**Lab Number: 10**

**Section Number: 001**

**Names: Barak Barclay**

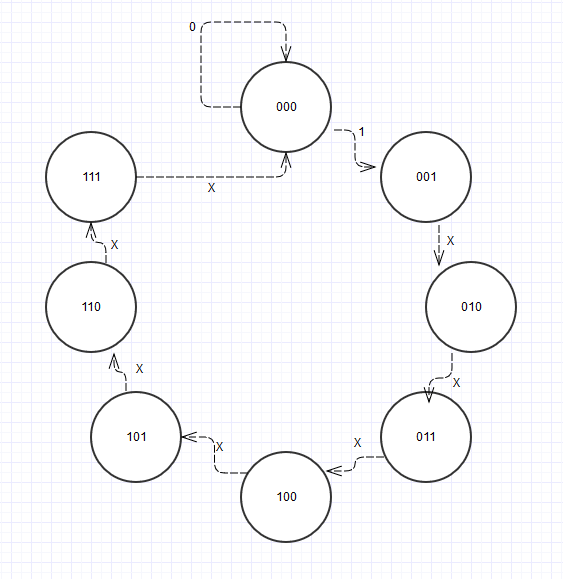
**Assigned Date: 04/28/2016**

**Due Date: 05/09/2016**

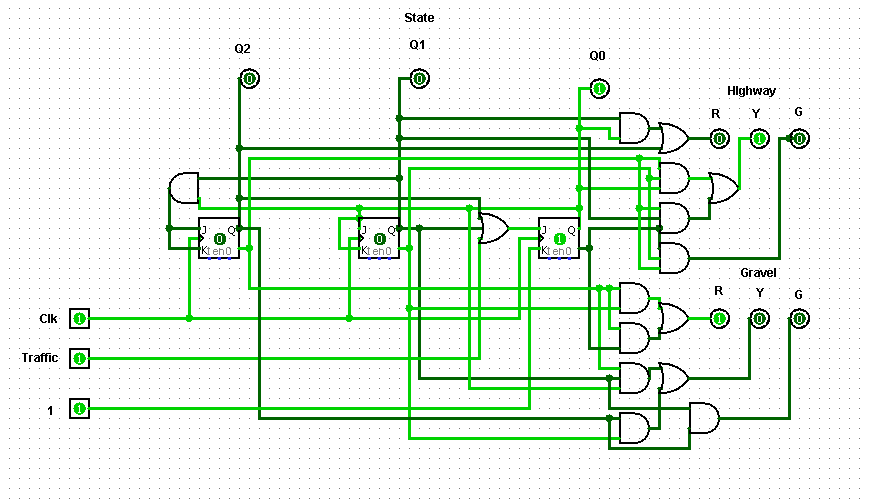
**Introduction:**

Part one of this lab is a state diagram, transition table, Karnaugh maps and Logisim schematic for a traffic light sequencer. As instructed, “upon detection of a vehicle on a gravel road, the traffic light sequencer changes the highway light from green to yellow for two clock pulses. It then changes the highway light to red and the gravel road light to green for three clock pulses. After that, it changes the gravel road light to yellow for two clock pulses and then changes the highway light to green and the gravel road light to red.” The modules in part two of this lab – written in Verilog, and simulated in Modelsim – create a FSM simulation of a traffic light sequencer using behavioral modeling. How everything was made will be explained thoroughly in the conclusion.

**Part 1:**

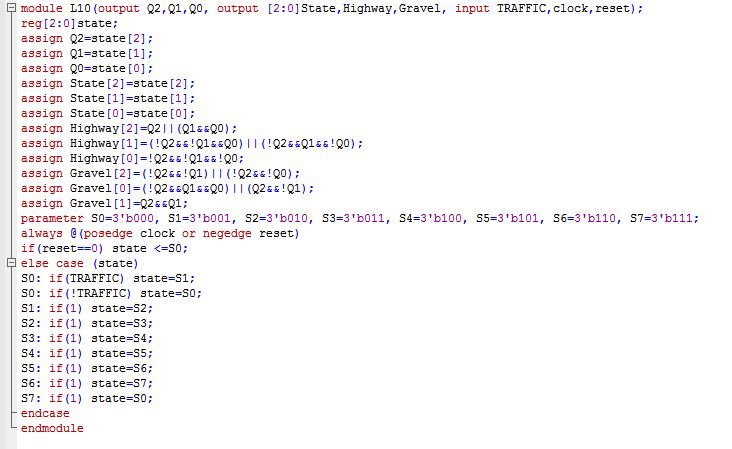


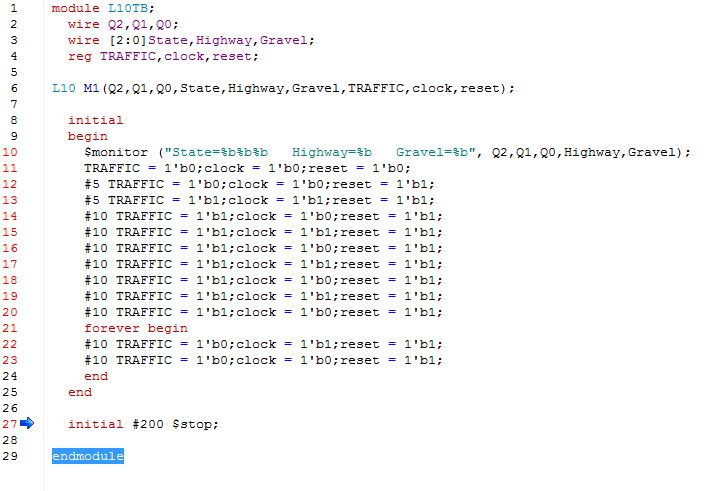
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Q2** | **Q1** | **Q0** | **T** | **FF INPUTS (present state)** | | |
| **Present to Next** | **Present to Next** | **Present to Next** |  | **J2K2** | **J1K1** | **J0K0** |
| 0-->0 | 0-->0 | 0-->0 | 1 | 0X | 0X | 0X |
| 0-->0 | 0-->1 | 1-->0 | 1 | 0X | 1X | X1 |
| 0-->0 | 1-->1 | 0-->1 | 1 | 0X | X0 | 1X |
| 0-->1 | 1-->0 | 1-->0 | 1 | 1X | X1 | X1 |
| 1-->1 | 0-->0 | 0-->1 | 1 | X0 | 0X | 1X |
| 1-->1 | 0-->1 | 1-->0 | 1 | X0 | 1X | X1 |
| 1-->1 | 1-->1 | 0-->1 | 1 | X0 | X0 | 1X |
| 1-->0 | 1-->0 | 1-->0 | 1 | X1 | X1 | X1 |
| 0-->0 | 0-->0 | 0-->1 | 0 | 0X | 0X | 1X |
| 0-->0 | 0-->1 | 1-->0 | 0 | 0X | 1X | X1 |
| 0-->0 | 1-->1 | 0-->1 | 0 | 0X | X0 | 1X |
| 0-->1 | 1-->0 | 1-->0 | 0 | 1X | X1 | X1 |
| 1-->1 | 0-->0 | 0-->1 | 0 | X0 | 0X | 1X |
| 1-->1 | 0-->1 | 1-->0 | 0 | X0 | 1X | X1 |
| 1-->1 | 1-->1 | 0-->1 | 0 | X0 | X0 | 1X |
| 1-->0 | 1-->0 | 1-->0 | 0 | X1 | X1 | X1 |

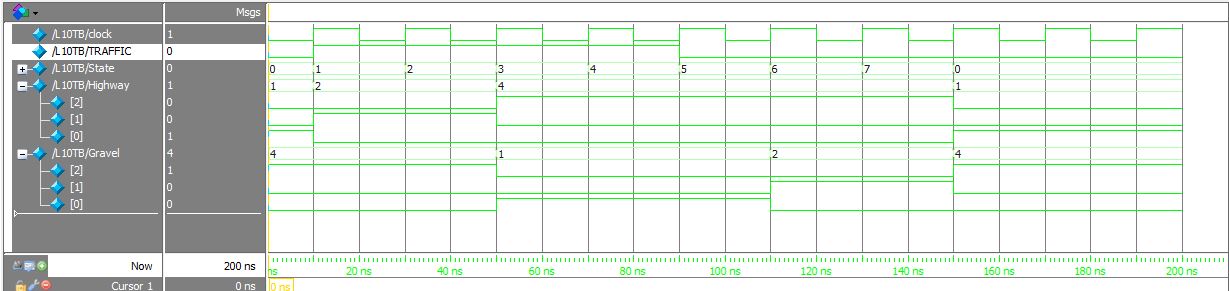


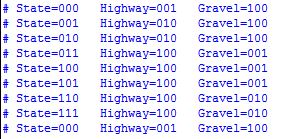
Karnaugh maps and sign-offs are on the last page.

**Part 2:**









**Conclusion:**

In part one, the state diagram in part one is a three-bit counter with the addition of an input that makes it so when the state is at 000, the counter will not move to the next state unless the input is 1. The transition table was created from the state diagram and the modified function table for a JK flip-flop. The Logisim schematic was built using three flip-flops and primitive gates using the equations derived from the Karnaugh maps. The snip of the schematic shows the completed circuit including LEDs simulating the traffic lights for both the highway and gravel road.

In part two, as I said above, behavioral modeling was used to create the modules. In the first module: the ports (outputs and inputs) were declared – with vector notation when needed, - a register was declared with vector notation, Q values were assigned to the specific register bit, output values were assigned to pass the register values, and all of the light outputs were assigned to specific register bits, parameters were declared. The rest of the module creates a FSM simulation that runs at each rising edge of the clock or a falling edge of the reset. The if statement checks if reset equals zero and if it is, sets state equal to S0. The else case statement runs if the previous if statement was false. This runs the register, state, through the values in the state diagram depending on the input, TRAFFIC.

The second module in part two is a test bench to run the first module. Outputs of the first module were declared as wires and inputs were declared as registers for the test bench. The following is the instantiate the first module. The “initial begin” allows the start of setting input values. The first line of code causes the snip of the transcript box in part two. The monitor statement will display the output values whenever any of the output values change. The following lines of code set the input values at certain times using time propagation. A forever loop was used after returning the TRAFFIC input back to 0. The module ends with a time propagation and stop command that allows the module to simulate for 200ns before stopping. The snips of the timing diagram and Transcript box show the simulation of the traffic light sequencer indicating the conditions of both traffic lights for all lighting conditions