PROJECT 1
INTRODUCTION

Deadline: 23:55 on Thursday 1st December, 2022

Last updated*: 16:06 on Saturday 12th November, 2022

Instructions

(1) The project must be implemented and submitted individually.
(2) Submissions in pairs or groups are not allowed. Plagiarism will not be tolerated.
(3) Submit exactly 1 Logisim (".circ") file under the appropriate submission box on Moodle.
The file must be named `ID_introduction.circ`, with `ID` replaced by your 9 digit ID number. For example, if your ID is 123456789, your filename should be `123456789_introduction.circ`.
(4) Use the provided `template_introduction.circ` file as a template, and implement your designs in the respective circuits. *Do not* move or modify the input/output ports, the "blackbox" layout, and the names of the circuits!
(5) If you need to use a constant 0 or 1 as a component in your circuits, use the "Constant" component from the "Wiring" library.
(6) Use tunnels to keep your circuit clean and free of spaghetti wiring. Do not remove the tunnels provided in the template. Use tunnels with matching names to connect the inputs/outputs to your circuit. Note: There is no limit on the number of tunnels corresponding to a signal. As long as they have the same name, any number of tunnels can be used for a single signal.
(7) All submissions will be graded using an automated grading system. There will be *no manual grading*, and the grade you receive will be based *exclusively* on the functionality of your circuits. No credit will be given purely for "attempted" solutions.
(8) In order to ensure that your file is compatible with the automated grading system, and to get a preliminary evaluation of the functionality, you are provided access to a validation system. In order to use this validation system, you need to send an email to `DLSBodek@gmail.com` with your Logisim (".circ") file attached.
(9) If your file contains compatibility issues and/or errors, the validation system will send you a reply with a list of error which you need to fix.
(10) If your file contains no compatibility issues and errors, the validation system will run a small number of tests on your circuits, and will

*This file may be updated once the project has been released, in order to fix mistakes and add clarifications. It is recommended to always download and use the latest file.

return the percentage of correct outputs observed. Note that this validation system checks only a small fraction of the possible inputs to your circuit. If all of these outputs are as expected, it means that your implementation may or may not be correct. However, if some of these outputs are not as expected, it means that your implementation is certainly not correct. In other words, if the preliminary score given by the validation system is less than 100 %, you still need to fix your implementation; but a preliminary score of 100 % does not guarantee a final grade of 100 %.

(11) Before submitting the file to Moodle, make sure you have validated it, and have received a "No compatibility issues detected." remark. In addition, make sure that your preliminary score is as high as possible.

(12) You *may not* use gates with fan-in[†] larger than 2. Splitters with any fan-in are acceptable.

(13) Unless specified otherwise, you may use components only from the following libraries provided by Logisim:
   (a) Wiring
   (b) Gates (basic gates only[‡])
   (c) Base

(14) Unless specified otherwise, your implementation of a circuit in Exercise m of Project n may use circuits you designed in Exercises $m - i$ in Projects $n - j$, for all $0 < i \leq m$, $0 \leq j \leq n$.

[†]The fan-in of a gate g is the number of input terminals of g, i.e., the number of bits in the domain of the Boolean function that specifies the functionality of g.

[‡]Basic gates include NOT, AND, OR, NAND, NOR, XOR, and NXOR. Parity gates, controlled buffers, and controlled inverters do not fall under basic gates.

1. Propositional Logic

Exercise 1.

Complete the circuit `comb` from the template. Use only AND, OR, NOT gates to implement it.

Input: $a, b, c \in \{0, 1\}$
Output: $y \in \{0, 1\}$
Functionality: $y = \overline{a} + b \cdot c$

Check the truth-table (in the "Analyze Circuit" option) for correctness.