King Fahd University of Petroleum and Minerals
Department of Computer Engineering

COMPUTER ARCHITECTURE COE 308

Homework 2

Problems	Grading
1	
2	
3	
4	
TOTAL	

QUESTION 1: COMPUTER ARITHMETICS (Integer multiply) (10 points)

Consider the product of following signed integers (+6) * (-3) = 0110 * 1101. Carry out the multiplication of above numbers according to Booth algorithm as follows:

a. Initialize the product register to 0000 1101 0 and the multiplicand register to 0110,
b. Carry out the multiplication (Booth),
c. Show the result.

Solution:

Operation	Multiplicand	Product	next?
0. initial value	**0110**	**0000 1101 0**	**10 -> sub**
1a. P = P - m	**1010**	**+1010**	
		1010 1101 0	**shift P (sign ext)**
		1101 0110 1	**01 -> add**
		+0110	
	0110	**0011 0110 1**	**shift**
	0110	**0001 1011 0**	**10 -> sub**
	0110	**+1010**	
		1011 1011 0	**shift P (sign ext)**
		1101 1101 1	**11 -> nop, shift**
		1110 1110 1	
		1110 1110	**done (-18)**

The result is 1110 1110 which is -18.

QUESTION 2: FLOATING-POINT REPRESENTATION (10 points)

1. Consider the IEEE 754 Floating point standard representation.
    a) For what reason a bias of 127 is used in the representation of the exponent!
    b) Determine the range of real exponent allowed by the biased representation
    c) How to represent + and – infinities!
    d) How to represent a Not-a-Number!

2. Convert the decimal number 9.4 to the IEEE 754 Floating Point standard representation by considering only six digits to the right of the decimal point during the conversion operation. Make sure the final result is normalized.

Solution:

1. The IEEE 754 Floating point standard representation.
    a) Since a two's complement 8-bit exponent in the range of (-128, +127), a bias of +127 is added to the exponent in the IEEE 754 Floating Point standard representation with the image E = (-1, 0…, 127,…, 254, 255). The sub-range E = (1, …, 127, …, 254) is used for the FP exponent representation which facilitates comparison of FP numbers by just checking the exponent.
    b) The range of the exponent is (1, …, 127, …, 254) – 127 = (-126, …, 0, …, 127).
    c) The combination of E=255 and nil significant is used to encode infinity where + or – are encoded using the sign bit.
    d) The combination of E=255 and non-nil significant is used to encode Not-a-Number such as the result from SQRT (negative argument).

2. The IEEE 754 Floating point standard representation.

9 / 2 = 4	remainder 1	0.4 * 2 = 0.8
4 / 2 = 2	0	0.8 * 2 = 1.6
2 / 2 = 1	0	0.6 *2 = 1.2
1 / 2 = 0	1	0.2 * 2 = 0.4
		0.4 * 2 = 0.8
		0.8 * 2 = 1.6

This gives 9,4 = 1001.011001
To normalize the result 9,4 = (-1)^(0) * 1.001011001 * 2 ^(3)
The stored exponent is 3 + 127= 130
For which the representation is (0, 1000 0010, 0010110010….0)
Note that the exponent becomes unsigned integer.

QUESTION 3: IEEE 754 FLOATING-POINT STANDARD (10 points)

1. Consider the IEEE 754. The result of some binary arithmetic operation is 101110.01101*2^{11001010}, where 101110.01101 is the significand and 11001010 is the biased exponent. Normalize the above result and rewrite its normalized form.

2. The normalised result of some binary arithmetic operation is 1.11111110101*2^{10000100}. Round the above result to 7 bit significand storage and re-normalise if rounding produces a result that is not normalised. Rewrite its normalized form.

3. Add the two floating-point numbers and normalize the result.

Sign	Exponent	Significand
0	0111 1010	1010 1000 0000 0000 0000 000

Sign	Exponent	Significand
0	0111 1100	1101 0000 0000 0000 0000 000

Give the result of the addition in the IEEE 754 format given above.

Solution -Q1: The normalised result will be 1.0111001101*2^{11001111}.

Solution – Q2: The rounded result is 10.0*2^{10000100}. We need to re-normalise the result because the above rounding produced a result that is not normalised. Re-normalizing gives 1.0*2^{10000101}.

Solution – Q3: The answer is (-1)^0 * 1.00011101 * 2^{0111 1101} or

Sign	Exponent	Significand
0	0111 1101	0001 1101 0000 0000 0000 000