Digital Logic & Design

Group Project

19th November 2020

Contents

1 Acknowledgement

The following document has been typed using Overleaf; online LaTeX editor.

The rest of this page is intentionally left blank.

2 Introduction & Logistics

Project Title: Connect-Four Video Game on FPGA

Group Members

All names appear in alphabetical order, any other apparent sequence is purely coincidental.

1. Ali Ur Rehman - ar05104
2. Maaz Saeed - ms05050
3. Mohammad Sameer Faisal - mf04709
4. Shoaib Mustafa Ali - sa04275

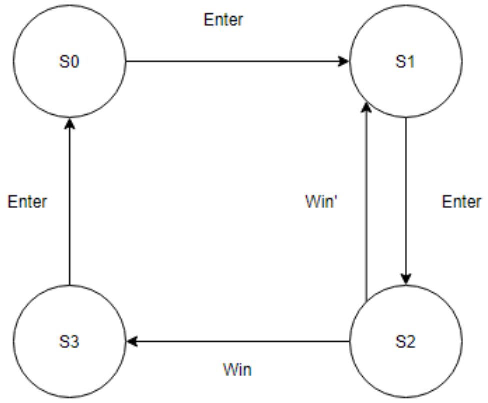
3 Design & Research

3.1 Finite State Machines (FSM)

3.1.1 Game

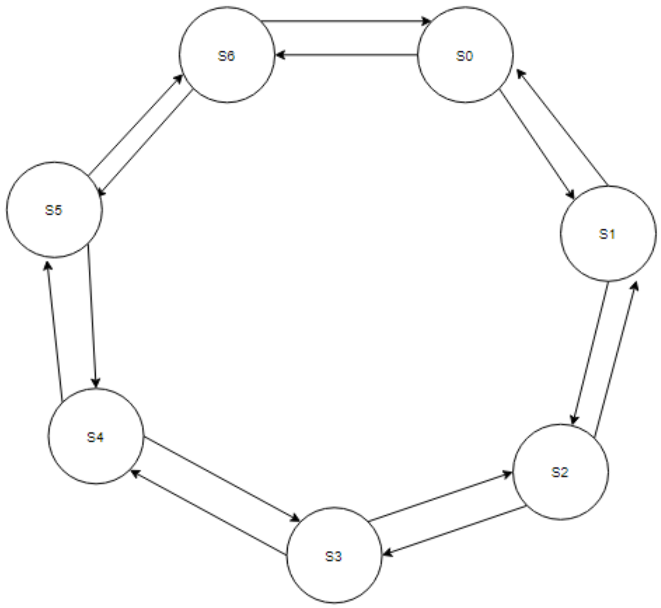
The 'game' finite state machine in our project comprises of the four states S0, S1, S2, & S3, which can be defined as follows:

1. S0 : Default State or Reset State.
   The reset state can only be exited once the user presses the enter button (an input taken from a switch).
2. S1: Column Selection State.
   In this state the player can switch between columns to pick one that he finds suitable to throw his chip in. The user transitions from this state to another only when the enter button is HIGH.
3. S2: Matrix Update and Win Checking State.
   This state updates the matrix so that the new chip is placed in the column selected by the player. It then checks whether the update in the matrix causes any of the win combinations to become HIGH. If it does then we transition to S3. If not then we go back to S1 after switching players.
4. S4: Win State.
   This state shows whether any of the players have one by displaying the winning player's colour on all matrix cells. The players can exit this state by pressing enter again.



3.1.2 Column Selection

The column selection finite state machine is fairly simple. There are seven states, each representing a single column on the game board. By default, at the start of a turn the 0th column is highlighted (and so the machine is at it's 0th state). The player can then either choose to move in either the left or right directions by one column, any number of times until they decide to place their chip. Depending on the direction they choose to move in, the states change. For e.g if you move left from the 1st column, you end up at the 7th column and so the machine moves from state S0 to S6. Similarly moving from column 0 to column 1 would mean going from S0 to S1.

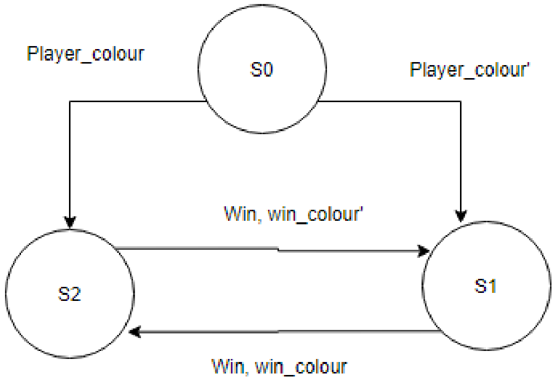


3.1.3 Cell

The game contains 42 cells in total (the matrix is 6x7). Each of these chips can have any of the following states:

1. S0: Inactive State.
   None of the players have placed their chip in this spot
2. S1: Blue State
3. S2: Red State

Due to the presence of three states of any cell, the data has to be represented by 2 bits. The MSB signifies whether or not the cell is active. The LSB signifies the colour of the chip placed in the cell, if active. The state transitions from the inactive state to either of the coloured states depending on the colour of the player who selected the cell. The transition is also only enabled when the column number, cell number and change input (indicating when the matrix has to be updated) is HIGH for this particular cell. The colours will also only change once a win condition has been detected by the win checker which would then promptly enable the colours to switch to display the colour of the player who has won.



3.2 Design Overview

Following is graphical illustration of our machine, along with brief descriptions of each component.

Game FSM

The game finite state machine allows users two options. Either to start a new instance of the game at startup, or to restart the game after a previous round has been won by a player.

Column Choosing FSM

The column choosing FSM takes input from users, to move along the board. Players can move from one column to another until they decide where to deposit their chip.

Player T Flip Flop

The player T Flip Flop enables the changing of the colours representing each player respectively. It determines the colour of the chip being deposited into the matrix.

Matrix

The matrix can be imagined as type of stack. It contains all the deposited chips in a column. By default, all columns are empty and the chip falls the bottom most cell in the selected column, then on-wards if the same column is selected the chip will be deposited one step above the bottom and so on. The matrix is also linked to the win-check.

Win Check

At each chip deposit, the matrix will be checked to see if four chips of the same colour have been placed in a row, column or diagonal. If not, the game continues normally. If yes, the win condition is fulfilled and a victor is declared. The game then moves back to the 'game' finite state machine where we can restart and play another round.

Columns

Inside the matrix, are seven columns, each representing one of seven columns on the game board. Players can choose which column to drop their chip inside of.

Cell Selector

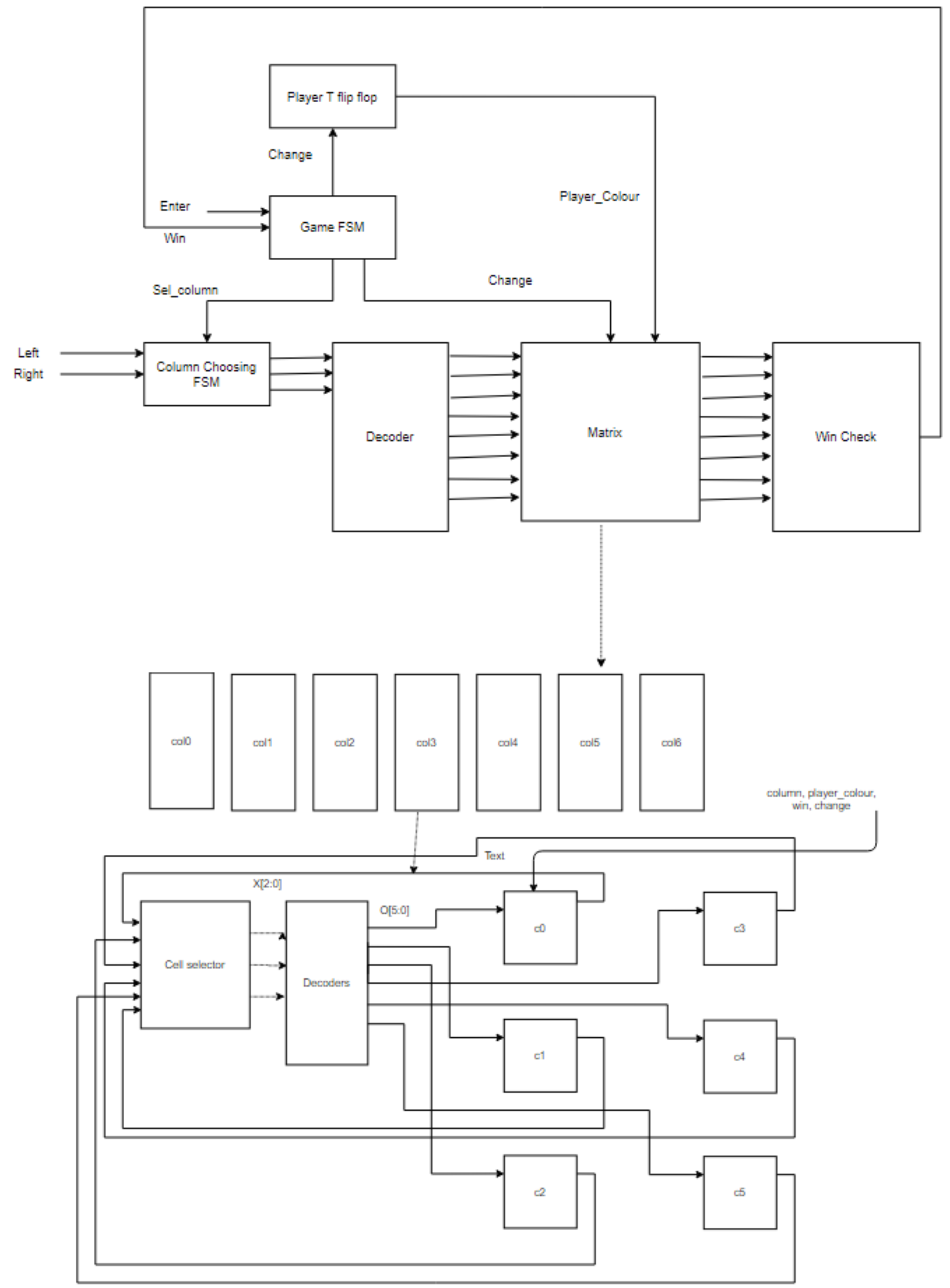
Each column on the game board contains six cells, chips are deposited into these cells starting from the bottom-up, stacked on top of one another. The cell selector manages which cells are already occupied and where the newly deposited chip will fall.

Cells

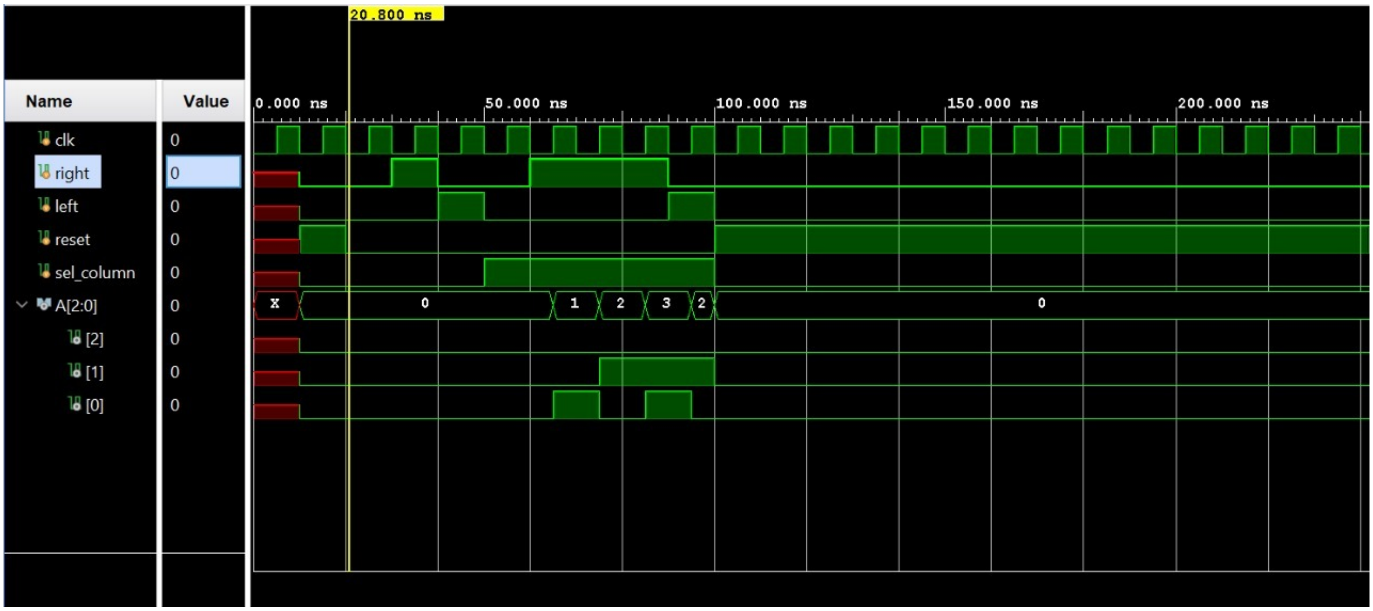
Each cell is able to contain one deposited chip. The cells are connected to the cell selector and enable it to decide whether a chip can be deposited at a certain cell or not (is the cell available or already filled).

Decoders

The decoders are used to determine which column as well as which cell has to be activated for the update to occur.


Player T flip flop
Change
Enter
Win
Game FSM
Player_Colour
Sel_column
Change
Left
Right
Column Choosing FSM
Decoder
Matrix
Win Check
col0
col1
col2
col3
col4
col5
col6
column, player_colour, win, change
Text
X[2:0]
O[5:0]
Cell selector
Decoders
c0
c1
c2
c3
c4
c5

3.3 Simulation Snippet


20.800 ns
Name
Value
clk
0
right
0
left
0
reset
0
sel_column
0
A[2:0]
0
[2]
0
[1]
0
[0]
0
0.000 ns
50.000 ns
100.000 ns
150.000 ns
200.000 ns
X
0
1
2
3
2
0