Date:________________

Homework for Chapter 3 Arithmetic for Computers

Name:____

1. This problem covers 4-bit binary multiplication. Fill in the table for the Product, Multiplier and Multiplicand for each step. You need to provide the DESCRIPTION of the step being performed (shift left, shift right, add, no add). The value of M (Multiplicand) is 1011, Q (Multiplier) is initially 1010.

Product	Multiplicand	Multiplier	Description	Step
0000 0000	0000 1011	1010	Initial Values	Step 0
				Step 1
				Step 2
etc ...				

2. This problem covers floating-point IEEE format.

(a) List four floating-point operations that cause NaN to be created?

(b) Assuming single precision IEEE 754 format, what decimal number is represent by this word:

`1 01111101 00100000000000000000000`

(Hint: remember to use the biased form of the exponent.)

3. 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-7 coding. 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 the 8-bit IEEE format:

(1) 0.0011011_{binary}

(2) $16.0_{decimal}$

(b) Decode the following 8-bit IEEE number into its decimal value: `1 1010 101`

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

(1) `0 0100 100` and `0 0100 111`

(2) `0 1100 100` and `1 1100 101`

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

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

4. The floating-point format to be used in this problem is a normalized format with 1 sign bit, 3 exponent bits, and 4 mantissa bits. The exponent field employs an excess-4 coding. 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 in the above format:

(1) 1.0_{binary}

(2) 0.0011011_{binary}

5. 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).

6. 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).

7. We're going to look at some ways in which binary arithmetic can be unexpectedly useful. For this problem, all numbers will be 8-bit, signed, and in 2's complement.

(a) For x = 8, compute x & (−x). (& here refers to bitwise-and, and − refers to arithmetic negation.)

(b) For x = 36, compute x & (−x).

(c) Explain what the operation x & (−x) does.