Name: _____

ES 4 Problem Set 1

Due on Gradescope, Tuesday September 17 at class time (1:30pm)

Fall 2024

Problem Set 1 will give you practice with the concepts from Sections 1.5, 2.1, and 2.2 in the textbook.

After solving the problems, look at the solutions posted on the course website and categorize your work for each problem on the following scale:

- ● Completely correct
- ◕ Nearly correct, but made a small mathematical or copying error
- ◒ Solved part of the problem correctly
- ◔ Started some work in the right direction
- ○ Incorrect, or didn't even know where to start on the problem
- Include a question mark (?) in addition to one of the above symbols if you don't feel like you understand the question or the solution well enough to make a definite judgement.

Problems with an asterisk (*) are optional.

Binary numbers	1.a	1.b*	1.c	1.d	1.e	1.f*

Truth tables & canonical forms	2.a	2.b*	2.c	2.d*

Logic diagrams	3.a	3.b*	3.c*	3.d

What questions do you have about these concepts and skills?

What things are you uncertain about (even if you don't have a specific question)?

Approximately how long did it take you to complete the homework?

How long did you take going over the solutions and writing this reflection?

Turn in this self-assessment sheet on Gradescope. You do not need to turn in anything else, although we're happy to look at your work if you have questions!

Part 1: Numbers

(a) Write 35 and 22 in binary, and add them.

(b) Write 63 and 4 in binary, and add them.

(c) What is the result of 01100101 + 01110110? Write the answer in both binary and decimal.

(d) Suppose you have a circuit that can store only 6 bits, which currently has the value 20. If you add 46 to this and store the result back, what will the resulting 6-bit value be (in decimal)?

(e) What is the largest positive integer you can represent with 7 bits?

(f) Suppose you have a circuit that can store only 8 bits, which currently has the value 12. If you add 255 to this and store the result back, what will the resulting 8-bit value be (in decimal)? Do you notice anything interesting about the result?

Part 2: Truth tables

(a) Write a truth table for the logic equation $Y = A \oplus \overline{B} \oplus C$.

(b) Write a truth table for the logic equation $Y = \overline{A \oplus \overline{B}}\, C$.

(c) Write a logic equation in canonical sum-of-products and product-of-sums form for the following truth table:

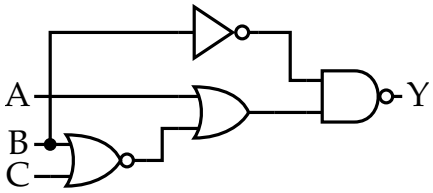
A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

(d) Write a logic equation in canonical sum-of-products and product-of-sums form for the following truth table:

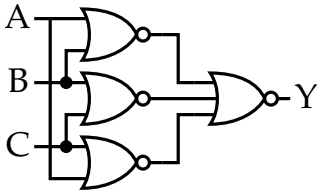
A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Part 3: Logic diagrams

(a) Write a logic equation representing the circuit below:



(b) Write a logic equation representing the circuit below:



(c) Draw a logic diagram for the boolean equation $Y = (\overline{AB} \oplus \overline{BC}) + ABC$

(d) Draw a logic diagram of the equation $AB+\overline{C}D+\overline{A}CD$ using only inverting logic (NAND/NOR/NOT). You can use whatever techniques you'd like to manipulate the equation/circuit.

Just for fun

Here's a fun theoretical problem that will stretch your analytical boolean logic skills. If you have an infinite supply of a single kind of logic gate, which other gates can you build? For example, suppose you only had inverters — you could build a buffer (using two inverters), but that's it. You can't build an AND gate or anything else. But what could you build with the other types?

You can use power and ground as constant inputs to gates if you need.

It's ok to let this one soak in your mind and keep coming back to it — with patience you'll find some neat solutions!

Additional practice

These are selected problems from the textbook (at the end of each chapter) which may be helpful for practice and review. The answers to these problems are online at `https://booksite.elsevier.com/9780128000564/solutions.php`.

- 1.13 (binary to decimal)
- 1.25 (decimal to binary)
- 1.53 (adding binary numbers)
- 2.1 (sum-of-products form)
- 2.3 (product-of-sums form)
- 2.5 (minimizing equations)
- 2.13 (minimizing equations)
- 2.7 (equations $\rightarrow$ circuit)
- 2.27 (bubble pushing)