**COMP1411 (Spring 2023) Introduction to Computer Systems**

Individual Assignment 1 Duration: 00:00, 11-Feb-2023 ~ 23:59, 12-Feb-2023

| | |
|---|---|
| *Name* | **Zhu Jin Shun** |
| *Student number* | **22101071d** |

**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 6 exponent bits, which are then
- followed by 25 fraction bits.

(1) **Convert** decimal value -28.40625 into the above 32-bit IEEE floating-point format. Write out the result in the hex-decimal form.

*Answer*:

Negative number so s=1

28/2=14…0

14/2=7…0

7/2=3…1

3/2=1…1

1/2=0…1

28=$11100_2$

0.40625*2=0.8125

0.8125*2=1.625

1.625*2=3.25

3.25*2=6.5

6.5*2=13

13=$1101_2$

0.40625=$0.01101_2$

$11100.01101_2$=$1.110001101_2$*2^4

Exp=4+(2^(6-1)-1)=35=$100011_2$

Frac=110001101…0

IEEE format

S 6-bits exp 25-bits frac

1 100 011 1 1000 1101 0000 0000 0000 0000

Hex-decimal form:0xC78D0000

(2) Assume this 32-bit number is stored on a little-endian machine in the addresses 0x300~0x303. 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.

| Address | Byte in the Address |
|---|---|
| 0x0300 | 0x00 |
| 0x0301 | 0x00 |
| 0x0302 | 0x8D |
| 0x0303 | 0xC7 |

**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 * 84;**

*Answer*

X^7=128

X^2=4

X^3=8

X^5=32

84=128-32-4-8

y=(x<<7) - (x<<2) - (x<<3) - (x<<5);

**(2)** Given the C-language statement y = x * b, assume that y, x, and b are all unsigned integers, and the result of x * b does not overflow.
Can the statement y = x * b always be re-written into a C-language statement only using << and – operations? Please answer yes or no, and explain why. Only answering yes or no without any explanation will receive a zero mark for this question.

*Answer*:

Yes, because any odd or even number could be represented by using << and – operations, by – one x can make all numbers possible in the language statement. For odd numbers b, we could use << operations to make an even number first then minus x by making it into an odd number, for example if b=7 then y=(x<<3)-x. Also, for negative numbers, by – x we could also get the number b, for example, b=-3 then y=x-x-x-x-x. Therefore, any y = x * b can always be re-written into a C-language statement only using << and – operations.

**Question 3**. [1.4 marks]

Consider a 12-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 7 fraction bits.

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

(2) **Compute** the decimal value of the bit vector 0xCB60 with the above floating-point format. Write the result in decimal form.

*Answer*:

1)

12 bit-largest-

S 4-bit exp 7-bits frac

0 1110 111 1111

S=0 positive

Exp=1110=14

Bias=2^ (4-1) -1=7

E=exp-Bias=14-7=7

Frac=1111111

M=$1.1111111_2$=1.9921875

Decimal value=1.9921875*2^7=255

2)

C=12=1100

B=11=1011

6=0110

0=0000

Because 12 bits then C deleted

IEEE format

S 4-bit exp 7-bits frac

1 011 0110 0000

S=1-negative

Exp=$0110_2$=6

Bias=2^ (4-1)-1=7

E=exp-bias=6-7=-1

Frac=0110 0000

M=$1.01100000_2$=1.75

Decimal Value= 1.75*-1*$2^{-1}$=-0.375

**Question 4**. [1 mark]

Suppose that x, y, z, and a are all 16-bit unsigned integers.

(1) Assume that the left-most bit is the highest bit. Write a single C-language statement to set the value of a such that:

   a. the left-most 5 bits of a are the same as the right-most 5 bits of x;
   b. the right-most 4 bits of a are the same as the left-most 4 bits of y;
   c. the middle 7 bits of a are the same as the left-most 7 bits of z.

Note that:

- You are only allowed to use logical bit shift operations and bit operations, including |, ^, and &, to set the value of a;
- NO arithmetic or if-then-else test (in any form) is allowed;
- Introducing new variables (other than x, y, z, and a) is NOT allowed;
- Using mask numbers is NOT allowed.

*Answer*:

x=x>>11>> 11;

y= y<<12>> 12;

z= x<<9>> 9;

a=x | z | y;

(2) If x = 0xDC9E, y = 0x36A7, and z = 0x9928, what the be the resulting value of a? Please write the value of a in hex-decimal form starting with the prefix 0x.

*Answer*:

D=1101

C=1100

9=1001

E=1110

x=1101 1100 1001 1110

3=0011

6=0110

A=1010

7=0111

y=0011 0110 1010 0111

9=1001

9=1001

2=0010

8=1000

z=1001 1001 0010 1000

x=11110

z=1001100

y=0011

a=1111 0100 1100 0011

1111=F

0100=4

1100=C

0011=3

a=0xF4C3