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

Class:
Subject: Digital Logic Design
Time Allowed: Till Next lab start
Name: Suvonkulov Abdulaziz

Date: October 9th , 2024
Instructor: Dr. M. Bilal Qureshi
Max Marks: 10
Student ID # 220456

Lab Assignment 4- Fall 2024

Understanding the Conversion between Boolean Expressions and Logic circuits and (understanding how analog input is interpreted by Arduino or other controllers)

Objectives:

- To convert logic circuits to Boolean expression
- To convert Boolean expression to logic circuits
- To determine truth tables for logic circuits using Multisim

Introduction:

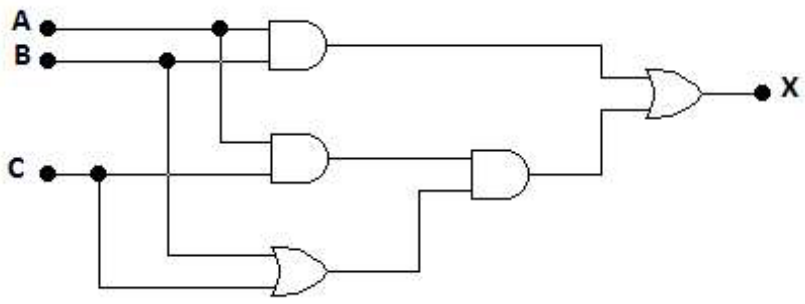
Logic circuits are constructed from a combination of different logic gates. Boolean algebra is the mathematics of digital circuits. The operation of logic circuits can be expressed in a concise manner through Boolean algebra. Logic can be represented in three ways.

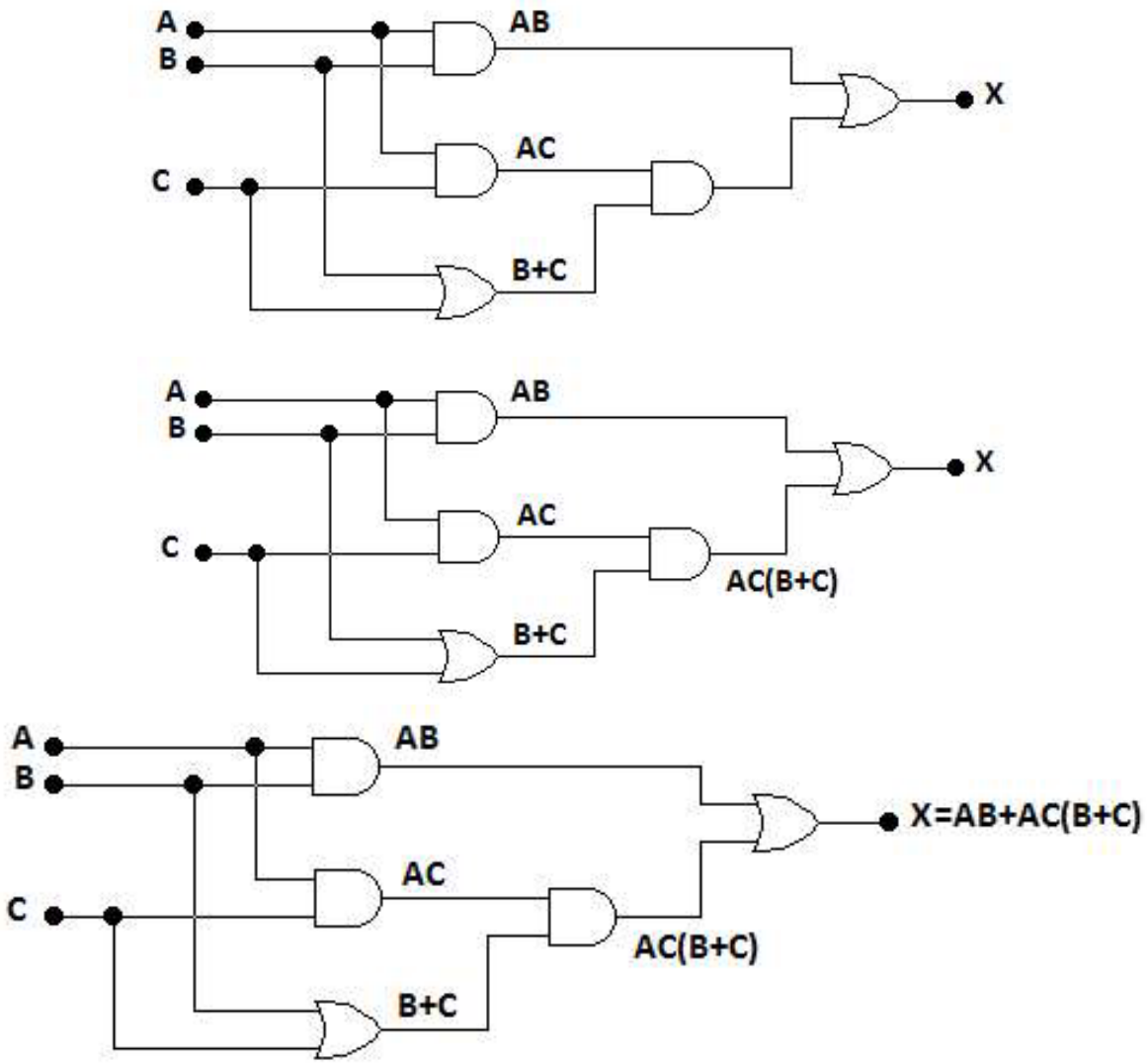
1. Boolean Expressions
2. Boolean Circuits
3. Truth Tables

It is therefore important to know the inter-conversion between them.

Converting Logic Circuits to Boolean Expressions:

To determine Boolean expression of any logic circuit, start at the left most inputs and work towards the final output, writing logic expressions at the output of each gate, as shown below.





Converting Boolean Expressions to Logic Circuits:

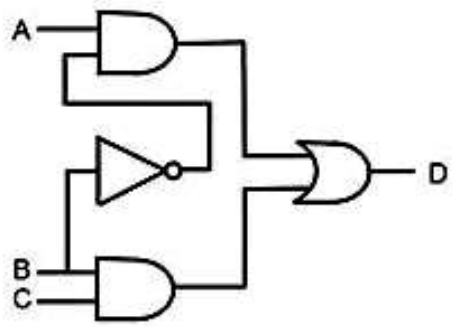
To convert a Boolean expression to logic circuit, requires the knowledge of order of operation. The order of operation is as follow.

1. Bracketed quantities
2. NOTs
3. ANDs
4. ORs

LAB TASKS:

TASK-I Using AND, OR and NOT ICs

Use AND, OR and NOT ICs from the previous lab to implement the circuit below, using TinkerCad. Record your observations in the table below and write the Boolean expression for the circuit.

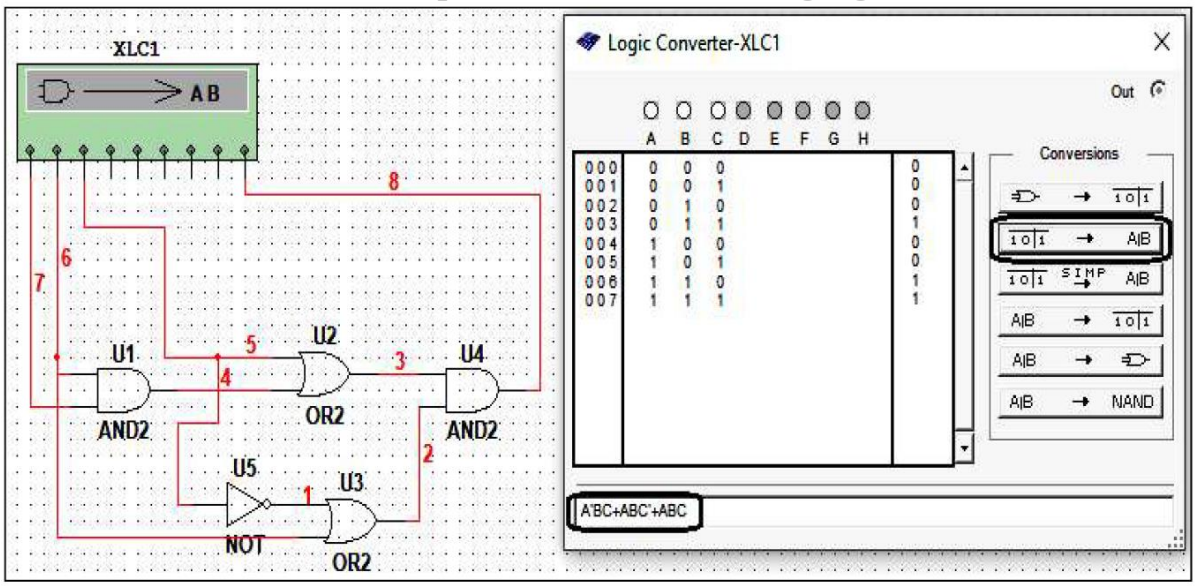


A	B	C	D
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Boolean Expression D=(A*B)+(A(bar)*C)

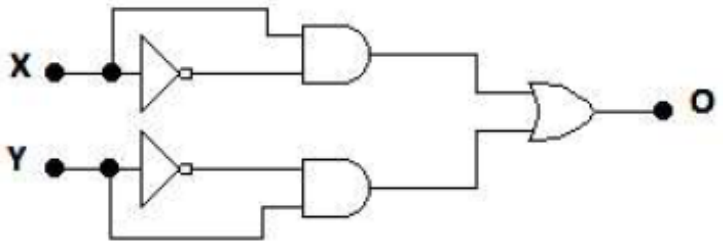
TASK-II Finding Boolean Expression of Logic circuits:

Draw the circuits below on Multisim, run them to fill in their respective truth tables and find the Boolean expressions of circuits using logic converter.



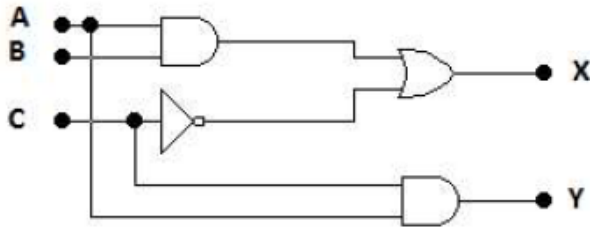
Logic Circuit – I

X	Y	O
0	0	1
0	1	0
1	0	0
1	1	1



Boolean Expression: O=(X(bar)*Y(bar)+(X+Y)

Logic Circuit – II



A	B	C	X	Y
0	0	0	1	0
0	0	1	0	0
0	1	0	1	0
0	1	1	0	0
1	0	0	1	0
1	0	1	0	1
1	1	0	1	0
1	1	1	1	1

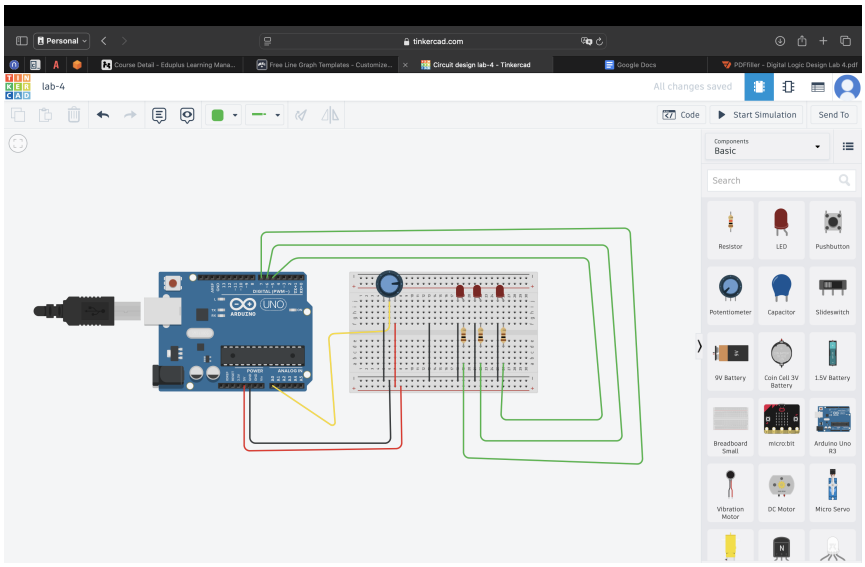
Boolean Expression X: X=(A*B)+C(bar)

Boolean Expression Y: Y=A*C

TASK-III Using Arduino Uno, LEDs, Potentiometer

Write an Arduino UNO sketch that controls the direction of blinking for three LEDs using a potentiometer (variable resistor). When the potentiometer's voltage is greater than 2.5 V (knob rotated to the right), the LEDs should blink from left to right. When the voltage is less than 2.5 V (knob rotated to the left), the LEDs should blink from right to left. Provide a circuit diagram, demonstrate the LEDs blinking in both directions, and properly format the code. Perform the task in Tinker Cad then using the hardware.

TinkerCad Circuit Picture: (simulation)



Real Circuit Picture: (hardware)

Arduino Code with proper comments:

```
void setup()
{
  pinMode(A0, INPUT);
  pinMode(7, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(5, OUTPUT);
}

void loop()
{
  if (analogRead(A0) < 512) {
    digitalWrite(7, HIGH);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(7, LOW);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(6, HIGH);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(6, LOW);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(5, HIGH);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(5, LOW);
  } else {
    digitalWrite(5, HIGH);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(5, LOW);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(6, HIGH);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(6, LOW);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(7, HIGH);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(7, LOW);
  }
}
```