COMP1411 (Spring 2022) Introduction to Computer Systems

Individual Assignment 1 Duration: 00:00, 19-Feb-2022 ~ 23:59, 20-Feb-2022

Name	**ZHOU, Siyu**
Student number	

Question 1. [0.5 marks]

Suppose that `x` and `y` are unsigned integers.

Rewrite the following C-language statement by using `<<` and `-`.

```
y = x * 77;
```

Introducing new variables (other than `x` and `y`) is not allowed.

Show your steps. Only giving the final result will NOT get a full mark of this question.

Answer:

x << 7 == x * 128

x << 6 == x * 64

x << 5 == x * 32

x << 4 == x * 16

x << 3 == x * 8

x << 2 == x * 4

x << 1 == x * 2

x << 0 == x * 1

Because 77 is an odd number, there will be x << 0

Also 77 < 128 and 77 > 64, then (x << 7) > x * 77

(x << 7) – (x * 77) – (x << 0) = x * 50

Because 50 < 64 and 50 > 32, then (x << 5) < x * 50

(x * 50) – (x << 5) = x * 18

Because 18 < 32 and 18 > 16, then (x << 4) < x * 18

(x * 18) – (x << 4) = x * 2

Because 2 = 2, then (x << 1) =x * 2

(x * 2) – (x << 1) = 0

Then Final result is :

y = (x << 7) – (x << 4) – (x << 3) – (x << 1) – (x << 0) == x * 77

Question 2. [1 mark]

Suppose that a, b, c and z are all 32-bit unsigned integers.

(1) Assume that the left-most bit is the highest bit. Write C-language statements to set the value of z, such that:

a. the left-most 10 bits of z are the same as the right-most 10 bits of a;
b. the right-most 14 bits of z are the same as the left-most 14 bits of b;
c. the middle 8 bits of z are the same as the right-most 8 bits of c.

Note that:

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

(2) If a = 0xC9E3BA75, b = 0x268DBA83, and c = 0x63ABE432, what the be the resulting value of z? Please write the value of z in hex-decimal form starting with prefix 0x.

Show your steps. Only giving the final result will NOT get a full mark of this question.

Answer:

(1)

(z >> 22) == (a Log. >> 22)

(z Log. >> 18) == (b >> 18)

((z Log. >> 10) >> 14) == (c Log. >> 24)

(2)

a in binary representatives is 11001001111000111011101001110101

b in binary representatives is 00100110100011011011101010000011

c in binary representatives is 01100011101010111110010000110010

The left-most 10 bits of z is (a Log. >> 22) = 1001110101.

The right-most 10 bits of z is (b >> 18) == 00100110100011.

The middle 8 bits of z is (c Log. >> 24) == 00110010.

z in binary representatives is 10011101010011001000100110100011

z in hex-decimal form is 0x9D4C89A3

Question 3. [2 marks]

Assume on a big-endian machine, a 32-bit single-precision floating-point number is stored in the addresses 0x0200 ~ 0x0203 is as follows:

Address	Byte in the Address
0x0200	0xC1
0x0201	0x94
0x0202	0x02
0x0203	0x3F

Convert the above floating-point number to a decimal number.

For the converted decimal number, leave only 3 digits after the decimal point and discard all the rest digits; DO NOT write the result in the exponential form of the power of 2 or 10.

Show your steps. Only giving the final result will NOT get a full mark of this question.

Answer:

0xC194023F

C194023F in binary form is 11000001100101000000001000111111

s = 1 means negative

exp = $10000011_2 = 131_{10}$

Bias = 2^{8-1}-1 = 127_{10}

M= $1.01000000001000111111_2 = 1.2505483627_{10}$

E = exp – Bias = $131_{10} – 127_{10}$ = 4

v = $(–1)^s$ M 2^E = $(-1)^1$ * 1.2505483627 * 2^4 = -20.0087738 = -20.009

Question 4. [1.5 marks]

Consider a 10-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 5 fraction bits.

Question 1: What is the largest positive normalized number? Write the numbers in both the binary form and the decimal value.

Question 2: **Convert** the decimal number 12.875 into the above 10-bit IEEE floating-point format. Write the result in the binary form.

Show your steps for both Question 1 and Question 2. Only giving the final result will NOT get a full mark of this question.

Answer:

1:

$v = (-1)^s M 2^E$

As for largest positive normalized number, s equals to 0.

Exponent E = exp – Bias

Bias = 2^{k-1}-1, where k is the number of bits for exponent bits, then Bias equals to 2^{4-1}-1= 7

The biggest exp equals to 1110 in binary form (the following 4 exponent bits), which is 14 in decimal form. Then E = 14 – 7 = 7

Because the range of M for normalized value is [1.0, 2.0), then maximum M approaches 2.

Then $v = (-1)^0 * 2 * 2^7 = 2^8 = 256$ in decimal value

And the binary form is 100000000

2:

v = 12.875

12.875 is positive, then s = 0

$12.875 = (-1)^0 M 2^E$

Bias equals to 2^{4-1}-1= 7

If exp does not equal to 0000 and 1111:

Exponent E = exp – Bias, and Bias equals to 7

Then 12.875 = $(–1)^0$ M 2^{exp-7} = M * 2^{exp-7}

12.875 in binary form is 1100.111

Then 2^{exp} * M in binary form is 11001110000.

According to definition of normalized value, M in range [1.0, 2.0). M can be calculated

by using 11001110000 can only shift binary point to the form of 1.xxx…xxx.

M equals to 1.100111 in binary form, and M is represented in 5 fractional bits, then exp

equals to 10, and M = 1.1010

exp = 10 = 1010_2

s=0

frac = 11010

Then result in binary form is 0101011010

Else if exp equals to 0000:

Exponent E = 1 – Bias = -6

Then M > 1 and M = 0.0000_2

It contradicts, and this assumption does not hold.

Else (exp equals to 1111):

The value will be infinity or there's no numeric value

It also contradicts, and this assumption does not hold.

Overall, the result in binary form is 0101011010