Cpr E 281 HW06

ELECTRICAL AND COMPUTER ENGINEERING

IOWA STATE UNIVERSITY

Representation and Arithmetic

Assigned: Week 7

Due Date: Oct. 9, 2023

P1 (8 points): Express -58_{10} in the following binary formats. If it is not possible simply write **Not Possible**. Indicate how many bits are needed.

a. Unsigned
b. Sign & Magnitude
c. 1's Compliment
d. 2's Compliment

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.

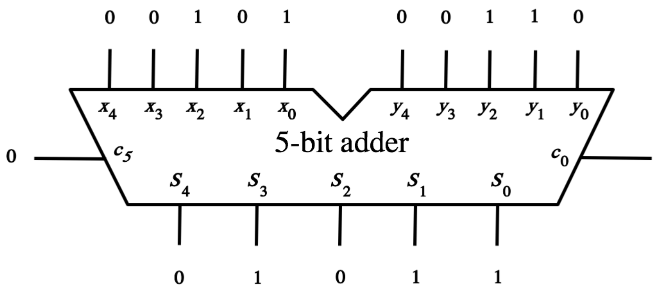
```
+ 110010       + 001011       + 101100
  010100         011001         011110
---------      ---------      ---------


- 100101       - 011110       - 101011
  110011         001100         010111
---------      ---------      ---------
```

P3 (10 points) Draw the complete circuit diagram for a 3-bit ripple-carry adder. You are allowed to use 2-input and 3-input logical gates (of any type), but you can't use any higher-level abstractions (e.g., can't use half-adders or full-adders).

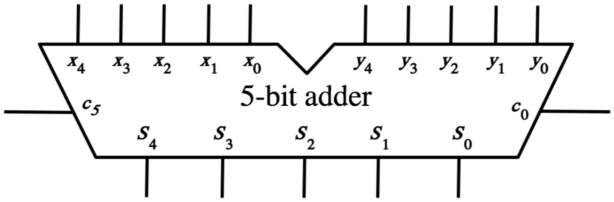
P4 (10 Points): You are given a 4-bit adder as a black box (say a microchip that you can't modify). The adder is too small for what you need to do and also does not compute an overflow bit. Draw a circuit that uses the 4-bit adder and any additional elements that you think are necessary to implement a 6-bit ripple-carry adder that also computes an overflow bit. Label all inputs, outputs and components.

P5 (15 Points): In all problems below, the binary numbers are in 2's complement representation. Assign either a 0 or a 1 to each input and output of the 5-bit adder such that it computes the given expression. The problem in a) is already solved.

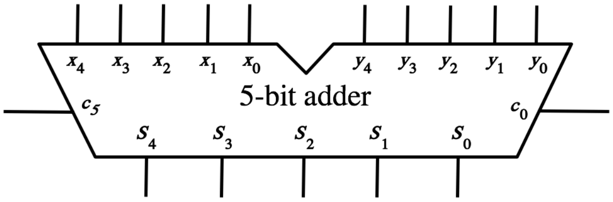
a) **(+5) + (+6) = +11**


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

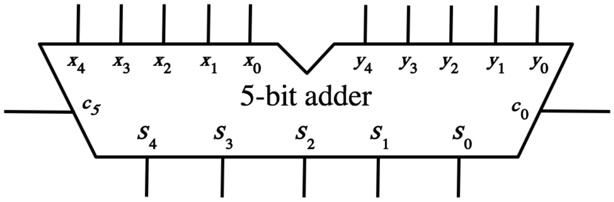
b) **(+13) + (+2) =**


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

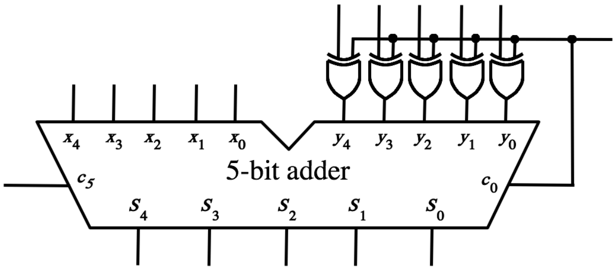
c) **(-12) + (+5) =**


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

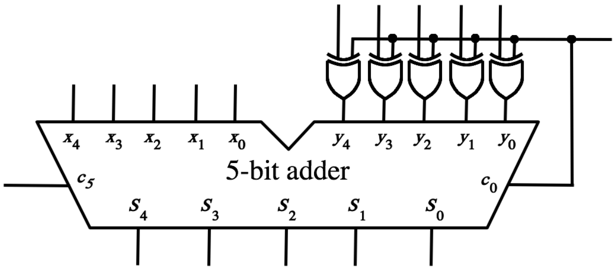
d) **(+14) + (-6) =**


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

e) **(+7) + (-11) =**


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

f) **(-9) - (-5) =**


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 (15 Points): Perform the following multiplications using 2's complement binary numbers. Show all your work using **binary numbers**:

a. 011_2 * 010_2

b. 0101_2 * 0110_2

c. 10010_2 * 00101_2

d. -6_{10} * 3_{10}

e. 10_{10} * 20_{10}

P7 (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) -72
b) 21
c) 54
d) 46
e) -105

P8 (15 points): Convert the following numbers from IEEE 754 Single-Precision Floating Point format to **decimal**. Show your derivations.

a) $01000001100100000000000000000000_2$
b) $11000001110010000000000000000000_2$
c) 42040000_{16}
d) $C2280000_{16}$
e) $C2B80000_{16}$