University of Washington – Computer Science & Engineering

Spring 2023 Instructor: Justin Hsia 2023-04-25

# CSE 369 QUIZ 1

**Name:** _____

**Student ID Number:** _____

## Please do not turn the page until 2:30.

## Instructions

- This quiz contains 3 pages, including this cover page. You may use the backs of the pages for scratch work.
- 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, and watches.
- You have 20 (+5) minutes to complete this quiz.

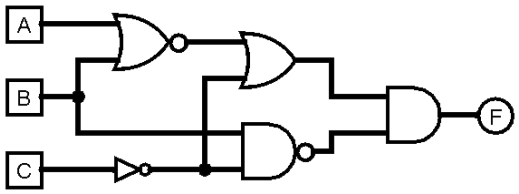
## Advice

- 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. You are here to learn.

| Question | Points | Score |
|---|---|---|
| (1) CL Gates | 8 | |
| (2) K-map | 5 | |
| (3) Waveforms & Verilog | 13 | |
| **Total:** | **26** | |

## **Question 1:** Combinational Logic Gates [8 pts]

(A) Write out a Boolean expression for the circuit diagram below. *No need to simplify*. Remember to use + (OR), · (AND), and ¯ (NOT) as well as any necessary parentheses to make your answer unambiguous. [2 pts]




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

$$F = \overline{\overline{A}\overline{B}} + \overline{C}\overline{D}$$

---

## **Question 2:** Karnaugh Maps [5 pts]

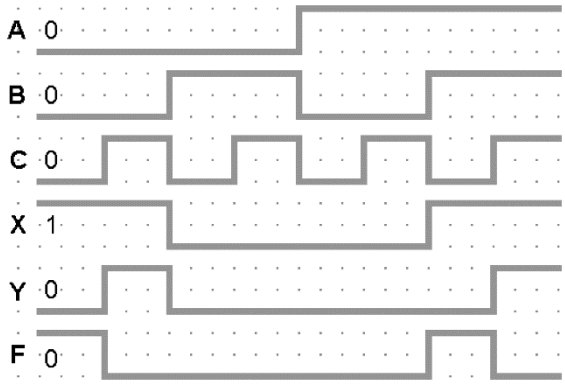
Find the *minimum sum-of-products solution* for the K-map shown below.

| C \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | X | 1 | 1 | 1 |
| 01 | X | 0 | 0 | 1 |
| 11 | 0 | 1 | X | 0 |
| 10 | 0 | 0 | X | 0 |

D

## Question 3: Waveforms & Verilog [13 pts]

(A) Consider the Verilog simulated testbench waveforms shown. If we know that X and Y are outputs of 2-input logic gates, complete the module `Mystery` below. [7 pts]




```
module Mystery (F, A, B, C);
   output logic F;
   input  logic A, B, C;
   logic        X, Y;

   ____________________________________;

   ____________________________________;

   xor    G3 (F, X, Y);
endmodule
```

(B) A testbench for the `Mystery` module (with inputs A, B, C) is shown below. Which input combinations are NOT currently being tested? You may not need all of the blanks. [3 pts]

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

   initial begin
                              #10;
      A = 0; B = 0; C = 0;    #10;
      A = 1;                  #10;
            B = 1;            #10;
                  C = 1;      #10;
      A = 0; B = 0;           #10;
            B = 1; C = 0;     #10;
            B = 0;            #10;
      A = 1;                  #10;
   end
endmodule
```

Missing combinations:

1. {A, B, C} = 3'b_____;
2. {A, B, C} = 3'b_____;
3. {A, B, C} = 3'b_____;
4. {A, B, C} = 3'b_____;

(C) Circle the value of A at the beginning of the simulation of `Mystery_testbench`: [1 pt]

0 1 X Z

(D) Give a brief piece of advice on how to improve the above testbench. [2 pts]