DSM LAB REPORT – 2

MACHARLA HARISH (2020102062)

Objective: To familiarize with the logic gates, De Morgan's law and to implement a full adder circuit using gates.

Description:

In this experiment, you will be introduced to some of the basic logic gates (NOT, AND, OR, NAND, NOR, XOR etc) available commercially in the IC (Integrated Circuit) form.

Two families of digital ICs are commonly used: the **TTL 74LSxx series** and the **CMOS CD 40xx** series. Many of these ICs have 14 pins, and some have 16 or more. Two pins are used for power supply connections.

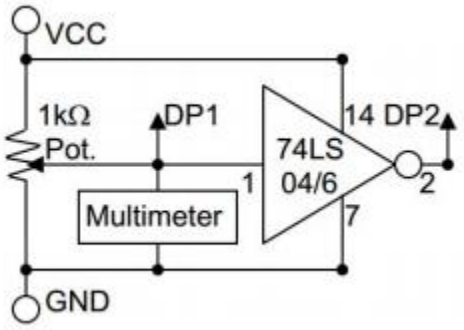
Thus 12 pins are available in a 14-pin IC for gate inputs and outputs. A 2-input gate requires three pins per gate (two for inputs and one output), and so ICs that implement 2-input logic functions generally have 4 gates per IC.

TTL ICs require a fixed d-c power supply voltage VCC having the nominal value of 5V and a tolerance of 5%. Thus, these ICs are not guaranteed to function with VCC below 4.75V and VCC in excess of 5.25V can damage the IC.

A 0.1µF ceramic capacitor should be connected between the VCC and Ground pins of each TTL IC to suppress spikes that may otherwise be created due to the current drawn from the power supply.

Most CMOS ICs can work with $3V \leq VCC \leq 15V$ and do not require capacitors at each VCC pin.

Part A: Logic Levels



Procedure:

1. Set up the circuit shown in Fig 2 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 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.

NOT GATE:

Link:

https://www.tinkercad.com/things/dphviiGvp3r-amazing-fulffy/editel?sharecode=8jSGoZ-1guSq5FAaHJUSdDGozG1maUl9QYHDgBKJ68w

AND GATE:

Link:

https://www.tinkercad.com/things/g9trFN0blbY-stunning-stantia-maimu/editel?sharecode=Nd64PLqabnR4g-pBX8Mec2YIuLUbSx9bhuZoPcRYrUE

Conclusion:

We can verify the logic gates by changing the readings of potentiometer.

Part B: Verifying the truth table of Logic Gates

Objective: The goal of this part of the experiment is to take input from the serial monitor and verify the truth table of logic gates: (NOR, OR, AND, XOR, NOT, NAND) using TTL 74XX family of ICs.

Procedure:

1. Place the IC on breadboard and give V cc and Gnd connection to it.

2. Take inputs from the Serial Monitor for values of A and B 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.

Code for Serial Monitor:

```
int x,y;
void setup() {
Serial.begin(9600);
  pinMode(5, OUTPUT);
  pinMode(4, OUTPUT);
  //opens serial port, sets data rate to 9600 bps
}
void loop() {
if(Serial.available() > 0) {
x=Serial.read();
y=Serial.read();
x = x - '0';
        y = y - '0';
  }
digitalWrite(5, x);
        digitalWrite(4, y);
Serial.println(x);
Serial.println(y);
}
```

AND GATE:

Truth Table:

A	B	output
0	0	0
0	1	0
1	0	0
1	1	1

Link:

https://www.tinkercad.com/things/bvVy2tyVRax-stunning-juttuli-tumelo/editel?sharecode=3bm5HqveBpHzvz5gtNvxef9GehKz_a5qXqQhcCSmuEw

OR GATE:

Truth Table:

A	B	output
0	0	0
0	1	1
1	0	1
1	1	1

Link:

https://www.tinkercad.com/things/4KJcSGrc9DQ-tremendous-vihelmo-waasa/editel?sharecode=2tI6cllPjfCpnv4dWIKX0M0FN9rngC0xbAbT2ait3nk

XOR GATE:

Truth Table:

A	B	output
0	0	0
0	1	1
1	0	1
1	1	0

Link:

https://www.tinkercad.com/things/1kRZGjn1Ek5-powerful-sango-allis/editel?sharecode=kYrn1i8YlMNNTn8zGoYJ150TFnHr9MCySOiXOFm4C4k

NOR GATE:

Truth Table:

A	B	output
0	0	1
0	1	0
1	0	0
1	1	0

Link:

https://www.tinkercad.com/things/9A5qBs7QeX6-spectacular-maimu-elzing/editel?sharecode=3YKRw9Zv6f8yv46WqELE0l5Cx7Qe73b7nb_b_1uv5K0

NAND GATE:

Truth Table:

A	B	output
0	0	1
0	1	1
1	0	1
1	1	0

Link:

https://www.tinkercad.com/things/hXZWyZ4SQ8b-stunning-borwo-rottis/editel?sharecode=k0fYXktW4_Db9akjRpgRzl10Ma9eNuXrr4jWaEBJrJQ

NOT GATE:

Truth Table:

Input	output
0	1
1	0

Link:

https://www.tinkercad.com/things/juRFwge9iQE-smashing-gaaris/editel?sharecode=dQxZIl44eRUyOejftbQfmvFBlKVrXaZnhRt-f5Ex6BM

Conclusion:

I verified all the truth tables of six logic gates by performing experiment using tinkercad software.

Part C: De Morgan's Law

De Morgan's theorems state that (A + B) '= A' · B' and (A · B)' = A' + B'

Objective: To verify De Morgan's Laws

Procedure:

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

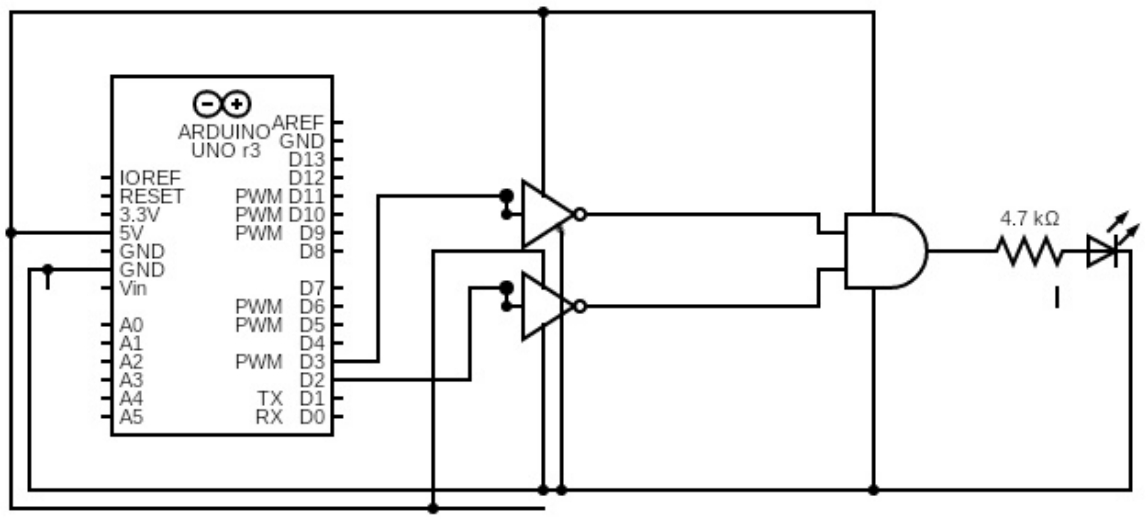
2. Obtain the truth table of this circuit by noting the output of the function for different values of A and B. Verify that the output of the function is same as that of the NOR gate.

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 for First Law: **(A + B) ' = A' · B'**

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

Reference Circuit:



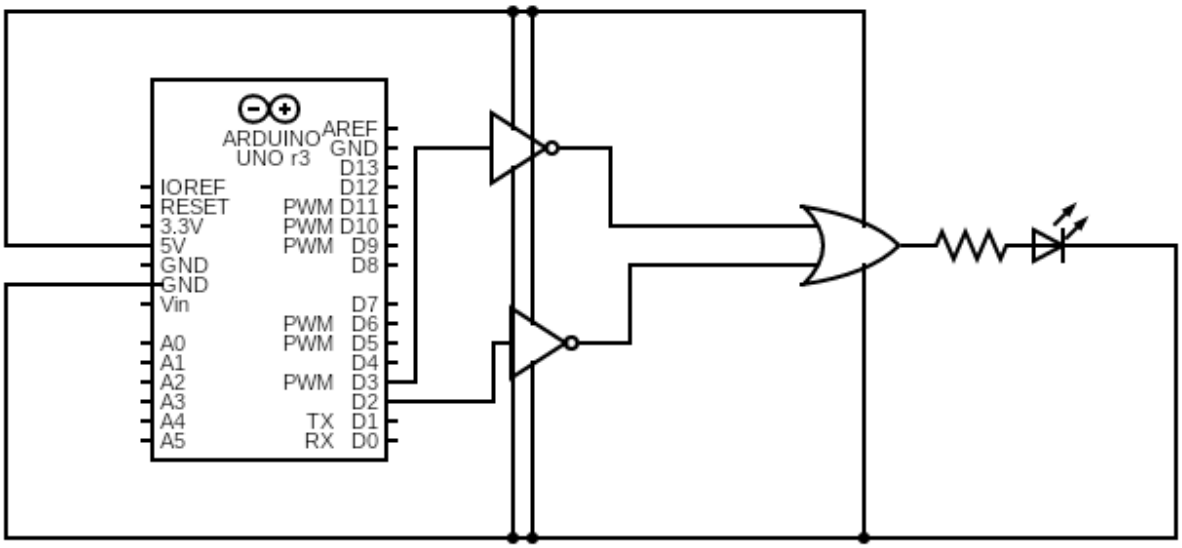
Link:

https://www.tinkercad.com/things/9rc6b4yCTRX-bodacious-fyyran/editel?sharecode=jRzsGyDEudSEESkgQ5YPK9gyQoL5Qw7jG8gHplEytf4

Truth Table for Second Law: **(A · B)' = A' + B'**

A	B	A'	B'	A.B	(A.B)' NAND	A'+B'
0	**0**	**1**	**1**	**0**	**1**	**1**
0	**1**	**1**	**0**	**0**	**1**	**1**
1	**0**	**0**	**1**	**0**	**1**	**1**
1	**1**	**0**	**0**	**1**	**0**	**0**

Reference Circuit:



Link:

https://www.tinkercad.com/things/cNBtGDJwEXc-brave-leelo/editel?sharecode=g8F1UoMjdQjPzAMBZt4JRF4yTJKcjxAX2iwCqDUgwj8

How would you realize the above circuit if you have only NAND gates instead of NOT gates? I.e. How would you use NAND gates to perform function of NOT gates?

By De Morgan's theorem, a two-input **NAND gate's** logic may be expressed as (A.B)'=A'+B', **making** a **NAND gate** equivalent to inverters followed by an OR **gate**. The **NAND gate** is significant because any Boolean function can be implemented by **using** a combination of **NAND gates.**

Conclusion:

Both the Truth Tables of Demorgan's Law are verified. They satisfied the NOR and NAND gates Truth tables.

Part D: Binary Full Adder

Objective: To verify binary full adder

Description:

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:

$$S1 = A' \cdot B + A \cdot B' = A \oplus B \text{ and } C1 = A \cdot B$$

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:

$$SUM = S1 \oplus C$$

$$C2 = S1 \cdot C$$

Procedure:

1. Set up the circuit of a Half Adder using an XOR gate and an AND gate. Apply the inputs 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.

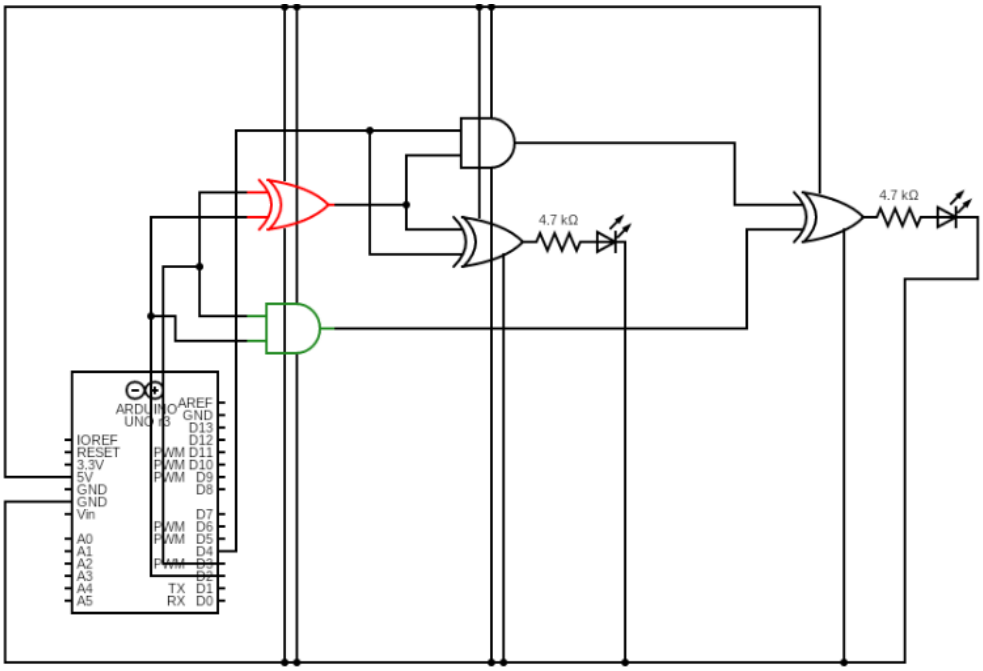
3. Generate the final CARRY output from the intermediate carry outputs C1 and C2, using the unused gates in the XOR and AND ICs deployed so far.

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.

Truth Table for Binary Full adder:

A	B	C	S1 $A \oplus B$	C1 A.B	C2 S1.C	SUM $S1 \oplus C$	CARRY $C1 \oplus C2$
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	1	0	0	1	0
0	1	1	1	0	1	0	1
1	0	0	1	0	0	1	0
1	0	1	1	0	1	0	1
1	1	0	0	1	0	0	1
1	1	1	0	1	0	1	1

Reference Circuit:



Orange – Carry

Blue – Sum

Pink – C1

Brown – C2

Yellow – S1

Link:

https://www.tinkercad.com/things/gnzROqlLxHY-bodacious-vihelmo/editel?sharecode=F8001OC2ap1A3nxxNroFNHLydL4TNVEdUcSu2LKmn6o

Conclusion:

All the truth tables(S1,C1,C2,SUM,CARRY) are verified.