Assignment 1 Report

Dhruv Sagar, Rodrigo Mejia

**Challenges and Implementation**

The core challenge of this assignment was designing the state machine that was to handle the changing states. The difficulty in this was both making sure that we went through all the states in the appropriate order, as well as making sure that all the states were cycled through without user input. For the most part, this wasn’t too large of an obstacle. Implementing the state machine was done through two registers, once for counting the time passed in a certain state and another to keep track of the state that we were currently in. Once the appropriate amount of time had passed we would increment the state register. In this same way we handled whether within a larger macro-state the lights should be on or off during the time they were supposed to be blinking. We implemented some “mini-states” that we would cycle through to determine whether within the current macro-state the currently blinking lights should be on or off. This was how we handled both the aspect of timing in which we had to keep lights on or off for a certain amount of time, and also how we handled which lights should be the ones that were blinking.

Handling how fast the lights should be blinking was on the easier side of the challenges. We simply kept track of the number of clock cycles a “blink” should be and incremented or decremented this value by a quarter-second depending on the registered key press.

One last difficulty would have been to have the key presses registered correctly. Because a clock cycle is faster than an individual person can press and release a key only once, there was a need for an implementation that would register a key press only once, and not repeatedly. This was done for us by using a register that held the opposite value of the current key-press state. Then the assertion of the key press was a wire that held the and of the negation both of these aforementioned values. Because the key press register is updated every clock cycle with the opposite value of the key press, the previously mentioned key press wire can only hold its assertion for one cycle before losing it due to the key register flipping values on the next cycle. This then solves the problem of possibly registering multiple key presses from one single key press.