System on Chip: Class Report 2

Noel Sengel and John Westbrook

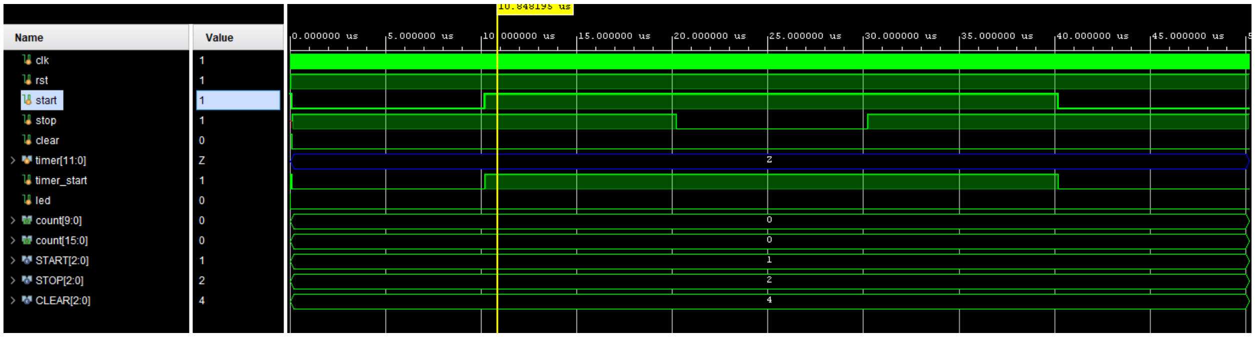
November 25, 2023

Summary

In this class report, our goal was to create a reaction game using an LED on our NEXYS board. Once the LED lit up, a timer started to see how long it took the person playing to press a button and react. To accomplish this, we took code from a previous class which converted binary to Binary Coded Decimal and displayed it on the 7 segment display. Then, we added our own code and edited the modules to display the millisecond timer while it was running. Our main development occurred in one module, ”reaction_timer”.

Results

Here is a test we did for reaction_timer module which ensures its functionality



Name	Value
clk	1
rst	1
start	1
stop	1
clear	0
timer[11:0]	Z
timer_start	1
led	0
count[9:0]	0
count[15:0]	0
START[2:0]	1
STOP[2:0]	2
CLEAR[2:0]	4

Figure 1: reaction_timer Simulation

In this GitHub link provided below, there is a video of our board working.

Code

Here is the GitHub repo with our modules: https://github.com/JhnWstbrk/ELC4396_ClassReport2

In Listing 1, you can see a section of our main module, ”reaction_timer”, which is the main logic that implements the game part. This code was very hard to write at first, but after some revisions, John and I created a working timer that functioned as necessary. When the code starts, it displays ”HI” and then when the Start button is pressed, a certain amount of time waits and then the LED will turn on. Once the button is pressed during the 1 second, it displays the recorded time, else it displays ”1000.” If the user presses before the timer goes off, it displays 9999. We learned from this code that it is very hard to drive an SSEG when there is multiple values you want it do display based off of the state machine. It took a lot of playing around to get it to work correctly.

Listing 1: Main Logic of reaction_timer

```
...
   always_comb begin
           if(reaction_state == START) begin
                 led = 1'b1;
                 led_on = 1'b1;
                  timer_start = 1'b1;
           end
           if(reaction_state == STOP && timer_start == 1'b1) begin
                //display last time on the screen
               timer_start = 1'b0;
               led = 1'b0;
           end
           if(timer == 1000) begin
              timer_start = 1'b0;
              led = 1'b0;
          end

           if(reaction_state == STOP && led_on != 1'b1) begin

           end
           if(reaction_state == CLEAR || rst == 1'b1) begin
               led = 1'b0;

           end
       end
...
```