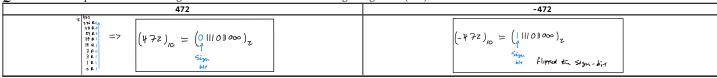
## Assignment 1 Part 2 (posted 02/07/2023, DUE as assigned on Canvas)

Score (by grader):

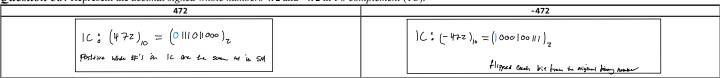
(Don't have to type but must be legible - no credits for any parts I have difficulty reading/understanding.)

5 (8 points) For each (sub-)part, clearly show (may include concise explanations) how you get your answer to not risk incurring penalty.

Question 5a: Represent the decimal signed whole numbers 472 and -472 in sign-magnitude (SM).



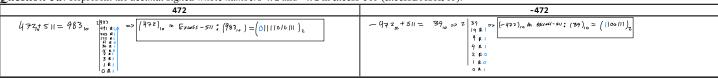
Question 5b: Represent the decimal signed whole numbers 472 and -472 in 1's-complement (1C).



Question 5c: Represent the decimal signed whole numbers 472 and -472 in 2's-complement (2C).



Question 5d: Represent the decimal signed whole numbers 472 and -472 in excess-511 (excessDASH511).



6 (3 points) Using Horner's scheme, represent in decimal (evaluate the decimal value of) the 2's complement number 10011010.

CAUTION: You will earn NO CREDITS if you don't use *Horner's scheme* to more efficiently evaluate polynomial.

You MUST apply Horner's scheme directly on the polynomial expansion for the number as given (NOT on the polynomial expansion for the 2's complement of the given number).

(This means that you should NOT, for instance, evaluate 1011 as -<corresponding positive value of 1011 using Horner's scheme>.)

(It also means that you should NOT, for instance, evaluate 1011 as <unsigned value of 1011 using Horner's scheme > - 24.)

You MUST apply Horner's scheme to cover all contributing bits, i.e., all contributing bits (not just the positive contributing bits, for instance) must participate in the repeated factoring.

You will earn NO CREDITS if you simply show the final result and not clearly show working (i.e., intermediate steps).

$$ZC^{\circ}(10011010)_{z} = 25 \text{ Sm}^{\circ}(11100010)_{z}$$

$$f[\text{orner's Scheme}; (11100010)_{z}$$

$$= 2 \cdot 111$$

$$= 2 \cdot 3 + 1$$

$$= 2 \cdot 114 \cdot 04$$

$$= 2 \cdot 113 + 0$$

$$= 2 \cdot 113 + 0$$

$$= 2 \cdot 113 + 0$$

$$= 2 \cdot 13 + 0$$

- 7 (8 points) For each (sub-)part, do or answer the following:
  - Show how you would add the two 8-bit binary numbers by filling in each of the blank boxes with a 0 or 1.
     NOTE1: Addition of 2's complement numbers is done just like the addition of unsigned whole numbers, so you only need to show each addition once.

NOTE2: There are 9 blank boxes for the Carry bits and 8 blank boxes for the Sum bits.

- CAUTION: Every blank box should be filled with a 0 or 1; if you leave any boxes blank, you risk incurring penalty.
- Circle <u>one</u> observation from the list below that is applicable and useful to help determine whether overflow has occurred, if the numbers are *unsigned whole numbers* and only 8 bits are available for holding the result.

  <u>CAUTION</u>: There will be penalty for additional observations circled that are not applicable/helpful.
- Circle Y or N to indicate your inference on whether overflow has occurred or not (if the numbers are *unsigned whole numbers* and only 8 bits are available for holding the result) based on the circled observation(s).
- Circle <u>two</u> observations from the list below either of which is applicable and useful to help determine whether overflow has occurred, if the numbers are 2's complement numbers and only 8 bits are available for holding the result.

  <u>CAUTION</u>: There will be penalty for observations circled that are not applicable/helpful.
- Circle Y or N to indicate your inference on whether overflow has occurred or not (if the numbers are 2's complement numbers and only 8 bits are available for holding the result) based on the circled observation(s).

List of observations ( $\rightarrow$  each is in terms of the state(s) of certain bit(s) in 1 or 2 positions -- good for hardware implementation):

- (A) Carry-out is 0 when bits at most-significant-bit position are added.
- (B) Carry-out is 1 when bits at most-significant-bit position are added.
- (C) Addends have opposite signs.
- (D) Addends are both positive, sum is positive.
- (E) Addends are both positive, sum is negative.
- (F) Addends are both negative, sum is positive.
- (G) Addends are both negative, sum is negative.
- (H) Carry-in and carry-out at most-significant-bit position are the same.
- (I) Carry-in and carry-out at most-significant-bit position are not the same.

