CSCI 360 **Assignment 2: Binary, Hex and Absolute Addresses**
60 points

Print this document, carefully print, show your work and circle your answers. Scan your finished assignment and submit it on Blackboard. Or, if you prefer, you may also carefully type your work into this Word document, still showing your work by typing it in, and submit the finished Word document. If you type into this document, change the answer to bold red. Photographs of your finished assignment are ***not*** allowed.

1. Convert the following **unsigned binary** numbers to their decimal representations: (8 points)

```
                  Power of twos place     0 = No Digit Present
a. 110                      4 2 1 ->          4 2 0 = 6
b. 1101                   8 4 2 1 ->        8 4 0 1 = 13
c. 1101011       64 32 16 8 4 2 1 -> 64 32 0 8 0 2 1 = 107
d. 0101                   8 4 2 1 ->        0 4 0 1 = 5
```

2. Convert the following **unsigned hexadecimal** numbers to their decimal representations: (8 points)

```
a. 14                        (1*16) + (4*1)  = 20
b. C1      C=12              (12*16) + (1*1) = 193
c. CE9     E=14  (12*256) + (14*16) + (9*1) = 3305
d. B19     B=11. (11*256) + (1*16)  + (9*1) = 2841
```

3. Convert the following **unsigned decimal** numbers to both hex **and** binary representations: (8 points)

```
           Number subtracted with the power of twos place
a. 14                                        14 - 8-4-2 = 1110
b. 456                     456 - 256-128-64-0-0-8-0-0-0 = 111001000
c. 48                                   48 - 32-16-0-0-0 = 11000
d. 4095    4095 - 2048-1024-512-256-128-64-32-16-8-4-2-1 = 111111111111
           Number subtracted with power of 16s place
                                              14 = E
                 (1*256) and (12*16) and (1*8) = 1C8
                                              48 = 30
            4095 - (15*256) - (15*16) - (1*15) = FFF
```

4. Do the following **unsigned binary** arithmetic giving the answer in binary: (8 points)

```
a. 10110 + 01101 = 0100011
b. 11001 + 00101 = 0011110
c. 10110 - 01111 = 0000111
d. 11111 - 01101 = 0010010
```

5. Do the following **unsigned hexadecimal** arithmetic giving the answer in hexadecimal: (8 points)

```
a. 829D + 1A82 = 9D1F
b. E2C + A32   = 185E
c. FA28 - 3254 = C7D4
d. E2C - AB1   = 37B
```

6. Do the following arithmetic as if these were **five-bit signed representations** and indicate if overflow occurs and, if so, why. Note: Remember that you want to add. So, for signed subtraction, always convert the subtrahend (the number being subtracted) to its 2's complement and add it. Do this whether or not the subtrahend is negative OR positive and still check for overflow! (8 points)

a. `10110 + 01101`

```
10110
01101 = 110011 <- Overflow, carried bit into signed bit
```

b. `11001 + 00101`

```
11001
00101 = 11110 <- Overflow, carried bit into signed bit
```

c. `10110 - 01101` Twos complement the subtrahend, now becomes 10110 + 10011

```
10110
10011 = 11001 <- Overflow, carried bit into signed bit
```

d. `11111 - 01011` Twos complement the subtrahend, now becomes 11111 + 10101

```
11111
10101 = 110100 <- Overflow, carried bit into signed bit
```

7. Assume that

Register `0` contains `0007F144`
Register `1` contains `00000128`
Register `7` contains `EC0735C8`
Register `9` contains `00000C22`

If they are valid, calculate the absolute `D(X,B)` addresses for the representations below. If they are not valid, explain why. (12 points)

a. `56(,1)` **Valid: 56 in Hex is 38**

```
   000128
+     38 = 000160
```

b. `0(0,1,7)` **Invalid, you can have 1 or 2 registers, but not 3**

c. `6(7,0)` **Valid: 6 in Hex is 6**

```
   735C8
+     6 = 735D1
```

d. `12(9)` **Valid: read 12 (C in Hex) off of R9**

```
   000C22
+     C = 000C2E
```

e. `255(9,1)` **Valid: 255 in Hex is FF**

```
   000C22
   000128
+     FF = E49
```

f. `11(1,7)` **Valid: 11 in Hex is B**

```
   000128
   0735C8
+     B = 736FB
```