**DSM Lab Report**

**Name:** Anirudh Kaushik

**Roll No:** 2020111015

**Group Number:** 5

**Experiment Part A**

**Objective**

After completing this lab experiment, you will have built an RS latch.

**Experiment setup/ procedure**

**Materials required:**

|  |  |
| --- | --- |
| 1 | Arduino Uno R3 |
| 1 | Quad NOR gate |
| 2 | 1 kΩ Resistor |
| 1 | Blue LED |
| 1 | Red LED |
| 1 | Pushbutton |
| 1 | Voltage Multimeter |

**Procedure:**

1. Assemble a NOR latch using two NOR gates as shown in figure 1.
2. R and S are to be taken as input from user.
3. Display Q and Q’ outputs on two LEDs.

**Code:**

int a = 13;

int b = 12;

int A,B;

int x;

void setup() {

pinMode(a, OUTPUT);

pinMode(b, OUTPUT);

digitalWrite(13,LOW);

digitalWrite(12,LOW);

Serial.begin(9600); //opens serial port, sets data rate to 9600 bps

}

void loop() {

if(Serial.available() > 0)

{

while(!Serial.available())

{}

A = Serial.read(); // x would be an integer between 0 and 255

// depending on the ascii value of the character read

A = A - '0'; // Subtracting ascii value of 0 from x.

if (A == 1)

{

digitalWrite(13,HIGH);

Serial.print("R = ");

Serial.println(A);

}

if(A == 0)

{

digitalWrite(13,LOW);

Serial.print("R = ");

Serial.println(A);

}

while(!Serial.available())

{}

B = Serial.read(); // x would be an integer between 0 and 255

// depending on the ascii value of the character read

B = B - '0';

if(B == 1)

{

digitalWrite(12,HIGH);

Serial.print("S = ");

Serial.println(B);

}

if(B == 0)

{

digitalWrite(12,LOW);

Serial.print("S = ");

Serial.println(B);

}

Serial.println("====");

}

//Use the value of x in code.

}

**Observations**

|  |  |  |  |
| --- | --- | --- | --- |
| **S** | **R** | **Q** | **Q’** |
| 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |

**When given the above inputs, explain till when the latch can be expected to operate**

**correctly and why. Cross check your theoretical understanding with the observed**

**behaviour of the latch:**

The latch is expected to correctly behave up until we input 1,1 to the SR latch after which both the LEDs turn off. When we resume our stream of inputs everything is same as it was as is evident from the I/O table. This is in accordance with the theory as 1,1 is considered the undetermined state and hence it is hard to predict how the SR latch will behave. In our case for 1,1 the output was 0,0.

**Conclusion**

Successfully made an SR latch using NOR gates.

**Tinkercad link with circuit**

**Circuit Diagram**

|  |
| --- |
|  |

**Experiment Part B**

**Objective**

After completing this lab experiment a JK Master-Slave Flip-Flop.

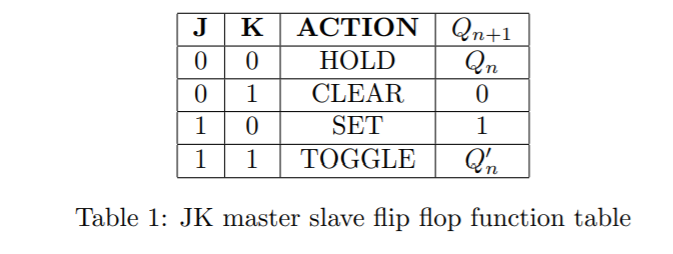
**Experiment setup/ procedure**

**Materials required:**

|  |  |
| --- | --- |
| 1 | Arduino Uno R3 |
| 1 | Triple 3-Input NAND gate |
| 2 | Quad NAND gate |
| 1 | 917.3156689848968 Hz, 5 V, 2.5 V, Square Function Generator |
| 1 | Hex Inverter |
| 1 | Voltage Multimeter |
| 2 | 1 kΩ Resistor |
| 1 | Blue LED |
| 1 | Red LED |
| 1 | Pushbutton |
| 1 | Slideswitch |

**Procedure:**

1. The JK master-slave flip flop consists of two latches: a master latch and a slave latch.
2. The master latch changes its values on the leading edge of the clock, whereas the slave
3. latch changes its values on the trailing edge of the clock.
4. Add a ”Power supply” to your working area. Make sure Power supply is set to 5V
5. and 5A• Assemble NAND gates as shown in the figure 2 to form this flip flop. Ideally, you can complete this circuit using only one 74HC10 (triple input NAND) gate and two 74HC00 gates (two input NAND). However, feel free to use more gates for convenience.



**Code:**

int a = 13;

int b = 12;

int A,B;

int x;

void setup() {

pinMode(a, OUTPUT);

pinMode(b, OUTPUT);

digitalWrite(13,LOW);

digitalWrite(12,LOW);

Serial.begin(9600); //opens serial port, sets data rate to 9600 bps

}

void loop() {

if(Serial.available() > 0)

{

while(!Serial.available())

{}

A = Serial.read(); // x would be an integer between 0 and 255

// depending on the ascii value of the character read

A = A - '0'; // Subtracting ascii value of 0 from x.

if (A == 1)

{

digitalWrite(13,HIGH);

Serial.print("J = ");

Serial.println(A);

}

if(A == 0)

{

digitalWrite(13,LOW);

Serial.print("J = ");

Serial.println(A);

}

while(!Serial.available())

{}

B = Serial.read(); // x would be an integer between 0 and 255

// depending on the ascii value of the character read

B = B - '0';

if(B == 1)

{

digitalWrite(12,HIGH);

Serial.print("K = ");

Serial.println(B);

}

if(B == 0)

{

digitalWrite(12,LOW);

Serial.print("K = ");

Serial.println(B);

}

Serial.println("====");

}

//Use the value of x in code.

}

**Observations**

|  |  |  |  |
| --- | --- | --- | --- |
| **J** | **K** | **Q** | **Q`** |
| 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 |

These observations are in accordance to the function table provided, hence it can be concluded that out JK Master Slave flip-flop works as expected.

**Conclusion**

Successfully made a JK Master Slave flip-flop and verified its reading.

**Tinkercad link with circuit**

**Circuit Diagram**

|  |
| --- |
|  |

**Experiment Part C**

**Objective**

After completing this lab experiment, you will have a 4-bit Up-Down Counter in Tinkercad.

**Experiment setup/ procedure**

**Materials required:**

|  |  |
| --- | --- |
| 1 | Arduino Uno R3 |
| 4 | 1 kΩ Resistor |
| 4 | Red LED |

**Procedure:**

1. Using the Timer library, implement a 4-bit counter. The bit outputs of the 4-bit ripple
2. counter will be represented by LEDs. (One LED for each bit)
3. Initialize a Timer t and use t.oscillate function to toggle the pin values in a predefined time period (each pin will have a different time period)
4. Down counter: once the ripple counter reaches 15 (1111), make it go down to 0.
5. You will need to stop existing timers using t.stop. You can use t.every to fire a function after a set-interval of time (hint: 2^4 time units) that will do two tasks: stop all timers and restart them in opposite direction.

**Code:**

Timer t;

int pin13 = 13;

int pin12 = 12;

int pin11 = 11;

int pin10 = 10;

int eventId13;

int eventId12 ;

int eventId11 ;

int eventId10;

int eventId130;

int eventId120;

int eventId110;

int eventId100;

void setup()

{

t.stop(eventId130);

t.stop(eventId120);

t.stop(eventId110);

t.stop(eventId100);

int condition = 0;

Serial.begin(9600);

pinMode(pin13, OUTPUT);

pinMode(pin12, OUTPUT);

pinMode(pin11, OUTPUT);

pinMode(pin10, OUTPUT);

digitalWrite(13, LOW);

digitalWrite(12, LOW);

digitalWrite(11, LOW);

digitalWrite(10, LOW);

//int count = 0;

//delay(1000);

// starts from high, oscillates on the pin every 500ms

eventId13 = t.oscillate(pin10, 500, LOW);

eventId12 = t.oscillate(pin11, 1000, LOW);

eventId11 = t.oscillate(pin12, 2000, LOW);

eventId10 = t.oscillate(pin13, 4000, LOW);

//int eventId130 = t.oscillate(pin13, 500, HIGH);

//int eventId120 = t.oscillate(pin12, 1000, HIGH);

//int eventId110 = t.oscillate(pin11, 2000, HIGH);

//int eventId100 = t.oscillate(pin10, 4000, HIGH);

if (eventId13 < 0 || eventId12 < 0 || eventId11 < 0 || eventId10 < 0)

{

Serial.println("Could not initialize timer");

}

condition = 1;

t.every(8000, stopAllTimers);

}

void loop()

{

t.update();

count++;

// Serial.print("count = ");

//Serial.println(count);

}

// "every" X milliseconds

void stopAllTimers()

{

t.stop(eventId13);

t.stop(eventId12);

t.stop(eventId11);

t.stop(eventId10);

//delay(500);

eventId130 = t.oscillate(pin10, 500, HIGH);

eventId120 = t.oscillate(pin11, 1000, HIGH);

eventId110 = t.oscillate(pin12, 2000, HIGH);

eventId100 = t.oscillate(pin13, 4000, HIGH);

t.every(8000,setup);

}

**Observations**

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.

**Conclusion**

Successfully made a 4 bit up-down counter in Tinkercad.

**Tinkercad link with circuit**

**Circuit Diagram**

|  |
| --- |
|  |