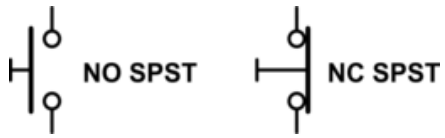
CS 250 Spring 2017 Homework 02 SOLUTION
Due 11:58pm Wednesday, January 25, 2017
Submit your typewritten file in PDF format to Blackboard.

1. Write out the POS expression for 2-intput XNOR.
   Answer: The truth table for 2-input XNOR is

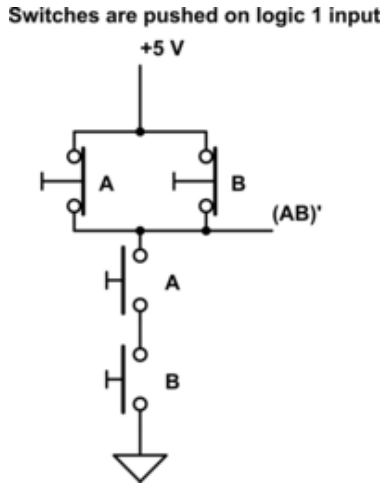
A	B	(A⊕ B)’
0	0	1
0	1	0
1	0	0
1	1	1

   The POS factors come from the rows of the table that are 0 for XNOR. These rows are the middle two. The factors are the ORed variable valueas for these rows that will generate a 0 output. Thus, for the row where A=0 and B=1, the POS factor is (A + B’) because (A + B’) = 0. The row with A=1 and B=0 yields the factor (A’ + B). Therefore the complete POS expression for A XOR B is **(A + B’)(A’ + B)**.

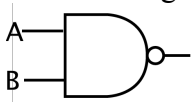
2. The normally-open (NO) SPST push button switch (same type as in the lab kit) transitions from high resistance to low resistance when pushed. The normally-closed (NC) SPST push button switch reverses this behavior. The schematic symbols for these two switches are shown here.



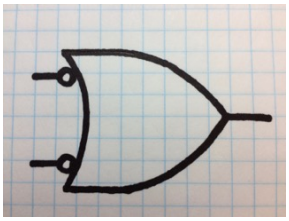
   Using these two switch types, wire, and connections to +5 V and ground, draw a schematic to implement (AB)’ and clearly label your inputs and output.
   Answer: The circuit schematic is



3. Examine the gate shown below, then draw an equivalent single gate that does not contain a NAND gate nor an AND gate.



Answer: Using DeMorgan's Law we should draw an OR gate with active low inputs.



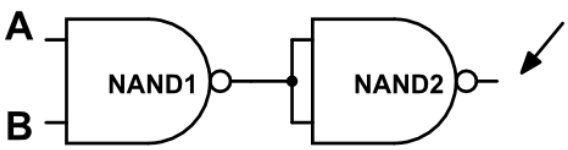
4. The most important logic gate parameter for the prevention of error is the
   a. width of the voltage band representing logic 1
   b. width of the voltage band representing logic 0
   c. power supply voltage
   d. gap between the highest logic 1 voltage and the lowest logic 0 voltage
   e. **gap between the lowest logic 1 voltage and the highest logic 0 voltage**

5. A revised version of the Lab01 take home problem has been assigned. You are to rebuild the XNOR gate using only 2-input NAND gates, with one difference. Because you are practicing to qualify to compete in the Olympic Games, you must operate the circuit using the tip of an epée while in a fencing pose as shown at the web link and standing far enough away from your breadboard that by stretching you can only just reach the circuit with the tip of the epée.
   See https://en.wikipedia.org/wiki/Épée - /media/File:Fencing_epee_valid_surfaces.svg to become familiar with an epée.
   Can you re-build the XNOR circuit using fewer than 5 two-input NAND gates as was done originally in lab? If yes, describe why and how.
   Answer: You can build the epée-operated XNOR using only 2 two-input NAND gates. Why? With only the epée to push the input switch buttons, instead of all your fingers, you will be able to push at most one button at a time. The Lab 01 circuit inputs are active low. Therefore, using the epée to push the buttons only the input combinations A=1,B=1 (epée tip not pushing either button) and A=0,B=1 (epée pushing button A) and A=1,B=0 are possible. The input combination A=0, B=0 is not possible. So the truth table for XNOR as possible using the epée is

A	B	(A⊕ B)' as performed using the epée, where A=0, B=0 is an impossible input
0	0	X
0	1	0
1	0	0
1	1	1

   How? If we choose the don't care entry in this truth table to be 0, rather than 1, then the table corresponds to the two-input AND function. We can build an AND circuit using 2 two-input NAND gates. One NAND gate computes (AB)' and a following NAND gate

is configured to perform as an inverter, yielding ((AB)')' = AB.



Output of "EpéeXNOR" gate for which the input A=B=0 is not possible. Thus, we can design assuming that 0 EpéeXNOR 0 = Don't Care. Choosing output equals zero changes the truth table in a way that this much simpler circuit satisfies.

Touché!