

CSE 140 Homework Two

February 1, 2008

*Only **Problem Set Two** will be graded. Turn in only **Problem Set Two** which will be due on Feb. 6, 2008 (Wednesday) at 11:00am.*

1 Problem Set One

- textbook 2.12(a)(c)
- textbook 2.17(b)(c)
- textbook 2.19(a)(d)
- textbook 2.24
- textbook 2.25
- textbook 2.29
- textbook 3.1(a)(d)
- textbook 3.3
- textbook 3.6(a)(c)
- textbook 3.7
- textbook 3.9

2 Problem Set Two

1 (Floating Point Number) The IEEE754 floating point number system is used extensively in computer arithmetic, but is not the only one that has been employed in processor implementations. Actually, IBM has used in their IBM System/360 computers a **hexadecimal floating point number system** that uses a similar approach to IEEE 754, but with the following differences.

- The 32-bit number is stored in a binary form, with **1 bit sign**, **7 bit exponent** and **24 bit mantissa**.
- The radix is **16**.
- There is **no** hidden bit in the mantissa.
- In the normalized form, the first (leftmost) digit of the mantissa should be a **nonzero hexadecimal** digit.
- The bias of the exponent is **64**.

We provide you below as an example the representation for the number -64 . This number would need to be expressed as -0.25×16^2 . The normalized hexadecimal radix floating point representation of this number is as follows:

Sign	Exponent	Mantissa
1	1000010	0100 0000 0000 0000 0000 0000

The arithmetic on this system is also similar to that of IEEE754 in that the last step of arithmetic operations is always a possible *normalization* of the exponent (E) and mantissa (M). We list two sets of candidate normalization operations¹ to E and M for the last normalization step of the algorithms. However, only some of them can really happen in the computation.

Operations for M	Operations for E
A. $M \ll 1$	a. $E + 1$
B. $M \ll 2$	b. $E + 2$
C. $M \ll 3$	c. $E + 3$
D. $M \ll 4$	d. $E + 4$
E. $M \ll n$, where $n > 4$	e. $E + n$, where $n > 4$
F. $M \gg 1$	f. $E - 1$
G. $M \gg 2$	g. $E - 2$
H. $M \gg 3$	h. $E - 3$
I. $M \gg 4$	i. $E - 4$
J. $M \gg n$, where $n > 4$	j. $E - n$, where $n > 4$
K. Do nothing	k. Do nothing

(Part A) Specify which operations can happen in the final normalization step of the **Add/Sub** algorithm and which operations cannot, by filling in each box of the table with some letter combination of the operations on M and E . *Please note that multiple answers may hold for each part, in which case for full credit, please supply all correct answers.*

Operations on M		Operations on E	
Can happen	Never happen	Can happen	Never happen

¹In the table, “ $\ll$ ” means left shift and “ $\gg$ ” means right shift.

(Part B) Given a pair of floating point numbers N_1 and N_2 in this representation, we are going to perform some operations on them. Unfortunately, only a part of information is available with these two numbers.

1. The signs of these two numbers are identical, denoted by u .
2. The exponents are known exactly.
3. The first eight bits of the mantissa are already known, but the last 16 bits are unknown.

The following table shows the available information.

	Sign	Exponent	Mantissa
N_1	u	1001001	1111 0010 ???? ???? ???? ???? ?
N_2	u	1001000	0110 0100 ???? ???? ???? ???? ?

For the various cases specified below, fill in each box of the table with some combination of the operations on M (Mantissa) and E (Exponent) which exactly describes all the possibilities that can occur in that case.

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

2 (Error Correction Code) Hamming Weight is defined as the number of 1's in a codeword. For example, in a 7-bit Hamming code (which consists of 3 parity bits and 4 information bits defined in the table below), codeword 0000000 has a Hamming Weight $HW(0000000)=0$. Similarly, $HW(0011001)=3$.

Position	7	6	5	4	3	2	1
Bit	$m0$	$m1$	$m2$	$p0$	$m3$	$p1$	$p2$

$$p0 = m0 \oplus m1 \oplus m2$$

$$p1 = m0 \oplus m1 \oplus m3$$

$$p2 = m0 \oplus m2 \oplus m3$$

1) Suppose c is a valid 7-bit Hamming codeword with $HW(c)=4$; what is the Hamming distance between c and codeword 1111111?

2) Can you find a 7-bit Hamming codeword with Hamming Weight 2? If yes, give an example; if not, explain your reason.

3) Write down all the 7-bit Hamming codewords of Hamming Weight 4.

3 (Boolean Algebra) This question concerns a Boolean Algebra of 8 variables, 0, A, B, C, D, E, F, and 1. We are given two operations “+” and “.”, with identity elements, 0 and 1, respectively, over the set {0, A, B, C, D, E, F, 1}. Two partially filled tables provide us some information about this particular Boolean Algebra, but not enough. You need to complete these two tables step by step, by applying the Boolean axioms and theorems we have discussed in class.

•	0	A	B	C	D	E	F	1
0								
A					?	?		
B		0			?		?	
C		0	0			?	?	
D		?	?	0				
E		?	0	?	A			
F		0	?	?	B	C		
1	0	A	B	C	D	E	F	1

+	0	A	B	C	D	E	F	1
0	0	A	B	C	D	E	F	1
A					?	?		
B					?		?	
C						?	?	
D		?	?	1				
E		?	1	?				
F		1	?	?				
1	1							

1) At the first step, you are asked to fill in the entries in the “.” table that are left blank, by applying **identity** and **commutativity**.

2) Based on the information you get from the first step, you are asked to fill in the entries in the “+” table that are left blank, by applying **De Morgan’s law**.

3) Again, based on the information you get from the first step, you can fill in the entries in the “+” table that are marked with “?”, by applying **absorption**.

4) At the last step, based on the information you get from the third step, you are asked to fill in the entries in the “.” table that are marked with “?”, by applying **De Morgan’s law**.