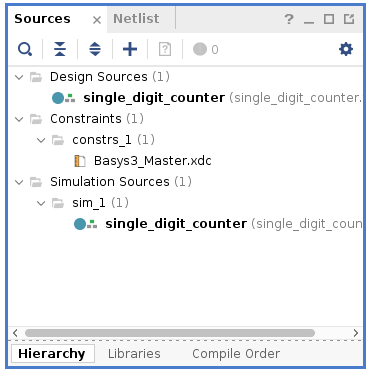
Lab 3 Sequential Circuit – Counter Design Name: Jacob Hillebrand

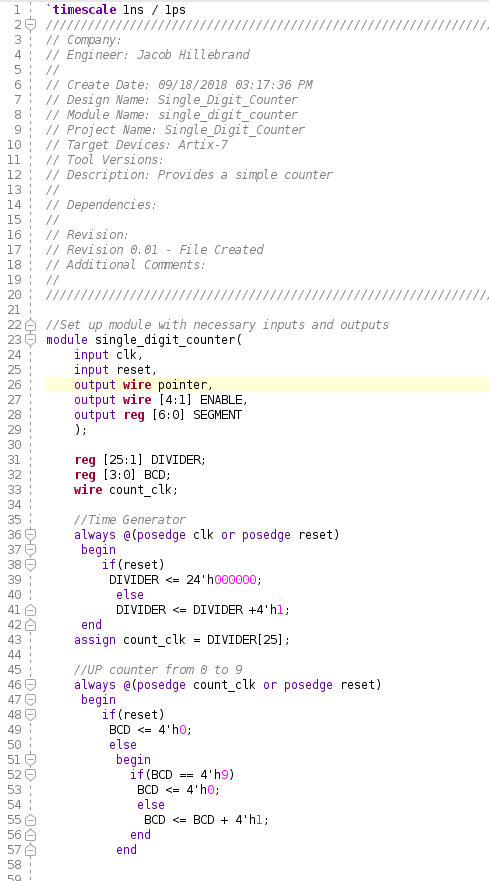
September 19, 2018

In this particular lab, we continued to learn how to use the functionality of Vivado to load a simple counting circuit onto our FPGAs. The idea was simple; the design would implement a slow clock based on the system clock, which would in turn signal a counting circuit. This counting circuit would cycle through the numbers 0-9, and reset to 0 when it hit nine. The number cycle would be triggered by the rising edge of the slow clock circuit. This counting sequence would be connected to the 7-pin display on the board, and would accurately display the current state of the counter. Additionally, each rising edge of the clock would be indicated with a flash of the decimal point on the 7-pin display.

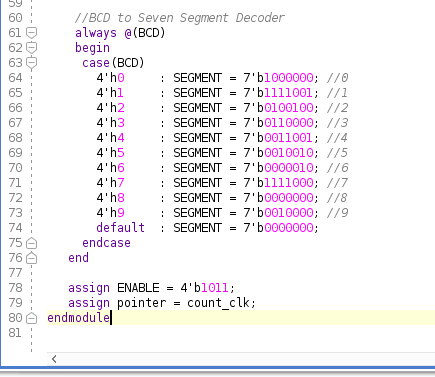
To begin with, a new project in Vivado was created. As this was a simple design, only two files were necessary, the Baysis3\_Master.xdc constraint file and a single\_digit\_counter.v. The file hierarchy is as shown below.



The single\_digit\_counter.v file consisted of a few primary parts, as shown in the screenshot on the next page.

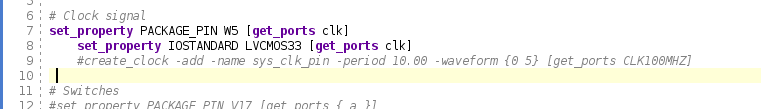


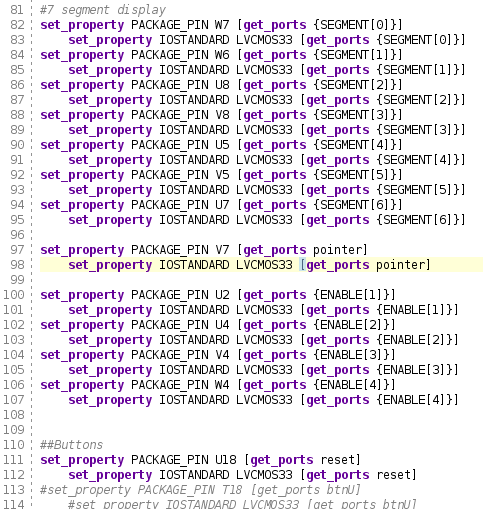
The first part of the file simply set up the module with the inputs necessary, as well as the outputs that would be used to display the counter. The section under //Time Generator was responsible for taking the clock and dividing its cycles into sections. This would allow the clock cycle rate to be slow enough for a human to watch it cycle. Next, the section under //UP counter from 0 to 9 created the numerical cycle that would depend on the previously created slow clock cycle. The numerical cycle is designed to start at 0, cycle through all the integers until it reached 9, then reset to 0 and begin the cycle again.



The second part of single\_digit\_counter.v, pictured above, was the section responsible for displaying the correct number on the 7-pin display. Each case in the switch corresponds a numerical count from the counting circuit to a binary representation of the correct segments to turn on in the 7-pin display. Finally, at the very end of the module, the ENABLE was assigned, and the pointer variable was linked to the rising edge of the clock circuit.

Next came the constraint file Baysis3\_Master.xdc, shown below. The first part of the constraint file was simply creating a connection between the clock on the FPGA and the clock circuit created in the design file. This would allow the clock circuit to use the system clock as a basis for its operation.





The second part of the constraint file was connecting the 7-pin display the registry of values (SEGMENT) from the design file, with the pins W7, W6, U8, V8, U5, V5, U7. Next, the decimal point display was connected to the pointer variable from the design file, so it could display the rising edge of the clock circuit. After this, the enable ENABLE registry from the design file was connected to its proper pins. And finally, the reset button was mapped to the reset variable from the design file to allow the number cycle to be reset to 0.

With the files created, the design was synthesized, bitstream generated, and downloaded to the FPGA. Thankfully, it ran successfully on my first attempt, but several of my classmates struggled to complete this lab. It seems it is very easy to make typos when creating the constraint files, so several of them spent extra time looking through their work. This is one of the reasons I think this lab was an excellent experience. Not only was it my first attempt at writing my own files to load onto the FPGA, but it also taught me the importance of doing my work carefully, to avoid errors. Overall, this lab was an excellent experience, and I can’t wait until the next lab to expand this counter to a 2-digit counter.