# Pre-Lab Submission 3

Due: Thu, 02 Feb 2023 14:30:00 (approximately 6 days from the time this page was loaded)
[40 points possible] [100 penalties possible] [0 penalties graded so far]

Complete this **before** doing your lab experiment.

Prototyping is an important skill to have in courses that follow ECE 270. The only way to gain that skill is to practice with increasingly complicated circuits. In this lab experiment, you will implement two logic functions:

```
F1(W,X,Y,Z) = W·Z' + W·X' + X'·Y + Y·Z'
F2(W,X,Y,Z) = (W'·Y')' · (X·Z)'
```

It just happens to be that these two functions have exactly the same logical result. They differ in the number of logic gates they require to *realize* the function and in the propagation delays that they incur in producing their results. This is an appropriate situation to consider as we start to look at the toping of *Mapping and Minimization* in lecture module 2.

Recall, from the previous experiment as well as the lecture notes, that an AND-OR tree of gates that F1 has, can be turned into a NAND-NAND tree by using DeMorgan's Law. DeMorgan's Law can also be used to change F2 from the NAND-AND tree into a NOR-NOR tree.

In this prelab, you will generate a truth table for either function, apply DeMorgan's Law to obtain the NAND-NAND and NOR-NOR realizations of the functions, and build a schematic version of the circuit in the simulator.

# Academic Honesty Statement [0 ... -100 points]

By typing my name, below, I hereby certify that the work on this prelab is my own and that I have not copied the work of any other student (past or present) while completing it. I understand that if I fail to honor this agreement, I will receive a score of ZERO for the lab, a one letter drop in my final course grade, and be subject to possible disciplinary action.

I also understand that I am to use only my own hardware for this and every other lab and not share it with any other student. I have used only my own hardware for this lab (except when explicitly permitted otherwise by the course staff). If a discovery is made otherwise, I will receive a zero for this lab.

Miller Kodish

Save

# 1: Function Representation [10 points]

Fill out the truth table, below, for the function you are to implement in this lab experiment as described in the lab document.
Note that there is no partial credit for this question. You must take care to get it correct. Check your work.

| W | X | Y | Z | F(W,X,Y,Z) |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |

| | | | | |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 |
| Score | | | | |

Save

---

# Step 2: Rewrite F1 [5 points]

Use DeMorgan's law to rewrite the Boolean function **F1(W,X,Y,Z) = W·Z' + W·X' + X'·Y + Y·Z'** in a form that can be implemented with only 2- and 4-input NAND gates and inverters. Make sure that all inputs of your new expression evaluate to the same result for F1. (Use * + ' to represent AND, OR, and NOT, respectively.) **Do NOT simplify the expression - your final expression must have a 4-input NAND gate, or you will not get credit for this question.**

Note that there is no partial credit for this question. You must take care to get it correct. Check your work.

F1(W,X,Y,Z) = ((W*Z')'*(W*X')'*(X'*Y)'*(Y*Z')')'

Save

---

# Step 3: Rewrite F2 [5 points]

Use DeMorgan's law to rewrite the Boolean function **F2(W,X,Y,Z) = (W'·Y')' · (X·Z)'** in a form that can be implemented with only 2-input NOR gates and inverters. Make sure that all inputs of your new expression evaluate to the same result for F2. (Use * + ' to represent AND, OR, and NOT, respectively.)

Note that there is no partial credit for this question. You must take care to get it correct. Check your work.

F2(W,X,Y,Z) = ((W+Y)'+(X'+Z')')'

Save

---

# Step 4: Schematic entry [20 points]

Use KiCad EESchema to construct the circuit you specified above with only

- 74HC00-style 2-input NAND gates
- 74HC02-style 2-input NOR gates
- 74HC04-style inverters
- 74HC08-style 2-input AND gates
- 74HC10-style 3-input NAND gates

See the instructions in the lab for details.

Choose File No file chosen Upload

```
(kicad_sch (version 20211123) (generator eeschema)

  (uuid 177ccae7-09da-4be1-93fc-0d7e2fb3f386)

  (paper "A4")

  (lib_symbols
    (symbol "74xx:74HC00" (pin_names (offset 1.016)) (in_bom yes) (on_board yes)
      (property "Reference" "U" (id 0) (at 0 1.27 0)
        (effects (font (size 1.27 1.27)))
```

Save

---

# Netlist file submission

To easily autograde your schematic, we will need a netlist file generated by Eeschema.

To generate this file, finish your schematic, and then click File > Export > Netlist..., and click Export Netlist. Save the file with any name you want as long as it ends in ".net", and upload it here.

Choose File No file chosen Upload

```
    (comp (ref "U4")
      (value "74HC00")
      (datasheet "http://www.ti.com/lit/gpn/sn74hc00")
      (libsource (lib "74xx") (part "74HC00") (description "quad 2-input NAND
gate"))
      (property (name "Sheetname") (value ""))
      (property (name "Sheetfile") (value "Prelab Schematic.kicad_sch"))
      (sheetpath (names "/") (tstamps "/"))
      (tstamps "16e1a8a2-76c2-40ea-b174-a6d9b85a2994"))
    (comp (ref "U5")
```

Save

Remember to complete the wiring for your circuit before your lab begins.