

ECE371 Design of Digital Circuits and Systems  
  
Winter 2019

**Lab 1 - Parking Lot Occupancy Counter**

Student: Jeff Josephsen  
Professor: Dr R. Hussein

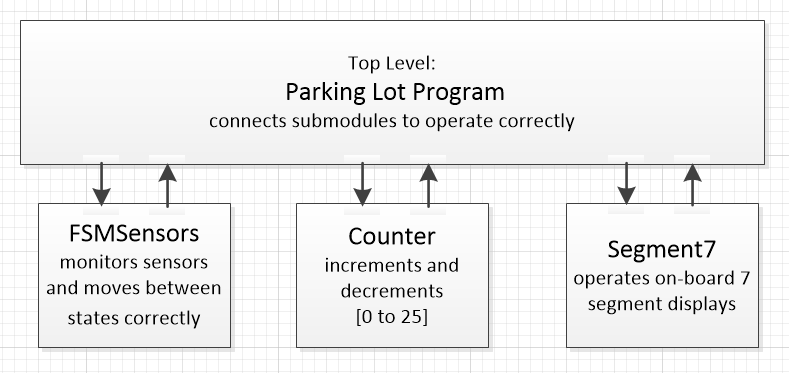
1-14-2019

**Abstract:**

In this lab, the student worked within the framework of the Quartus II and ModelSim software suites to program the Altera DE1-SoC Cyclone V Field Programmable Gate Array (FPGA).  
  
Finite State Machine (FSM) was implemented which simulated on the hardware a system which monitored the count of automobiles parked in a parking lot. It also displayed that count to a seven segment display. The system was dependent on the use of sensors for the monitoring of the entering or exiting of vehicles. The system showed when the parking lot had become full or empty. Note, no actual parking lots were harmed in the testing of this product.

**Introduction:**

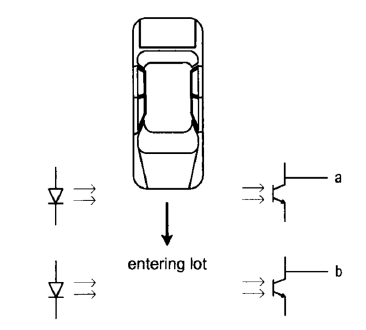
The objective of this lab was to design a system that would count and display the number of parked vehicles in a parking garage with the capacity of 25 parking spaces. This implementation was done by use of SystemVerilog HDL and FPGA hardware. The students were given the alignment and labels of sensors, basic structure of the modules, a few modules of code from which to base their approach.  
  
From that point, the students were asked to design the appropriate Finite State Machine, and write the corresponding SystemVerilog code. The output code was then simulated for each of the modules on ModelSim. The modules are shown in the following figure.



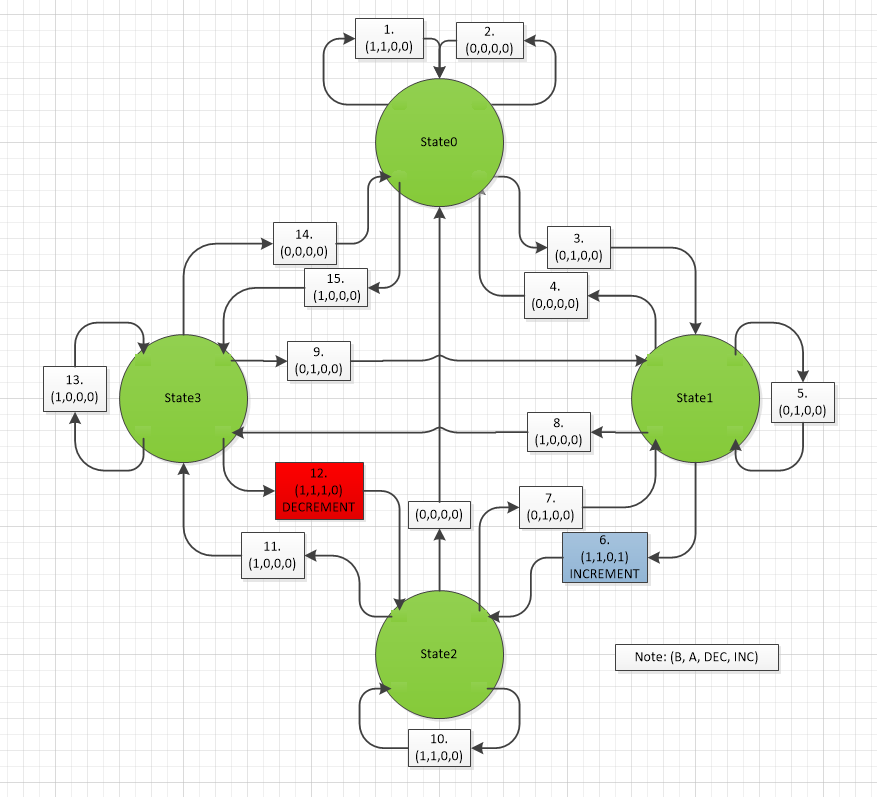
Next, the code was upload to the FPGA hardware. The program as implemented on a real FPGA with buttons, switches, and Seven Segment displays were then tested for correct operation.

**Procedure:**

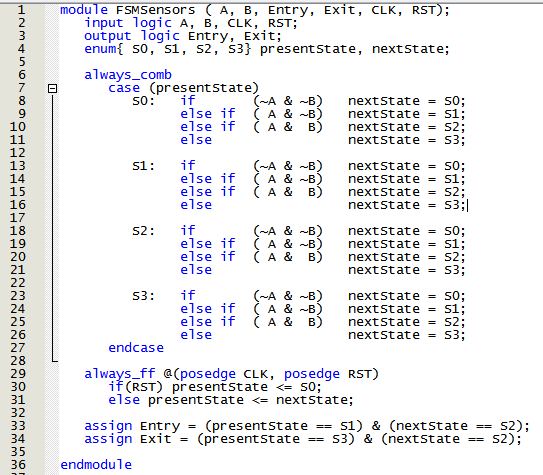
The students were asked to imagine a parking lot with sensors that would be used to monitor cars as they entered or left a parking garage. The students were given the arrangement of the sensors as in the image below.



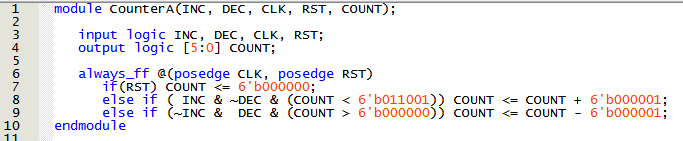
The student used a layout of the input sensor and output increment or decrement of count in parameters as following: (Sensor B, Sensor A, Decrement, Increment). These parameters were then laid out in a Finite State Machine as can be seen in the following image.



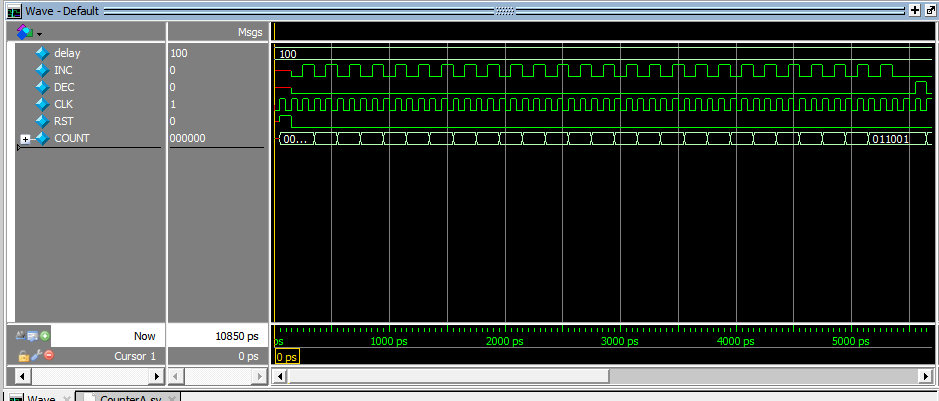
From this point a module was added in SystemVerilog by which the sensors would move the system to the correct state of the Finite State Machine. This code can be seen in the following image.



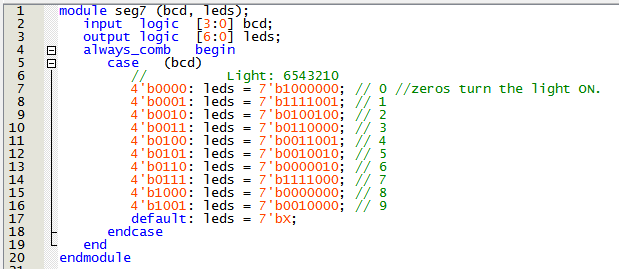
Another module was added which would increase or decrease the count of the automobiles in parking lot as follows in the image below.



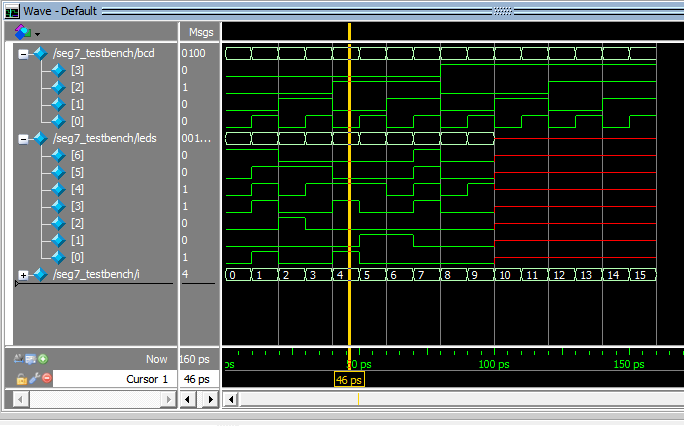
The testing of this module by a testbench module and simulation in ModelSim was completed. This testing showed the output of the module with all of the potential inputs. The output of this testing can be seem in the following screenshot image.



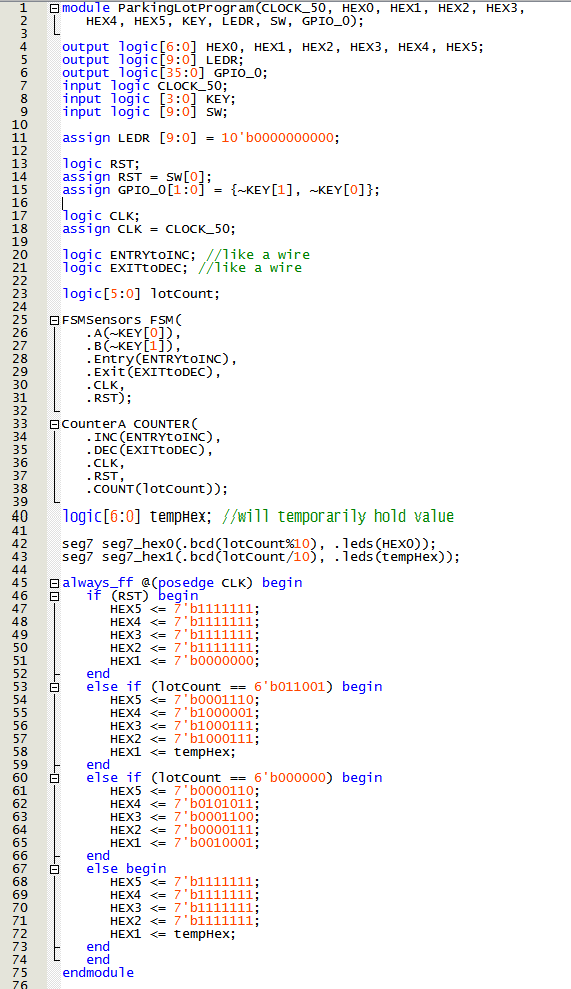
Next, a module to display the number of automobiles counted was added to the system. This module would also allow for the display of "Full25" or "Empty" in correspondence with the number of cars in the garage. This module can be seen in the following image.



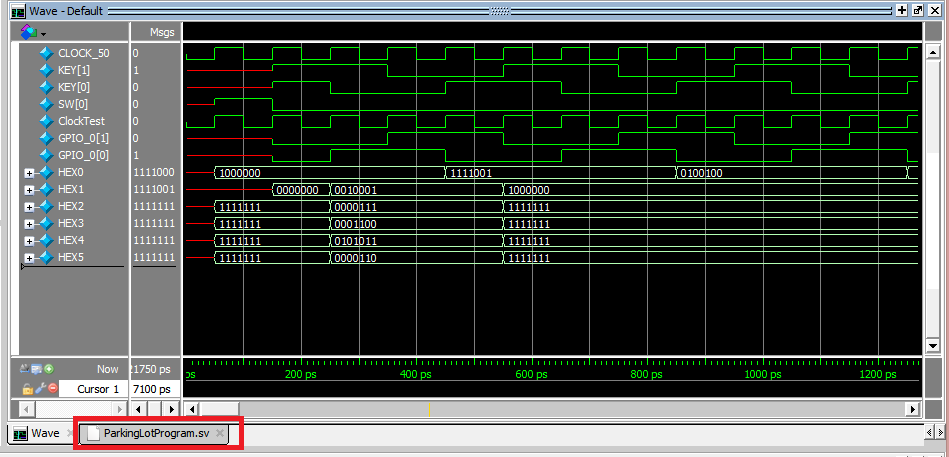
This module was tested and the results of the simulation can be seen in the following image.



Lastly, the modules were brought together under a top level Parking Lot Program module. In this modules the other modules are connected. The code can be seen in the following image.



This module was then tested in conjunction with the others by test bench and the results can be seen in the following image.



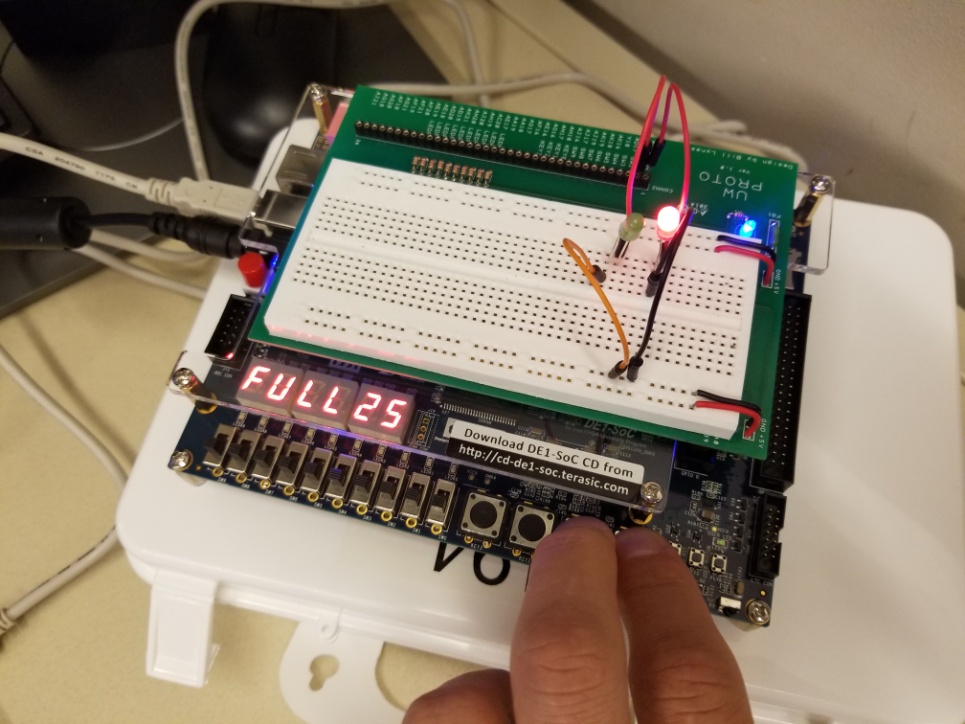
Lastly, the code was uploaded to the Altera DE1-SoC Cyclone V Field Programmable Gate Array (FPGA).  
  
The results of which can be found in the following results section of this report.

**Results:**

The completed system yielded a parking lot automobile counter that worked fairly well. The system counted up to 25 and then read-out "Full25", also, the system reset to "empty" upon complete countdown or clicking of the reset switch.

There was a bug that, at one point, jumped the count to 29. I'm not sure how that happened, and I could not replicate the problem a second time.

A photograph of the working system can be seen below.



The main challenges in this lab were getting the up and rolling with Quartus Prime Lite and ModelSim. There was a problem in having the Simulations continue running where a reinstantiation of objects via the Library Menu was required. Also, I had a bit of trouble uploading the to the Altera Board at the beginning for a few missing steps in the Programmer's Hardware set-up. I was able to elicit help from another student named Haomin who had taken the class the previous semester.

**Conclusion:**

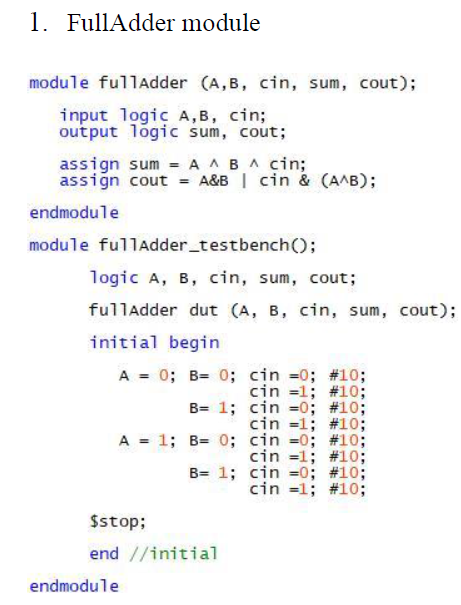
This Laboratory was useful in reviewing Finite State Machines and SystemVerilog Coding principals. The ModelSim simulations were very helpful in allowing the visualizations of the use and output of the code. I feel that this project was a great way to allow me, with my limited experience, to see how VHDL systems might be used in the real world with sensors, finite state machines, and seven segment displays.

**References:**

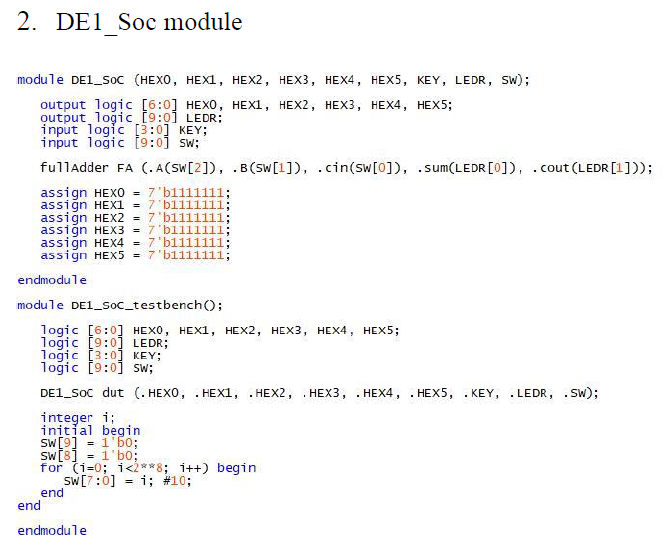
I would like to give credit to student Grant Wheatley, who was helpful in tutoring me in parts of this lab including differences between Verilog and System Verilog,

Also, the following modules were delivered to the students for a starting off point in the coding work:

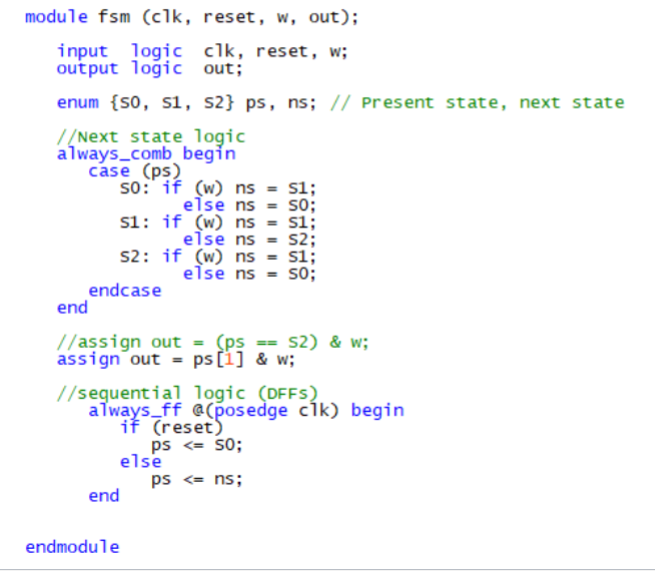
First, a full adder module, which was given in the lab instructions. It can be seen in the following image.

****

Next, a high level module was given in the lab instructions to help the students get the first of their HDL programs uploaded to the board. This code could be amended for use in the final top-level module of the system. It can be seen in the following image.

****

Next, the following string recognizer, which was given in the Classroom lectures, was useful as a version of the state follower used by the student.

****