Lab 7: Finite State Machine

Mert Örnek EEE 102-3 22302052 03.05.2024

Purpose

The purpose of this lab session is to build an arbitrary finite state machine by first planning the states, transitions and outputs and showing them on tables, then implementing the finite state machine with circuit components and a breadboard.

Design Specifications

The arbitrarily chosen finite state machine in this design will consist of two states and a single input to keep the required components to a minimum due to a lack of space on the breadboard as well as lack of available logic gates in the lab which inhibits the possibility of a higher number of states and inputs. The system works on the premise of there being two states (Q=1 and Q=0), which stay the same when the button input is '0' but switch to the other state at the rising edge of the clock when the button input is '1'.

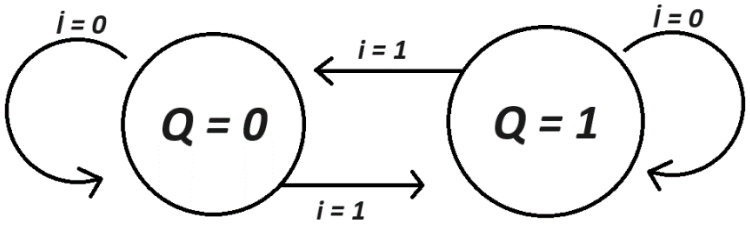

Figure 1:State Transition Diagram

Current State (Q)	Next State (D)		Output (LED)	
	i= 0	i = 1	i= 0	i = 1
0	0	1	0	0
1	1	0	1	1

Figure 2: Output Table

This Logic can be simplified as (D = i XOR Q) and (LED = Q)

Here, it is worth emphasizing again that the oversimplicity of the planned logic circuit was a lack of available components in the lab space. Finite state machines of greater complexity could have been achievable if there was a sufficient supply of common logic gates (AND, OR etc.) in the lab space or a large enough supply of functional LEDs to visualize said states.

The proposed finite state machine used the following logic gates: a single Quadruple 2-Input XOR gate (SN74HC86N), one Dual D-Flip Flop (SN74HC74N), one red LED representing Q, a number of 1.2 kOhm resistors, a button meant to act as the input and jumper wires to put the system together on the breadboard. The clear and preset pins of the D-Flip Flop were connected to the positive DC current strip of the breadboard due to them being set-up as active low. The button was also set up as active low by attaching a resistor to connect it to the positive strip of the breadboard and the other side of the button to the ground, with it having the preset logic value of '1' unless pressed, at which point it has an output of logic '0'. The output of the button was put though an XOR function with the Q of the D-FF by using the XOR gate, which in turn was connected to the D pin of the active side of the D-FF. The clock signal was generated by a signal generator at an arbitrary but visible frequency. The LED was connected to the strip that contains the Q, supported by yet another resistor. Of course, in order for the XOR gate and DFF to function, the Vcc and GND pins of both gates were connected to the potential difference generated by the DC Power Supply.

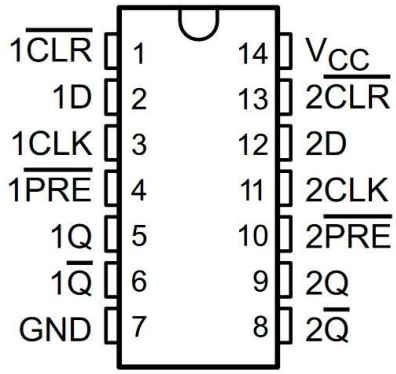

Figure 3.1: Pin layout of D-FF (Ja-Bots.com)

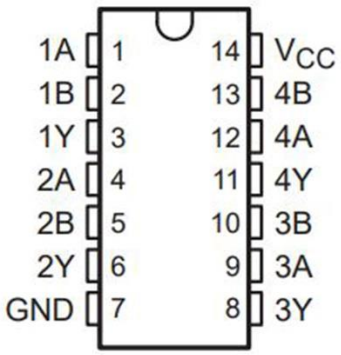

Figure 3.2: Pin layout of XOR gate (5 X SN74HC86N)

The resulting system consists of an oscillating LED output that stops at either '0' or '1' when the button is triggered. This system -in function- is comparable to real life equivalents such as a primitive form of game of chance which can be though of as a binary dice, an automatic door that unlocks or shuts down when triggered or the snooze function of an alarm clock which halts its oscillatory beeping as long as it is pressed.

Methodology

The methodology of this lab session consists of implementing the setup described in the design specifications through state transition diagrams and next state/ output tables by using the predetermined components that are available in the lab in conjunction with jumper wires and the breadboard to implement the desired circuit.

Results

Due to the limited number of inputs and states used in this setup, there are three distinct possibilities: The button is not pressed, and the LED oscillates between logic ‘0’ and ‘1’, the button is pressed, and the LED is logic ‘1’, or the button is pressed, and the LED is logic '0’.

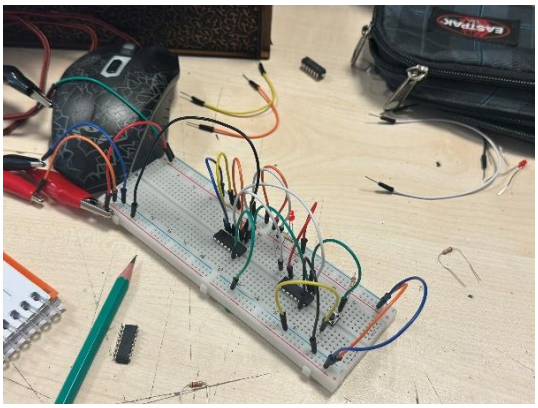

Figure 4.1: Button is logic ‘1’ and the LED is oscillating.

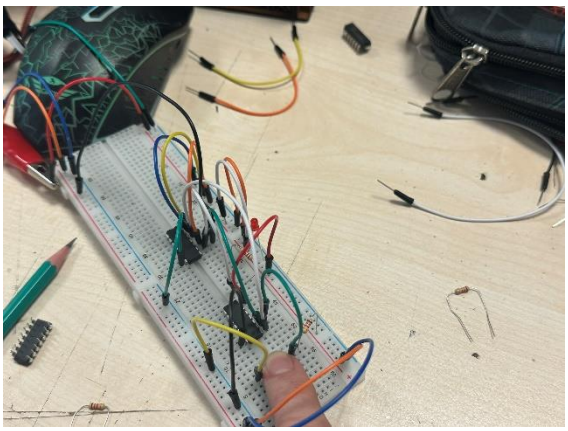

Figure 4.2: Button is logic ‘0’, LED is constant at ‘0’

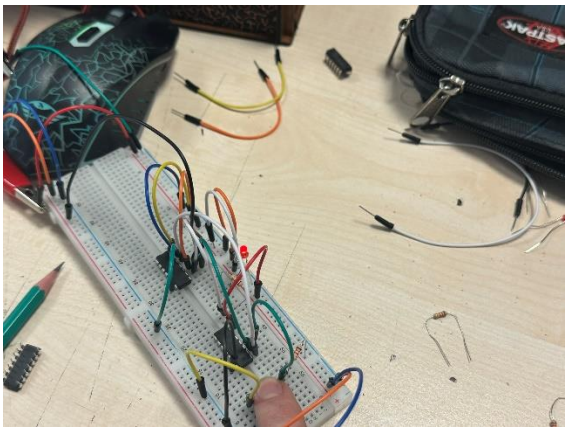

Figure 4.3: Button is logic ‘0’, LED is constant at ‘1’

These input/output combinations match the setup described in the design specifications section. On these grounds, it can be concluded that the implemented system is correct and works as intended.

Conclusion

In conclusion, the purpose of this experiment was to implement an arbitrary finite state machine we planned by using state transition and output/next state tables by using a breadboard, jumper wires and physical circuit components to show that our setup works as intended. Throughout this specific lab session, all these criteria mentioned in the lab instructions were met. One of the biggest issues faced in this lab session was the lack of available logic gates, LEDs and other circuit components in the lab environment. This is also the primary reason for the setup of this experiment being as uncomplicated as it is, as I had to make-do with logic gates that were generally “less desirable” or “less preferable”, such as the XOR gate used in this experiment. Outside of this

single significant issue, the lab session went without a hitch. The lab session also constituted a good exercise opportunity for developing physical logic circuit-building skills alongside theoretical ones.

Works Cited

Ja-Bots.com. “Flip Flop Tipo D SN74HC74N - Ja-Bots.” *Todo Lo Que Necesites En Robótica De Competencia*, 6 Mar. 2023, ja-bots.com/producto/flip-flop-tipo-d-sn74hc74n.

5 X SN74HC86N DIP14 Quadruple 2 Input Exclusive OR Gates Through Hole | All Top Notch. alltopnotch.co.uk/product/5-x-sn74hc86n-dip14-quadruple-2-input-exclusive-or-gates-through-hole.