CSCI 2500 — Computer Organization
Lab 08 (document version 1.0)

- This lab is due by the end of your lab session on Wednesday, October 26, 2022.
- This lab is to be completed **individually**. Do not share your code with anyone else.
- You **must** show your code and your solutions to a TA or mentor and answer their questions to receive credit for each checkpoint. **If you do not show your code to the lab TA/mentor to get a checkpoint, you will not receive any credit**.
- Labs are available on Tuesdays before your lab session. Plan to start each lab early and ask questions during office hours, on the Discussion Forum on Submitty, and during your lab session.

You are given a 3-bit binary unsigned integer number `A` ($A_2A_1A_0$) and a 2-bit binary unsigned integer number `B` (B_1B_0). Define a Boolean function `D` which is 1 when `A` is divisible by `B`, and 0 otherwise. When `B` is 0, the value of `D` is undefined because of the division by zero and can be represented by a "don't care" value in the truth table. In addition, define another function `ERR` which is 1 whenever function `D` is undefined, and 0 otherwise.

A "don't care" term (typically marked as `X` or `*` in a truth table and K-map) doesn't affect the output and can therefore be taken as either 0 or 1, whichever makes your logic design easier and simpler. Please read additional information about using "don't care"s at https://www.allaboutcircuits.com/textbook/digital/chpt-8/dont-care-cells-karnaugh-map/

To help you get started, here are 3 example rows from the truth table for D and ERR.

A			B		D	ERR	Comments
A_2	A_1	A_0	$B1$	$B0$			
1	0	1	0	0	X	1	The divisibility function D is undefined because 5/0 cannot be computed in ordinary arithmetic; ERR is 1
1	0	1	1	0	0	0	5 is not divisible by 2; ERR is 0
1	1	0	1	0	1	0	6 is divisible by 2; ERR is 0

1. **Checkpoint 1:** For the first checkpoint, fill out the remaining rows of the truth table.
2. **Checkpoint 2:** For the second checkpoint, transfer the truth table you created in checkpoint 1 to K-map representation. Then use the K-map method to produce minimal (i.e., using the smallest number of terms **and** variables) Boolean formulas for both `D` and `ERR`. Show all work, including the tables/groupings for the K-map method but do not forget that the final result is two Boolean formulas.
3. **Checkpoint 3:** For the third checkpoint, implement Boolean functions `D` and `ERR` from checkpoint 2 using the minimum number of logic gates `AND`, `OR`, and `NOT` where each gate has the smallest number of inputs. You may use gates with any number of inputs.

   Draw a corresponding logic circuit in 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. Test your circuit extensively to make sure that it works correctly.