CSC 362 Homework Assignment #1

Due Date: Friday, August 30

**Shishir** **Ghorashainee**

Word Processor all answers (the tables in question 5 can be hand drawn). Each part of each problem (e.g,. 1a, 1b, 2a, 3b, etc) is worth 5 points. Show your work for a chance at partial credit – if you don’t show your work and you get an answer wrong, you get it all wrong! NOTE: remember to include minus signs for any negative numbers in 1, 2 and 6.

1. Convert 11001000 to decimal assuming the number is stored in each of the following representations:
   1. Unsigned magnitude = **200**
   2. Signed magnitude = -**72**
   3. One’s complement = -**55**
   4. Two’s complement = -**56**

1. Convert 1001100110001010 to decimal assuming that the number is stored in
   1. Signed magnitude = -**6538**
   2. Two’s complement = -**26230**

1. Convert -8632 from decimal to each of the following binary representations
   1. 16-bit signed magnitude = **1010000110111000**
   2. 16-bit one’s complement = **1101111001000111**
   3. 16-bit two’s complement = **1101111001001000**

1. Perform the following binary subtraction problem by converting the second number into its two’s complement negation and adding the results together. Both numbers are two’s complement so ignore any carry out produced by the addition.

01011101 – 00001110 = **01001111**

1. Do the following binary multiplication and division problems. Use the **tabular approach** (as covered in class, see the sample problems on the web site and power point notes). **If you do not show the work via the tabular approach, you will get no credit**. For a, d & e, the numbers are unsigned magnitude. For b & c, the numbers are two’s complement.

The multiplication problems use 5-bit numbers, the division problems use 6-bit numbers.

* 1. 11011 \* 11001 (use the unsigned multiplication algorithm) = **1010100011**
  2. 10101 \* 01010 (use Booth’s algorithm) = **0001101110**
  3. 11010 \* 11101 (use Booth’s algorithm) = **0000010010**
  4. 011110 / 000111 (use the unsigned division algorithm)

**Result: Q=000100, A = 000010**

**Remainder = 2 and Quotient = 4**

* 1. 110111 / 000110 (use the unsigned division algorithm)

**Result: Q=001001, A = 000001**

**Remainder = 1 and Quotient = 9**

1. Using the 14-bit floating point representation from chapter 2 (figure 2.2) where exponents are represented using excess-16, convert the following
   1. 01011011011001 to decimal **= 54.25**
   2. -13.3125 to binary = **11010011010101**
   3. 10111010110000 to decimal **= - 0.171875**
   4. 44.625 to binary **= 01011010110010**

1. Assume we are using even parity. We have a byte of 00001111 and a parity bit of 0. Is there an error? Yes, no or uncertain? Explain.

**= No, there’s not an error because the numbers of ones are even which means the parity bit will be 0.**

**If there are an odd number of 1’s with the parity bit of 0 or an even number of 1’s with parity bit of 1, then there would be an error.**

A paper with writing on it

Description automatically generated

A paper with writing on it

Description automatically generated

A paper with numbers and numbers

Description automatically generated

A white paper with writing on it

Description automatically generated

A piece of paper with writing on it

Description automatically generated

A piece of paper with writing on it

Description automatically generated

A paper with writing on it

Description automatically generated

A piece of paper with writing on it

Description automatically generated

A piece of paper with writing on it

Description automatically generated

A paper with writing on it

Description automatically generated

A white paper with writing on it

Description automatically generated

A paper with writing on it

Description automatically generated

A paper with writing on it

Description automatically generated