

Fundamentals of Digital Logic Design

ECE 3700

Practise Exam - I

Spring 2018

Note: Time yourself for 1 hour and 20 minutes. Closed book, closed notes, open minds, Do not panic. Good luck!!

- (2 points) Suppose that we are given a circuit that implements an arbitrary Boolean function $f(a, b, c)$, i.e. the circuit takes a, b, c as inputs and produces f as the output. If we invert the inputs, and simultaneously invert the output, do we **always** get back the same function? In other words, is $f(\bar{a}, \bar{b}, \bar{c}) = \overline{f(a, b, c)}$? Demonstrate your answer on: i) $f(a, b, c) = ac + b\bar{c}$; and also on ii) $f(a, b, c) = ab + ac + bc$.
- (3 points) Using a K-map, minimize $f(a, b, c, d) = \sum m(0, 4, 8, 10, 11, 12, 13, 15)$. Implement the resulting two level logic function using only NAND gates.
- (3 points) Consider two 2-bit numbers $D = \{d_1, d_0\}$ and $C = \{c_1, c_0\}$. Design a combinational circuit that takes these two **unsigned** numbers as inputs and produces a 2-bit output $R = \{r_1, r_0\}$. The output R should be the remainder after dividing D by C . For example, when $d_1, d_0 = 11$ and $c_1, c_0 = 10$, then the output $r_1, r_0 = 01$ (as $D = 3$ divided by $C = 2$ results in remainder $R = 1$). Note that since *divide by zero* is not defined, $c_1, c_0 = 00$ will never occur.
 - Write the truth table for the circuit with D, C as inputs and R as outputs. Keep in mind the don't cares.
 - Fill in the Karnaugh map and write simplified Boolean expressions for r_1 and r_0 .
 - Draw a schematic that implements the circuit. Assume that the signals d_1, d_0, c_1, c_0 are available only in TRUE form.
- (3 points) Consider $f(a, b, c) = \sum(0, 2, 3, 6)$. Use (repeated application of) Shannon's expansion to implement this function using only multiplexers (2-to-1 MUXes, with 1-bit data and 1-bit control inputs) and NOT gates. Can you design this circuit using only 3 1-bit MUXes and 1 NOT gate?

5. (4 points) You are asked to design a circuit that:

- takes two 4-bit vectors $X[3 : 0], Y[3 : 0]$ as inputs that are given in **4-bit two's complement form**;
- produces a 4-bit output $Z[3 : 0]$;
- When X, Y have the same sign (either both X, Y are positive numbers, or both are negative numbers), then the circuit produces $Z = X - Y$;
- When X, Y have different signs (i.e. X is positive and Y is negative, or X is negative and Y is positive), then the circuit produces $Z = X + Y$.

Design a circuit that implements such a function. You may assume that you have the following pre-designed circuit blocks already available to you:

- 4-bit adders ($A[3 : 0], B[3 : 0], C_{in}$ inputs, $S[3 : 0], C_{out}$ as outputs)
- Any and all types of multiplexors (MUXes), e.g. feel free to use these MUXes as black-boxes: 4-bit data with 1-bit control ($F[3 : 0] = c ? B[3 : 0] : A[3 : 0]$)
- Any Boolean logic gates AND/OR/NOT/XOR/XNOR/NAND/NOR/whatever