Robot Arm Control Final Report

Nathan Ruetten

Eva Dinelli

5/4/13

*Overview*

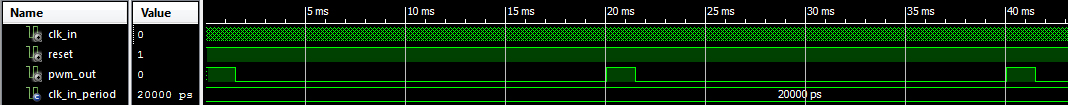
In the Robot Arm Control lab, we used VHDL to control the position of a robotic arm. This lab refined and taught us more about VHDL such as the use of hex numbers. The arm has two motors, one for panning and the other tilting the arm. The arm is supplied with five volts of DC power. We limited the arm control to between 45 and 135 degrees, so the arm would not damage itself by slamming into the table. We didn’t go beyond the call of duty and went for simply refining our VHDL.

*Design Plan*

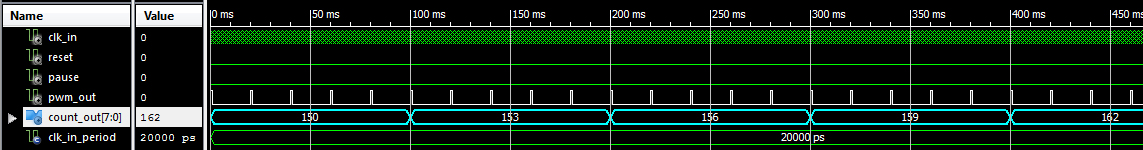
As always, to run any this through this language, there was a clock divider. It made all of the “checks” work together. The mode generator is how we control the arm. Depending on the combination of the switches, the mode generator interfaces with other comports to produce a position for the arm. The counters loads and passes the correct vector to the PWM. Its primary purpose is to split the pan and tilt data, whether to pass the inputted data, and know if the arm is already at the requested position. The PWMs arm to output the data to the arm and feed back to the counter the arm has been moved. There is one more component that is needed for when both the pan and tilt are requested at the same time. We called it D\_swep. Its job is to count till the pan finishes its sweep, and then allow tilt to sweep. We also have a debouncer to cleanly reset all the modules other than the clock divider.

*Major Simulation Results*

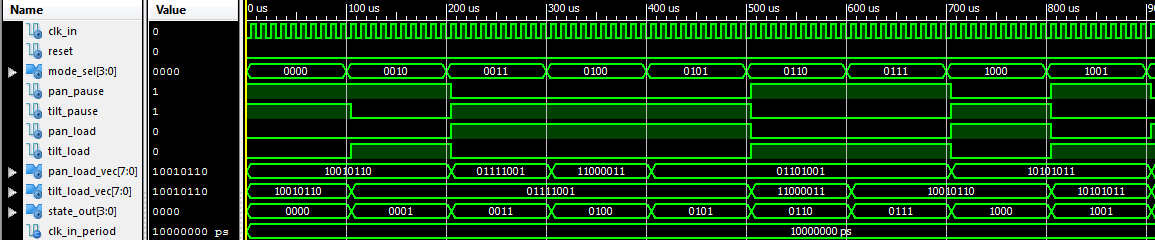
Combination for week 1(clk\_div & pwm)



Combination for week 2(clk\_div, counter, & pwm)



Mode Generator



*Problems*

We had issues from the start but most of these for week one was refreshing VHDL. We had some major problems with the counter. The issues involved understanding hex numbers. We added an extra library and had some of the cases sent from the mode generator throw off the counter. We had a little problem with our Dual Sweep. It won’t tilt after panning. Instead, it would go to a position. We also had a little confusion how the mode generator was supposed to be done. The lecture in class for the rotary encoder threw us behind a little. We also had a few speed bumps syncing the program to the board. We had to check that everything was located in the correct places.

*Conclusion*

From this lab we know how to control a robotic arm using VHDL. The knowledge from this lab can be carried on to other big operating systems that require multiple robotic inputs and outputs such as heavy construction equipment and factory robots. We were successful through dividing the work load, troubleshooting each other’s work, simulating every module to make sure it worked correctly. From simulating and applying practical digital design, we were able to control the robotic arm within the specifications needed.

Everything seems pretty good. I fixed a few spelling mistakes but that’s about it. It might be worth considering splitting up the overview into the objective and design plan. The conclusion should explain that the lab was successful because of our results and application of VHDL with practical digital design. Maybe relate it back to the objective as well

*Block Diagram*