Allan Frederick

EE316 Digital Logic

12/1/2019

Lab 6 Report: Stopwatch

Design Approach

The way I approached this lab was realizing that the lab 6 stopwatch was essentially the application of several modules we had created in previous labs, much like how pieces fit together to form a larger puzzle. This way, it was easier to visualize and design such a solution.

The stopwatch module consists of a main module that instantiates several of the modules we had previously created. The design of the stopwatch breaks down into 3 main components: the display, the system inputs, and the counter itself. It was important to realize that much of the work for the actual display output had already been created. Specifically, the modules related to the display included everything we did from lab 4.

The start/stop button functionality is created in its own module which also includes the implementation of a falling edge detector module. One key realization was that to ensure the start/stop button functioned properly, it was important to consider the noise generated upon release of the button. This is why I included a flag that detected the falling edge signal of when that button is pushed, which was done by utilizing the edge detector module from lab 4. In other words, the results of the button push would only take effect after the release of it.

The reset button was also implemented in its own module, however, no edge detection was necessary. The switch inputs are directly integrated into the main stopwatch module because I thought that was easiest considering that they did not require any other external modules.

The counter itself is implemented in the main module. The most important aspect to this is that it requires a separate clock that operates at a different frequency than the seven-segment display FSM. The counter clock runs at 10 ms. The design of the counter is pretty straight forward- depending on the mode, the appropriate values are loaded into certain registers representing the digits on the display, and these digits are updated using non-blocking at the speed determined by the counter clock divider. Once the values in these registers reach the terminal count values, a flag is given that results in these registers being loaded with the terminal count values until the reset button is pushed. This gives the appearance that the counter has "stopped".

The first key milestone was coming up with an actual diagram of the system that helped me visualize which parts were needed and how all these parts would work together. In other words, I was able to realize which modules would need to be created. Another key milestone was realizing that two seperate clock dividers were needed and finding out the count value necessary for the timer.

There were two main key setbacks. One was not being able to test on a real board until the day it was due, which delayed my ability to turn in this assignment on time. Another setback was using Vivado itself on Windows bootcamp, on my Mac. Because Windows is not native to my computer, certain drivers and software like Vivavdo do not function smoothly making the entire process much more frustrating than it needs to be.

Summary

The stopwatch main module contains several smaller modules, some of which we had previously created. These modules are: stopwatch (main), hex2seg converter, seven-segment display FSM, display clock divider, start/stop signal, edge detector, reset signal, and counter clock divider. The functionality of the stopwatch breaks down into 3 critical components: the display, the system inputs, and the counter itself. The stopwatch inputs consisted of a start/stop button, a reset button, and switch inputs to determine the external load values and mode. It also contains a seven segment display that outputs the numbers ranging from 00.00 to 99.99 seconds.

Block Diagram