**CSC 3210**
**Computer Organization and Programming**
**Assignment #1**
**Spring 2023**
**Due on Wednesday 01/25/2023, 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 is assembly language not usually used when writing large application programs?

   Assembly language is not typically used for writing large application programs because it is a low-level programming language that is difficult to read and write, and is not portable across different types of computers. Programs written in assembly language are also difficult to debug and maintain. Additionally, it is less efficient than high-level languages such as C or Python because it requires the programmer to manually manage memory and handle other low-level details. This makes assembly language better suited for specific tasks such as writing operating system components or device drivers, where fine-grained control over the hardware is required.

2. **(1 point)** Assume that you have three 8-bit storages (memory) named A, B, and C to store binary numbers. Memory A contains 11110100 and memory B contains 10110111. 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.**

```
Carry:    1110100_
          11110100
+         10110111
          110101011
```

3. **(2 points)** Assume that you have 4-bit storage to store the numbers. Calculate the following operations using **two's complement method**. Show all the computations in details. (assuming 4-bit register is used)

   -3 – 1 -1

   [Hint: Perform the computation in binary system, then convert it back to decimal]

   3: 0011 -> -3:1101
   1: 0001 -> -1:1111

```
-3-1 = (-3)+(-1)
Carriers:   111_
              1101
          + 1111
            11100
4bits => 1100(-4)
```

```
-4-1 = (-4)+(-1)
Carriers:   111_
              1100
          + 1111
            11011
4bits => 1011 (-5)
```

   ⇨ **Answer: 1011(two's complement) or -5**

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

10001010010101011110000110011011 7

10001010010101011110000110011011 7 = 0001 0001 0100 1010 1011 1100 0011 0011 0111

| Binary | Decimal | Hexadecimal |
|---|---|---|
| 0001 | 1 | 1 |
| 0001 | 1 | 1 |
| 0100 | 4 | 4 |
| 1010 | 10 | A |
| 1011 | 11 | B |
| 1100 | 12 | C |
| 0011 | 3 | 3 |
| 0011 | 3 | 3 |
| 0111 | 7 | 7 |

⇨ **114ABC337**

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

-58

**Two's complement:**

58 in binary = 0000 0000 0011 1010 (16-bit)
-58 in two's complement = 1111 1111 1100 0110 (16-bit)

| Binary | Decimal | Hexadecimal |
|---|---|---|
| 1111 | $2^3+2^2+2^1+2^0=15$ | F |
| 1111 | $2^3+2^2+2^1+2^0=15$ | F |
| 1100 | $2^3+2^2=12$ | C |
| 0110 | $2^2+2^1=6$ | 6 |

⇨ **-58 = FFC6**

**Without two's complement:**

58/16 = 3, Remainder = 10(A)
3/16 = 0, Remainder = 3

58 => 3A

⇨ **-58 = -3A**

**6.** **(2 points)** What is the decimal representation of each of the following *signed binary* numbers? Show the computation.

a. **(1 point)** 01110111

$(0\times(-2^7)) + (1\times2^6) + (1\times2^5) + (1\times2^4) + (0\times2^3) + (1\times2^2) + (1\times2^1) + (1\times2^0) = 0+64+32+16+0+4+2+1$

⇨ **119**

b. **(1 point)** 11110001

$(1\times(-2^7))+ (1\times2^6) + (1\times2^5) + (1\times2^4) + (0\times2^3) + (0\times2^2) + (0\times2^1) + (1\times2^0) = (-128) + 64 + 32 + 16 + 0 + 0 + 0 + 1$

⇨ **-15**

**7.** **(2 point)** Evaluate the following Hexadecimal expression. **All the numbers are hexadecimal**. Show all the steps of computation and the carries.

A1C + CCF – FFE

*Carries*: 101_
A1C
\+ CCF
**16EB**

Step 1: 12 (C) + 15 (B) = 27, 27/16 = 1 (carry = 1) and remainder is 11 (B)
Step 2: Since carry = 1, 1 + 1 + 12 (C) = 14 (E) , no carry because 14 not greater than 16
Step 3: 10 (A) + 12 (C) = 22. Because 22>16, 22/16 = 1 (carry = 1) and remainder is 6
Step 4: Because there is no number to calculate, we drop the carry down. It's 1.
⇨ **16EB**

*Carries*: 111_
16EB
\- FFE
**6ED**

Step 1: 11 (B) < 14 (E) so we borrow a unit from E(in front of B) (carry = 1). 11+16-14 = 13 (D)
Step 2: Carry = 1, so 14 (E) -1 = 13, 13 < 15 (F) so we borrow a unit from 6 (carry =1). 13+16-15 = 14 (E)
Step 3: Carry = 1, so 6 – 1 = 5, 5 < 15 (F) so we borrow a unit from 1 (carry =1). 5+16-15 = 6
Step 4: Carry = 1, so 1 – 1 = 0 and nothing to divide with. So it is 0 (or nothing because it stands at the beginning)

⇨ **6ED**

**8.** **(1 point)** Is it possible to store **-10** in a 4-bit storage. If your answer is YES, then show how to store **-10** in 4-bit register. If your Answer is No, Explain why.

No, because -10 is a signed decimal so we have to use two's complement or one bit at the beginning to store sign.
With two's complement, we can only store integer from $-2^3$ to $(2^3-1)$ or -8 to 7, not include **-10**
With sign bit, we spend 1 bit for sign so we have 3 bit left, which can store number from 0 to $2^3-1 = 7$, not include **10**

⇨ **It is impossible to store -10 in a 4-bit storage.**

**9.** **(1 point)** What is the smallest decimal value you can represent, using a **120-bit signed integer?** You can write the number in exponent form.

⇨ **$-(2^{119})$**

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

| x | y | z | P |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

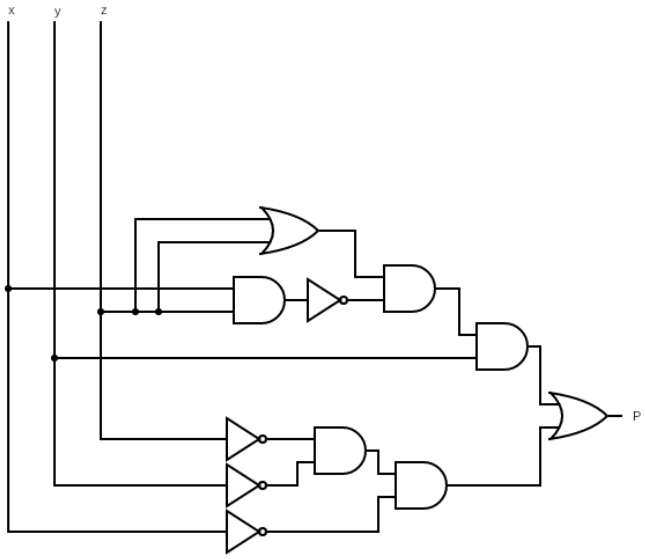
Design a circuit that can produce output P for inputs x, y, and z as expressed in the table above.

**Solution:**

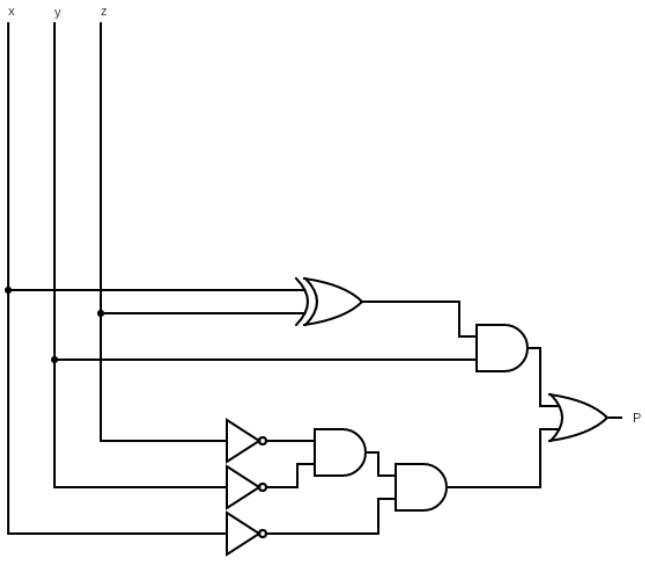
P = $\overline{xyz} + y(x + z)\overline{xz}$

Circuit:

- Without XOR gate:




- With XOR gate:




**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.

**Late submission:**

A late penalty will be applied to any submission after the due date.

- If you submit the assignment within 1 day after due date, the late penalty is 10% of the grade.
- If you submit the assignment within 2 day after due date, the late penalty is 15% of the grade.
- If you submit the assignment within 3 day after due date, the late penalty is 20% of the grade.
- If you submit the assignment within 4 day after due date, the late penalty is 25% of the grade.
- For any assignment submitted after 4 day, the late penalty is 35%.

**How to calculate the late penalty?**

Let's assume that the assignment has total point of 15. And you submitted the assignment within 2 days after the due date. After grading you received 13 out of 15 in the assignment. The late penalty will be 15% of 15 = 2.25 points. After late penalty deduction your grade is 10.75 out of 15.