University of Windsor
Faculty of Science
School of Computer Science

COMP-2650
Computer Architecture I: Digital Design
Winter 2022

Assignment#	Date	Title	Due Date	Grade Release Date
Lec04	Week 04	**Logic Gates**	March 02, 2022, Wednesday 4 AM EDT	Feb. 21, 2022

The objectives of the lecture (weekly) assignments are to practice on topics covered in the lectures as well as improve the student's critical thinking and problem-solving skills in ad hoc topics that are closely related but not covered in the lectures. Lecture assignments also help students with research skills, including the ability to access, retrieve, and evaluate information (information literacy.)

Deliverables

You should answer 2 of the below questions based on your preference using an editor like MS Word, Notepad, and the likes or pen in papers. In the latter case, you have to write and scan the papers clearly and merge them into a single file. In the end, you have to submit all your answers in one single pdf file `lec04_UWinID.pdf` containing the question ids for the answer. Please note that if your answers cannot be read, you will lose marks. Please follow the naming convention as you lose marks otherwise. Instead of UWinID, use your own UWindsor account name, e.g., mine is hfani@uwindsor.ca, so my submission would be: `lec04_hfani.pdf`

Questions

(select only 2 questions based on your preference)

1. Using the truth table, show that the NAND is *not* associative. That is Z ↑ (Y ↑ X) ≠ (Z ↑ Y) ↑ X.

2. We can perform logical operations on strings of bits by considering each pair of corresponding bits separately (called **bitwise operation.**) Given two eight-bit strings A = 10110001 and B = 10101100, what are the eight-bit result after a) XOR and b)NAND.

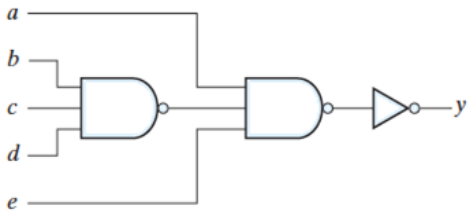
3. Draw logic circuits (diagram) for the following Boolean expressions considering the **precedence** order between logic operations, that is, when there is no parenthesis, NOT > AND > OR.
   a) [example] F=u+x'y = (u + ((x')y))
   b) F = u + x'y' + z
   c) F = u ⊕ y' + x ; *The precedence of xor is not given. Search and find out yourself!*

4. Find the **complement** (i.e., NOT) of the following expressions:
   a) (a + c) (a + b') (a' + b + c')
   b) z + z'(v'w + xy)

5. Design the logic circuits for the Boolean function F = xy + x'y' + y'z
   a) With OR and inverter gates
   b) With NAND only
   c) With NOR only

6. Given n binary variables in the input, how many different Boolean functions are available? For example, given 1 binary variable x, there would be 4 different Boolean functions: F1=0, F2=1, F3=x', F4=x. Look at Table 2.7 in the book for 2 binary variables.

7. Write Boolean expressions and construct the truth tables describing the outputs of the circuits described by the logic diagrams:



8. Design the logic circuit that output 1 (equivalently unlock the door/turn the light on/unlock the safebox, etc) when there is an *even* number of zeros in the inputs. Assume there are 3 binary variables in the inputs A, B, and C.