Digital Logic Design (EL-1005)

LABORATORY MANUAL
Spring-2022

LAB 02
Basic Logic Gates

Syed Muhammad Huzaifa
STUDENT NAME

21K-4948
ROLL NO

BSCS-2J
SEC

INSTRUCTOR SIGNATURE & DATE

MARKS AWARDED: /02

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES (NUCES), KARACHI
Date: 7th Feb 2022

OBJECTIVES:

The objectives of this lab is:

- To learn and understand the working of basic gates (AND, OR, NOT).
- To learn and understand troubleshooting of logic circuits
- Identify gates by their symbols
- Write logical expressions of gates and draw their truth tables

APPARATUS:

- Logic trainer
- Logic probe

COMPONENTS:

ICs 74LS08, 74LS32, 74LS04, Jumper Wire

Introduction:

Digital circuits are the electronic circuits that manipulate binary information. Logic gates are the basic building blocks in constructing digital circuits. Logic gate has one output and one or more inputs. Each logic gate performs a specific logical operation. The interconnection between inputs and outputs of gates form a digital circuit. Any digital circuit can implemented using three basic logical operations called AND, OR, and NOT. That is why AND, OR, and NOT gates are referred as basic logic gates. AND OR logic functions exhibit the phenomenon of dominance. In both cases, there is an input value that will force the output of the gate to a known value regardless of the state of other inputs. This value is known as the dominant value of the gate. The dominant value of an AND gate is zero, while the dominant value of an OR gate is one. The **Output** of logic gate also provides two nominal values of voltage only_e.g. 0V and 5V representing logic 0 and logic 1 respectively. There is always a time delay between an input being applied and the output responding

In this experiment, we will use 74LS08, 74LS32, and 74LS04 ICs for the implementation of AND, OR, and NOT logical operations.

And Gate IC And Its Functionality:

74LS08 IC contains four 2-input AND gates. AND gate is an electronic circuit that gives a high output only when all of its inputs are high & if there is a single input which is low its output is low. So basically AND gate works on the principle of multiplication.

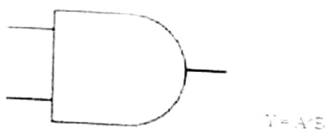


Figure 1: AND Gate Symbol

Function Table:

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

Table 1: AND Gate Truth Table

H= Logic High, L= Logic Low

Procedure:

1. Install the IC 74LS08 on the ETS-7000 Trainer's breadboard.
2. Wire the circuit according to the IC's pin configuration given in its data sheet. 3. Use the logic switches S0 and S1 for input to the AND gate.
4. For output indication use any one of the LEDs
5. Supply the VCC = +5V and GND as indicated in the diagram
6. Test at least two gates of the IC being used
7. Test the output for all possible combination of inputs and record your results in following Truth Table

Connection Diagram:

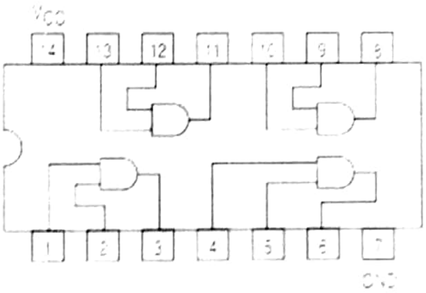


Figure 2: AND Gate IC Configuration

OR Gate IC And Its Functionality:

74LS32 IC contains four 2-input OR gates. OR gate is an electronic circuit that gives a high output if one or more of its inputs are high & gives low output when all of its inputs are low. So basically OR gate works on the principle of addition.

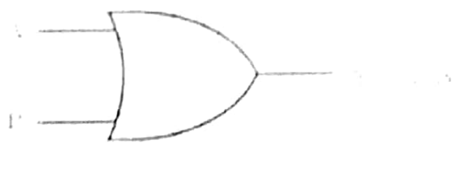


Figure 3: OR Gate Symbol

Scanned with CamScanner

Function Table:

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

Table 2: OR Gate Truth Table

H= Logic High, L= Logic Low

Procedure:

1. Install the IC 74LS32 on the ETS-7000 Trainer's breadboard.
2. Wire the circuit according to the ICs pin configuration given in its data sheet. 3. Use the logic switches S0 and S1 for input to the AND gate.
3. For output indication use any one of the LEDs
4. Supply the VCC = +5V and GND as indicated in the diagram
5. Test at least two gates of the IC being used
6. Test the output for all possible combination of inputs and record your results in following Truth Table

Connection Diagram:

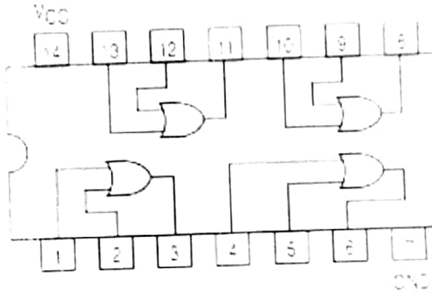


Figure 4: OR Gate IC Configuration

NOT Gate IC And Its Functionality:

74LS04 IC contains Six 1-input NOT gates. NOT gate is an electronic circuit that is used to invert a digital logic, hence called as an **inverter**. It always has exactly a single input and a single output. Whatever logical state is applied to the input, the opposite state will appear at the output.

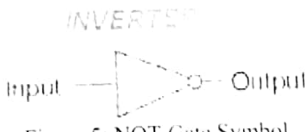


Figure 5: NOT Gate Symbol

Function Table:

Input	Output
A	Y
L	H
H	L

Table 3: NOT Gate Truth Table

H= Logic High, L= Logic Low

Procedure:

1. Install the IC 74LS04 on the ETS-7000 Trainer's breadboard.
2. Wire the circuit according to the ICs pin configuration given in its data sheet. 3. Use the logic switches S0 and S1 for input to the AND gate.
3. For output indication use any one of the LEDs
4. Supply the VCC = +5V and GND as indicated in the diagram
5. Test at least two gates of the IC being used
6. Test the output for all possible combination of inputs and record your results in following Truth Table

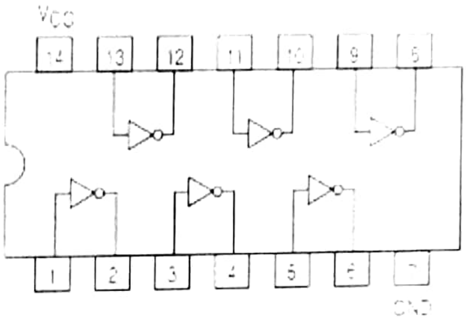


Figure 6: NOT Gate IC Configuration

Testing of ICs:

Before starting implementation of a specific logic circuit, all basic gate ICs should be tested in order to make sure that the ICs are working properly. Using the function table (truth table) for each gate, in a particular IC, apply all input combinations one by one and check its output logic level on LED.

Troubleshooting:

After testing all required number and type of ICs we need to implement a digital circuit, we start implementing the circuit on logic trainer. Once we complete the implementation, we need to test the output of the circuit to make sure that whether the circuit is working accurately or not. Using the truth table that represents the functionality of the logic circuit, we apply all input combinations one by one and check its output logic level on LED.

Lab Task:

Lab Task#1: Implement the following Basic Logic AND logic – OR Logic and Inverter Circuit on logic trainer

Lab Task#2: Implement the following logic circuit on logic trainer, and write truth table in the space provided below:

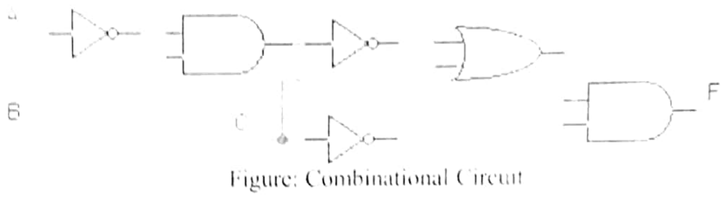


Figure: Combinational Circuit

A	B	C	$\bar{A}$	$\overline{\bar{A}\cdot B}$	$\overline{\bar{A}\cdot B + C}$	$\bar{C}$	F
0	0	0	1	1	0	1	1
0	0	1	1	1	1	0	0
0	1	0	1	0	1	0	1
0	1	1	1	0	1	1	0
1	0	0	0	1	0	1	1
1	0	1	0	1	1	0	0
1	1	0	0	1	0	1	1
1	1	1	0	1	1	0	0

Lab Task#3 Write the Boolean expression for each of the logic circuits in Figure Also implement the given circuits on breadboard and draw Truth tables:

Figure: Combinational Circuit

1

A	B	C	A·B	A+B+C
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	0	1
1	0	0	0	0
1	0	1	0	1
1	1	0	0	0
1	1	1	1	1

2

A	$\bar{A}$	B	$\bar{A}\cdot B$	$\overline{\bar{A}\cdot B}$
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1

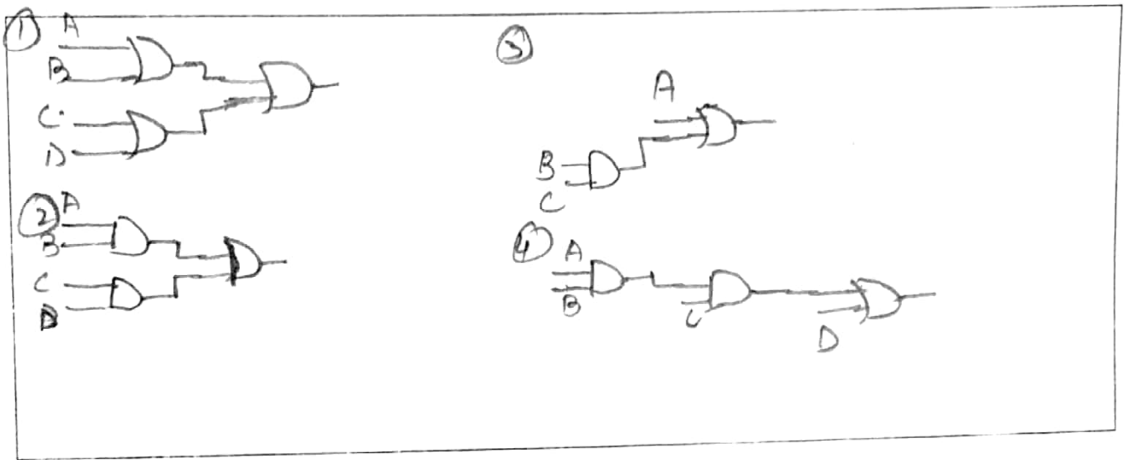
3

A	B	C	A+B	A+B·C
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	1	0
0	0	1	0	0
0	1	1	1	1
1	0	1	1	1
1	1	1	1	1

Lab Task#4

Draw the logic circuit represented by each of the following expression:

1. A+B+C+D
2. ABCD
3. A+BC
4. ABC+D



Lab Task#5: Implement the following logic circuit on logic trainer, and write truth table in the space provided below:

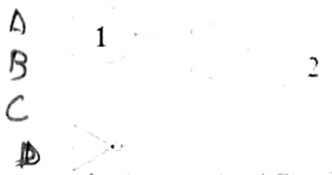


Figure: Combinational Circuit

A	B	C	D	A•B	A·B+C	D'	(A·B+C)+D'
0	0	0	0	0	0	1	1
0	0	0	1	0	0	0	0
0	0	1	0	0	1	1	1
0	0	1	1	0	1	0	1
0	1	0	0	0	0	1	1
0	1	0	1	0	0	0	0
0	1	1	0	0	1	1	1
0	1	1	1	0	1	0	1
1	0	0	0	0	0	1	1
1	0	0	1	0	0	0	0
1	0	1	0	0	1	1	1
1	0	1	1	0	1	0	1
1	1	0	0	1	1	1	1
1	1	0	1	1	1	0	1
1	1	1	0	1	1	1	1
1	1	1	1	1	1	0	1