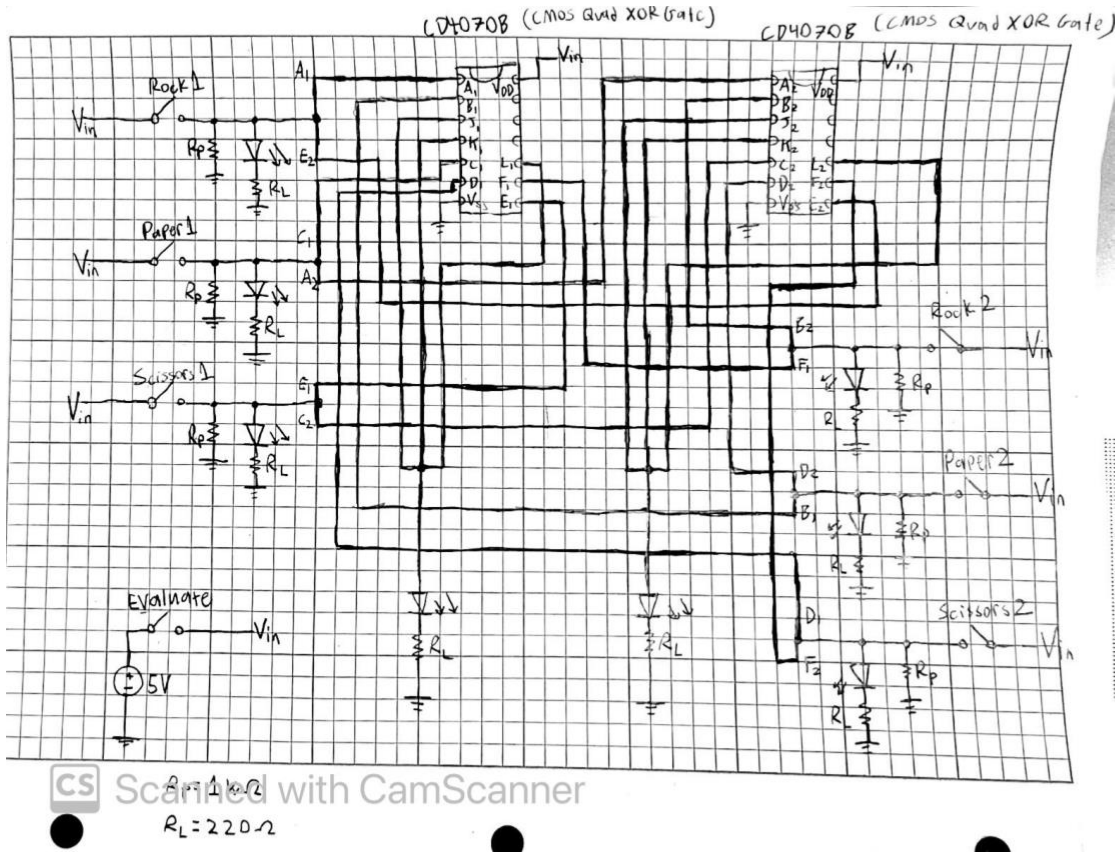
E1201 Project Report: Rock, Paper, Scissors

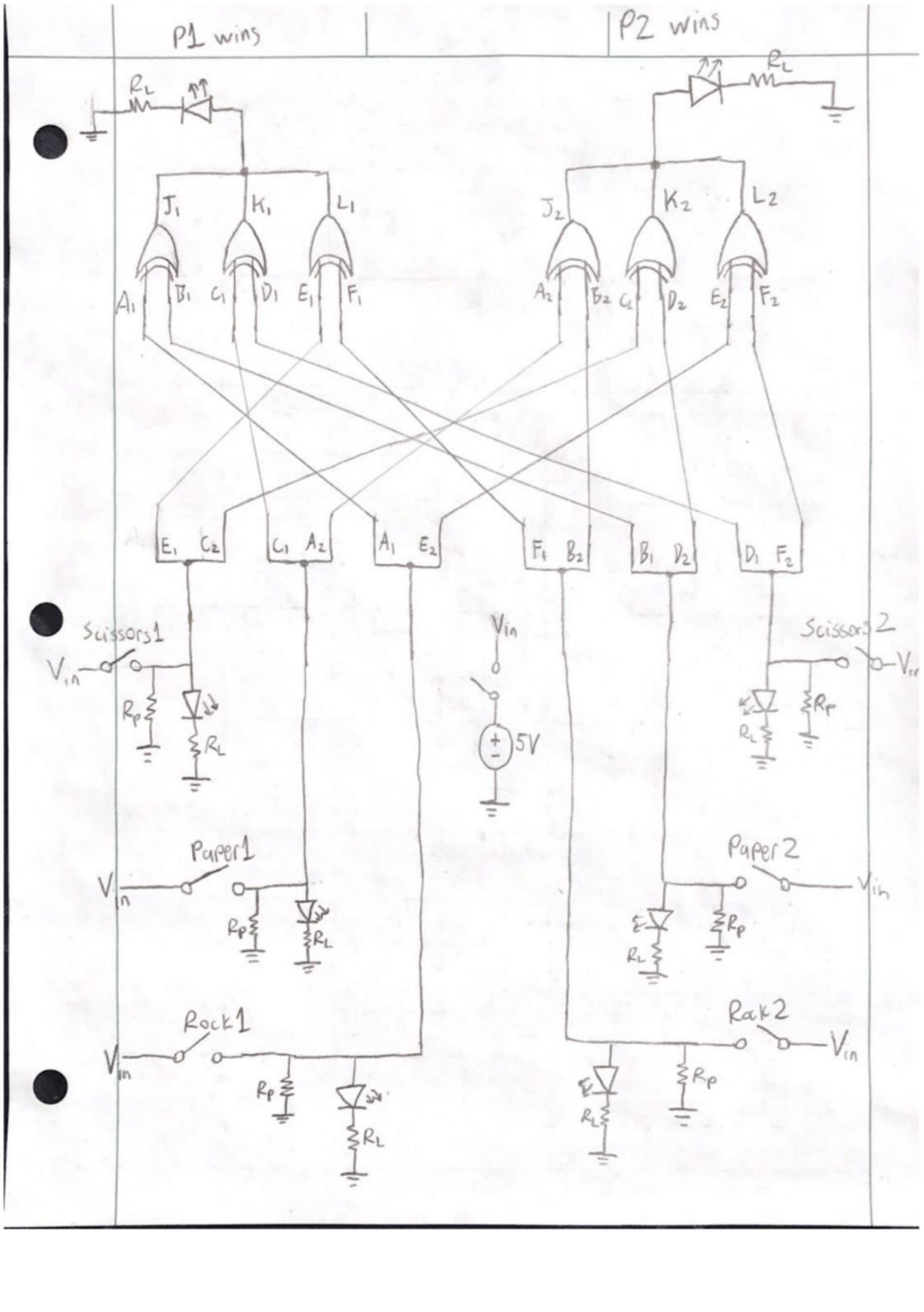
Christopher Ades; Jason Jin
UNI: ca2972 ; UNI: hj2602

Abstract

This project simulates a game of “rock, paper, scissors” using just 6 XOR gates. Two players play and each player makes one selection out of three choices (rock, paper, or scissors). After both players choose, an “evaluate” switch is toggled on to show what each player chose and the winner between the two.
With the rules: a) Rock beats Paper
b) Paper beats Scissors
c) Scissors beats Paper

Circuit Schematics


CD4070B (CMOS Quad XOR Gate)
CD4070B (CMOS Quad XOR Gate)
Rock 1
Paper 1
Scissors 1
Rock 2
Paper 2
Scissors 2
Evaluate
V_{in}
5V
R_p
R_L
A_1 B_1 J_1 K_1 C_1 D_1 V_{SS} V_{DD} L_1 F_1 E_1
A_2 B_2 J_2 K_2 C_2 D_2 V_{SS} V_{DD} L_2 F_2 E_2
$R_p = 1k\Omega$
$R_L = 220\Omega$



P1 wins
P2 wins
R_L
J_1
K_1
L_1
J_2
K_2
L_2
A_1
B_1
C_1
D_1
E_1
F_1
A_2
B_2
C_2
D_2
E_2
F_2
Scissors 1
Scissors 2
Paper 1
Paper 2
Rock 1
Rock 2
V_{in}
R_p
5V

The two schematics above are virtually equivalent. The difference is that the first one incorporates two CD4070B CMOS Quad XOR Gates while the second one just shows the logic gates (with the same input and output labels) themselves.

A DC 5V power supply coming from an arduino was connected in series with the evaluator switch. When the evaluator switch is open, the circuit receives no power. When the switch is closed, power runs through the circuit as normal. The purpose of the evaluator switch is so that the players' choices aren't revealed prematurely. It ensures that the choices each player makes and the winner of the match are shown all at once when the switch is closed.

In series with the evaluator switch is the node Vin, which is connected to 6 different choice switches (Rock1, Paper1, Scissors1, Rock2, Paper2, or Scissors2). Player 1 can close only one of the three switches: Rock1, Paper1, Scissors1. Similarly, Player2 can close only one of the three: Rock2, Paper2, or Scissors2. Assuming the evaluator switch is closed (to supply power), when a choice switch is closed, a corresponding LED (connected in parallel to the respective closed choice switch) is turned on.

We chose RL=220Ohms to get a decent amount of brightness from each LED. "Pull-down" resistors (RP=1kOhm) were connected in parallel with each gate input. They eliminated any error in the inputs of the two IC's and ensured the correct state of the inputs and therefore the correct state of the outputs.

Each choice switch corresponds to two inputs. For example, let's say Player1 chooses Rock1. The inputs corresponding to Rock1, inputs A1 and E2, will be given a state of 1. Because Player1 chose Rock1, this means that the inputs corresponding to Paper1 (C1 and A2) and Scissors1 (E1 and C2) will be given a state of 0. Continuing the example, let's say Player2 picks Scissors2. This would mean inputs D1 and F2 will have states of 1 and inputs B1, D2, F1, B2, will all have states of 0. Following the logic of the XOR gates, this would give $J1=A1\oplus B1=1$, $K1=C1\oplus D1=1$, $L1=E1\oplus F1=0$. If at least one of the outputs J1, K1, or L1 has a state of 1 then the winning LED is turned on for Player1. And, since $J2=A2\oplus B2=0$, $K2=C2\oplus D2=0$, and $L2=E2\oplus F2=0$, the winning LED for Player2 is left off. So Player1 won this match, concluding the case of rock beats scissors This example can be seen in the third row of the truth table below.

Truth Table for Rock, Paper, Scissors using Six XOR Gates

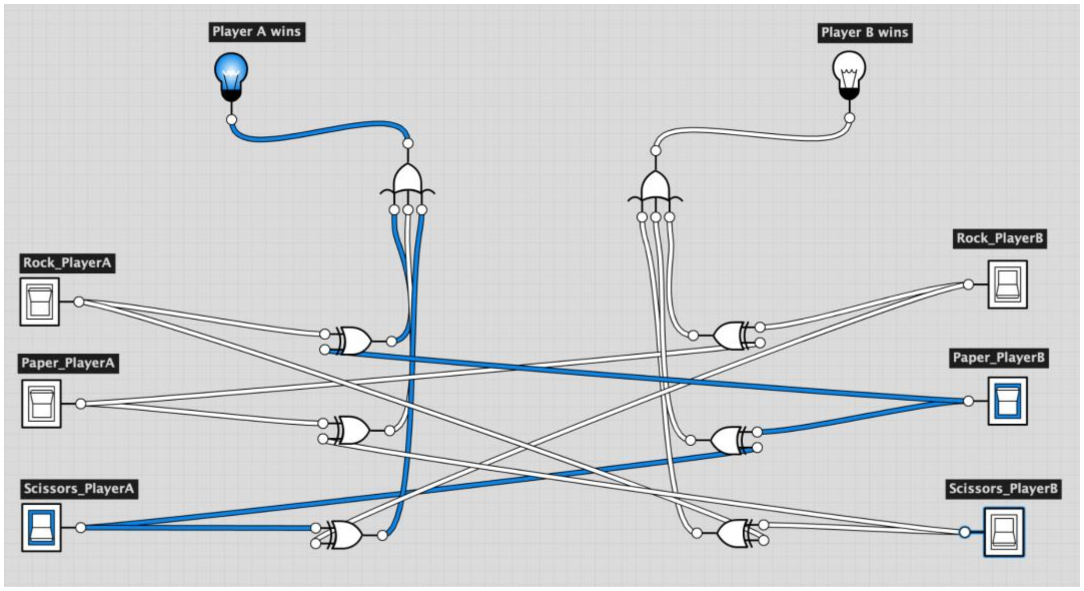
Rock 1	Paper 1	Scissors 1	Rock 2	Paper 2	Scissors 2	Player1			Player2		
A1 E2	C1 A2	E1 C2	B2 F1	D2 B1	F2 D1	$J1= A1\oplus B1$	$K1= C1\oplus D1$	$L1= E1\oplus F1$	$J2= A2\oplus B2$	$K2= C2\oplus D2$	$L2= E2\oplus F2$
1	0	0	1	0	0	1	0	1	1	0	1
1	0	0	0	1	0	0	0	0	0	1	1
1	0	0	0	0	1	1	1	0	0	0	0
0	1	0	1	0	0	0	1	1	0	0	0
0	1	0	0	1	0	1	1	0	1	1	0
0	1	0	0	0	1	0	0	0	1	0	1
0	0	1	1	0	0	0	0	0	1	1	0
0	0	1	0	1	0	1	0	1	0	0	0
0	0	1	0	0	1	0	1	1	0	1	1
0	0	0	0	0	0	0	0	0	0	0	0

The inputs are color coded to their respective outputs. The magenta highlights indicate an instance where a player's winning LED is turned on. In the case when it's a tie, as in the first row of the table (Rock1 and Rock2 are chosen), both player's LEDs light up.

It should be noted that there was never a case where all 3 XOR gates for a player output a value of 1. In the case where a player's LED turned on, it was always due to two out the three of a player's XOR gates outputting a value of 1.

Notes

Logicly:



Used Logicly before building to brainstorm(not the exact schematic)

We wanted to minimize the amount of gates used in the project. We figured that at least six gates were necessary to make the game work properly since there will always be six “choices” made in a match. For example if Player1 chooses Rock1=1 then that implies that they choose Paper1=0 and Scissors1=0, for a total of three “choices” made. Similarly, Player 2 technically makes three choices as well. So in total there would be six choices made each match. We also thought that each gate must take input from both players. Just browsing through the truth tables for the different gates to use, the XOR gate turned out to fit in perfectly with our six gate idea.

While building the circuit there was a considerable amount of time dedicated to figuring out how the CD4070B worked. Although all the inputs were properly placed the outputs kept malfunctioning and not following the truth table for an XOR gate. Sometimes the output would just flash on and off at random or if our hands got close to the circuit. The point is that gates were sensitive and errorful. After doing some research we came across a video on youtube by EE Wave (https://www.youtube.com/watch?v=NpPq0eXMQzQ) showing how to use an IC similar to the one we were using. What he did differently than us was add “pull-down” resistors in parallel with the inputs. Without a “pull-down” resistor, when one of the “choice” switches is open, the associated inputs are unconnected from either a defined 1 or 0 state. This means that the unconnected inputs have the potential to bounce between the states of 0 and 1. The value of the unconnected input can change at the smallest interference or noise from other nearby inputs. A “pull-down” resistor ensures that the state of an input is 0 by shorting it to ground whenever a “choice” switch is open. Once we implemented “pull-down” resistors to each input, the circuit worked flawlessly.