DIGITAL LOGIC SYSTEMS - SPRING 2019
PROJECT 1: INTRODUCTION TO LOGIC DESIGN

Deadline: Saturday 27th April, 2019

This project consists of four assignments in which you must implement several simple logical circuits.

Submission Instructions

(1) Submit exactly 1 Logisim (".circ") file. The file must be named `ID1_ID2_introduction.circ`, with `ID1` and `ID2` replaced by each partner's 9 digit ID number.

(2) 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!

(3) Only one of the students in a pair needs to upload the submission. Do not upload the same work twice!

(4) You may not use gates with fan-in larger than 2.

Assignment 1 - Propositional Logic

(1) Complete the circuit `implication` from the template. Use only AND, OR, INV gates to implement it.
Input: $x, y \in \{0, 1\}$
Output: $z \in \{0, 1\}$
Functionality: $z = 1$ iff $x \to y$
Check the truth-table (in the "Analyze Circuit" option) for correctness.

(2) Complete the circuit `parity` from the template. Use only AND, OR, INV gates to implement it.
Input: $x, y \in \{0, 1\}$
Output: $z \in \{0, 1\}$
Functionality: $z = x \oplus y$
Where $\oplus$ is the XOR operator. You may not use the XOR-gate. Check the truth-table (in the "Analyze Circuit" option) for correctness.

(3) Complete the circuit `equivalence` from the template. Use only AND, OR, INV gates, or circuits from the previous sections to implement it.
Input: $x, y \in \{0, 1\}$
Output: $z \in \{0, 1\}$
Functionality: $z = 1$ iff $x = y$
Check the truth-table (in the "Analyze Circuit" option) for correctness.

(4) In the circuit `tautology` from the template, place a circuit instance of your `parity` function, and a circuit instance of Logisim's XOR

gate. Feed both the XOR gate and your `parity` circuit with the same inputs (x,y) and plug the outputs of both circuits into an instance of your `equivalence`. Connect the output of the `equivalence` circuit to the output port z. Can you observe a tautology in the truth table?

Assignment 2 - Trees

A zero-tester of an n-bit binary sting is denoted ZT(n) is defined as follows.

Input: $x[n-1:0] \in \{0,1\}^n$
Output: $y \in \{0,1\}$
Functionality: $y = 1$ iff $x[n-1:0] = 0^n$

(1) Design ZT(8) based on an OR-tree. You may also use inverters if needed. Complete the circuit `ztor` using your design.
(2) Design ZT(8) based on an AND-tree. You may also use inverters if needed. Complete the circuit `ztand` using your design.

Assignment 3 - Masks

Consider the definition of Parity-Filter circuit, hereinafter denoted as PF(n):

Input: $x[n-1:0] \in \{0,1\}^n$
Output: $y[n-1:0] \in \{0,1\}^n$
Functionality:

$$y[n-1:0] = \begin{cases} x[n-1:0] & ; \text{ if } \left(\sum_{i=0}^{n-1} x[i]\right) \bmod 2 = 1 \\ 0^n & ; \text{ otherwise} \end{cases}$$

(1) Preliminary design (not to be submitted): Design by hand, the PF(n) design suitable for every n, analyze its asymptotic cost and delay.
(2) Complete the circuit `pf4` with your implementation of PF(4).

Assignment 4 - NAND Gates

Complete the circuit `CMOS_NAND` from the template. Use only P-type and N-type transistors provided by Logisim.

Input: $x, y \in \{0,1\}$
Output: $z \in \{0,1\}$
Functionality: $z = \text{NAND}(x,y)$

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

Reminder:

(1) From Logisim's “Wiring” library, use the components “Power”, “Ground”, and “Transistor”.
(2) P-type transistors must be connected to “Power” and N-type transistors must be connected to “Ground”.
(3) When low voltage is applied to the gate of a P-type transistor's gate, it is activated.
(4) When high voltage is applied to the gate of an N-type transistor's gate, it is activated.

(5) Your aim is to arrange the transistors in a way that will cause the output to be connected to "Power" whenever logical 1 output is required, and to be connected to "Ground" whenever logical 0 output is required.
(6) A CMOS inverter is implemented as in fig. 1.

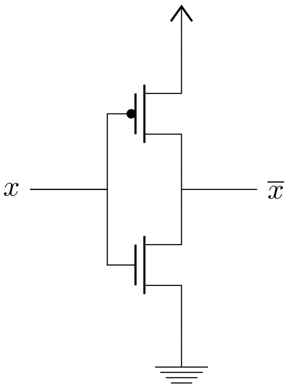

FIGURE 1. Structure of CMOS Inverter. The upward pointing arrow stands for the supply voltage (referred to as "Power" in Logisim).