ECEN 2350 Project 3

Mason Milligan

November 25, 2020

Problem Statement

The purpose of this project was to emulate the behavior of the taillights on a 1965 Ford Thunderbird using the DE10-Lite board. This design covers the functionality of the brake lights, the hazard lights, and the turn signals. The controls for the system are as follows:

- KEY[0] is a system reset
- SW[0] is the hazard light switch. It enables the hazard lights in the UP position and disables in the DOWN position.
- SW[1] is the turn signal switch. It enables the turn signals in the UP position and disables in the DOWN position.
- SW[2] is the brake switch. It enables the brake lights when UP and disables when DOWN.
- The hazard lights override both the turn signals and the brakes.

Additionally, the design has an automation feature that can be enable by moving SW[9] to the UP position. This automation feature cycles through the system's states, simulating different inputs to enter each state. This feature takes advantage of the board's available ROM through the use of Quartus IP.

Theory of Operation

Unlike the previous two projects, project 3 functions as a state machine. Since the output behavior of this system is more complex, the use of a state machine enables the design to have a more modular structure. A separate state can be defined for each unique output of the system, making for more efficient development, improved ease of debugging, and better code readability. Before development began, a state diagram was created to model the expected behavior. This diagram can be seen in figure 1.

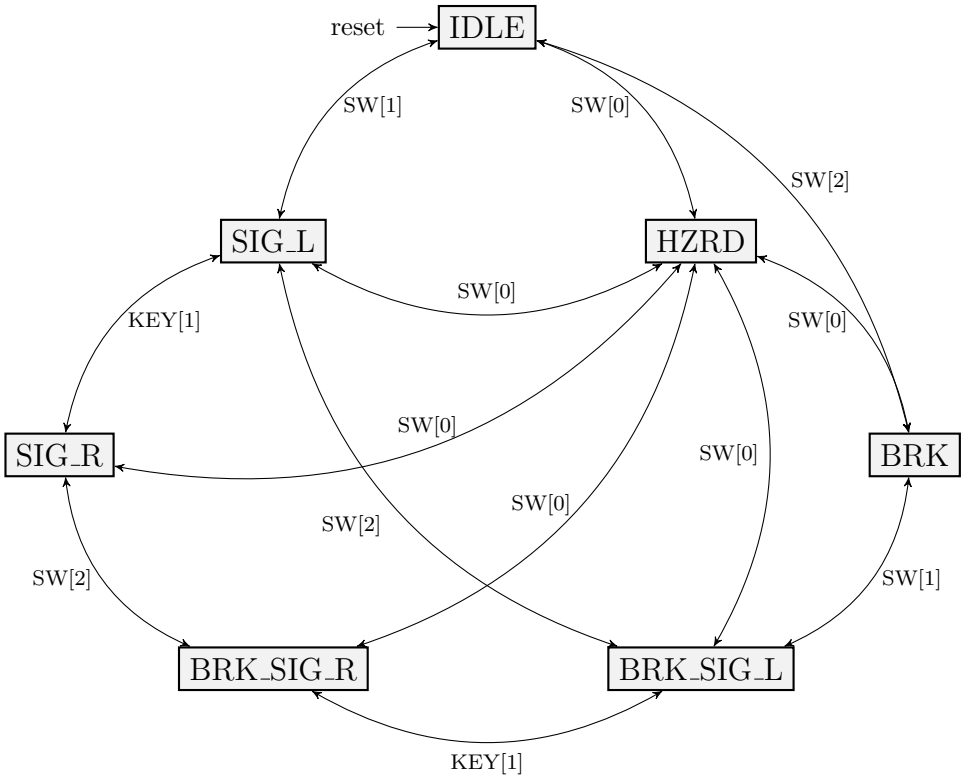

Figure 1: State diagram of system

While this diagram is a good representation of the behavior of the system, some possible transitions have been omitted to prevent the diagram from being overly complex. The reality is that the next state logic module checks all of the applicable inputs at all times, so any state can be entered from any other state.

Module Hierarchy

Figure 2 shows the hierarchy of modules that make up the functionality of the project. Text in bold represents modules, while non-bolded text represents instances of modules. The branch that begins with the module `rom` is entirely made up of Quartus IP modules, used for the automation portion of the design.

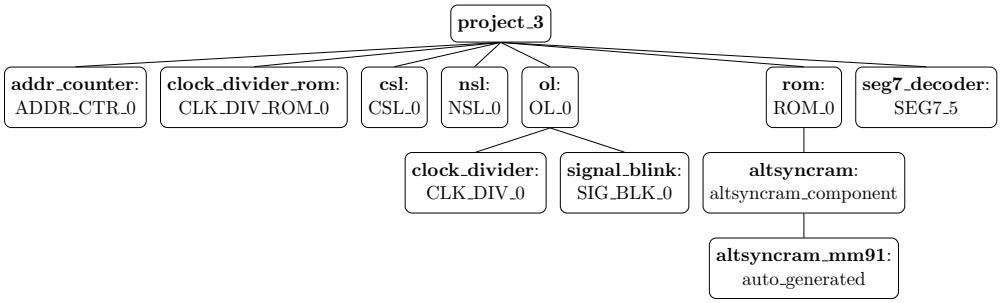

Figure 2: Hierarchy of project modules

Block Diagram

Figure 3 shows a schematic of the design to offer a visual on how the modules of the system interact.

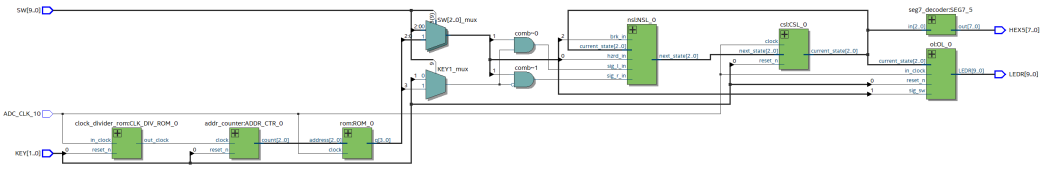

Figure 3: Schematic of system

Testbench Operation

A testbench was used to verify that the correct states were entered and that the correct outputs were presented for their respective inputs. While only

one testbench necessary, two views have been provided. The first, seen in figure 4 view demonstrates the system entering the appropriate states for all inputs. The second, seen in figure 5 demonstrates the system's outputs that vary based on the state. The use of testbenches streamlined the development process by rapidly displaying the behavior of the system without the need to compile for the DE10-Lite and interact with the board, which can take a very long time. Due to Icarus Verilog being unable to simulate Quartus IP modules, the automation portion of the project was not tested via testbench.

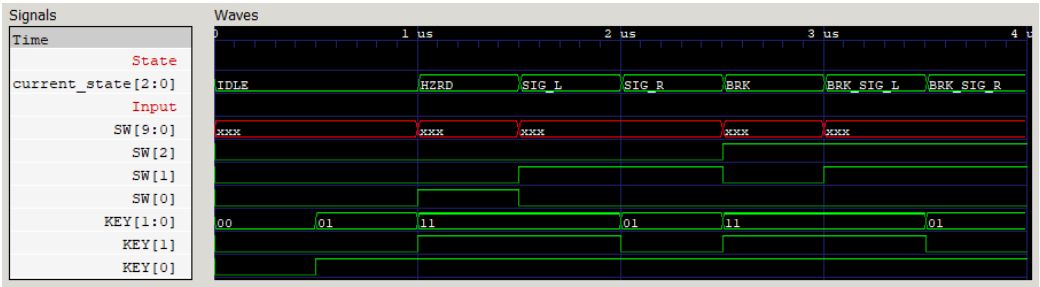

Figure 4: GTKWave output displaying inputs and their respective states

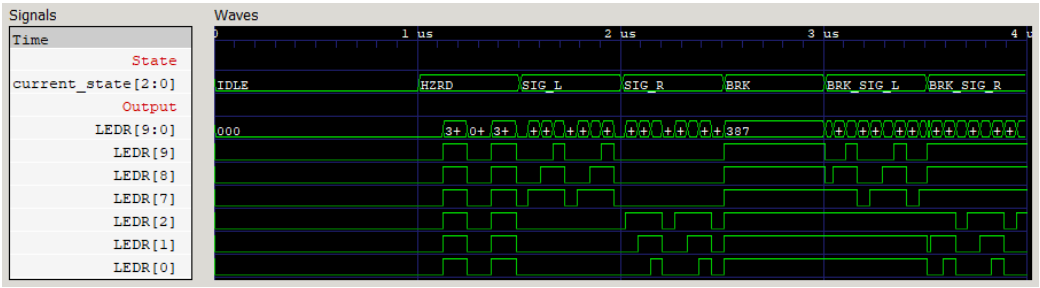

Figure 5: GTKWave output displaying states and their respective outputs

Closing Thoughts

With the aid of testbenches and the automation functionality, which was enabled through the use of Quarus IP, the expected behavior was achieved within the system. Upon compiling the design and flashing it to the board,

the final product worked as expected. All of the design specifications were met by the system and the LEDs responded as simulated. For this reason, the design can be considered a success.