

**EE/CSE 371:**  
**Design of Digital Circuits and Systems**

**Lab1: Parking Lot Occupancy Counter**

---

**Lab Objectives**

This lab provides a refresher of the finite state machines that you learned in EE 271 by designing a parking lot occupancy counter.

**Assigned Task**

**Parking Lot Occupancy Counter**

Consider a parking lot with a single entry and exit gate. Two pairs of photosensors are used to monitor the activity of cars, as shown in Figure 1. When an object is between the photo transmitter and the photoreceiver, the light is blocked, and the corresponding output is asserted to 1. By 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).

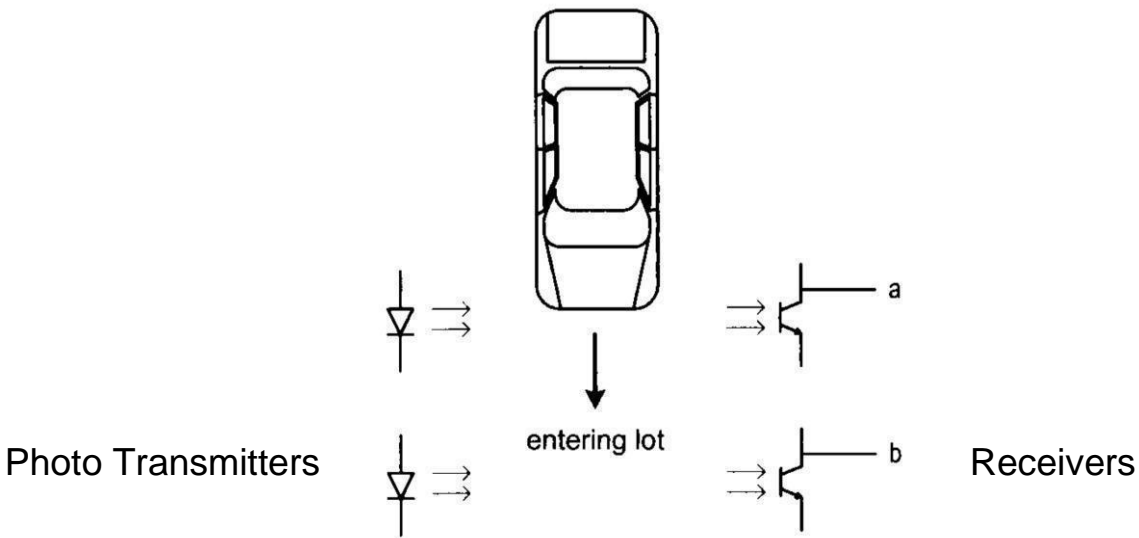


Figure 1. Parking lot sensors

**Design a parking lot occupancy counter as follows:**

1. Design an FSM with two input signals, *a* and *b*, and two output signals, *enter* and *exit*. The *enter* and *exit* signals assert true for one clock cycle when a car enters or exits the lot, respectively. Derive the SystemVerilog code for the FSM and simulate it in ModelSim. Do not assume that cars will not change direction while entering or exiting the parking lot.
2. Design a counter with two control signals, *inc* and *dec*, which increment and decrement the counter when asserted. Assume that the maximum capacity of the parking lot is 25 spots. however, you may test (and demo) your system with a maximum of 5 spots. Your system must work with both of these sizes. Derive the SystemVerilog code for the FSM and simulate it in ModelSim.
3. Combine the counter and the FSM and then model the parking lot, using two switches on the breadboard to mimic the two sensor outputs and the seven-segment displays to display the car count. Your system should have the following:
  - a. Display the car counts as they enter the parking lot on the seven-segment displays HEX0 and HEX1.
  - b. If the counter reaches 25 (or 5 for simulation and demo purposes), display the word "FULL" on HEX5-HEX2

- c. As cars exit the lot, the counter decrements and the corresponding number should be displayed on HEX0 and HEX1.
  - d. When the lot is empty. Display the word “CLEAR” on HEX5-HEX1 and display the number ‘0’ on HEX0.
  - e. Use 2 LEDs on the breadboard to represent the a and b signals. When a is 1, turn on the LED on the left, and when a is 0, turn off the LED on the left. Stimulatingly, when b is 1, turn on the LED on the right, and when b is 0, turn off the LED on the right.
  - f. Use the third switch on the breadboard to represent the reset signal.
4. Please read the “GPIO guide” for the instructions regarding how to wire and use switches and LEDs. The general rule is labeled as below.
    - a. Wire 2 LEDs to output ports (GPIO0\_0[26] and GPIO0\_0[27])
    - b. Wire 3 switches to any input ports (GPIO0\_0[5] – GPIO0\_0[22]) (Please pick any three you would like to use, and update your SystemVerilog code accordingly)

Breadboard - JP1

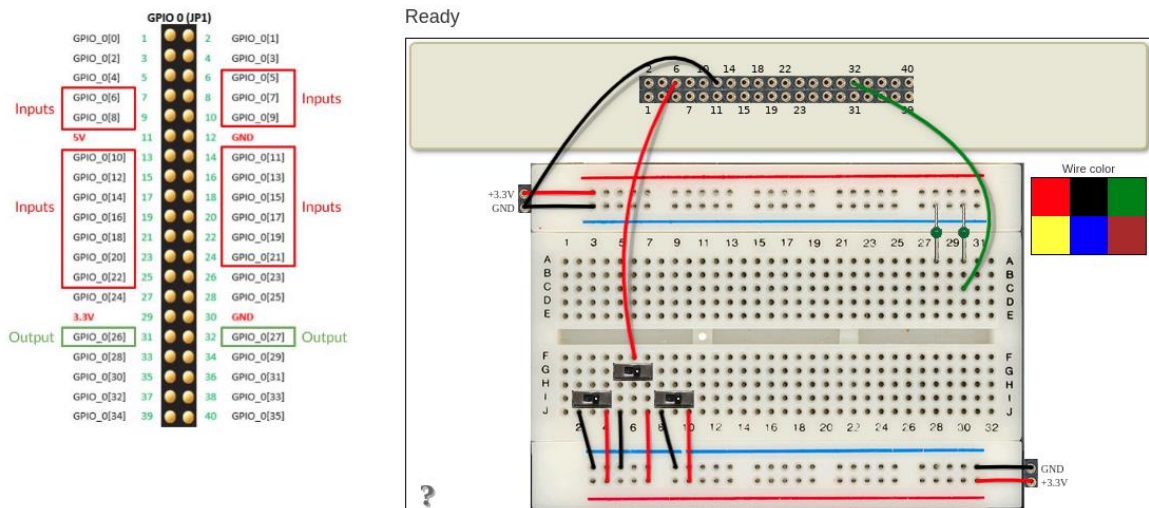


Figure 2. LabsLand GPIO headers

5. Simulate the system in ModelSim.
6. Upload the program onto LabsLand FPGA and record a 2-3 mins video to demonstrate your work. Note, we do not need to see your compilation part.
7. Create a block diagram for your system and include it in your lab report.
8. In the report, include the state diagram(s) that you used in designing this system

## Lab Demonstration and Submission Requirements

- Demonstrate the project on the remote lab's DE1\_SoC board in a video that shows the functionality of your project. You can use zoom to record the video or any other screencast programs. Upload your video to canvas. Please note that uploading videos may take time on canvas so plan accordingly and do not wait very close to the due date to avoid late submission.
- Write a Lab Report, as framed by the "Lab Report Outline" document on Canvas. Comment your code. Follow commenting guidelines as discussed in the "Commenting Code" document on Canvas. **A significant amount of grade will depend on the commenting style.**
- Submit your lab report as a pdf file and all of your SystemVerilog files on Canvas.