COMP1411 (Spring 2022) Introduction to Computer Systems

REFEREENCE ANSWERS FOR ASSIGNMENT 1

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

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:

STEP 1: representing the number 77 in the form of subtractions of numbers in the form of the power of 2.

$77 = 128 – 32 – 16 – 2 – 1 = 2^7 – 2^5 – 2^4 – 2^1 – 2^0$

STEP 2: rewrite the statement with << and –.

$y = x * (2^7 – 2^5 – 2^4 – 2^1 – 2^0)$

The new statement is:

y = (x << 7) – (x << 5) – (x << 4) – (x << 1) – x;

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) The statement is:
   z = (a << 22) | ((c << 24) >> 10) | (b >> 18)

(2) a = 1100 1001 1110 0011 1011 1010 0111 0101
   b = 0010 0110 1000 1101 1011 1010 1000 0011
   c = 0110 0011 1010 1011 1110 0100 0011 0010
   z = 1001 1101 0100 1100 1000 1001 1010 0011
   z = 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:

STEP 1:

The hex-decimal is: 0xC194023F

The binary is: 1100 0001 1001 0100 0000 0010 0011 1111

STEP 2:

The sign bit = 1, negative number

The exponent bits: 10000011, so e = 131 – 127 = 4

The fraction part: 1.00101000000001000111111

The number: -18.501

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:

Q1:

The largest positive normalized number is: 0 1110 11111

The value is: 252

Q2:

STEP 1: 12.875 = 1100.111 = 1.100111 * 2^3

STEP 2:

The sign bit: 0

The exp = 3 + 7 (bias) = 10 = 1010 (binary)

The fraction part: 100111 rounded to 10100

The 10-bit binary number: 0101010100