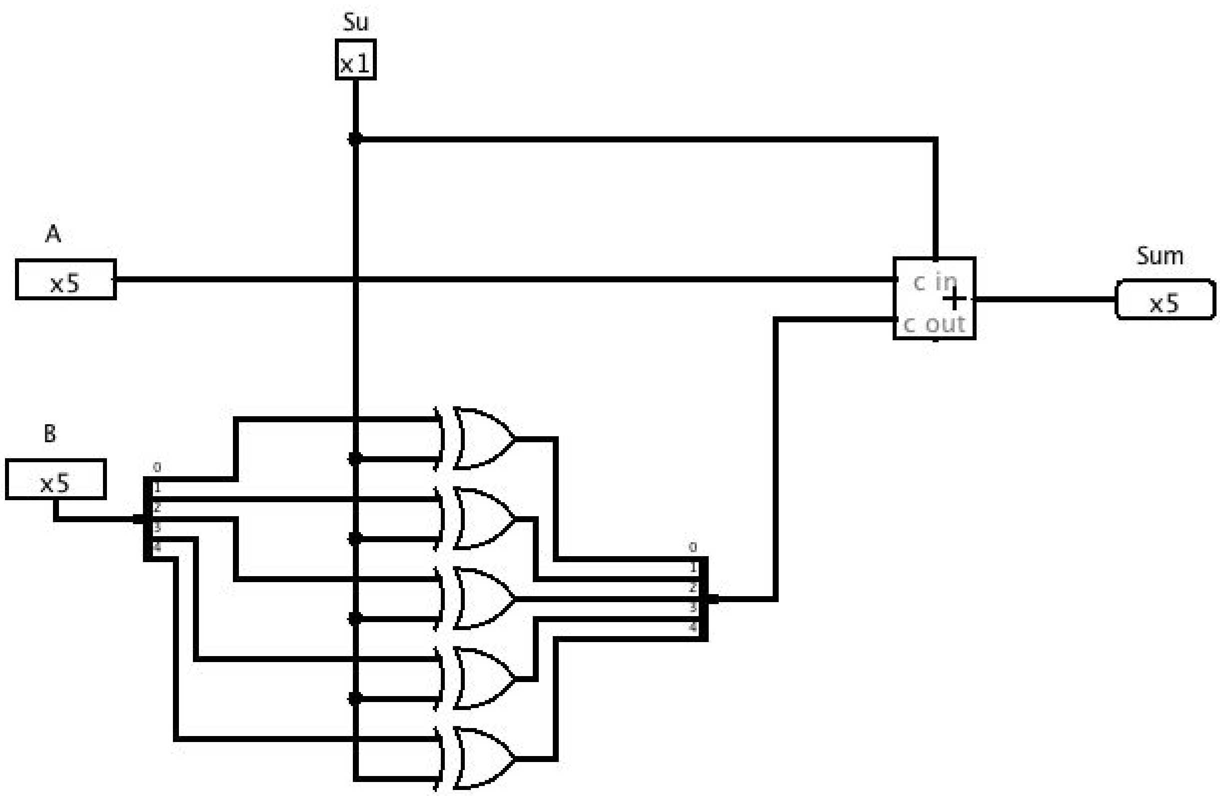
William "Chris" Fenton
CSF - Digital Logic
10/22/2015
Lab 4

Part 1

1. Build a 5-bit adder-subtracter using the Logisim adder along with a negation circuit on the subtrahend as shown in last figure of Section 4.3 of the text. Do not build the Borrow and Carry logic. The *inv* block is a Controlled Bitwise Not circuit. Do not use the Logisim Subtracter! Just use the adder.



2. Why does the *Sub* signal go into the *Cin* of the adder?

After Inverting, we need to add one bit in twos complement. So when the control is active for subtraction, we add one into the carry-in input of the adder.

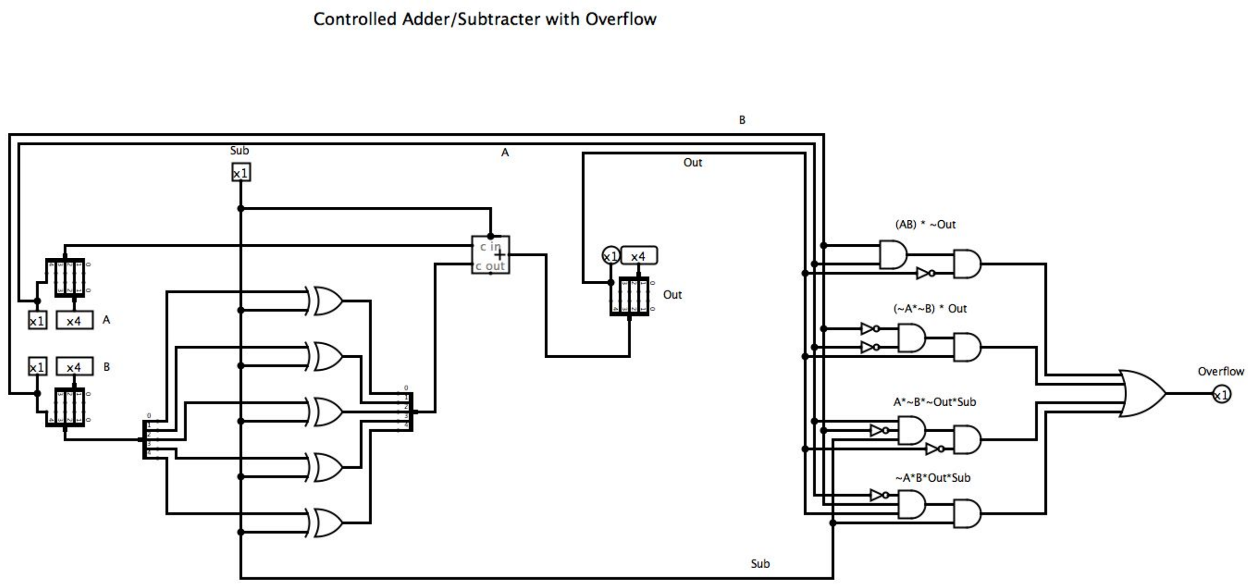
3. Build a function table that defines the Sum output of your adder-subtracter in terms of the A and B inputs. Assume all numbers are 5-bit signed numbers in two's complement representation.

A	B	Sub	Sum
x	y	false	if (x + y > 2^5 - 1) then x + y - 2^5
			else if (x + y < -2^5) then x + y + 2^5
			else x + y
x	y	true	if (x + y > 2^5 - 1) then x - y - 2^5
			else if (x + y < -2^5) then x - y + 2^5
			else x - y

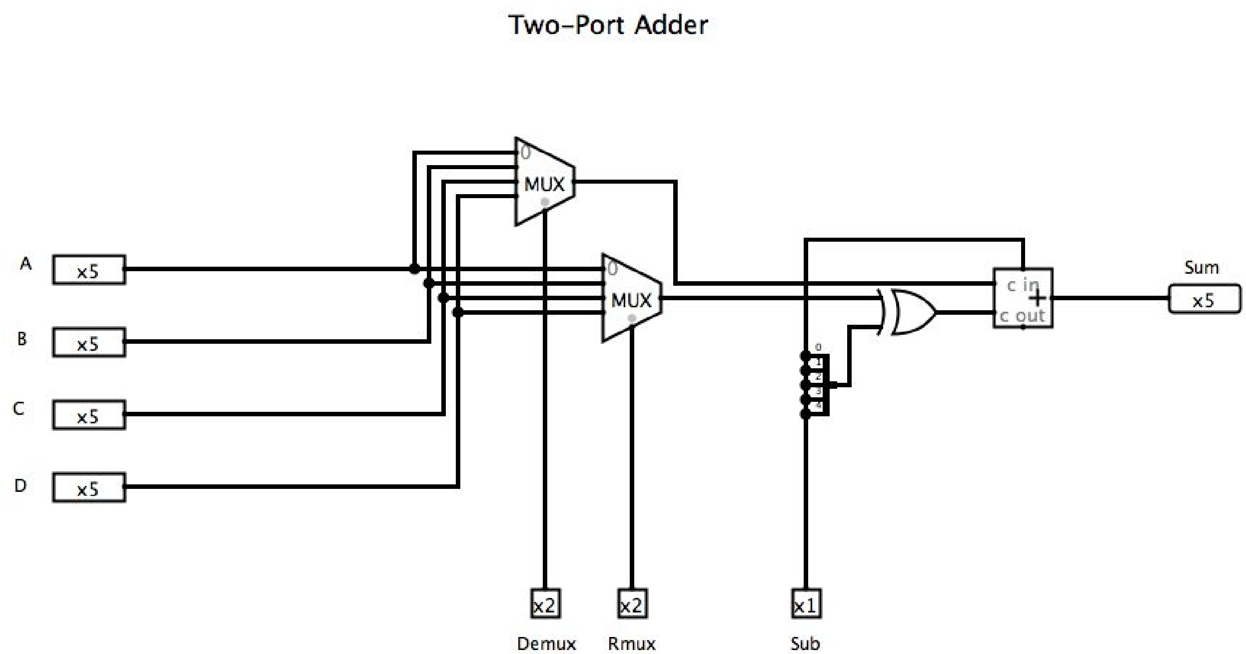
4. Give a logic table for the overflow signal *Ovfl* in terms of the sign bits of the adder operands *signA, signB*, the *Sub* control signal, and the sign bit of the adder output, *signOut*. Recall that overflow for addition in signed two's complement representation occurs when the two operands to the adder are the same sign (either both positive or both negative) and the result is the opposite sign. If the *Sub* signal is zero, then the overflow logic is based on the the signs of the two operands and the sign of the adder output. If the *Sub* signal is one, indicating a subtract, then the overflow logic is based on the sign of the A input and the opposite of the sign of the B input (because of the negation that will occur), along with the sign of the output of the adder. Note that the negation circuit itself can itself cause an overflow in case the argument to be negated is -8. UGH. See if you can figure out the logic for also detecting an overflow in the special case of attempting to subtract -8 (eg, x - (-8) for any x).

signA	signB	signOut	Sub	Ovfl
1	1	1	0	0
1	1	0	0	1
1	0	1	0	0
1	0	0	0	0
0	1	1	0	0
0	1	0	0	0
0	0	1	0	1
0	0	0	0	0
1	1	1	1	0
1	1	0	1	0
1	0	1	1	0
1	0	0	1	1
0	1	1	1	1
0	1	0	1	0
0	0	1	1	0
0	0	0	1	0

5. Split out the sign bits from the two inputs and the final output and along with the *Sub* signal build your overflow logic circuit from the logic you designed in the previous step. Be sure to test your logic!



Part 2



What is the purpose of the Xor gate?

To combine addition and subtraction into one circuit.

What, specifically, is the wiring of the splitter near the Xor gate?
The control code for the circuit is indicated by pins at the bottom of the circuit. How many control bits are there?

A bundler. 1bit

How many operations can the circuit perform?

3

Describe in one or two sentences what operations can be performed on the inputs A,B,C,D by the circuit.
(Optional) The Two-port adder has the weakness that you cannot move results from one register to another, and this action is sometimes needed. Can you think of a simple way to modify the circuit to allow transfers from one register to another? If so, explain how.

Make the output of the adder feedback into the registers.