CSCI 2500 — Computer Organization
Homework 04 (document version 1.0) — Due October 24, 2022
Onez and Seroz

- This homework is due by the Midnight EDT on the above date via a Submitty gradeable.
- This homework is to be completed **individually**. Do not share your code with anyone else.
- Homework assignments are available approximately seven calendar days before they are due. Plan to start each homework early. You can ask questions during office hours, in the Submitty forum, and during your lab session.

This will be an extra-special assignment. It involves writing no code. Whether or not you're preferential to this blend of extra-specialness is TBD. Regardless, answer the questions below. Show all of your work!

Overview

- Answer the questions and provide solutions to all problems given below.
- You MUST type up your answers. Handwritten solutions will not be accepted or graded, even if they are scanned into a PDF file.
- We recommend using LaTeX (`https://www.latex-project.org/`). If you have never used LaTeX, you might want follow some tutorial, like this one: `https://www.latex-tutorial.com/tutorials/`. There is a convenient online LaTeX editor called Overleaf (`https://www.overleaf.com/`) that has a free plan for students.
- Submit your answers on Submitty as a single PDF file named `hw4.pdf`.
- All circuit diagrams (schematic diagrams) must be submitted as both pictures in your PDF and OpenCircuits files (`https://nightly.opencircuits.io/`) uploaded to Submitty.
- This homework will be manually graded by our TAs.
- Explicitly denote any assumptions you need to make.
- **Show all work!** Because the majority of the problems below have online calculators which would make them otherwise trivial to answer correctly, **no work=no credit** for this assignment.
- Clearly label every problem. If a problem has a single answer, like `0x561a9d36`, make it bold, use a larger font, and enclose in a box for better visibility, like this: **0x561a9d36**. You still need to show all work!

Boolean Algebra

1. Simplify the following expressions using Boolean algebraic laws. Give each step of your simplification and denote which laws you're using for each step. Do not skip or combine steps!

   (a) $A * (\overline{A} + B * B) + \overline{(B + A)} * (\overline{A} + B)$

   (b) $\overline{C * B} + (A * B * C) + \overline{A + C + \overline{B}}$

   (c) $(A + B) * (\overline{A} + C) * (\overline{C} + B)$

2. Find all solutions of the following Boolean equations without using the truth tables:

   (a) $(\overline{A} + C) * (\overline{B} + D + A) * (D + A * \overline{C}) * (\overline{D} + A) = 1$

   (b) $(((\overline{K} * L * N) * (L + M)) + ((\overline{K} + L + N) * (K * \overline{L} * \overline{M}))) * (\overline{K} + \overline{N}) = 1$

3. Simplify the following expression by first constructing a truth table, using that truth table to construct a K-map, and then using that K-map to simplify.
   $Q = \overline{X} * \overline{Y} * Z + X * Y * \overline{Z} + \overline{X} * Y * \overline{Z} + X * \overline{Y} * \overline{Z}$

Logical Circuits

4. Convert the following truth table into its sum of products representation:

A	B	C	Output
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

5. Draw a logical circuit diagram that represents the above sum of products expression using OpenCircuits (https://opencircuits.io/). Clearly label all inputs/outputs and all components. Make sure you connect appropriate input components (e.g., buttons, switches, clocks, etc.) and output components (e.g., LEDs, displays, etc.) to facilitate testing of your circuit. Download your diagram using OpenCircuits' "Download" feature, rename it to `hw4_SOP.circuit`, and submit on Submitty along with your `hw4.pdf` file.

6. Test you circuit by supplying appropriate inputs and observing the expected values of the output. Explain why your set of tests is sufficient to prove that your logical circuit does in fact implement the required Boolean function. For each test, provide a picture (snapshot) of your circuit. Insert all such pictures in the `hw4.pdf` PDF file. You can download pictures (PNG, JPEG, or PDF) of your circuit diagram using OpenCircuits' "Export Image" feature.

7. Given inputs A and B, show that NOR $\{\overline{(A + B)}\}$ is functionally complete by giving logical circuits equivalent to AND $\{(A * B)\}$, OR $\{(A + B)\}$, and NOT $\{\overline{A}\}$ gates using *only* NOR gates in their construction.

Numerical Conversions and Arithmetic

All problems in this subsection are from the following section https://learn.zybooks.com/zybook/RPICSCI2500KuzminFall2022/chapter/9/section/13 in zyBooks. Saturating arithmetic means that when a calculation overflows, the result is set to the largest positive number or most negative number, rather than a modulo calculation as in two's complement arithmetic.

8. **Textbook Problem 9.13.1**
9. **Textbook Problem 9.13.2**
10. **Textbook Problem 9.13.6**
11. **Textbook Problem 9.13.10**
12. **Textbook Problem 9.13.11**
13. **Textbook Problem 9.13.20**
14. **Textbook Problem 9.13.21**
15. Give a reason why we use two's complement representation for negative numbers in computer arithmetic. Give an example of its usage.

Submission and Grading Criteria

For this assignment, you will submit your answers to the problems and your OpenCircuits diagrams into the Submitty gradeable.