LAB REPORT – 5

Name: Vanshita

Roll Number: 2021101102

Group: 6

Aim of the Experiment:

To build an RS latch, a JK Master-Slave Flip-Flop, and a 4-bit Up-Down Counter in Tinkercad.

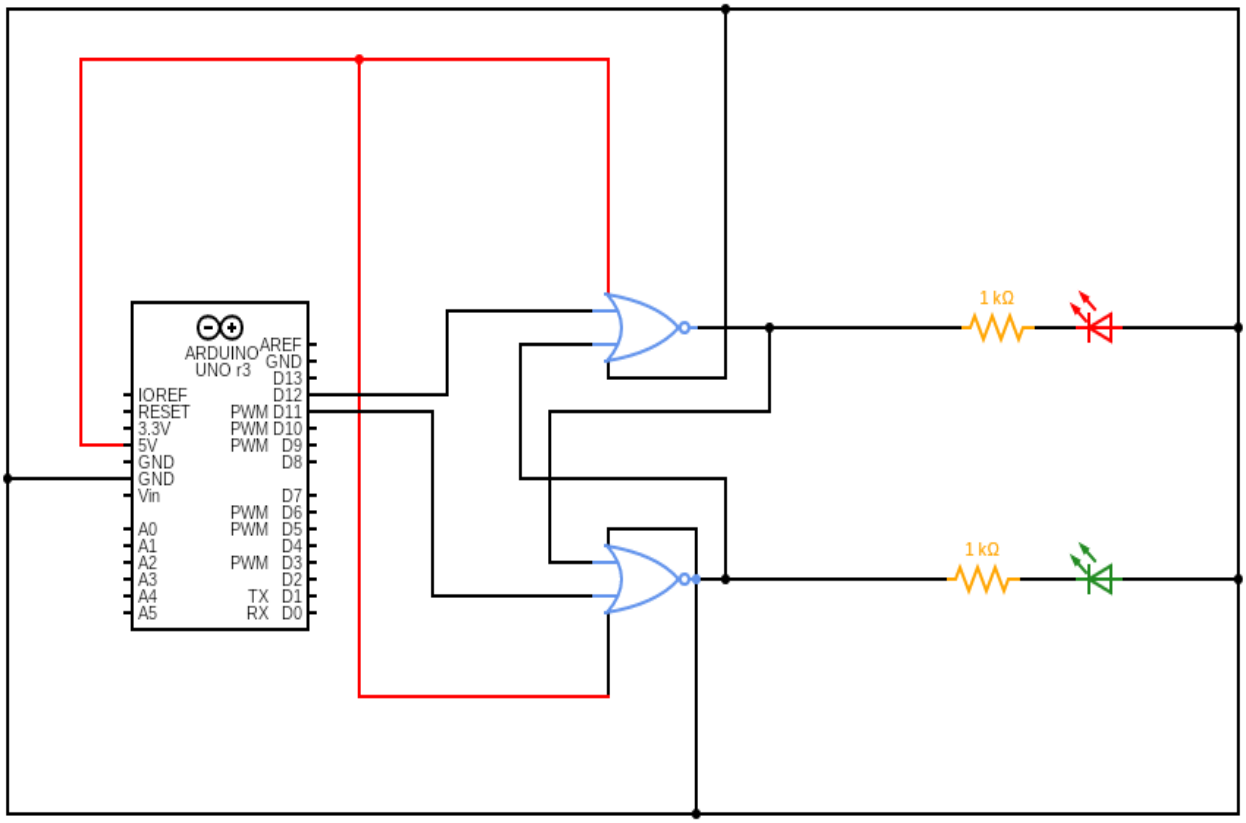
PART A: SR LATCH

Electronic components used:

1. Small Breadboard
2. 74HC02 IC (2)
3. Push Button
4. Connecting Wires
5. Arduino UNO
6. LEDs (2)

7. Resistors (3)

Reference Circuit:



Procedure:

1. Drag all electronic components to Tinkercad working area.
2. Assemble them to form a NOR latch.

3. Attach push button as shown in reference circuit.
4. Connect the LEDs to the output of each NOR gate.
5. Write appropriate code and start simulation.

Code:

```
int pin1 = 11;
int pin2 = 12;
int x,y,k;
void setup()
{
pinMode(pin1, OUTPUT);
pinMode(pin2, OUTPUT);
Serial.begin(9600);
}
void loop()
{

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

```
Serial.println(x);



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

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

Serial.print("Enter anything to go to read again:");
while(Serial.available() == 0){}
k = Serial.read();
}
```

Observation:

S	R	OUTPUT
0	1	Q'_n

0	0	Q'_n
1	0	Q_n
0	0	Q_n
0	1	Q'_n
1	0	Q_n
0	1	Q'_n
0	0	Q'_n
1	1	Both Off
0	0	Q'_n
1	0	Q_n
1	1	Both Off

Q. Till when can latch be expected to behave correctly?

A. We can see the latch behaves correctly till it encounters the input S=R=1. At S=R=1, both the LEDs are OFF. After that, despite S=R=0 being the hold state, the output changes and Q'_n LED is lit up, when in fact the previous output should have been shown. Hence, inputting S=R=1 makes S=R=0 an unpredictable state hence it is forbidden.

Conclusion:

We observe that on giving the input S=R=1, both the LEDs show same output and then change their output on giving S=R=0, defying the fact that 0,0 is the hold state for an S-R Latch using NOR gates. Hence, 1,1 is the forbidden input in a NOR gate S-R Latch.

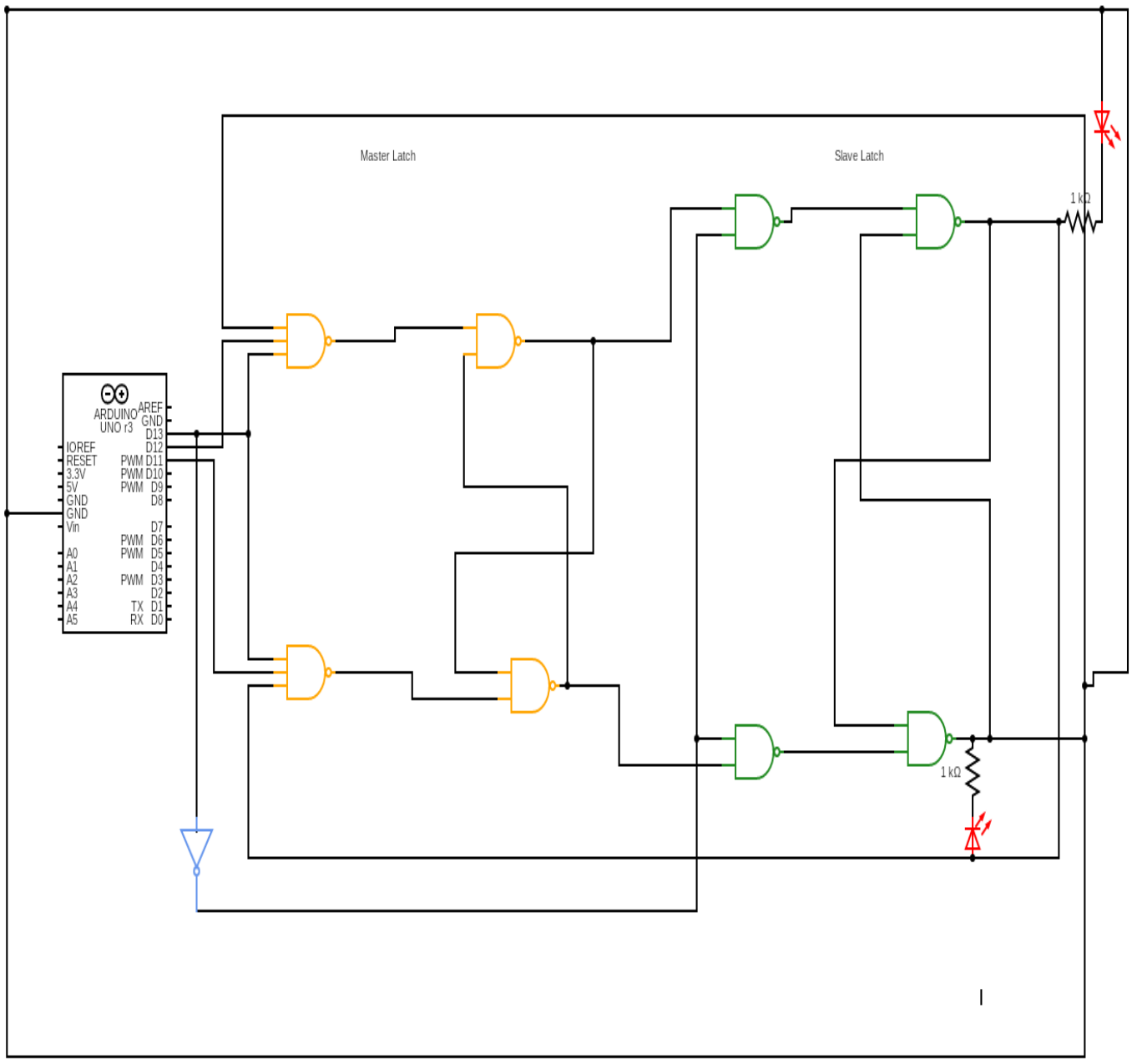
Link for Tinkercad Simulation Circuit part A

PART B: JK MASTER-SLAVE FLIP FLOP

Electronic components used:

1. Arduino UNO
2. Breadboard
3. Connecting Wires
4. Push Button
5. 74HC04 IC
6. 74HC00 IC(4)
7. LEDs (2)
8. Resistors (2)

Reference Circuit:



Procedure:

1. Drag all the components to the Tinkercad working area.
2. Assemble the circuit as shown in the figure.

3. Make the master latch using 2 3-input NAND gates and 2 Quad NAND gates as shown.
4. Make the slave latch using 4 Quad NAND gates as shown.
5. Attach 1 LED each to the two outputs of slave latch.
6. Write the required code and start simulation.

Code:

```
int pin1 = 11;
int pin2 = 12;
int pin3 = 13;
int x,y,z,k;

void setup()
{
pinMode(pin1, OUTPUT);
pinMode(pin2, OUTPUT);
pinMode(pin3, OUTPUT);
Serial.begin(9600);
}
void loop()
```

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

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

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


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

```
digitalWrite(pin3, z);


Serial.print("Enter anything to go to read again");
while(Serial.available()==0){}
k=Serial.read();
}
```

Observation:

J	K	ACTION	NEXT STATE(Q_{n+1})
1	0	Set	Q_n
0	0	Hold	Q_n
0	1	Reset	Q'_n
0	0	Hold	Q'_n
1	0	Set	Q_n
0	1	Reset	Q'_n

0	0	Hold	Q'_n
1	1	Toggle	Q_n
0	0	Hold	Q_n
0	1	Reset	Q'_n

Conclusion:

It can be observed that on giving different inputs to J and K, desired outputs in terms of glowing LEDS are obtained as shown in the table above.

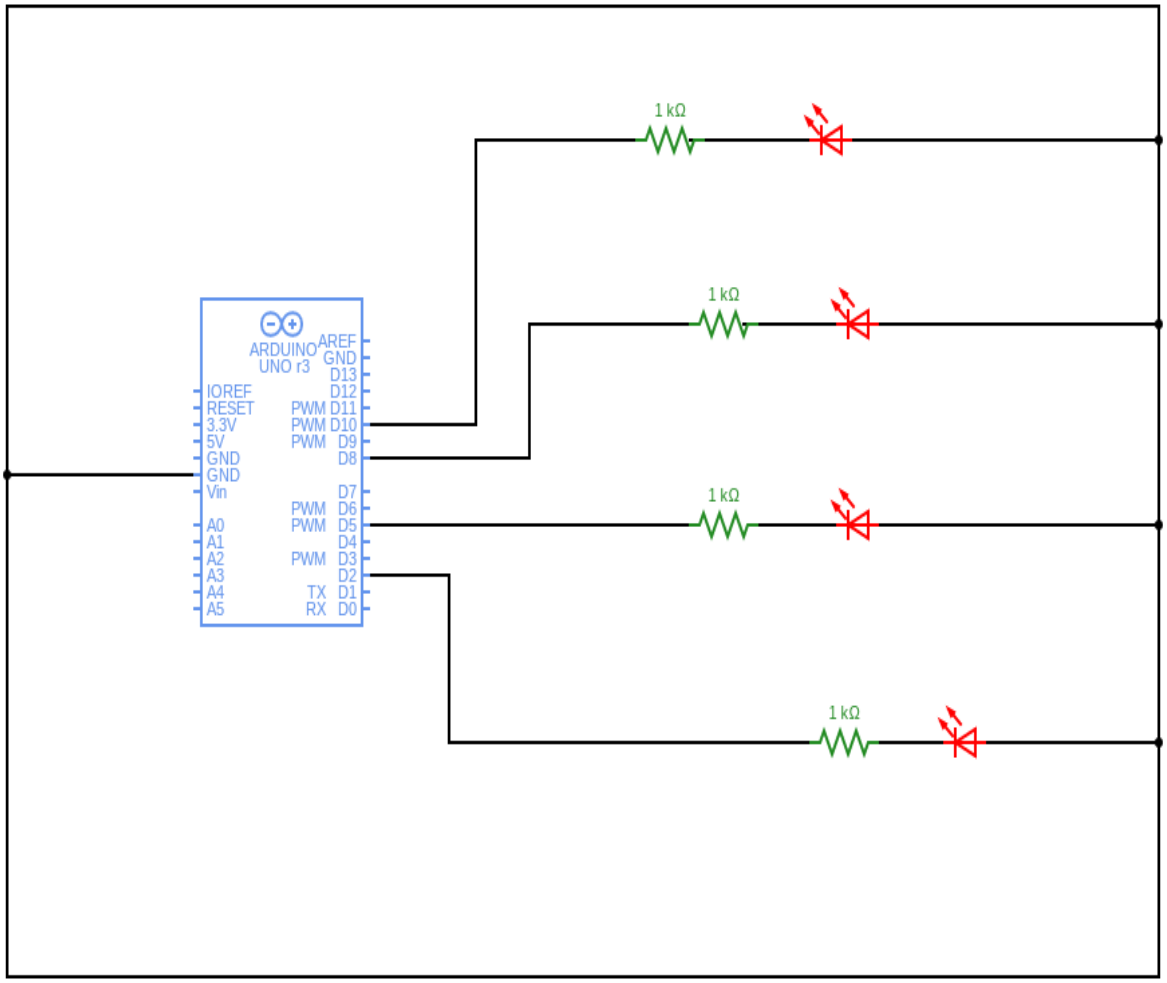
Link for Tinkercad Simulation Circuit part B

PART C: 4 BIT UP-DOWN COUNTER

Electronic components used:

1. Small Breadboard
2. Resistors (4)
3. LEDs (4)
4. Arduino UNO

Reference Circuit:



Procedure:

1. Make the arrangement as shown in the figure above by connecting 4 LEDs to Arduino inputs and one resistor each.
2. Write the required code in the code section such that the LED pertaining to the most significant bit has a rest time of 2000 milliseconds, which is 8 times that of the LED pertaining to the most significant bit which has a rest time of 250 milliseconds.

Code:

```
Timer t;
int eventId1;
int eventId2;
int eventId3;
int eventId4;
 void setup()
{
Serial.begin(9600);
pinMode(2, OUTPUT);
pinMode(5,OUTPUT);
pinMode(8,OUTPUT);
```

```
pinMode(10, OUTPUT);


Void upcount();
{


If (eventId1 < 0)
{
Serial.println("Could not initialize timer");
}
if (eventId2 < 0)
{
Serial.println("Could not initialize timer");
}
if (eventId3 < 0)
{
Serial.println("Could not initialize timer");
}
if (eventId4 < 0)
{
Serial.println("Could not initialize timer");
}
}
```

```
// 1 unit of your timer = 1000ms in real time


void loop()
{
t.update();
}


void upcount()
{
                        // starts from low, oscillates on the pin every 1000ms
eventId1 = t.oscillate(2, 1000, LOW);
                        // starts from low, oscillates on the pin every 2000ms
eventId2 = t.oscillate(5, 2000, LOW);
                        // starts from low, oscillates on the pin every 40000ms
eventId3 = t.oscillate(8, 4000,LOW);
                        // starts from low, oscillates on the pin every 8000ms
eventId4 = t.oscillate(10, 8000,LOW);


t.after(8005,stopper);


t.after(8010,downcount);
```

```
}

void stopper()
{
t.stop(eventId1);
t.stop(eventId2);
t.stop(eventId3);
t.stop(eventId4);
}
Void downcount()
{
                    // starts from high, oscillates on the pin every 1000ms
eventId1 = t.oscillate(2, 1000, HIGH);
                     // starts from high, oscillates on the pin every 2000ms
eventId2 = t.oscillate(5, 2000, HIGH);
                    // starts from high, oscillates on the pin every 4000ms
eventId3 = t.oscillate(8, 4000,HIGH);
                    // starts from high, oscillates on the pin every 8000ms
eventId4 = t.oscillate(10, 8000,HIGH);
t.after(8005,stopper);
```

```
t.after(8010,upcount);
}

// "every" X milliseconds
void stopAllTimers()
{

}
```

Observation:

We see the following sequence of numbers on screen, if ON state has value 1 and OFF state has value 0:

0000

0001

0010

0011

0100

0101

0110

0111

1000

1001

1010

1011

1100

1101

1110

1111

We see that the leftmost (most significant bit) changes after every 8 outputs, the next bit changes after every 4 outputs, the next to rightmost bit changes after every 2 outputs and the rightmost (least significant bit) changes after every output in a sequence of binary digits from 0 to 15.

Conclusion:

This fact can be used to design a circuit with each LED corresponding to each bit having an appropriate time delay to display the sequence of binary numbers 0 to 15 on screen.

Link for Tinkercad simulation circuit part C