CprE 381: Computer Organization and Assembly-Level Programming, Spring 2019

Project Part 1 Report

Lab Partners _____Sean Gordon_____

_____Abdul-Salam Adedoja_____

_____

Section/Lab Time _____C/10:00_____

Refer to the highlighted language in the project 1 instruction for the context of the following questions.

a. [Part 0] With your project group members, create a list of best practices / tips for designing, compiling, and testing VHDL modules based on your experiences so far with these labs, both working individually and as a group.
   - Distinctly comment code
   - Leave lots of space
   - Separate inputs and outputs
   - Align all assignments to the same point horizontally
   - Break parts up into reusable pieces
   - Do files = fast

b. [Part 1 (a)] Describe the difference between logical (srl) and arithmetic (sra) shifts. Why does MIPS not have a sla instruction?

   Srl shifts right without copying the sign bit, srl shifts right while copying the sign bit. Mips does not have an sla because the sl instruction shifts left towards the sign bit, and the least significant bit has no meaning when pertaining to sign.

c. [Part 1 (b)] In your writeup, briefly describe how your VHDL code implements both the arithmetic and logical shifting operations.

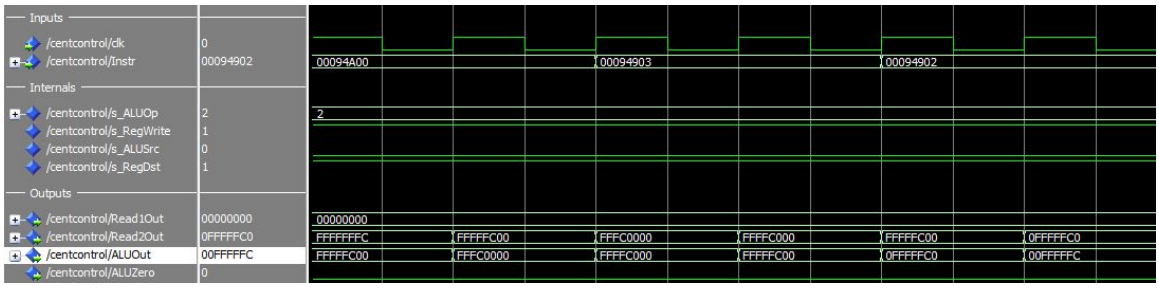
   Designed STRUCTURALLY (Who decided that, btw), the shifter uses cascaded two-to-one multiplexers to shift the incoming value 1, 2, 4, 8, or 16 times depending on the selector signal. Each set of cascaded muxes can be activated in parallel, so the value can be shifted 0-31 bits.

d. [Part 1 (c)] In your writeup, explain how the right barrel shifter from part b) can be enhanced to also support left shifting operations.

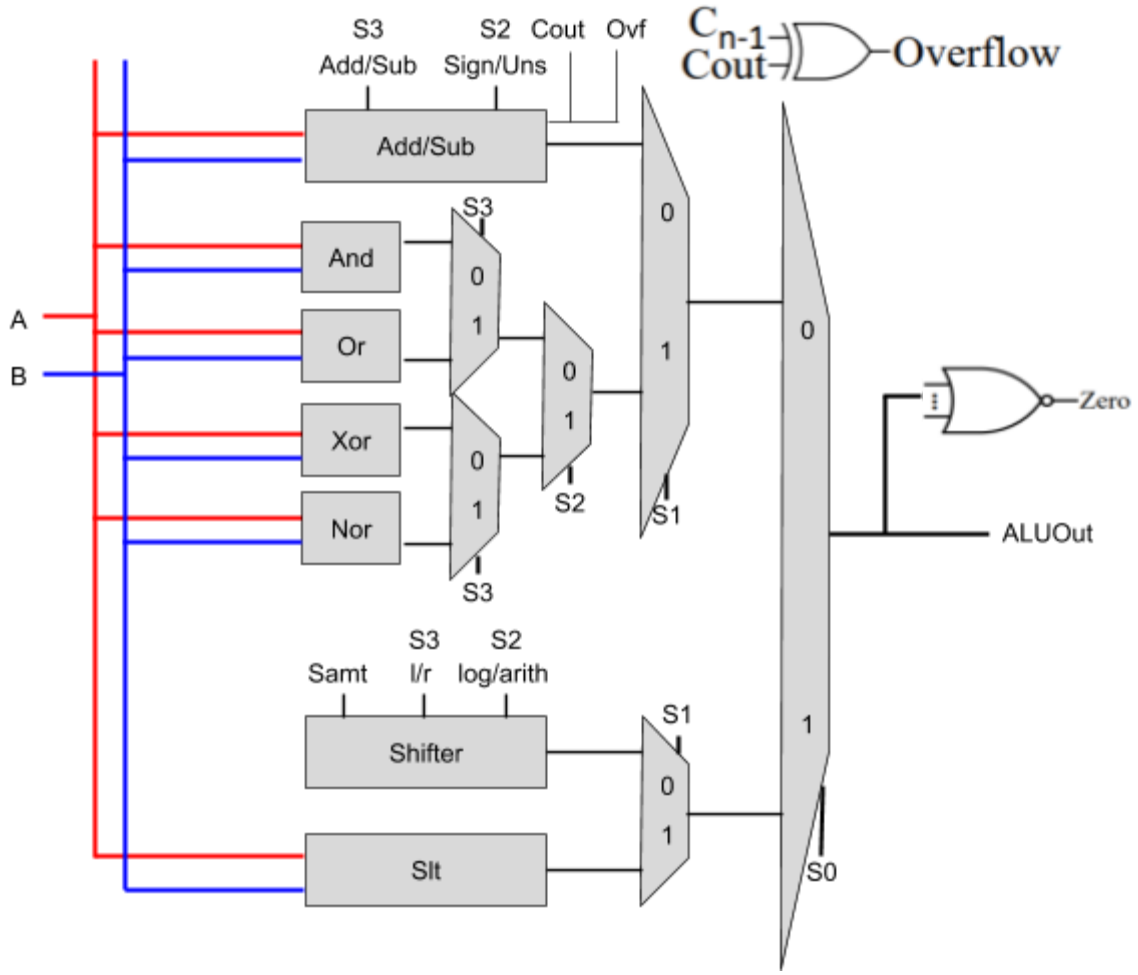
   Flip it inside out so that it places the shifted bits at the front of the signal rather than the end.

e. [Part 1 (d)] Describe how the execution of the different shifting operations corresponds to the Modelsim waveforms in your writeup. Waveforms. □

Shifting the number 0xFFFF_FFFC left 8 bits two times results in the number 0xFFFC_0000
Shifting the number 0xFFFC_0000 right arithmetically 4 bits two times results in the number 0xFFFF_FC00
Shifting the number 0xFFFF_FC00 right logically 4 bits two times results in the number 0x00FF_FFFC



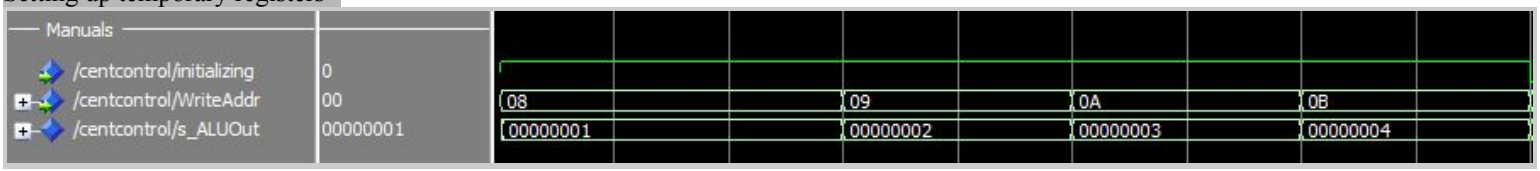
f. [Part 2 (a)] Draw a simplified schematic for this 32-bit ALU. Consider the following questions: how is Overflow calculated? How is Zero calculated? How is slt implemented?



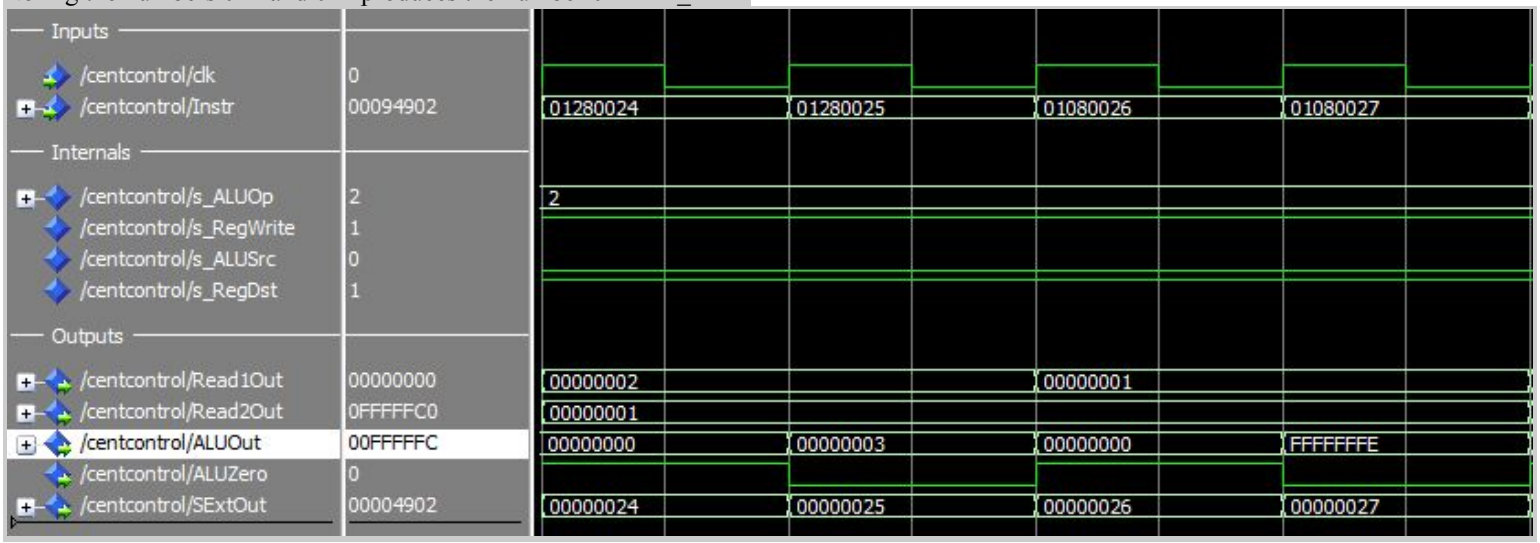
Overflow is calculated by taking the most significant bit and xoring it with the carry out.
Zero is calculated by checking if the ALU output is all zeros with an or gate.
SLT is implemented by subtracting the two numbers in question, and comparing the carry out flag with 0 to find unsigned < or >.

g. [Part 2 (c)] Describe how the execution of the different operations corresponds to the Modelsim waveforms in your writeup .Waveforms.

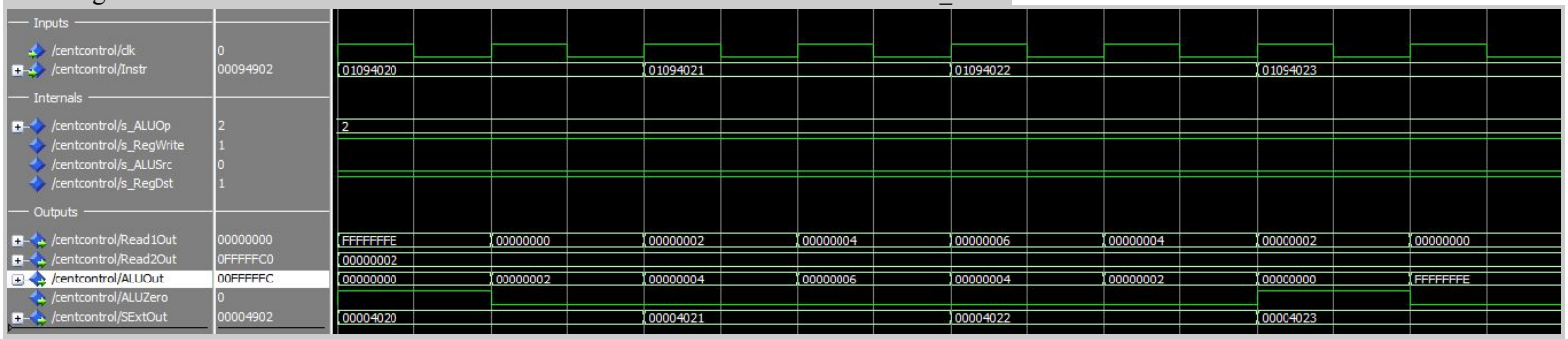
Setting up temporary registers -



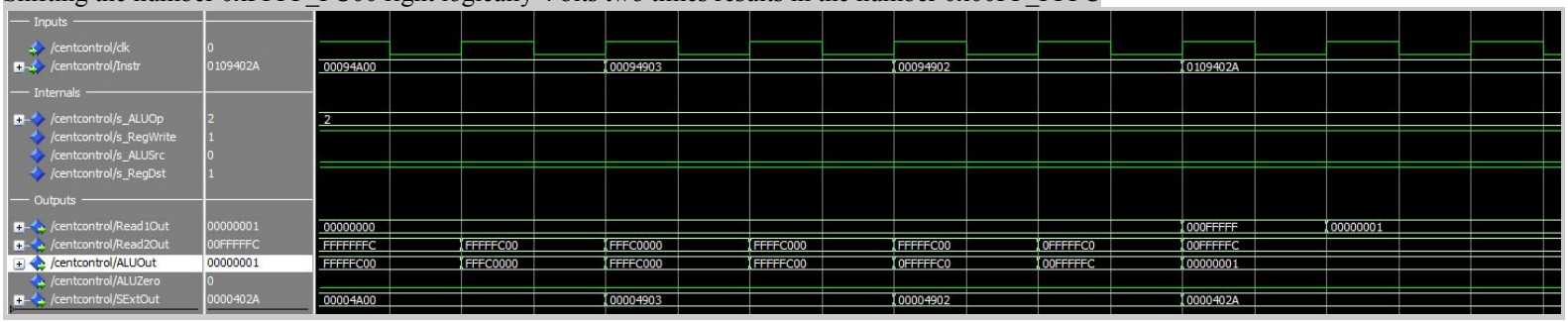
Anding the numbers 0x1 and 0x2 produces the number 0x0
Oring the numbers 0x1 and 0x2 produces the number 0x3
Xoring the numbers 0x1 and 0x1 produces the number 0x0
Noring the numbers 0x1 and 0x1 produces the number 0xFFFF_FFFE



Adding the number 0x2 to the number 0xFFFF_FFFE two times results in the number 0x2
AddU-ing the number 0x2 to the number 0x2 two times results in the number 0x6
Subtracting the number 0x2 from the number 0x6 two times results in the number 0x2
SubU-ing the number 0x2 from the number 0x2 two times results in the number 0xFFFF_FFFE



Shifting the number 0xFFFF_FFFC left 8 bits two times results in the number 0xFFFC_0000
Shifting the number 0xFFFC_0000 right arithmetically 4 bits two times results in the number 0xFFFF_FC00
Shifting the number 0xFFFF_FC00 right logically 4 bits two times results in the number 0x00FF_FFFC



h. [Part 3(b)] justify why your test plan is comprehensive. Include waveforms that demonstrate your test programs functioning.

My test plan is comprehensive because it tests each of the signals in a suitable situation, similar to how they would be used in a program.

i. [Feedback] You must complete this section for your lab to be graded. This will be turned in INDIVIDUALLY in a separate assignment called "Proj 1 Feedback". Write down the first response you think of; I expect it to take roughly 5 minutes (do not take more than 10 minutes).

1. How many hours did you spend on this lab?

Task	During lab time	Outside of lab time
Reading lab		
Pencil/paper design		
VHDL design		
Assembly coding		
Simulation		
Debugging		
Report writing		
Other:		
Total		

2. What fraction of the above time did you spend working with your partner (either in a lab or remotely)?

3. If you could change one thing about the lab experience, what would it be? Why?

4. What was the most interesting part of the lab?