Central Asian University, Tashkent, Uzbekistan
Department of Computer Science, School of Engineering

Class: CS1/CS2/CS3/CS4/CS5
Subject: Digital Logic Design
Time Allowed: Till Next lab start
Name: Suvonkulov Abdulaziz

Date: October 8th to 11th , 2024
Instructor: Dr. M. Bilal Qureshi
Max Marks: 10
Student ID #220456

Lab Assignment 5- Fall 2024

Implementation of Boolean Functions Using Universal Gate and Interfacing Ultrasonic sensor with Arduino UNO in TinkerCad

Objectives:

- To implement Boolean function using only NAND gates in Multisim and TinkerCad
- To implement Boolean function using only NOR gates in Multisim and TinkerCad
- To implement Boolean function using only NAND gates, using the Multisim logic converter feature
- Interfacing of ultrasonic sensor with Arduino UNO

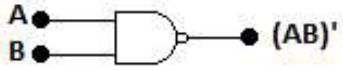
Introduction:

Universal Gates

NAND and NOR are referred to as universal gates, because individually both these gates can implement any Boolean function without the need to use any other type of logic gate. NAND and NOR are both economical and easier to fabricate, compared to the rest. Because of this reason even an AND gate is typically implemented as a NAND gate followed by a NOT gate, and not the other way around. Similarly, an OR gate is typically implemented as a NOR gate followed by a NOT gate, and not the other way around.

NAND Gate:

A NAND gate has more than one input and one output. Logic symbol of a two inputs NAND gate is shown in the figure below.



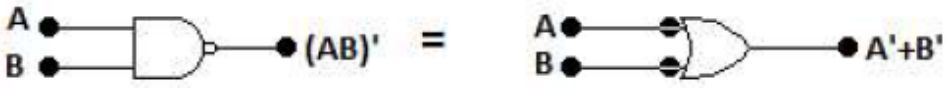
If the two inputs are A and B, the Boolean expression for a NAND gate will be (AB)'. The output of NAND gate is LOW, if and only if all the inputs are HIGH, else LOW. Truth Table for two input NAND gate is given below.

A	B	(AB)'=NAND
0	0	1
0	1	1
1	0	1
1	1	0

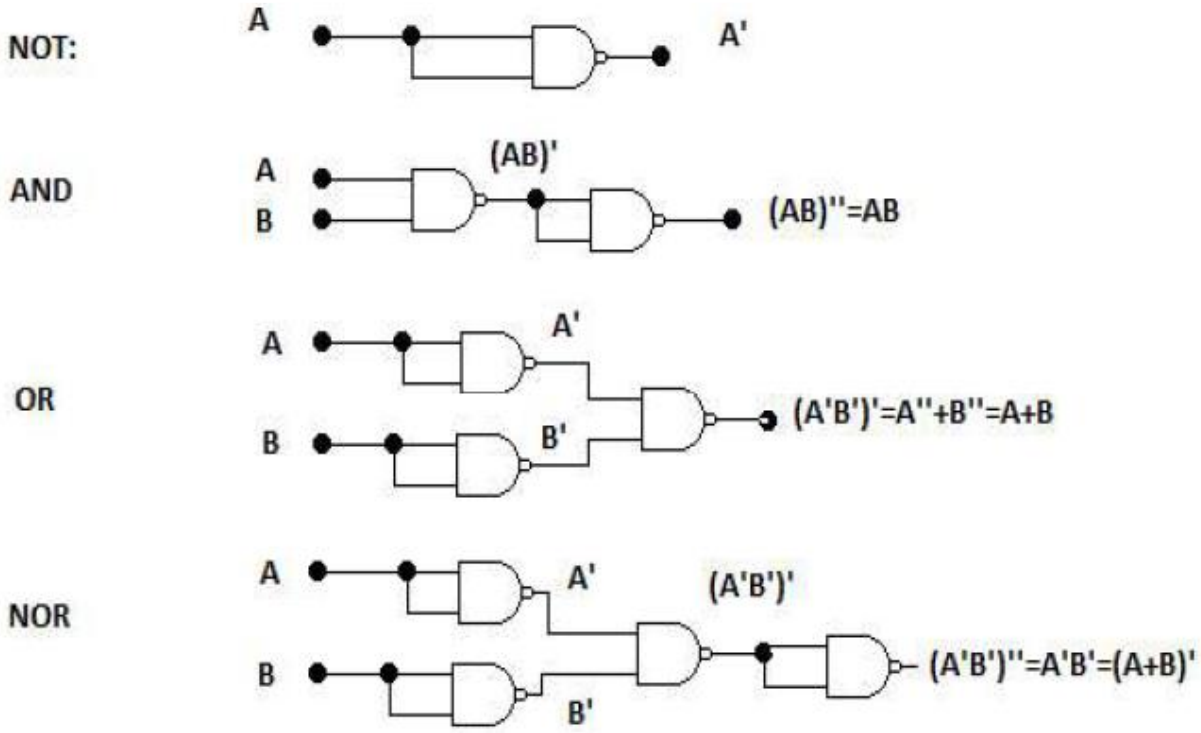
As you already know according to De-Morgan's theorem:

$$(AB)' = A' + B'$$

Hence, both NAND and negative OR gates are equivalent.

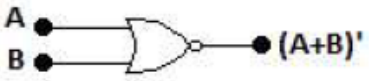


Construction of NOT, AND, and NOR gates from NAND gate(s):



NOR Gate:

A NOR gate has more than one input and one output. Logic symbol of a two inputs NOR gate is shown in the figure below.



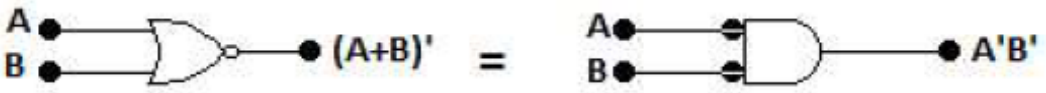
If the two inputs are A and B, the Boolean expression for a NOR gate will be (A+B)'. The output of NOR gate is HIGH, if and only if all the inputs are LOW, else HIGH. Truth Table for two input NOR gate is given below.

A	B	(A+B)' = NOR
0	0	1
0	1	0
1	0	0
1	1	0

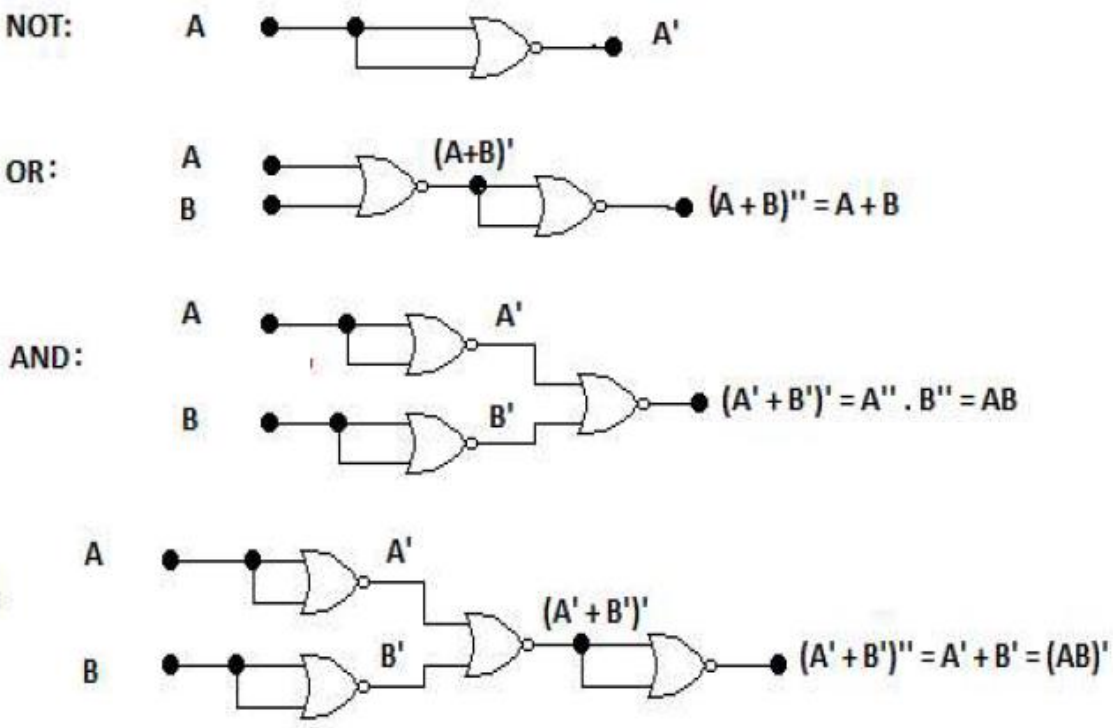
As you already know according to De-Morgan's theorem:

$$(A+B)' = A'B'$$

Hence, both NOR and negative AND gates are equivalent.



Construction of NOT, AND, and NAND gates from NOR gate(s):



Implementation of a Two-Level Boolean Function with only NAND gates:

Follow the steps below, to implement a two level Boolean function with only NAND gates.
1. Simplify the Boolean function and express it in sum of products form.
2. Draw a NAND gate, for each product term.
3. Draw a NAND gate, with inputs coming from the outputs of step-2 NAND gates.

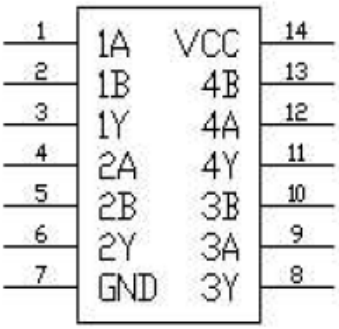
Implementation of a Two-Level Boolean Function with only NOR gates:

Follow the steps below, to implement a two level Boolean function with only NOR gates.
1. Simplify the Boolean function and express it in product of sums form.
2. Draw a NOR gate, for each sum term.
3. Draw a NOR gate, with inputs coming from the outputs of step-2 NOR gates.

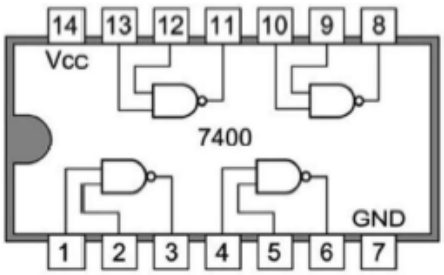
Pin Configuration of NAND and NOR ICs:

7400 (Quad 2-In NAND)

- **1A and 1B:** Inputs of NAND-1.
- **2A and 2B:** Inputs of NAND-2.
- **3A and 3B:** Inputs of NAND-3.
- **4A and 4B:** Inputs of NAND-4.
- **1Y, 2Y, 3Y and 4Y:** Outputs of NAND-1, NAND-2, NAND-3 and NAND-4, respectively.
- **GND and Vcc:** Supply connection lines.

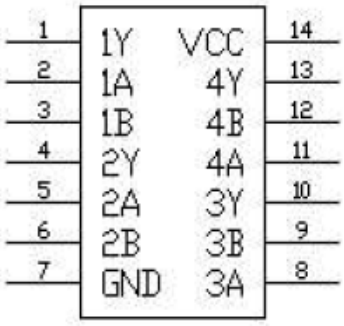


7400

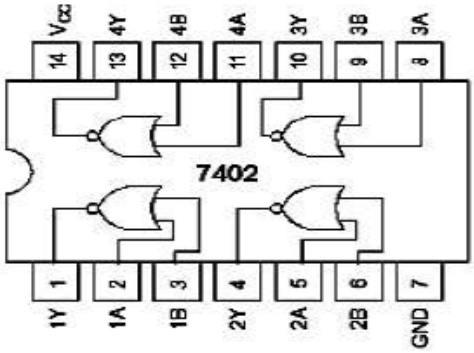


7402 (Quad 2-In NOR)

- **1A and 1B:** Inputs of NOR-1.
- **2A and 2B:** Inputs of NOR-2.
- **3A and 3B:** Inputs of NOR-3.
- **4A and 4B:** Inputs of NOR-4.
- **1Y, 2Y, 3Y and 4Y:** Outputs of NOR-1, NOR-2, NOR-3 and NOR-4, respectively.
- **GND and Vcc:** Supply connection lines.



7402



LAB TASKS:

TASK-I: Construct NOT, AND, OR and NOR with NAND gates

Construct NOT, AND, OR and NOR gates with only NAND gates on logic trainer and fill the tables below.

A	(A.A)' = NOT
0	1
1	0

A	B	(A.B)'	(A.B)'' = AND	A'	B'	(A'.B')' = OR	(A'.B')'' = NOR
0	0	1	0	1	1	0	1
0	1	1	0	1	0	1	0
1	0	1	0	0	1	1	0
1	1	0	1	0	0	1	0

TASK-II: Construct NOT, AND, OR and NAND with NOR gates

Construct NOT, AND, OR and NOR gates with only NOR gates on Multisim and fill the tables below.

A	(A+A)'=NOT
0	
1	

A	B	(A+B)'	(A+B)''= OR	A'	B'	(A'+B')' = AND	(A'+B')'' = NAND
0	0	1	0	1	1	0	1
0	1	0	1	1	0	0	1
1	0	0	1	0	1	0	1
1	1	0	1	0	0	1	0

TASK-III: Implement a Two Level Boolean Function with NAND gates

Implement the following Boolean function with only NAND gates on Multisim and draw the circuit diagram.

F = A.B.C + A.C + D

Circuit Diagram:

TASK-IV: Implement a Two Level Boolean Function with NOR gates

Implement the following Boolean function with only NOR gates on logic trainer and draw the circuit diagram.

$$F = A.B.C + A.C + D$$

Circuit Diagram:

Note: Since you don't have logic trainer perform them on TinkerCad using IC.

TASK-V: Implement a Boolean Function with NAND gates, using Logic Converter

Implement the function below with NAND gates using the Multisim's logic converter and draw the circuit diagram.

$$F = A.(B'.C + C') + B.C' + A$$

Circuit Diagram:

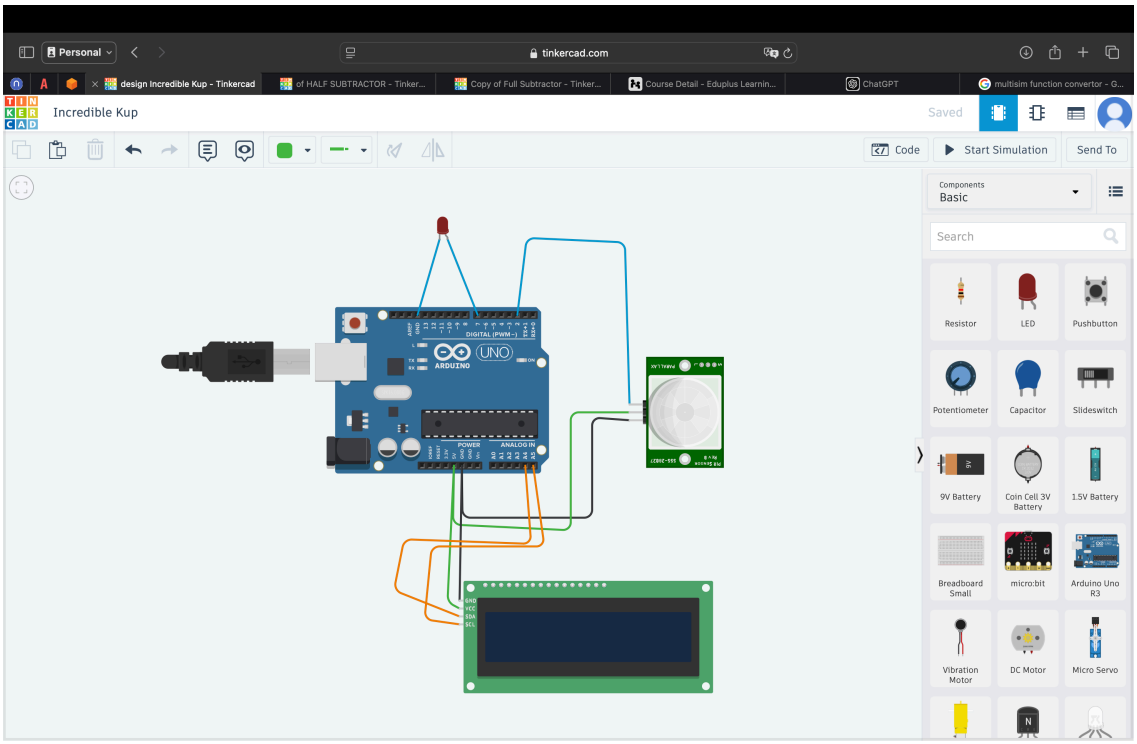
Write the Boolean expression in the logic trainer and press button.

TASK -VI:

Interfacing Ultrasonic sensor, LED and Buzzer with Arduino in TinkerCad, write an Arduino sketch that turns ON the LED when distance measurement from ultrasonic gets below 100 cm else turn it OFF, turn ON the buzzer When distance measurement from ultrasonic sensor gets below 50 cm else turn it OFF.
Use builtin function of input from ultrasonic sensor or look into its datasheet and find a way to convert sound waves into appropriate distance measurement.

TinkerCad Circuit Diagram Screenshot:

Arduino Code with comments:




```
// C++ code
//
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
const int led=7;
const int PIR=2;
LiquidCrystal_I2C lcd(32, 16, 2);
void setup()
{
  pinMode(PIR, INPUT);
  pinMode(led, OUTPUT);
  Serial.begin(9600);

  lcd.init();
  lcd.backlight();
  lcd.setCursor(3, 0);
  lcd.print("privet");
}

void loop()
{
  int value = digitalRead(PIR);
  if(value == 1){
    digitalWrite(led,HIGH);
  }else{
    digitalWrite(led,LOW);
  }

  delay(10); // Delay a little bit to improve simulation
```