Introduction to Computer Science – Honors I

CS 181 – Fall 2013

Assignment 1 (due 9/11)

Complete the following exercises and submit your answers either on paper or (preferably) electronically in pdf format through Moodle.

Under the Stevens Honor System, you are required to sign the Honor pledge in all material that you hand in for grading. The Stevens Honor pledge is reported below:

"I pledge my honor that I have abided by the Stevens Honor System."

Please copy the pledge to the top of your answer sheet and sign your name. In the case of electronic submission simply enter your full name after the pledge.

Maximum points = 70. The last subject (8) gives you a 10-point bonus and it is optional. The bonus points can be used to cover any errors in your previous answers (1-7).

Subjects:

1. What are the 8 positive, odd integers immediately larger than **0100 0011 0111 0110**. Provide your response in binary. (8 points)

2. Transform the following numbers to the decimal numerical system: $(\mathbf{1001\ 1001\ 0001})_{2}$, $(\mathbf{652})_{8}$, $(\mathbf{31210})_{4}$, $(\mathbf{652})_{16}$, and $(\mathbf{b1c})_{16}$. Write down your calculations to produce the result. (Note: just the result in decimal will not be accepted as a correct answer.) (10 points)

3. Transform the following numbers to the binary system: $(\mathbf{24})_{10}$, $(\mathbf{500})_{10}$, $(\mathbf{1024})_{10}$, and the following to the hexadecimal system: $(\mathbf{68534})_{10}$, $(\mathbf{150421})_{10}$**.** (15 points)

4. (a) Negate **0101 0011 0111 0110** using 16-bit two's complement. Write down the result (3 points). (b) What are the 8 negative integers in binary, smaller than the negative integer calculated in (a) (8 points).

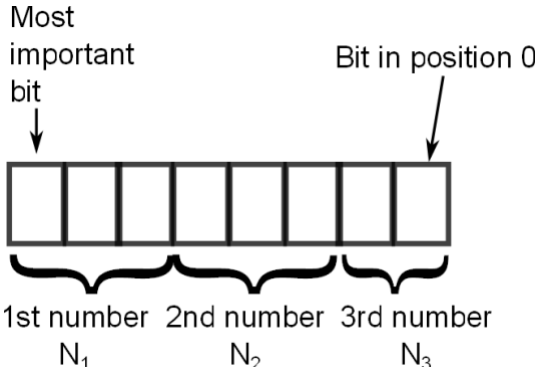
5. Transform the following negative integers to 16-bit binary numbers using two's complement notation: **-90, -967, -4095, -100** (12 points).

6. Add each of the following pairs of binary numbers. If the result falls outside the range representable in 16-bit two's complement notation, indicate this by writing "Overflow." (9 points)

```
  1100 1010 0110 0010        0110 1111 1010 1111      1110 1011 1100 1101
+ 0111 0000 1001 0110      + 0011 1010 1101 1111    + 1101 1111 1110 1110
```

7. Can overflow take place when negating a number? Explain. (5 points)

8. You need to store 3 unsigned integers, two 3-bits long, and one 2-bits long. Even though you do have multiple 8-bit registers, you want to store the numbers using a single register. You decide to use the following encoding:



Describe how you can recover the 3 numbers N_1, N_2, N_3 that are stored in an 8-bit register R using arithmetic and/or bitwise operations. If you use constants, write them down in hexadecimal format. (Bonus subject! 9 points)