NAME : Malla Sailesh
ROLL NUMBER :2021101106
GROUP NUMBER:6

LAB 2
DIGITAL STRUCTURES AND MICROCONTROLLERS

AIM:

1) **Part A: Checking the output voltages on multimeter by varying the input voltage**
2) **Part B : Verifying the truth table of logic circuits**
3) **Part C: Verifying the Demorgan's law**
4) **Part D: Binary Full Adder Circuit**

Electronic Components:

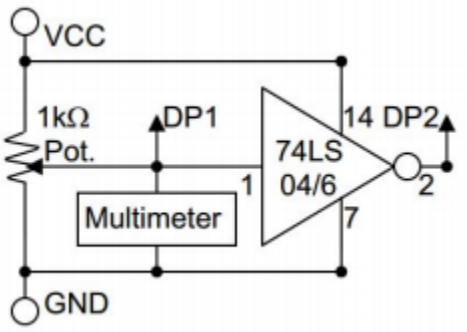
1) **Arduino**
2) **Led**
3) **Resistor**
4) **Breadboard**
5) **Integrated circuits (Not gate, And gate , Or gate ,Nand gate ,Nor gate ,Xor gate)**

Links to Reference Circuit :Copy the links and paste it in chrome to view it

1) **Part A : https://crcit.net/c/8908b67e570f41e093980d2144e34569**
2) **Part B(Not gate) :https://crcit.net/c/5107910fe4244d9192a0be8c97eadc97**
3) **Part B(Or gate):https://crcit.net/c/ee481ab0d25b441ba423099284b61594**
4) **Part C:https://crcit.net/c/b3cd57d6955b4138b068f77955aa1f10**
5) **Part D: https://crcit.net/c/f74c86678cb34adaaaf3cef7b7b8d307**

Procedure:

1) **Part A:**



1. Set up the circuit shown in Figure on the breadboard and turn the potentiometer shaft to one end so that the multimeter reads 0V.
2. DP1 and DP2 are LEDs connected with appropriate resistors. DP2 must be glowing.
3. Now rotate the potentiometer shaft gradually up to the other end and tabulate the transitions in DP1 and DP2.

We are intending to find the tipping point voltage for the IntegratedCircuit(IC) which it considers to be HIGH or LOW in both INPUT and OUTPUT pins. You may add another multimeter at the output of the NOT gate to monitor the output voltage as well. Compare these voltages with the specifications for binary logic level for a 0-5 V range.
0 <= Vol <= 0.4, 0 <=Vil <= 0.8, 2.0<= Vih <= 5.0, 2.4 <= Voh <= 5.0
Where Vil is Voltage input low , Vol is Voltage output low , Vih is Voltage input high and similarly Voh

2) **Part B :**

The goal of this part of the experiment is to take input from the serial monitor and verify the truth table of logic gates: (NOT, OR, AND, XOR, NOT, NAND)
1. Place the IntegratedCircuit on breadboard and give Vcc(Power) and Gnd(Ground) connection to it.
2. Take inputs from the Serial Monitor for values of x and y if two inputs and x if one input and route them to the input pins of the IC.
3. Connect an LED with appropriate resistor to the output of the GATE.
4. Note the output of the chosen gate for different values of input in a truth table.
The code : 1) If we consider Not gate

```
int x;
void setup()
{
  pinMode(13, OUTPUT);
  Serial.begin(9600);//sets the data rate to 9600 bps
}

void loop()
{
  Serial.print("\nx=");
  while(Serial.available()==0){}// so that if input is blank do nothing
   x=  Serial.read();
  x=x-'0';//takes ascii value if 0 it takes 48 so 48 - 48 =0
  Serial.println(x);

  digitalWrite(13,x);//so that values considered are to be digital

  Serial.print("Enter the value again");
```

```
  while(Serial.available()==0){}

}
```

2) if we consider Nand gate

```
int x,y;
void setup()
{
  pinMode(13,OUTPUT);
  pinMode(12,OUTPUT);
  Serial.begin(9600);//sets the data rate to 9600 bps;
}
void loop()
{
  Serial.print("\nx=");
  while(Serial.available()==0){}
  x=Serial.read();
  x=x-'0';
  Serial.print(x);

  Serial.print("\ny=");
  while(Serial.available()==0){};
  y=Serial.read();
  y=y-'0';
  Serial.println(y);

  digitalWrite(13,x);
  digitalWrite(12,y);

  Serial.print("Enter the values again");
  while(Serial.available()==0){};

}
```

3) **Part C :**

De Morgan's theorems state that (A + B)' = A' · B' and (A · B)' = A' + B'. Verify these theorems by proceeding step by step as follows:

1. Set up a circuit consisting of two NOT gates and one AND gate to perform function Y = A' ·B',

2. Obtain the truth table of this circuit by noting the output of the function for different values of A and B. Verify that the output of the function is same as that of the NOR gate.
3. Repeat steps 1 and 2 using an OR gate instead of an AND gate to verify that the truth table is same as that of the NAND gate.

Truth table of nor gate is
(X = 1& y = 1)= 0;
(X = 1& y = 0)= 0;
(X = 0& y = 0)= 1;
(X = 0& y = 1)= 0;

Code is :

```
int x,y;
void setup()
{
  pinMode(13,OUTPUT);
  pinMode(12,OUTPUT);
  Serial.begin(9600);//sets the data rate to 9600 bps;
}
void loop()
{
  Serial.print("\nx=");
  while(Serial.available()==0){}
  x=Serial.read();
  x=x-'0';
  Serial.print(x);

  Serial.print("\ny=");
  while(Serial.available()==0){};
  y=Serial.read();
  y=y-'0';
  Serial.println(y);

  digitalWrite(13,x);
  digitalWrite(12,y);

  Serial.print("Enter the values again");

}
```

4) **Part D :**

A binary Full Adder adds two bits A and B along with a carry in C to generate SUM and CARRY bits as output. The first step to achieve this is to make a binary Half Adder, which adds two binary inputs A and B to give a sum S1 and a carry C1 according to the following Boolean expressions for the outputs S1 and C1:

S1 = A' · B + A · B' and C1 = A · B
Another Half Adder is then used to generate the final SUM by adding the third binary input C to the S1 bit generated by the first Half Adder:
SUM = S1'.C+S1.C'
The carry bit generated by this Half Adder is given by
C2 = S1 · C
FInal Carry = C1 + C2;
Write down the complete truth table of a Full Adder, including columns for the intermediate outputs S1, C1 and C2. Find out the logic for generating the final CARRY output from C1 and C2. As XOR and AND gates are going to be used for the Half Adders, try to obtain a logic for CARRY using the same type of gates, so that the complete realisation of the Full Adder is possible without necessitating a third IC.
1. Set up the circuit of a Half Adder using an XOR gate and an AND gate. Apply the inputs A and B from two input pins and observe the outputs S1 and C1 on two LED displays for all combinations of the inputs. Tabulate these values and verify the operation of the Half Adder.
2. Set up another Half Adder using another XOR and another AND gate out of the same ICs used in step 1, and connect the C input and the S1 output generated by the first Half Adder as its inputs to generate the final SUM output and the C2 output.
3. Generate the final CARRY output from the intermediate carry outputs C1 and C2.
4. Verify the truth table experimentally by applying the inputs A, B and C through three input pins and displaying the S1, C1, C2, SUM and CARRY outputs.

Code :

```
 int x,y,z;
void setup()
{
  pinMode(13,OUTPUT);
  pinMode(12,OUTPUT);
  pinMode(11,OUTPUT);
  pinMode(10,OUTPUT);
  pinMode(9,OUTPUT);
  pinMode(8,OUTPUT);
  Serial.begin(9600);//sets the data rate to 9600 bps
}
```

```
void loop()
{
  Serial.print("\nx=");
  while(Serial.available()==0){};
  x=Serial.read();
  x=x-'0';
  Serial.print(x);

  Serial.print("\ny=");
  while(Serial.available()==0){};
  y=Serial.read();
  y=y-'0';
  Serial.print(y);

  Serial.print("\nz=");
  while(Serial.available()==0){};
  z=Serial.read();
  z=z-'0';
  Serial.println(z);

  digitalWrite(13,x);
  digitalWrite(12,y);
  digitalWrite(11,x);
  digitalWrite(10,y);
  digitalWrite(9,z);
  digitalWrite(8,z);

  Serial.print("Enter the values again");
}
```

Conclusion:

1) **Part A:** It gives 0 volt output and the output LED doesn't glow for input of 5volt and it gives 4.87 volt output for 0 volt input and now the output LED glows.
2) **Part B :** For NOT gate

X	Output
1	0
0	1

For AND gate

X	Y	Output
1	1	1

1	0	0
0	1	0
0	0	0

For OR Gate

X	Y	Output
1	1	1
1	0	1
0	1	1
0	0	0

For NAND Gate

X	Y	Output
1	1	0
1	0	1
0	1	1
0	0	1

For NOR gate

X	Y	Output
1	1	0
1	0	0
0	1	0
0	0	1

For XOR gate

X	Y	Output
1	1	0
1	0	1
0	0	0
0	1	1

3) **Part C :**

A	B	A'.B'
1	1	0
1	0	0
0	1	0
0	0	1

4) **Part D:**

A	B	C	S1	FINAL SUM	C1	C2	FINAL CARRY
1	1	1	0	1	1	0	1
1	1	0	0	0	1	0	1
1	0	1	1	0	0	1	1
1	0	0	1	1	0	0	0
0	1	1	1	0	0	1	1
0	1	0	1	1	0	0	0
0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0

Link of Tinkercad Simulation :Copy the links and paste it in chrome to view it

1) **PartA :**
   https://www.tinkercad.com/things/2nt6WaZE3eN-glorious-bruticus-jofo/editel?sharecode=S7EOZp12DcBkV9lEC9lWzGfwosLGtZsCFi_AtLrt4_s
2) **Part B:**
   For Not :
   https://www.tinkercad.com/things/0xSSvjtZIlg-not-gate-partb/editel?sharecode=79vWZKVv5cl84ZudleA4OLex1Ehwv-R_NE5RGhOLKns

   For And :
   https://www.tinkercad.com/things/cEHom9tG6o4-and-gate-partb/editel?sharecode=dwB40A25T7538HKvUJJef7vVv7-Mrly_bYSFlSMblOc

   For Or :
   https://www.tinkercad.com/things/6gC65Uax16n-or-gate-part-b/editel?sharecode=NrPgCUlfX3zDAz49GcJShGlMuDS8nnlKZnnyK5bds_Q

   For Nand :
   https://www.tinkercad.com/things/5llT2ilSci9-nand-gate-partb/editel?sharecode=xJ0JwWOg5xtwg-3UuEVlMsPR7yY6wKprg8wpcSt_yuE

For Nor :
https://www.tinkercad.com/things/99v8HgjaA14-nor-gate-part-b/editel?sharecode=oSJk0Xix8WYrTa0nnHEcahJcQixbvpExP2-iP7xhTrY

For Xor:
https://www.tinkercad.com/things/86Aom2aDjqg-xor-gate-partb/editel?sharecode=TJShfxHknh3lfnseFaZECVCpqZHrB_thlk4OemfcnPM

3) **Part C:**
https://www.tinkercad.com/things/jlo0eRWt0UV-part-c-demorgans-law/editel?sharecode=wyUUWDczv65imL3Ph8Hta89bvVqcy1yAtnkR5DlhHtQ
4) **PartD:**
https://www.tinkercad.com/things/67wwoL3QO6U-full-adder-partd/editel?sharecode=Xl0AWF02oovEOVi9_In8kEseRSYydnDDZW3jlNdgBzs