# COMP-273, Winter 2020, Assignment 1

School of Computer Science
McGill University
Available On: January 20th, 2020
Due Date: January 31st, 2020, by 11:59pm.
Submit your solution in electronic form using mycourses.
Read the assignment submission instructions carefully!
Late policy: 10% off per day late, for up to 2 days.
No submissions accepted after then.

## Question 1: Number Representation (30 marks)

1. In the table below, in each row, you are given an *unsigned* number in a particular base. Complete the table by providing the representation of that same number in the other 3 bases. For each conversion you must show your working to get credit. You cannot just provide the final answer. (**24 marks**)

| Binary | Octal | Decimal | Hexadecimal |
|---|---|---|---|
| **10011001** | | | |
| | **333.75** | | |
| | | **4366.8** | |
| | | | **DBE.C** |

2. Perform the following calculation using an 8-bit 2's complement representation. The numbers are given in base 10. (**6 marks**)

$$(78)_{10} - (50)_{10}$$

# Question 2: IEEE Floating Point Numbers (15 marks)

1. Represent **101.00011** as an IEEE single precision floating point number.
2. Represent **-2.45609** as an IEEE single precision floating point number

You must show all your work to get full credit.

# Question 3: Boolean Algebra (15 marks)

Derive the minimized sum-of-products forms for the following functions. You may use a combination of truth tables and applications of the laws of Boolean algebra, but be sure to show your work. You will get no credit for just writing down the final answers.

1. $f(A, B, C, D) = (A + \overline{B} + \overline{C}).(\overline{A} + \overline{D}).(A + C)$
2. $f(W, X, Y, Z) = (W.Y + W.\overline{Z}).\overline{(\overline{X} + \overline{Y}).(\overline{X} + Z)}$

# Question 4: Gray Code Incrementer (40 marks)

A *Gray code* is an alternate binary integer representation with the special property that the encodings of successive numbers differ only by a single binary digit (bit). They have applications in engineering, mathematics and computer science. An encoding of the first 16 unsigned integers using a Gray code is given here:

| Decimal | Gray |
|---|---|
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0011 |
| 3 | 0010 |
| 4 | 0110 |
| 5 | 0111 |
| 6 | 0101 |
| 7 | 0100 |
| 8 | 1100 |
| 9 | 1101 |
| 10 | 1111 |
| 11 | 1110 |
| 12 | 1010 |
| 13 | 1011 |
| 14 | 1001 |
| 15 | 1000 |

Note that we can wrap around from 15 back to 0 by also changing only a single bit. In this question you will design a Gray code incrementer, which is a circuit which accepts a 4-bit Gray code $A_3A_2A_1A_0$ as input and outputs the Gray code $B_3B_2B_1B_0$ For the next number (rolling over to the representation of $(0)_{10}$ if the input represents $(15)_{10}$ .

1. Construct the truth table for this circuit showing the outputs (the $B_i$'s) as functions of the inputs. (**10 marks**)

2. Write down a valid sum-of-products Boolean expression for each of the outputs. Then, using the laws of Boolean algebra, simplify the expressions. (**15 marks**)

3. Draw the circuit diagram for your 4-bit Gray code incrementer in **logisim-evolution**. Try to use as few gates as possible. Be sure to test your circuit by verifying its behaviour in logism-evolution. (**15 marks**)

# Assignment Submission Instructions

By handing in this your assignment you acknowledge that the work you are submitting is your own, and that you have read the COMP 273 FAQs document under "Content" in *mycourses*.

1. The circuit you submit must be saved as a logism-evolution file, so that we can test it, while the document showing your working for everything else should be a PDF file. It is alright to generate a scan of handwritten solutions, provided that your handwriting is legible. But if your handwriting is not clear, it is better to type your answers using an editor.

2. The Logisim circuit must run under logism-evolution, to be graded.

3. Zip your PDF and logism files into a single file, and rename it with your student ID number. For example, 260763964_A1.zip

4. Submit this single compressed file on *myCourses* under Assignment 1.

5. Once you have submitted your assignment, download the zip file you uploaded and check that it is indeed what you intended us to grade. This step is critical because a non-trivial number of you will submit the wrong zip file, or a corrupted version. You cannot submit a corrected file later, i.e., after the submission deadline and the two day "late" window have passed.