UNCA CSCI 320
Exam 2 Spring 2014
22 April, 2014

This is a closed book and closed notes exam. It is to be turned in by 3:00 PM.

Communication with anyone other than the instructor is not allowed during exams. Furthermore, calculators, PDA's, cell phones, and any other electronic or communication devices may not be used during exams.

To ensure compliance with these rules, anyone needing a break during exams must leave their exam with the instructor. The only allowed break excursion is a direct walk to and from the classroom and the restrooms outside RRO 239 (the ATMS classroom). Cell phones or computers may not be used during breaks.

Violation of these rules will result in a grade of 0 for this exam.

Name:_____

Problem 1: IEEE 754 to floating point (10 points)
Consider 3FC00000 to be the hexadecimal representation of a 32-bit IEEE floating point number. Translate that number into a C/Java-style floating point number. You must show your work to get credit for this problem.

Problem 2 (20 points) Memory caches
Suppose a computer has 20-bit addresses. (Pretend it's one of the original IBM PC's.) Show how addresses are broken into tag, set (if any), and block offset fields for a **16 k byte cache** for the following **three** cache structures.

Use the 20-bit hexadecimal address 240AC to illustrate your answer.

1: *A direct cache with blocks of size 256 bytes.*

2: *A 4-way cache with blocks of size 64 bytes.*

3: A fully associative cache with blocks of size 1024 (or 1 k) bytes.

In case you've forgotten, the geometric progression for powers of 2 are

1	2	4	8	16	32	64	128	256	512

Problem 3 (10 points) Truth table to Boolean expression
On the right below is a Boolean function. Write the equivalent boolean expression for this function on its left.

G	P	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Problem 4 (10 points) Truth table to SystemVerilog
Now write a SystemVerilog module that implements the truth table seen in Problem 4. You may just want to build on your Problem 3 answer.

Call your module `problem4` and be sure to complete the `module` header.

```
module problem4(
```

Problem 5 (5 points) Testing from VSIM — Combinatorial

Suppose you were debugging this circuit using `vsim`, like we did in the lab. What commands would you type (in `vsim`'s `transcript` window) to set all three inputs of your module to 1.

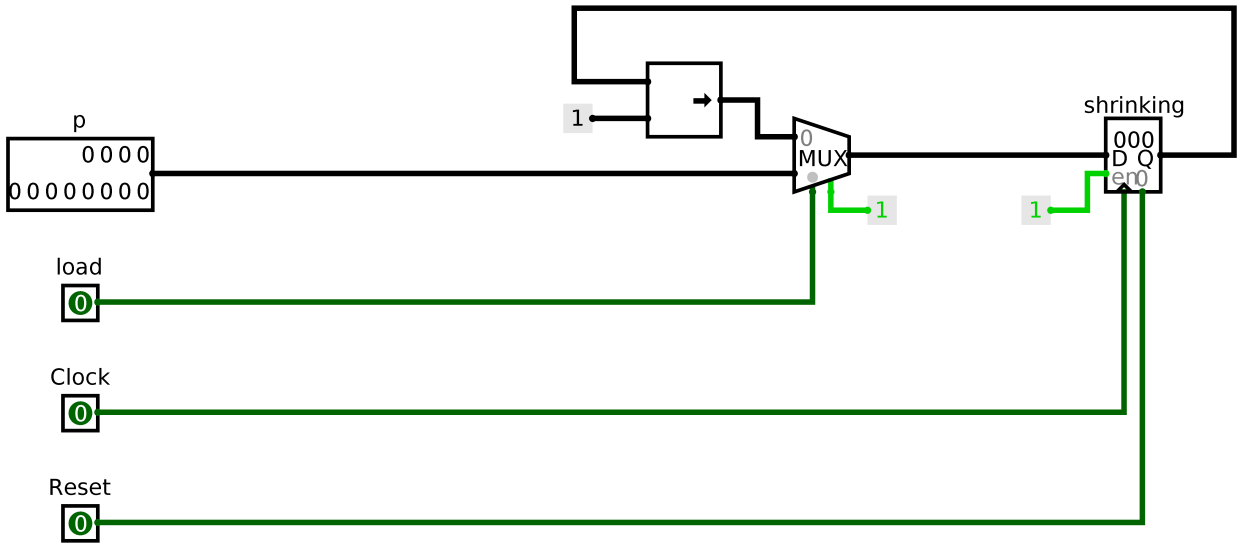
Use the lined paper provided by the instructor for the remaining problems.

Problem 6 (15 points) FSM
Take a look at the finite state machine called `patternMoore` that is described on the second page (example 4.31) of the "Example SystemVerilogs programs" handout distributed for this exam.

Draw a finite state machine diagram (that's the one where labeled arrows connect states) for this module.

Problem 7 (10 points) Testing from VSIM — Sequential
Suppose you were debugging the `patternMoore` circuit using `vsim`. What commands would you type (in `vsim`'s `transcript` window) to make the FSM output the value 1. I estimate that at least eight commands are needed. If you are unsure of your answer (or the `vsim` commands), explain what you are trying to do.

Problem 8 (20 points) SystemVerilog Datapath
Below is a drawing of a circuit that should be familiar from Homework 6.



The circuit has a 12-bit *input* called `p`, three 1-bit *inputs* called `load`, `clock`, and `reset`, and a 12-bit *output* called `shrinking`. Here is what it does:

- When `reset` is 1, `shrinking` is set to 0.
- When `reset` is 0 and `load` is 1, `shrinking` is set to `p` on the *positive edge* of `clock`.
- When `reset` is 0 and `load` is 0, `shrinking` is shifted right one place on the *positive edge* of `clock`.

In this problem, write a SystemVerilog module that implements the circuit. The module doesn't require any internal variables, just the four input and one output variable mentioned above.