CMPEN/EE 331 – Computer Organization and Design, Chapter 3 Review Questions

1. Assume 185 and 122 are signed 8-bit decimal integers stored in sign-magnitude format. Calculate 185 + 122. Is there overflow, underflow, or neither?

2. Assume 185 and 122 are signed 8-bit decimal integers stored in sign-magnitude format. Calculate 185 - 122. Is there overflow, underflow, or neither?

3. List four floating-point operations that cause NaN to be created?

4. Assuming single precision IEEE 754 format, what decimal number is represent by this word:

   1 01111101 00100000000000000000000

5. The floating-point format to be used in this problem is an 8-bit IEEE 754 normalized format with 1 sign bit, 4 exponent bits, and 3 mantissa bits. It is identical to the 32-bit and 64-bit formats in terms of the meaning of fields and special encodings. The exponent field employs an excess- 7coding. The bit fields in a number are (sign, exponent, mantissa). Assume that we use unbiased rounding to the nearest even specified in the IEEE floating point standard.
   a) Encode the following numbers to the 8 bit IEEE format
      1) $0.0011011)_{binary}$

      2) $16.0)_{decimal}$

   b) Perform the computation 1.011binary + 0.0011011binary showing the correct state of the guard, round bits and sticky bits. There are three mantissa bits.

   c) Decode the following 8-bit IEEE number into their decimal value: 1 1010 101

   d) Decide which number in the following pairs are greater in value (the numbers are in 8-bit IEEE 754 format):
      1) 0 1000 100 and 0 1000 111

e) In the 32-bit IEEE format, what is the encoding for negative zero?

f) In the 32-bit IEEE format, what is the encoding for positive infinity?

6. Using 32-bit IEEE 754 single precision floating point with one(1) sign bit, eight (8) exponent bits and twenty three (23) mantissa bits, show the representation of -11/16 (-0.6875).

7. What is the smallest positive (not including +0) representable number in 32-bit IEEE 754 single precision floating point? Show the bit encoding and the value in base 10 (fraction or decimal OK)

1. Assume 185 and 122 are signed 8-bit decimal integers stored in sign-magnitude format. Calculate 185 + 122. Is there overflow, underflow, or neither?

   Neither (65)

2. Assume 185 and 122 are signed 8-bit decimal integers stored in sign-magnitude format. Calculate 185 - 122. Is there overflow, underflow, or neither?

   Overflow (result =-179, which does not fit into 8-bit format)

3. List four floating-point operations that cause NaN to be created?

   Four operations that cause Nan to be created are as follows:

   (1) Divide 0 by 0
   (2) Multiply 0 by infinity
   (3) Any floating point operation involving Nan
   (4) Adding infinity to negative infinity

4. Assuming single precision IEEE 754 format, what decimal number is represent by this word:

   1 01111101 00100000000000000000000

   The decimal number

   = (-1)* (2^(125-127))*(1.001)2
   = (-1)*(0.25)*(1.125)
   = -0.28125

5. The floating-point format to be used in this problem is an 8-bit IEEE 754 normalized format with 1 sign bit, 4 exponent bits, and 3 mantissa bits. It is identical to the 32-bit and 64-bit formats in terms of the meaning of fields and special encodings. The exponent field employs an excess- 7coding. The bit fields in a number are (sign, exponent, mantissa). Assume that we use unbiased rounding to the nearest even specified in the IEEE floating point standard.

   g) Encode the following numbers to the 8 bit IEEE format

      1) $0.0011011)_{binary}$

         This number = $1.1011_2 * 2^{-3}$
         = 0 0100 110 in the 8-bit IEEE 754
         (after applying unbiased rounding to the nearest even)

      2) $16.0)_{decimal}$

         This number = $(1000.0)_2$
         = $(1.000)_2 * 2^{(11-7)}$

= 0 1011 000 in the 8-bit IEEE 754

h) Perform the computation 1.011binary + 0.0011011binary showing the correct state of the guard, round bits and sticky bits. There are three mantissa bits.

1.0110
+ 0.0011 (The least two significant bits are the guard and round bits)

1.1001

After rounding with sticky bit, the answer then is (1.101)2

i) Decode the following 8-bit IEEE number into their decimal value: 1 1010 101

The decimal value is (-1)* 1.625 *$(2)^{(10-7)}$ = - 13

j) Decide which number in the following pairs are greater in value (the numbers are in 8-bit IEEE 754 format):

1) 0 1000 100 and 0 1000 111

The first number = $2^{(8-7)}$*(1.5) = 1.5
The second number = $2^{(8-7)}$*(1.875) = 1.875
The second number is greater in value

k) In the 32-bit IEEE format, what is the encoding for negative zero?

The representation of negative zero in 32-bit IEEE format is
1 00000000 00000000000000000000000

l) In the 32-bit IEEE format, what is the encoding for positive infinity?

The representation for positive infinity in 32-bit IEEE format is
0 11111111 00000000000000000000000

6. Using 32-bit IEEE 754 single precision floating point with one(1) sign bit, eight (8) exponent bits and twenty three (23) mantissa bits, show the representation of -11/16 (-0.6875).

The representation of -0.6875 is:

1 01111110 01100000000000000000000

7. What is the smallest positive (not including +0) representable number in 32-bit IEEE 754 single precision floating point? Show the bit encoding and the value in base 10 (fraction or decimal OK)

The smallest positive representable number 32-bit IEEE 754 single precision floating point value is:
0 00000001 00000000000000000000000

- Its value is = $\pm 1.0 \times 2^{-126} \approx \pm 1.2 \times 10^{-38}$