## Abstract:

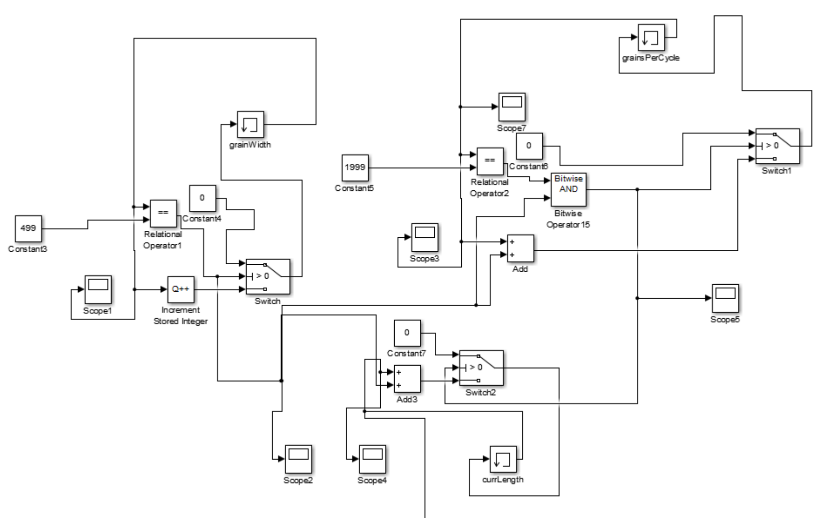
This lab aims to bring together the digital logic that we have been using through Simulink in MATLAB, the FPGA that is on the Zedboard, and the robotic arm that we aim to control. The Robotic arm has 5 servo motors that control different joints. The goal of this lab is to develop a Simulink digital design that when written to the FPGA on the Zedboard, we would be able to control the robotic arm using the buttons and the switches located on the Zedboard. In order to do that we must fully understand and be able to implement pulse width modulation in the digital design. This allows us to fully master the design of digital logic and allows us to see the power that an FPGA can have in getting jobs done. Also we get to have a little fun while playing with a robot.


## Materials:

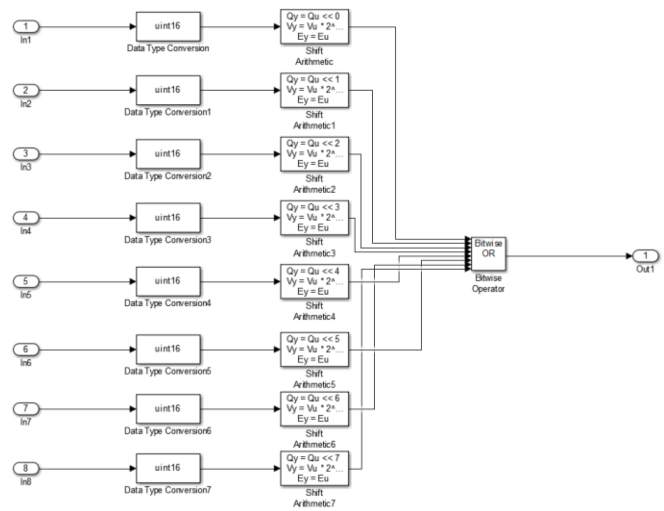
- Zedboard
- Micro USB to USB cord
- 4 GB SD card
- Windows computer running Simulink on the software MATLAB
- Robotic arm
- Daughter board to connect to Zedboard

Procedure:

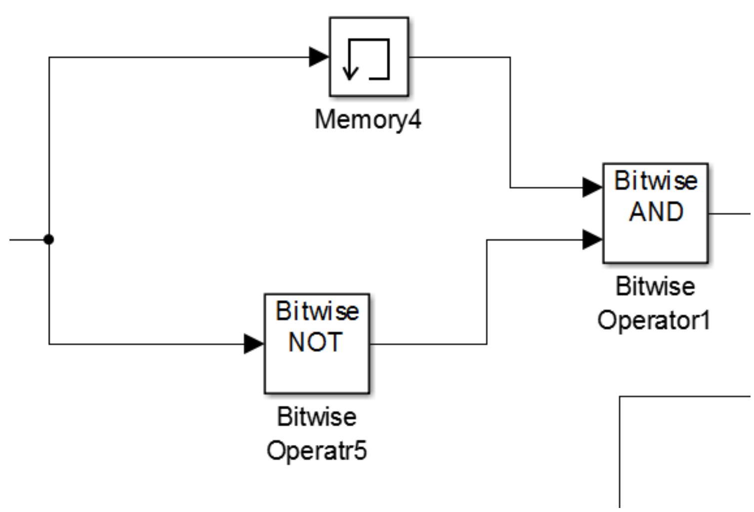
In this lab we are using the Programmable Logic side of the board, or digital logic, to control the robotic arm. In previous labs we used a C program to control the motion of the arm but now we are moving on to control it completely through logic. In order to this we reused and reworked different programs that we had designed in the previous labs. In order to get the robot to work we need a frequency of 50 MHz. We did achieve that in a previous lab when we made a counter that could be sped up and slowed down with the buttons on the Zedboard. We used that same design again to generate the same frequency. Besides using this previously developed program to change the frequency we also want to generate a pulse width of 10 microsecond resolution. We also developed this in a



previous lab so we just had to again implement this program and work it into our new logic. The goal result which we achieved is a pulse with a frequency of 50 MHz and a width of 10 microsecond resolution. This value is fed into a bit of logic that we designed to generate the Pulse Width Modulation. This value we call the current length.

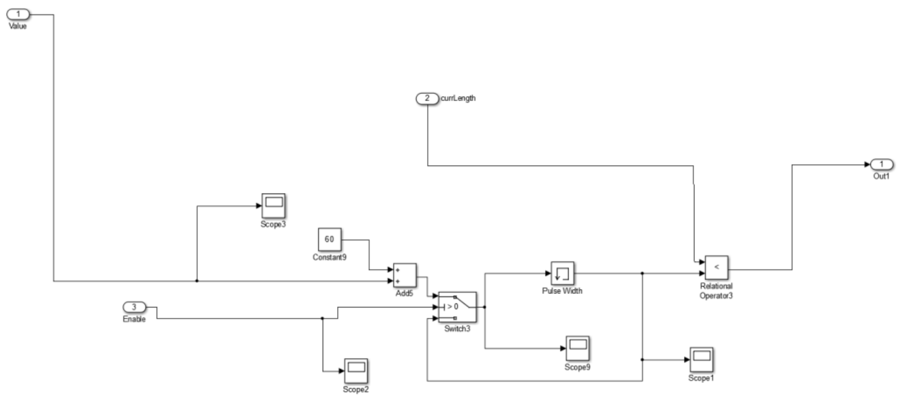



We after creating the logic that makes the correct frequency we now aim to create logic that will allow us to change the pulse width with the switches that are located on the Zedboard. The way that we have it set up is that there are eight switches that are available on the Zedboard. These switches represent an eight bit integer. The left most switch is the most significant bit and the right most switch represents the least significant bit. We made it so that the logic converts this number to a range from 600 to 2400 microseconds which is the input that the robotic arm takes in as the angle in which the motor is at. The logic that we developed allows the user to put in an eight bit number that corresponds to the angle in degrees of what the motor is at. So if you want a specific motor to go to 90 degrees the binary number we set the switches as would have to be 90 in eight bit binary form. This logic converts the eight bit integer that we input through the switches into a decimal number , which can be read by the PWM generator. We feed this into the PWM generator as the value.




The buttons on the Zedboard we used as the enable for each of the servo motors. Each motor was enabled by a different button. So the way that it worked is that the user sets the angle at which they want the motor to go to and then following the setting of the value they would push the button of the motor that they want to move to that specified angle. We have logic that goes along with it make sure that the signal sent out from the button is interpreted as on when pushed down and off when it is no longer pressed. This logic is located on the right hand side of the page This is fed into the PWM generator as the enable. This logic is repeated 5 times or one for each button. In our design we had the center button control the base of the robot, the bottom button control the bicep, the left button to control the elbow, the top button to control the wrist and the right button to control the gripper

All three of these preceding logic designs, the value, the current length and the enable, all are fed into the PWM generator that allows the robotic arm to understand the information that is coming into it to perform the action designated by the user. It brings together allow the values generated




throughout the design and allows them to come together in a form that can be understood and performed by the robot. The PWM is repeated five different times, one for each of the servo motors located on the robot. Each motor has its own PWM generator so that the motors can all be controlled individually.