University of Washington – Computer Science & Engineering

Winter 2026 Instructor: Naomi Alterman 2026-02-03

CSE 369 QUIZ 1

Name: Sam Solutionista

Student ID Number: 369369369

Please do not turn the page until 12:30.

Instructions

- This quiz contains 6 pages, including this cover page. You may use the blank page at the back for scratch work, but clearly label which problems your work corresponds to.
- Please clearly indicate (box, circle) your final answer.
- The quiz is closed book and closed notes.
- Please silence and put away all cell phones and other mobile or noise-making devices.
- Remove all hats, headphones, smart glasses, watches, and other digital wearables.
- You have 20 (+5) minutes to complete this quiz.

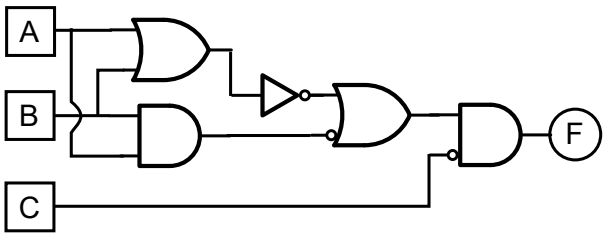
Advice

- Read questions carefully before starting. Read *all* questions first and start where you feel the most confident to maximize the use of your time.
- There may be partial credit for incomplete answers; please show your work.
- Relax. If you've been practicing, you got this. If you haven't, you'll learn something now.

Question	Points	Score
(1) CL Gates	10	
(2) K-map	5	
(3) Waveforms & Verilog	14	
Total:	**29**	

Question 1: Combinational Logic Gates [10 pts]

(A) Write out a Boolean expression for the circuit diagram below. *No need to simplify*. Please the following notation: + (OR), · (AND), ¯ (NOT), ⊕ (XOR), as well as any necessary parentheses to make your answer unambiguous. [4 pts]

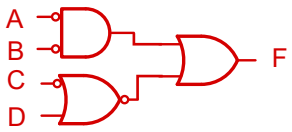


$$(\overline{(A + B)} + \overline{(A \cdot B)}) \cdot \bar{C})$$

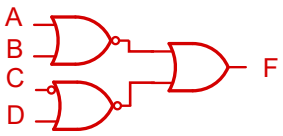
(B) Find a minimal implementation of the function below using only **2-input NOR gates**. *We will only accept circuit diagrams.* [6 pts]

$$F = (\bar{A} \cdot \bar{B}) + \overline{(\bar{C} + D)}$$

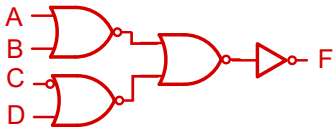
1) Initial gate implementation:



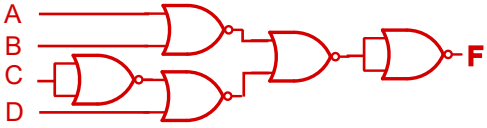
2) Apply DeMorgan's law



3) Double-invert the output to turn OR into ~NOR



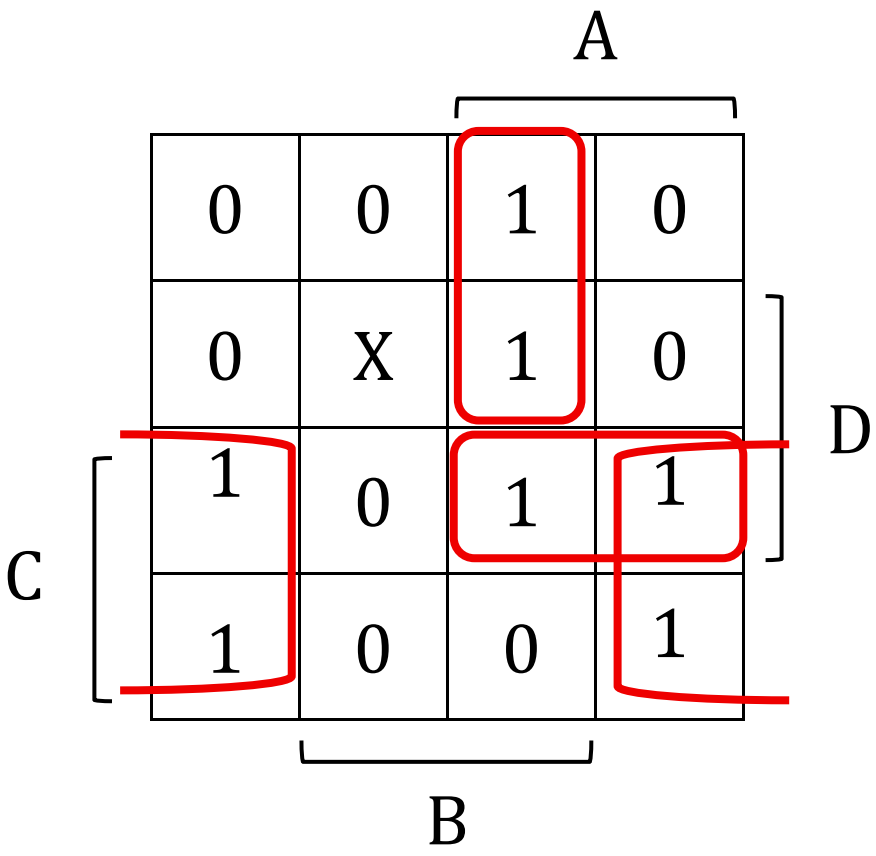
4) Implement NOT gates as X nor X



Question 2: Karnaugh Maps [5 pts]

Find a ***minimum two-level sum-of-products solution*** for the K-map shown below. Show your work by circling groups on the map and writing out the resulting Boolean expression.

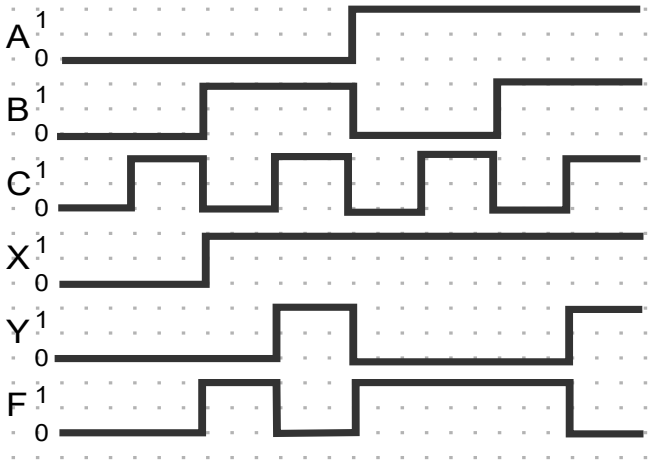
One possible solution:



$\overline{B}C + ACD + AB\overline{C}$

Question 3: Waveforms & Verilog [14 pts]

(A) Consider the following simulation output:



Fill in the blanks below to implement the SystemVerilog of the testbench that generated the above waveform. Assume that one tick ("#1") corresponds to the horizontal distance between a pair of dots on the waveform grid. You may not need all of the blanks. [7 pts]

```
module Mystery_tb ();
   logic F, A, B, C;

   initial begin

      a=0; b=0; c=0; _____; #3;

      c=1; _____; #3;

      b=1; c=0; _____; #3;

      c=1; _____; #3;

      a=1; b=0; c=0; _____; #3;

      c=1; _____; #3;

      b=1; c=0; _____; #3;

      c=1; _____; #3;

   end
   Mystery dut(.F, .A, .B, .C);
endmodule
```

(B) Oh noooooooo, we've got a Sony TV and a Nintendo Switch and they don't seem to be compatible with each other. They both transmit image data as three 8-bit color values grouped into 24-bit "pixels," but they put the colors in opposite orders. The TV expects those colors to be in the order RED, GREEN, BLUE, while the Nintendo transmits them as BLUE, GREEN, RED. Fill in the blanks below to build a hardware adapter: [7 pt]

```
module Swizzle (pi, po);

  input logic ___[23:0]____ pi; // Pixel from Nintendo Switch

  output logic __[23:0]___ po; // Pixel to Sony TV

  logic ___[7:0]___ R;

  logic ___[7:0]___ G;

  logic ___[7:0]___ B;

  assign {B, G, R} = _____pi_____________________;

  assign po = {R, G, B};

endmodule
```

This page reserved for scratch work