ECE 2504: Introduction to Computer Engineering
Homework Assignment 5 (60 points)

Make a reasonable effort to show your work. Clearly indicate your answers.

Problem 1 (12 points)
Express each of the following signed decimal numbers using **8-bit signed magnitude representation** and **8-bit 2's complement representation**.

	Signed-magnitude	2's complement
a. +41	00101001	00101001
b. +84	01010100	01010100
c. +103	01100111	01100111
d. -37	10100101	11011011
e. -88	10110001	10101000
f. -106	11101010	10010110

a) +41 = 00101001

41
-32
9
-8
1
-1
0

b) +84 = 01010100

84
-64
20
-16
4
-4
0

c) +103 = 01100111

103
-64
39
-32
7
-4
3
-2
1
-1
0

d) -37

37
-32
5
-4
1
-1
0

+37 = 00100101
Comp = 11011010
+ 1

2s Comp 11011011 = -37

Sm = 10100101 = -37

e) −88

+88 = 01011000

$$\begin{array}{r} 88 \\ -64 \\ \hline 24 \\ -16 \\ \hline 8 \\ -8 \\ \hline 0 \end{array}$$

So SM = 11011000

2s comp = 01011000

10100111

+ 1

10101000

f) −106

+106 = 01101010

$$\begin{array}{r} 106 \\ -64 \\ \hline 42 \\ -32 \\ \hline 10 \\ -8 \\ \hline 2 \\ -2 \\ \hline 0 \end{array}$$

So SM = 11101010

2s comp = 01101010

10010101

+ 1

10010110

Problem 2 (18 points)
Perform the following arithmetic operations in **binary** using **8-bit 2's complement representation**.

- Perform subtraction by finding the 2's complement of the subtrahend and then performing addition.
- Give an appropriate indicator of overflow, if it occurs. *Make sure that you identify the specific sign in your arithmetic that indicates overflow.*
- I did not choose the numbers in this problem by accident; where appropriate, you need not repeat your work to represent the values you use in solving this problem.

a. (+41) + (+84)
b. (-106) + (+41)
c. (-37) + (+83)
d. (+103) – (+41)
e. (+41) + (+103)
f. (-88) – (+84)

a) 00101001 = 41
+ 01010100 = 84
01111101

b) -106 = 10010110 (2s comp)
41 = + 00101001
10111111

c) -37 + 83
+37
= 00100101
-37 = 11011011
+ 01010011
1 00101110
over flow

d) 41 = 00101001
comp = 11010110
+ 1
11010111 = -41
01100111 = 103
+ 11010111 = -41
1 00111110
overflow

e) 41 + 103

```
  00101001   = 41
+ 01100111   = 103
-----------
  10010000
```

f) +88

```
        01011000
comp =  10100111
      +        1
      ----------
        10101000
```

+84

```
        01010100
comp =  10101011
      +        1
      ----------
        10101100
```

```
   10101000
 + 10101100
 ----------
  101010100
```

Problem 3 (8 points)

Represent the number $(+26.6875)_{10}$ as a 16-bit floating point number. This particular floating point format, uses signed-magnitude to store numbers in the mantissa (a normalized fraction) and the exponent (an integer). The mantissa has 10 bits (including the mantissa's sign bit) and the exponent has 6 bits (including the exponent's sign bit).

26
−16
10
−8
2
−2
0

= 1 1010.1011
= 26.6875

.6875 x2 = 1
.375 x2 = 0
.75 x2 = 1
.5 x2 = 1

11010.1011

$= .110101011 \times 2^5$

S	mantissa	S	exponent
0	110101011	0	00101

10 bits | 6 bits

= 0110101011000101 for FP

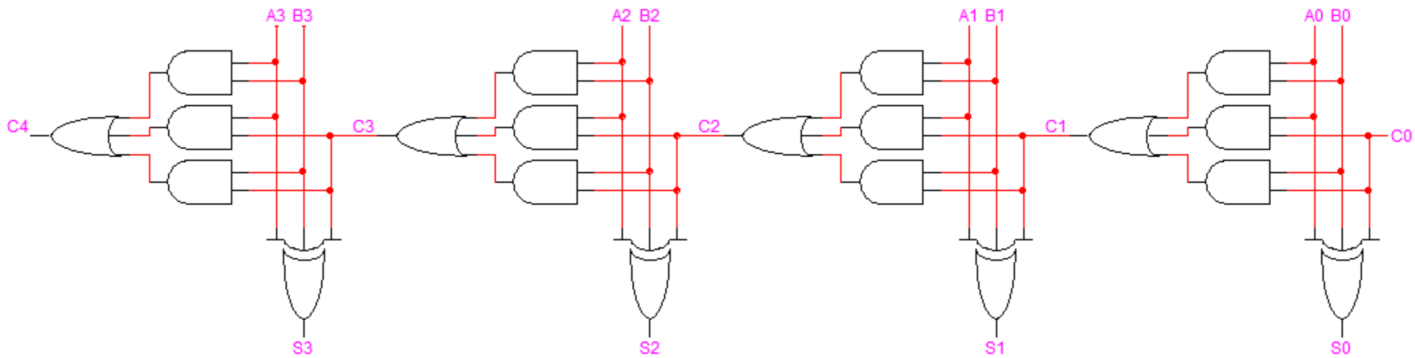
Problem 4 (2 points)
One day, I overheard Michael Mistake-Maker expressing wondrous amazement about binary floating point numbers:

"You really can get something for nothing! For a particular number of bits, you can define a floating-point representation that gives a greater range and a greater precision than the same number of bits give in a fixed-point representation. Why would you ever use a fixed-point representation?"

Based on your own knowledge or upon research, gently deflate Michael's enthusiasm. I am not looking for essay-length answers; a few sentences will suffice. Answer the question in your own words, and briefly cite any sources that you use.

Problem 5 (20 points)
Use the “design by contraction” strategy to derive the equations that implement a 4-bit increment-by-3 circuit from the equations that implement a standard adder.

If it helps to visualize the circuit, imagine that you are starting with a 4-bit adder:



Your strategy should consist of choosing specific values for B (B3 B2 B1 B0) and C0 that will cause the output S (S3 S2 S1 S0) to always equal A (A3 A2 A1 A0) plus 3, and then using Boolean algebra to reduce the equations that contain constants.