CSE 140 Midterm

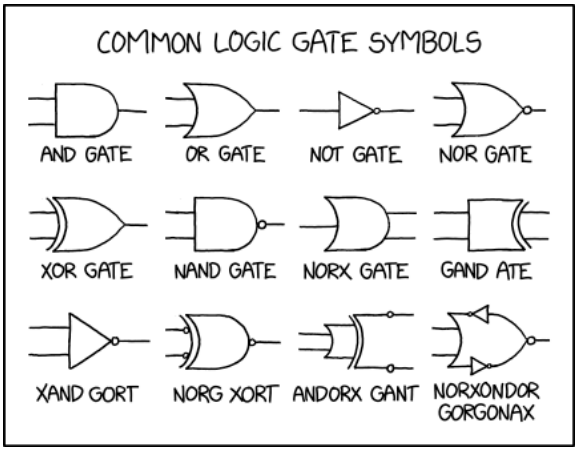
Name: _____ PID: _____

Please record all your answers on the bubble sheet. Exam policies:

- You are allowed to use one cheat sheet, letter size, and double-sided.
- This exam is closed-book. No phones or calculators allowed.
- You can use the back of the question papers as scratch sheets.
- We do not take questions during the exam.
- If you think a question is wrong or unclear, write your reasonings down on the back of the bubble sheet. We will consider that and may give points to everyone.
- There are 26 questions. Each question is worth 4 points. There's only one correct answer to each question. You will be graded out of 100 points. (4 bonus points!)

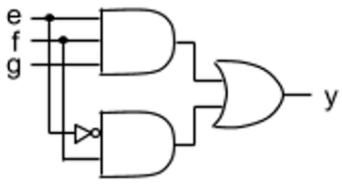
========== Section A: Logic gates and transistors ==========

1. In the comic below, how many logic gates are real? (Source: XKCD) Think about the logic gates we covered in class. (This is a fun question; don't overthink it.)

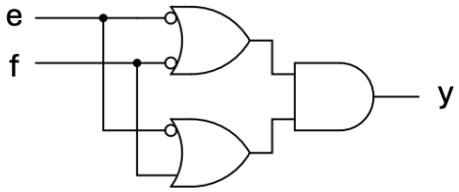


A. 6 B. 8 C. 10 D. 12

For Questions 2 and 3, select the Boolean expression that directly corresponds to each circuit's output



Question 2

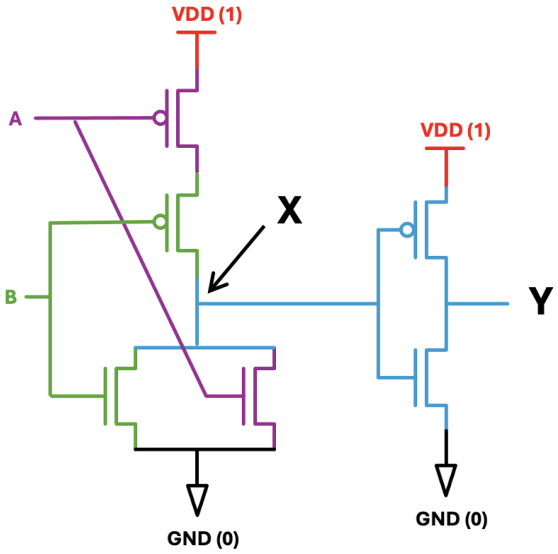


Question 3

2. A. (e+f+g)(e'+f) B. (e'+f'+g')(e+f') **C. efg+e'f** D. (efg+e'f)'

3. A. (e+f)(e+f') **B. (e'+f)(e'+f')** C. ef+ef' D. ef'+e'f

This is a logic gate in CMOS. Specify the logic levels at X and Y under different input conditions in Questions 4 and 5.



4. A=0, B = 1

A. X=0, Y= 0 **B. X=0, Y=1** C. X=1, Y= 0 D. X=1, Y=1

5. A=0, B=0

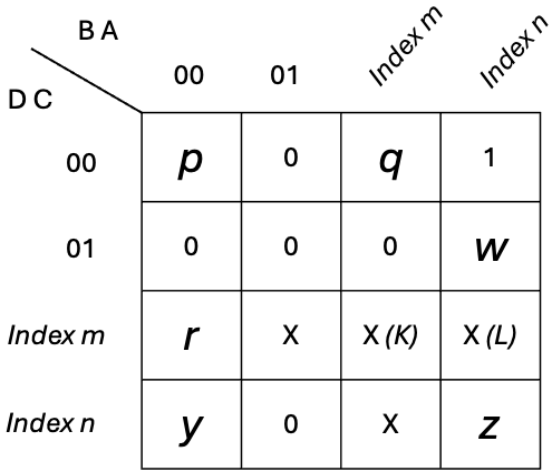
A. X=0, Y= 0 B. X=0, Y=1 **C. X=1, Y= 0** D. X=1, Y=1

========== Section B: K-map and logic minimization ==========

Consider the following truth table from the segment "e" of a *seven-segment decoder.* Any other combination of inputs will result in the output being "don't care".

D	C	B	A	Output "e"
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0

Let's start drawing the K-map from the truth table as below. Answer the following questions. Cells with value "X" mean "don't care".



6. What should the values at the missing indices of this K-map be?

A. Index m = 00, Index n = 01 B. Index m = 10, Index n = 11

C. Index m = 11, Index n = 10 D. Index m = 01, Index n = 00

For questions 7 to 12, choose from : A. 0 B. 1 C. X

7. Find p B
8. Find q A
9. Find w B
10. Find r C
11. Find y B
12. Find z C

Assume all the "X" in the K-map are treated as "0". Perform logic minimization using the K map.

13. Which one is the correct minimum SOP expression assuming all the "X" are 0?

A. *A'BD' + A'B'C'* B. *AB'D' + A'B'C'*

C. *A'BD' + A'B'C'D' + A'B'C'D* D. *AB'D' + A'B'C'D' + A'B'CD'*

Now, assign 0 or 1 to each don't-care cell as needed to obtain the minimum-literal SOP expression.

14. Which value should be assigned to cell (K)?

A. **0** B. 1

15. Which value should be assigned to cell (L)?

A. 0 **B. 1**

16. Which one is the correct minimum-literal SOP expression after properly assigning "0" and "1" to all the "X" cells?

A. A'C B. A'C' + A'BD' **C. A'B + A'C'** D. A'B + A'C' + B

========= Section C: Quant-ReLU and SystemVerilog ==========

A student writes the following, possibly incorrect code to build the Quantized ReLU module for Assignment 2. It takes an input, shifts and rounds it to the nearest even value, clamps it to the maximum and minimum, and applies ReLU. Ignore X and Z values and assume all signals are binary.

```
01: module quant_relu #(
02:   parameter int W_ACC=16, W_Y=8, SHIFT=4,
03:   parameter bit RELU = 1'b1
04: ) (
05:   input  logic signed [W_ACC-1:0] acc,
06:   output logic signed [W_Y  -1:0] y
07: );
08:   logic signed [W_ACC-1:0] quotient, remainder, rounded, clamped;
09:   localparam logic signed [W_ACC-1:0]
10:     MIN  = -(1 <<< (W_Y - 1)),
11:     MAX  =  (1 <<< (W_Y - 1)) - 1,
12:     HALF = (SHIFT > 0) ? (1 <<< (SHIFT - 1)) : '0;
13:
14:   always_comb begin
15:     // ----------- Round to nearest even -----------
16:     quotient = acc >> SHIFT;
17:
18:     if (SHIFT == 0) rounded = acc;
19:     else begin
20:       remainder = acc - (quotient << SHIFT);
21:
22:       if (remainder > HALF)                      rounded = quotient + 1;
23:       else if (remainder == HALF && quotient[0]) rounded = quotient + 1;
24:       else                                       rounded = quotient;
25:     end
26:
27:     // ----------- Clamp to MAX and MIN -----------
28:     if      (rounded < MIN) clamped = MIN;
29:     else if (rounded > MAX) clamped = MAX;
30:     else                    clamped = rounded;
31:
32:     // ----------- ReLU -----------
33:     y = W_Y'(clamped);
34:     if (acc < 0 && RELU) y = '0;
35:   end
36: endmodule
```

17. Find the error (if any) in line 16.

A. Shifting acc by SHIFT results in loss of precision; therefore, it is wrong given the spec.

B. It should be: `quotient = acc >>> SHIFT;`

C. It should be: `quotient = SHIFT << acc;`

D. There is no error

18. What does `quotient[0]` in line 23 do?

A. Selects the quotient at the 0th clock cycle

B. Selects the MSB of the vector named quotient

C. Selects the 0th multi-bit word of the array named quotient

D. Selects the LSB of the vector named quotient

19. A student changes **`quotient[0]`** in line 23 to **`!quotient[0]`**. For a tie (**`remainder == HALF`**), which statement is always true about the value assigned to rounded?

A. It is rounded toward zero
B. It is rounded away from zero
C. It is an odd number
D. It is rounded toward positive infinity

20. Find the error (if any) in lines 33 and 34:
A. acc is signed, therefore <<< should have been used in place of <
B. == should have been used instead of =
C. The same signal cannot be driven from two statements
D. There is no error

For questions 21 - 23, assume all errors in the code are fixed, if any.

21. Assume `W_ACC=8; W_Y=4; SHIFT=2; RELU=0`; and `acc = 30` (`8'b00011110`);
What are `quotient, remainder, rounded, clamped,` and `y`?
A. 7, 2, 7, 7, 7 **B. 7, 2, 8, 7, 7** C. 7, 1, 7, 7, 7 D. 7, 2, 8, 8, -8

22. Assume `W_Y=8; RELU=1`, with all other parameters retaining their default values. What are the smallest and largest outputs of the whole module?
A. 0, 255 **B. 0, 127** C. -128, 127 D. -127, 127

23. For `W_ACC=6; W_Y=5; RELU=0`; What are the values of localparams `MIN` and `MAX`?
A. -32 and 31 B. 0 and 31 **C. -16 and 15** D. -15 and 15

For questions 24 - 26, choose from these answers:

A. `make gds DESIGN=quant_relu`
B. `make sim DESIGN=quant_relu`
C. `make show_layout DESIGN=quant_relu`
D. `gtkwave sim/quant_relu/quant_relu.fst`

24. Which command should we use if we wish to run the simulation of our design with the testbench, to see if the results are correct? B

25. We encountered some errors and would like to check the waveform of some signals for debugging. Which command should we use? D

26. After we fix any errors and are satisfied with our functional design, we need to run the synthesis, placement, and routing flow to generate the final implementation outputs. Which command should we use? A