**Student Info (1 pt)**
Name: _____
Section: _____
netID: _____

# CS/ECE 252 Introduction to Computer Engineering

Spring 2017
Instructor: Radhakrishnan Rahul Nayar
TAs: Annie Lin, Mohit
URL: http://pages.cs.wisc.edu/~rrahulnayar/cs252/Spring2017/index.html

# Homework 2 [Due at lecture on Wed, Feb 1]

Primary contact for this homework: Annie [elin23 at wisc dot edu]
You must do this homework **alone**. Please staple multiple pages together.

Problem 1 (2 pts):
Convert the following IEEE floating point number into decimal. **Show your work for full credit.**
`1 10000000 10101000000000000000000`

`-1 * 2^(128-127) * 1.65625 = -3.3125`

Problem 2 (2 pts):
Given the following decimal numbers, complete the table with the respective **8-bit** conversions.

| **Decimal** | **2's complement** | **1's complement** |
|---|---|---|
| 41 | 00101001 | 00101001 |
| -41 | 11010111 | 11010110 |

Problem 3 (2 pts):
Show the result of the following operations.

(a) `(0110 AND 1001) AND NOT(1110)`

`0000`

(b) `NOT(1011 AND 1010) OR (0010 OR 1110)`

`1111`

Problem 4 (4 pts):
Add the following **2's complement** numbers together. **Show your work for full credit.**

a) 1010 + 010101

```
 010101
+111010
---------
 001111
```

b) 0100 + 1100000

```
 0000100
+1100000
--------
 1100100
```

c) Why is sign-extension important when performing arithmetic with 2's complement numbers?

Sign-extension allows you to increase the number of bits while preserving the original value and sign. If we add only 0's to the left of a negative number, then it becomes an even number in two's complement.

d) For part (a), check your answer by translating the operands and results into decimal.

1010 = -6
010101 = 21
21 - 6 = 15 = 001111

Problem 5 (3 pts):

a) In 2's complement, how many distinct numbers can be represented using 16 bits?

2^16 = 65536

b) What is the largest **unsigned** integer that may be represented using 16 bits?

2^16 - 1 = 65535

c) What is the **minimum** number of bits is required to represent the number -135 in 2's complement?

9 (1 01111001)

Problem 6 (2 pts):
For the following **8-bit 2's complement numbers**, perform the stated operation. **Show your work for full credit.**

(a) 01110000 - 00101000

-00101000 = 11011000
01110000 + 11011000 = 01001000

(b) 01010001 + 01010101
0 10100110

(c) Which of the above (if any) create overflow?
(a): No overflow, as the sum of a positive and negative number is always between the addends.
(b): Overflow because the sum of two 8-bit positive numbers creates an 8-bit negative number, which is obviously incorrect. We need an extra bit in front to denote the positive sign.

Problem 7 (2 pts):
Convert the following to their hex equivalent.

a) The decimal number 21

00010101 -> 15

b) The ASCII string “cat” (do not include quotation marks)
63 61 74