Mustafa Bahaeddin Bozan

22102838

EEE-102-001

# LAB7: Finite State Machine (FSM)

## A) Introduction

The purpose of the 7th lab session is to design a Finite State Machine (FSM) and implement the design on the breadboard by using logic gates and D-Flip-Flops.

## B) Methodology of Fatih Sultan Machine:

In this lab session one can build an FSM that mimics a conquest game, and it will be called Fatih Sultan Machine abbreviated as FSM too. First of all, one needs to determine the states of the design which will change according to the present state and the input. I recommend 4 states namely, idle, army preparation, war and conquer states. The state diagram can be seen in the Figure 1.

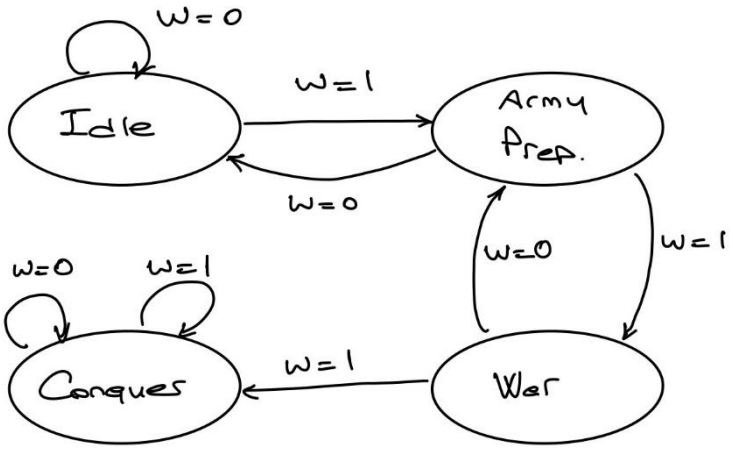


Figure 1. State diagram of Fatih Sultan Machine (w is the input).

After determining the states and the relationships between the inputs and the next states, one should put this information on the state table. The state table can be checked from the Table 1.

| Present State | | State Name | Next State | | | | C |
|---|---|---|---|---|---|---|---|
| $Q_1$ | $Q_0$ | | W=0 | | W=1 | | |
| | | | $Q_1$ | $Q_0$ | $Q_1$ | $Q_0$ | |
| 0 | 0 | Idle | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | Army prep. | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | War | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | Conquer | 1 | 1 | 1 | 1 | 1 |

Table 1. State table of the FSM design.

The next step should be determining the equations of C which is the output and $D_0$ and $D_1$ which are the inputs of the D-Flip-Flops. Since C is easy to observe, one does not need to use K-Map method but to obtain the logic expressions for $D_0$ and $D_1$, one needs to use K-Maps.

$$C = Q_1 * Q_0$$

The equations for the inputs can be seen from the Figure 2 and Figure 3.

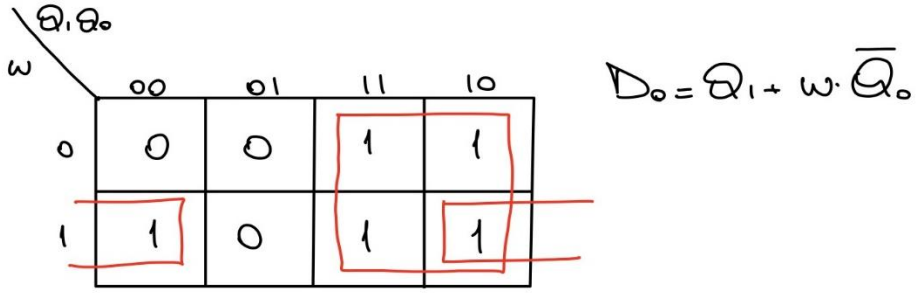


Figure 2. K-Map and the logic expression for $D_0$.

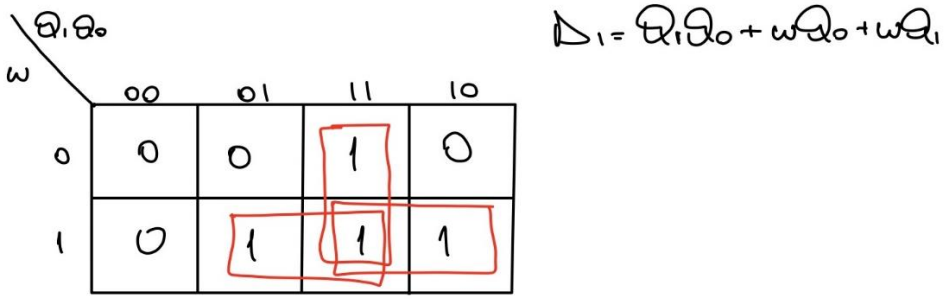


Figure 3. K-Map and the logic expression for $D_1$.

After finding the logic expressions for the inputs of the D-Flip-Flops and C, the design can be drawn as in the Figure 4.

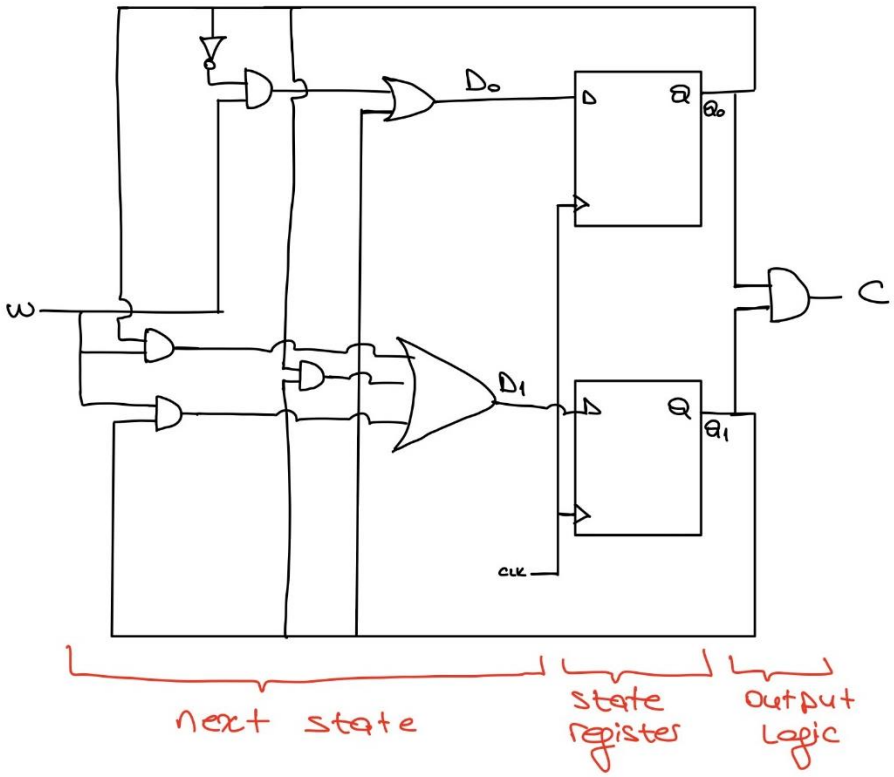


Figure 4. Scratch of the FSM design.

Then, one can move on to implement the design one the breadboard after checking the integrated circuit pin configurations of the AND (code: SN74HC08) OR (code: SN74HC32) NOT (code: SN74HC04) and D-Flip-Flops (code: SN74HC74). One should connect the VCC and GND connections to high and low respectively. Also, there is one critical point which is connecting the pre-set and clear connections to high since they are not used, and they are active low. Then, the design is ready to be controlled by observing the LEDs.

## C) Results of Fatih Sultan Machine:

After completing the steps explained in the Methodology part, the breadboard looks like to the Figure 1.

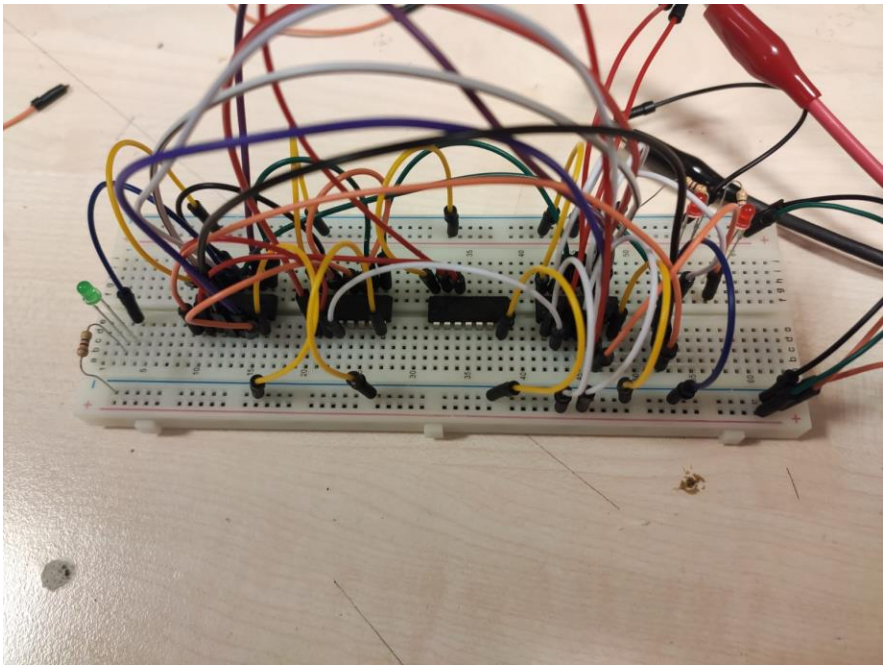

Figure 1. The final form of the breadboard.

Since I could not manage giving the input through button design, I tested my design manually. The results can be seen from the Figure 2-6.

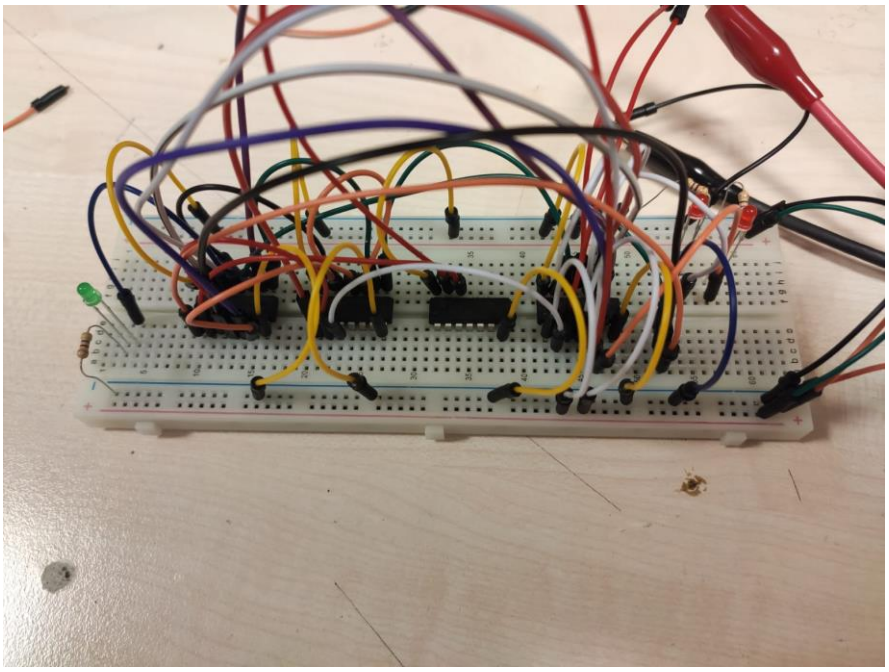

Figure 2. Present State = 00 and W = 0, Next State = 00.

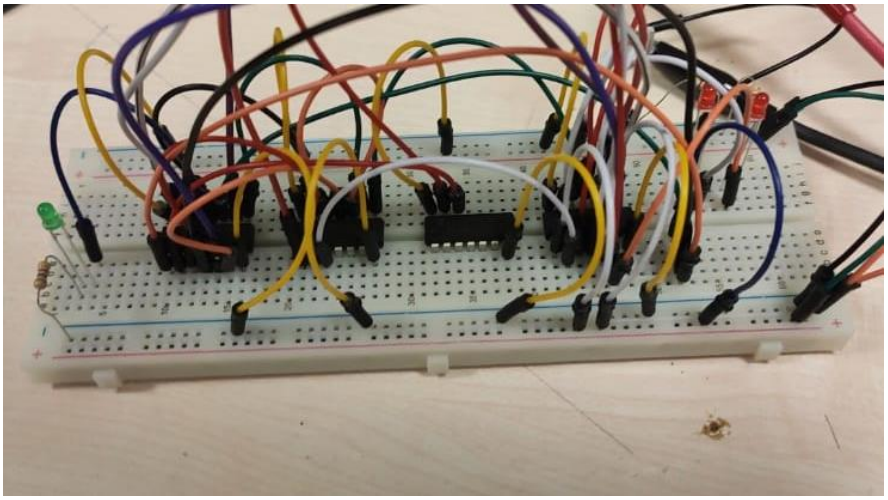

Figure 3. Present State = 00 and W = 1, Next State = 01, C = 0.

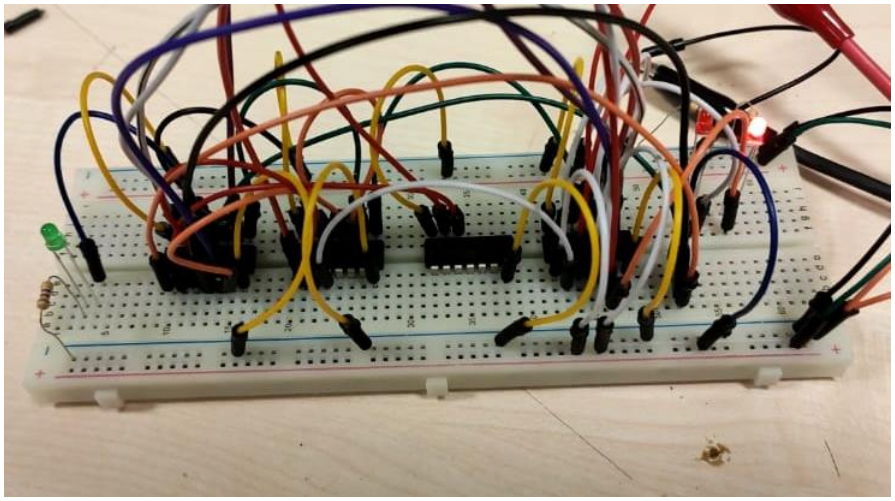

Figure 4. Present State = 01 and W = 1, Next State = 10 C = 0.

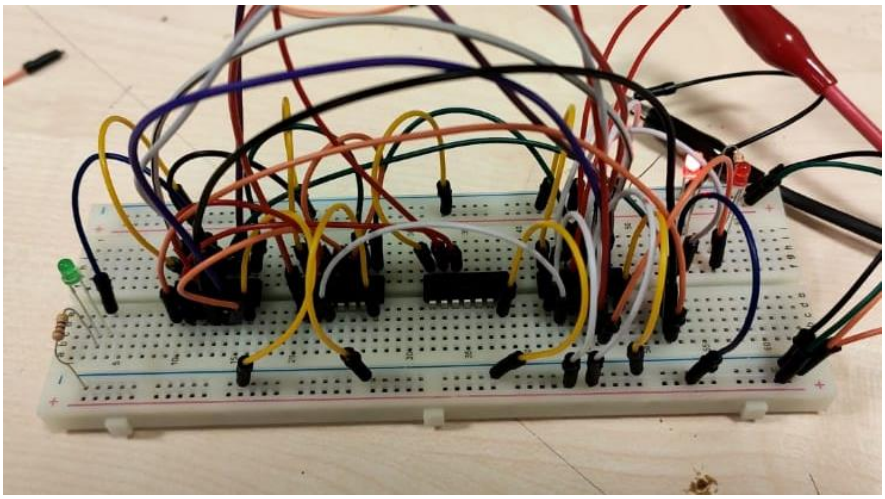

Figure 5. Present State = 10 and W = 1, Next State = 11 C = 0.

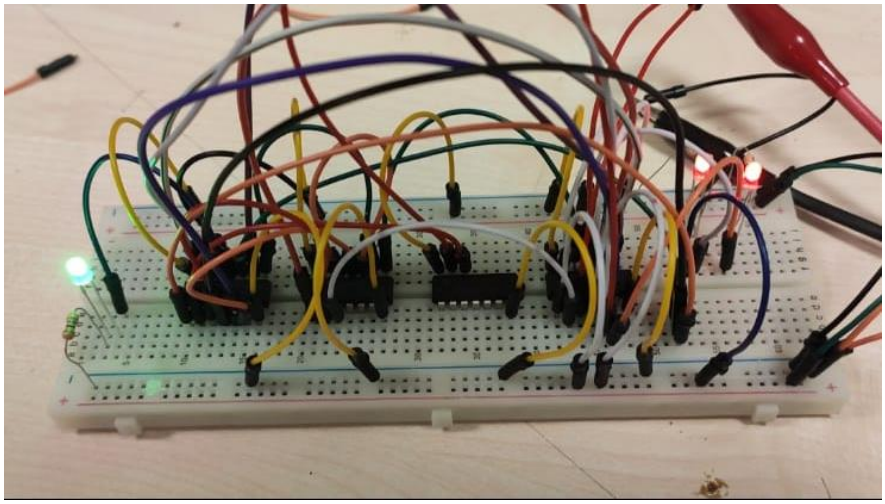

Figure 6. Present State = 11 and W = 1, Next State = 11 C = 1.

Also, I checked other state transitions as well. They are all working as predicted. However, since it will be hard to observe the results of those transitions from the photographs, they are not photographed.

## D) Conclusion

The purpose of the 7th and the last lab session is to implement an FSM design onto the breadboard by using logic gates and D-Flip-Flops. The chosen design mimics a conquest game and according to the logic expression of the output, the design was an example of a Moore Machine since the given input does not determines the output directly and determines only the states.

The design can be improved by adding other states about the conqueror city; however, since it will cost more gates and D-Flip-Flops, 4 states was decided to be enough to exemplify how a Moore type FMS works.

All in all, this lab session was very helpful and illuminating about a completely different concept called FSM. The design part was educative about how an FSM can be built.