University of Washington – Computer Science & Engineering
Autumn 2024 Instructor: Chris Thachuk 2024-10-29

CSE 369 QUIZ 1

Name: ____ Molly_Model_____

Student ID Number: ____1234567_____

Please do not turn the page until 12:30.

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.

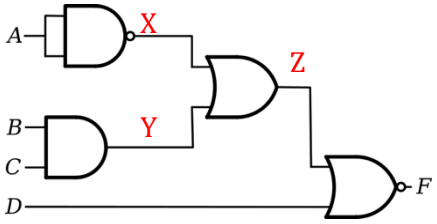
Advice

- 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	8
(2) K-map	5	5
(3) Waveforms & Verilog	13	13
Total:	**26**	**26**

Question 1: Combinational Logic Gates [8 pts]

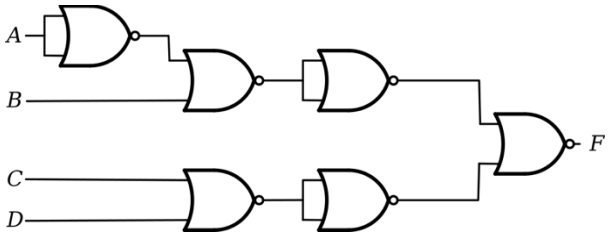
(A) Write out a Boolean expression for the circuit diagram below. *No need to simplify.* Remember to use + (OR), · (AND), and ¯ (NOT) as well as any necessary parentheses to make your answer unambiguous. [2 pts] $\mathbf{F = \overline{((\bar{A} + BC) + D)}}$



$X = \bar{A}$ [0.5 pt]
$Y = BC$ [0.5 pt]
$Z = X + Y$ [0.5 pt]
$F = \overline{D + Z}$ [0.5 pt]

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

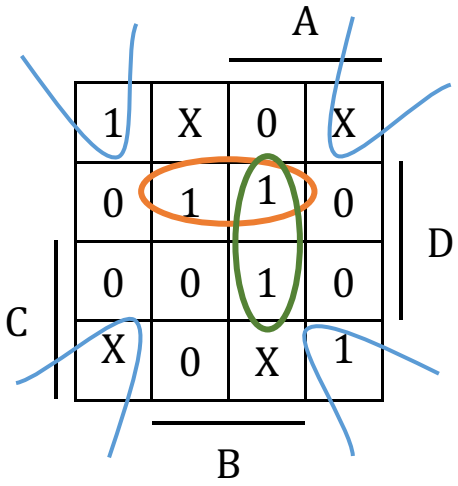
$$F = A\bar{B}\overline{(C + D)}$$



[1 pt] Valid gate conversion from expression
[2 pt] DeMorgan's applications (either in expression or gates)
[3 pt] Conversion of extra NOTs to NORs

Question 2: Karnaugh Maps [5 pts]

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

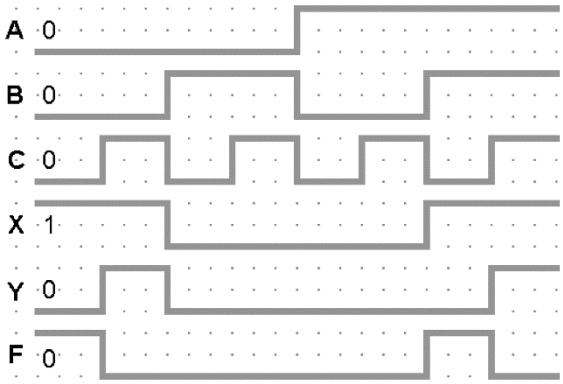


$\mathbf{F = \bar{B}\bar{D} + ABD + B\bar{C}D}$
Or $+ ABD + \bar{A}B\bar{C}$
Or $+ ABC + B\bar{C}D$

[2 pt] X choices
[1 pt each] correct term/grouping
[-0.5 pt each] smaller grouping used
[-0.5 pt each] extra grouping included

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;

   xnor G1 (X, A, B);
   // or assign X = ~(A ^ B);

   and  G2 (Y, C, X);
   // or assign Y = C & X;

   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; // XXX
      A = 0; B = 0; C = 0;    #10; // 000
      A = 1;                  #10; // 100
            B = 1;            #10; // 110
                  C = 1;      #10; // 111
      A = 0; B = 0;           #10; // 001
            B = 1; C = 0;     #10; // 010
            B = 0;            #10; // 000
      A = 1;                  #10; // 100
   end
endmodule
```

Missing combinations:

1. `{A,B,C} = 3'b011;`
2. `{A,B,C} = 3'b101;`
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]

Many acceptable answers here:

- Use a for-loop (from 0 to 7) to guarantee going through all combinations.
- Go through combinations more systematically (*e.g.*, 000 → 001 → 002 → ...).
- Explicitly write out the A, B, and C values on each line for readability.
- Define signals from the start (*i.e.*, eliminate the first `#10;`).