LAB REPORT : 5

NAME: ROMICA RAISINGHANI

ROLL NUMBER: 2021101053

GROUP: 1

AIM OF THE EXPERIMENT:

PART A:

To design and simulate a Set-Reset Latch(SR Latch) with the help of two NOR Gates as shown in figure 1 below and to show how the memory functions inside a processing unit.Moreover,observing the changes while operating with the SR Latche.

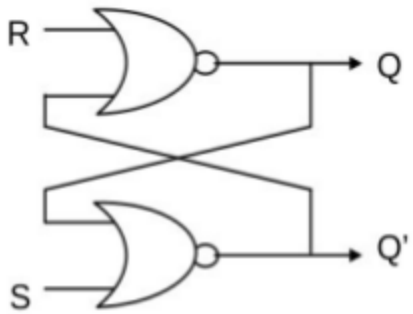


Figure 1: Schematic diagram for one RS latch

PART B:

To design and simulate a JK Master-Slave Flip-Flop with the help of one triple input NAND Gate and two Quad Nand Gates as shown in figure 2 below and to show how the memory functions inside a processing unit.Here we want to make a flip flop using 2 JK Latches named as Master and Slave that will function as a flip flop here. We then verify the functioning of the JK Master Slave Flip Flop by tabulating our observations with the theory.

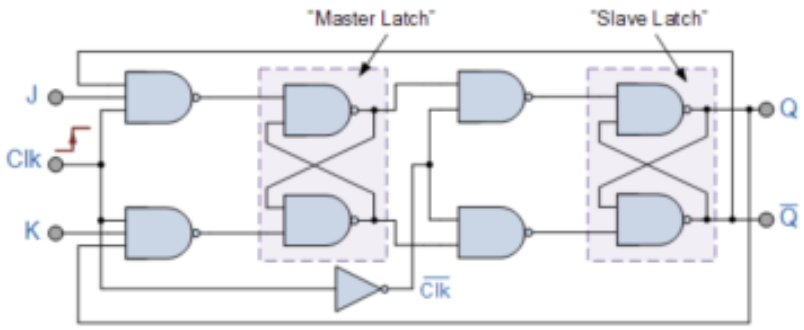


Figure 2: Schematic diagram for one master slave JK flip flop, figure extracted from [1]

PART C:

To design and simulate a 4-bit counter by representing the bit outputs of the 4-bit ripple counter by LEDs (one LED for each bit).We want to create a counter of 4 bit that counts from 0 to 15 and backwards from 15 to 0. This is called an up-down counter and the 4 LEDs and their glowing pattern represents the binary value of the count.

ELECTRONIC COMPONENTS USED:

PART A:

Arduino Uno R3, breadboard, 74HC02 Quad NOR Gate, two input LEDs R (orange) and S (green), two output LEDs Q (blue) and Q' (red), four resistors for each LEDs (200ohms each)

and some connecting wires.

PART B:

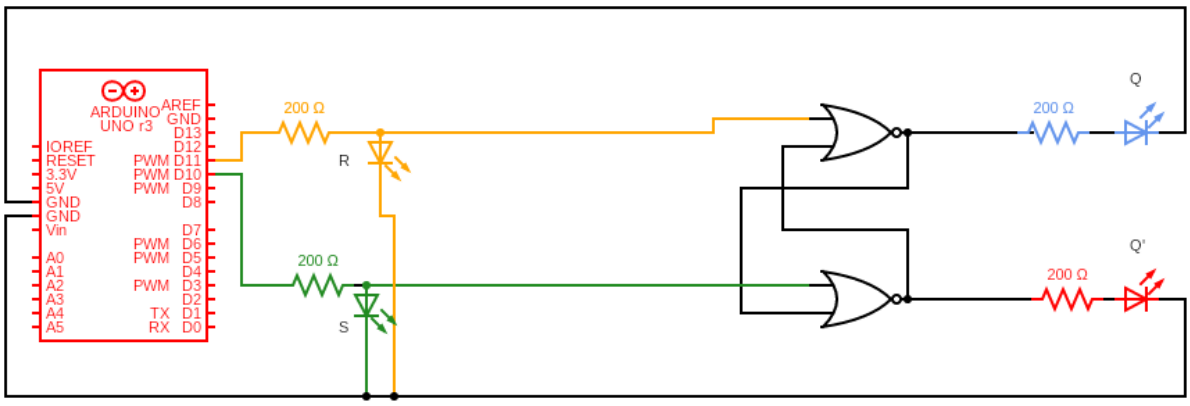
Arduino Uno R3, breadboard, one 74HC10 Triple 3-input NAND Gate, two 74HC00 Quad NAND Gates , one 74HC04 Hex Inverter, two input LEDs R (orange) and S (green), two output LEDs P (blue) and R (blue) for output of Master Latch, two output LEDs Q (red) and Q’ (red), one LED representing the clock, seven resistors for each LEDs (200ohms each), a pushbutton and some connecting wires.

PART C:

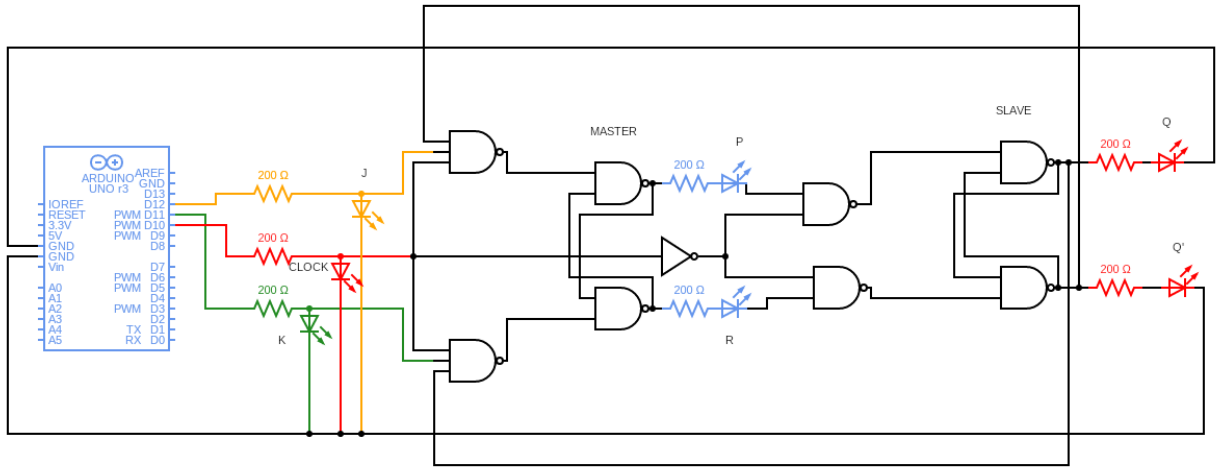
Arduino Uno R3, breadboard, four LEDs representing one bit each in binary, four resistors (200 ohms each) and some connecting wires.

REFERENCE CIRCUITS:

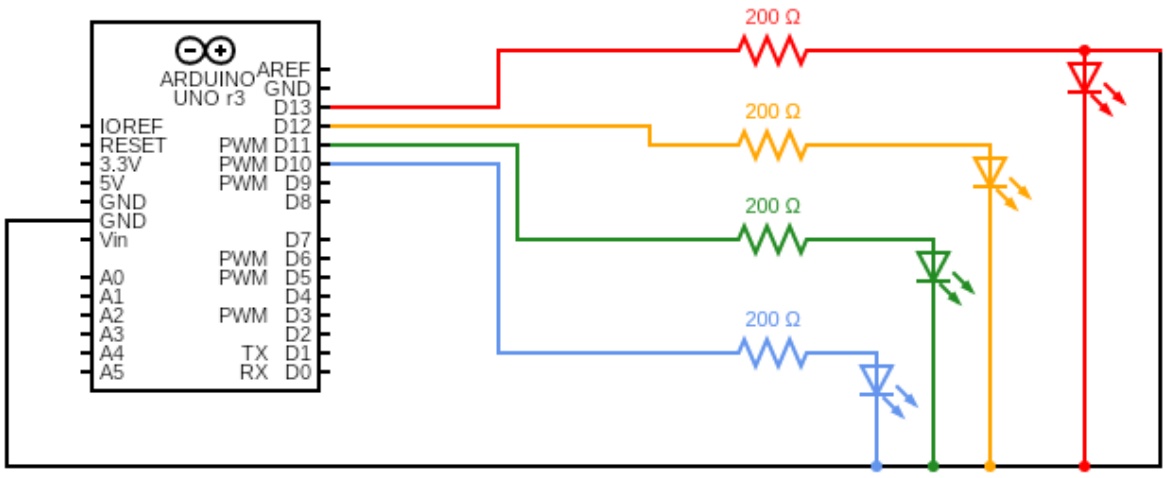
PART A: CIRCUIT DIAGRAM TO ILLUSTRATE AN SR LATCH



PART B: CIRCUIT DIAGRAM TO ILLUSTRATE A JK MASTER SLAVE FLIP-FLOP



PART C: CIRCUIT DIAGRAM TO ILLUSTRATE A 4 BIT RIPPLE COUNTER



PROCEDURE:

PART A:

1. Give power from the arduino to the breadboard and connect to nor Ic on the breadboard and give them power.

2. Now take two inputs from the user (i.e serial monitor) for this to connect two pins from the arduino to one of the two inputs of two NOR gates .

3. Now we make the cross connection for sequential circuit i.e we connect the output of other nor gate to the input of the first nor gate, similarly we connect the output of the first NOR Gate to the input of the second NOR Gate.

4. Then attach two LEDs , one each on the output of one NOR Gate. And then write the code for taking inputs.

5. Once code is written our circuit still doesn't work as it is stuck because of criss-cross connection, so we give it a nudge i.e w e give one of the outputs an initial value after which our circuit will work just fine.

6. For this place a push button on board and make all the power connections and then pass one of its wire to output say Q . Now enter the inputs from the serial monitor and observe the results.

PART B:

1 . Give power from the arduino to the breadboard and connect to nor Ic on the bread board and give them power.

2 . Now take two inputs from the user (i.e serial monitor) for this to connect two pins from the arduino to one o f the two inputs o f two 3 input NAND Gates also in each gate insert clock, now take the output and feed it as an input to a NAND Gate each.

3 . Now we make the cross connection for sequential circuit i.e we connect the output of other NAND Gate to the input of the first NOR Gate, similarly we connect the output of the first NAND Gate to the input of the second NAND Gate.

4.In turn feed the outputs of the two NAND Gate in two other NAND Gates along with inverted clock in each and feed the output to cross coupled NAND Gates.

5. Now take the Q ' and feed it back to one of the three inputs on the NAND Gate at start having J as input, similarly feed Q to NAND Gate with K.

6 . Then attach two LEDs , one each on the output of one NAND Gate. And then write the code for taking inputs.

7.Once code is written our circuit still doesn’t work as it is stuck because of criss- cross connection, so we give it a nudge i.e we give one of the outputs an initial value after which our circuit will work just fine.

8.For this place a push button on board and make all the power connections and then pass one of its wire to output say Q . Now enter the inputs from the serial monitor and observe the results.

PART C:

1. Given the code for the timer.h we have implement some changes.
2. Connect 4 leds to 4 pins in the Arduino (here pin1 = 10;pin2 = 11;pin3 = 12;pin4 = 13) and define the pinMode in void setup .
3. Define a function rippleUp for the counter to begin counting from 0 to 15 in binary which will be observed by the glow of the LEDs .
4. In the rippleUp function,define four events namely event1,event2,event3 and event4 for pins 1,2,3 and 4 respectively having time lags of 500ms each starting from position LOW using t.oscillate.
5. Define another function namely rippleDown for the counter to begin counting from 15 to 0 in binary which will be observed by the glow of the LEDs.
6. In the rippleDown function,define four events namely event5,event6,event7 and event8 for pins 1,2,3 and 4 respectively having time lags of 500ms each starting from position HIGH using t.oscillate.
7. A function namely stopAllTimersUp is defined for thr rippleUp events using t.stop.
8. A function namely stopAllTimersDown is defined for thr rippleDown events using t.stop.
9. To restart the cycle we define another function execute( ) and when this function is called in the main using t.every, the cycle restarts i.e from count 0 to 15 and 15 to 0,again 0 to 15 is counted.The time for restart is mentioned in t.every (here 1600ms).
10. Lay out an appropriate code using above steps for the circuit to simulate effectively.

CONCLUSION:

PART A:

We have successfully made an Set-Reset Latch(SR Latch) using a NOR Gate and simulated it with the help of an Arduino.

QUES : Explain till when the latch can be expected to operate correctly and why ?

The given latch is expected to work fine until we give consecutive input of 11 and 00 , as 11 being an invalid state gives Q = 0 ; and Q ’ = 0 and at 0 0 input these states should hold theoretically, but since 11 is invalid input it does not hold true as one of the gates being faster always gives the result first and the next state is decided by it hence this behaviour can be explained.

The Truth Table for the SR Latch is:

S	R	Q_t	Q'_t
0	0	Q_{t-1}	Q'_{t-1}
0	1	0	1
1	0	1	0
1	1	0	0

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

PART B:

We have successfully made a JK Master Slave Flip Flop using NAND Gates, clock and a pushbutton and simulated it with the help of an Arduino.

We found a flip flop using two latches that helps in storing the input registered and can be used to set and reset the input element and an extra option like it toggles with the Q'.

The Truth Table for the JK Master Slave Flip Flop is:

J	K	C_{lk}	Q_t	Q'_t
0	0	↑	Q_{t-1}	Q'_{t-1}
0	1	↑	0	1
1	0	↑	1	0
1	1	↑	Q'_{t-1}	Q_{t-1}

J	K	Q	Q'
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	TOGGLING	TOGGLING
0	0	1	0
1	0	1	0
1	1	TOGGLING	TOGGLING
0	0	0	1
0	1	0	1
1	1	TOGGLING	TOGGLING
0	0	1	0

NOTE: At input 11, the circuit oscillates its values between Q and Q'. As soon as we input 00 after this toggling state, the output which was high

(either Q or Q') at that particular instant, retains its value as high. The above table is according to the outputs observed .

PART C:

We have successfully made a 4-bit counter by representing the bit outputs of the 4-bit ripple counter by LEDs and simulated it with the help of an Arduino.

We created a counter that counts from 0 to 15 and after 15 it falls from 15 to 0 its like basically a timer that has 15 secs as its limits and the count is displayed by the glowing pattern of LEDs which represents binary numbers.

The code used for this simulation is embedded below :

```
//// YOUR CODE STARTS HERE

Timer t;
int pin1 = 10;
int pin2 = 11;
int pin3 = 12;
int pin4 = 13;

int event1,event2,event3,event4,event5,event6,event7,event8;

void setup() {
  Serial.begin(9600);
  pinMode(pin1, OUTPUT);
  pinMode(pin2, OUTPUT);
  pinMode(pin3, OUTPUT);
  pinMode(pin4, OUTPUT);
  execute();
  int event= t.every(16000,execute);
}

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

void rippleup(){
  event1 = t.oscillate(pin1, 500, LOW);
  event2 = t.oscillate(pin2, 1000, LOW);
```

```
  event3 = t.oscillate(pin3, 2000, LOW);
  event4 = t.oscillate(pin4, 4000, LOW);
}

void rippledown(){
  event5 = t.oscillate(pin1, 500, HIGH);
  event6 = t.oscillate(pin2, 1000, HIGH);
  event7 = t.oscillate(pin3, 2000, HIGH);
  event8 = t.oscillate(pin4, 4000, HIGH);
}

void stopAllTimersup() {
  t.stop(event1);
  t.stop(event2);
  t.stop(event3);
  t.stop(event4);
}

void stopAllTimersdown() {
  t.stop(event5);
  t.stop(event6);
  t.stop(event7);
  t.stop(event8);
}
//to restart the cycle:
void execute(){
  int up1= t.after(1,rippleup);
```

```
  int stop1=t.after(8000,stopAllTimersup);
  int down1=t.after(8000,rippledown);
  int stop2=t.after(15750,stopAllTimersdown);
}

//// YOUR CODE ENDS HERE

```

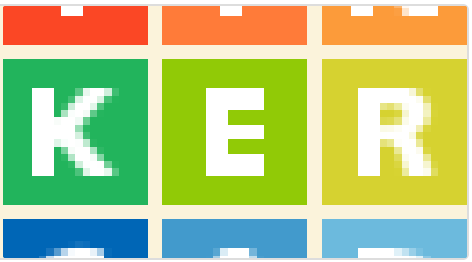
LINKS FOR TINKERCAD SIMULATION:

PART A:

Tinkercad | From mind to design in minutes

Tinkercad is a free, easy-to-use app for 3D design, electronics, and coding.

https://www.tinkercad.com/things/0m7cV4FWbxJ-lab5-sr-latch/edit?sharecode=4BqF0uZ1-lE-wIfTe8xVGpaTr2nqH-MNbhbjTxjuc-A

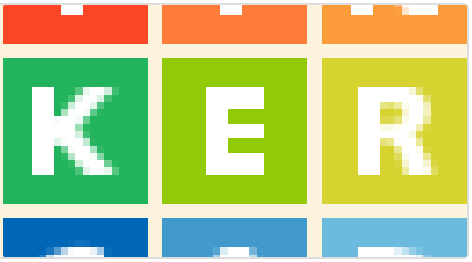


PART B:

Tinkercad | From mind to design in minutes

Tinkercad is a free, easy-to-use app for 3D design, electronics, and coding.

https://www.tinkercad.com/things/jAbKTf4qdi2-lab5-jk-master-slave-flip-flop/editel?sharecode=j-wGZNcVY9Budoj0aLwNivRmxReMz9RD3oe1AnIMaek

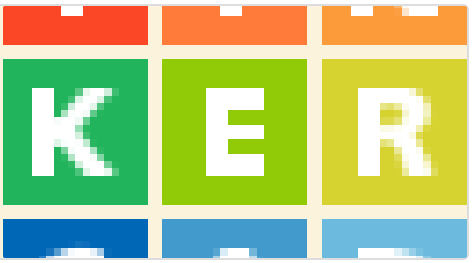


PART C:

Tinkercad | From mind to design in minutes

Tinkercad is a free, easy-to-use app for 3D design, electronics, and coding.

https://www.tinkercad.com/things/4qZQ8vHKirL-lab5-ripple-counter/editel?sharecode=wM3m47GRDs3G-cOtuLJbsJYbEs5ppu3xFB6Bqs14qwk



SCREEN RECORDINGS OF SIMULATION :

PART A:

https://youtu.be/RI6I_1Gr3o4

PART B:

https://youtu.be/iV3WkbI555c

THANK YOU!