LAB REPORT : 5

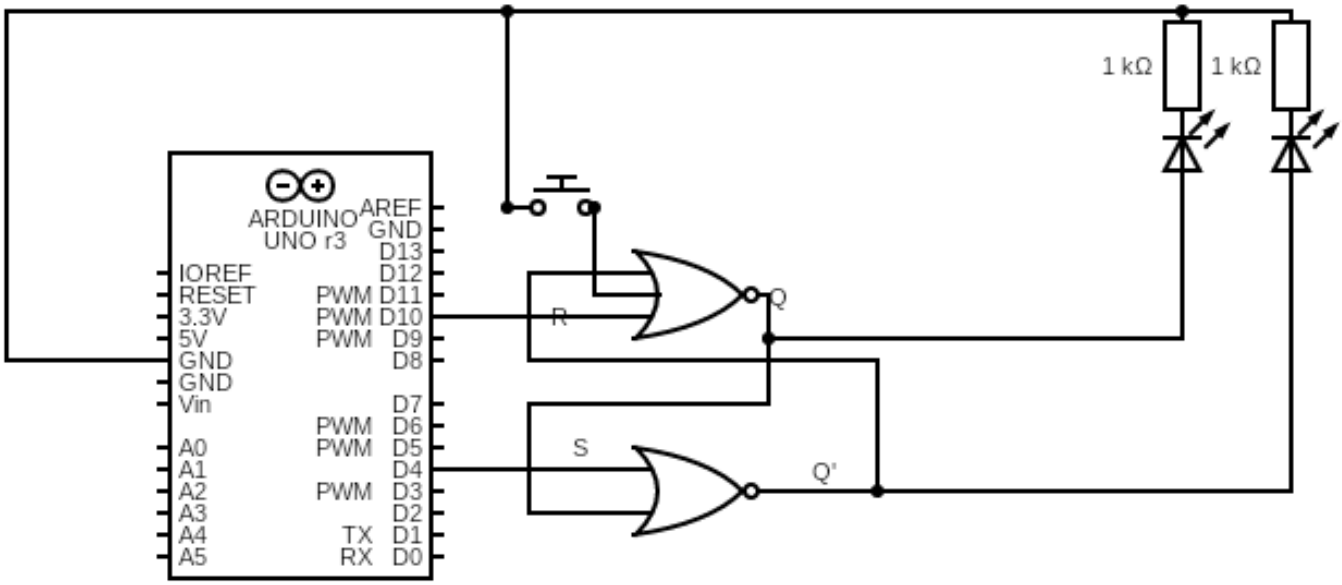
NAME: Arya Marda
ROLL NUMBER : 2021102021
GROUP NUMBER : 3

PART A: RS LATCH

AIM: To understand and build an RS latch. We are building a NOR latch using NOR GATE.

Electronic components used: Wires , LED, Resistors, Arduino uno, breadboard, XOR GATE IC'S, Push button.

Reference circuit:



Procedure: 1. Take inputs from the user of R and S.

2.Take two NOR gates and input R ans S in 2 different NOR gates, now draw the output of 1 NOR gate as an input for the other NOR gate (Do this for both thr gates) .

3.Add an Push-button to one of the gates as input. This is used to start the circuit or elese it will run very slowly.

4.Connect the outputs of the NOR gates to LEDS.

Code:

```
int pin1 = 4;
int pin2 = 3;

int x, y, k;

void setup()
{
pinMode(pin1, OUTPUT);
pinMode(pin2, OUTPUT);
Serial.begin(9600);
}

void loop()
{


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


Serial.print("R=");
```

```
while(Serial.available() == 0){}
x=Serial.read();
x=x-'0';
Serial.println(x);


digitalWrite(pin1,y);
digitalWrite(pin2,x);



}
```

Conclusion:

Q. We must Observe and tabulate the sequence of Q and Q’ in response to the following input sequence: S R = 01, 00, 10, 00, 01, 10, 01, 00, 11, 00, 10,11, 00, 01, 11, 00.

S	R	Q	Q’
0	1	0	1
0	0	0	1
1	0	1	0
0	0	1	0
0	1	0	1
1	0	1	0
0	1	0	1
0	0	0	1
1	1	0	0
0	0	Unpredictable	Unpredictable
1	0	1	0
1	1	0	0
0	0	Unpredictable	Unpredictable
0	1	0	1

1	1	0	0
0	0	Unpredictable	Unpredictable

Q. When given the above inputs, explain till when the latch can be expected to operate correctly and why.

ANS: The latch will work correctly till 9th input, i.e (1, 1) . On the 9th input thought the outpupt is as expected(0,0) but after this state, (10th input , i.e (0,0)) the device will enter in unpredictable state called meta state.
Therefore : input (1,1) is a wrong input which causes indeterminant state.

If both inputs are then switched to (0,0) , output will be (0,0) for time-being and then when output acts as input for latches the value of Q,Q' will toggle till it reaches a stable state which is unpredictable.

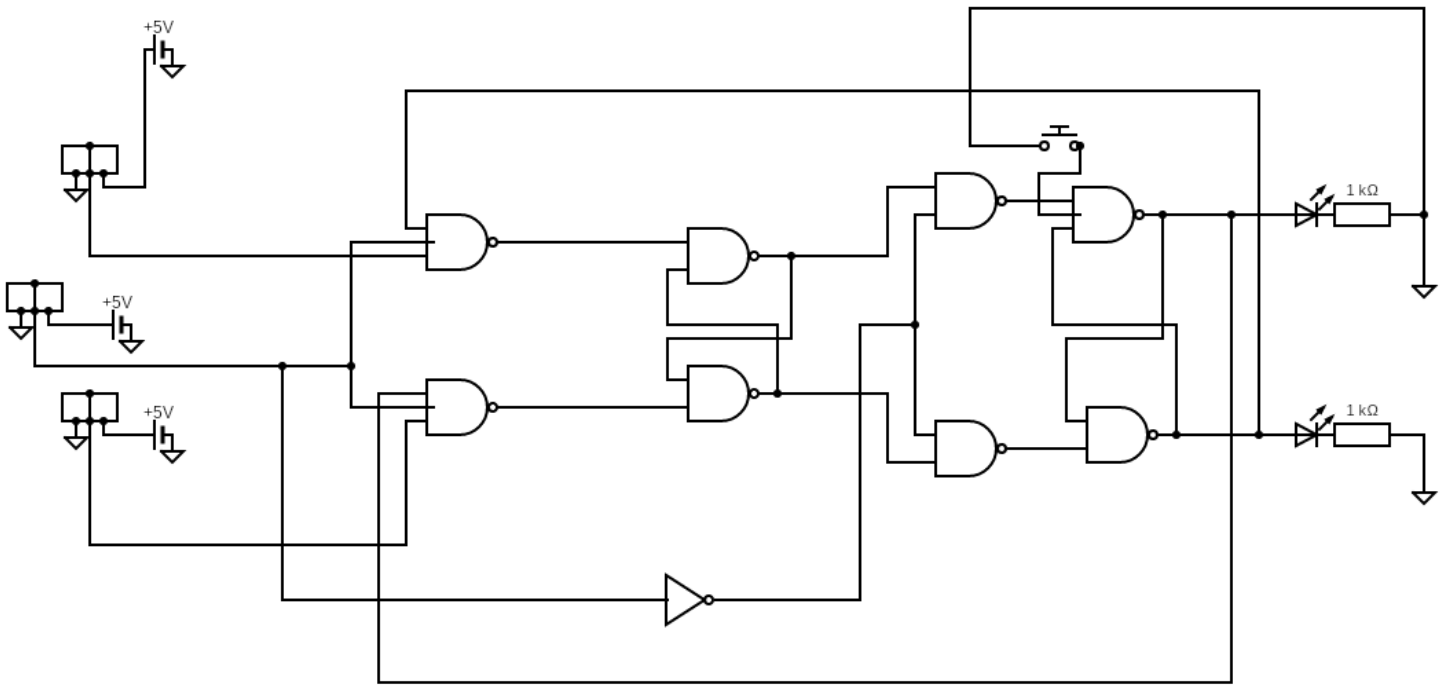
TinkerCAD Link : https://www.tinkercad.com/things/37xTxm3CCsX

PART B: JK Master-Slave Flip-Flop

AIM: To understand and build a JK Master-Slave Flip-Flop.

Electronic components used: Wires , LED, Resistors, slide-switch, 2-input NAND GATES IC'S, 3-input NAND gate IC'S, NOT gate, Push button.

Reference circuit:



Procedure:

1. The JK master-slave flip flop consists of two latches: a master latch and a slave latch. The master latch changes its values on the leading edge of the clock, whereas the slave latch changes its values on the trailing edge of the clock.

2.Form the cicruit as in the Reference circuit.

3.Slide the slide switches to see the output.

4.Output should be of form:

J	**K**	**ACTION**	Q_{n+1}
0	0	HOLD	Q_n
0	1	CLEAR	0
1	0	SET	1
1	1	TOGGLE	Q'_n

Code: No code required.

Conclusion :

Q. Observe and tabulate the sequence of Q and Q' in response to the following input se-
quence: J K = 10, 00, 01, 10, 01, 00, 11, 00, 10, 11, 00, 01, 11, 00.

				CURRENT OUTPUT	
J	K	Qn(presentstate of Q)	Qn'(presentstate of Q')	Qn+1	Qn+1'
1	0	X (Don't care)	X (Don't care)	1	0
0	0	1	0	1	0
0	1	1	0	0	1
1	0	0	1	1	0

0	1	1	0	0	1
0	0	0	1	0	1
1	1	0	1	1	0
0	0	1	0	1	0
1	0	1	0	1	0
1	1	1	0	0	1
0	0	0	1	0	1
0	1	0	1	0	1
1	1	0	1	1	0
0	0	1	0	1	0

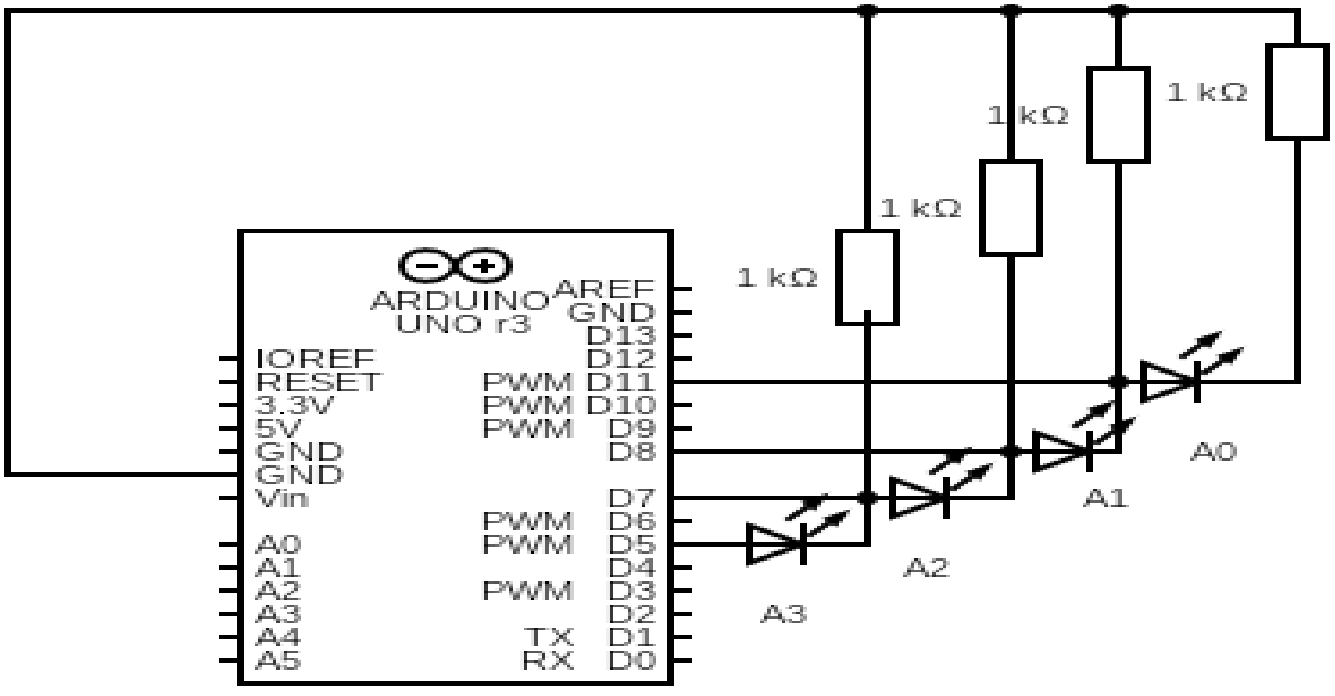
TinkerCAD link: https://www.tinkercad.com/things/0zXjb2XfBXI

PART C : JK Master-Slave Flip-Flop

AIM: To understand and build a 4-bit Up-Down Counter.

Electronic components used: Wires ,3 LED, Resistors, Arduino uno, breadboard.

Reference circuit:



Procedure :

1.Simply connect LEDS to pins on Arduino, where they will get their inputs.

2.Code is the main component of this practicall. We use timer library for this purpose.

Code: Its written in the code part of the TinkerCAD circuit, whose link is provided below.

Conclusion:
The ripple counter first goes UP from 0 (0000) to 15 (1111), then goes DOWN from 15 to 0, then goes UP, and this cycle repeats until the simulation is stopped manually.

If A3A2A1A0 represents the 4-bit output, and if x is the time preiod that A0 takes to toggle, A1 will take 2*x time, A2 will take 4*x and A3 will take 8*x time to toggle while going from (0000->1111 or 1111->0000).

TinkerCAD circuit link:
https://www.tinkercad.com/things/emzilHUn5cC