/Advanced Logic (2-INPUT) using IC on Logic Works.

Exercise#03 Design the circuit for XOR Logic circuit using Basic Gates (AND-OR-NOT) on Logic Works.

Exercise#04 Design the circuit for XNOR Logic circuit using Basic Gates (AND-OR-NOT) on Logic Works

Exercise#05 Implement the following scenario/ Logic on Logic Works

Two tanks store certain liquid chemicals that are required in a manufacturing process. Each tank has a sensor that detects when the chemical level drops to 25% of full. The sensors produce a HIGH level of 5 V when the tanks are more than one-quarter full. When the volume of chemical in a tank drops to one-quarter full, the sensor puts out a LOW level of 0 V.

It is required that a single red light-emitting diode (LED) on an indicator panel show when both tanks are more than one-quarter full. Show how a NAND gate can be used to implement this function.

Exercise#06 Analysis and Design Logic Circuit on Logic Works for the following scenario/ Logic.

For the process described in Exercise 01 it has been decided to have a red LED display come on when at least one of the tanks falls to the quarter-full level rather than have the green LED display indicate when both are above one quarter. Design circuit on logic works that shows how this requirement can be implemented.

Exercise#07 Analysis and Design Logic Circuit on Logic Works for the following scenario/ Logic.

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.

Exercise#08 Design the circuit following scenario/ Logic on 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 **design the circuit following scenario/ Logic on Logic Works**.

TRUE/FALSE

1. An inverter performs a NOT operation.

2. A NOT gate cannot have more than one input.

3. If any input to an OR gate is zero, the output is zero.

4. If all inputs to an AND gate are 1, the output is 0.

5. A NAND gate can be considered as an AND gate followed by a NOT gate.

6. A NOR gate can be considered as an OR gate followed by an inverter.

7. The output of an exclusive-OR is 0 if the inputs are opposite.