

**LAB # 2**



**DATED:**

**17th October, 2022**

**SUBMITTED TO:**

**Engr. Rehmat Ullah**

**CSE-202L Digital Logic Design Lab**

**Fall 2022**

**SUBMITTED BY:**

**Ali Asghar(21PWCSE2059)**

**Suleman Shah(21PWCSE1983)**

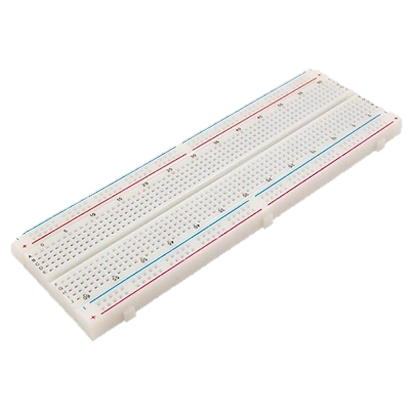
**Abu Bakar(21PWCSE2004)**

**Department of Computer Systems Engineering**

**University of Engineering & Technology, Peshawar**

STUDY OF BASIC GATES

CSE-202L: Digital Logic Design Laboratory

OBJECTIVE:

* To study basic logic gates

APPARATUS:

* Breadboard

Power Supply

* Power Supply

Breadboard

* Connecting Wires

COMPONENTS:

* IC’s
  + 7400 Quad-2-Input NAND Gate
  + 7402 Quad-2-Input NOR Gate

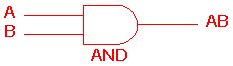
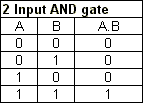
Connecting Wires

* + 7404 Hex Inverter
  + 7408 Quad-2-Input AND Gate
  + 7432 Quad-2-Input OR Gate
  + 7486 Quad-2-Input XOR Gate
* LED’s
* Switches

DIGITAL LOGIC GATES:

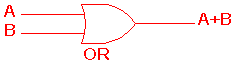
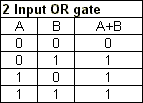
Digital systems are said to be constructed by using logic gates. These gates are the AND, OR, NOT, NAND, NOR, EXOR and EXNOR gates. The basic operations are described below with the aid of truth tables.

AND gate:

**** 

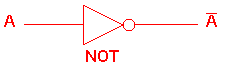
The AND gate is an electronic circuit that gives a high output (1) only if all its inputs are high.  A dot (.) is used to show the AND operation i.e., A.B.  Bear in mind that this dot is sometimes omitted i.e., AB

OR gate:

**** ****

The OR gate is an electronic circuit that gives a high output (1) if one or more of its inputs are high.  A plus (+) is used to show the OR operation.

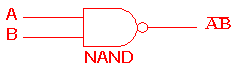
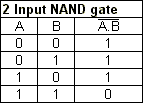
NOT gate:

**** ****

The NOT gate is an electronic circuit that produces an inverted version of the input at its output.  It is also known as an inverter.  If the input variable is A, the inverted output is known as NOT A.  This is also shown as A', or A with a bar over the top, as shown at the outputs. The diagrams below show two ways that the NAND logic gate can be configured to produce a NOT gate. It can also be done using NOR logic gates in the same way.

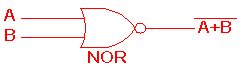
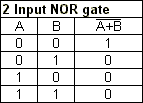
**http://www.ee.surrey.ac.uk/Projects/CAL/digital-logic/gatesfunc/graphics/NOT.gif**

NAND gate:

**** ****

This is a NOT-AND gate which is equal to an AND gate followed by a NOT gate.  The outputs of all NAND gates are high if any of the inputs are low. The symbol is an AND gate with a small circle on the output. The small circle represents inversion.

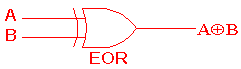
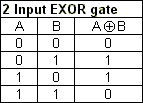
NOR gate:

**** ****

This is a NOT-OR gate which is equal to an OR gate followed by a NOT gate.  The outputs of all NOR gates are low if any of the inputs are high.

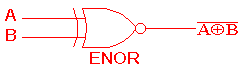
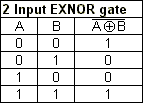
The symbol is an OR gate with a small circle on the output. The small circle represents inversion.

EXOR gate:

**** ****

The 'Exclusive-OR' gate is a circuit which will give a high output if either, but not both, of its two inputs are high.  An encircled plus sign (**http://www.ee.surrey.ac.uk/Projects/CAL/digital-logic/gatesfunc/graphics/enplus.gif**) is used to show the EOR operation.

EXNOR gate

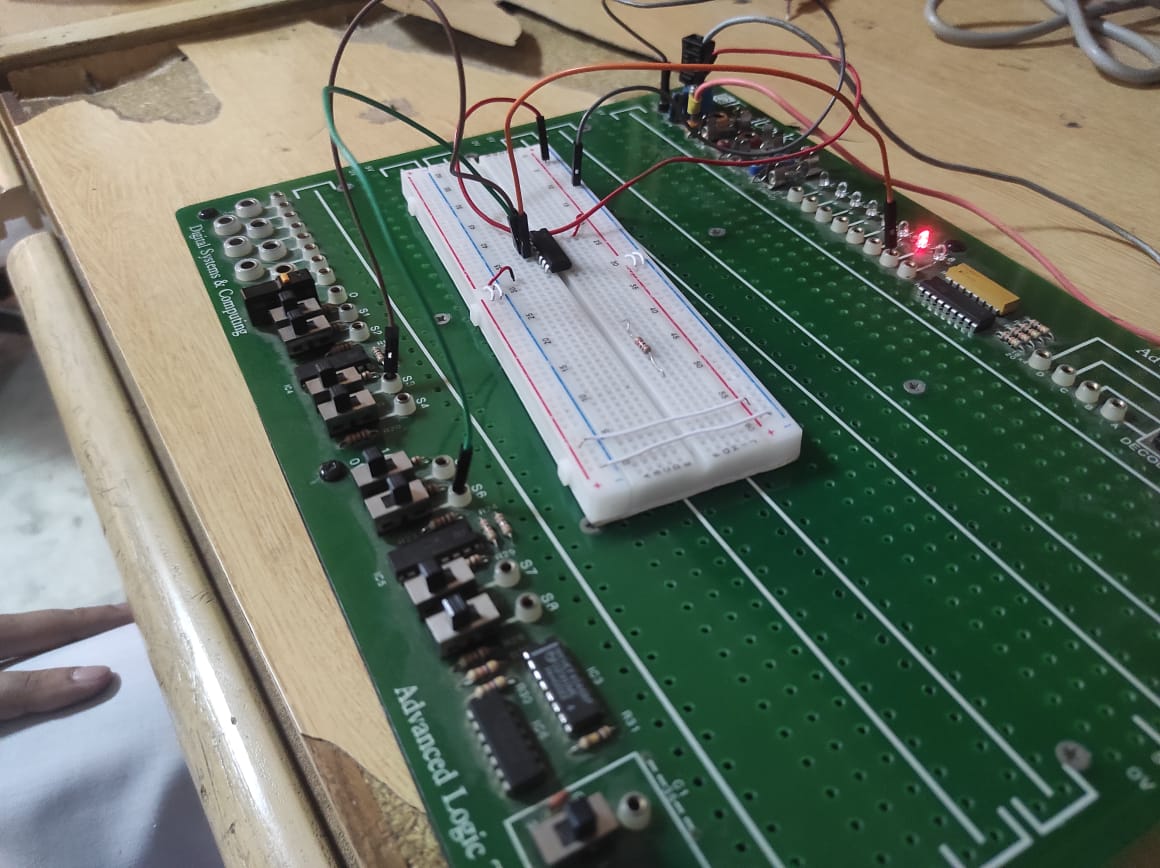
**** ****

The 'Exclusive-NOR' gate circuit does the opposite to the EOR gate. It will give a low output if either, but not both, of its two inputs are high. The symbol is an EXOR gate with a small circle on the output. The small circle represents inversion.

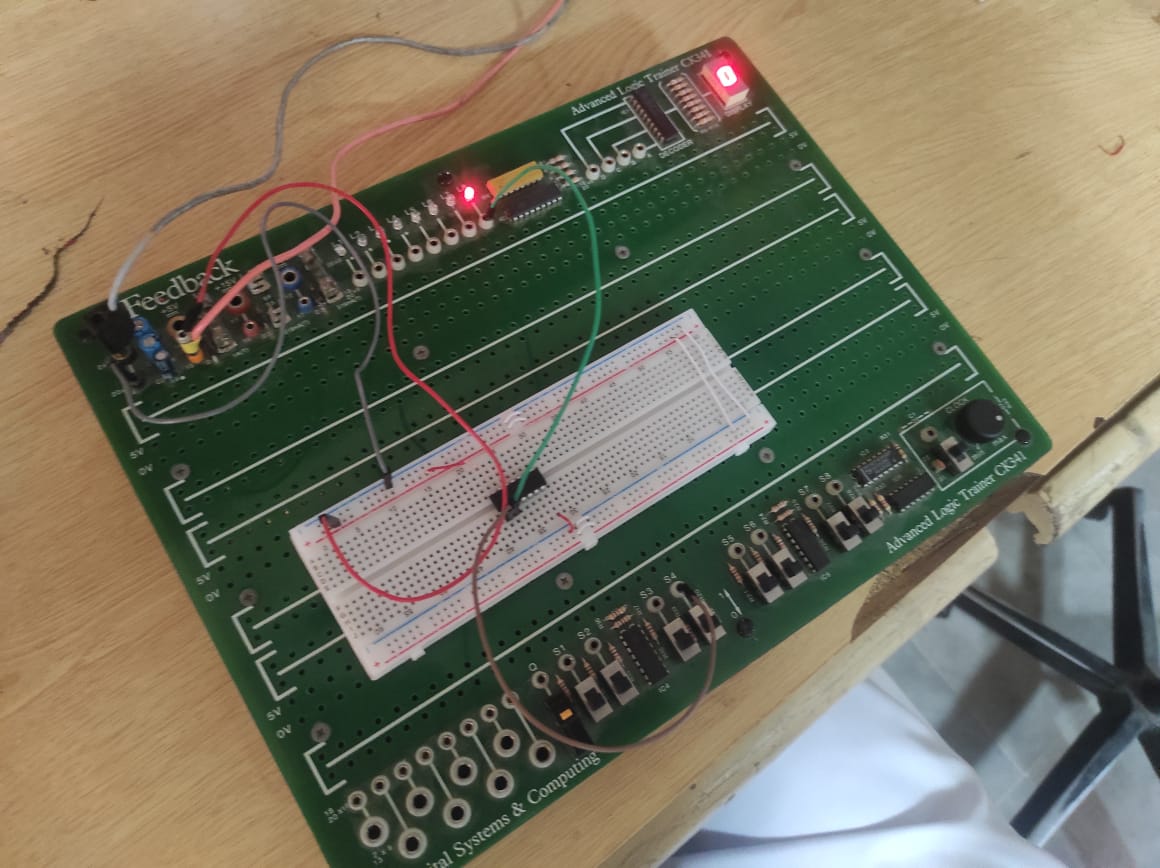
PIN OUT DIAGRAMS OF 74XX SERIES LOGIC IC’s

PROCEDURE:

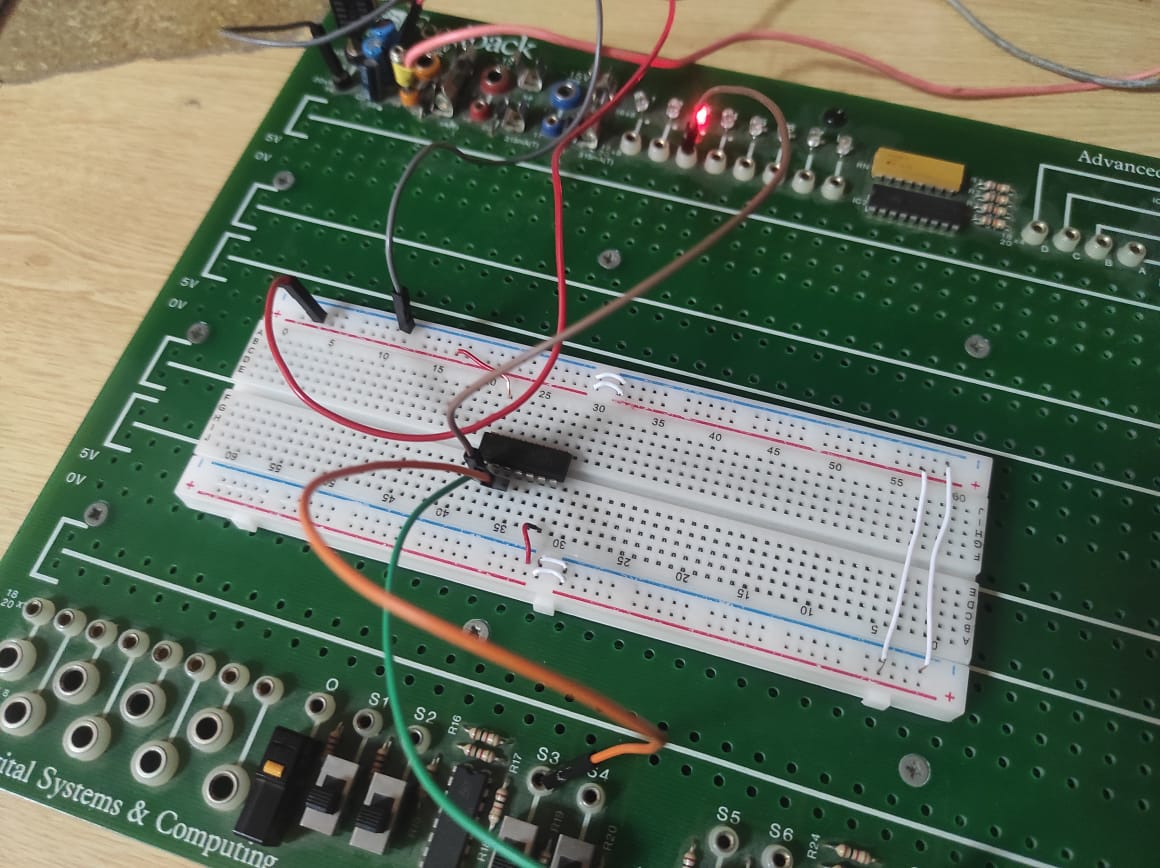
1. First of all, give biasing to the TTL IC by connecting pin # 7 to Ground and Pin # 14 to Vcc.
2. Then connect the input pins of the given IC to the switches.
3. After that, connect the output to any LED.
4. Output is considered 0 when LED is off. Conversely, it is considered 1 when LED is on.

LAB WORK:

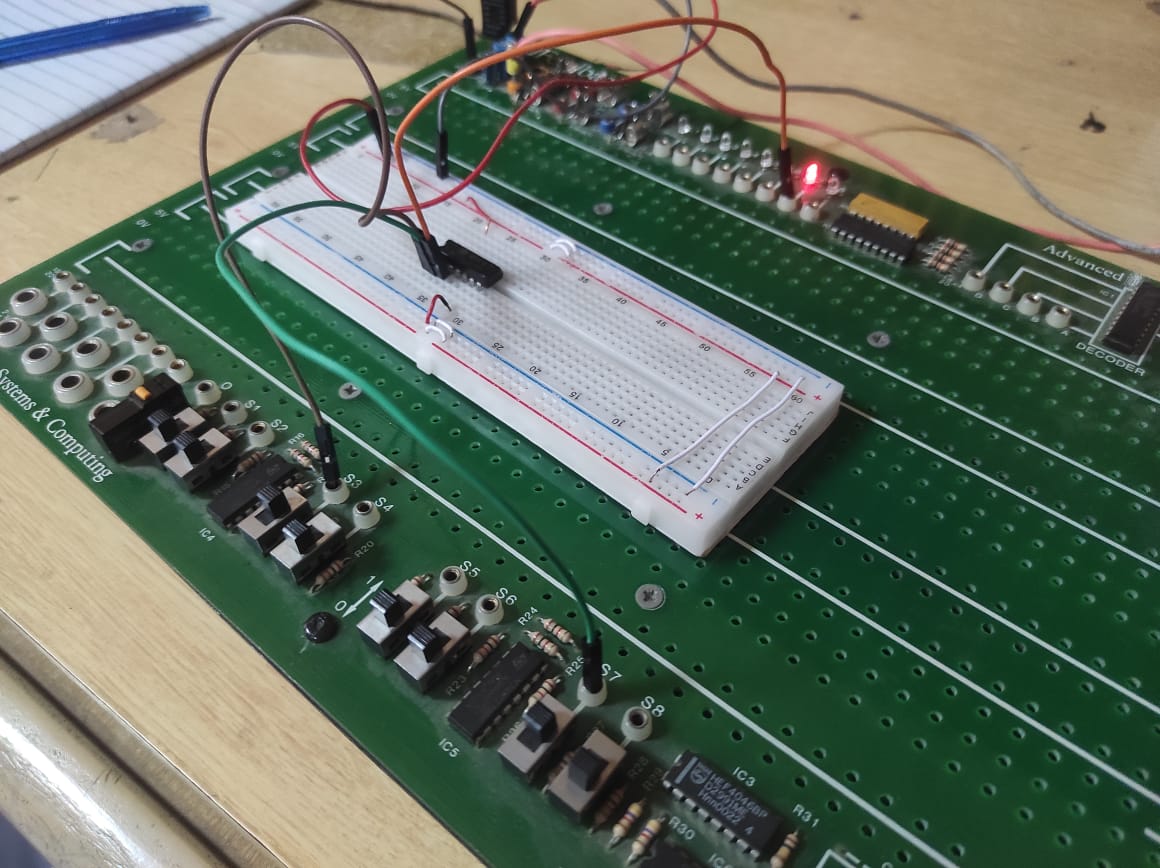
NAND Gate Implemented Using 7400 IC



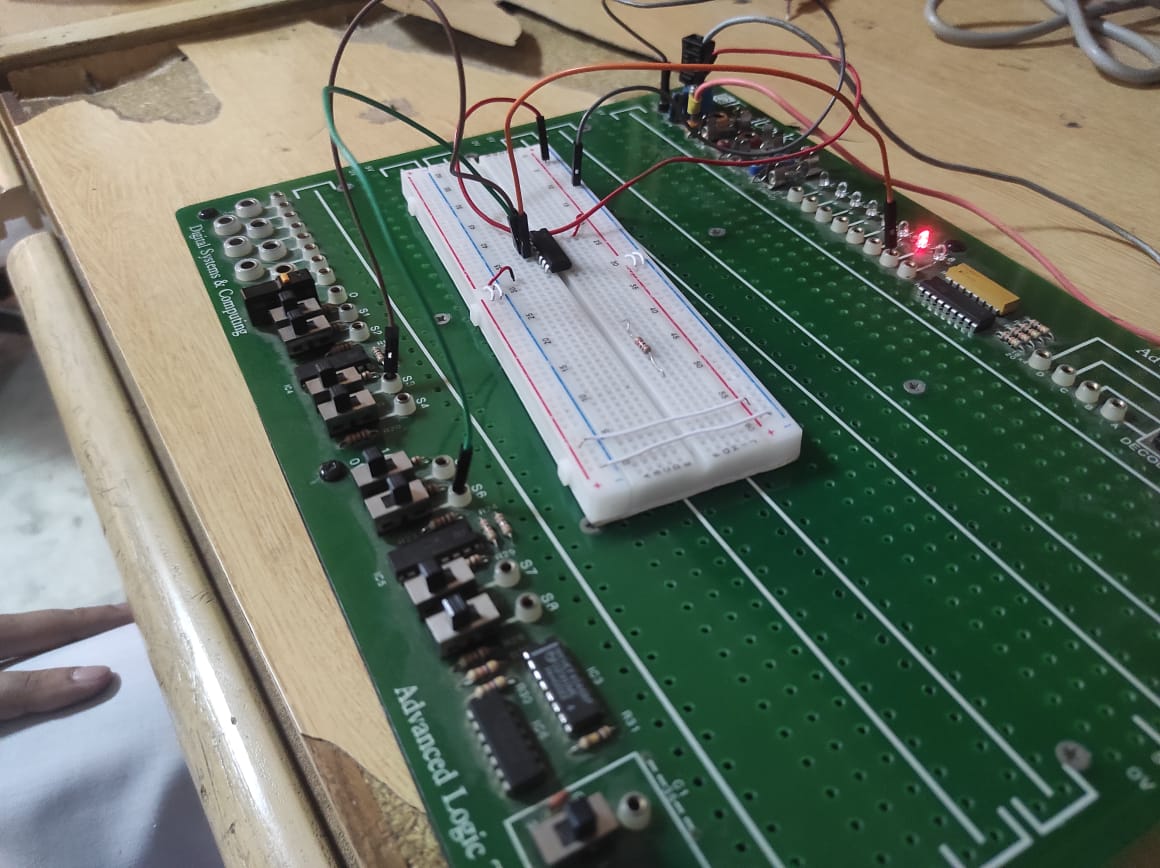
NOT Gate Implemented Using 7404 IC’s



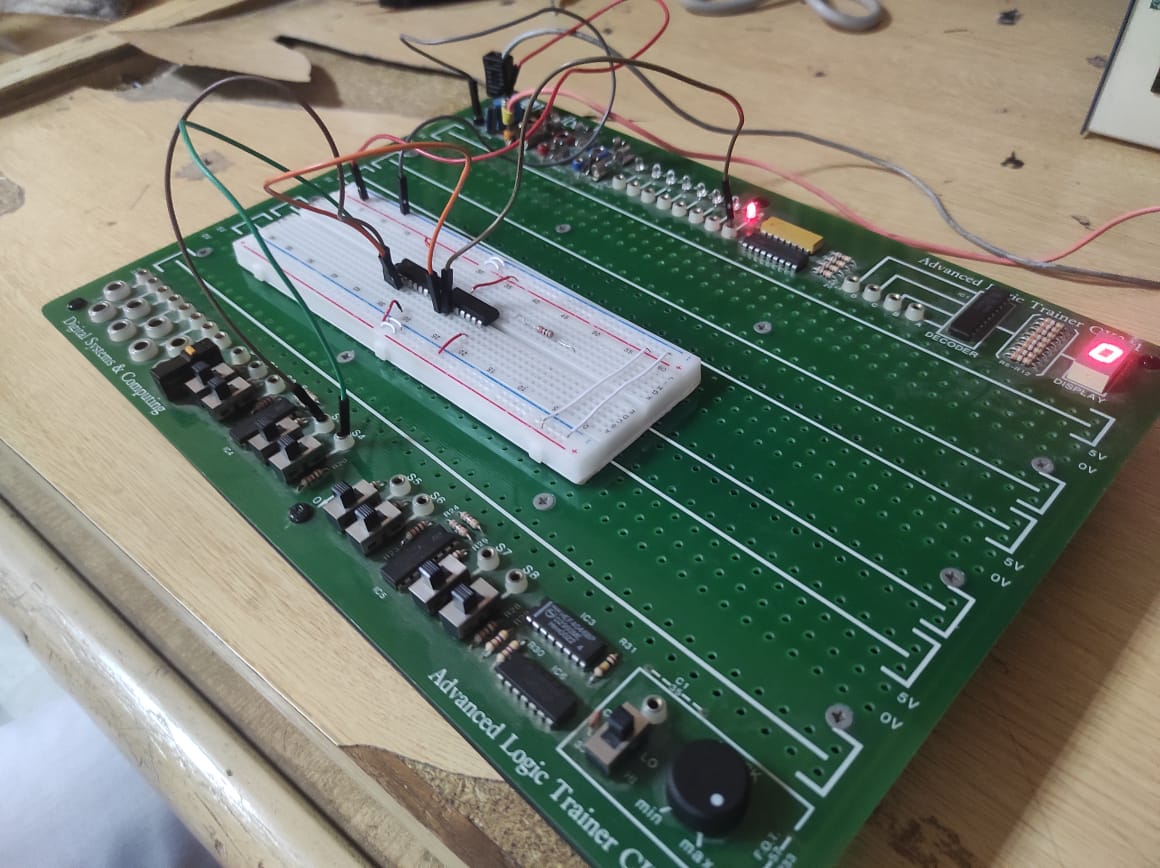
NOR Gate Implemented Using 7402 IC



AND Gate Implemented Using 7408 IC



OR Gate Implemented Using 7432 IC



XOR and XNOR Gate Implemented Using 7404 and 7486 IC’s

LAB READINGS:

Table 1: Truth Table for Inverter (NOT Gate)

|  |  |
| --- | --- |
| A (High/Low) | Y (High/Low) |
| Low | High |
| High | Low |

Table 2: Truth Table for AND & OR Gates

|  |  |  |  |
| --- | --- | --- | --- |
| A (H/L) | B (H/L) | AND2 (H/L) | OR2 (H/L) |
| Low | Low | Low | Low |
| Low | High | Low | High |
| High | Low | Low | High |
| High | High | High | High |

Table 3: Truth Table for NAND, NOR, & XOR Gates

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A (H/L) | B (H/L) | NAND2 (H/L) | NOR2 (H/L) | XOR2 (H/L) |
| L | L | High | High | Low |
| L | H | High | Low | High |
| H | L | High | Low | High |
| H | H | Low | Low | Low |

CONCLUSION:

By performing this experiment, we have successfully verified the truth tables of 2-Input Logic gates.

REVIEW QUESTIONS:

QUESTION#1:

A burglar alarm for a car has a normally LOW (grounded) switch on each of four doors. If any door is opened, the output of that switch goes HIGH. The alarm is set off with an active- LOW output . What type of gate will provide this logic?

ANSWER:

This question has various parts.

* LOW switch on four doors means there are four inputs with 0 as active state on positive logic.
* When the door is opened, the output gets HIGH which means even one input is changed (in binary logic the only available value apart from 0 is 1) then the output goes HIGH.
* HIGH output triggers the Alarm
* Alarm stops triggering on LOW output.

So, for this case, we will use an OR gate with four inputs at LOW state.

QUESTION#2:

If more than two input AND & OR gates are available, how will you connect its inputs so that they work as two input gates? Perform it for three and four input AND & OR gates.

ANSWER:

If we have AND and OR gates with more than two inputs than we will short extra inputs to have a total of two main inputs at the end.