Lab 1: Sequential Circuits and FSM

Introduction:

The aim of this lab was to design a finite state machine capable of detecting cars entering and exiting a car park with a total capacity of 15. Once this was achieved, the following step was to target the PYNQ-Z2 FPGA board and modify the design of the modules to use the buttons and LEDs featured on the board.

Method & discussion:

Given that the detection of the entry and exit of cars into the car park was obtained using two sensors, ‘a’ & ‘b’, the sequences that indicated the states were as follows:

- Default state (both a & b unblocked) input = “00”.
- Sensor a is blocked, input = “10”.
- Both sensors are blocked, input = “11”.
- Sensor a is unblocked, b is blocked; input = “01”.
- Default state (both a & b unblocked) input = “00”.

Knowing these sequences, for a car to be detected as entering the carpark, all of these states must be iterated in the order provided above. To detect a car exiting the carpark, the states must be iterated through in reverse order.

With this information, a Mealy FSM was designed with a total of 7 states, this can be seen in *figure 1*. The reason that a mealy state was chosen for this problem was due to the fact that current input determines the output of the state. This differs from a Moore machine that merely depends on the state and not the current inputs.

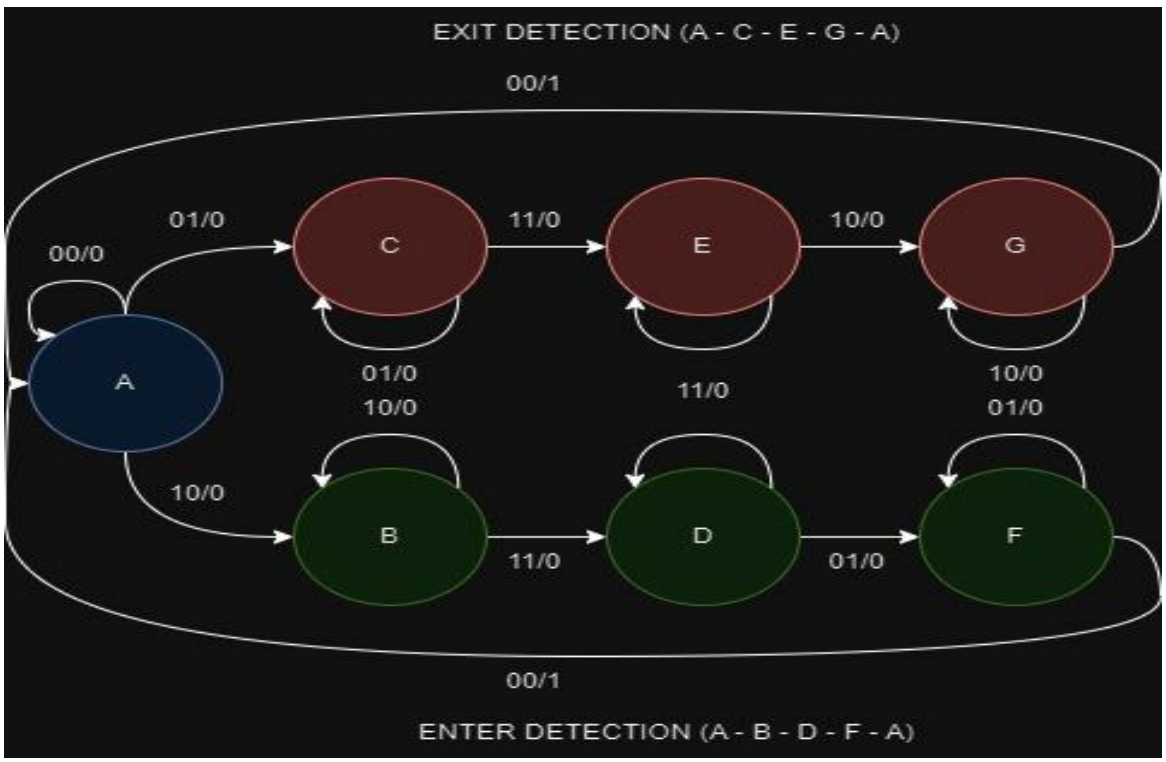

Figure 1: Mealy FSM for entry(green) and exit(red)

Following the design of this FSM, a counter module to detect the number of cars in the carpark was created. This took the entry & exit values from the FSM module, and depending on various factors, like the carpark being full, or completely empty, the counter would increase or decrease.

A testbench was developed and *figure 2*, is an example of one of the tests, where two cars enter the carpark, and then one exits.

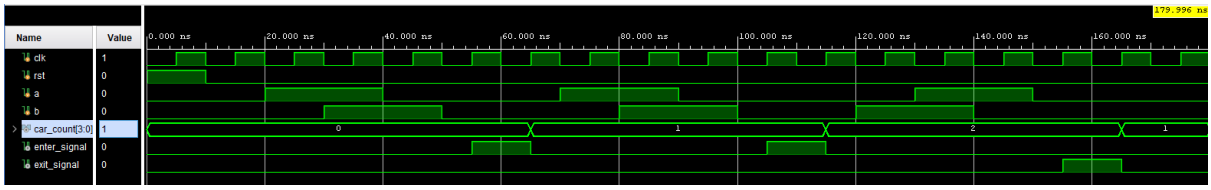

Figure 2: Basic example of cars entering and exiting carpark.

Given that the FSM was working as intended, the next step was to modify the top module. This is where all the rest of the modules were being called, but now it had to be modified to take inputs in from the buttons on the PYNQ board and display the count on the LEDs. A debouncer was integrated into the design to improve the accuracy of the buttons and the design worked as intended. *Figure 3* is a photo of the PYNQ board displaying that 3 cars were in the carpark.

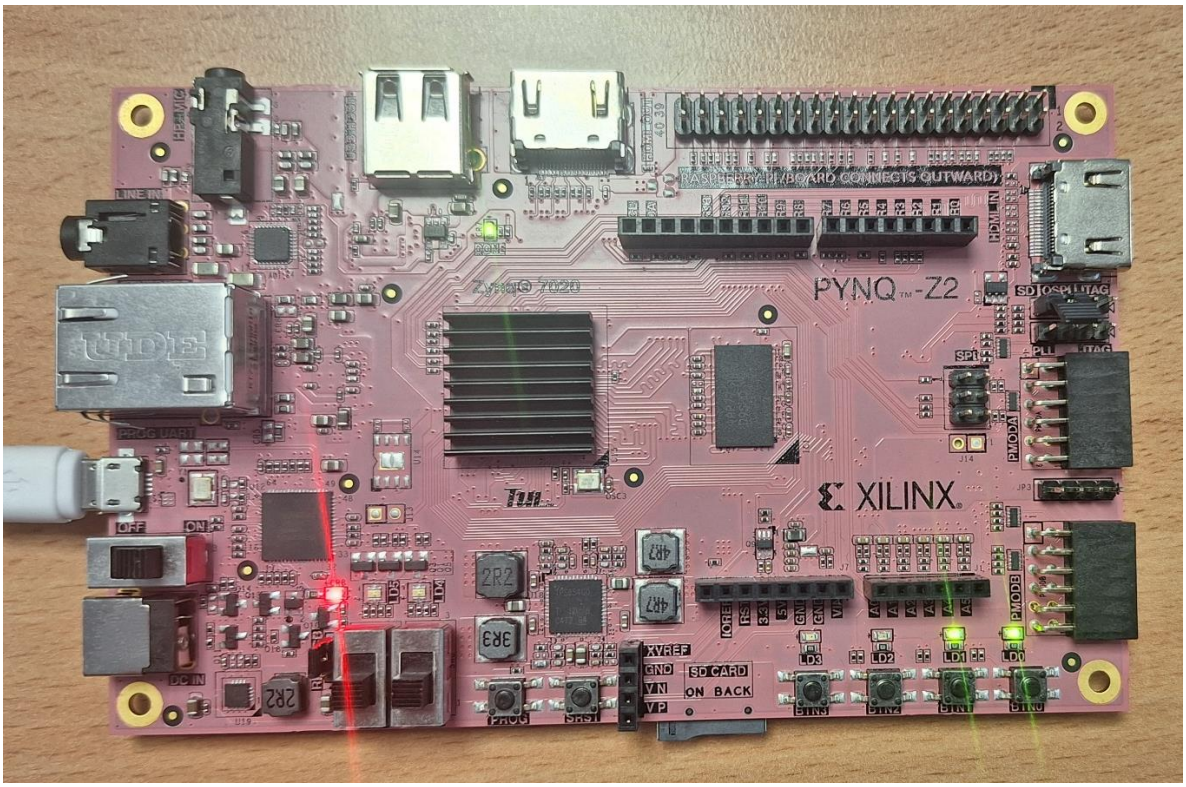

Figure 3: PYNQ-Z3 displaying carpark count after inputting values using buttons.

Conclusion:

The outcome of the lab was as intended, a FSM capable of detecting the entry and exit of car into a carpark was achieved. In addition, the design had the capability to detect if a car were to reverse in the case that it didn't fully enter the carpark.

Attached with this report, Pkavanagh_lab1.xpr.zip can be found, where all files sought can be found.