605.611 - Foundations of Computer Architecture

Assignment 01

Sabbir Ahmed

February 1, 2021

1. Convert the following from decimal to binary. Express numbers in 16 bits.
   (a) 14
   (b) 97
   (c) 947
   (d) 32768 (What is special about this number that makes it easy to calculate?)
   (e) 32767 (What is special about this number and the last one that makes it easy to calculate?)
2. Convert the following numbers from binary to decimal
   (a) 0000 0000 0000 1010
   (b) 0000 0000 1001 1010
   (c) 0000 0001 1011 1100
3. Convert the following numbers from binary to hexadecimal
   (a) 0000 0000 0000 1010
   (b) 0000 0000 1001 1010
   (c) 0000 0001 1011 1100
4. Convert the following numbers from hex to binary.
   (a) 0x000C
   (b) 0x07AD

(c) 0xA49F

5. Convert the ASCII string "Java.2" to binary. You must include the character for the period.

6. Is the binary number 1001 1100 0111 1011 1101 1001 odd or even?

7. What is the base 10 value of the following two's complement numbers.

   (a) 01001 0111

   (b) 1101 1000

8. Add the following numbers in 4-bit two's complement format. Specify if there is an overflow or not.

   (a) 5+3

   (b) 7+5

   (c) 7+ (-5)

   (d) (-7) + (-5)

   (e) (-5) + (-3)

9. Using only the bitwise complement operation and addition, implement a function in the language of your choice to calculate the negative value of an int. The bitwise complement is the operator in Java and C/C++. The program should be properly formatted and documented, including:

   ☐ (a) A preamble statement that includes the name of the program, the purpose of the program, the author, and the data that it was written. The author attribute in Java can be documented using the @author Javadoc attribute.

   ☐ (b) Functions should include a purpose and any parameters and return values. The parameters and return values can be documented a using the @param and @return Javadoc attributes.

   ☐ (c) Proper style including variable naming, indentation, placement of brackets, etc.

☐ (d) Students unfamiliar with proper style should refer to: https://www.oracle.com/technetwork/ja
150003.pdf

10. Implement the following multiplication problems using only bit-shift operations and addition.

    (a) 7 * 4

    (b) 8 * 3

    (c) 9 * 13

11. Implement the division of 13/4 using bit shift operations. What happens to the remainder. What does this tell you about the efficient implementation integer division? Does this match what you know about integer division in Java or C/C++?

12. Can 14 / 3 be implemented using simply bit shift operations? Why or why not?

13. Write a program in Java or C/C++ to convert a character to upper case, regardless of whether it started as upper of lower case. This is simply OR'ing the byte for the character 0x20. The bitwise OR operation in Java or C/C++ is "|". The same rules of program style apply to the program as for Problem 9.

14. Write a program in Java or C/C++ to convert a character to lower case, regardless of whether it started as upper of lower case. This is simply AND'ing the byte for the character 0xCF. The bitwise AND operation in Java

15. Write a program in Java or C/C++ to convert a numeric value in a String to a valid integer. You must use the following algorithm, you may not use functions like "parseInt" or any other Java or C/C++ (or any language) libraries. The same rules of program style apply to the program as for Problem 9. The algorithm for this operation is:

```
int toInt(char[] inputNumber) {
  retval == 0;
  foreach character c in the inputNumber {
```

```
    retval = retval * 10;
    retval = retval + c - '0';
  }
  return retval;
}
```