Ege Ozan Özyedek
Lab07
EEE102-02
Lab Report

- Design Methodology

This week, we were assigned to design and implement a Finite State Machine. The design process started with the purpose of the FSM. I decided that I would design it based on a boss fight from a video game. The boss has 100 health points in the first state (HP100) and the player has two attack options. The first one is light attack (LA) which decreases the opponent's health points by 25. The other one is strong attack (SA) which decreases the opponent's health points by 50. The player wins (W, the output) if the enemies health point is 0. To implement this FSM, the first step was the design procedure which started with the state diagram.

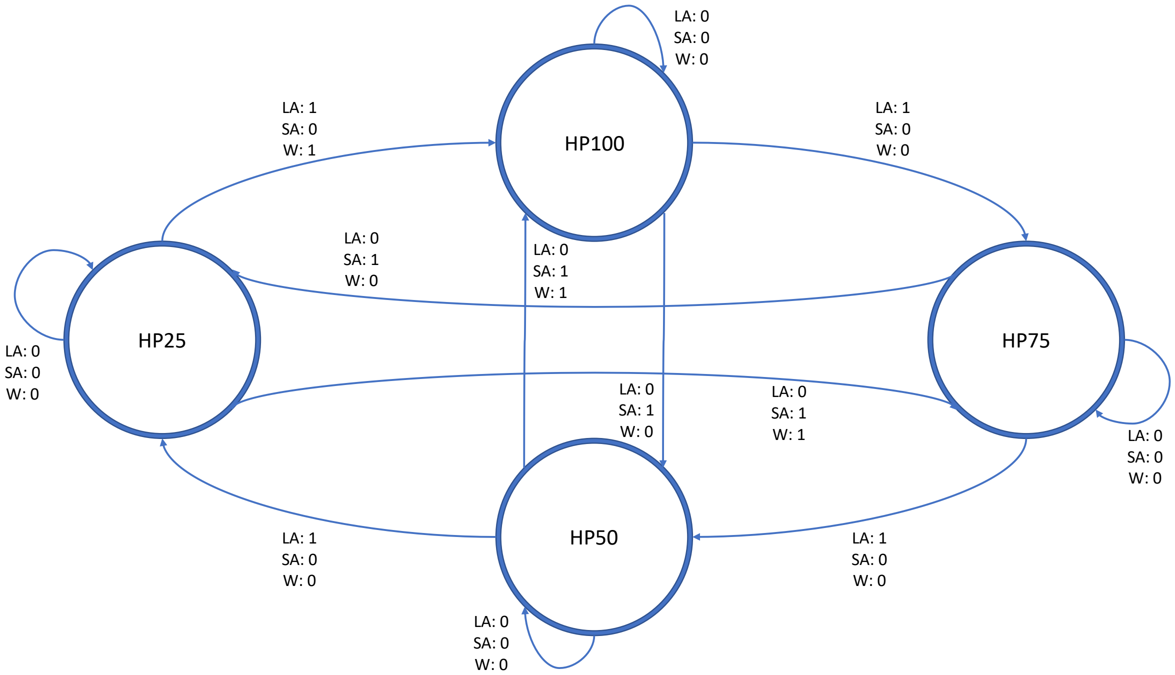

Figure 1: The State Diagram of the desired FSM

The state diagram and the input/output relations can be viewed on the above diagram. After the diagram was drawn, the next step was to find the next state logic part of the FSM. This required me to draw the Karnaugh Map of output (W) and the inputs of the D flip flops. These Karnaugh Maps can be seen below.

Table 1: Karnaugh Map of D0

Q/LA,SA	00	01	11	10
00	0	0	d	1
01	1	1	d	0
11	1	1	d	0
10	0	0	d	1

Table 2: Karnaugh Map of D1

Q/LA,SA	00	01	11	10
00	0	1	d	0
01	0	1	d	1
11	1	0	d	0
10	1	0	d	1

Table 3: Karnaugh Map of W

Q/LA,SA	00	01	11	10
00	0	0	d	0
01	0	0	d	0
11	0	1	d	1
10	0	1	d	0

By using these maps, we can find the formulas for the next state logic part of the FSM. These formulas can be found below.

$$W = Q_1 \cdot S + Q_1 \cdot Q_0 \cdot L$$

$$D_1 = \overline{Q_1} \cdot S + Q_1 \cdot \bar{S} \cdot \bar{L} + L \cdot Q_1 \cdot \overline{Q_0} + L \cdot \overline{Q_1} \cdot Q_0$$

$$D_0 = Q_0 \cdot \bar{L} + \bar{S} \cdot \overline{Q_0} \cdot L$$

Now that we found the next state logic formulas, we can draw and see which gates we will have to use to create the FSM. The schematic of the next state logic part with the flip flops and the output can be found below.

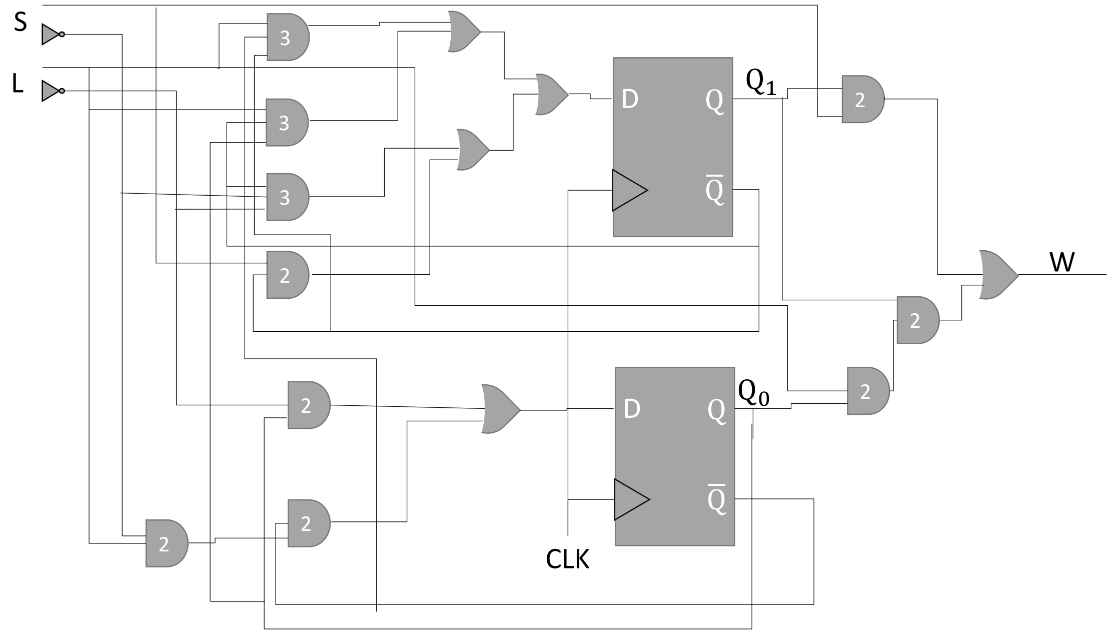

Figure 2: The Next State Logic, Flip-Flops and the Output of the FSM

Now that we know which gates are required, we can find the required integrated circuits that we need to use to create this FSM. For full efficiency, I created the circuit diagram above with the gates used in mind. For example, instead of using a 4-input OR-gate (which does not exist in the lab), I used three 2-input OR-gates. In some places, instead of using a 3-input AND-gate, I used two 2-input AND gates to use all gates of the used IC's. I decided to do this since I had one single breadboard in hand and didn't have much place to put any more IC's on it. I will now proceed to shortly explain all the gates used.

- 74LS/HC163 (x1) : The 4-bit counter which I utilize as my two inputs (LA and SA) and also the clock for the flip-flops. When the input clock of the counter has a period of 1 second; the first output (used as the clock) has a 2 second period, the second output (LA) has a 4 second period and the third output (SA) has an 8 second period. This means that I get the same inputs 2 times for the D Flip-flops.
- 74LS/HC74 (x1) : The D flip-flops. It contains two of them inside, which is enough for the FSM that I have created.
- 74LS/HC11 (x1) : The triple 3-input AND-gate. This is also enough for my design, the exact amount actually. I used this to create the data for the flip-flop inputs.
- 74LS/HC04 (x1) : The inverter, which I used to create the inverted inputs of LA and SA.
- 74LS/HC08 (x2) : The 2-input AND gates, one of these is used for the output and the other is used while creating the next state.
- 74LS/HC32 (x2) : The 2-input OR gates, similar to the 2-input AND gates these are also used for both the next state logic and the output.

After the determination of which IC are going to be used, it was time to design the circuit on the breadboard. The IC's with the corresponding connections can be found on the next page below.

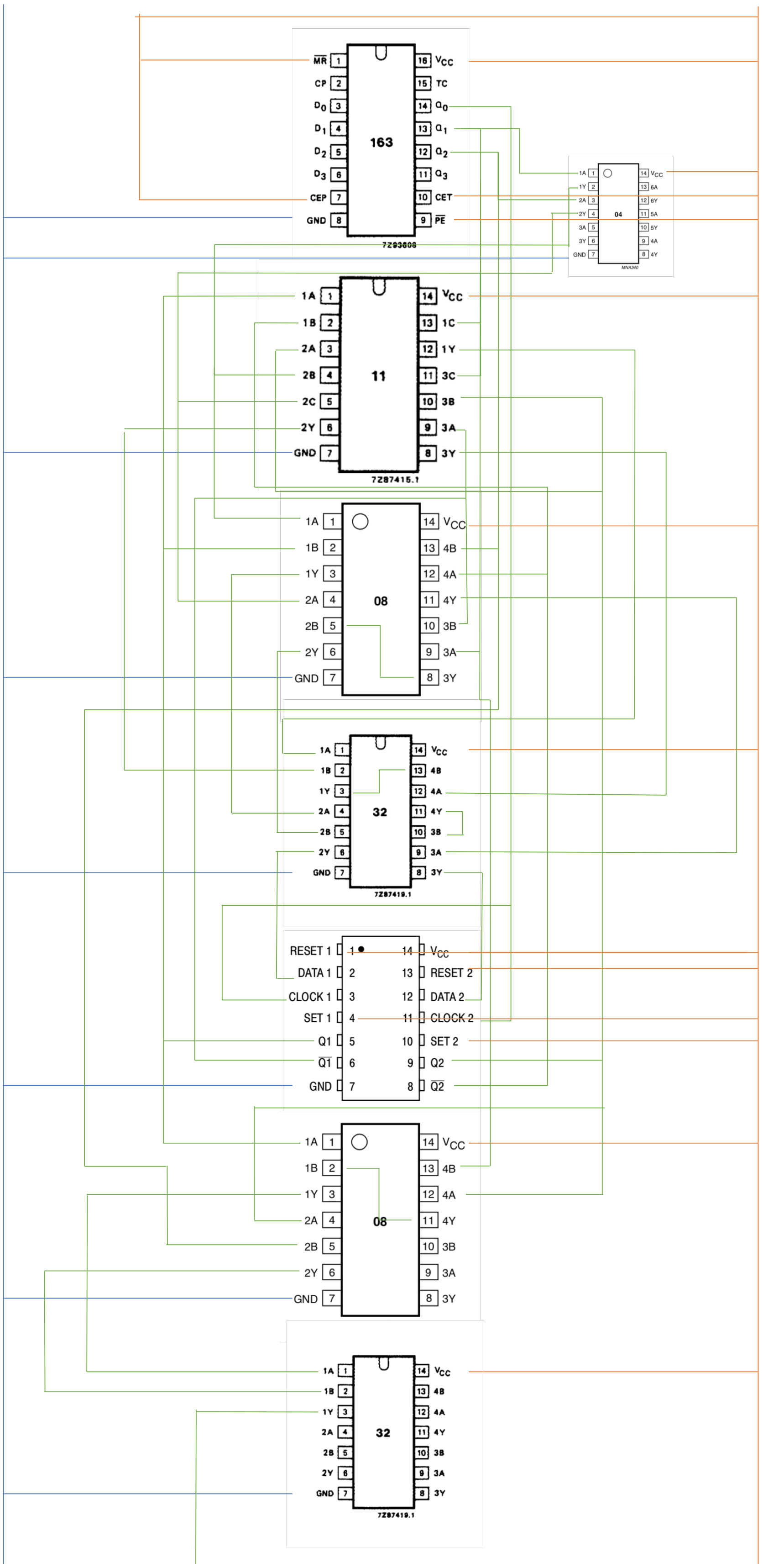

Figure 3:The Circuit Diagram with IC's and their corresponding connections

- Results

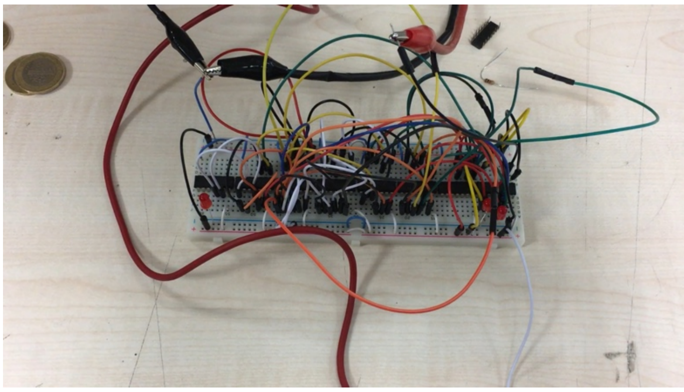

Figure 4: LA:0, SA:0, D:00, Q:00, W:0

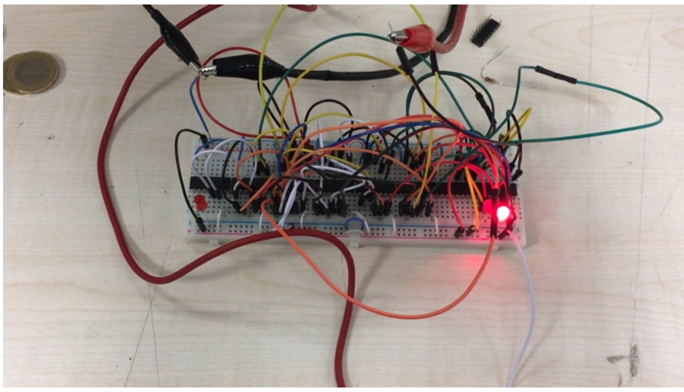

Figure 5: LA:1, SA:0, D:01, Q:00, W:0

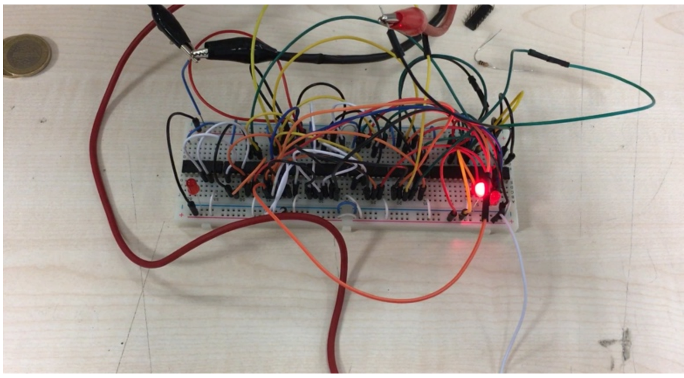

Figure 6: LA:0, SA:1, D:00, Q:10, W:0

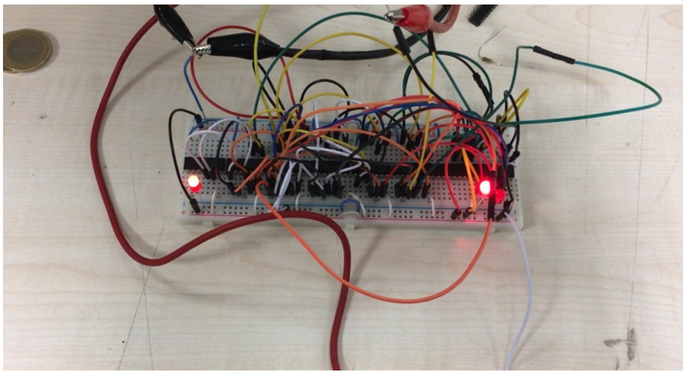

Figure 7: LA:0, SA:1, D:10, Q:00, W:1

Before getting into the results and their explanation, I have to note something so that the figures above are clearly understood. First off, since the clock has a 2 second period,

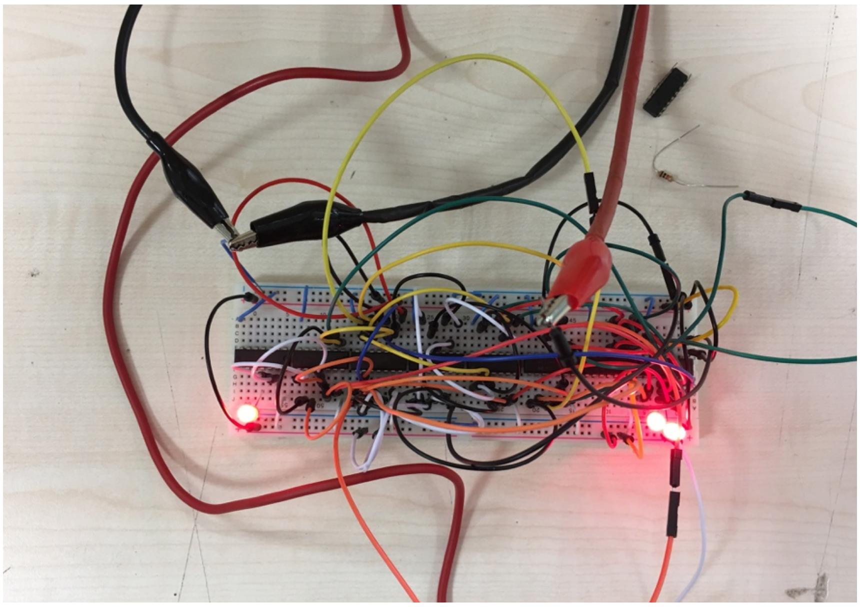

Figure 8: The implemented FSM circuit

it receives the same input 2 times since there exists 2 rising edges with the same inputs. It should also be noted that the output (W), is the output of the next state.

The results were as expected and thankfully the circuit was assembled correctly. The circuit was very complicated when built and was very hard to implement onto a single breadboard. The assembly process did have some problems which I will explain in detail on the next part, which is the conclusion.

- Conclusion

  The experiment has familiarized me with the FSM concept and the design process of it. It also familiarized me with certain IC’s which I believe will prove useful in the future of my academic life. There were some problems along the way unfortunately. The breadboard had some connectivity issues since some jumper cables were loose, this was very hard to solve since there doesn’t exist a way to find the solution this problem. There were also some issues with my signal generator which was solved when I changed it and set some other values such as its amplitude and offset to the correct values. All in all, this lab was pretty beneficial and I believe that I now know how to design and implement an FSM.