Last (family) Name: _____

First (given) Name: _____

netID: _____@wisc.edu

Exam 4

CS/ECE 252 Section-2 (MWF 11:00)

Monday, December 14

Write legibly, especially for your name/netID. <- this is the version number. “name/netID” is version A. “name&netID” is version B. “name+netID” is version C.

Read all questions carefully.

There is a total of 25 points (28 with extra credit). Try to limit yourself to 2 minutes per a point value to keep yourself on pace.

Good luck.

Quote of the day:

“Be yourself, everyone else is already taken.”

-- Oscar Wilde, playwright [1854 - 1900]

Comic:

1. What is “Moore’s Law”? **(2)**

answer: "Moore's law" is the observation that, over the history of computing hardware, the number of transistors in a dense integrated circuit has doubled approximately every two years.
18 months through 36 months are also acceptable.

2. Multiple choice (circle only 1 letter): Which terminal of a transistor determines whether the transistor is opened (off) or closed (on)? **(1)**
   a. source
   b. gate
   c. drain

3. Multiple choice (circle only 1 letter): What is the purpose of the gate oxide in the electrical operation of a transistor? **(1.5)**
   a. is the conducting path for electrons to flow
   b. allows gate to set up an electrical field for electrons to flow
   c. helps in the doping of source and drain
   d. all of the above
   e. none of the above

4. Find the boolean expression for *C* from following truth table. The boolean expression should be in terms of A and B and in sum-of-products (SOP) form. Simplifying is not recommended. **(3)**

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

```
C = A'B + AB' + AB        // this is full credit
C = A + B
```

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

```
C = A'B' + AB' + AB       // this is full credit
C = A + B'
```

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

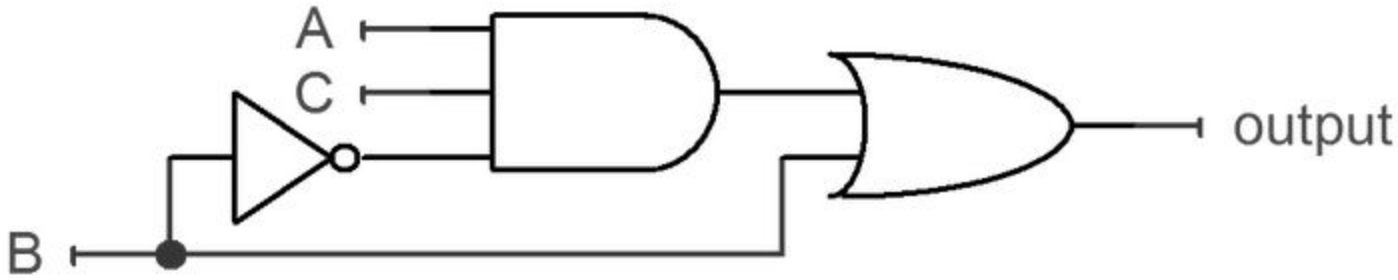
```
C = A'B' + A'B + AB       // this is full credit
C = A' + B
```

answer: _____

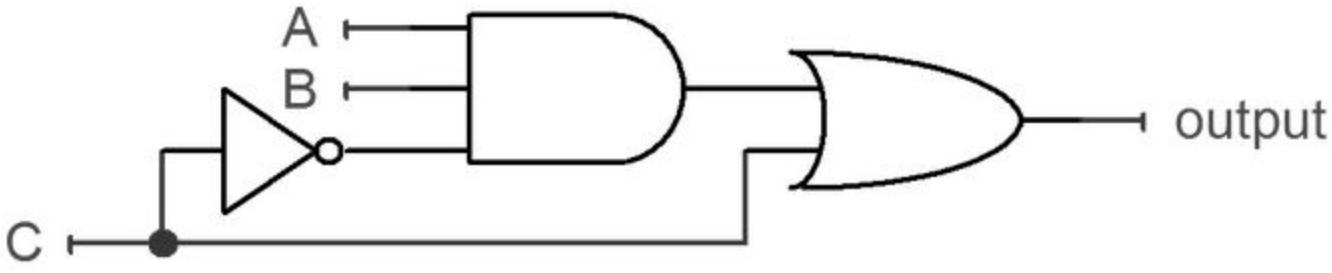
5. Implement the following boolean equation with logic gates. You are allowed to use 2-input, 3-input, 4-input, and n-input gates. You only have inputs A, B, C available.
   output = B+AB'C
   output = C+ABC'
   output = A+A'BC

(3.5)

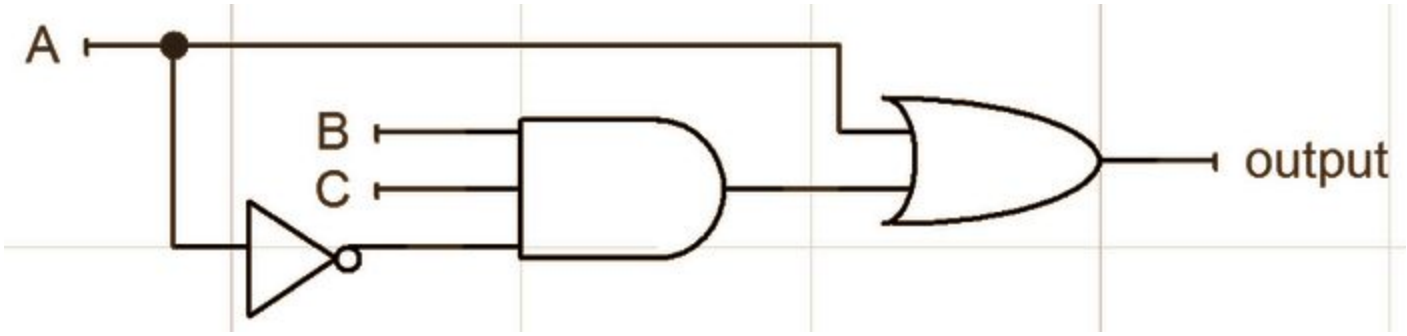
Version A sample solution:



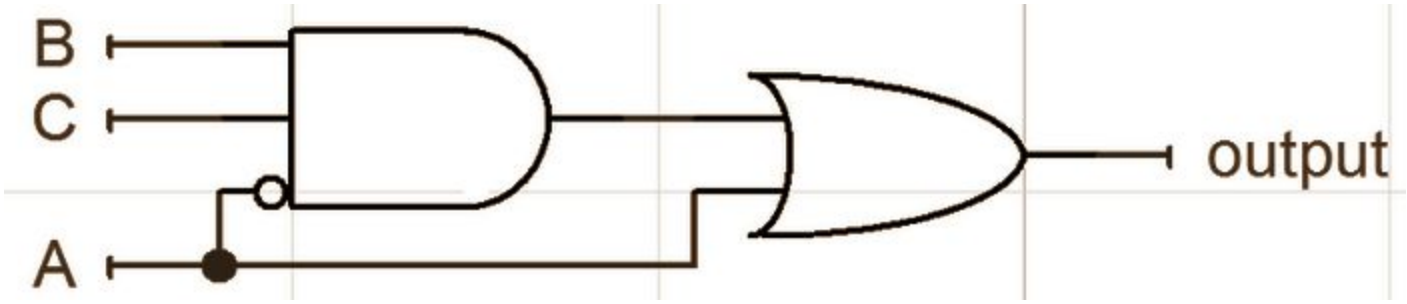
Version B sample solution:



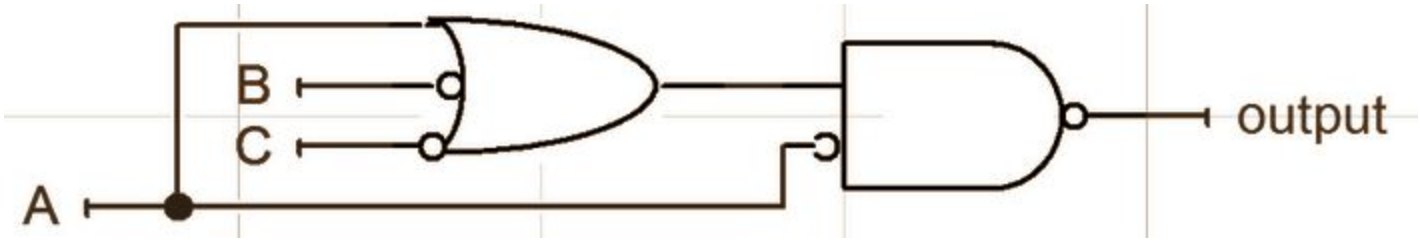
Version C sample solution 1:



Version C sample solution 2 (using non-basic logic gates):



Version C sample solution 3 (using DeMorgan's Law):



6. Compute the negation of the following expression using De Morgan's Law. **(3)**

CD'+BC'D

```
negation: (CD'+BC'D)'
      = (CD')'(BC'D)'      // technically, this is full credit
      = (C'+D)(B'+C+D')
```

C'D+BCD'

```
negation: (C'D+BCD')'
      = (C'D)'(BCD')'
      = (C+D')(B'+C'+D)
```

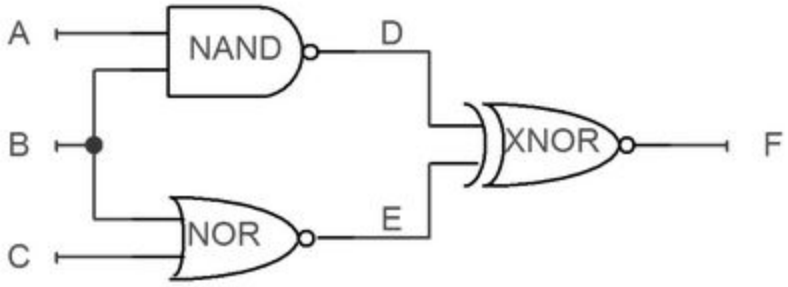
AC'+A'BC

```
negation: (AC'+A'BC)'
      = (AC')'(A'BC)'
      = (A'+C)(A+B'+C')
```

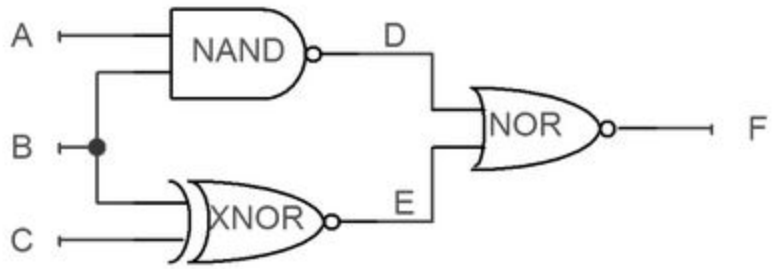
negation: _____

7. Fill out the truth table for the logic gate level circuit below. **(7)**

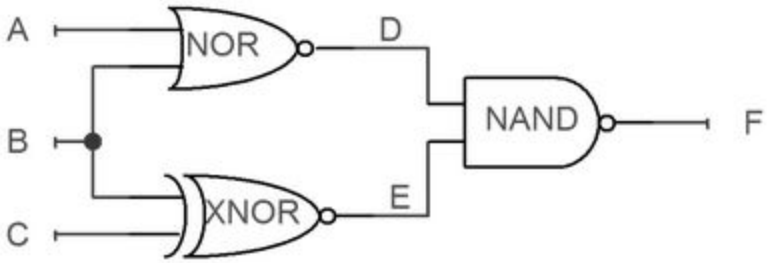
Suggestion for grading: do not deduct all points for F if D or E is incorrect.



A	B	C	D	E	F
0	0	0	1	1	1
0	0	1	**1**	**0**	**0**
0	1	0	**1**	**0**	**0**
0	1	1	**1**	**0**	**0**
1	0	0	**1**	**1**	**1**
1	0	1	**1**	**0**	**0**
1	1	0	**0**	**0**	**1**
1	1	1	**0**	**0**	**1**

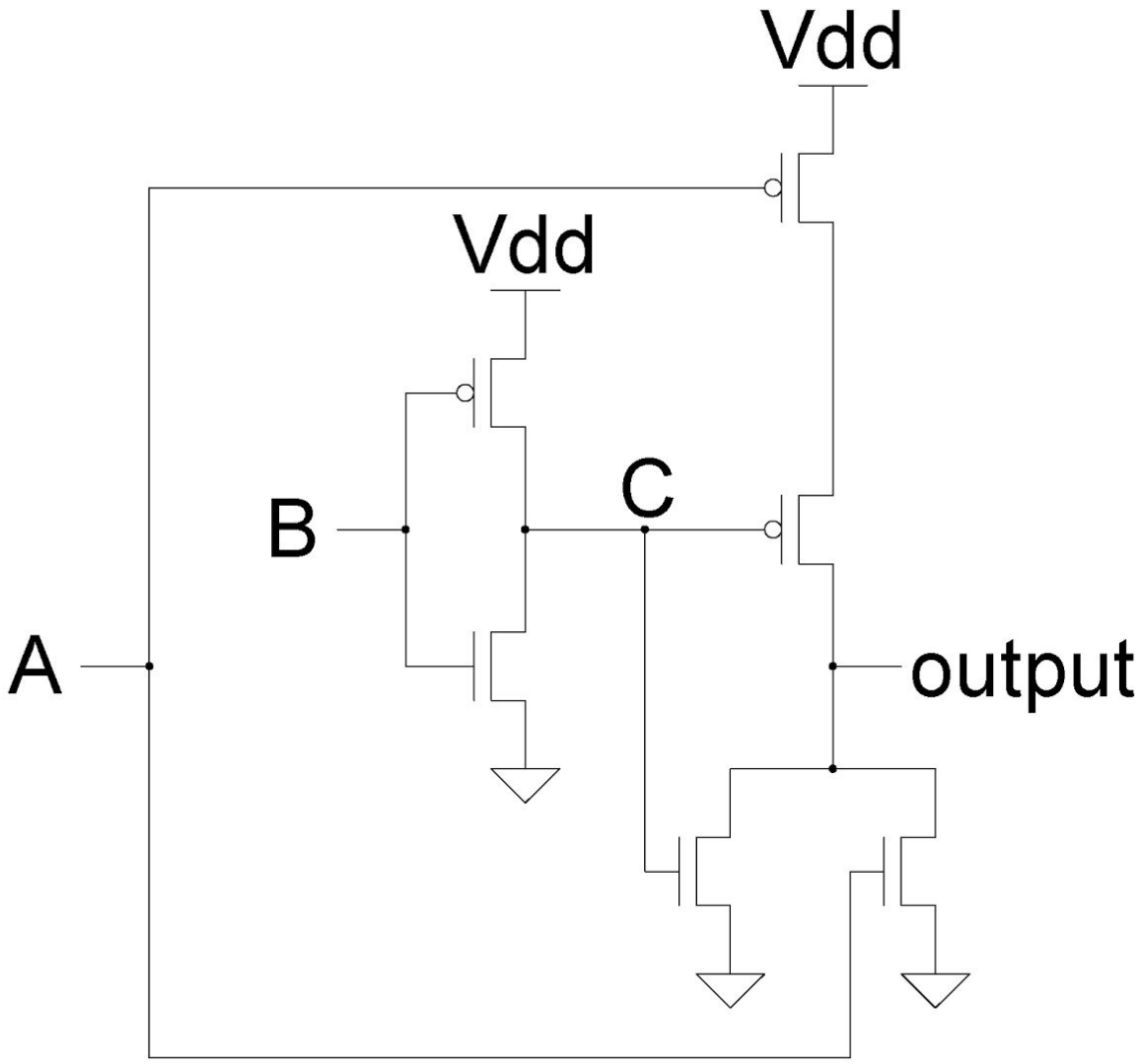


A	B	C	D	E	F
0	0	0	1	1	0
0	0	1	**1**	**0**	**0**
0	1	0	**1**	**0**	**0**
0	1	1	**1**	**1**	**0**
1	0	0	**1**	**1**	**0**
1	0	1	**1**	**0**	**0**
1	1	0	**0**	**0**	**1**
1	1	1	**0**	**1**	**0**



A	B	C	D	E	F
0	0	0	1	1	0
0	0	1	**1**	**0**	**1**
0	1	0	**0**	**0**	**1**
0	1	1	**0**	**1**	**1**
1	0	0	**0**	**1**	**1**
1	0	1	**0**	**0**	**1**
1	1	0	**0**	**0**	**1**
1	1	1	**0**	**1**	**1**

8. Fill out the truth table for the transistor level circuit below. **(4)**



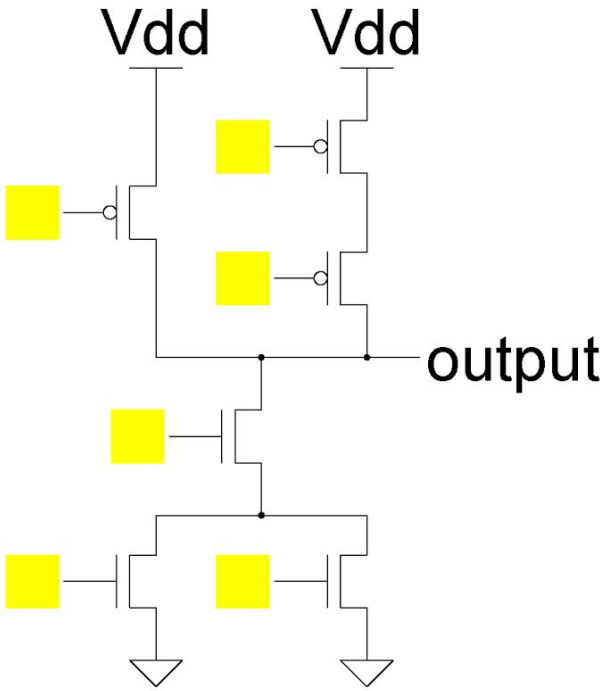
A	B	C	output
0	0	**1**	**0**
0	1	**0**	**1**
1	0	**1**	**0**
1	1	**0**	**0**

Suggestion for grading: do not deduct all points for output if C is incorrect.

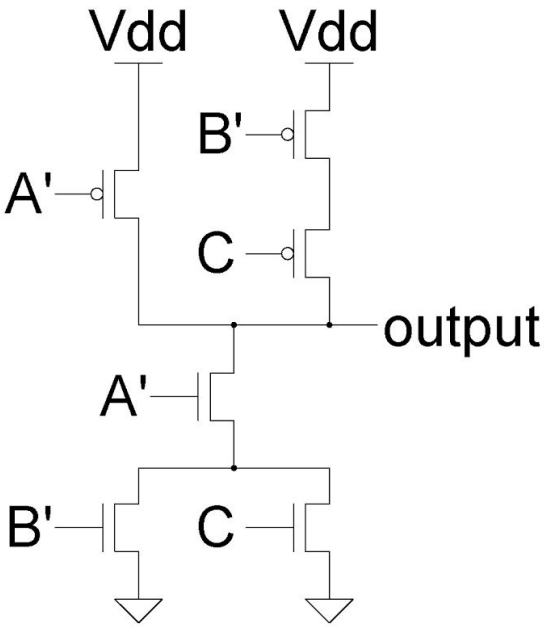
9. **EXTRA CREDIT:** Fill in the input of each transistor below to implement the following boolean equation. The input locations are highlighted in yellow. You have inputs A, B, C, **A', B', and C'** available.

   output = A+BC'
   output = A+B'C
   output = A'B+C

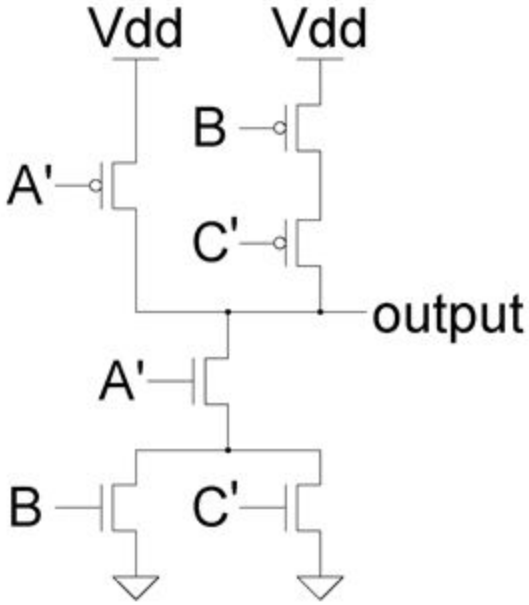
(1.5 Extra Credit)



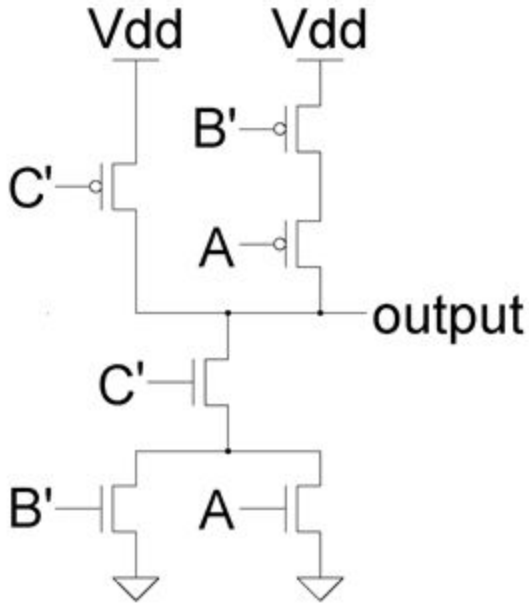
sample solution for version A:



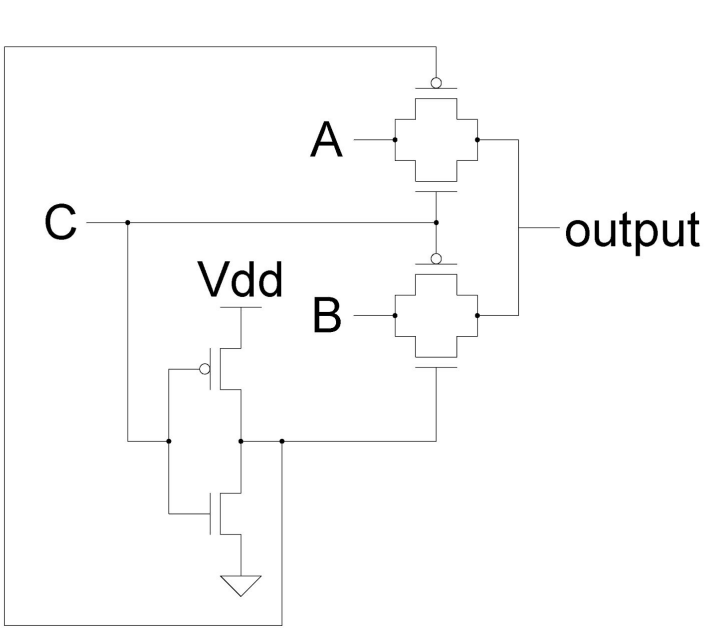
sample solution for version B:

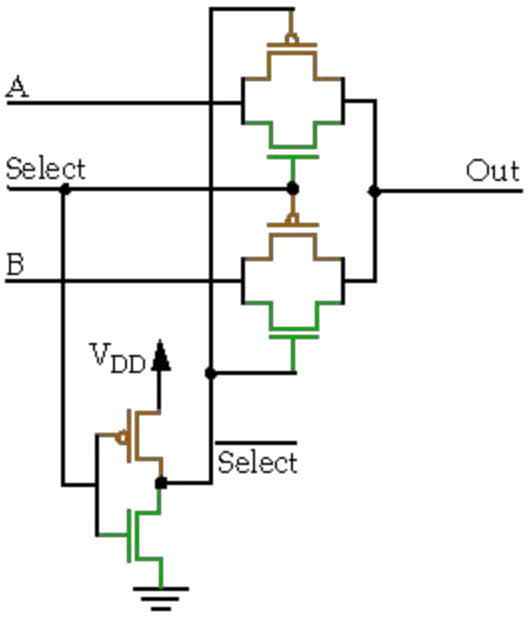


sample solution for version C:



10. **EXTRA CREDIT:** Fill out the truth table for the transistor level circuit below. **(1.5 Extra Credit)**





Calling the input "select" might give away the fact that it is a mux. Therefore, I drafted it to rename the input signal and remove overlaps. Also, Vdd and ground now looks more similar to the book.

A	B	C	output
0	0	0	**0**
0	0	1	**0**
0	1	0	**1**
0	1	1	**0**
1	0	0	**0**
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

Alternative solution that is easier to visualize:

C	output
0	**B**
1	**A**