

Homework 1

DDL: 23:59, Thursday October 13

Hand in: Pack up your answers (PDF) and program as Zip file and send to ouc14@fudan.edu.cn before DDL

Problem 1

Please convert the numbers below in given base to target radix.

- (a) $(BD2)_{16} \rightarrow (_\[1]_)_{10}$
- (b) $(10000101)_2 \rightarrow (_\[2]_)_{10}$
- (c) $(5655)_{10} \rightarrow (_\[3]_)_{16}$
- (d) $(000100001)_2 \rightarrow (_\[4]_)_{16}$
- (e) $(5392)_{10} \rightarrow (_\[5]_)_2$
- (f) $(11)_{16} \rightarrow (_\[6]_)_2$
- (g) $(11)_8 \rightarrow (_\[7]_)_2$

Problem 2

Suppose A and B are both 8-bit variables. A has value 0xA3 and B has value 0xDA.

Calculate the following expression. (Give the answer in hexadecimal form)

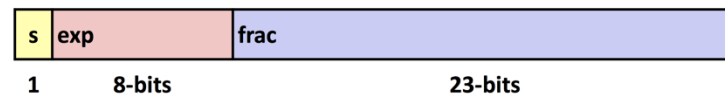
- (a) $A \mid B$
- (b) $\sim A \mid \sim B$
- (c) $A \& B$
- (d) $A \&\& B$
- (e) $A \wedge (\sim B)$
- (f) $(A \ll 2) \gg 4$ (arithmetic)
- (g) $B \gg 3$ (logical)

Problem 3

Number Conversion: IEEE Standard 754 for floating point arithmetic is illustrated as bellow:

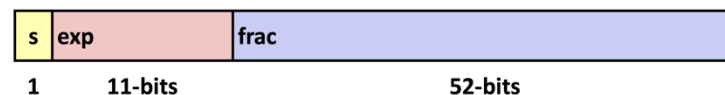
■ **Single precision: 32 bits**

≈ 7 decimal digits, $10^{\pm 38}$



■ **Double precision: 64 bits**

≈ 16 decimal digits, $10^{\pm 308}$



Normalized: $(-1)^{sign} * (1.frac) * 2^{exp-127}$ (exp = 1 ~ 254)

Denormalized: $(-1)^{sign} * (0.frac) * 2^{-126}$ (when exp = 0, frac \neq 0)

Zero: all 0s in all 3 fields.

- (1) Convert the number -23.15213 into IEEE 754 float point single representation (in hex). That is, you should write in the form like “0xABCDEF”.

$$(-23.15213)_{10} = (\quad)_{16}$$

- (2) What is the equivalent value as a decimal floating-point number?
(Note: You may have to use “Infinity” and “Nan”).

$$(0100\ 0011\ 0101\ 0100\ 0000\ 0000\ 0000\ 0000)_2$$

$$(0111\ 1111\ 1011\ 0100\ 0000\ 0000\ 0000\ 0000)_2$$

- (3) Can the number 2^{-127} be represented? If yes, show the number in IEEE 754 float point representation in hex. If no, explain the reason.

Problem 4

Write a C program (NOT C++) to implement ADD & MULTIPLY operation on IEEE 754 float point single precision representation number.

Problem 5

Write a C program to ‘re-explain’ between INT and FLOAT. That is, given a variable of type int, your program should output the equivalent value of the same binary representation explained as IEEE 754 float point single precision representation.

For example, given int a = 1, its binary representation is 0...01 and the equivalent value of this binary representation explained as IEEE 754 float point single precision representation is $(0.1) * 2^{-126}$.

You should handle both directions, int->float & float->int.