**Explain the process of developing your alu.circ. How did you get started? What is the sequence of steps you took to complete it?**

First we started by individually creating all the separate circuits for every operation that could be performed. We used tunnels for inputs and outputs of each circuit. The outputs were then connected by a tunnel to the multiplexer to its respective input pin. All of the ALU flags were determined by checking to see if the result coming from the multiplexer was equal to zero (Z flag), less than zero (N flag), if there was a carry out from the adder (C flag) and if there was an overflow if the inputs met the certain conditions necessary (V flag).

**Explain the process of developing your alu-control.circ. How did you get started? What is the sequence of steps you took to complete it?**

Used pencil and paper to come up with an algorithm for the ALUControl. Checked whether each of the inputs was a certain value to determine which input (op, cmd(4 middle bits of funct), sh) would determine the instruction to be performed on the input to the ALU. Used truth tables to try and determine what the output of the ALUControl would be.

**Explain the difference between overflow and carry for two's complement adding and subtracting.**

The overflow flag indicates that the sign bit has been changed during adding or subtracting operations but carry flag means adding or subtracting two registers has a carry or borrow bit. Also, the overflow flag doesn't really tell you anything when working with unsigned numbers but it can alert you of errors when working with signed numbers and vice versa with the carry flag.

**Explain how your implementation of alu-control.circ works. What approaches did you use to implement all the different cases?**

Used comparators to check whether op, funct, sh were something then the control would be a certain output. Wanted to use a multiplexer to hook up all the different operations that could be performed. We were not able to fully get the ALUControl to function correctly.

**Debugging story.**
**a. Describe a specific time when you observed an unexpected behavior of your circuit. Say what the observed behavior was and the expected behavior.**

When trying to make the asr part of the alu circuit work I kept getting the wrong result. It was returning a signed number when it should have been unsigned.

**b. For the observation above, what specific steps did you take to investigate the cause?**

Went and looked at the asr test circuit to see what was happening to the input when being shifted. Noticed that it was not shifting the number correctly.

**c. What was your attempted fix? This should follow logically from your investigation. What was the result when you tested your circuit again? (maybe it still didn't get the result you wanted, but we want to know what that specific new result was).**

Realized that I was using the logical shift right function of the shifter instead of the arithmetic shift right function of the shifter. After making this change the asr test was passed.