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

Individual Assignment 1 Duration: 00:00, 11-Feb-2022 ~ 23:59, 12-Feb-2022

|  |  |
| --- | --- |
| *Name* | **Jyotsna Venkatesan** |
| *Student number* | **22108825d** |

**Question 1**. [0.5 marks]

Suppose that x and y are unsigned integers.

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

y = x \* 137;

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*:

137 = 2^8 - 2^6 - 2^5 - 2^4 - 2^2 - 2^1 - 2^0

y = x \* 137

Hence,

y = 1 \* x \* 28 - 1 \* x \* 26  - 1 \* x \* 25  - 1 \* x \* 24  - 1 \* x \* 22  - 1 \* x \* 21  - 1 \* x \* 20

Thus,

y = (x << 8) - (x << 6) – (x << 5) – (x << 4) – (x << 2) – (x << 1) – (x << 0)

**Question 2**. [1.5 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:
   1. the left-most 11 bits of z are the same as the left-most 11 bits of a;
   2. the right-most 14 bits of z are the same as the right-most 14 bits of b;
   3. the middle 7 bits of z are the same as the middle 7 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;

1. z=(a&0b1111 1111 1110 0000 0000 0000 0000 0000)
2. z=(b&0b0000 0000 0000 0000 0011 1111 1111 1111)
3. z=(c&0b0000 0000 0001 1111 1100 0000 0000 0000)
4. Write another C statement to set the value of z to the same results in (1), but do NOT use masks.
5. 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*:

Binary form of a:

1100 1001 1110 0011 1011 1010 0111 0101

Binary form of b:

0010 0110 1000 1101 1011 1010 1000 0011

Binary form of c:

0110 0011 1010 1011 1110 0100 0011 0010

Thus in binary, z=1100 1001 1110 1011 1111 1010 1000 0011

In hexadecimal z=0xC9EBFA83

**Question 3**. [1.5 marks]

Assume on a little-endian machine, a 32-bit single-precision floating-point number is stored in the addresses 0x0100 ~ 0x0103 is as follows:

|  |  |
| --- | --- |
| Address | Byte in the Address |
| 0x0100 | 0x3F |
| 0x0101 | 0x02 |
| 0x0102 | 0x94 |
| 0x0103 | 0xC1 |

**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*:

Hexa-decimal = 0x3F0294C1

Binary = 00111111000000101001010011000001

First Digit is sign bit

Exponent: 01111110 = 126

Exp – Bias = -1

Significand: 00000101001010011000001

M = 1.00000101001010011000001 = 1.02016460895538330078

Decimal = 5.100e-1

**Question 4**. [1.5 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 5 exponent bits, which are then
* followed by 6 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 23.875 into the above 12-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*:

Question 1:

0 11101 111111

Binary:

Exp = 11101 (Binary) = 29 (Decimal)

Exp – Bias = 29 – (2\*4 – 1) = 29 – 16 = 13

M = 1.111111 (binary) = 1.984375 (Decimal)

Ans = (-1)^s \* M \* 2^E

= (-1)^0 \* 1.984375 \* 2^6 = 127

Question 2:

23.875 (decimal) = 10111.111 (Binary)

1 01111 111111

Exp = 01111 (binary) = 15 (decimal)

Exp – Bias = 15 – (2^4 -1) = 15 – (16 – 1) = 0

M = 1.111111 (binary) = 1.984375 (Decimal)

Ans = (-1)^1 \* 1.984375 \* 2^6 = -127