William “Chris” Fenton
CSF - Digital Logic
HW4

1. Do a binary addition by hand of the 4-bit binary numbers 0110 and 0011. Express the same addition in decimal.

```
0110
0011
----
1001 -> 9 base10
```

2. For this problem assume signed bounded integers in two’s complement representation in 4 bits. Do a binary addition by hand of 1110 and 0111. Express the same addition in decimal. Does this addition result in an overflow? Explain your answer.

```
1110
0111
----
0101 -> 5 base10
```

No overflow (neg + pos number). There was a carry bit, but we drop that in two’s complement

3. When an overflow condition occurs in an adder working on bounded integers, is the result on the output bus arithmetically correct? What should the processor do when an overflow occurs? (Note - the question about what a processor should do is a provocative thought question not covered by the chapter material).

No, the result is not correct. The processor should at least tell you if there is an overflow error. We can check this by looking at the signs of the numbers and their result. If two negative numbers produce a sum that is positive, we know we have a problem. Likewise, if two positive numbers produce a sum that is negative, we have an overflow. Adding a positive and a negative number will not produce an overflow because the sum will always move closer to zero (the signs tend to work against each other).

4. For this problem assume signed bounded integers in two’s complement representation in 8 bits. Do a hex addition by hand of 0xBA and 0x7F. Express the same addition in decimal. Does this addition result in an overflow? Explain your answer.

0x BA
0x 7F

0x 39 -> 57 base10

There is no overflow because you are adding a negative number (BA) with a positive number (7F). This does result in a carry bit, however.

5. For this problem assume signed bounded integers in two’s complement representation using 4 bits. Do a binary addition by hand of 1110 and 1001. Express the answer in decimal. Does this addition result in an overflow? Explain your answer.

1110 -> -2 base 10
1001 -> -7 base 10

0111 -> 7 base 10

Yes, this results in an overflow. The sum of two negatives numbers should be negative, yet the sum is positive in our case. If this was in 5bits, our carry bit would make the number 10111, as negative number in two’s complement.

6. Give a function table for the Sum output of the 8-Bit Adder block circuit of Section 4.3 assuming a signed interpretation in two’s complement representation. Feel free to use a simple conditional programming statement to express the function. Do not include the carry out signal in the table because the overflow detection for the signed interpretation is handled in the next exercise.

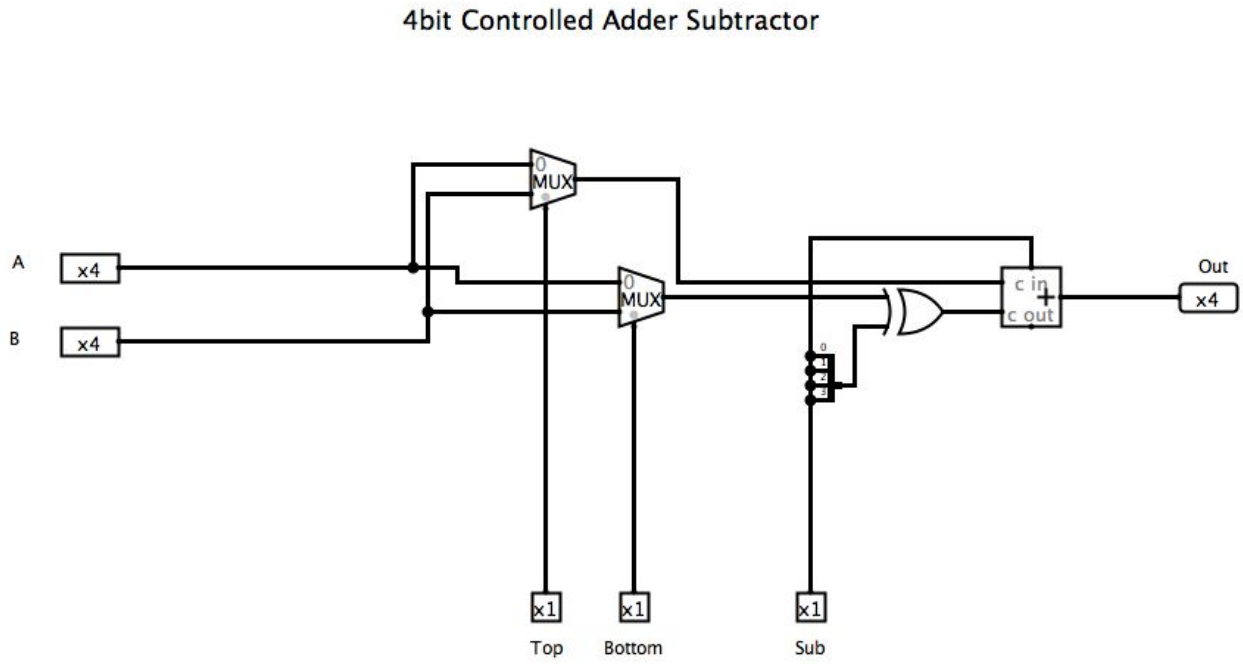
X	**Y**	**SUM**
x	y	if (x + y) > 2^7 -1 then x + y - 2^8
		else if (x + y) < -2^7 then x + y + 2^8
		else x + y

7. For this problem assume signed bounded integers in two’s complement representation. Also assume you are working with the Adder-Subtracter with Negate circuit near the end of Section 4.3. Overflow in addition can occur when two positive numbers are added or two negative numbers are added. Overflow can also occur when the Neg circuit input is the largest negative number. Give a logic table and a logic expression for an overflow signal Ovfl in terms of the sign bits of the two operands Sa, Sb, the sign bit of the output of the negation circuit Snegb, and the Ss sign bit of the Sum output.

sA	sB	sS	sNeg	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

Ovfl = (sA*sB*~sS) + (~sA*~sB*sS) + (sA*~sB*~sS*sNeg) + (~sA*sB*sS*sNeg)

8. Draw a dataflow circuit using block components that lets you add or subtract two input numbers in all possible ways. Your circuit will have two inputs X, Y and one output Out. You'll have an adder and you'll need to use multiplexers on each input to the adder. Clearly specify the number and type of control bits needed.



10. Why is the subtract control signal Sub connected to the carry in Cin signal of the Two's Complement Adder-Subtracter circuit at the end of Section 4.3?

Because you need to add 1 after inverting in two's complement.

11. In Section 4.3 the Adder-Subtracter with Negate is built with separate Neg and Adder blocks. In the same section, shortly after, the Two's Complement Adder-Subtracter uses a Controlled Bitwise Not (Labeled Inv in the figure). How many gates does the latter circuit save over the former circuit?

8. One for each bit's XOR gate.

12. What does the Borrow signal signify in the Two's Complement Adder- Subtracter circuit at the end of Section 4.3? What does it say about the bit pattern representing the Sum?

It signals overflow. When there is overflow, the Sum is not arithmetically correct.