CprE 2810 HW06
ELECTRICAL AND COMPUTER ENGINEERING
IOWA STATE UNIVERSITY

Representation and Arithmetic
Assigned: Week 7
Due Date: Oct. 13, 2025

P1 (8 points) Express the hexadecimal number -**9D** in the following binary formats. If it is not possible simply write *Not Possible*. In all cases, indicate how many bits are needed.

a. Unsigned
b. Sign and Magnitude.
c. 1's complement.
d. 2's complement.

P2 (12 points): Perform the following operations on the numbers and indicate if overflow occurs for each operation. All numbers are 6 bits wide (stored in 2's complement). Show your work and all carry bits.

```
  000001        101010        110110
 +             +             +
  001001        010101        111010
_________     _________     _________


  010000        001011        001100
 -             -             -
  000001        001001        010101
_________     _________     _________
```

P3 (10 points) Draw the circuit diagram for a 3-bit ripple carry adder. You are allowed to use only half-adders (as black boxes) and 2-input OR gates. You can't use any other gates or high-level abstractions. Label the inputs and outputs to the circuit and the inputs and outputs to all half-adders.

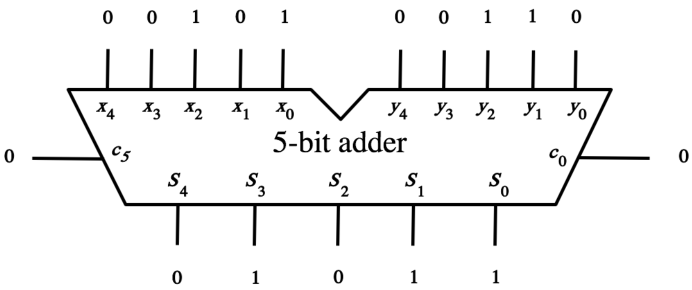
P4 (15 points) Convert the following numbers to IEEE 754 Single-Precision Floating Point format. Write your answer as a 32-bit number. Show your derivations.

a) +15
b) -21
c) -62
d) +6.75
e) -7.125

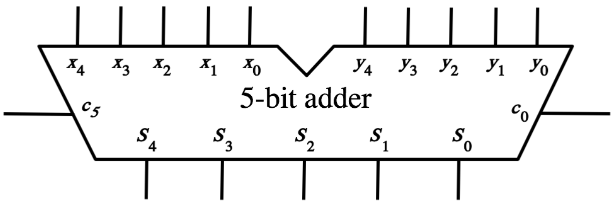
P5 (15 Points) Computations with Adders

In all problems below, the binary numbers are stored in 2's complement representation. For each of the following, assign either a 0 or a 1 to each input and output of the 5-bit adder such that it computes the given expression. Also, for each problem circle either "yes" or "no" to indicate if an overflow will occur. The problem in a) is already solved.

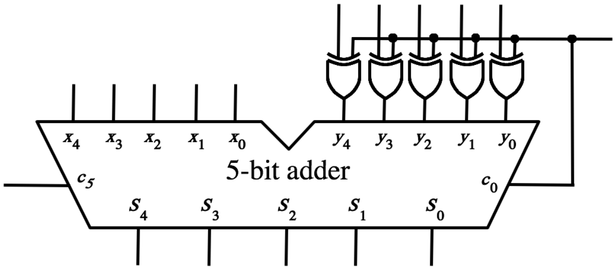
a) (+5) + (+6) = +11 Overflow: yes [✓ no]


0 0 1 0 1
0 0 1 1 0
x_4 x_3 x_2 x_1 x_0
y_4 y_3 y_2 y_1 y_0
5-bit adder
0
c_5
c_0
0
s_4 s_3 s_2 s_1 s_0
0 1 0 1 1

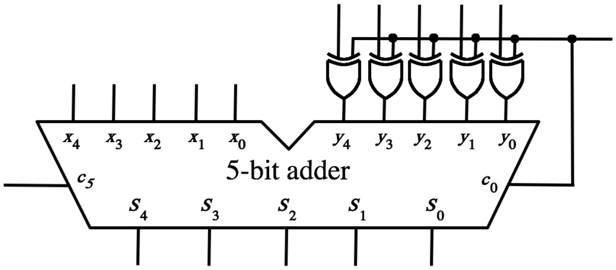
b) (+12) + (+14) = Overflow: yes / no


x_4 x_3 x_2 x_1 x_0
y_4 y_3 y_2 y_1 y_0
5-bit adder
c_5
c_0
s_4 s_3 s_2 s_1 s_0

c) (+9) + (-5) = Overflow: yes / no


x_4 x_3 x_2 x_1 x_0
y_4 y_3 y_2 y_1 y_0
5-bit adder
c_5
c_0
s_4 s_3 s_2 s_1 s_0

d) (+8) - (-13) = Overflow: yes / no


x_4 x_3 x_2 x_1 x_0
y_4 y_3 y_2 y_1 y_0
5-bit adder
c_5
c_0
s_4 s_3 s_2 s_1 s_0

P6 (10 Points) Convert the following numbers from IEEE 754 Single Precision Floating Point format to decimal. The numbers are in hexadecimal. Show your derivations, not just the final result.

a) $C1A00000_{16}$

b) $42F00000_{16}$

P7 (5 Points) Using only one 5-bit full-adder, derive a circuit that performs 10X + 1 on an unsigned 2-bit number X. Clearly show both the 5-bit inputs to the full-adder. The output may be assumed as unsigned.

P8 (10 Points) Multiply using 2's complement binary numbers. Show all your work using **binary numbers**.

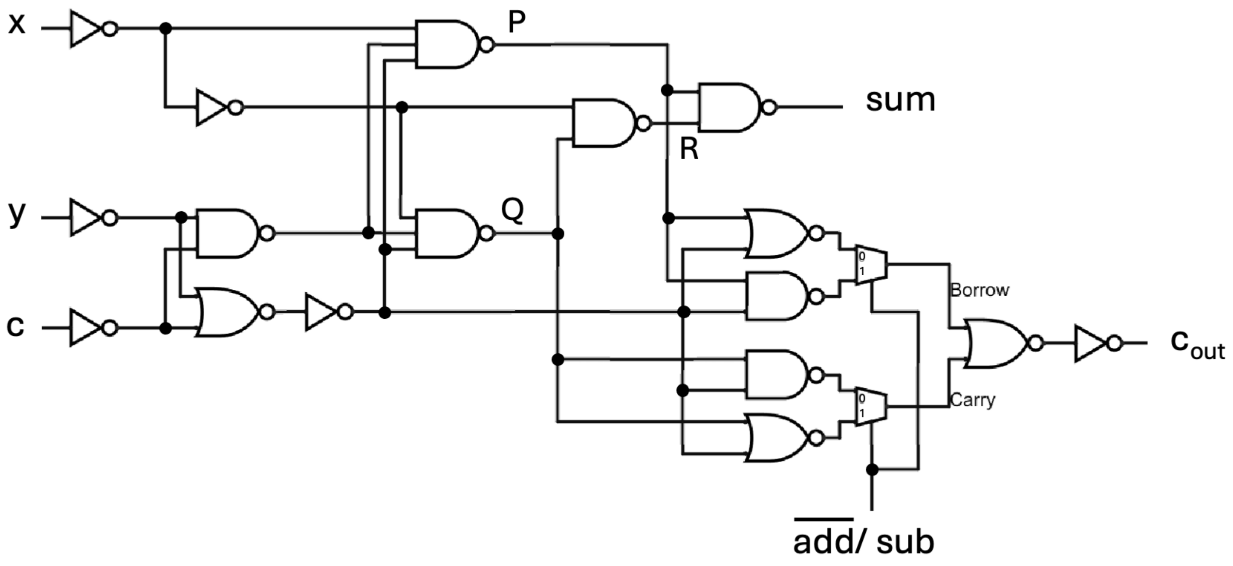
a) 01010 * 01110

b) 10101 * 01010

P9 (15 Points) The ABC's Full-Adder Circuit

The Atanasoff-Berry Computer (ABC) was constructed at Iowa State University between 1939 and 1942. It was the first electronic digital computer. It was also the first parallel computer. It used 30 full-adder circuits that ran in parallel. Each was implemented with vacuum tubes, which favor inverting logic gates. In modern notation, the ABC's full-adder is equivalent to the circuit shown below. Your mission, should you choose to accept it, is to use Boolean algebra to prove that the sum output of this circuit is equal to the sum output of the full-adder covered in the lectures.

Derive the expression for: a) P; b) Q; c) R; d) sum. e) Prove that sum is equal to the sum bit of a modern full adder.


x
P
sum
R
y
Q
c
0
1
Borrow
0
1
Carry
C_{out}
$\overline{\text{add}}$/ sub