CSCI 2500 —- Computer Organization
Homework 04 —- Due October 25, 2024

- This homework is due by 11:59pm EDT on the above date via a Submitty gradeable named *"Homework 04"*.
- This homework is to be completed individually. Do not share your solutions with anyone else.

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.
- You are recommended to use LaTeX (https://www.latex-project.org/). If you have never used LaTeX, you might want to follow some tutorials, 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. You can also use Microsoft Word or free Google Doc (in your Google drive) both of which have equation editors.
- Submit your answers for this homework on Submitty as a single PDF file named `hw04.pdf`.
- All circuit diagrams (schematic diagrams) must be submitted as both pictures in your PDF and OpenCircuits files (https://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, and use a larger font for better visibility, like this: **0x561a9d36** . You still need to show all the work!
- There are 10 problems in this homework.

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! *(Character * means AND operator)*

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

   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: *(Character * means AND operator)*

   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. *(Character * means AND operator)*

   $Q = \overline{X} * \overline{Y} * Z + X * Y * \overline{Z} + \overline{X} * Y * \overline{Z} + X * \overline{Y} * \overline{Z}$

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 `hw04_SOP.circuit`, and submit on Submitty along with your `hw04.pdf` file.

6. Test your 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 `hw04.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 *(Please note that you only need to solve Problems 8 and 9 if we cover Floating Point numbers during the lectures by Thursday, October 24 . Otherwise, just answer Problem 10.)*

8. For each of the following numbers, convert them to their closest single precision IEEE 754 floating point representation. First, denote the binary values of the sign, fraction, and exponent. Then provide a 32-bit hexadecimal value. Show your steps.
    a. 50.4375
    b. 0.0
    c. $-Infinity$
    d. 1.0000001

9. For each of the following hexadecimal values, convert them from single precision IEEE 754 floating point representation to decimal rational numbers. You may leave large powers of two in the exponential form, and you may express your answer as a ratio (e.g., $-\frac{5}{8}$, $\frac{1}{2^{64}}$, etc.). Show your steps.

    a. `0xc349a000`
    b. `0xffe00001`
    c. `0x80000000`
    d. `0x00400000`

10. Give a reason why we use 2's complement representations 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.