ECEN 2350 Digital Logic:

Project 2 Report

Ramon Martinez

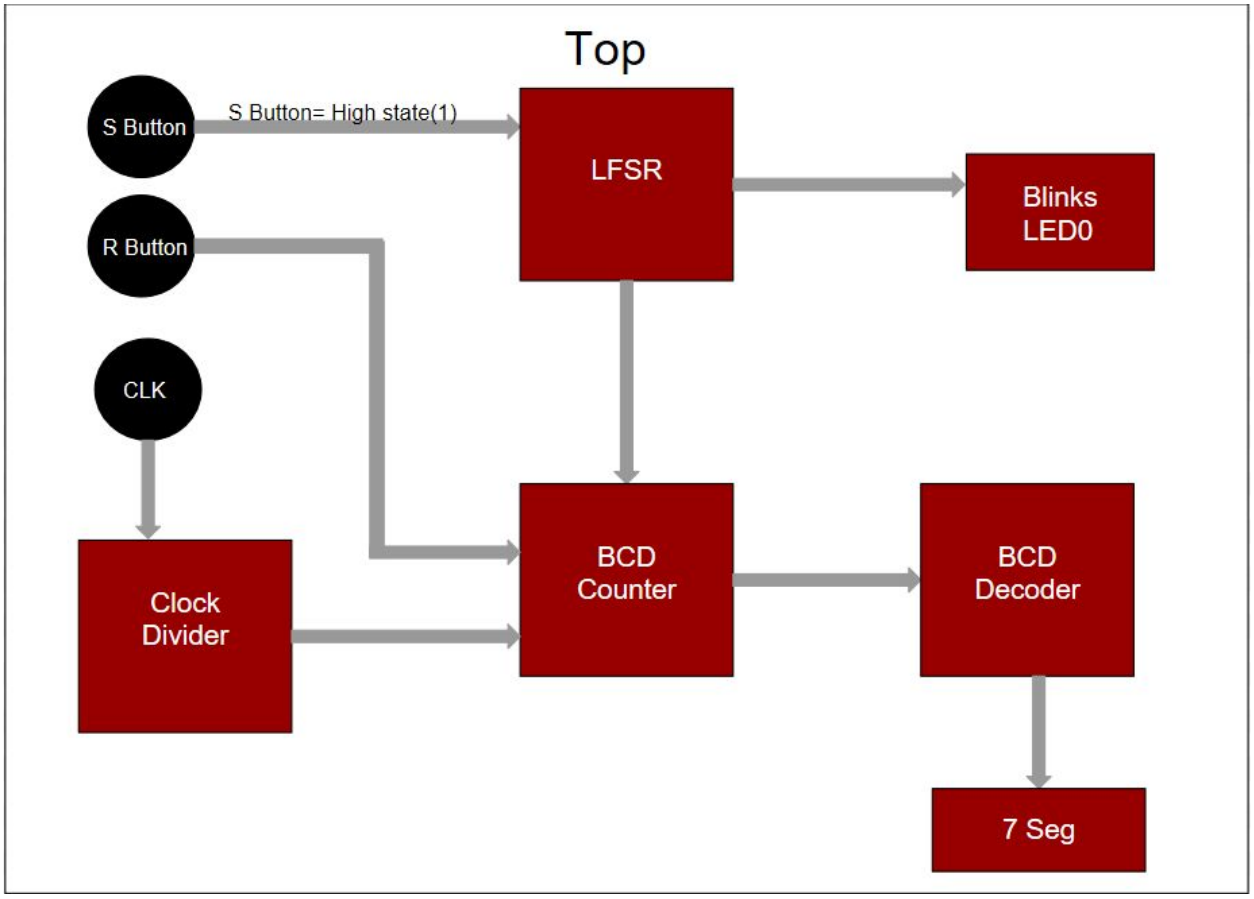
Sean DeVaney

Rafael Espinoza

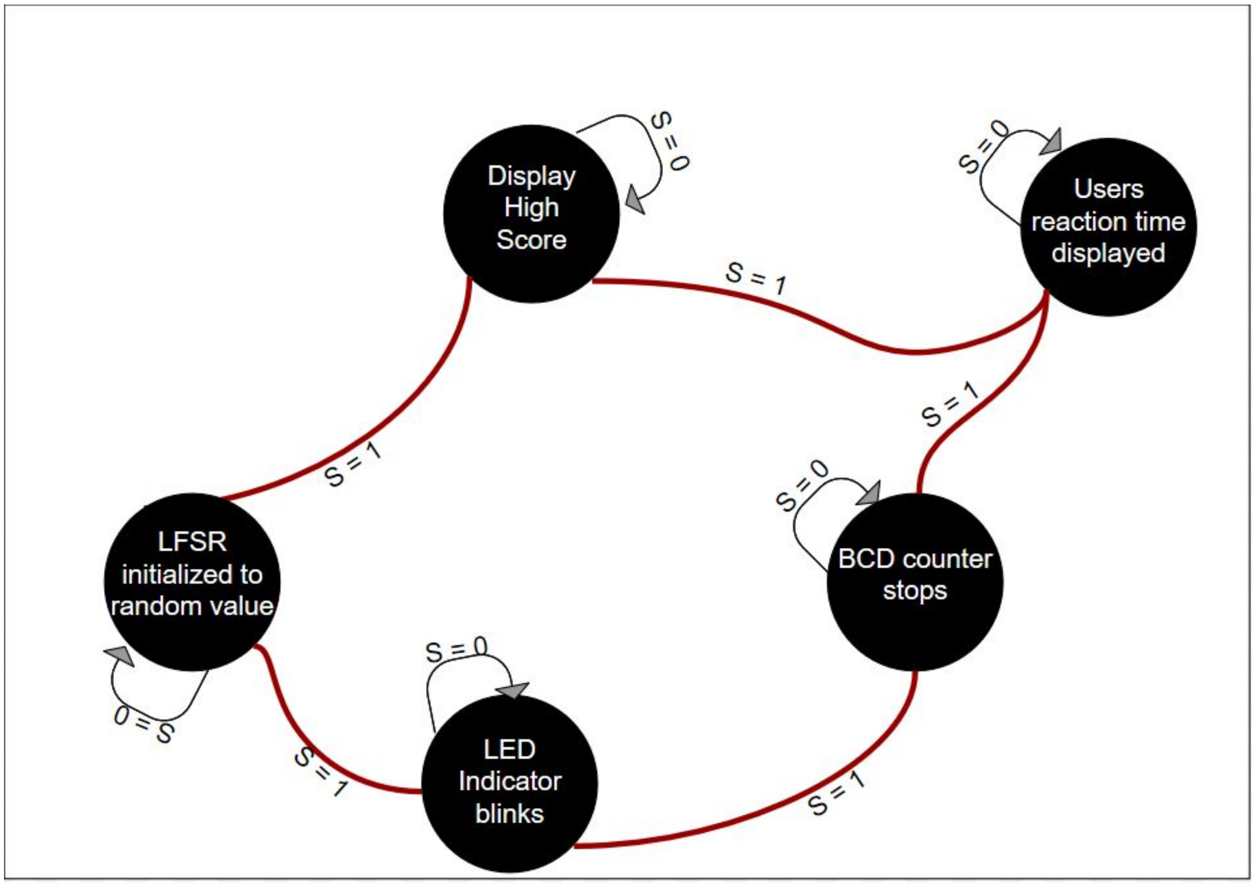
Introduction:

For this project, we’ve developed code to create a reaction timer using a state machine. To start, the user must press Key0. This causes a countdown for a random amount of time. Once the time is up, an LED on the board will light up. The user reacts to this LED and presses Key1 to test their reaction time. This reaction time is displayed, and hitting Switch0 will display the high score. From the high score screen, the user is able to press Key0 once again to repeat the test.

Block Diagram:



State Machine:



Description of Code:

Project2_top.v: This module by running the Clk_divider, which will create a slower clock in our machine. Then we generate a random number with the LSFR which will be counted down by count_down. Then we call our state machine, which will dictate which state our machine is. We then call LEDLighter to decide which LEDs should be turned on to signal the state the machine is in, and assign those LEDs. Then, we call BCD_counter, which will activate and countdown whenever the enabler is turned on, and then we call BCD_encoder, which will determine which numbers from the counter the Seven Segment display should show. After that, we assign those numbers to the Seven Segment displays.

Clk_divider.v: This module slows down the clock by taking in the device's clock and creating a new clock that switches every 2500 counts of the clock. The new clock counts in milliseconds.

BCD_counter.v: This module takes in our clock clk, a reset button, an enabler, and 5 output S1, S2, S3, S4 and S5. This module updates the always block a clock cycle has passed. Then, if the enabler is 00 or 01, it sets the output to 0. If the enabler is 03, which is our counter state, it will begin incrementing our S outputs accordingly.

BCD_encoder.v: This module takes in a bcd input and led output, and will assign the decimal number from the bcd to the led, depending on what that is.

StateMachine.v: The state machine takes in clk, the modified clock, clear, our reset switch, Start and Stop, our buttons, Result, our LSFR output, and z, our state machine output. We also have two variables y and Y. The main bulk, the code checks if there’s a change in y, and then proceeds to the four case statements. The first case, if Start is pressed, then Y is changed to B, if not it stays as A. The second case, if the result is 0 and the Start button hasn’t been pressed, then Y is set to C, if not it stays as B. For our third case, if the Stop button is pressed, then Y is set to D, if not it stays at C. Finally, if the reset button is switched, Y is set to A, if not it stays at D.

Then there’s the second always block, which is updated every clock cycle and when the reset switch is hit. In it, if the reset switch is hit, y is set to A, sending the state machine to state A. If not, y is set to Y, which will then set which state is being run. The final always block updates everytime y changes. This one is in charge of selecting the module’s output depending on what state y is currently set to.

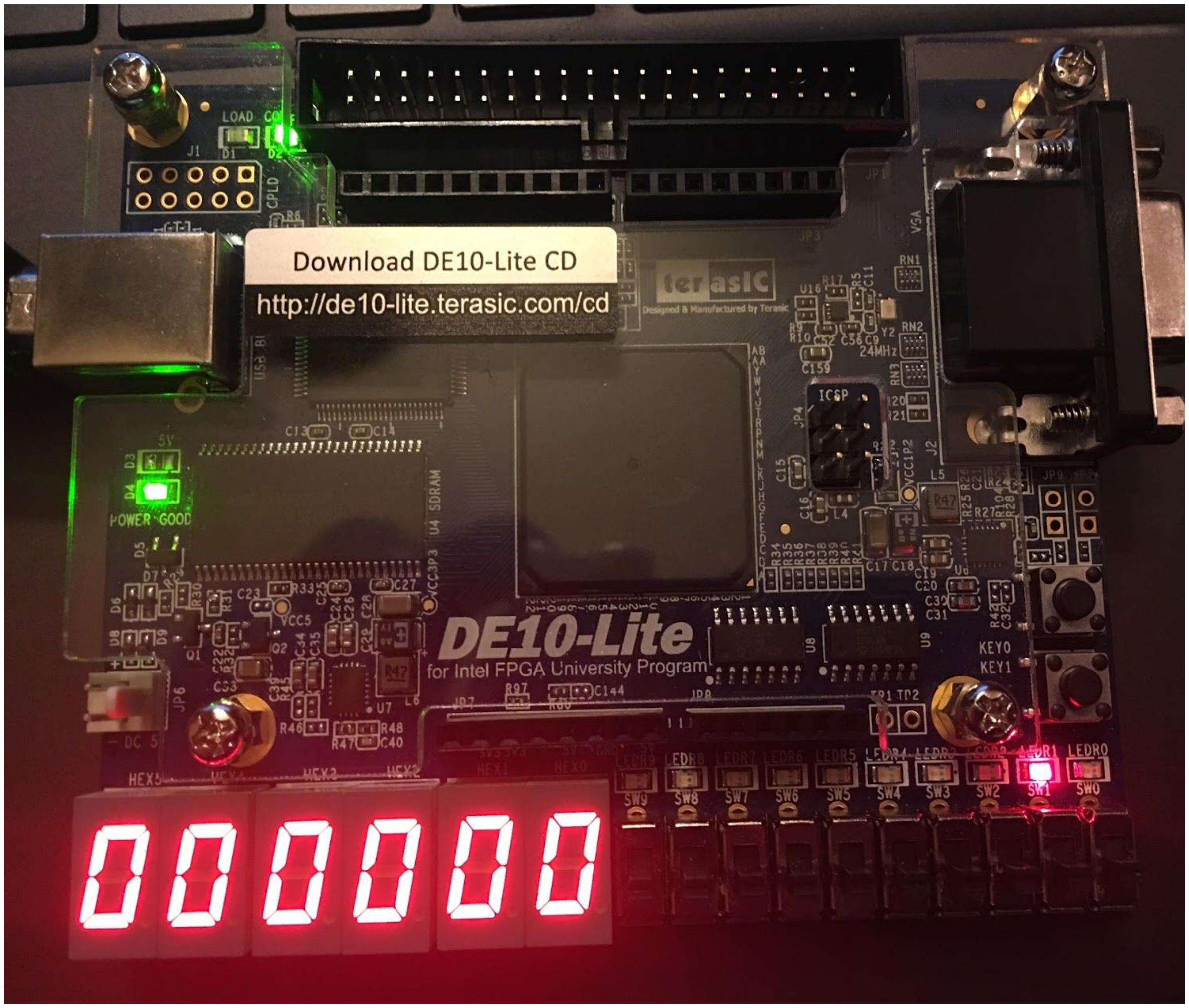
LEDlighter.v: This module takes the output from the State machine, as well as five inputs A, B, C, D and LED. The always block updates every time there is a state change. Inside, we assign values to A, B, C, D and LED depending on which state the machine is currently in.

LSFR.v: The LSFR module creates a random number and then counts down to that number and outputs a zero when the counter has reached zero. When this happens LED0 turns on and indicates to the user that the reaction timer has begun.

Reaction Time Measurements:

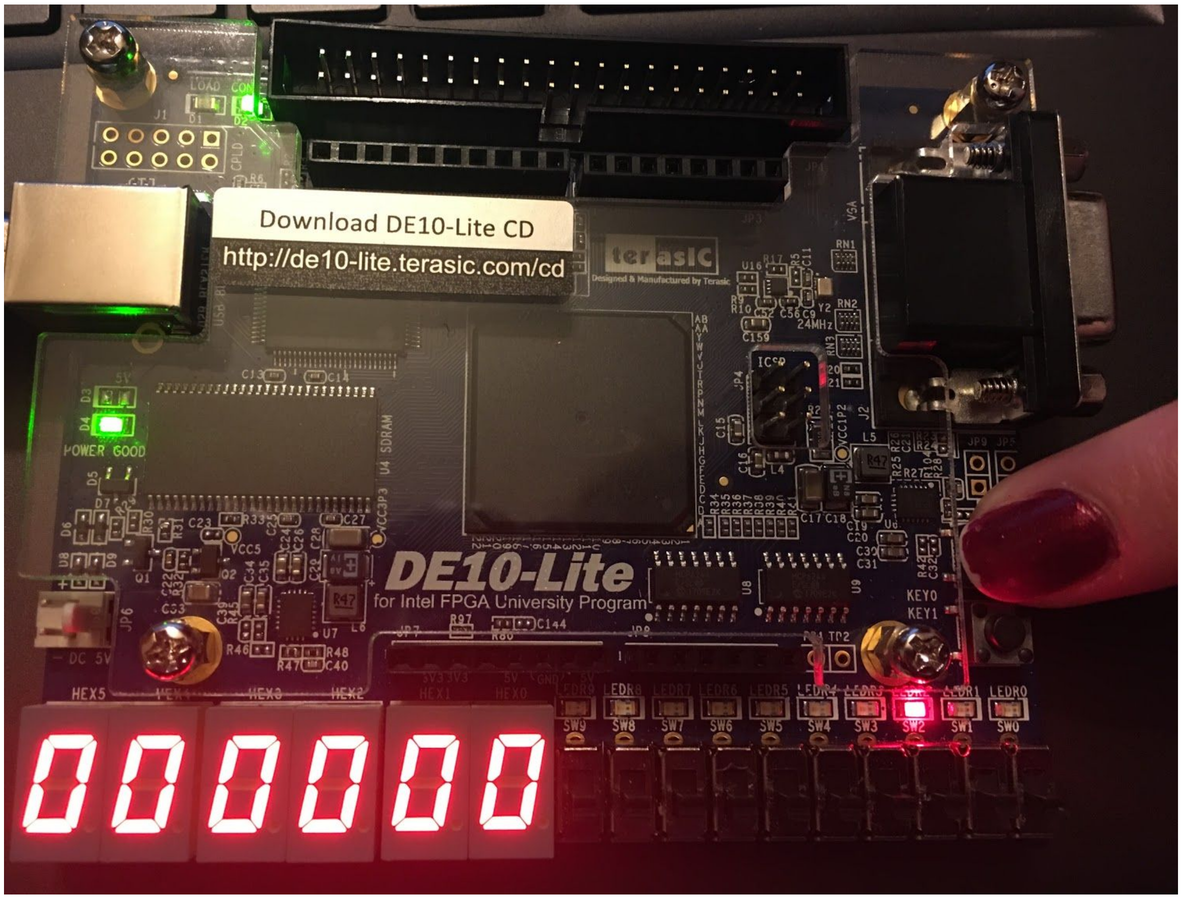
Measurement:	Reaction Time:
1	920 ms
2	172 ms
3	184 ms
4	244 ms
5	145 ms
6	70 ms
7	536 ms
8	364 ms
9	232 ms
10	131 ms

Idle State:


Download DE10-Lite CD
http://de10-lite.terasic.com/cd
DE10-Lite
for Intel FPGA University Program

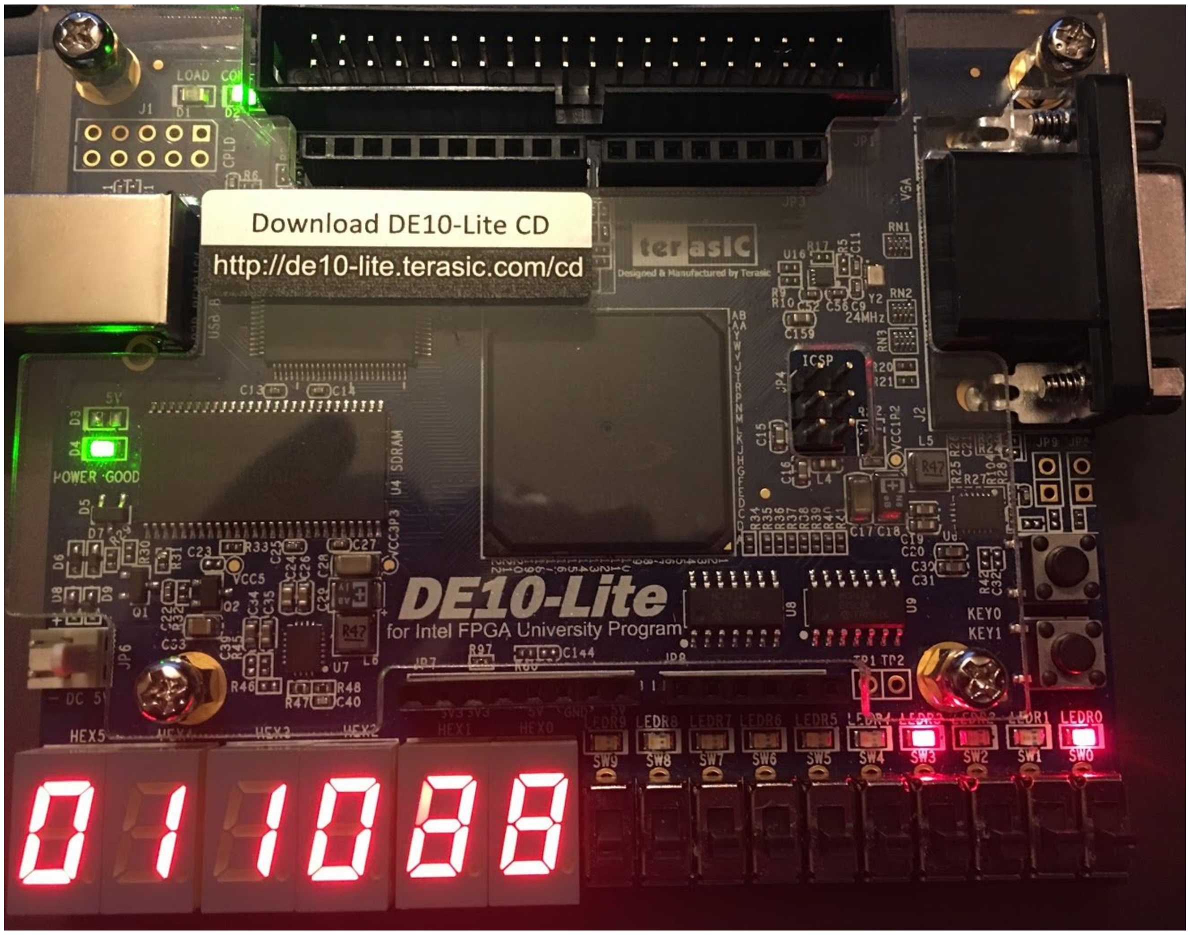
When in the idle state LED1 turns on and the seven segment display is set to zero.

Delay State:.



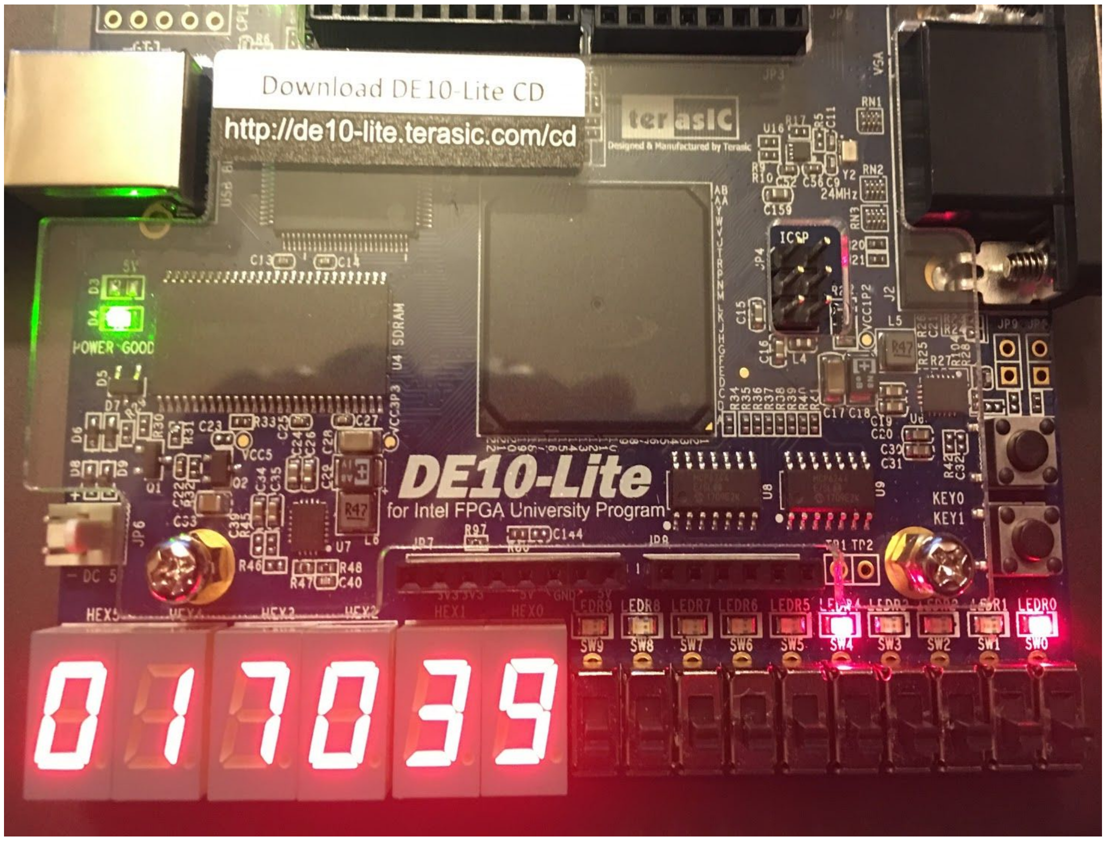
The next state is when the delay is set LED2 is turned and stays on tell the delay has been meet.

Reaction Timer:


Download DE10-Lite CD
http://de10-lite.terasic.com/cd
DE10-Lite
for Intel FPGA University Program

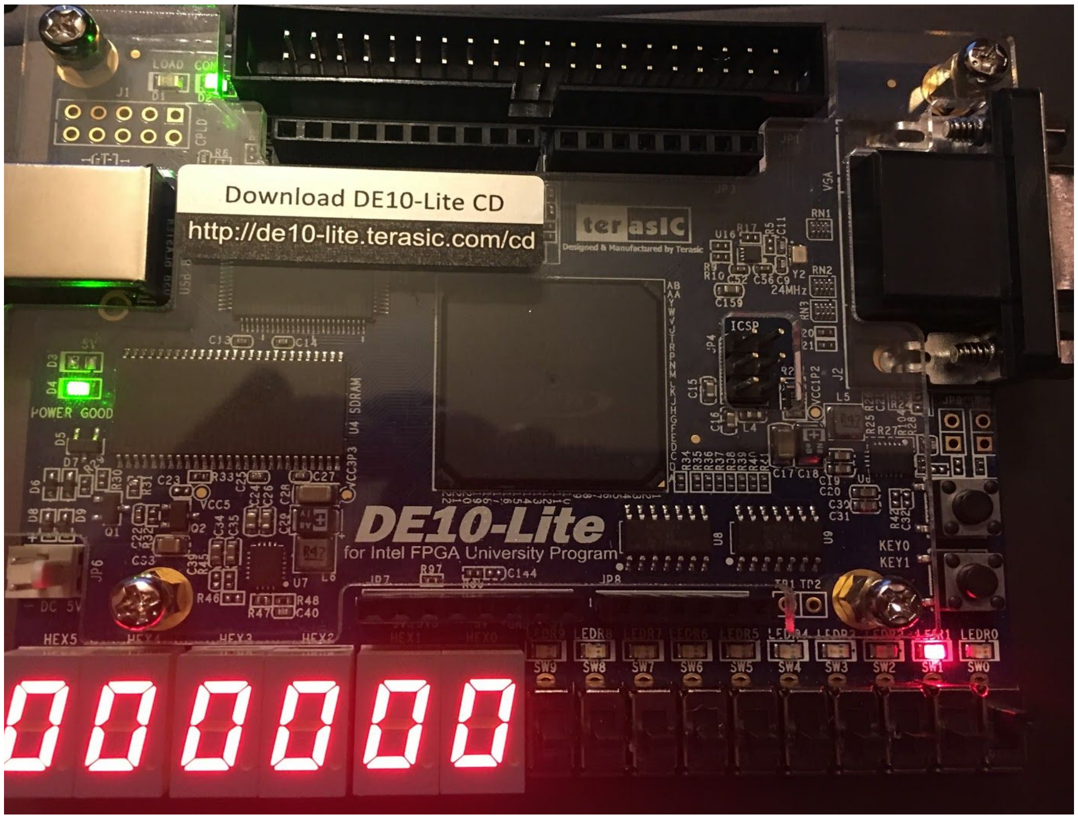
When users reaction time is being measured LED0 and LED3 are on and the seven segment display updates as time passes.

Reaction TIme Displayed:


Download DE10-Lite CD
http://de10-lite.terasic.com/cd
DE10-Lite
for Intel FPGA University Program
017039

Once user has hit the stop button LED4 turns on and the seven segment display stops updating to display time it took for user to stop the timer.

Reset State:



Once users is content with seeing their score they can reset the game to try to get a faster time. The seven segment display is cleared and LED0 is turned off while LED1 is turned back on.

Conclusion:

For this project we implemented a moore state machine to measure the reaction time of the user. Our DE10 board startes in the idle state and waits for user to interaction. Once user has started interacting with the board, the BCD counter is called and keeps track of the elapsed time in milliseconds until user has hit the stop button.Our boards display in continually updated until the stop button it. The biggest hurdle we meet during the project was getting the state machine to work with our other modules and to make sure we could get in and out of states properly.