EECS 203 Homework 4 Solutions

3. (5 pts)

(a) 39 = $(27)_{16}$

(b) 21 = $(15)_{16}$

(c) 126 = $(7E)_{16}$

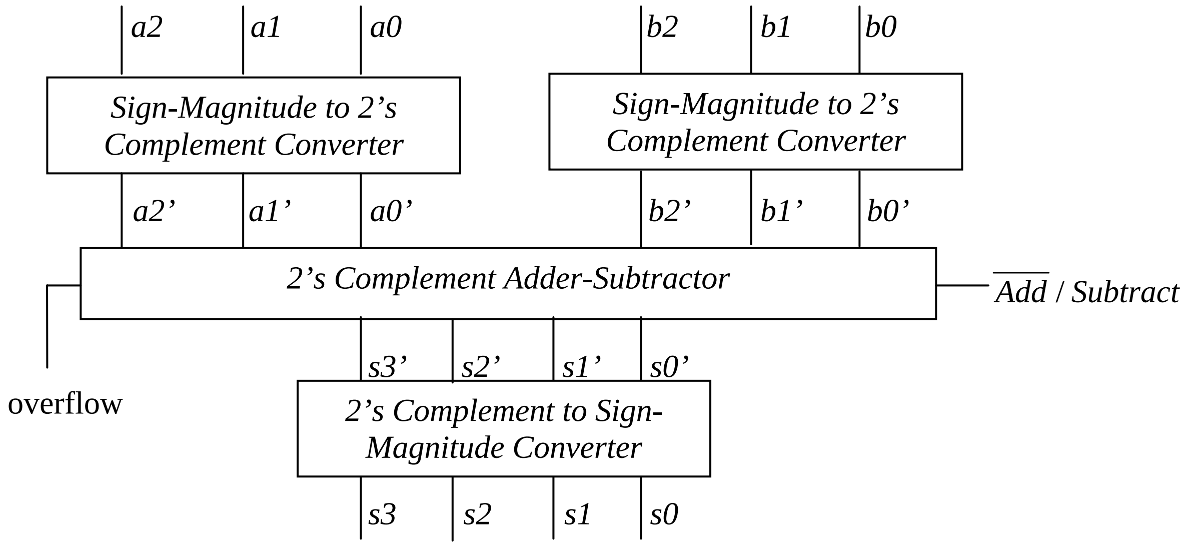
(d) 119 = $(77)_{16}$

4. (5 pts)

Decimal	Sign Magnitude	2's Complement	Sign Mag. (8-bit)	2's Comp. (8-bit)
-12	11100	10100	1000 1100	1111 0100
34	0100010	0100010	0010 0010	0010 0010
-6	1110	1010	1000 0110	1111 1010
7	0111	0111	0000 0111	0000 0111

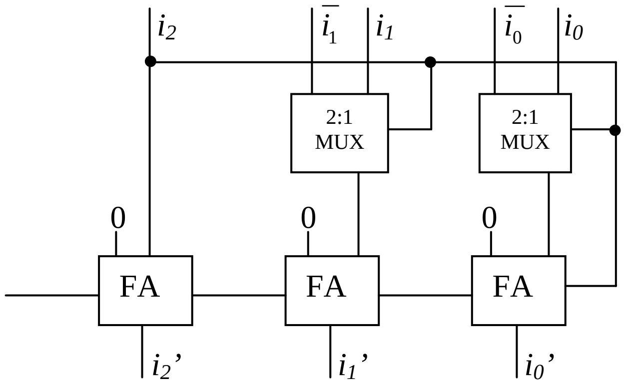
5.

(a) (10 pts)

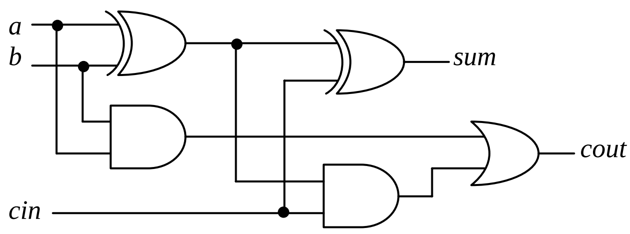


The *2's Complement Adder-Subtractor* is shown in Problem 5(b).

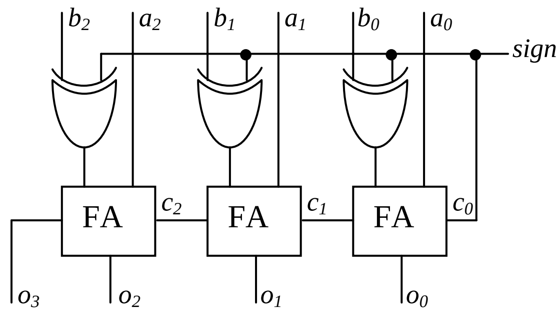
The *Sign-Magnitude to 2's Complement Converter* is same as the *2's Complement to Sign-Magnitude Converter*, and the 3-bit version is shown below. The 4-bit version is just an extension of this with an extra 2:1 MUX and Full-Adder



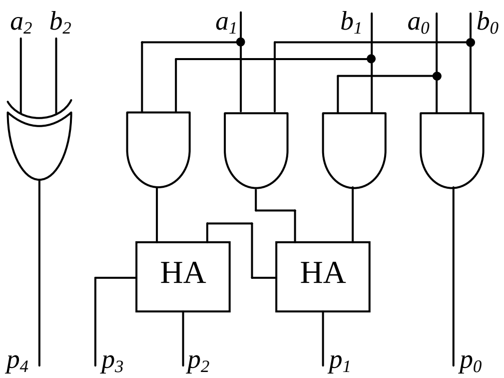
The Full Adder is implemented as shown below. Half Adders could also be used, but for the sake of minimizing different functional pieces in the design, I've chosen to use only Full Adders.



(b) (5 pts)
Shown below is a *3-bit Ripple Carry 2's Complement Adder/Subtractor* with the Full Adder used above.



6) (15 pts)
The sign bit is simply an XOR of the two input sign bits. And 1-bit multiplication in binary is the same as using an AND gate. Hence, using the math shown below on the right, we can implement the following function drawn on the left.

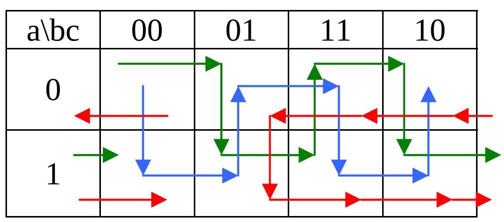


$$
\begin{array}{rrrr}
 & & a_1 & a_0 \\
\times & & b_1 & b_0 \\
\hline
 & & a_1b_0 & a_0b_0 \\
+ & a_1b_1 & a_1b_0 & \\
\hline
p_3 & p_2 & p_1 & p_0
\end{array}
$$

$p_4 = a_2$ XOR b_2

7) (10 pts)
This answer can vary. It may help to use a Karnaugh map. Examples of three possible solutions are shown below with the corresponding Karnaugh map traversal shown on the right.

Decimal	Answer 1	Answer 2	Answer 3
0	000	000	000
1	001	100	010
2	101	101	011
3	111	001	001
4	011	011	101
5	010	111	111
6	110	110	110
7	100	010	100

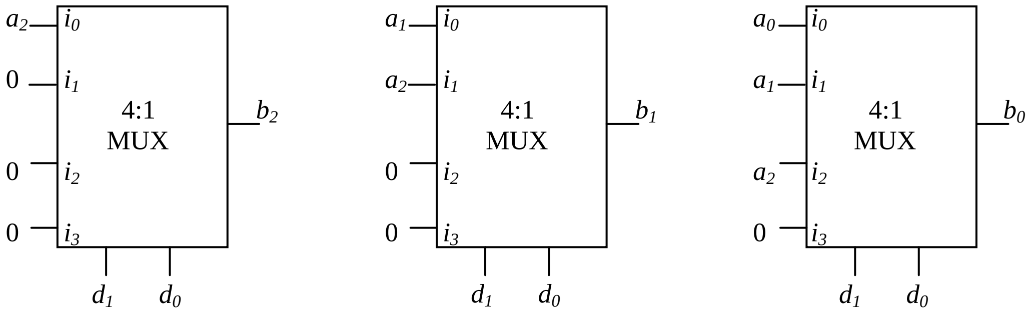


8) (10 pts)

The function of $b = \left\lfloor \frac{a}{2^d} \right\rfloor$ is simply a right-shift of each bit of a. Thus the truth table is shown below.

a	d_1	d_0	2^d	b_2	b_1	b_0
$a_2a_1a_0$	0	0	1	a_2	a_1	a_0
$a_2a_1a_0$	0	1	2	0	a_2	a_1
$a_2a_1a_0$	1	0	4	0	0	a_2
$a_2a_1a_0$	1	1	8	0	0	0

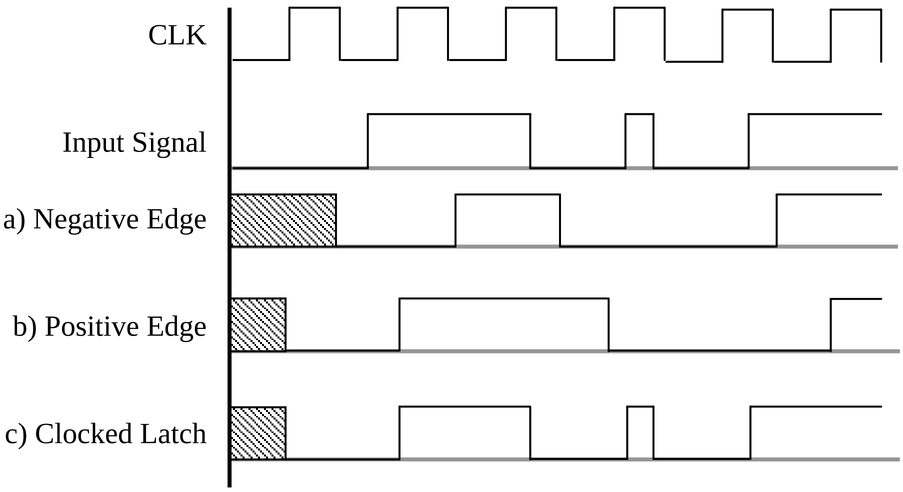
Therefore, the MUX implementation of this is as follows:



9) (10 pts)

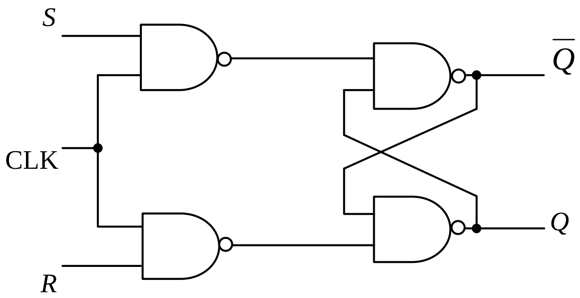
a	b	Old Q	New Q
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	(ignore)
1	0	1	(ignore)
1	1	0	1
1	1	1	1

10) (10 pts)



11) (10 pts)

Other solutions are possible, simply by “bubble-pushing” this solution



CLK	S	R	Q’
0	X	X	Q
1	0	0	Q
1	0	1	1
1	1	0	0
1	1	1	*undef.*

12) (10 pts)