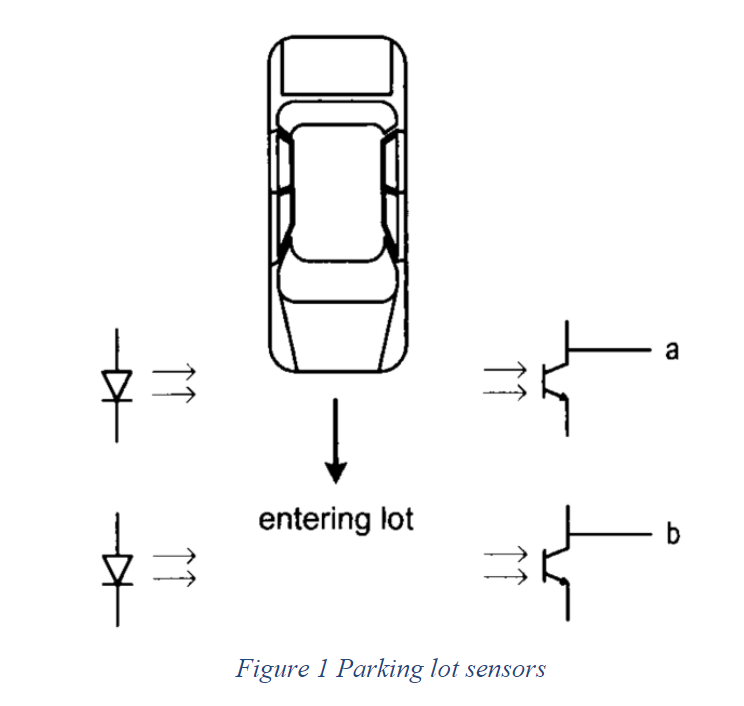
Abstract

The lab is a refresher to the finite state machines that we learned in EE 271 by designing a parking lot occupancy counter.

Introduction

Two pairs of photo sensors are used to monitor the activity of cars, as shown in Figure 1. When an object is between the photo transmitter and the photo receiver, the light is blocked, and corresponding output is asserted to 1. B monitoring the events of two sensors, we can determine whether a car is entering or exiting, or a pedestrian is passing through. For example, the following sequence indicates that a car enters the lot:

* Initially, both sensors are unblocked (i.e., the a and b signals are "00").
* Sensor a is blocked (i.e., the a and b signals are “10").
* Both sensors are blocked (i.e., the a and b signals are "11").
* Sensor a is unblocked (i.e., the a and b signals are "01").
* Both sensors become unblocked (i.e., the a and b signals are "00")



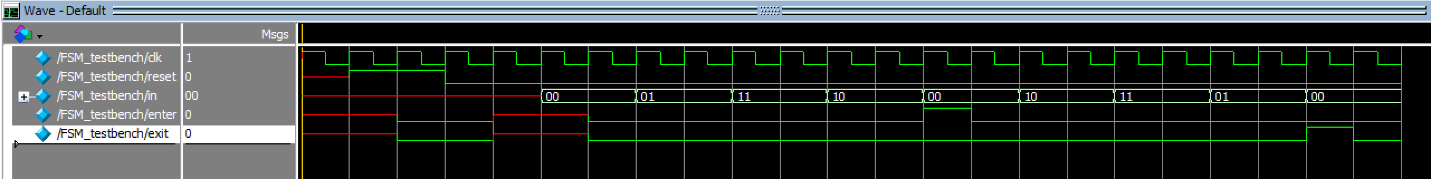
Procedures

1. Designed an FSM with two input signals, a (SW[0]) and b (SW[1]), and two output signals, enter and exit. The enter and exit signals assert true for one clock cycle when a car or exits the lot, respectively. (I assumed that cars will not change direction while entering or exiting the parking lot.)
2. Designed a counter with two control signals, enter and exit, (from the FSM) which increments and decrements the counter when asserted. The maximum capacity of the parking lot is 25 spots.
3. Combined the counter and the FSM and model the parking lot, using push buttons to mimic the two sensor outputs and the seven-segment displays to display the car count. My System have the following features:
4. Displays the car counts as they enter the parking lot on the seven-segment displays HEX0 and HEX1.
5. If the counter reaches 25, displays the word “FULL” on HEX5-HEX2.
6. As cars exit the lot, the counter decrements and the corresponding number is displayed on HEX0 and HEX1.
7. When the lot is empty. Displays the word “EMPTY” on HEX5-HEX1 and displays the number ‘0’ on HEX0.
8. Uses 2 off-board LEDs to represent the a and b signals. When a is 1, turns on a red LED, and when a is 0, turns off a red LED. Stimulatingly, when b is 1, turns on a green LED, and when b is 0, turns off a green LED.

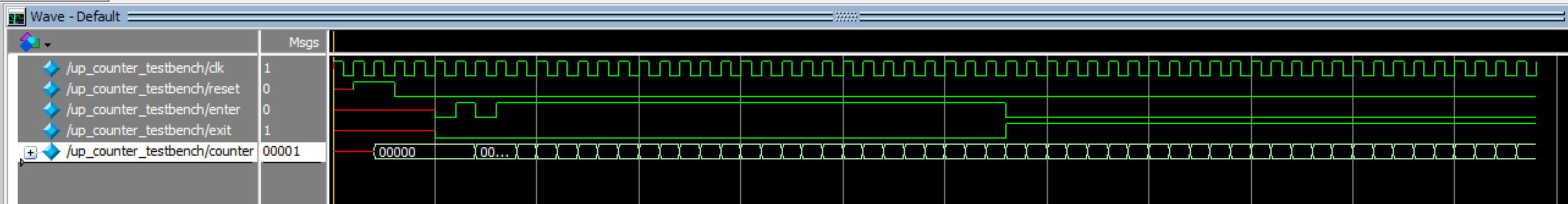
Results

The car enters from the right. So, SW[0] is a and SW[1] is b. Also, the reset switch is SW[9].

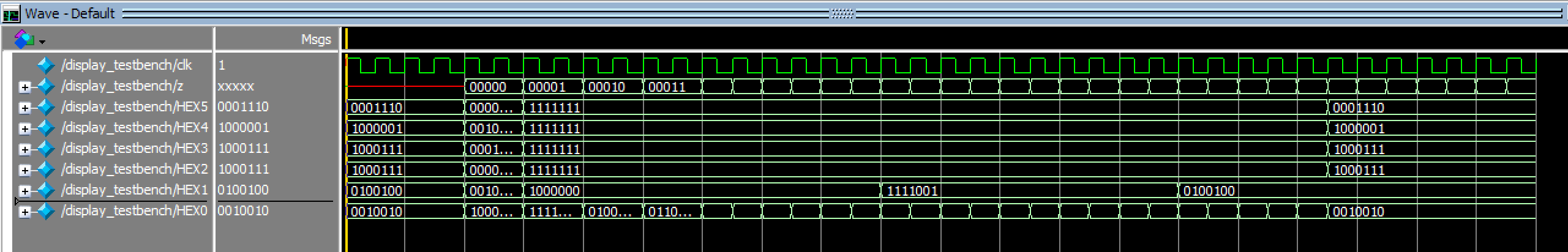
1. FSM counter waveform:



1. Up\_couter waveform:

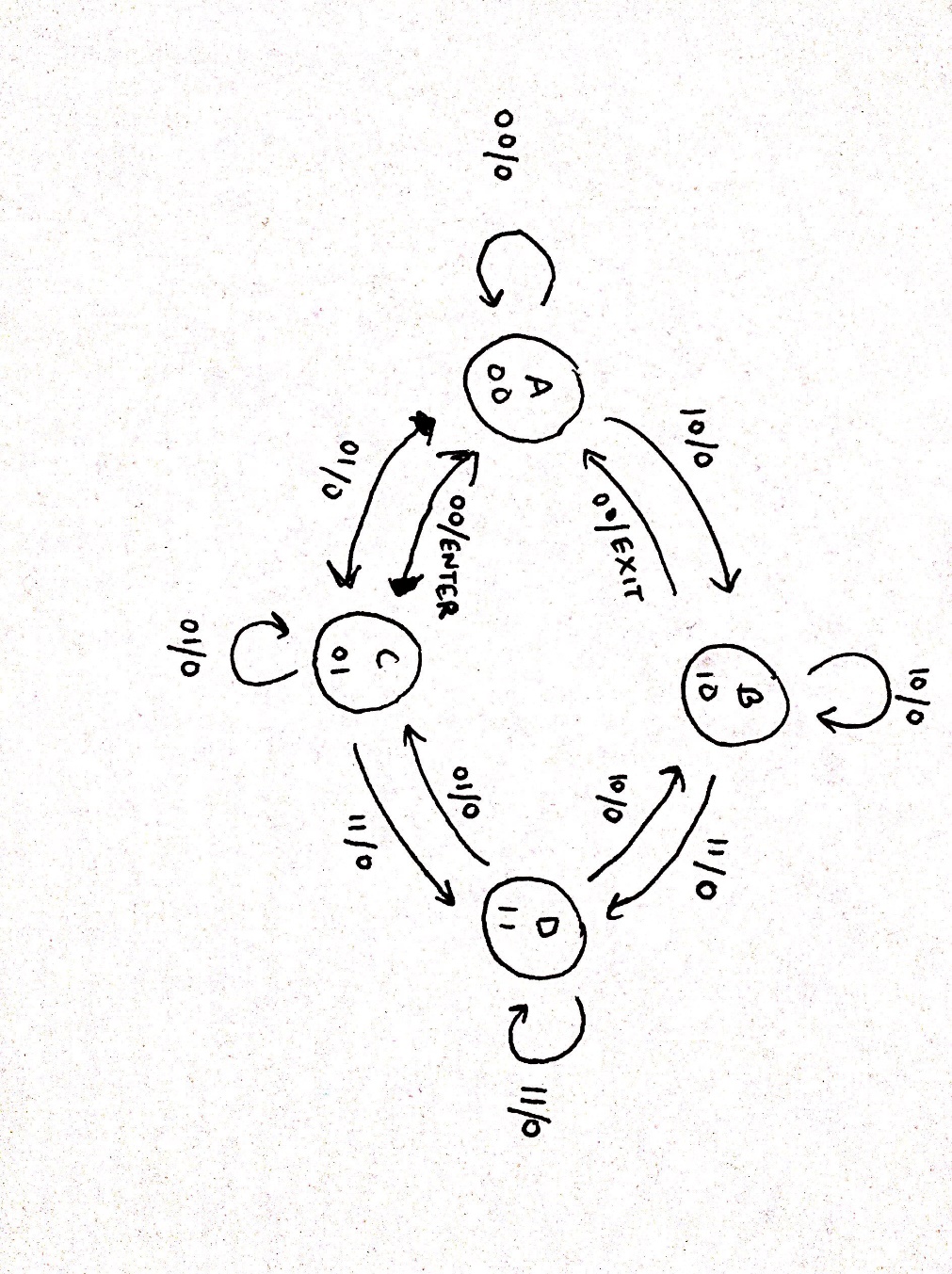


1. Display waveform:

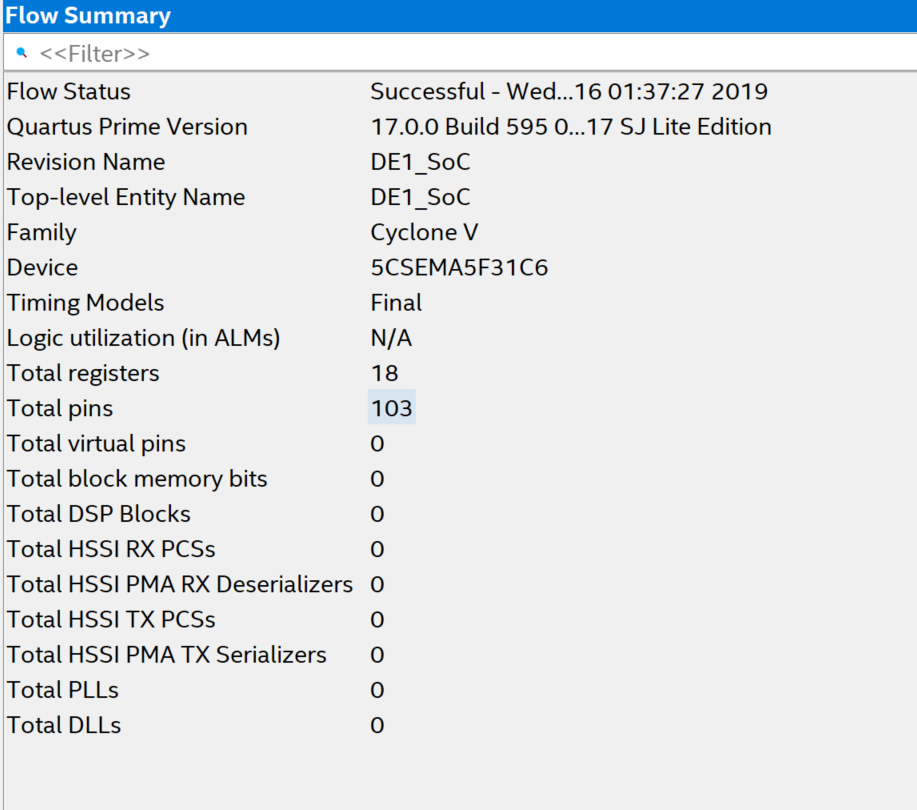


All the outputs were expected. All the pdfs of the SystemVerilog code is attached with this report.

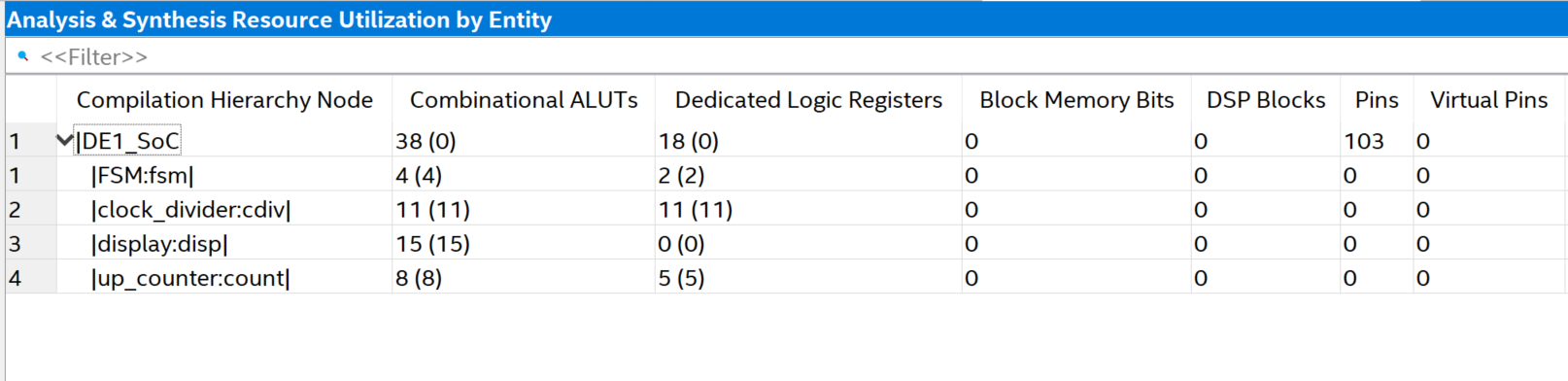
1. State Diagram for FSM:



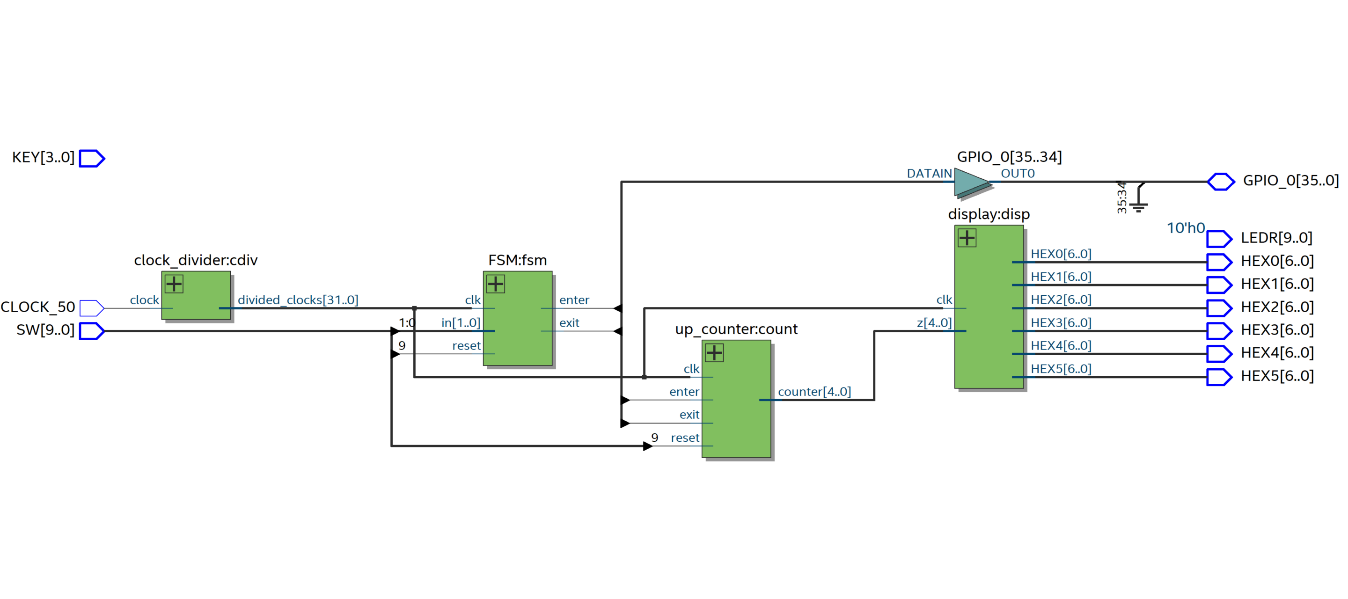
1. Flow Summary:



1. Resourse Utilization by Entity:



1. Block Diagrams of DE1\_SoC:



Analysis

This lab covered most of the topics from EE 271 which were state machines, GPIO\_0, Combinational and sequential logic.

Conclusion

This lab took a total time of 15 hours including reading, planning, design, coding, debugging, testing etc.