HW 1

Thursday, January 20, 2022 2:28 PM

CSC 3210
Computer Organization and Programming
Assignment #1
Spring 2022
Due on 02/03/2021, 11:59 PM Eastern Time (US and Canada)

Objective: Learn some core concepts closely relating to assembly language.

Total 15 points

1. **(1 point)** Why 2's complement is necessary to perform subtraction in hardware level? Explain your answer.

2. **(1 point)** Assume that you have three 8-bit storages (called registers) named A, B, and C to store binary numbers. Register A contains 10100111 and register B contains 11110110. Compute A+B and store the value in C register. What is the content of register, C after the computation? **Show the computation in details with carries.**

3. **(2 points)** Assume that you have 8-bit storage to store the numbers. Calculate the following operations using **two's complement method (in binary)**. (Assuming 8-bit registers are used)
   70 – 10 – 42
   [Hint: Perform the computation in binary system, then convert it back to decimal]

4. **(1 point)** What is the hexadecimal representation of the following binary numbers? Show the conversion in details.
   1101001011110110010111 0001

5. **(2 points)** What is the *16-bit* hexadecimal representation of the following *signed decimal* integer? Show all the steps of conversion in details.
   -90

6. **(2 points)** What is the decimal representation of each of the following *signed binary* numbers?
   a. **(1 point)** 11110101
   b. **(1 point)** 00110101

7. **(2 point)** Evaluate the following Hexadecimal expression. **All the numbers are hexadecimal**. Show all the steps of computation and the carries.
   ABC + CDE – 51E

8. **(1 point)** Is it possible to store **-19** in a 5-bit storage. If your answer is YES, then show how to store **-19** in 5-bit register. If your Answer is No, Explain why.

9. **(1 point)** What is the smallest decimal value you can represent, using a **145-bit signed integer?** You can write the number in exponent form.

10. **(2 points)** What is the Boolean expression for P?

x	y	z	P
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Design a circuit that can produce output P for inputs x, y, and z as expressed in the table above.

Note:
- **Make sure to justify all answers – show all work.**
- The Assignment **must be submitted electronically** through iCollege.
- You can do your work in a text editor (Microsoft word, open office, etc.)
- Or you can do it in a piece of paper, then scan or take a picture of the paper.
- Upload the answers in a **pdf file** to iCollege in the respective assignment dropbox.
- All work must be neat and legible. Illegible work will receive no credit. This includes work where the print contrast or darkness are too faint.
- The work that you turn in must be your own --- copying is not allowed for any assignments.
- Using another student's work as your own, allowing another student to use your work as their own, is academic misconduct.

1. 2's compliment is a way to turn binary subtraction into binary addition. In 2's compliment a negative value is indicated by the most significant bit (MSB) being a 1 and the value of a negative is obtained by subtracting 1 and inverting the bits. In other words one forms a 2's complement by inverting all the bits and adding 1.

   -3=1101

   1101–1=1100→0011=3

   This is important because computers need different hardware to do addition and subtraction. However by using 2's compliment subtraction becomes addition so less hardware is required.

2. (A) 10100111 ← Carries
   (B) + 11110110
   = *1 10011101 ← 8-bit

   C = 10011101

Dividend	Remainder
70/2 = 35	0
35/2 = 17	1
17/2 = 8	1
8/2 = 4	0
4/2 = 2	0
2/2 = 1	0
1/2 = 0	1

3. 70 in 8-bit binary is 01000110

   10 in 8-bit binary is 00001010

   One's Complement = 11110101

   Two's Complement = 11110101 + 1 = 11110110

Dividend	Remainder
10/2 = 5	0
5/2 = 2	1
2/2 = 1	0
1/2 = 0	1

   So, -10 in 8-bit binary is 11110110

   42 in 8-bit binary is 00101010

   One's Complement = 11010101

   Two's Complement = 11010101 + 1 = 11010110

Dividend	Remainder
42/2 = 21	0
21/2 = 10	1
10/2 = 5	0
5/2 = 2	1
2/2 = 1	0
1/2 = 0	1

   So, -42 in 8-bit binary is 11010110

   70 – 10 – 42
   = 70 + (-10) + (-42)
   = 01000110 + 11110110 + 11010110 (in binary)

   01000110
   + 11110110
   = *00111100

   00111100
   + 11010110
   = *00010010

   So, the sum in binary is 00010010

   Now knowing that... 00010010 in decimal is 18

4. 1101 | 0011 | 0111 | 1101 | 1100 | 1111 | 0001
   8421 | 8421 | 8421 | 8421 | 8421 | 8421 | 8421
   13 | 3 | 7 | 13 | 12 | 15 | 1
   D | | | D | C | F |

   $D37DCF1_{16}$

5. 90 in 16-bit binary is 0000000001011010

   One's Complement = 1111111110100101

   Two's Complement = 1111111110100101 + 1 = 1111111110100110

Dividend	Remainder
90/2 = 45	0
45/2 = 22	1
22/2 = 11	0
11/2 = 5	1
5/2 = 2	1
2/2 = 1	0
1/2 = 0	1

   So, -90 in 16-bit binary is 1111111110100110

   -90 = 1111 | 1111 | 1010 | 0110
   8421 | 8421 | 8421 | 8421
   15 | 15 | 10 | 6

$-10 = 1111\ 1111\ 1010\ 0110$

8 4 2 1 | 8 4 2 1 | 8 4 2 1 | 8 4 2 1

15 15 10 6

↓ ↓ ↓

F F A

$-90 = FFA6_{16}$

6a. sign bit → 1 1 1 1 0 1 0 1

$-128 + 64 + 32 + 16 + 4 + 1 = \boxed{-11}$

6b. sign bit → 0 0 1 1 0 1 0 1

$32 + 16 + 4 + 1 = \boxed{53}$

7.

$$\begin{array}{r} {\scriptstyle 1\ 1} \\ A\ B\ C \\ +\ C\ D\ E \\ \hline 1\ 7\ 9\ A \end{array} \qquad \begin{array}{r} {\scriptstyle 8\ \ 26} \\ 1\ 7\ \cancel{9}\ \cancel{A} \\ -\ 5\ 1\ E \\ \hline 1\ 2\ 7\ C \end{array} \qquad = \boxed{127C_{16}}$$

8. No, it is not possible to store -19 in a 5-bit storage. To store -19, you need at least a 6-bit storage. Out of the 5 bits, one is used as a sign bit. Now you're only left with 4 bits. In the 4 bits it can be represented as $2^4 - 16$ Numbers.

The 5-bit range is -16 to +15 → -2^4 to (2^4-1)

9. The range of signed integers of n-bits = $-(2^{n-1})$ to $+(2^{n-1}-1)$

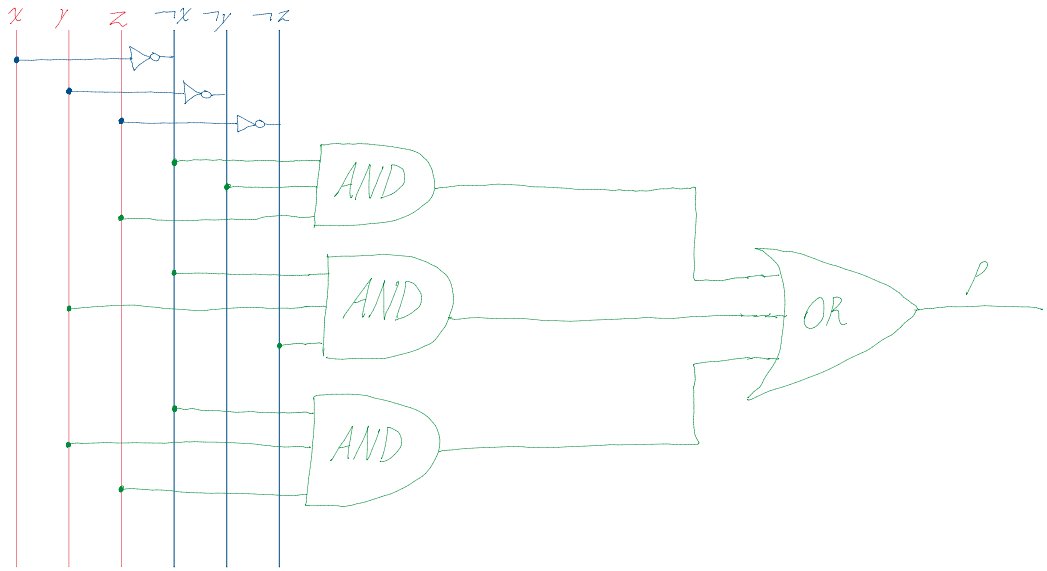
$n = 145$, so $-(2^{(145-1)}) = -2^{144}$

So, the smallest decimal value you can represent in a 145-bit signed integer = $\boxed{-2^{144}}$

10.

x	y	z	P
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

$\boxed{P = (\neg x \wedge \neg y \wedge z) \vee (\neg x \wedge y \wedge \neg z) \vee (\neg x \wedge y \wedge z)}$


x
y
z
¬x
¬y
¬z
AND
AND
AND
OR
P