

CSE 140 Homework One

January 8, 2013

*Only **Problem Set Part B** will be graded. Turn in only **Problem Set Part B** which will be due on January 23, 2013 (Wednesday) at 3:00pm.*

1 Problem Set Part A

- textbook 1.4
- textbook 1.5(b)(c)
- textbook 1.6(a)
- textbook 2.1(a)
- textbook 2.5(d)
- textbook 2.6(b)(d)
- textbook 2.7(a)(d)
- textbook 2.8(b)(d)
- textbook 2.9
- textbook 2.12(a)
- textbook 2.13(a)
- textbook 2.15(a)
- textbook 2.16(a)
- textbook 2.17(c)
- textbook 2.18(b)(c)
- textbook 2.19(a)

1 (Number Representation, Addition and Subtraction)

(Part A) Specify whether an overflow can occur in the operations on the incomplete 2's complement numbers listed in the following table where the x 's represent unknown values. Fill in the blanks with a "Yes" if you think the corresponding operation will generate an overflow; fill it in with a "No", otherwise.

Operations	Overflow?
$011xx0x1 + 1x1111x1$	
$01x10xxx - 100xxx1x$	
$111xx1xx + 1x1xx1x1$	
$0x11x00x + 01x1x001$	
$1x1x10xx - 1xxxx1x1$	

(Part B) Decimal numbers are usually encoded in the BCD format. For example, 57_{10} can be represented by two 4-bit BCD digits 0101 0111. To complete the addition/subtraction operations on these decimal numbers, a possible correction step might need to be applied to the result to attain the correct BCD code for the digit of the sum/difference being computed. The candidate correction operations include the following:

A Do nothing

B Add 3

C Add 6

D Add 10

The table below contains several arithmetic operations on 1-digit BCD codes where the x 's represent unknown values. For each case, fill in the box with the index of the appropriate correction operation.

	Correction operation
$0100 + 0111$	
$0xxx - 1xxx$	
$xx1x + 1xxx$	
$01xx - 00x1$	

(Part C) If $\alpha 1_{r^3} = r 2 \beta_{r^2}$, what are the values for the integers α , β and r ? (**Hint:** perhaps expanding the two sides of the given equation to a partial radix r representation may help you reason about this question more effectively.)

2 (IEEE754 Representation)

This question concerns the addition, subtraction, multiplication and division of floating point numbers in the IEEE754 single precision format. As we have learned in class, the value of a single precision IEEE754 floating point number is typically given by the following formula:

$$N = (-1)^S 2^{E-127} (1.M)$$

Therefore, the normalized representation of the number 3 is as follows:

Sign	Exponent	Mantissa
0	10000000	1000000 00000000 00000000

We provide as follows an overview of the algorithms for Add/Sub and Mult/Div operations, in case you need a refresher on the IEEE754 floating point arithmetic.

Add/Sub:

1. Right shift the mantissa of the number with the *lesser* exponent and increase that exponent until it matches the exponent of the larger number.
2. Depending on both the signs of the two numbers and the operation specified, either add or subtract the mantissas.
3. Adjust the resulting exponent E and mantissa M to the normalized form.

Mult/Div:

1. Depending on the operation specified, add/subtract the exponent and then subtract/add the bias.
2. Multiply/Divide the mantissa depending on the operation specified.
3. Adjust the resulting exponent E and mantissa M to the normalized form.

(Part A) Given one pair of numbers, N_1 and N_2 , represented in the IEEE754 single precision floating point format, you are asked to compute $N_1 + N_2$ and $N_1 * N_2$ and normalize your results according to IEEE754 rules. Please clearly show your final results by filling in each box of the table below.

	Sign	Exponent	Mantissa
N_1	1	10010001	1010000 00000000 00000000
N_2	1	10001111	1001000 00000000 00000000
$N_1 + N_2$			
$N_1 * N_2$			

(Part B) In the algorithms for floating point arithmetic operations, the last step is always a possible *normalization* of the resulting exponent (E) and mantissa (M). We list below two sets of candidate normalization operations to E and M for the last normalization step of the algorithms. However, only some of them can actually happen in the computation.

Operations on M “<<” = left shift, “>>” = right shift.	Operations on E
A. $M << 1$	a. $E + 1$
B. $M << 2$	b. $E + 2$
C. $M << n$, where $n > 2$	c. $E + n$, where $n > 2$
D. $M >> 1$	d. $E - 1$
E. $M >> 2$	e. $E - 2$
F. $M >> n$, where $n > 2$	f. $E - n$, where $n > 2$
G. Do nothing	g. Do nothing

Given a pair of IEEE754 single precision floating point numbers N_1 and N_2 , we are going to perform some operations on them. Unfortunately, only two pieces of information are available for these two numbers.

1. The signs of these two numbers are identical, denoted by u .
2. The first 4 bits of the exponent are already known, but the last 4 bits are unknown.

The following table shows the available information.

	Sign	Exponent	Mantissa
N_1	u	1011????	M_1
N_2	u	1010????	M_2

For the various cases specified below, fill in each box of the table with some combination of the operations on M and E which exactly describes **all possibilities** that can occur in that case. *Please note that multiple answers may hold for each case, in which case for full credit, please supply **all correct** answers.* As an example, the result of the first case has been filled in for you.

	Operations on M	Operations on E
$N_1 + N_2$	D, G	a, g
$N_1 + N_1$		
$N_1 - N_2$		
$N_1 * N_2$		
N_1 / N_2		

(Part C) As we have learned in class, for normalized floating point numbers, the standard single precision IEEE floating point representation includes a hidden ‘1’ right before the binary point, thus making the mantissa effectively 24 bits. In this part of the question, we will perform a slight modification to this representation by moving this hidden ‘1’ to the position right **after** the binary point. In other words, the value of a floating point number in this *modified* IEEE754 single precision format is given by the following formula:

$$N = (-1)^S 2^{E-127} (0.1M)$$

Therefore, the normalized representation of the number 3 in this slightly altered system would be as follows:

Sign	Exponent	Mantissa
0	10000001	1000000 00000000 00000000

Given two numbers N_1 and N_2 represented in this *modified* IEEE754 single precision format, you are asked to analyze the possible operations that may occur at the normalization step. The available information about the number and the set of possible operations on M and E are exactly identical as what has been listed on the previous page. For the various cases specified below, please perform the same analysis as in (Part B) and fill in each box of the table. Again, please supply **all possibly correct** answers for each case to get full credit.

	Operations on M	Operations on E
$N_1 + N_2$		
$N_1 + N_1$		
$N_1 - N_2$		
$N_1 * N_2$		
N_1 / N_2		