**Name:**  Mark Demore  
  
**Section:** M1A  
  
**Documentation:** None  
  
Due Date: **Beginning of Class (BOC), Lesson 3**

**Authorized Resources**

This review is **individual effort** and will be counted as a quiz grade. You may seek help from any instructor and reference any publication in its completion. You may **not** reference ECE382 skills reviews from previous years. Normal documentation is required.

**Part 1: Software**

Install the Code Composer Studio (CCS) Integrated Development Environment (IDE) using the instructions from [Computer Exercise 1](http://ece.ninja/382/labs/compex1/index.html) - the "Installing Code Composer Studio (CCS)" block . Do not do anything else from the computer exercise.

This will not be checked, but **it must be accomplished by BOC on Lesson 5**. Failure to accomplish this will result in a **50% penalty** on this assignment.

**Part 2: Introduction to Numbering Systems**

Objectives:

* Explain, by means of an example, the idea of positional weighting in numbering systems.
* Convert an arbitrary base Q number to a decimal number.
* Convert a decimal number to a number with base Q.

1. (5pts each) Convert the following numbers to decimal numbers. **You must show your work for credit**.

a. D209 (base 16)   
  
D = 1101 2 = 0010 0 = 0000 9 = 1001

D209 = **53769**  
  
b. 40A7 (base 11)   
  
4 = 0100 0 = 000 A = 1010 7 = 0111  
  
40A7 = **5441**  
  
c. 672 (base 8)   
  
6 = 110 7 = 111 2 = 010  
  
672 = **442**  
  
d. 10101110 (base 2)   
  
1010 (0000) = 160 1110 = 14  
  
10101110 = **174**

1. (5pts each) Convert the following numbers to the base(s) indicated.

a. 547 (base 10) to base 2 and base 16   
  
547 / 2 = 273+1 … 547 = **10 0010 0011**  
  
0010 = 2 0010 = 2 0011 = 3 547 = **223**  
  
b. B36D (base 16) to base 2 and base 8   
  
B = 1011 3 = 0011 6 = 0110 D = 1101 B36D = **1011001101101101**  
  
001 = 1 011 = 3 001 = 1 101 = 5 101 = 5 101 = 5 B36D = **131555**

**Part 3: Representation of Negative Numbers**

Objectives:

* Understand the concept of a sign bit.
* Understand the concept and uses of two's complement numbers.

1. (5pts each) Find the 8-bit two's complement representation of the following numbers.

a. 116 (base 10)

116 / 2 = 58 … 116 = **01110100**  
  
  
  
  
b. -37 (base 10)

37 / 2 = 18 +1 … = 0010 0101  
 -37 = 1101 1010 +1 = **11011011**  
  
  
  
c. -2 (base 10)   
  
2 / 2 = 1 … = 10  
-2 = 01 + 1 = **10**

1. (5pts) What are the largest (most positive) and the smallest (most negative) 8-bit two's complement numbers in hexadecimal **and** decimal formats?   
     
     
   01111111 = 127 = 7F  
   10000000 = -127 = 80

**Part 4: Binary Addition and Subtraction, Concept of Overflow**

Objectives:

* Explain the concept of overflow and when it occurs.
* Know how to add and subtract two binary numbers and note whether an overflow occurs.

1. (5 pts each) All of the numbers below are represented using 8-bit, two's complement notation. Perform the specified operations and specify whether or not an overflow occurs.

a.  
    1110 1101  
+ 1010 1100  
  
**1 0001 1001 – overflow**  
  
  
  
  
b.  
  1001 1000  
- 1110 0111 = +0001 1001  
  
**1011 0001 - carry**

1. (5pts) When an overflow occurs, a common solution is to expand the number of bits used in the calculation (known as sign extension). How would you extend the negative 8-bit two's complement number with a hexadecimal value of BC to 16 bits?   
     
   B = 1011 C = 1100 BC = 10111100  
     
   16-bit BC = **1111 1111 1011 1100**
2. (10pts each) Perform the specified operations and specify whether or not an overflow occurs. Give the answer in unsigned 16-bit hexadecimal format.

a.  
  0xE76D (unsigned 16-bit)  
+     0x14 (2's comp 8-bit)  
  
E = 1110 7 = 0111 6 = 0110 D = 1101 1110 0111 0110 1101  
1 = 0001 4 = 0100 0001 0100  
  
**1110 0111 1000 0001**  
  
b.  
  BD72 (unsigned 16-bit)  
-     D6 (2's comp 8-bit)  
  
B = 1011 D = 1101 7 = 0111 2 = 0010 1011 1101 0111 0010  
D = 1101 6 = 0110 1101 0110  
  
**1011 1110 0100 1000**

**Part 5: Digital Logic**

Objectives:

* Know how to use bit masking to change registers.
* Demonstrate the ability to design a digital logic circuit.

1. (5 pts) You are given a 16-bit register called R6 that contains a certain value. You need to ensure that the 0th and the 6th bits are set (in other words, they have a value of one) in order for your program to function correctly, but you don't want to change any other bits in the register. Describe what operation needs to be performed and the operands involved in order to set these bits. Pseudocode will suffice.

add R6 and 65 but to avoid altering other bits can use BIS(.B) src, dst  
  
65 = 0100 0001

1. (5 pts) You forgot you needed to modify one other 16-bit register called R7 to allow your program to work correctly. In R7, the 0th, 6th, and 15th bits need to be cleared (set to zero) without affecting the other bits in the register. Describe what operation needs to be performed and the operands involved in order to clear these bits. Pseudocode will suffice.

BIC(.B) src, dst operate with 65601  
  
65601 = 1000 0000 0100 0001

1. (5pts) Draw a schematic for a circuit that will generate the following equation. You may use only the following gates: AND, OR, and NOT. **No simplification is allowed**.

F = X'Y' + YZ' +X'Y'Z

X  
  
  
Y F  
  
  
  
Z