



**School of Computer Science
Faculty of Science**

**COMP-2650: Computer Architecture I: Digital Design
Winter 2021**

#	Date	Title	Due Date	Grade Release Date
Lec04	Week 04	Boolean (Logic) Gates	Feb. 02, 2021 Tuesday Midnight AoE Wednesday 7 AM EDT	Feb. 08, 2021

The objectives of the lecture (weekly) assignments are to practice on topics covered in the lectures as well as improving 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).

Lecture Assignments 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. You have to write and scan the papers clearly and merge them into a single file in the latter case. In the end, you have to submit all your answers in one single pdf file COMP2650_Lec04_UWinID.pdf containing the following items:

1. Your name, UWinID, and student number
2. The question Id for each answer. Preferably, the questions should be answered in order of increasing Ids. *Please note that if your answers cannot be read, you will lose marks.*
3. Including the questions in your submission pdf file is optional.

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: COMP2650_Lec04_hfani.pdf

Lecture Assignments

(select only 2 questions based on your preference)

1. Using the truth table, show that the NAND logic gate is *not* associative. That is $Z \uparrow (Y \uparrow X) \neq (Z \uparrow Y) \uparrow X$.
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 $B = 10101100$, evaluate the eight bit result after the following logical operations:
 - a) XOR
 - b) NAND
3. Draw logic circuits (diagram) to implement the following Boolean expressions considering the **precedence** order between logic operations, that is, when there is no parenthesis, NOT > AND > OR. For example, $F = u + x'y = (u + ((x')y))$.
 - a) $F = u + x'y' + z$
 - b) $F = u \oplus y' + x$; *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 $F = xy + x'y' + y'z$

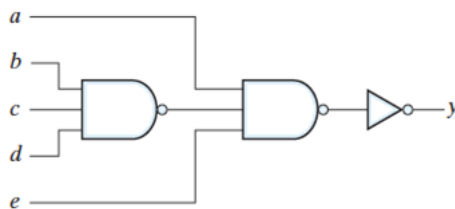
a) With OR and inverter gates

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: $F_1=0$, $F_2=1$, $F_3=x'$, $F_4=x$. 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.