Simple Combinational Circuit: Rock, Paper, Lizard, Spock

Background

For this project, you will be using Logisim 2.7.1 to create a simple combinational circuit that plays a simple matching game. The circuit requires nothing but basic logic gates, input and output pins, wires, and two LED units.

You should familiarize yourself with Logisim by working your way through the built-in tutorial.

The weekly TV documentary program *Big Bang Theory*[1] introduced a game based on the classic rock-paper-scissors game that children have played for decades. In the *BBT* game, each player chooses one of five options: rock, paper, scissors, lizard or Spock. The winner, if any, is determined by a simple set of rules that rank the "power" of each option in such a way that no single option is a guaranteed winner.

Rock-Paper-Lizard-Spock Game Circuit

In order to simplify this assignment, we will modify the *BBT* version by eliminating "scissors" as an option. (Really, we should not allow most of the cast of *BBT* to handle anything sharp anyway.)

The winner of our simplified game is determined according to the following rules:

- *rock vs paper* rock tears paper, so rock wins
- *rock vs lizard* lizard sits on rock, so lizard wins
- *rock vs Spock* Spock vaporizes rock, so Spock wins
- *paper vs lizard* lizard eats paper, so lizard wins
- *paper vs Spock* paper disproves Spock, so paper wins
- *lizard vs Spock* Spock eats lizard, so Spock wins

Other combinations, such as rock vs rock, produce no winner.

You will implement a simple circuit, named RockPaperLizardSpock, that takes inputs indicating the option chosen by each of two players, and sets two LED output components to indicate the winner, based upon those input settings and the rules given above.

When designing and implementing a circuit like this, where the input values are not naturally viewed as integers, one of the first steps is to decide how to represent the input and output data. In this case, each player must make a choice among four options, so it's natural to represent the choices (in hardware) as two-bit integers (0, 1, 2, and 3). In order to make the grading simpler, we will mandate that you represent the input values using the following mappings:

Choice	Representation
rock	00
paper	01
lizard	10
Spock	11

Therefore, the circuit will take two 2-bit inputs (so 4 input bits altogether).

As for output, there are three cases: player 1 wins, player 2 wins, and no one wins. We will represent those outcomes by 00, 11, and 01, respectively. (There's a reason we defined these in that manner; it actually simplifies the analysis of to produce a Boolean function for each output bit.)

For the output to be attractive, the circuit must be able to clearly indicate which of the two players has won, or that neither has won. A simple way to do that is to employ an LED component for each player, and to turn on a player's LED if and only if that player has won. The LED outputs should be set so that the winner, if any, is indicated by a green LED, and

non-winners are indicated by red. Note that the LED components must be controlled by using the two output bits set by the circuit, and that both LEDs should be red if there is no winner.

There will be two deliverables for the assignment. The first is a text file containing a truth table corresponding to the operation of the circuit, the Boolean functions for each of the two output bits, and simplified versions of those functions (including the algebraic derivation of the simplification.). To make grading easier, you will structure the truth table as follows:

A1	A0	B1	B0	W1	W0
0	0	0	0	?	?
0	0	0	1	?	?
0	0	1	0	?	?
.	.	.			
1	1	0	1	?	?
1	1	1	0	?	?
1	1	1	1	?	?

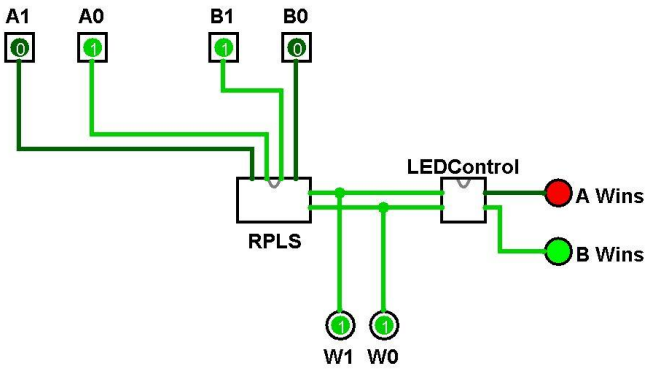
Here, A1 and A0 form the option chosen by player **A**, B1 and B0 the option chosen by player B, and W1 and W0 the output bits indicating the winner (if any). Note the rows of the table are ordered so that the 4 input bits "count" from 0000 to 1111.

The second deliverable is a single Logisim file containing your implementation of the circuit. Your main circuit must be named RockPaperLizardSpock, and must include the following interface elements:

- at the top: four 1-bit inputs, labelled A1, A0, B1, and B0, from left to right
- at the bottom: two 1-bit outputs, labelled W1 and W0, from left to right
- on the right side: two LED components, labeled A Wins and B Wins, from top to bottom

The layout and labelling of the input and output pins is important, because that determines the way Logisim will generate a truth table for your circuit, and we may use that truth table in our grading.

You are encouraged to create sub-circuits in your implementation, since that makes it possible to create a much simpler abstract view of the final design. In my case, I've implemented two sub-circuits that are shown below. RPLS takes the options choses by players A and B and computes the output bits W1 and W0. LEDControl takes W1 and W0 as input and sets the correct input signals for the two LED components. (I actually have two more sub-circuits that are not shown. Those perform computations that are used inside RPLS.)



You are not required to achieve a design that uses a minimal number of gate levels; in some cases, it's better to have more levels in order to create a more modular design.

What to submit

You will submit a single, uncompressed tar file containing your text file and your Logisim file. If you worked with a partner, be sure the text file contains a copy of the Partners Template (see the Assignments page), including the PID and name of each partner. Otherwise, only the partner who made the submission will receive credit.

The *Student Guide* and other pertinent information, such as the link to the proper submit page, can be found at:

http://www.cs.vt.edu/curator/

[1] It is a common misconception that Big Bang Theory is a situational comedy (sitcom). It is actually a documentary study of the sort of people we interact with on a daily basis.