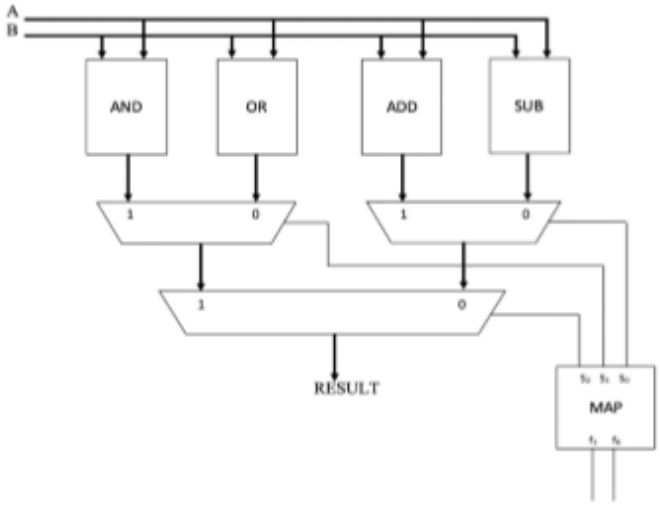
Austin Smothers
Professor Bustamante
CSC 137

Assignment 4

1. Design the minimum logic for the MAP module that selects the ALU operations according to the following table. a) First fill table accounting for any possible "don't cares," b) derive the minimal equations for the select lines s_2, s_1, and s_o. c) draw the logic for s_2, s_1, and s_o using AND/OR/NOT gates.



ALU function	f_1	f_0	s_2	s_1	s_0
ADD	0	0	0	d	1
SUB	0	1	0	d	0
OR	1	0	1	0	d
AND	1	1	1	1	d

b) s2

f1	f0	0	1
0		0	0
1		1	1

$s_2 = f_1$

s1

f1	f0	0	1
0		d	d
1		0	1

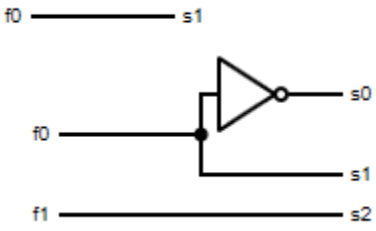
$s_1 = f_0$

s0

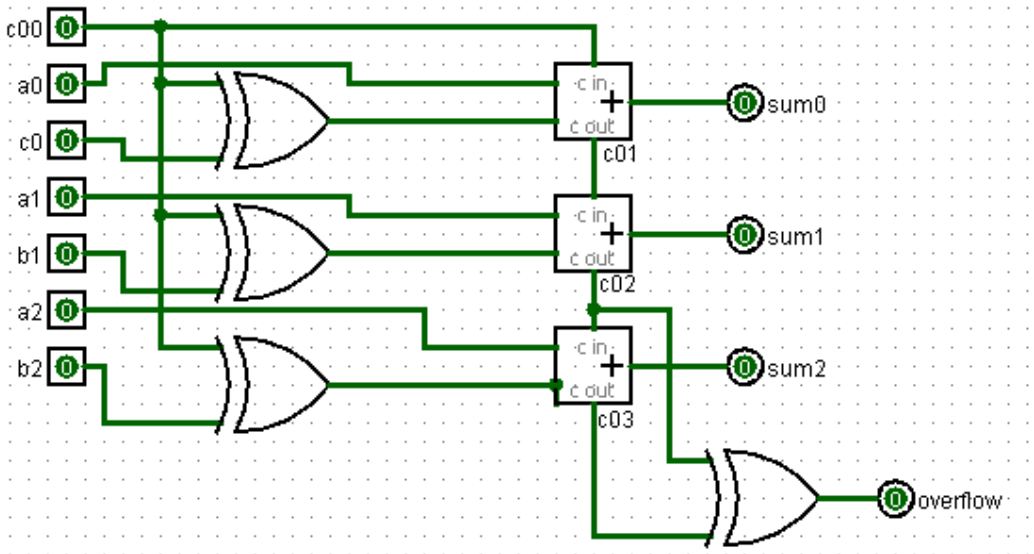
f1	f0	0	1
0		1	0
1		d	d

$s_0 = \bar{f_0}$

c)



2 a) Draw a 3-bit Adder-Subtractor module with Overflow detection with Full Adders (FA) and XOR gates. Use FA symbols and XOR gates in your implementation and label all your wires with numerical sub indexes, i.e. $a_{[0-2]}$, $b_{[0-2]}$, $c_{i[0-2]}$, $c_{o[0-3]}$, $sum_{[0-2]}$,

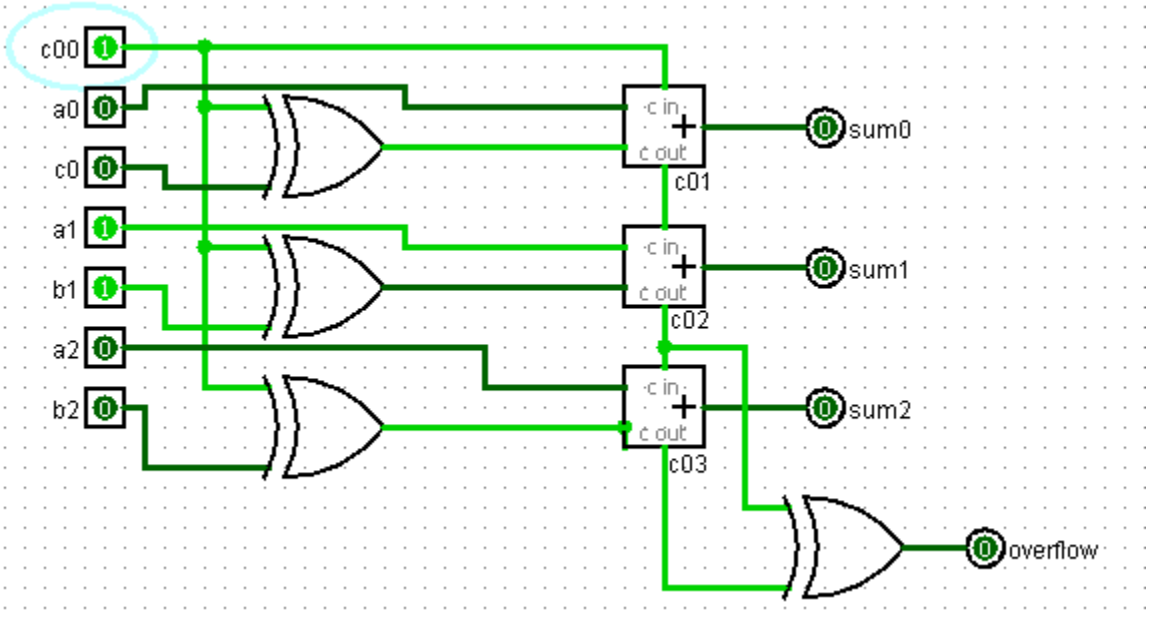


b). Using your 4-bit Adder-Subtractor drawing, show the inputs and outputs values as well as the values of each wire involved when it performs the following operation.

Operation => A - B

where: A = 2 and B = 2

c00	a0	c0	a1	b1	a2	b2	sum0	sum1	sum2	overflow
1	0	0	1	1	0	0	0	0	0	0



3. Perform the Floating-Point Arithmetic Operation. Sum = A + B given that

A = 9.25 and B = 5.625

Assume that A and B are 16-Bit FP numbers consisting of the following:
1-bit sign,
7-bit exponent
bias = 63
8-bit fraction

Use the procedure outlined in page 133-135 of the textbook. Show your work below.

Write the binary representation of A, B and their Sum. The IEEE scientific format is $1.F \times 2^E$

16	8	4	2	1	.5	.25	.125	.0625	.03125
0	1	0	0	1	0	1	0	0	0
0	0	1	1	0	1	0	1	0	0
+					=				
0	1	1	1	1	1	1	1	0	0

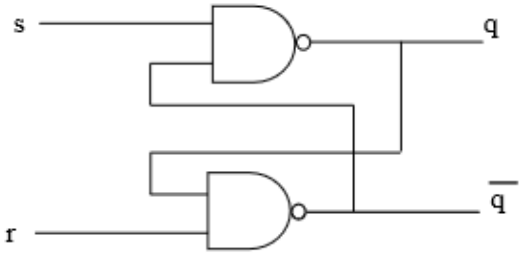
= 15.875

A 1 (s) 1 0 0 0 0 1 0 (Biased Exponent (E)) 0 0 1 0 1 0 0 0 (Fraction(F))

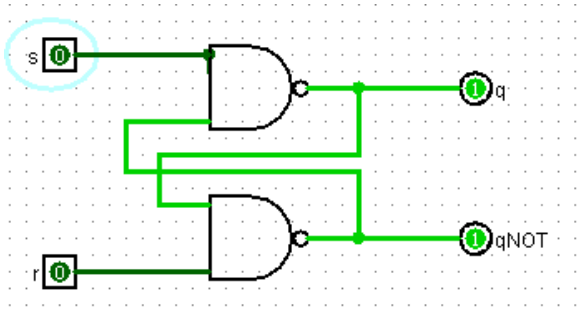
B 1 (s) 1 0 0 0 0 0 1 (Biased Exponent (E)) 1 0 1 0 1 0 0 0 (Fraction(F))

Sum 1 (s) 1 0 0 0 0 1 0 (Biased Exponent (E)) 1 1 1 1 1 1 0 0 (Fraction(F))

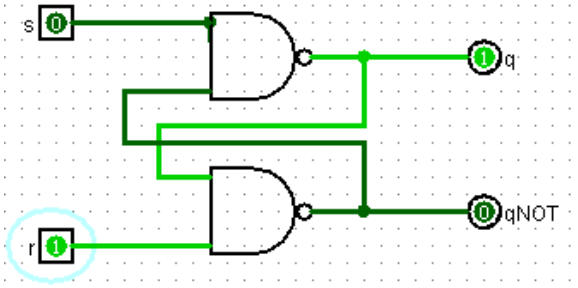
4. The following SR-Latch is built using NAND gates. Find the final values for the outputs Q and $\overline{Q}$ for all latch combinations. Indicate if the function would be set/reset/(hold last Q)/invalid for each specific value of s and r. Show your work and summarize your results in the table below.



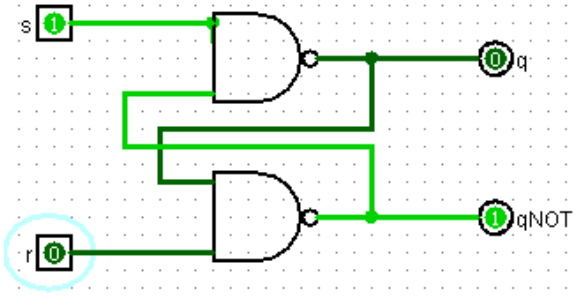
s	r	q	$\overline{q}$	function
0	0	1	1	Reset q and $\overline{q}$ to 1
0	1	1	0	Set q to 1
1	0	0	1	Set q to 0
1	1	-	-	Invalid, causes oscillation



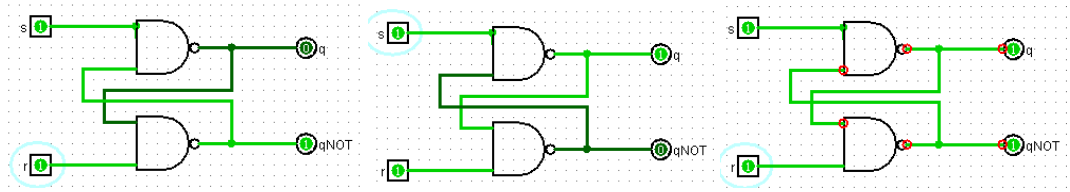
s=0, r=0 sets q and $\overline{q}$ to 1



s=0, r=1 sets q = 1 and $\overline{q}$ = 0

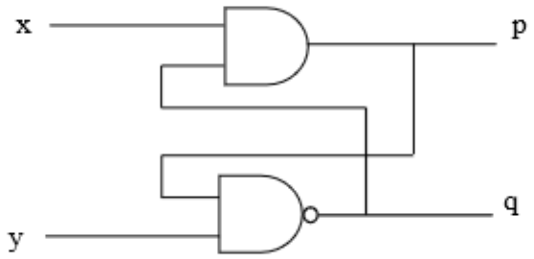


s=1, r=0 sets q = 0 and $\overline{q}$ = 1



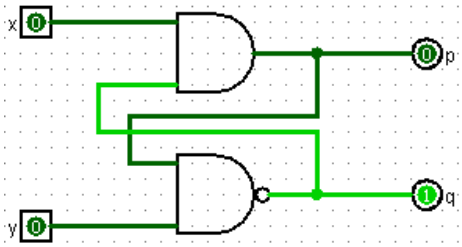
s=1, r=1 will store the last value of q if q != $\overline{q}$, but causes oscillation if q = $\overline{q}$.

5) The following circuit is built using one AND gate and one NAND gate. Find the final values for the outputs p and q of the of the circuit for all combinations. Show your work and summarize your results in the table below. Briefly explain why or why not it could be used as a Latch.

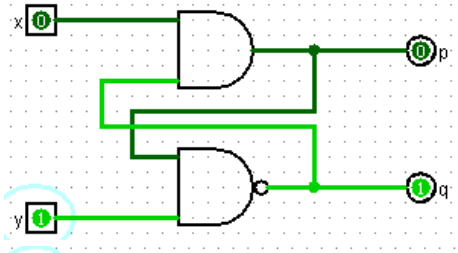


x	y	p	$\overline{q}$	function
0	0	0	1	Sets p to 0
0	1	0	1	Sets p to 0
1	0	1	1	Sets p to 1
1	1	-	-	Invalid, causes oscillation

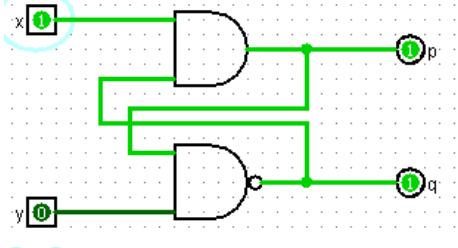
This cannot be used as a latch, as it functions as a wire for p=x and q=1, but adds an oscillation case. It also will not store the value of p when turned off, so it has no "memory" so to speak.



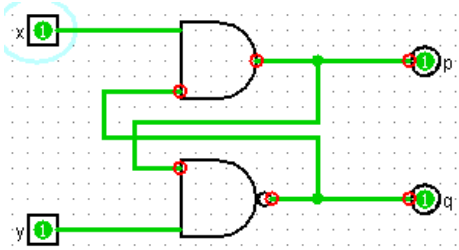
x=0, y=0 sets p=0 and q=1



x=0, y=1 sets p=0 and q=1



x=1, y=0 sets p=1 and q=1



x=1, y=1 causes oscillation

6. Draw the data that is stored by either a D-Latch or a D-FF. Assume that the initial values for each is zero.