THE UNIVERSITY COLLEGE
OF THE CARIBOO

COMPUTING 253

Small Computer Systems: Organisation and Architecture

Take Home Review – Numeric Representation
"Fixed and Floating—What's the Point?"

---

*For completeness, **show your work** wherever possible.*

1. Using the direct method of calculating the decimal form of any 2's complement number (whether the number is positive or negative).

   Direct method:
   - let the msb (*sign bit*) have a data value of $-(2^{(n-1)})_{10}$; n is the *integer word size* (to the left of *point*)
   - add the unit values of the remaining bits to the first number (which is either zero, or a large negative)
   - since the msb unit value is larger than all other bits put together, if the msb = 1 the outcome is negative, if msb = 0 the outcome is positive
   - *note: this method does perform any strange "flip & add 1" techniques*

   Use this technique to determine the values of the binary fixed-point numbers below,

   using 8-bit word, 2's complement,

   a) $0000\ 0011_2 = ?_{10}$ b) $1000\ 0011_2 = ?_{10}$

   using 8-bit word, 2's complement, 3-bit precision,

   c) $00010.101_2 = ?_{10}$ d) $10010.101_2 = ?_{10}$

2. Convert and calculate the following with fixed-point on an 8-bit word and 4-bit precision (all-positive),

   a) What is the *largest value* that can be stored? (Answer in binary and decimal.)
   b) $10.50_{10} = ?_2$ c) $6.0625_{10} = ?_2$
   d) $1010.1010_2 = ?_{10}$ e) $82_{16} = ?_{10}$

3. Convert and calculate the following in fixed-point form with: 8-bit word, 2's complement, and 3-bit precision,

   a) What are the *largest positive* and *largest negative* values? (Answer in binary and decimal.)
   b) $-10.50_{10} = ?_2$ c) $13.375_{10} = ?_2$ d) $-13.375_{10} = ?_2$
   e) $10101.010_2 = ?_{10}$ f) $A2_{16} = ?_{10}$
   f) in binary, calculate the result of the value in c) – the value in e)

4. Express the following decimal values in floating-point form with: 16-bit word, 7-bit exponent, and 8-bit mantissa,

   a) 0.0_{10} b) 1.0_{10} c) -0.5_{10} d) -5.62_{10} e) $1/64_{10}$

5. Express the following floating-point numbers in decimal ($\_\_{}_{10}$). (use floating-point structure as in question 4.)

   a) 0|000 0000|1000 0000
   b) 0|000 0010|1111 0000
   c) 1|111 1111|1010 0000
   d) 1|000 0000|1001 0100
   e) 0|111 1111|1100 0000
   f) What is the *largest number* and *smallest number* that can be stored in this FP format?

6. The following floating-point numbers are invalid, and consider NaN ("not-a-number") in floating-point representation. Indicate why.

   a) 0|000 0010|0101 0100
   b) 1|000 0000|0000 0000
   c) 0|000 0100|0000 0100
   d) 1|100 0000|0000 0001