UNIVERSITY OF MASSACHUSETTS
Dept. of Electrical & Computer Engineering

Digital Computer Arithmetic
ECE 666

Mid-Term I
Sample Exam

Israel Koren
Fall 2010

1. Show the exact steps in the non-restoring division with the (negative) dividend X=1011001 in two's complement representation and the divisor D=0110.

2. (a) Show the representation of the following operands in the IEEE short format (use hexadecimal notation), perform the multiplication and show the final result in all four rounding schemes (nearest-even, toward zero (truncate), toward +∞, and toward -∞). (1+2^{-23}) × (1+2^{-22}).
Note: (1+2^{-23}) is the number 1.00000000000000000000001.

(b) Show the result of the following subtraction of numbers in the IEEE short format in all four rounding schemes. The operands are already given in the hexadecimal notation. 3F80 0000 - 3EFF FFFF

3. (4.10) Write down the post-normalization steps that might be needed when performing addition, subtraction, multiplication, and division with two floating-point operands in the IEEE short format. Indicate how many guard digits are needed in each case.

4. Prove that the optimal way to implement a two-level combinatorial shifter for k bits, where k=m^2, is for the first level to shift by multiples of m, and the second level to shift from 0 to m. Assume that the speed is proportional to the number of destinations for each line in the two levels. Can you generalize this result for any value of k?

5. (4.12) Two normalized floating-point numbers A and B in the short IEEE format were added, and the result was equal to A. Does this imply that B=0?

(b) (4.13) Given a floating-point number A with an exponent E_A (in any format), its successor has either the same exponent or the exponent E_A+1. Is the distance between A and its successor the same in both cases?

6. (4.14) (a) Compare the error involved in the serial evaluation of the product of four numbers, performed as $(((A_1 \times A_2) \times A_3) \times A_4)$ to that of its parallel evaluation performed as $((A_1 \times A_2) \times (A_3 \times A_4))$. Decide whether one of these methods has a smaller upper bound for the error when forming the product of n numbers.

(b) Repeat (a) for the sum of four numbers, then n numbers. Can we get lower error bounds if we know that the numbers are in some order; e.g., ascending order?