ECEN 2350 Lab 3 Report
Adam Chehadi - 12/11/19

Problem Statement

The purpose of Lab3 is to emulate the behavior of tail lights found on a 1965 Ford Thunderbird. The LEDR[9:7] and LEDR[2:0] on the DE10-Lite board mimic the behavior of the left and right sides respectively of the Ford Thunderbird. The idle, hazard, left turn, and right turn behavior of the Ford Thunderbird is emulated with a state machine on the DE10-Lite board. The state machine transitions states based on SW and KEY user interaction, and automation of design using the MAX10 memory block.

Theory of Operation

Lab3 contains four states: idle, hazard, left turn signal, and right turn signal. In the idle state, all of the LEDs are deactivated. In the hazard state, LEDR[9:7] and LEDR[2:0] blink together once every clock cycle. In the left turn state, LEDR[2:0] are always deactivated, LEDR[7], LEDR[8], and LEDR[9] sequentially activate, and once LEDR[9:7] are all activated they deactivate and repeat that process. In the left turn state, LEDR[9:7] are always deactivated, LEDR[0], LEDR[1], and LEDR[2] sequentially activate, and once LEDR[2:0] are all activated they deactivate and repeat that process.

The updating clock frequency is created by dividing the ADC_CLK_10 clock provided by the DE10-Lite board. A clock_divider.v module is created and passed in the desired frequency parameter, the system ADC_CLK_10, the reset latch from KEY[0], and a slow clock wire to output the slower clock frequency. The clock_divider.v module takes those values and flips the value of the slow clock every time it's one less than the desired frequency parameter. clock_divider.v is instantiated in Lab3.v for 1Hz.

KEY[0] is the system reset, and is a latched pushbutton. SW[0] controls the hazard state, 0 = off, and 1 = on. SW[1] enables turn signals, 0 = off, and 1 = on. KEY[1] is a latched pushbutton, and toggles between the left and right turn signal. The left turn signal activates when KEY[1] = 0, and the right turn signal activates when KEY[1] = 1. Each state has a numeric value. Idle state parameter = 3'b000, hazard state parameter = 3'b001, left turn state parameter = 3'b010, and right turn state parameter = 3'b011. HEX0 is utilized to display the numeric value of the current state register.

A CSL.v module is created to utilize current state logic. CSL.v takes in a clock, reset, NextState, and CurrentState. CSL.v has an always block with a sensitivity list that contains the clock and reset. If the reset = 0, CurrentState returns to idle, otherwise CurrentState = NextState.

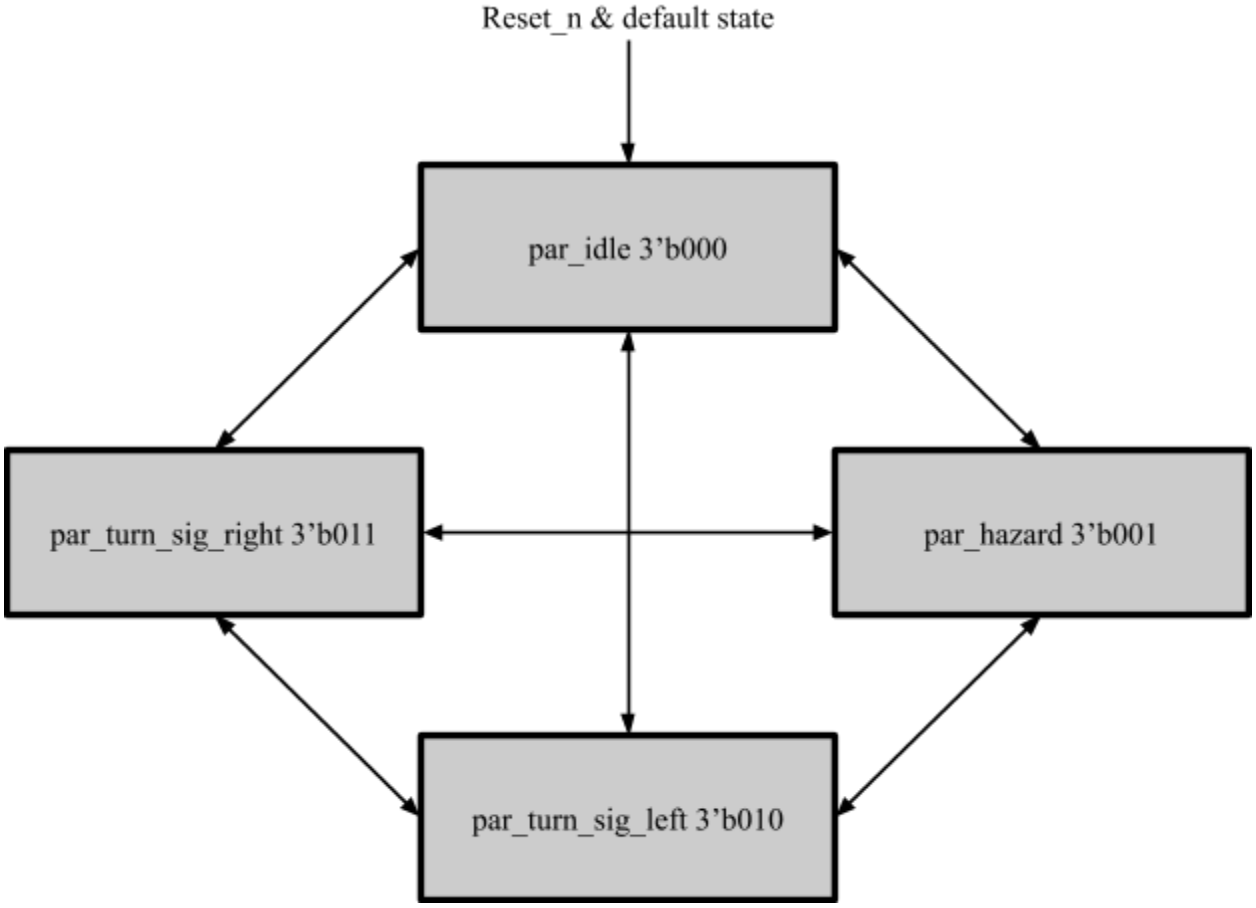
A NSL.v module is created to utilize next state logic. NSL.v takes in the KEY[1] latched turn signal, SW[1:0], CurrentState, and NextState. NSL.v has an always block with a sensitivity list that contains CurrentState. If SW[0] = 1, NextState becomes hazard. If SW[1] = 1 and

KEY[1] = 0, NextState becomes left turn. If SW[1] = 1 and KEY[1] = 1, NextState becomes right turn.

A OL.v module is created to utilize next output logic. OL.v takes in a clock, reset, CurrentState, SW[1:0], HEX0, LEDR[9:7], and LEDR[2:0]. OL.v has an always block with a sensitivity list that contains clock and reset. If reset = 0, all of the LEDs are deactivated. Otherwise, a case statement is entered that evaluates CurrentState. If the CurrentState is hazard, a reg toggles on/off to activate/deactivate the LEDs. If the CurrentState is left turn, LEDR[2:0] are deactivated, and LEDR[9:7] cycles through the values 3'b001, 3'b011, and 3'b111. If the CurrentState is right turn, LEDR[9:7] are deactivated, and LEDR[2:0] cycles through the values 3'b100, 3'b110, and 3'b111. The numeric value of the CurrentState is displayed on HEX0 as well.

Lab3.v is the top module of the project. Lab3.v blanks LEDR[6:3] and HEX5-1, latches KEY[0], latched KEY[1], creates the 1Hz clock, and creates 3-bit wires for CurrentState, NextState, CurrentState_mem, and CurrentState_manual. The CSL, NSL, and state_machine modules are instantiated. A state_machine.v module is created that counts through addresses in the .mif file. This address, the system clock, and a memory_output wire are passed into the M9K quartus module. The memory_output is then passed into an always block and evaluated in a case statement to set the CurrentState. An output_mux.v module is instantiated that determines if the CurrentState is based on the memory module or manually controlled user input. After all of this, the output logic module is instantiated.

State Bubble Diagram and State Table

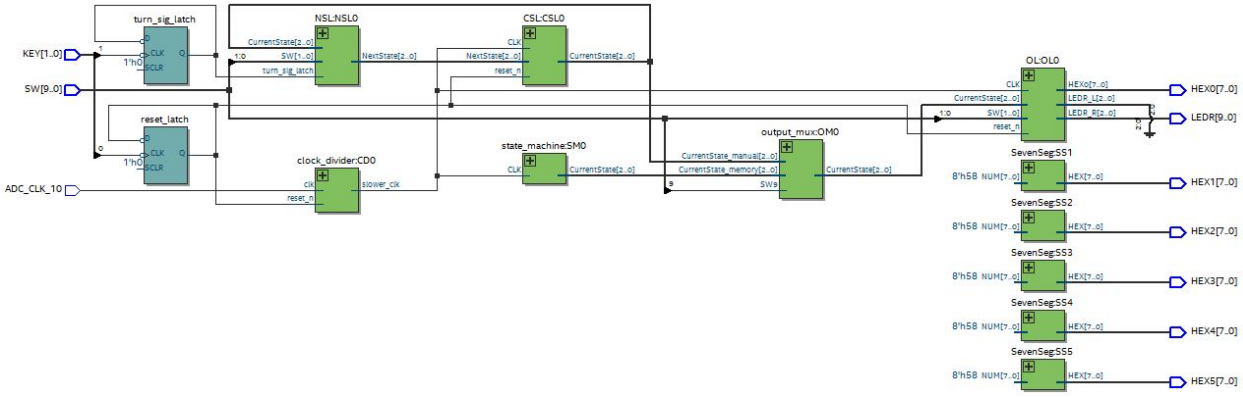


State Bubble Diagram: Finite state machine, state nodes, and transitions between states.

KEY[0]	KEY[1]	SW[0]	SW[1]	Current State
0	x	x	x	3'b000
1	0	0	0	3'b000
1	x	1	x	3'b001
1	0	1	0	3'b010
1	1	1	0	3'b011

State Table: Current state based on inputs to the state machine

Hierarchy of Source Files



RTL Viewer: Block Diagram of verilog modules with their inputs and outputs.

Lab3.v is the top level of this project. Lab3 instantiates CSL, NSL, state_machine, output_mux, and OL. Lab3.v also instantiates SevenSeg and clock_divider. OL.v instantiates SevenSeg.

Description of Testbench Operation and Output

A single testbench is used to evaluate the behavior of Lab3. The testbench begins by instantiating Lab3.v, and triggering CSL and NSL, which instantiates OL. The testbench includes switches and keys in order to emulate user interaction on the DE10-Lite board. The outputs of GTKWave and screenshots are contained in the main project directory. The CurrentState, NextState, LED, and HEX0 values can all be seen updating with the clock.

Summary of Project Success, What Works and What Doesn't

The project fulfills all of the project requirements. The design is properly able to switch Ford Thunderbird taillight states based on user input and M9K ROM.