Lab 7: Finite State Machine

1) Purpose

Aim of this lab is to learn how to design and implement finite state machines. It is required to design a finite state machine and indicate its state diagram and output tables. Then the FSM design implemented on the breadboard using integrated circuits to implement digital circuit components.

2) Methodology

Designed FSM circuit operates a number guessing game. User gives a binary 2 bit unsigned number in decimal from 0 to 3. Users are expected to guess 3 numbers correctly. In each state if true number is given then next state comes, otherwise circuit stays in the same state. After 3 numbers are correctly guessed circuit gets to finish state and lights a winning led, state diagram and output table of the circuit is given below.

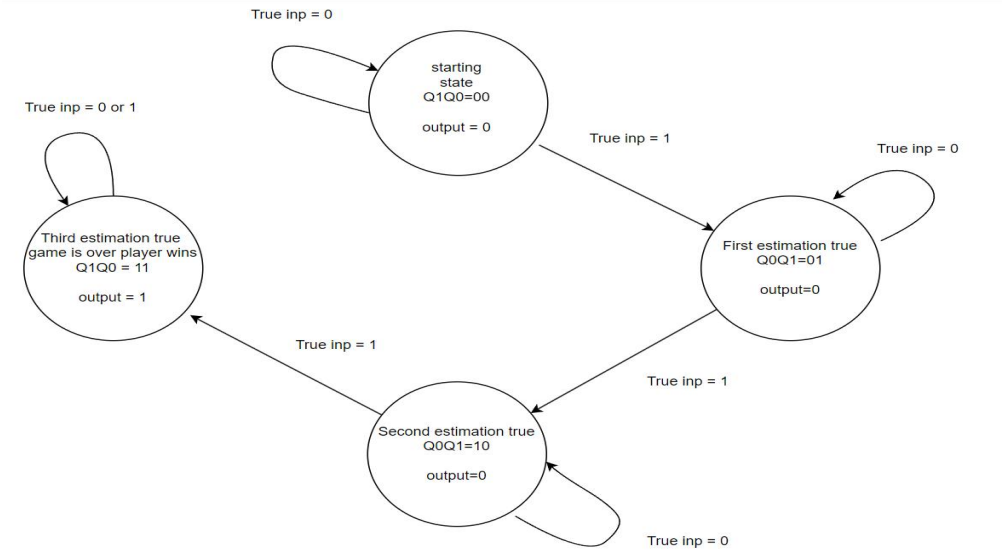

Figure 2.1: State diagram of number guessing FSM circuit.

States	Q1	Q0	True Guess Actual INP = 1			Wrong Guess Actual INP = 0			Winning output
			Next state	D1	D0	Next state	D1	D0	
S0	0	0	S	0	1	S	0	0	0
S	0	1	S	1	0	S	0	1	0
S	1	0	S	1	1	S	1	0	0
S	1	1	S	1	1	S	1	1	1

Figure 2.2: Output table of number guessing circuit.

Using K-maps we can derive that

D1 = Q1 or (Q0 and Actual INP)

D2 = (Q0' and Actual INP) or (Q0 and (Not (Actual INP) or Q1)

Where Actual INP is a signal that indicates given number is truly guessed. In state S0 10, in S1 00 and in S2 11 should be given to win the game therefore actual INP can be expressed as given number's MSB is In1 and LSB is In0.

Actual INP = Q0.In1.In0 + In1.(Q1.In0 + In0'.Q1'.Q0').

Using these boolean expressions we can produce next state combinational circuit.
To reset the game D-Flip Flop's active low 2RD and 1RD leads given with low voltage. To implement the circuit 74 HC 74 IC (2 D_Flip flop included), 74 HC 04 (6 inverters on 1 IC), 74 HC 08(4 2 input AND gates)and 74 HC 32 (4 two input OR gates)

are used on breadboard. As clock signal signal generator is connected to provide square wave with given frequency for flip flops.

3) Results

The results of the breadboard implementation can be seen in figures below. Different 2 bit numbers given in each state and circuit response's and given input signals (In1 In2) are shown by led's. In pictures led's resemble from left to right winning, q1, q0, input(1) and input (0).

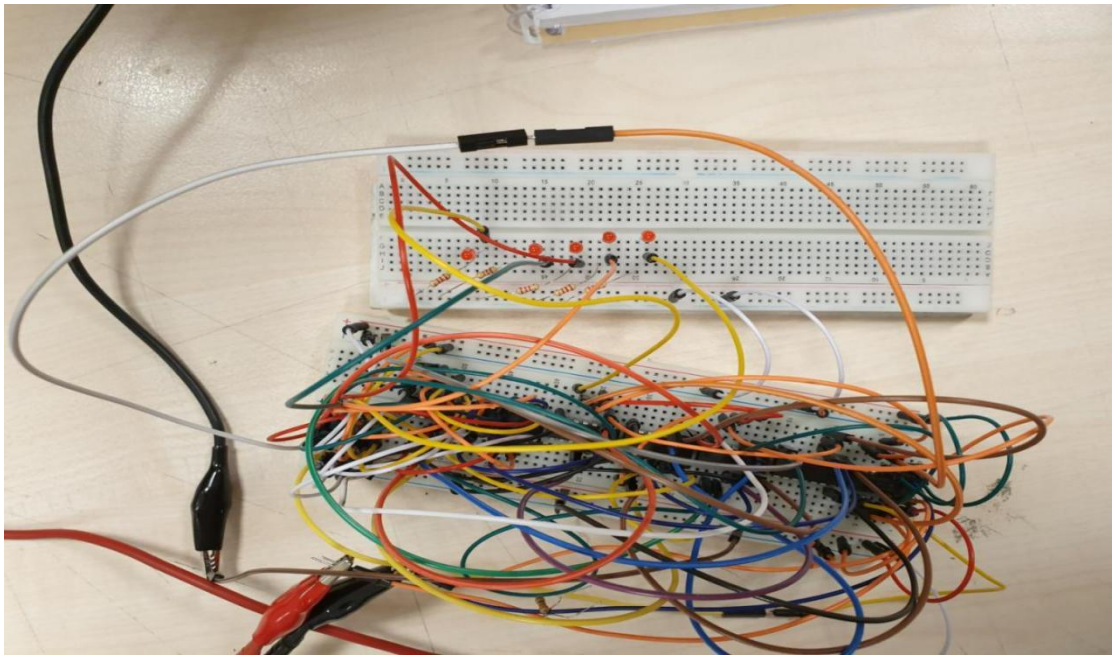

Figure 3.1: winning = 0, state = 00, input = 00

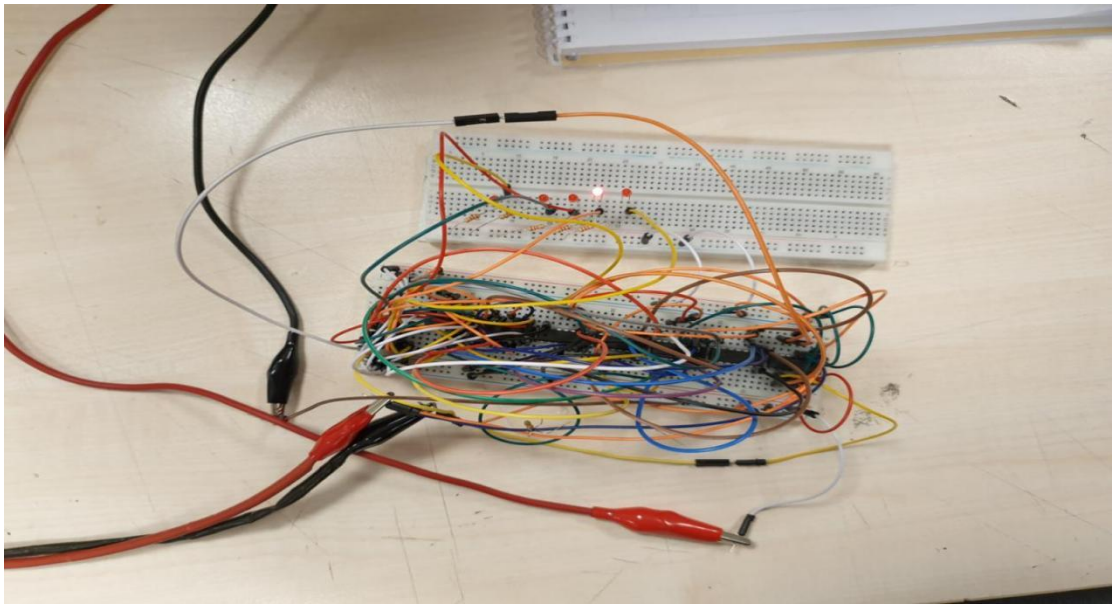

Figure 3.2: winning = 0, state = 00, input = 10

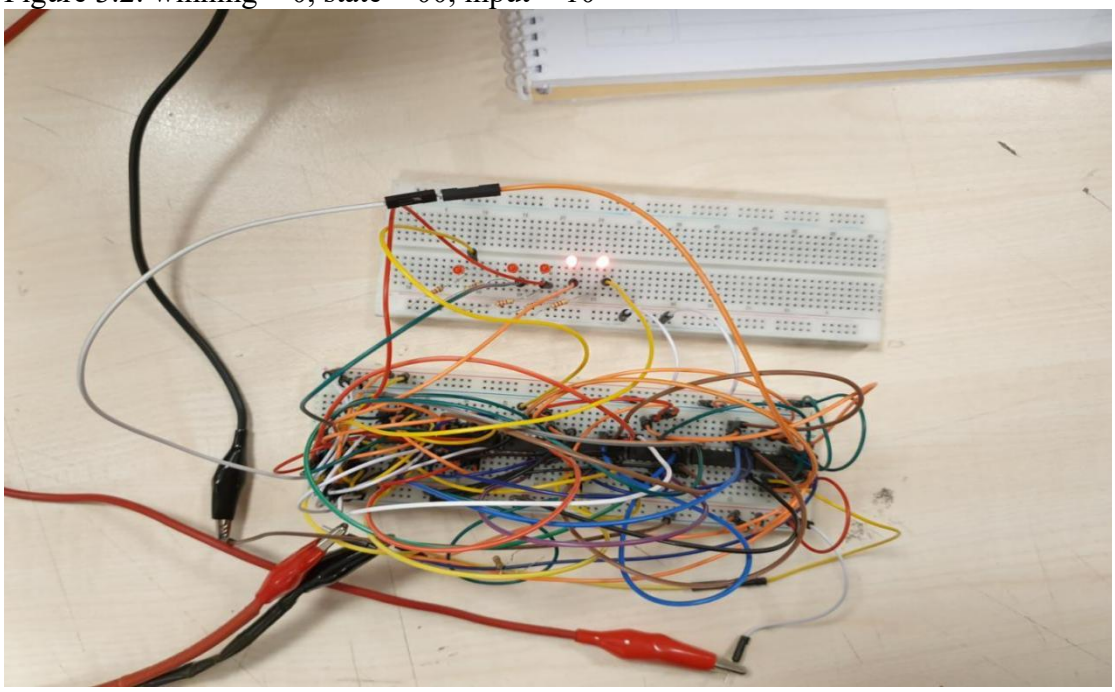

Figure 3.3: winning = 0, state = 00, input = 11

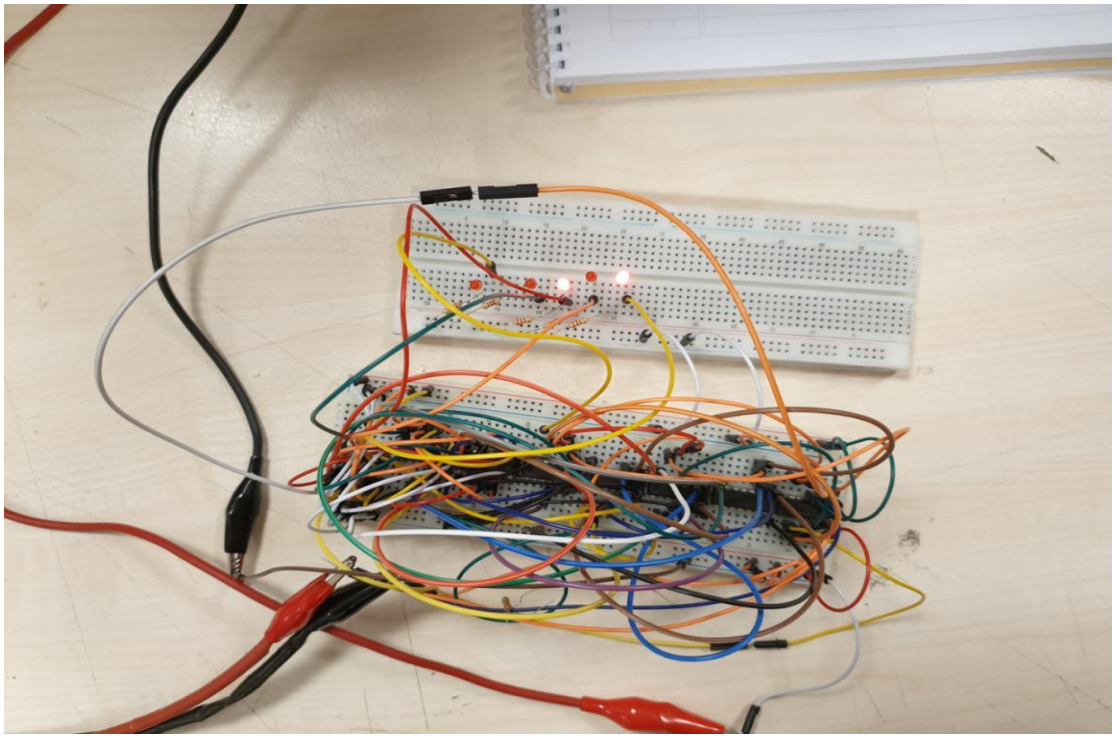

Figure 3.4: winning = 0, state = 01, input = 01, as 01 is guessed correctly at state 00 it gets state 01

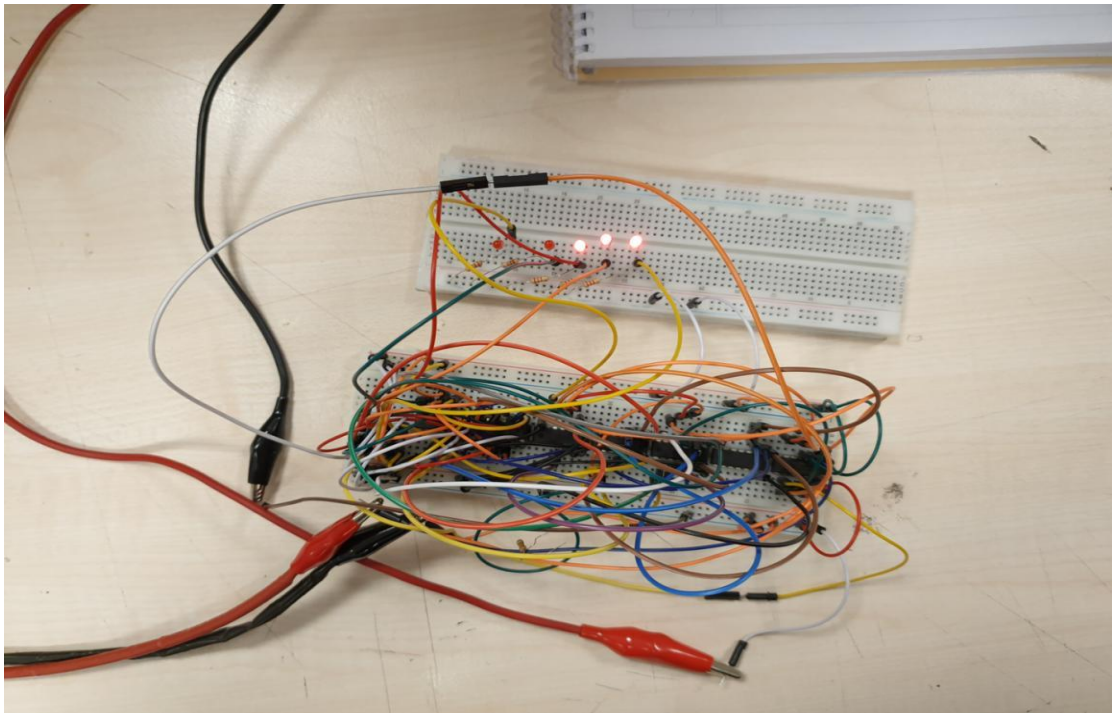

Figure 3.5: winning = 0, state = 01, input = 11

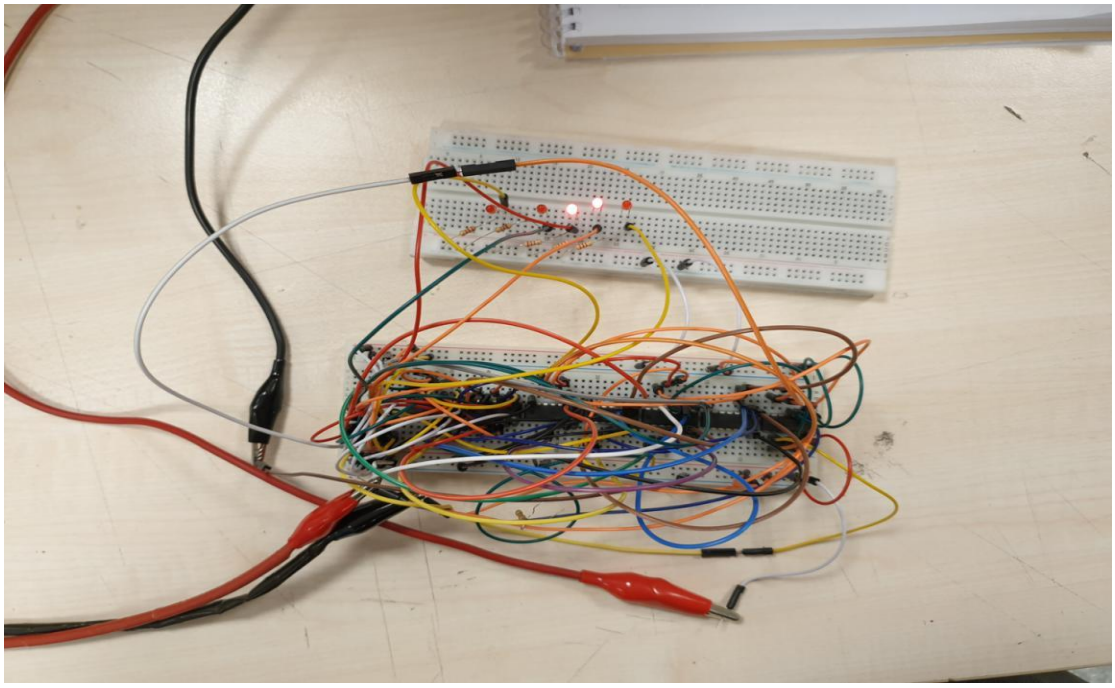

Figure 3.6: winning = 0, state = 01, input = 10

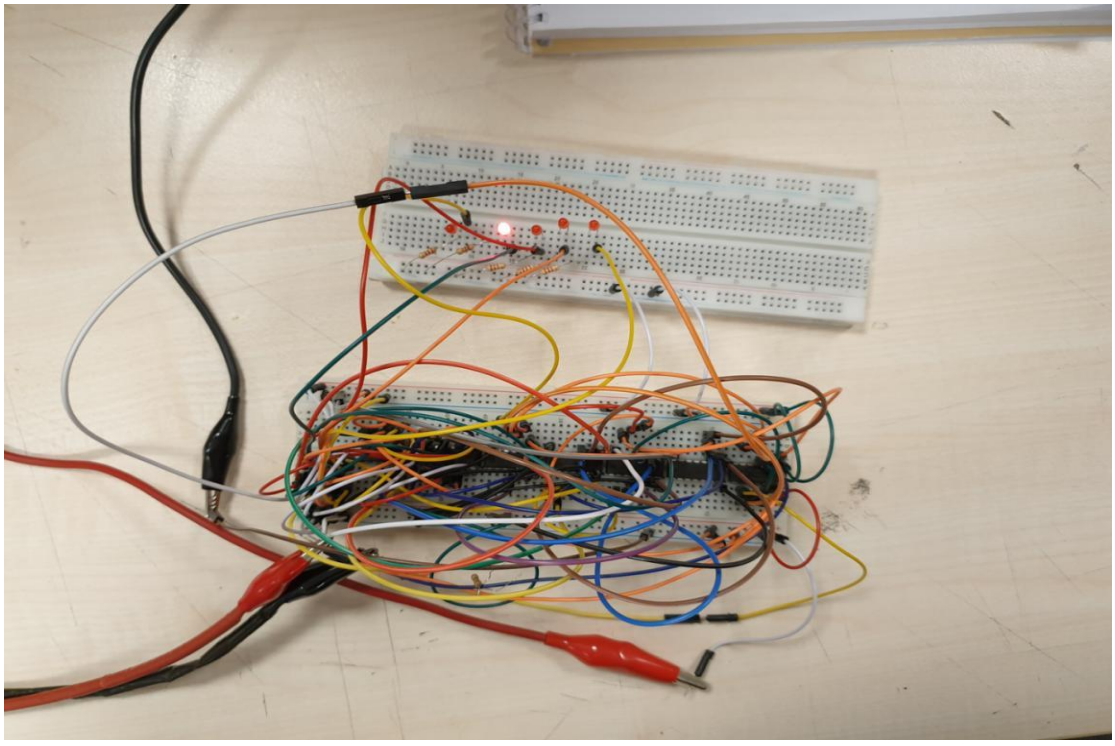

Figure 3.7: winning = 0, state = 10, input = 00 , as 00 is guessed correctly at state 01 it gets state 10

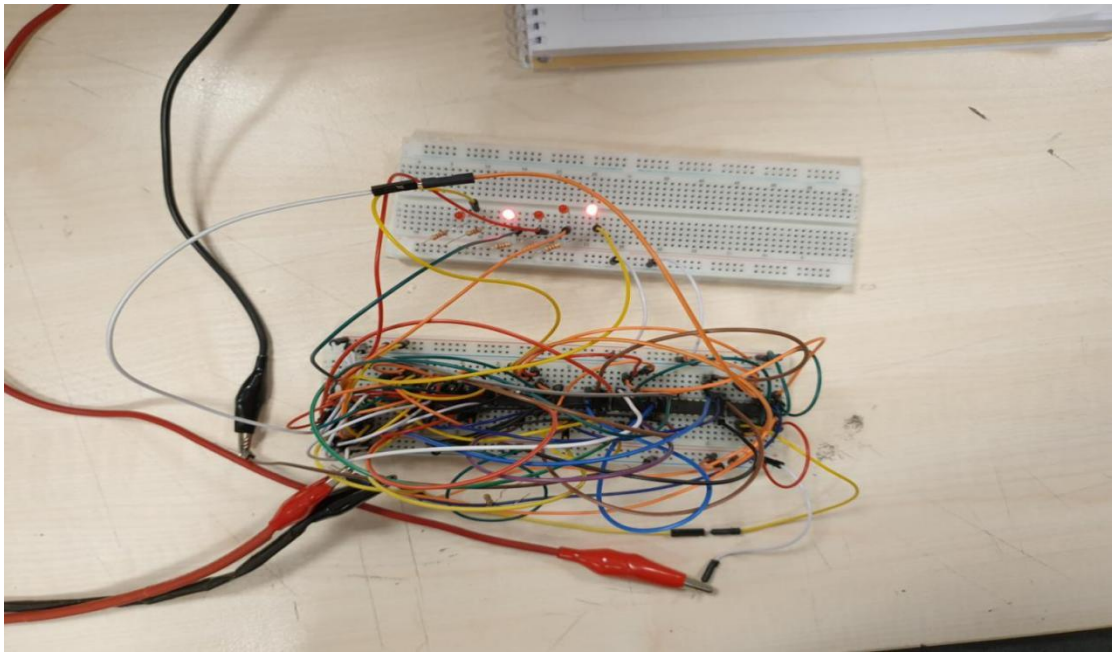

Figure 3.8: winning = 0, state = 10, input = 01

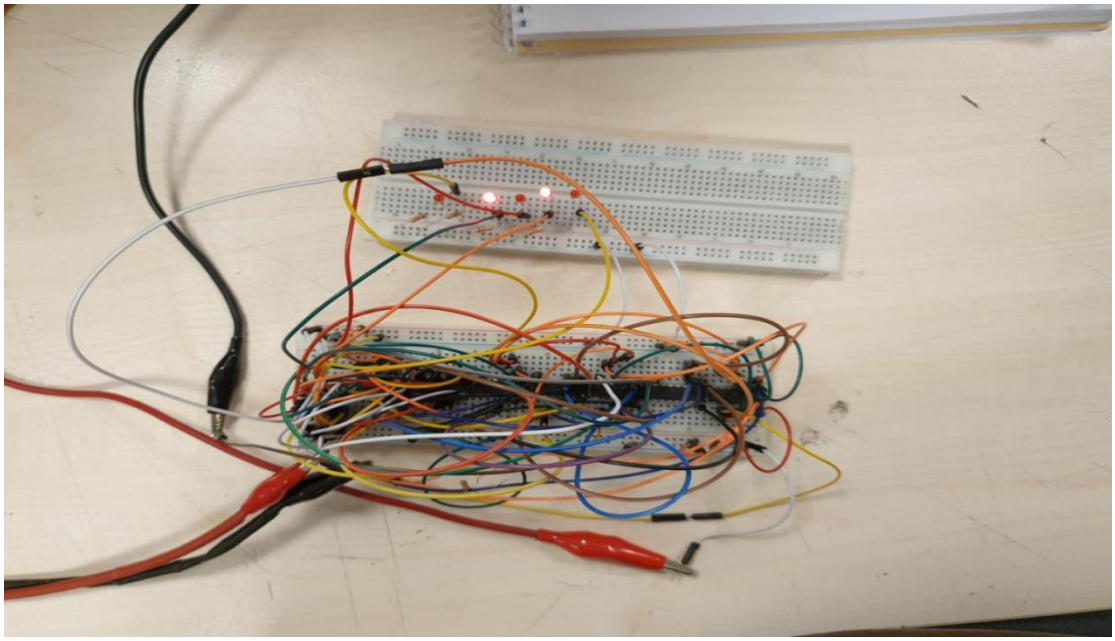

Figure 3.9: winning = 0, state = 10, input = 10

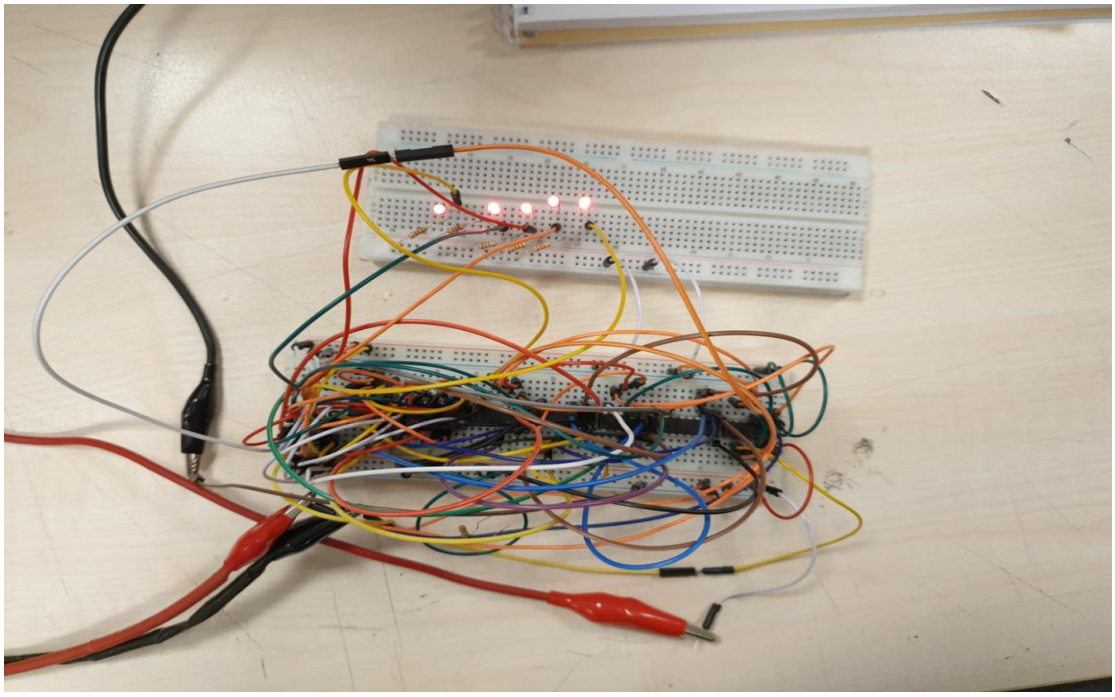

Figure 3.10: winning = 1, state = 11, input = 11 , as 11 is guessed correctly at state 10 it gets state 11 and game is over user wins.

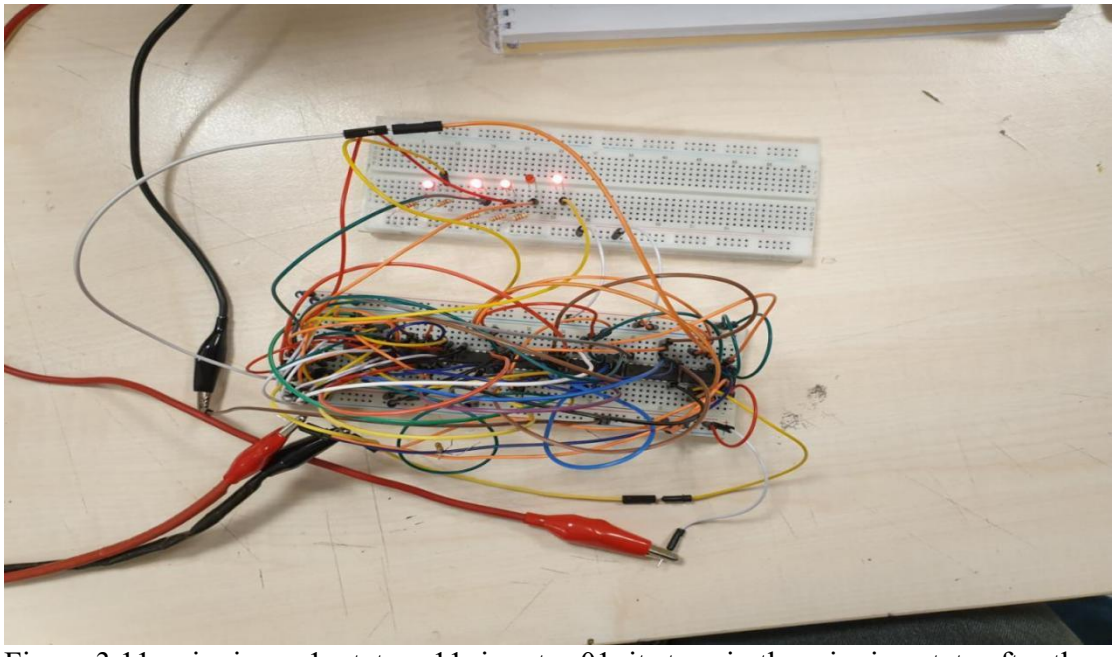

Figure 3.11: winning = 1, state = 11, input = 01, it stays in the winning state after the game is over to reset circuit D-flip flop’s asynchronous reset pins should be given with low voltage.

3) Conclusion

In conclusion we learned how to design finite state machines using basic digital circuit components. We also learned how to implement FSM designs on the breadboard only using integrated circuits.