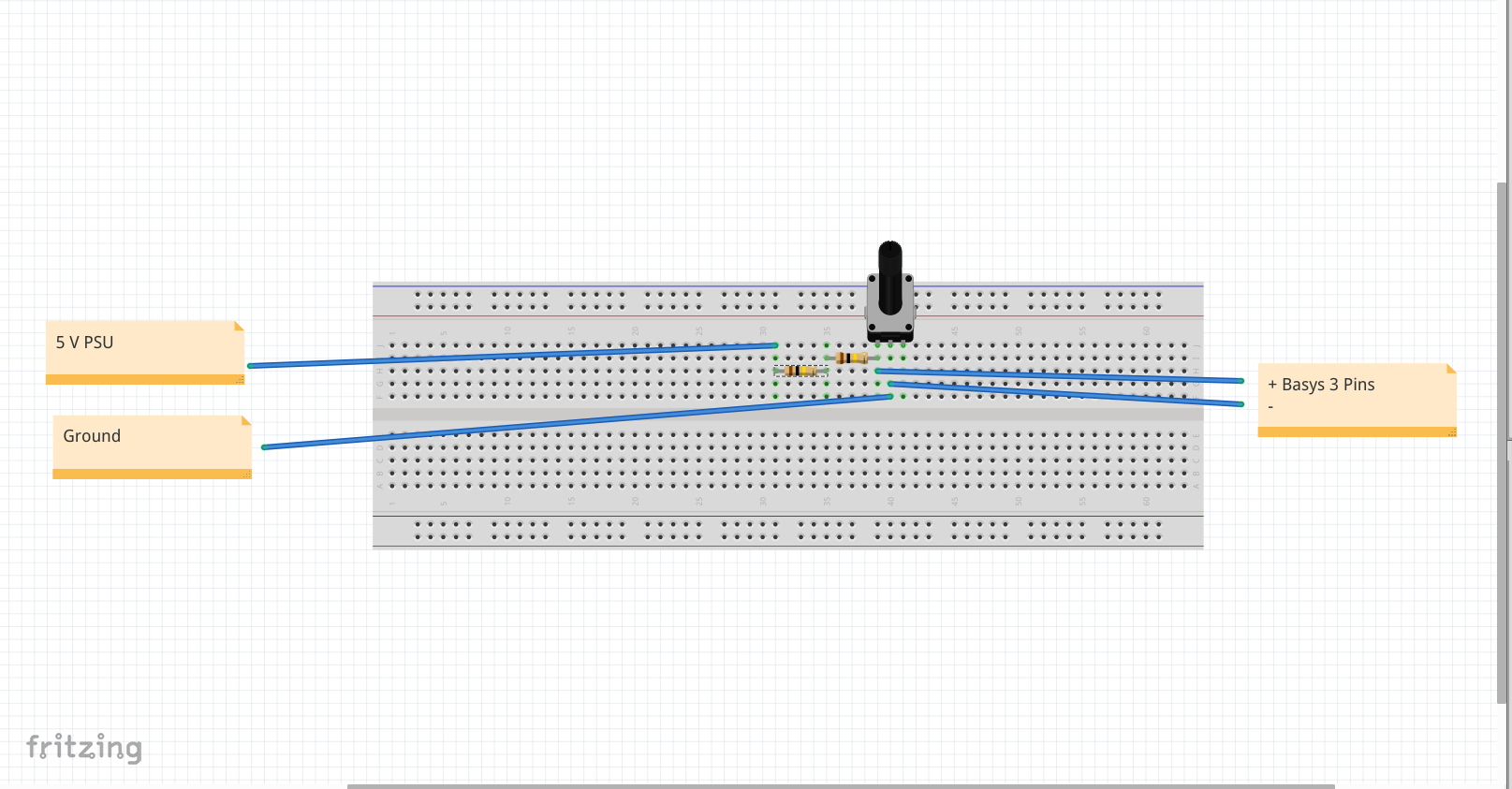
Lab 9 XADC – Analog to Digital Conversion Name: Jacob Hillebrand

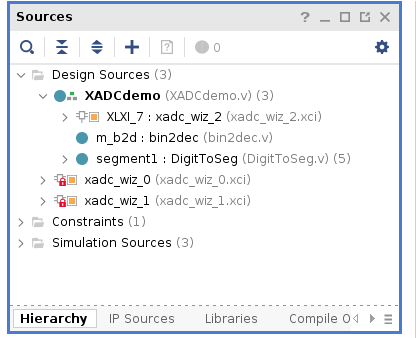
November 5, 2018

In this lab, we expanded our knowledge of the Basys 3 FPGA further by utilizing the onboard XADC components, which can convert an analog voltage input into a usable digital signal. One condition of this component is that it can only handle voltage values between 0 and 1 V, in order to avoid frying the onboard components. To give this input, we used an external circuit on a breadboard. The circuit used a power supply which converted 120 VAC wall power to 5 V DC power. We then passed the 5 V through a circuit with two resistors (2x100k **Ω)** and a potentiometer (0-50k **Ω)** to drop the voltage down so that, at maximum, the potentiometer would create a voltage of 1 V and, at minimum, a voltage of 0V. This final voltage was what we could pass to the XADC module on the FPGA.

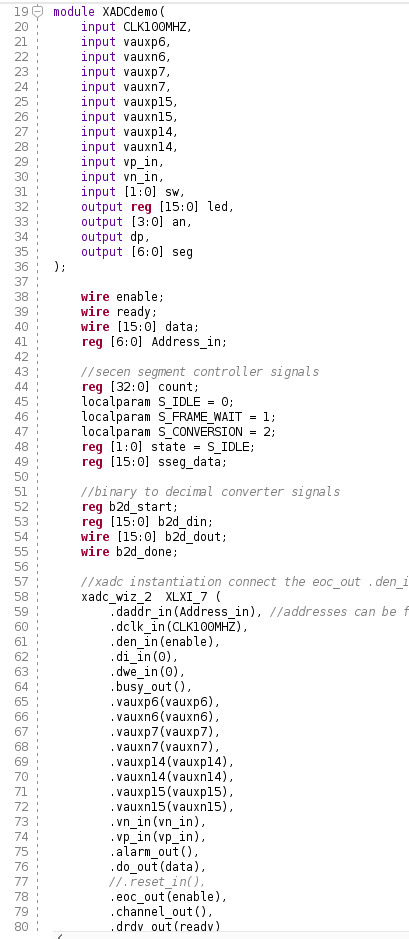
A diagram of the breadboard circuit is shown here



As for the Vivado design, the file hierarchy looked like this



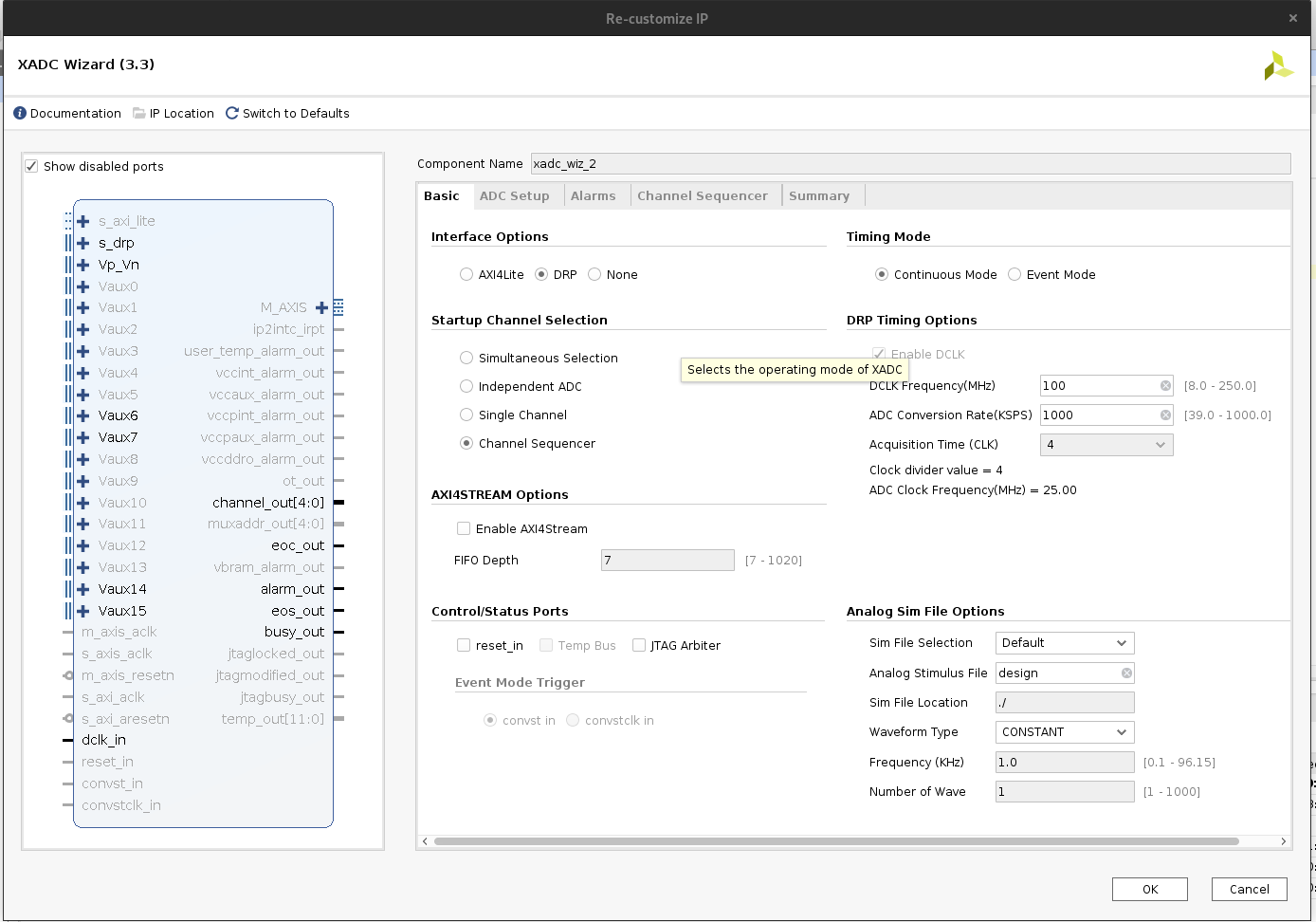
The XADCdemo.v file, shown below, was the top level file for the project.



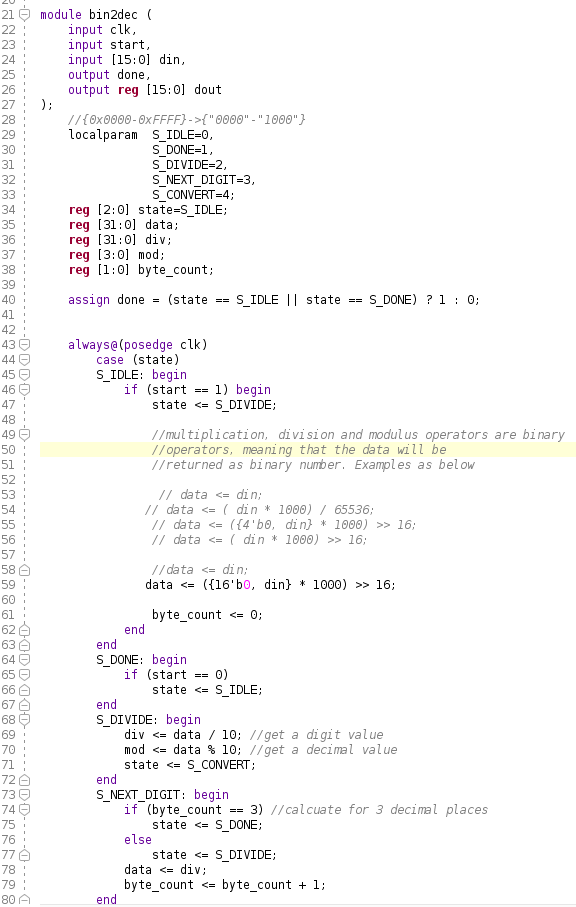


This file was responsible for taking in all of the inputs from the FPGA components, and passing them off to the various necessary functions. However, a few key functions were performed right in the XADCdemo module, for example, you can clearly see that around line 90, the number of leds to be turned on was decided, and about 20 lines later, the binary to decimal conversion process was carried out.

However, much of the work was done by other files in the hierarchy. For example, around line 58, you can see the XADCdemo call the xadc\_wiz\_2 IP, shown below. This IP was responsible for the functionality of the onboard XADC component, and ensured that the 0 – 1 V that our physical circuit passed in would be properly read and interpreted by the FPGA. The pin inputs (with analog values of DC Voltage) would be passed into the IP, and the IP would return an array of digital data.

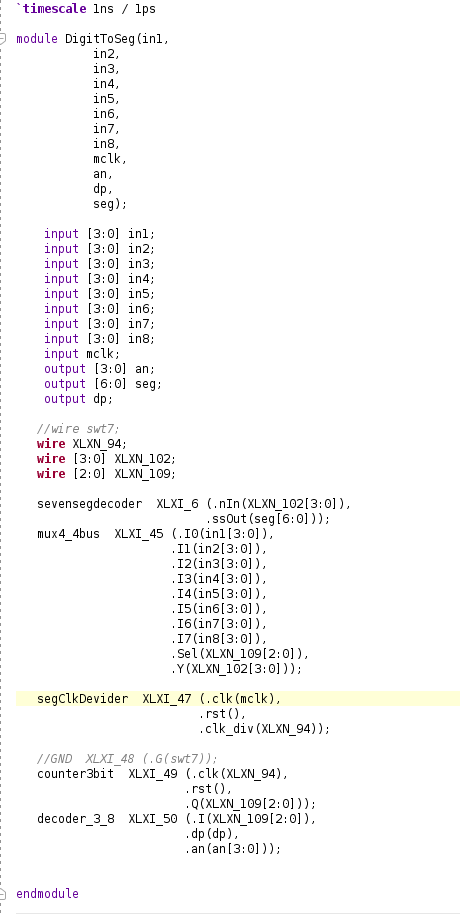


This digital data could then be passed back into the parent module XADCdemo, where it could be interpreted. The binary could be directly used to decide how many leds to turn on on the FPGA, but it had to be passed into a converter to display the proper decimal value of the voltage on the 7-segment display. This is where the bin2dec module came in (bin2dec.v), shown below.



This module took in the binary value given by the XADC\_wiz IP, and converted it to a decimal number. However, this module took things one step further, and also standardized all the numerical values coming in so they were proportionally between 0 – 1, thus giving a very accurate voltage reading on the FPGA’s 7-segment display. As a result, the FPGA could display the voltage to 3 decimal places.

Finally, this numerical decimal value could be passed off to the DigitToSeg module (DigitToSeg.v), shown below



This module filled in the last piece of the puzzle by taking the decimal value provided by the bin2dec module and appropriately displaying the digits on the 7-segment display. It looks like this module comes straight from Xilinx, so it is helpful to know the module is officially supported.

All in all, this was an incredibly informational lab, despite the fact that there were a plethora of hiccups along the way. The first issue I ran into along the way was a faulty multimeter, which caused several issues in trying to test my circuit. Additionally, I had trouble determining which pins to use on the potentiometer, but was able to figure it out after the issue with the multimeter had been resolved using a simple resistance test. Finally, I had two distinct issues with my code. In the XADCdemo module, I had to trial/error different inputs to the XADC\_wiz module before I found the set of inputs that worked, and in the bin2dec module, I had to do more trial/error to determine how to accurately calculate the value for the seven-segment display. In the end, though, I was able to get a working design, and learned much about using the onboard XADC with the Basys 3 FPAG.

