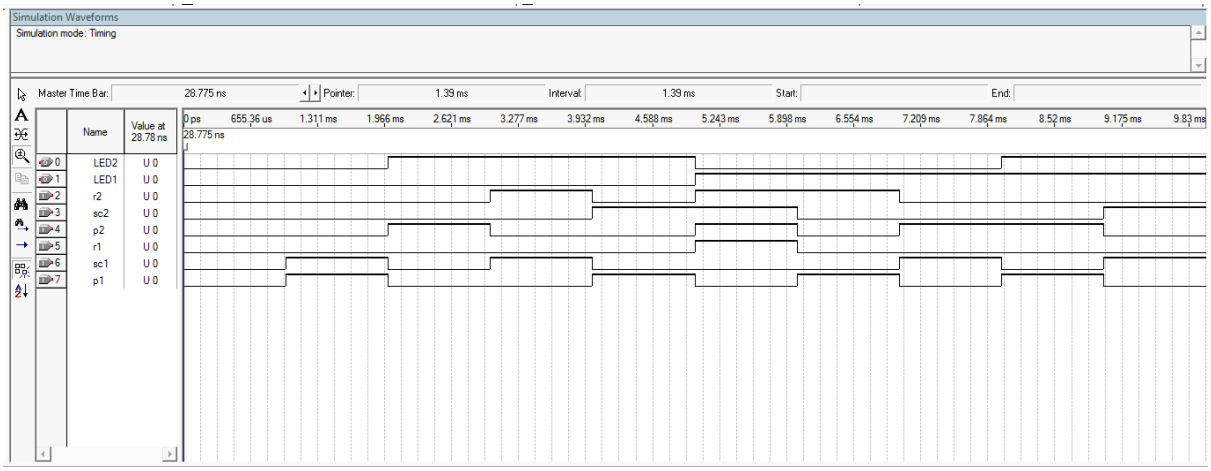
Simon Jones (100504440)
Salvador Ayala (100495832)
Group 14
Lab 2 - Final upload

Design Incentives for our VHDL implementation

For implementing the encoder blocks for player 1 and 2, we used a VHDL ‘case’ statement to first select out inputs where only one button was pressed. All other cases, including those in which players do not press a button or press multiple buttons, are captured in the final ‘others’ line of the case statement.

In effect, our implementation of the outcome logic block uses two multiplexers that use the output from the encoder block for player 1. However, the two multiplexers are created in a single case statement, because each block of the case statement determines outcomes for both LED1 and LED2, thanks to VHDL’s syntax allowing for it. We had originally tried using a type of ternary operation to easily block off the result of LED1 and LED2; however, this syntax was not supported in the version of VHDL the lab computer has. Instead, we rewrote it using the first and second canonical forms. We could have used if..else blocks to determine the output of this logical block; however we found this would take a large amount of space and would become difficult to read, so we chose this other option.

The final simulation graphic is the following one:



We considered the following cases:

- Both players do not play
- Both players also do not play, but one is due to a simultaneous press of buttons.
- One player does not play but the other does, so the one who plays wins.
- Player one wins (2 different cases)
- Player 2 wins (2 different cases)
- Player one wins (due to a simultaneous press of buttons that causes the second player’s play to be invalid and, therefore, player 1 wins).
- Two ties caused by two valid and identical plays.

From the wave function we can extract the following information, adding the proper intermediate signals for a proper conversion (encoding) of the play. The truth table would look like this:

	Player1					Player2					Result	
	P	S	R	S1	S0	P	S	R	S1	S0	LED1	LED2
Both noplay	0	0	0	0	0	0	0	0	0	0	0	0
Wrong play	1	1	0	0	0	0	0	0	0	0	0	0
No play (only1)	0	0	0	0	0	1	0	0	1	1	0	1
Win player 2	0	1	0	1	0	0	0	1	0	1	0	1
	1	0	0	1	1	0	1	0	1	0	0	1
Wrong play	0	0	1	0	1	1	1	1	0	0	1	0
Win player 1	1	0	0	1	1	0	0	1	0	1	1	0
	0	1	0	1	0	1	0	0	1	1	1	0
Tie	1	0	0	1	1	1	0	0	1	1	1	1
	0	1	0	1	0	0	1	0	1	0	1	1

In order for a proper understanding of the exercise, we also made the following truth tables so we can check both processes (encoders and logic block).

Paper-Scissors-Rock encoder truth table:

P	S	R	S1	S0
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	0
1	1	1	0	0

Player1-Player2-Result truth table (Logic Block):

S1(P1)	S0(P1)	S1(P2)	S0(P2)	LED1	LED2
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	1	1
0	1	1	0	1	0
0	1	1	1	0	1
1	0	0	0	1	0
1	0	0	1	0	1
1	0	1	0	1	1
1	0	1	1	1	0
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	0	1
1	1	1	1	1	1