COMP1411 (Spring 2024) **Introduction to Computer Systems**

Individual Assignment 1 Duration: 00:00, 24-Feb-2024 ~ 23:59, 25-Feb-2024

Name: WANG Yuqi
Student number: [redacted]

There are four questions in this assignment (some of the questions have sub-questions). Write down your answers in the blank area under each question. A total of 5 marks are distributed among the questions.

For any question, show your steps to obtain the final result. Only giving the final result will cause you to LOSE a significant mark on the questions.

□

Question 1. [2 marks]

Consider a 32-bit floating-point representation based on the IEEE floating-point format:

- the highest bit is used for the sign bit,
- the sign bit is followed by 5 exponent bits, which are then
- followed by 26 fraction bits.

(1) **Convert** decimal value -29.21875 into the above 32-bit IEEE floating-point format. Write out the result in the hex-decimal format.

Answer:

$(29)_{10}$ = 2|29 ---1
2|14 ---0
2|7 ---1
2|3 ---1
2|1 --- 1
0

$0.21875 \times 2 = 0.4375$
$0.4375 \times 2 = 0.875$
$0.875 \times 2 = 1.75$
$0.75 \times 2 = 1.5$
$0.5 \times 2 = 1$

$\therefore -29.21875 = -11101.00111$

Normalize: -1.110100111×2^4

Bias exponent: $4 + (2^{5-1} - 1) = (19)_{10} \rightarrow$

$= (10011)_2$

$$
\begin{array}{l|l}
2 & 19 \; \text{---} \; 1 \\
2 & 9 \; \text{---} \; 1 \\
2 & 4 \; \text{---} \; 0 \\
2 & 3 \; \text{---} \; 0 \\
2 & 1 \; \text{---} \; 1 \\
 & 0
\end{array}
$$

Therefor:

IEEE Float Point $= \underbrace{1}_{\text{sign}} \; \underbrace{10011}_{\text{Bias Exp}} \; \underbrace{11010011100000000000000000}_{\text{Significand.}}$

(2) Assume this 32-bit number is stored on a **big-endian** machine in the addresses 0x100~0x103. Please fill in the following table to show the byte stored in each address. To write a byte, please use the hex-decimal format starting with 0x.

$= 0x\,CF4E0000$

Memory Address	Byte in the Address
0x0100	?? CF
0x0101	?? 4E
0x0102	?? 00
0x0103	?? 00

Question 2. [0.6 marks]

Suppose that x and y are unsigned integers.

(1) **Re-write** the following C-language statement only using << and – operations. Introducing new variables (other than x and y) is not allowed. Please show your steps.

y = x * 78;

Answer:

$y = x(2^7 - 2^5 - 2^4 - 2^1)$

$= 2^7x - 2^5x - 2^4x - 2x$

$\therefore \; y = (x<<7) - (x<<5) - (x<<4) - (x<<1)$

□

Question 3. [1.4 marks]

Consider a 16-bit floating-point representation based on the IEEE floating-point format:

- the highest bit is used for the sign bit,
- the sign bit is followed by 4 exponent bits, which are then
- followed by 11 fraction bits.

(1) What is the **largest positive normalized number** with the above floating-point format? Write the number in binary form.

(2) **Compute** the decimal value of the bit vector 0x6D80 with the above floating-point format. Write the result in decimal format.

Answer:

□

(1) $\underbrace{0}_{\text{sign}}\ \underbrace{1110}_{\text{Bias Exp}}\ \underbrace{11111111111}_{\text{significand}}$

Bias E: $(1110)_2 = 14$

$E: 14-(2^3-1) = 7$

∴ Shift radix point to the right 7 bits:

$(11111111.1111)_2$

(2) $0x6D80 = (0\underline{110}\ 1101\ 1000\ 0000)_2$

Bias Exp $= (1101)_2 = (13)_{10}$

$Exp = 13-(2^3-1) = 6$

Shift radix point to right by 6 bits:

$(1101100.00000)_2$

$= 108.0$

Question 4. [1 mark]

Given the following C program:

```
#include "stdio.h"

void main()
{
    unsigned char a;
    char b;

    a = 0x9C;
    b = a;

    printf("b = %d\n", b);
    return;
}
```

(1) What is the output of this program?

(2) Explain why the output is generated in detail.

Answer:

(1) -100

(2)
1. Since ‘a’ is unsigned char, 0x9C is interpreted as 156 in decimal
2. However, when copied to ‘b’, due to signed nature of ‘b’, 0x9C is treated as 2’s complement.
3. ‘b’ is then passed to “printf” with a “%d” specifier, leading to integer promotion.
4. Finally the value of ‘b’, 0x9C, is interpreted as a negative signed integer -100.

Exact conversion:

0x9C = 0b10011100
To decimal:
10011100-1 = 10011011
reverse = 01100100 = 100
Add sign: -100