UNCA CSCI 255
Exam 1 Fall 2013
9 October, 2013

This is a closed book and closed notes exam. It is to be turned in by 11:15 AM. Calculators, PDA's, cell phones, and any other electronic or communication devices may not be used during this exam.
If you want partial credit for imperfect answers, explain the reason for your answer!

Name:_______________________________

Problem 1 (12 points) Decimal to two's complement conversion
Convert the following six signed decimal numbers into eight-bit two's complement representation. Some of these numbers may be outside the range of representation for eight-bit two's complement numbers. Write "out-of-range" for those cases.

-100	-17
-2	17
42	200

Problem 2 (8 points) Two's complement to decimal conversion
Convert the following four six-bit two's complement numbers into signed decimal representation.

000011	001100
110000	110011

Problem 3 (12 points) Adding
Add the following pairs of six-bit two's complement numbers **and indicate which additions result in an overflow by writing one of "overflow" or "no overflow" in each box**. You must write either "overflow" or "no overflow" in each box in addition to the result of the addition.

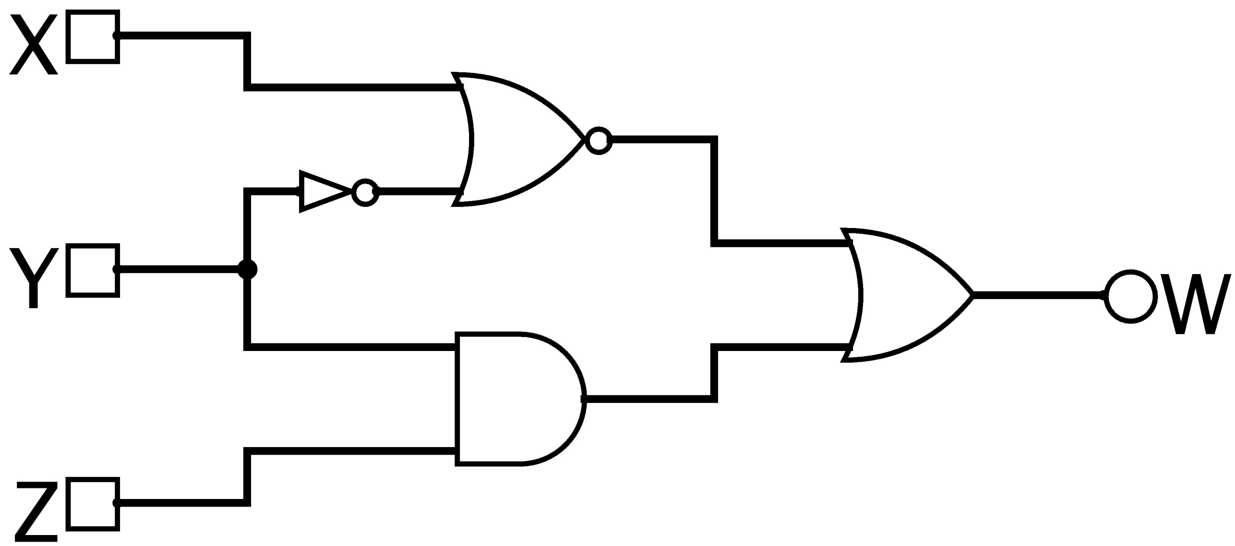
000101 + <u>000001</u>	011010 + <u>000110</u>
011011 + <u>001001</u>	001011 + <u>110100</u>
111011 + <u>101001</u>	111111 + <u>111111</u>

Problem 4 (8 points) Memories
An 8 MB memory has a 16-bit word size. How many words are contained in this memory?

A memory has 512 k words. Each word contains 32 bits. How many bytes are contained in this memory?

How many bits are required to address the 512 k words of this memory?

Problem 5 (13 points) Digital logic to truth table
A gate-level circuit is shown below with three inputs on the left and a single output on the right. Complete the truth table so that it corresponds to this digital logic circuit.



X	Y	Z	W
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Problem 6 (5 points) Digital logic to Boolean expression
Write a Boolean expression that corresponds to the logic circuit shown in Problem 5. You can build on your Problem 5 answer if that seems appropriate.

Problem 7 (13 points) Truth table to digital logic
Draw a logic circuit at the gate level that will implement the following truth table, where X, Y, and Z are inputs and W is the single output.

X	Y	Z	W
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Problem 8 (5 points) Truth table to Boolean expression
Write a Boolean expression that corresponds to the truth table shown in Problem 7. You can build on your Problem 7 answer if that seems appropriate.

Problem 9 (9 points) Boolean expression to truth table
Complete the truth table on the left below so that it corresponds to the following Boolean equation

$$X = A\,\overline{B}\,C + \overline{A}\,C$$

If you prefer that your inversions be primes, you can think of the equation as

$$X = A\,B'\,C + A'\,C$$

Or, if you really like Java conditional expressions, you can go with

```
X = A && !B && C || !A && C
```

Problem 10 (6 points) Boolean expression to digital logic
On the remainder of this page, draw a logic circuit at the gate level that will implement the Boolean equation given in Problem 9. You can build on your Problem 9 answer if that seems appropriate.

A	B	C	Z
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Problem 11 (6 points) Boolean expression to digital logic
A finite state machine (FSM) is described by the following state table.

present state	input	next state
α	0	α
α	1	β
β	0	β
β	1	β
γ	0	α
γ	1	γ

Assuming that the finite state machine is started in state α and the finite state machine is given the input sequence `0 0 1 0 0 1 0 0 1`, what state is the finite state machine in after each input?

Just fill in the blanks with the state entered after the input immediately to the left of the blank arrives. Keep in mind the state machine is not reset after each input. It makes a transition for the preceding state.

You do need to write any outputs. They aren't even given in the problem description.

α 0 _ 0 _ 1 _ 0 _ 0 _ 1 _ 0 _ 0 _ 1 _

Just to be clear on this. The first blank on the right above should be filled in with the new state that the finite state machine enters after it receives a 1 when the present state is α.