Ahmad Adil 12/5/20
EE 210
Project Report

https://www.youtube.com/watch?v=XJIDuqZPV3M&feature=youtu.be

Introduction:

Rock Papers Scissors is a synchronized zero-sum game that is played with usually two players and has only two possible outcomes which are a tie, win or loss for either player. The rules of the game are that rock beats scissors, scissors beats paper, and paper beats rock. The motivation for this project was to be able to explore and understand the basics of game logic. Rock Paper Scissors gives a unique look into this as it allows various states for the game and can be built in various different ways to achieve the same goal. This creates interesting problem complexity to the game as it can be extremely modular in design allowing for various features in game.The main goal of the circuit is to be able to successfully implement a rock paper scissors game in logisim with two player inputs, indicators for win, losses and ties and a scoreboard for each player for their wins. This is an interesting field as it allows me to explore game logic design at a fundamental level.

Method:

The way this problem was approached was by building first the core game logic of the circuit. The inputs of the game are in binary so I would need to assign values to the inputs that would correspond to rock paper scissors. For the game the bit pattern that was made 01 - rock, 10 - paper and 11 - scissors. The bit pattern 00, which is what both the inputs start at is considered illegal, and the evaluation circuits ignore. For the inputs I had labeled A,B to be the inputs for player 1 and C,D to be the inputs for player 2. This way it made the truth tables account for all possible inputs that each of the players could do. For the outputs I had assigned a win and a win/tie. The win output would occur if either player wins based on the rules. For the win/tie, I considered that if both players tie they both would get one point as well. The circuit is split into two parts, a left subcircuit and a right subcircuit. These subcircuits are similar to each other, however the big difference comes from which player it is counting the score for.

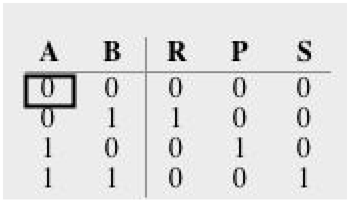
These circuits were built from a 4-bit adder and a 4-bit counter. The 4-bit counter combines the adder and a counter circuit to help count the current score for the players. This was a difficult part as I had gone through various iterations to figure out how I would take the output of each circuit and place it into a hex display in order for the circuit to work. This was solved by using a bit extender for the inputs once they had been processed by the corresponding left/right player subcircuit. Also implemented was indicators for if the current player won, lost or tied. This was built using AND gates with a NOT gate for the win and loss output indicator.

There is an implementation of an “AI” that replaces player 2 that has been implemented. This was built using logisim’s random number generator and by giving both inputs C and D a random generator with a different seed. A different seed allows it to randomize input from the clock alternating.

Bit patterns:
Rock: 01
Paper: 10
Scissors: 11

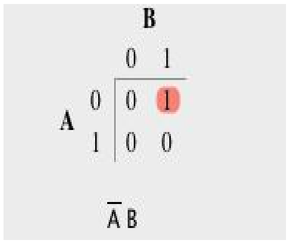
Outputs:
Win/tie: 1 if player wins or ties. 0 if loses or illegal
Win: 1 if player wins 0 for everything else

Truth table for player selection:



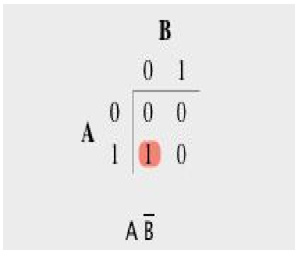
A	B	R	P	S
$\boxed{0}$	0	0	0	0
0	1	1	0	0
1	0	0	1	0
1	1	0	0	1

Kmaps:

R: 

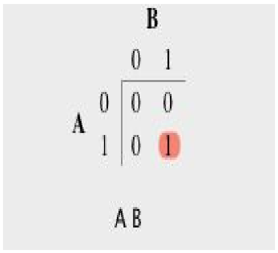
A \ B	0	1
0	0	1
1	0	0

$\bar{A}\,B$

P: 

A \ B	0	1
0	0	0
1	1	0

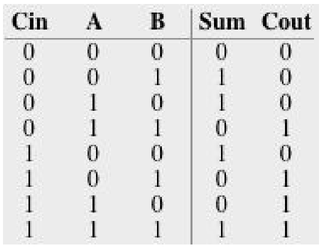
$A\,\bar{B}$

S: 

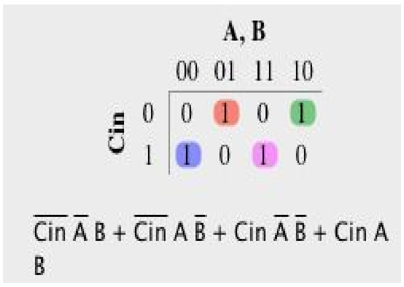
A \ B	0	1
0	0	0
1	0	1

$A\,B$

1- bit adder:

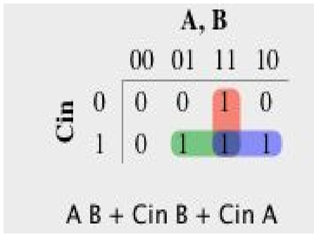


Cin	A	B	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Sum: 

Cin \ A, B	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$\overline{Cin}\,\bar{A}\,B + \overline{Cin}\,A\,\bar{B} + Cin\,\bar{A}\,\bar{B} + Cin\,A\,B$

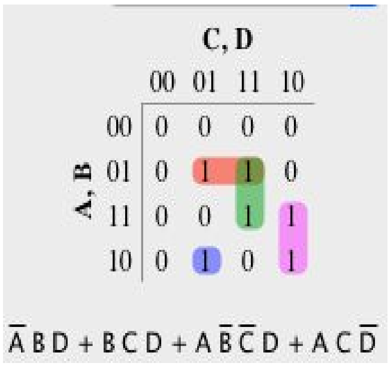
Cout: 

Cin \ A, B	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$A\,B + Cin\,B + Cin\,A$

Left player table:

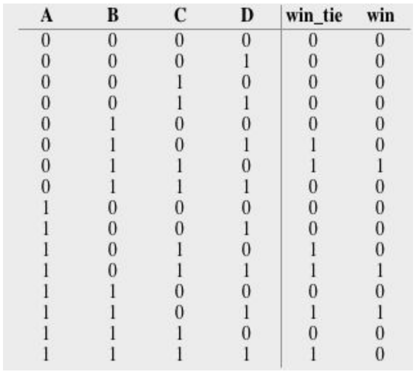
Left player Win/tie:



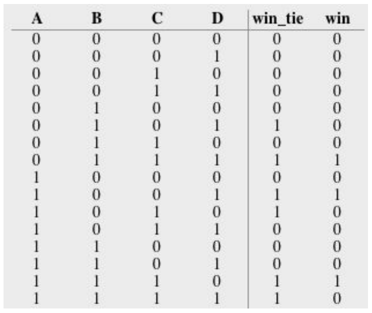
A, B \ C, D	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	0	0	1	1
10	0	1	0	1

$\bar{A}\,B\,D + B\,C\,D + A\,\bar{B}\,\bar{C}\,D + A\,C\,\bar{D}$

Ahmad Adil 12/5/20
EE 210
Project Report

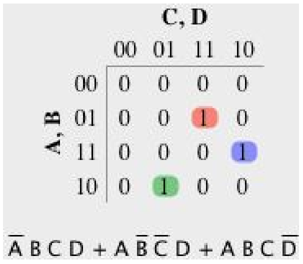


A	B	C	D	win_tie	win
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	0	0
0	1	0	1	1	0
0	1	1	0	1	1
0	1	1	1	0	0
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	1	0
1	0	1	1	1	1
1	1	0	0	0	0
1	1	0	1	1	1
1	1	1	0	0	0
1	1	1	1	1	0



A	B	C	D	win_tie	win
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	0	0
0	1	0	1	1	0
0	1	1	0	0	0
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	1	1	1
1	0	1	0	1	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	1	1
1	1	1	1	1	0

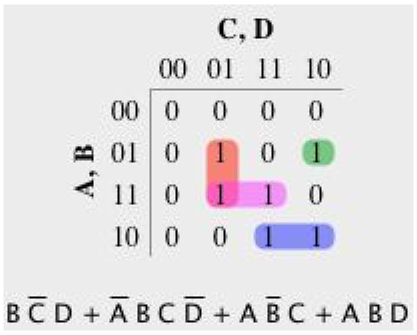
Left player win: Right player
table:



A, B \ C, D	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	0	0	1
10	0	1	0	0

$\overline{A}BCD + A\overline{B}\overline{C}D + ABC\overline{D}$

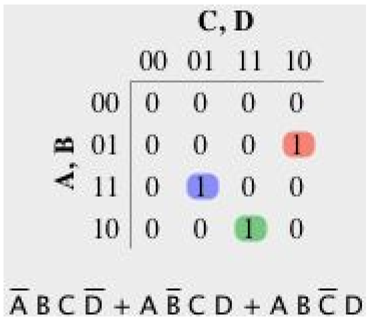
Right player win/tie:



A, B \ C, D	00	01	11	10
00	0	0	0	0
01	0	1	0	1
11	0	1	1	0
10	0	0	1	1

$B\overline{C}D + \overline{A}BC\overline{D} + A\overline{B}C + ABD$

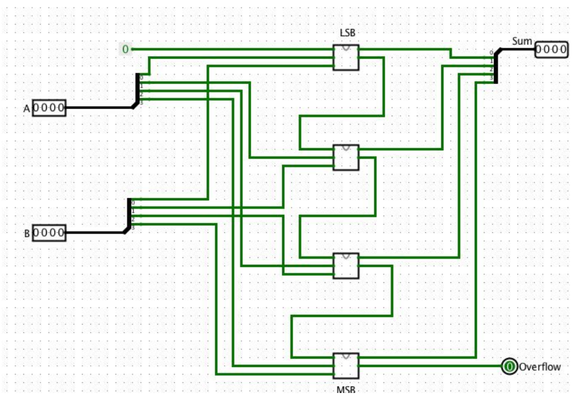
Right player win:



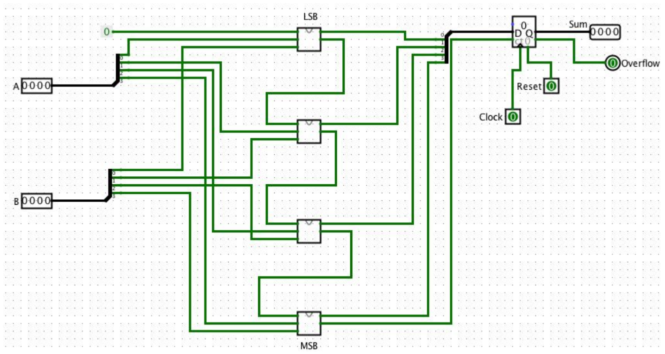
A, B \ C, D	00	01	11	10
00	0	0	0	0
01	0	0	0	1
11	0	1	0	0
10	0	0	1	0

$\overline{A}BC\overline{D} + A\overline{B}CD + AB\overline{C}D$

4-bit ripple adder:

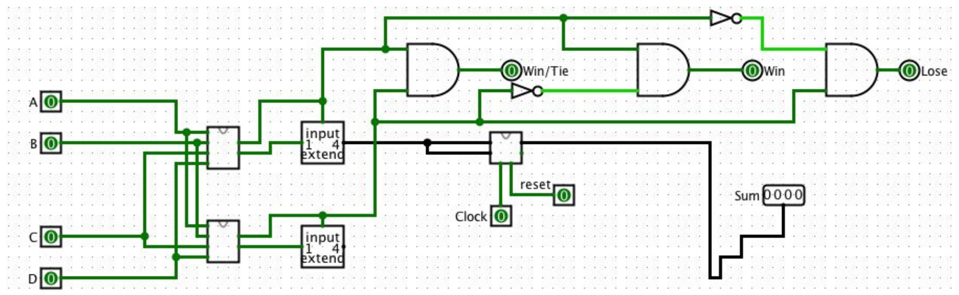


4-bit counter:



Ahmad Adil 12/5/20
EE 210
Project Report

Left/Right player subcircuit:



Results:

The final circuit that was built achieved the main goals of the project. That is to allow two player input, calculating the inputs and displaying the outputs of which player won, lost or if both tied. The inputs work based on the bit pattern assigned to them and both left player and right player subcircuits operate according to their given truth tables. One shortcoming that was noticed within the circuit is the speed of how each game is played. The game is dependent upon the clock alternating and what was noticed was if the clock input is at a higher frequency the players might not have enough time to change their inputs resulting in a wrong selection and causing the score to incorrectly update. A clock frequency of 0.5 Hz is recommended as a solution to the problem.

Another interesting result was found in the AI version of the game. In the AI circuit I used logisim built-in a random number generator to create an illusion of an AI in place of player 2. However, some of the setbacks from this was how AI doesn’t have enough randomness to it and can cause the other player to pick up on a pattern, which would develop a strategy against it. The AI also is connected to the same clock input as the subcircuits which result in a slower clock frequency for the AI.

Discussion:

In conclusion the final design of the project works as intended and can operate all of the base functions that were set out with the proposal. However, while these were achieved they can be improved later on with better designs. One such problem that can be improved is to add a delay to the inputs that would allow the players a chance to select their input. This can be implemented by using a flip flop and would overall improve the system consistency and can allow a higher base clock frequency for the subcircuits. Another part of the design that can be improved is the AI. The AI can be improved by building a custom random generator that follows the bit patterns of the game's logic rather than using the solution from the logisim. This combined with the improvements mentioned before can make for a fully functional rock paper scissors game that can be used by a wide variety of people.