UNCA CSCI 320
Exam 1 Spring 2016
3 March, 2016

This is a closed book and closed notes exam. It is to be turned in by ~6:25 PM.

Communication with anyone other than the instructor is not allowed during the exam. Furthermore, calculators, cell phones, and any other electronic or communication devices may not be used during this exam. Anyone needing a break during the exam must leave their exam with the instructor. Cell phones or computers may not be used during breaks.

If you want partial credit for imperfect answers, explain the reason for your answer!

Name:_____

Problem 1 (8 points) Decimal to two's complement conversion
Convert the following four signed decimal numbers into six-bit *two's complement* representation. Some of these numbers may be outside the range of representation for six-bit two's complement numbers. Write "out-of-range" for those cases.

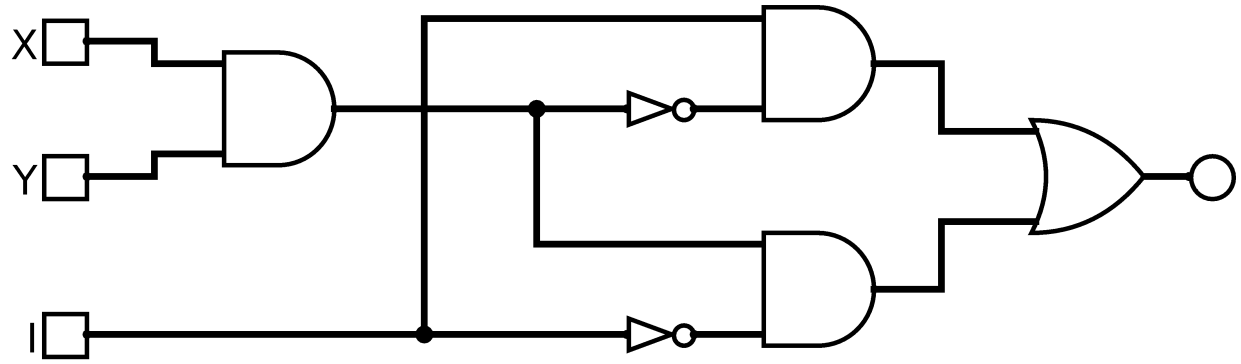
-6	320
-20	20

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

111111	100100
000010	011011

Problem 3 (13 points) Digital logic to truth table
A gate-level circuit is shown below with three inputs, X, Y, and I on the left and a single output on the right. Complete the truth table so that it corresponds to this digital logic circuit.

This circuit will be used two more times in this exam, so be sure you get the right answer.

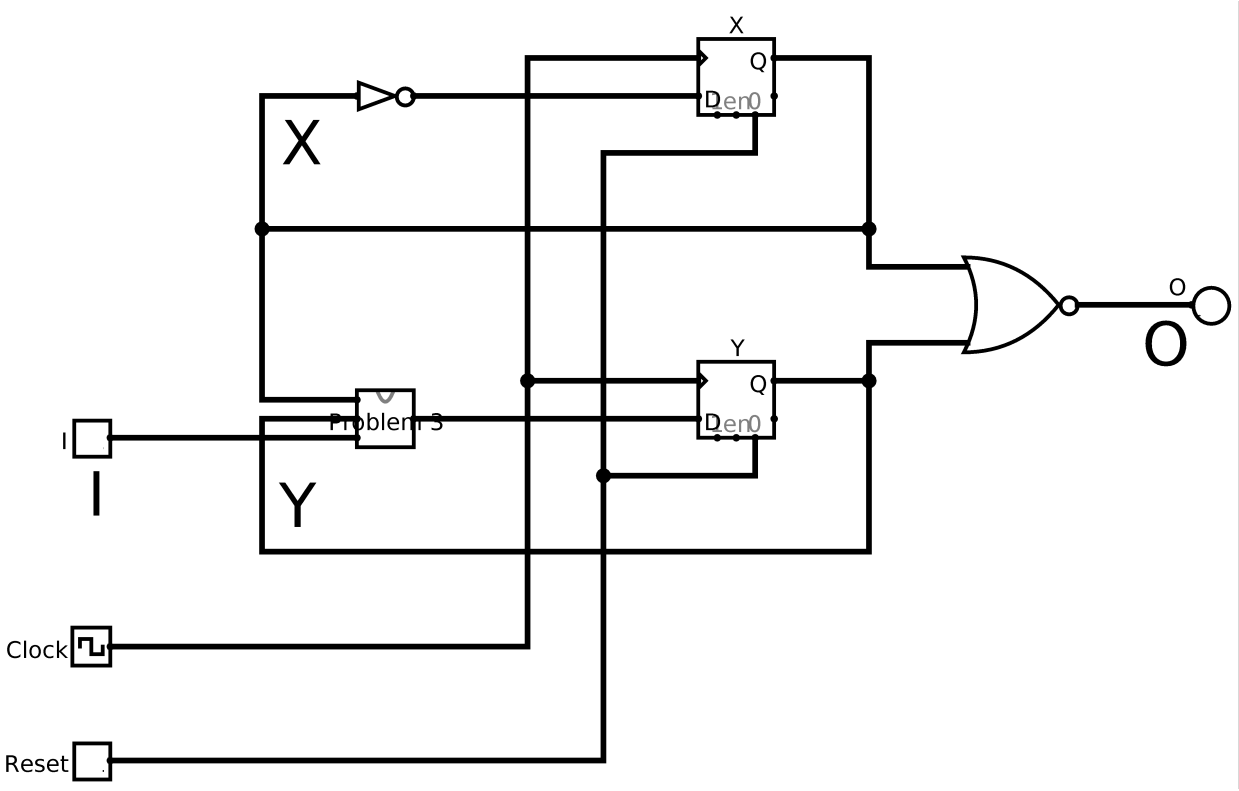


X	**Y**	**I**	*output*
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Problem 4 (5 points) Digital logic to Boolean expression
Write a Boolean expression that corresponds to the logic circuit shown in Problem 3. You can build on your Problem 3 answer if that seems appropriate.

From sequential circuit to FSM diagram and SystemVerilog module

Below is a sequential circuit that implements a finite state machine. You are going to use it in the next few questions.



Here are some relevant facts this circuit.

- The circuit has an external input labeled `I`.
- The circuit has an external output labeled O.
- The circuit has an external clock labeled Clock.
- The circuit has an eternal unclocked input to initiate the FSM labeled `Reset`.
- The circuit has two internal D flip-flops for its state. These are labeled X and Y.
- The D flip-flops are asynchronously initialized to 0 when `Reset` is asserted.
- The D flip-flops are triggered on the positive edge of the clock. (This really makes no difference in this problem.)
- The circuit element labeled Problem 3 is an “instance” of the combination circuit of Problem 3. The labels X, Y, and `I` are used similarly in both the “parent” and “child” circuits. (That is, the order is X, Y, and `I` from top-to-bottom in both.) The output of the combination circuit is on its right.

Answer the questions on the next few pages regarding circuit.

Problem 5 (8 points) Next State Table using 0 and 1
Complete the binary-level next state table for this circuit. Half of your answer is a repeat of your answer for Problem 3. The other half is even simpler.

present state		*input*	*next state*	
X	**Y**	**I**	**X'**	**Y'**
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 (6 points) Output table using 0 and 1
In the space below, draw and fill it an output table for the FSM. (The faint lines are to help you make the table look neat. You may use a different number of rows.)

Problem 7 (6 points) State encoding
In the space below, create a table giving a state encoding for the FSM. You will need to choose names for your states.

Problem 8 (8 points) Next state table using state names
Using your state names, create a next state table for the FSM.

Problem 9 (4 points) Output table using your state names
Using your state names, create an output table for the FSM.

Problem 10 (12 points) FSM diagram
Now draw the FSM diagram. Please be neat.

Problem 11 (12 points) SystemVerilog program
Write a module in SystemVerilog to implement the FSM by *filling in the blanks*.

```
module prob_11
      (input    logic                    clk,
       input    logic                    reset,
       input    logic                    i,
       output   logic                    o) ;


   typedef enum logic [1:0] {__________________} statetype ;
   statetype pres_state, next_state ;

   always_ff@(posedge clk, posedge reset)

      ____________________________________________________

      ____________________________________________________

      ____________________________________________________

      ____________________________________________________

      ____________________________________________________


   always_comb
     case (pres_state)

      ____________________________________________________

      ____________________________________________________

      ____________________________________________________

      ____________________________________________________

      ____________________________________________________

      ____________________________________________________

      ____________________________________________________

      ____________________________________________________

      ____________________________________________________

      ____________________________________________________

     endcase

   assign ____________________________________________;
endmodule
```

Problem 12 (10 points) Contamination and propagation delays
The table below gives the contamination and propagation delays of three gates.

Gate	Contamination	Propagation
NOT (inverter)	10 ps	15 ps
Two-input AND	25 ps	30 ps
Two-input OR	30 ps	40 ps

What is the contamination and propagation delays for the circuit shown in Problem 3. (Show your work by drawing some numbers on the connections between gates.)