UNIVERSITY OF TORONTO

Fall 2013 Midterm

CSC258: Computer Organization

Duration: 1.5 hours

October 24th, 2013

Last Name: _____

First Name: _____

Student Number: _____

Instructor: **Steve Engels**

Instructions:

- **Write your name on the back of this exam paper.**
- **Do not open this exam until you hear the signal to start.**
- Have your student ID on your desk.
- No aids permitted other than writing tools. Keep all bags and notes far from your desk before the exam begins.
- There are 4 questions on 8 pages. When you hear the signal to start, make sure that your exam is complete before you begin.
- Read over the entire exam before starting.
- If you use any space for rough work or have to user the overflow page, clearly indicate the section(s) that you want marked.

Mark Breakdown

Part A:	**/ 16**
Part B:	**/ 22**
Part C:	**/ 10**
Part D:	**/ 12**
Total:	**/ 60**

Part A: Short Answer (16 marks)

Answer the following questions in the space provided. When providing a written answer, write **as clearly and legibly as possible**. Marks will not be awarded to unreadable answers.

1. The depletion layer of a pn junction causes electrons to move from the p side of the junction to the n side, due to the electric field. What is the name of this current? **(1 mark)**

a) Distribution **b)** Drift

c) Diode **d)** Diffusion

2. The depletion layer of a pn junction causes holes to move from the n side of the junction to the p side, also due to the electric field. What is the name of this current? **(1 mark)**

a) Distribution **b)** Drift

c) Diode **d)** Diffusion

3. In the space below, write the 8-bit signed binary value for -20. **(1 mark)**

4. What is the largest value that an 8-bit signed binary number can have? **(2 marks)**

Binary value: _____ Decimal value: _____

5. Recall the lab where you displayed 4-bit BCD numbers on the seven-segment display. In the space below, assuming BCD inputs of `X[3]`, `X[2]`, `X[1]` and `X[0]`, write Verilog code that expresses the logic for segment `0` of `HEX0` for the seven-segment display. For full marks, this must be a single line, and may not have any `if` or `case` statements. **(4 marks)**

6. Which of the following Verilog statements set output Y high when inputs A and B are both high? Circle all that apply. **(2 marks)**

a) `assign Y = A && B;`

b) `and(Y, A, B);`

c) `assign Y = A || B;`

d) `assign Y = A & B;`

7. Given inputs A, B, C and D, which of the following are valid minterms? **(2 marks)**

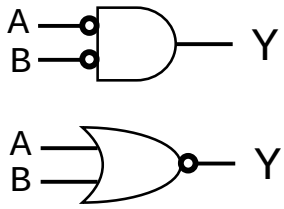
a) $A \cdot B \cdot C$

b) $A \cdot B + C \cdot D$

c) $D \cdot C \cdot B \cdot A$

d) $\overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D}$

8. What is the name of the law that declares the following gates logically equivalent? **(1 mark)**



9. Why are ripple counters considered to be asynchronous? Circle one. **(1 mark)**

a) Each T flip-flop is triggered by the clock signal

b) Not all of the T flip-flops are triggered by the clock signal

c) None of the T flip-flops are triggered by the clock signal

d) None of the above

10. Why are ripple carry adders considered to be asynchronous? Circle one. **(1 mark)**

a) Each adder units is triggered by the clock signal

b) Not all of the adder units are triggered by the clock signal

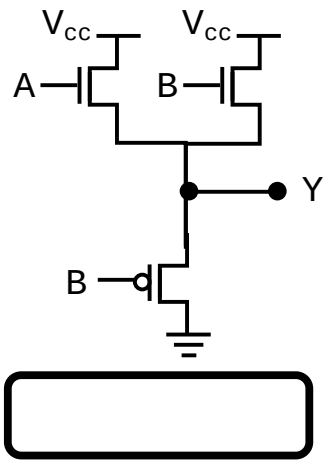
c) None of the adder units are triggered by the clock signal

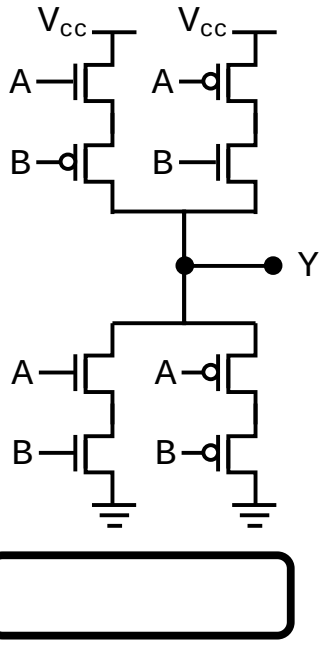
d) None of the above

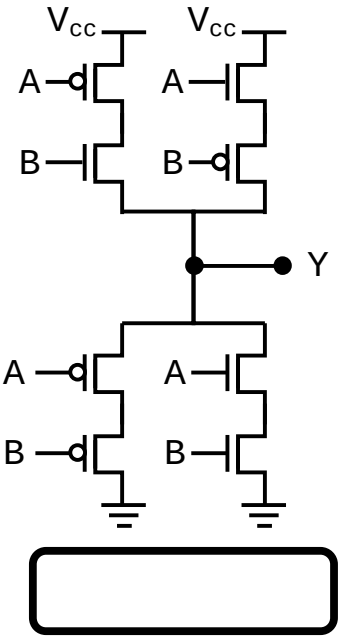
Part B: Slightly Longer Answer (22 marks)

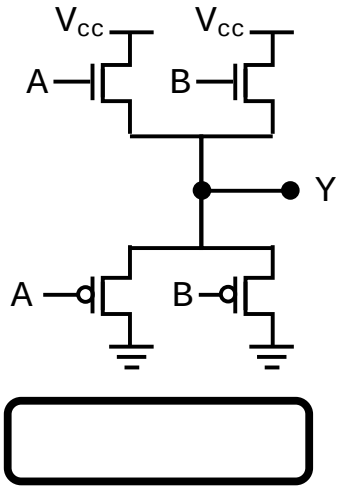
Answer the following questions in the space provided. The final answer is all that is necessary, but showing your work can help if your final answer isn't correct. Again, make sure to write legibly here.

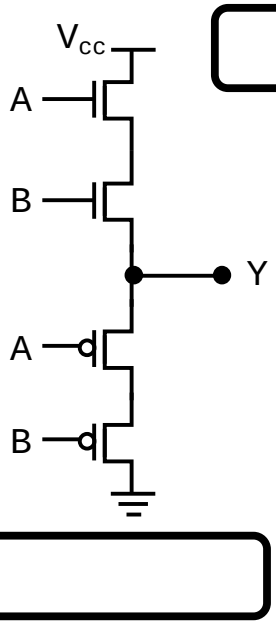
1. For each of the transistor circuits shown below, provide the name of the logic gate that the circuit implements in the blank space below each diagram. If the circuit does not implement a valid gate, write "INVALID" instead. **(12 marks)**

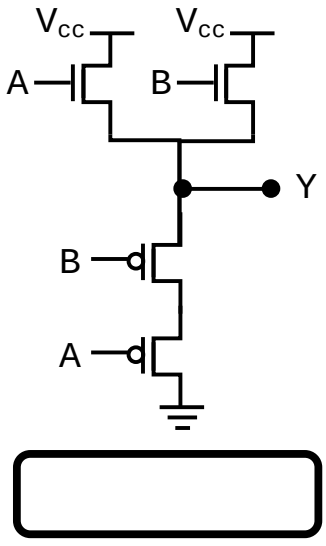


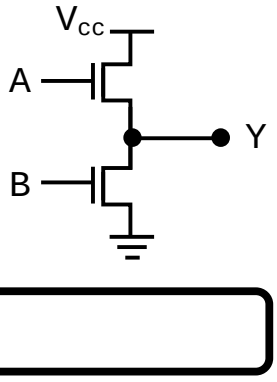


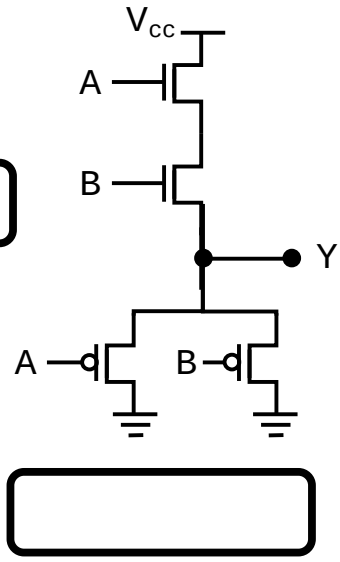




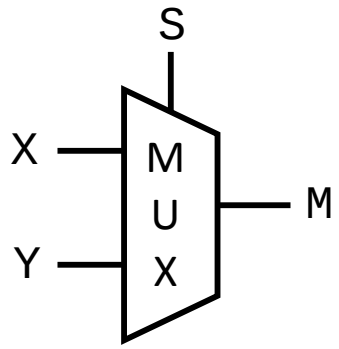




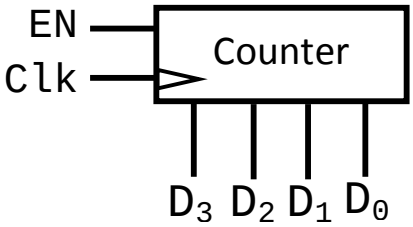




3. Consider the two-input multiplexer on the right. In the space below, show how a **four-input multiplexer** with inputs A, B, C, D and select bits S_1 and S_0 can be implemented as a combination of these two-input multiplexers. For full marks, no other gates or devices should be used in your solution. **(5 marks)**



4. Consider the 4-bit counter device on the right. In the space below, show how an 8-bit counter could be constructed out of these 4-bit counters. You may use other gates and devices in your design, but for full marks you should only use one or two extra gates. Be sure to label the input and output pins of your design. **(5 marks)**



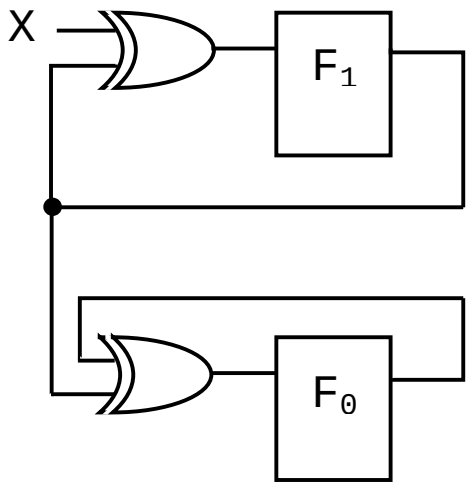
Part C: Circuit Design and Analysis (10 marks)

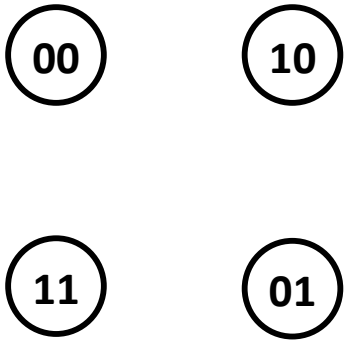
1. Given the minterms [m_1, m_2, m_4, m_7, m_8, m_{11}, m_{13}, m_{14}], fill in the Karnaugh map below. Make the minterm groupings that will result in a circuit with the lowest gate cost. **(4 marks)**

	$\overline{C} \cdot \overline{D}$	$\overline{C} \cdot D$	$C \cdot D$	$C \cdot \overline{D}$
$\overline{A} \cdot \overline{B}$				
$\overline{A} \cdot B$				
$A \cdot B$				
$A \cdot \overline{B}$				

2. Given the Karnaugh map groupings in the previous question, write the most reduced equivalent boolean expression in the space below. **(2 marks)**

3. Consider the flip-flop circuit below and the incomplete state diagram on the right. Given the circuit, complete the transitions of the state diagram, making sure to label each transition. You may assume that the two digits in each state are the values for flip-flops F_1 and F_0 respectively. **(4 marks)**





Part D: Verilog (12 marks)

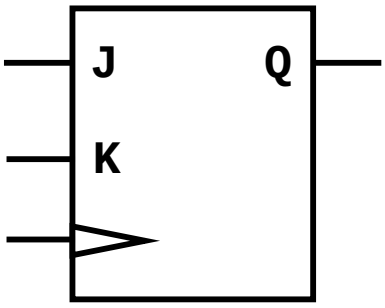
Consider the piece of Verilog code on the right.

```
module foo (c, i, d, o);
  input c, i, d;
  output o;
  reg [7:0] tmp;

  always @ (posedge c)
    begin
      if (d == 1'b0)
        tmp <= {tmp[6:0], i};
      else
        tmp <= {i, tmp[7:1]};
    end

  assign o = tmp;
endmodule
```

1. What does this piece of code do? **(3 marks)**

2. What function does the d signal perform here? **(1 mark)**

3. Consider the JK flip-flop diagram below.



In the Verilog code on the right, fill in the blanks that would complete this Verilog implementation of a JK flip-flop. **(8 marks)**

```
module jk_flipflop(J, K, clk, Q);
input J, K, clk;
output Q;
reg Q;

always @(posedge clk)
  if(J == 1 && K == 0)
    [                    ]
  else if(J == 0 && K == 1)
    [                    ]
  else if(J == 1 && K == 1)
    [                    ]
  else
    [                    ]
endmodule
```

This page is left blank intentionally for answer overflows.

Please enter your first and last name in the space below. Do NOT write your student number here.