Tic Tac Toe using Verilog

November 22

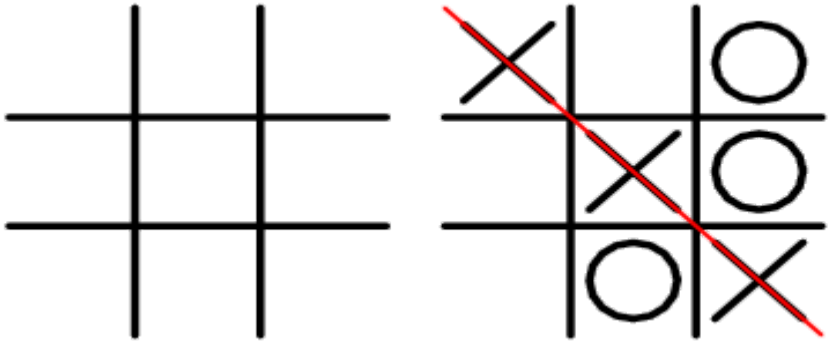
Rohan(2020CSB1117)
Raghav Patidar(2020CSB1115)

Instructor:
Dr. Neeraj Goel

Summary : **Tic-tac-toe**, noughts and crosses , or Xs and Os is a paper-and-pencil game for two players who take turns marking the spaces in a 3X3 grid with X or O. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row is the winner. It is a solved game, with a forced draw assuming best play from both players.

---

1. Introduction



This project is based on the implementation of a well known game called Tic-Tac-Toe using Verilog.
In our game, the player plays the Tic Tac Toe game with a computer. When the player/ computer plays the game, a 2-bit value is stored into one of the nine positions in the 3x3 grid. It stores :

- **2'b00** - when neither the player or computer played in that position.
- **2'b01** (X) - when the player played in the position
- **2'b10** (O) - when the computer plays in the position.

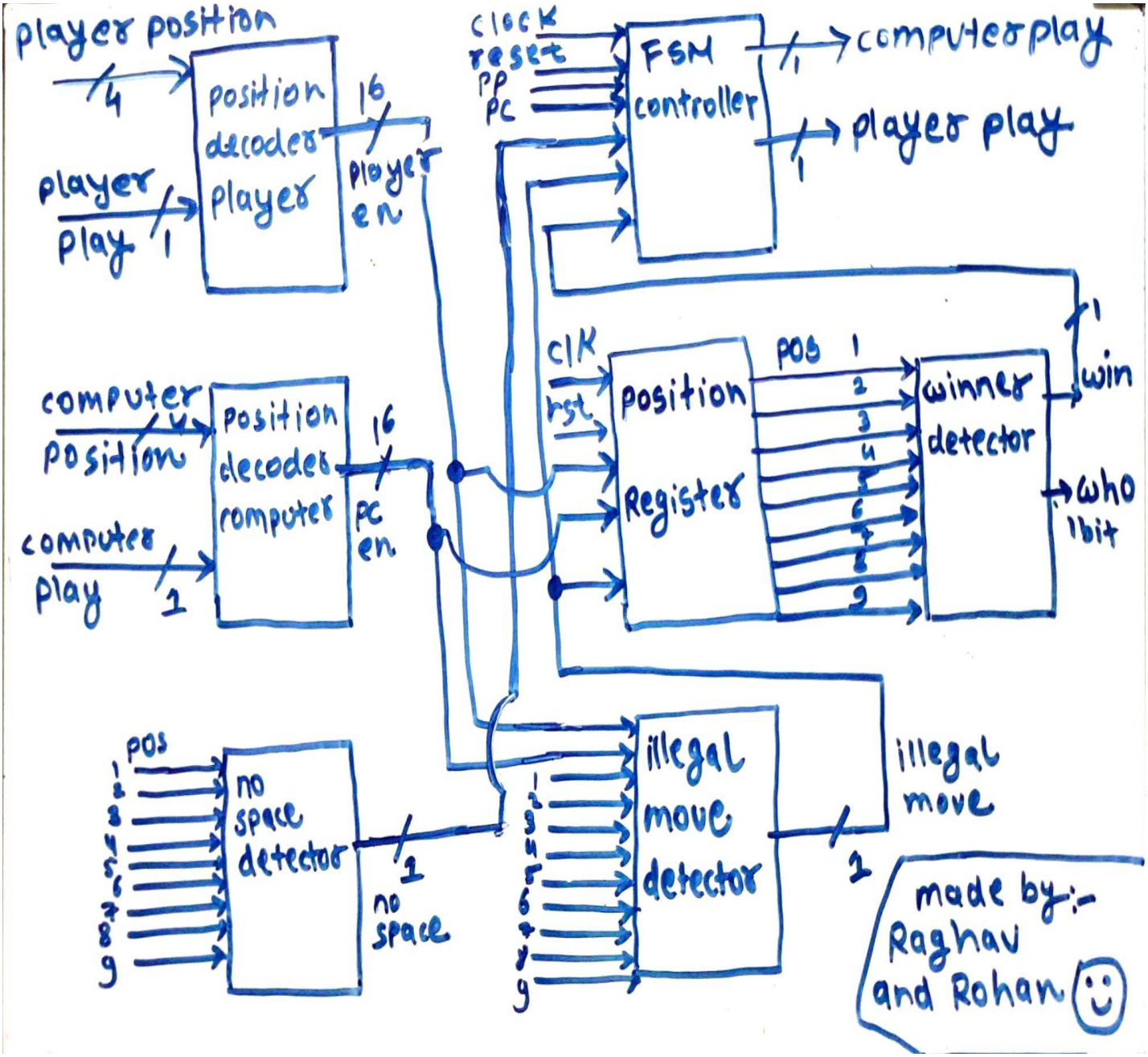
The winners is announced as a 2 bit value, where :

- **2'b01** -> Player won
- **2'b10** -> Computer won

For the implementation part, several modules are created which detects illegal moves, decodes position, stores values at positions,

2. Illustration and Algorithms

2.1 Diagram :



2.2 Example :

- Let us understand it better using an example :
  Consider the 3X3 grid given below :

1	2	3
4	5	6
7	8	9

- When a player plays, a 2 bit value gets stored at the position he wants to mark his symbol. In our implementation, we defined 00 for ideal state, 01 for player and 10 for computer.
- So, initially the 3X3 matrix is initialized with 00 states in all positions.
- The player/ computer wins the game when successfully placing three similar (01-Xs) or (10-Os) values in the following pairs of positions:
  row{(1,2,3), (4,5,6), (7,8,9)} or **column**{(1,4,7), (2,5,8);(3,6,9)} or **diagonal**{(1,5,9);(3,5,7)}.
- It is important to keep in mind illegal moves and space left in the 3X3 grid.
- Winner is announced as 2 bit value, which can be : **00** , **01** or **10** .

2.3 Algorithms/Logic :

2.3.1 DESIGN.v

Our main module is Tic-Tac-Toe. In this module we instantiate various modules which eventually helps in deciding the output.
Various modules are:

- **Position Register :** To store positions of players when FSM controller is enabled
- **FSM Controller :** Controls the gameplay
- **NoSpaceDtector :** To detect if there is more available space to play.
- **illegalMoveDetector :** Ensures fair gameplay by ensuring all moves are correct.
- **position_decoder :** To decode the position being played.
- **winner_detector_across_line :** Take value at 3 positions of grid and tell if there is a winner in it.
- **winner_detector :** Runs over all possible lines {rows, columns and diagonals} and checks if there is a winner from them.

Here is detailed explanation and working of modules :

Position_decoder : When a player plays at a particular position it will be any digit from 1 to 9 hence requires 4 bits, so this module has 4 to 16 decoder which decode the positions and enable writing to corresponding registers.

NoSpaceDtector : It takes all the positions (pos1,pos2…..pos9) as a input and checks the space in the 3*3 matrix and if space is available then it outputs nospace which is 1 if there is nospace otherwise 0.

Position Register : It takes clock,reset,illegalmove, PC_enable_signal, PL_enable_signal as an input and stores player and computer positions when enabled by fsm controller. Its outputs are pos1,pos2…...pos9.

illegalMoveDetector : This module checks the position chosen by the player/computer to play . If this position is already chosen then it is an illegal move and in this case output is 1 otherwise 0.It takes input pos1,pos2,pos3...pos9, PC_enable_signal,PL_enable_signal and outputs an illegal_move.

Winner_detector : This module takes input pos1,pos3,pos3,...pos9 and outputs the winning signals and the winner. It checks the positions (1,2,3); (4,5,6);(7,8,9); (1,4,7); (2,5,8);(3,6,9); (1,5,9);(3,5,7) and decides the winner.

FSM Controller : It basically controls the game play. It controls how the player and computer play the game.It has inputs which includes clock,reset,player chance , computer chance , illegal move, no space, win and has two outputs which are computer enable signals and player enable signals.

2.2.2 Test_bench.v

Here, we took 3 possible cases through which we can see that our design tells whether the player wins, the computer wins or it's an incomplete match.

First of all we instantiated the test_bench module with the main module of design. Then, we took three possible matches between players.

Initially the clock cycle is defined using the forever statement.

Player enable signals (denoted by pp) and Computer enable signals (denoted by pc) are enabled/disabled based on whose turn is going on.

Positions of computer and player are registered by computer_position and player_position respectively.

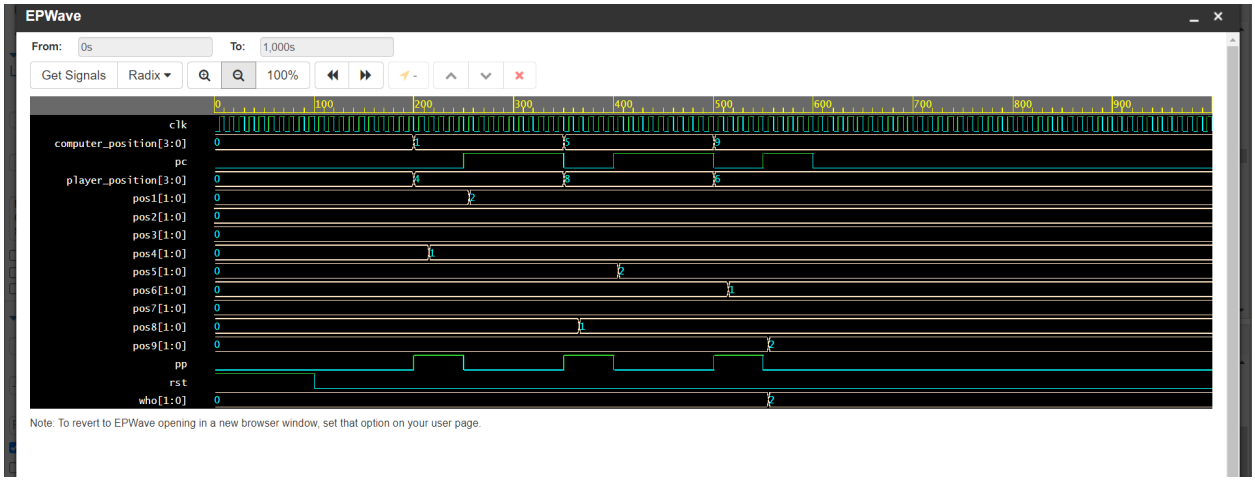
We have nine 2 bit positions represented by pos1, pos2, ... pos9, which are updated whenever any of the players register their positions. Initially all the positions are 00.

Example : If any player or computer registers position6, then pos6 is updated to 01(for player) or 10(for computer).

Reset button (denoted by rst) is for re-initializing the 3X3 grid to make all positions equal to 00. So, we made rst=1 at starting and rst is maintained 0 throughout a particular match.

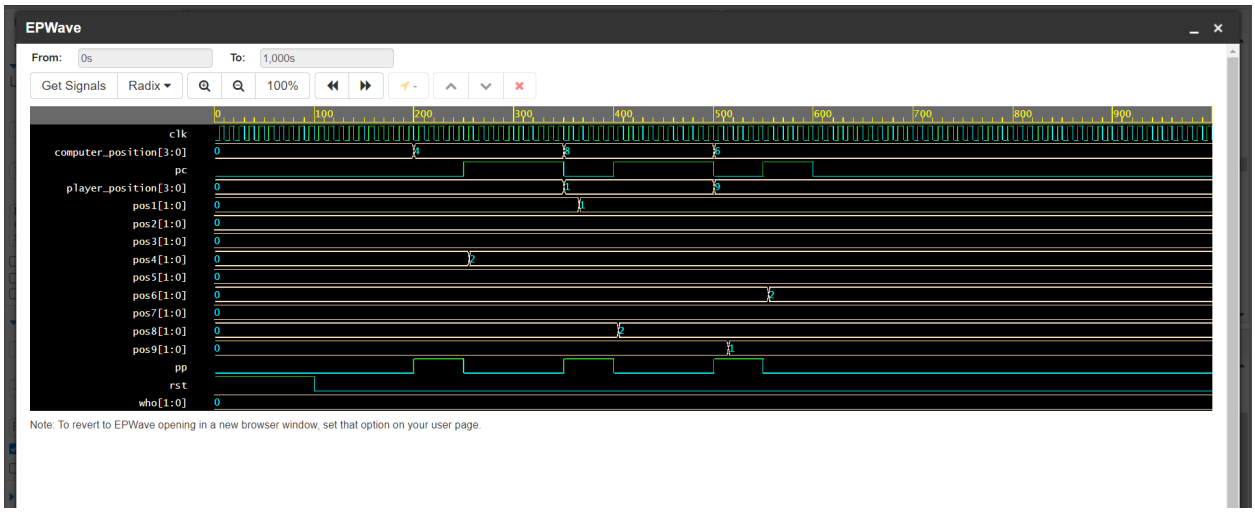
CASE I :

In the first case, we took an example where the computer is winning the match. At last, we will be able to see the final 3X3 matrix generated and the output winner : 10.



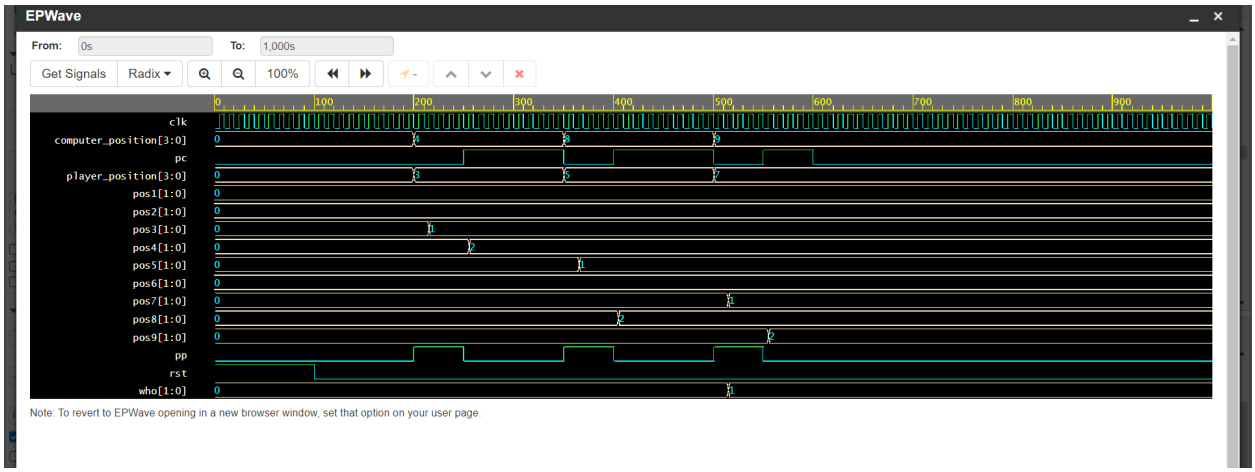
CASE II :

In the second case, we took an example where no one is winning the match. At last, we will be able to see the final 3X3 matrix generated and the output winner : 00.



CASE III :

In the third case, we took an example where the player is winning the match. At last, we will be able to see the final 3X3 matrix generated and the output winner : 01.



3. Conclusions

Tic-Tac-Toe is a 2 player game in which both players mark their symbols over a 3X3 grid turn by turn. The player who makes the first three of their marks in a diagonal, vertical, or horizontal row wins the game. Because of the simplicity of tic-tac-toe, it is often used as a pedagogical tool for teaching the concepts of good sportsmanship. This game contributes to children's developmental growth in numerous ways including their understanding of predictability, problem solving, spatial reasoning, hand-eye coordination, turn taking, and strategizing.

Acknowledgements

Me, Rohan and my team member Raghav would like to express special thanks to our teacher **Dr. Neeraj Goel** who gave us the golden opportunity to do this project on the topic Tic-Tac-Toe game using Verilog. This project really helped us doing a lot of research and we got to learn many new things.
Secondly, I would like to thank all my friends who helped me in finishing this project before time.
Rohan(2020CSB1117)
Raghav Patidar(2020CSB1115)

References

[1] Wikipedia : https://en.wikipedia.org/wiki/Tic-tac-toe
[2] Charles H. Roth, Jr. Larry L. Kinney *Fundamentals of Logic Design* . USA, 2010

Appendix A

Charles H. Roth, Jr. Larry L. Kinney *Fundamentals of Logic Design* . USA, 2010