LAB REPORT : 6

NAME: ROMICA RAISINGHANI

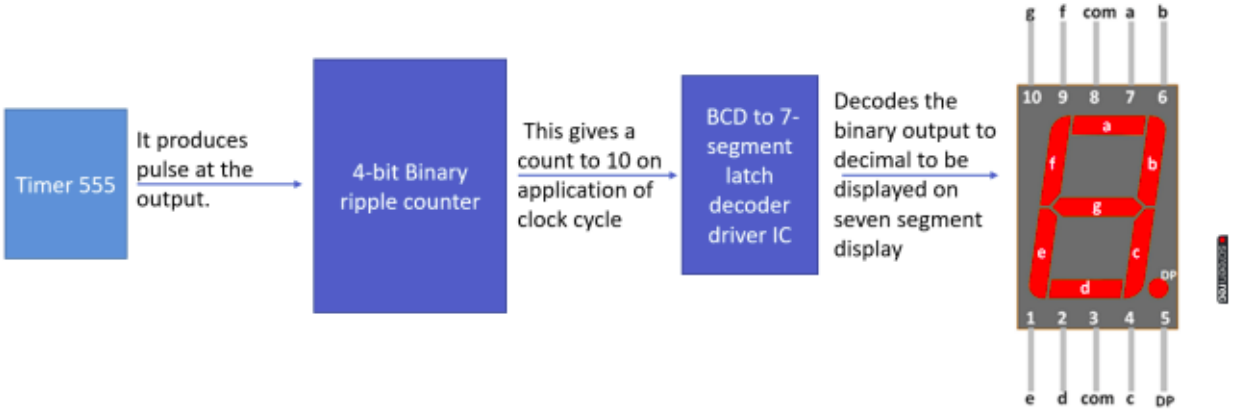
ROLL NUMBER: 2021101053

GROUP: 1

AIM OF THE EXPERIMENT:

PART A:

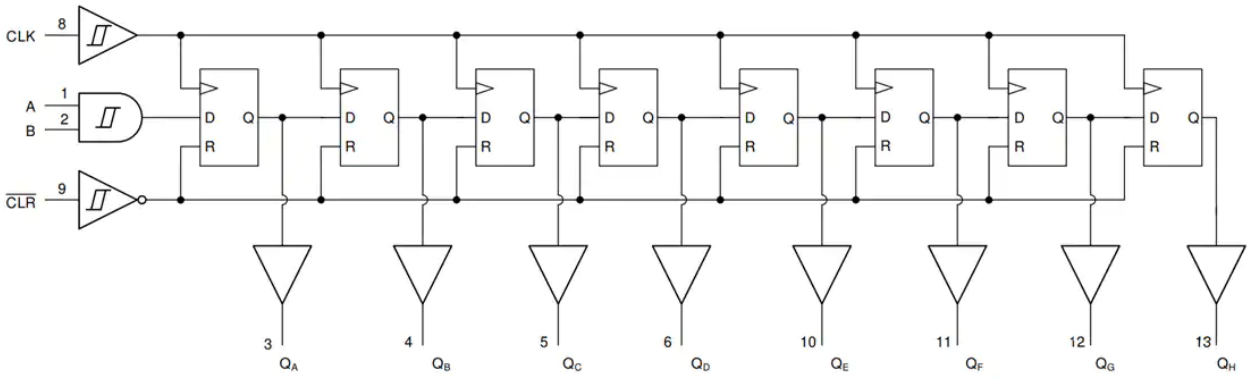
To design and simulate a decade counter that counts from 0 to 9 and then resets its value to use using a 555 timer, 4-bit binary counter, BCD to decimal converter,7-segment decoder and 7-segment LED display to display the decimal number.



Block diagram depicting components for a decade counter

PART B:

To design and simulate a circuit that explains the working of a 8-bit Shift Register by making it count from 0 to 255 in binary and then again resetting it to 0. The count is displayed by the glow of 8 LEDs in order.



Block diagram depicting the working of 8-Bit Shift Register

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:

1. Arduino Uno R3
2. 555 timer
3. 4-bit binary counter (IC 74HC93)
4. Breadboard
5. 10 uF, 16 V Polarized Capacitor

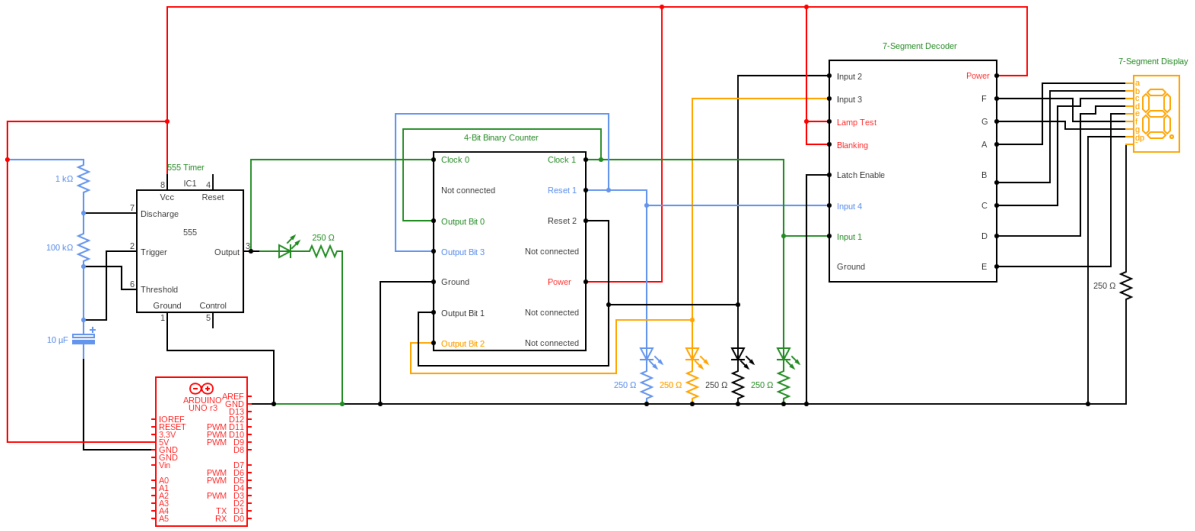
6. 100 kΩ Resistor,1 kΩ Resistor
7. Seven 250 Ω Resistor
8. Four Red LED
9. Seven Segment Decoder(IC CD4511)
10. Cathode 7 Segment Display
11. Green LED
12. A bunch of connecting wires

PART 2A and 2B:

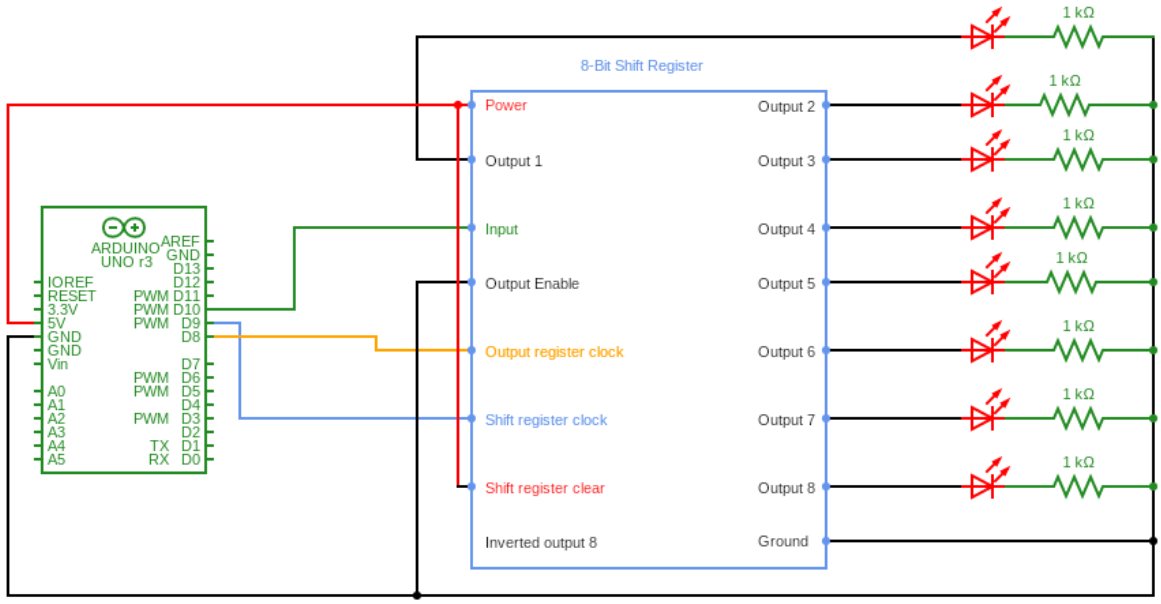
1. Arduino Uno R3,
2. Breadboard
3. 8-Bit Shift Register(IC 74HC595)
4. Eight Red LEDs
5. Eight 1 kΩ Resistors
6. A bunch of connecting wires

REFERENCE CIRCUITS:

PART 1: CIRCUIT DIAGRAM TO ILLUSTRATE A DECADE COUNTER



PART 2A & 2B: CIRCUIT DIAGRAM TO ILLUSTRATE WORKING OF A 8-BIT SHIFT REGISTER



PROCEDURE:

PART A:

1. Provide all the live wire and ground connections from the Arduino to all the breadboards first.
2. Take the capacitor of capacitance 10 micro farad and connect one of its ends to the ground wire and the other end to the Trigger Input of the 555 Timer.
3. Connect the Power Pin of the 555 Timer to the live wire and connect the Ground Pin of the 555 Timer to the ground wire.
4. Take the 1k ohm resistor and connect one of its ends to the live wire and the other end to the Discharge Pin.
5. Take the 100k ohm resistor and connect one of its ends to the Discharge pin of the 555 Timer and the other end to the Trigger Pin of the 555 Timer.
6. Connect the Threshold Pin of the 555 Timer and the Trigger Pin of the 555 Timer through a wire.
7. Take a wire from the Out Pin of the 555 Timer and connect it to the anode of LED X.
8. Take one end of the resistor and connect it to the cathode of the LED X and ground the other end.
9. Take a wire from the Out Pin of the 555 Timer and connect it to the clock 0 Pin of the 4-Bit binary counter.
10. Connect the Clock 1 Pin of the 4-Bit binary counter and Output Bit 0 of the 4-Bit binary counter with a wire.
11. Connect the the Reset 1 Pin of the 4-Bit binary counter and the Output Bit 3 of the 4-Bit binary counter with a wire.
12. Connect the the Reset 2 Pin of the 4-Bit binary counter and the Output Bit 1 of the 4-Bit binary counter with a wire.
13. Connect the Power Pin of the 4-Bit binary counter to the live wire and connect the Ground PIn of the 4-BIt binary counter to the ground wire.
14. Take a wire from the Output Bit 0 Pin of the 4-BIt binary counter and connect it to the anode of LED B.
15. Take one end of the resistor and connect it to the cathode of the LED A and ground the other end.

16. Take one end of the Output Bit 1 Pin of the 4-bit binary counter and connect it to the anode of the LED B and ground the other end.
17. Take one end of the resistor and connect it to the cathode of the LED B and ground the other end.
18. Take a wire from the Output Bit 2 of the 4-Bit binary counter and connect it to the anode of LED C.
19. Take one end of the resistor and connect it to the anode of the LED C and ground the other end.
20. Take a wire from the Output Bit 3 Pin of the 4-BIt binary counter and connect it to the anode of LED D.
21. Take one end of the resistor and connect it to the cathode of the LED D and ground the other end.
22. Connect the Output Bit 0, Output Bit 1, Output Bit 2, Output Bit 3 of the 4-BIt binary counter to Input 1, Input 2, Input 3 and Input 4 respectively of the 7-Segment Decoder.
23. Connect the Power,Lamp test and Blanking Pin of the 4-Bit binary counter to the live wire.
24. Connect the Latch Enable and Ground of the 4-Bit binary counter to the live wire.
25. Connect one end of the two resistors to the Common Pin of the 7-Segment Display and ground the other end.
26. Ground the DP Pin of the 7-Segment Display.
27. Connect the A,B,C,D,E,F,G pin of the 7-Segment Decoder to the A ,B,C,D,E,F,G Pin respectively of the 7-Segment Display.
28. Feed an approriate code into the Arduino to make the above circuit work and make sure all the connections are tight by crosscheck the wirings from the given circuit and make the necessary changes.

PART 2(A and B){Circuit Diagram}:

1 . Provide all the live wire and ground connections from the Arduino to all the breadboards first.

2 .Take a wire from pin 8 and connect it as Input into the 8-bit shift register.

3 . Take a wire from pin 10 and connect it as Output Register Clock PIn into the 8-bit shift register.

4. Take a wire from pin 9 and connect it as Shift Register Clock Pin into the 8-bit shift register.

5. Connect the Power and Shift Register Clear PIN to the live wire.

6.Connect the Ground and Output Enable Pin to the ground wire.

7.Take a wire from Output 1,Output 2,Output 3,Output 4,Output 5,Output 6,Output 7 and Output 8 and connect it to the anode of LED A,B,C,D,E,F,G and H respectively.

8. Take the 8 resistors of 1k ohm each and connect them respectively to the cathode of the respective LEDs and Ground the other ends of the resistors.

9. Feed an approriate code into the Arduino and make sure all the connections are tight by crosscheck the wirings from the given circuit and make the necessary changes.

PART 2B{CODE EXPLANATION}:

1. We initialise variables to the pins (input pin to pin 10,latch pin to pin 8 and cloc pin to pin 9).

2. Then we name a function setup in which we setup the respective 3 pins and print “Input a number between 0 to 7: “.

3. In the function loop, we initiate variable ‘A’(in the range [0,7]) and print it and we also print “Input a number between 0 to 7: “.

4. We also initiate a variable j = 1 and initiate a ‘FOR’ loop in which we initialise a variable i = 0 and iterate it through A values and increment its value by 1 in each cycle.

5. Then conditional statements of ‘IF’ are used inside the respective ‘FOR’ loop and if i = 0 then j = 1 otherwise j= 2*j.

6. When i = A (which is the maximum value that the ‘For’ loop can reach) we define the ‘shiftOut’ function which accepts 4 input pins namely ‘INPUT’ pin, ‘CLOCK’ pin,(the pin to toggle once the input Pin has been set to the correct value.) MSB/LSB first and the value of j. [shiftOut function —> Shifts out a byte of data one bit at a time. Starts from either the

most (i.e. the leftmost) or least (rightmost) significant bit. Each bit is written in turn to a data pin, after which a clock pin is pulsed (taken high, then low) to indicate that the bit is available.]

7. Further we declare the delay statement to increase the communication time between the input of the user and the Arduino.

CONCLUSION:

PART 1:

We have successfully made a decade counter that counts from zero to nine and then resets its value to zero again. To make this we used a 555 Timer, 4-Bit binary counter, binary to decimal decoder and a Seven Segment LED Display.

A decade counter also known as a binary-coded decimal is a serial digital counter that counts ten digits. It resets for every new clock input. It goes through 10 unique combinations of output. A decade counter counts in a sequence of ten and then returns back to zero after the count of nine. Obviously, to count up to a binary value of nine, the counter must have at least four flip-flops within its chain to represent each decimal digit.

PART 2A:

We have successfully made and simulated a circuit that depicts the working of a 8-Bit Shift Register. Using the circuit made above, we wrote the code to count from 0 to 255 and glow the 8 LEDs in order.

This sequential device loads the data present on its inputs and then moves or “shifts” it to its output once every clock cycle, hence the name **Shift Register**.

A *shift register* basically consists of several single bit “D-Type Data Latches”, one for each data bit, either a logic “0” or a “1”, connected together in a serial type daisy-chain arrangement so that the output from one data latch becomes the input of the next latch and so on.

Data bits may be fed in or out of a shift register serially, that is one after the other from either the left or the right direction, or all together at the same time in a parallel configuration.

PART 2B:

We have successfully made and simulated a circuit that depicts the working of a 8-Bit Shift Register. Using the circuit made above, we wrote the code to take input from the user in the range 0-7 and glow the corresponding LED. Here 0 represents the least significant bit that is the rightmost LED whereas 7 represents the most significant bit that is the leftmost LED.

The code used for this simulation is embedded below :

```
int input_pin = 10; //GREEN
int latch_pin = 8;  //ORANGE
int clock_pin = 9;  //TURQUOISE

int A;

void setup()
{
    Serial.begin(9600);
    pinMode(input_pin, OUTPUT);
    pinMode(latch_pin, OUTPUT);
    pinMode(clock_pin, OUTPUT);

    Serial.println("Enter a digit between 0 to 7 only:  ");
    //digit depicts which LED will glow starting from least significant bit
}

void loop()
{
    if (Serial.available() > 0)
    {
        A = Serial.read();
        A = A - '0';

        Serial.print("INPUT:  ");
        Serial.println(A);
```

```
27
28          int j = 1;
29          for (int i = 0; i <= A; ++i)
30          {
31              if (i == 0)
32              {
33                  j = 1;
34              }
35              else
36              {
37                  j = j * 2;
38              }
39              if (i == A)
40              {
41                  digitalWrite(latch_pin, LOW);
42                  shiftOut(input_pin, clock_pin, MSBFIRST, j);
43                  digitalWrite(latch_pin, HIGH);
44                  delay(100);
45              }
46          }
47      }
48  }
49
```

LINKS FOR TINKERCAD SIMULATION:

PART 1:

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/ifhkaz9VLjP-lab-6-c1/editel?sharecode=phzPNTCt6ADzmziOWoyWvmIyi_ZUrygwe9DdZAmw26U

PART 2A:

Tinkercad | From mind to design in minutes

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

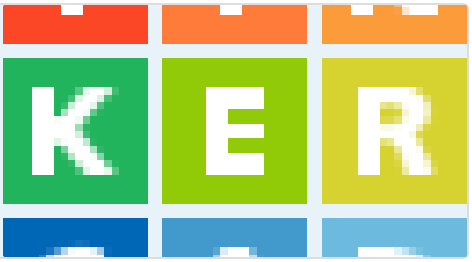
https://www.tinkercad.com/things/a10KmvaiNCw-lab6-c2a/editel?sharecode=NqcViOpeC3I-SNtGA3agREIAs7_VfBG_qlL1HqnhcJc

PART 2B:

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/ejK1jfbYPuq-lab6-c2b/editel?sharecode=KtTVyB2GHR9k7ZFfEUh6zt6pScRPW1Qs0IBv4u1_UpM



THANK YOU!