SANTA CLARA UNVERSITY

Fatemeh Tehranipoor, James Lewis, and Tokunbo Ogunfunmi **ELEN/COEN 21L**

Laboratory #3: Two Level Circuit Design

For lab sections Monday-Friday Oct. 12 – Oct. 16, 2020

I. OBJECTIVES

- To design, test, and implement a digital logic circuit based on a functional specification.
- To translate a problem statement into an algebraic representation and simplify it using Kmaps to find the minimized logic implementation.
- To use 7-segment displays to show an output and add a Verilog module to generate the 7-segment display.

PROBLEM STATEMENT

In this laboratory assignment, you will design a highway entrance ramp metering controller with the following specifications for controlling the release of traffic from an entrance ramp onto a highway:

There are three metered entrance lanes: one carpool lane and two other lanes for cars that are not carpool. Each lane has its own light ("red" for stop and "green" for go) and its own sensor to detect a waiting car. The carpool lane is given priority for a green light over the other two lanes. Otherwise, a "round robin" scheme is used in which the green lights alternate between the left and right lanes.

The controller has these **four inputs**:

CS	car pool lane car sensor	All three sensors are a "1" when a car is present, and a "0" when no car is present.
LS	left lane car sensor	
RS	right lane car sensor	
RR	round robin signal	RR is "1" or "0" to select the left or right lane, respectively

The three controller **outputs** are:

CL	car pool light	All three lights are a "1" for green, and a "0" for red.
LL	left lane light	
RL	right lane light	

The controller operates as follows:

1. If there is a car in the car pool lane, CL is 1.
2. If there are no cars in the car pool lane and the right lane, and there is a car in the left lane, LL is 1.
3. If there are no cars in the carpool lane and in the left lane, and there is a car in the right lane, RL is 1.
4. If there is no car in the car pool lane and there are no cars in the left and right lanes, then RL is 1.
5. If there is no car in the car pool lane, there are cars in both the left and right lanes, and RR is 0, then LL = 1.
6. If there is no car in the car pool lane, there are cars in both the left and right lanes, and RR is 1, then RL is 1.

If any of CL, LL, or RL is not specified to be 1 in conditions 1 to 6 above, then it has value 0 for that condition.

II. PRE-LAB

The Design process:

- Review lecture material on K-maps and logic design.
- Read the problem statement, and from the problem statement, write the truth table.
- Create K-maps for each of the three outputs and find the minimized **SOP** implementation using AND and OR logic gates. Assume that AND and OR gates with 2, 3, or 4 inputs are available.
- From the K-maps find minimized logic functions in **POS** form.
- Compare the cost of the POS form and the SOP form when the cost is measured by the *sum of the number of logic gates and the number of inputs to all logic gates*.

III. LAB PROCEDURE

Schematic Entry and Simulation:

1. Draw the schematic for your SOP or POS design using the schematic editor in the Quartus II program. Refer to the tutorials if needed. (See references in part V.) Do a screen capture of your schematic.

2. Simulate your design and verify that it functions correctly. You should be able to convince yourself (and the TA) of the following:
   a. If there were no cars entering the highway, the right lane light would be on (you can think of this as the default condition)
   b. For any case where there is just one car present, the correct light would be on
   c. The round robin signal correctly dictates which light turns on when there are cars in both the left and right lane
   d. A car in the carpool lane gets priority, i.e., the carpool lane light would be turned on even if there is a car in either the left or right lanes (or both)

   You might want to checkpoint yourself with the TA at this point before moving to the next step.

3. Now, extend the functionality of the circuit with two more outputs, which will be a count of the cars currently present. Call the two outputs T1 and T0. These signals will be a function of just inputs CS, RS, and LS. I.e., RR has no bearing on the count. If no sensors are on, T1 T0 = 0 0. If one sensor is on, T1 T0 = 0 1. If two sensors are on, T1 T0 = 1 0. If all three sensors are on, T1 T0 = 1 1.

4. Test your design for all possible input conditions and, when it is working, demonstrate it to your lab instructor. Make sure you demonstrate the complete functioning of your circuit.

IV. REPORT

To be completed and turned in by the deadline established by your TA.

- Write an introduction describing how your circuit should operate and the design choices you made.
- Include your schematic and simulation results.
- Describe the simulation strategy that you used to test your circuit design. Did it identify any errors in design or implementation? If so, what were they and how did you correct them?
- Do you think your simulation strategy would detect all design errors for this circuit? Why or why not?
- What logic would you add to create a new output, ERR1 which would be 1 if two or more lights (CL, LL, and RL) were turned on at the same time. How is that logic similar to the logic that created the T1 output?
- Describe how you would modify your circuit to include a fifth input TM, a timer that is turned on at intervals controlled by the traffic density. When TM is "1", the circuit operates exactly as specified in the problem statement. When TM is "0", all output lights are red. Show a schematic for your modified circuit which includes the timer input.
- In the specification for the circuit, the carpool lane always has priority. If there were a long line of cars in the carpool lane, all other traffic would stop completely until the last car in the carpool lane had entered the highway. Describe a possible strategy to prevent total blocking of the cars in the non-carpool lanes but still allow cars in the carpool lanes to wait less time than other cars.

V. REFERENCES FOR LAB

CAD Tools

Quartus Introduction Using Schematic Designs

ftp://ftp.intel.com/Pub/fpgaup/pub/Intel_Material/18.0/Tutorials/Schematic/Quartus_II_Introduction.pdf

Quartus Introduction Using Verilog Designs

ftp://ftp.intel.com/Pub/fpgaup/pub/Intel_Material/18.0/Tutorials/Verilog/Quartus_Std_Introduction.pdf

Simulations using VWF for versions 13 and later

ftp://ftp.intel.com/Pub/fpgaup/pub/Intel_Material/18.0/Tutorials/Verilog/Quartus_II_Simulation.pdf