PROJECT REPORT
TIC - TAC - TOE

Tic-tac-toe (noughts and crosses) or X and O is a classic game for two players, *X* and *O*, who take turns marking the spaces in a 3×3 grid. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row is the winner.

The game has been designed to play using FPGA switches to enter the position where the player wants to enter an X (01 in the programme) or O(10 in the programme) . The switches take four bit input. For example, if the first and last switch is on, the middle two are off, the position being entered is 9 (1 0 0 1)

1	2	3
4	5	6
7	8	9

Fig: The conventions of positions followed

Please note that the inputs 0000,1101,1011,1110,1111,1010 will be considered as invalid.

The Basis of Computer Logic:

- **If X plays corner opening move, O should take center, and then an edge, forcing X to block in the next move irrespective of what is the second move of X.**
- **If X plays edge opening move, O should take center or one of the corners adjacent to X.**
- **If 2 X are in same column or same row or same diagonal, O goes in the position left in the column/row/ diagonal to prevent X from winning.**

The logic used in the ComputerTurn module is the result of cases corresponding to the above situations and the result is that

"The Computer will never let the Player win"

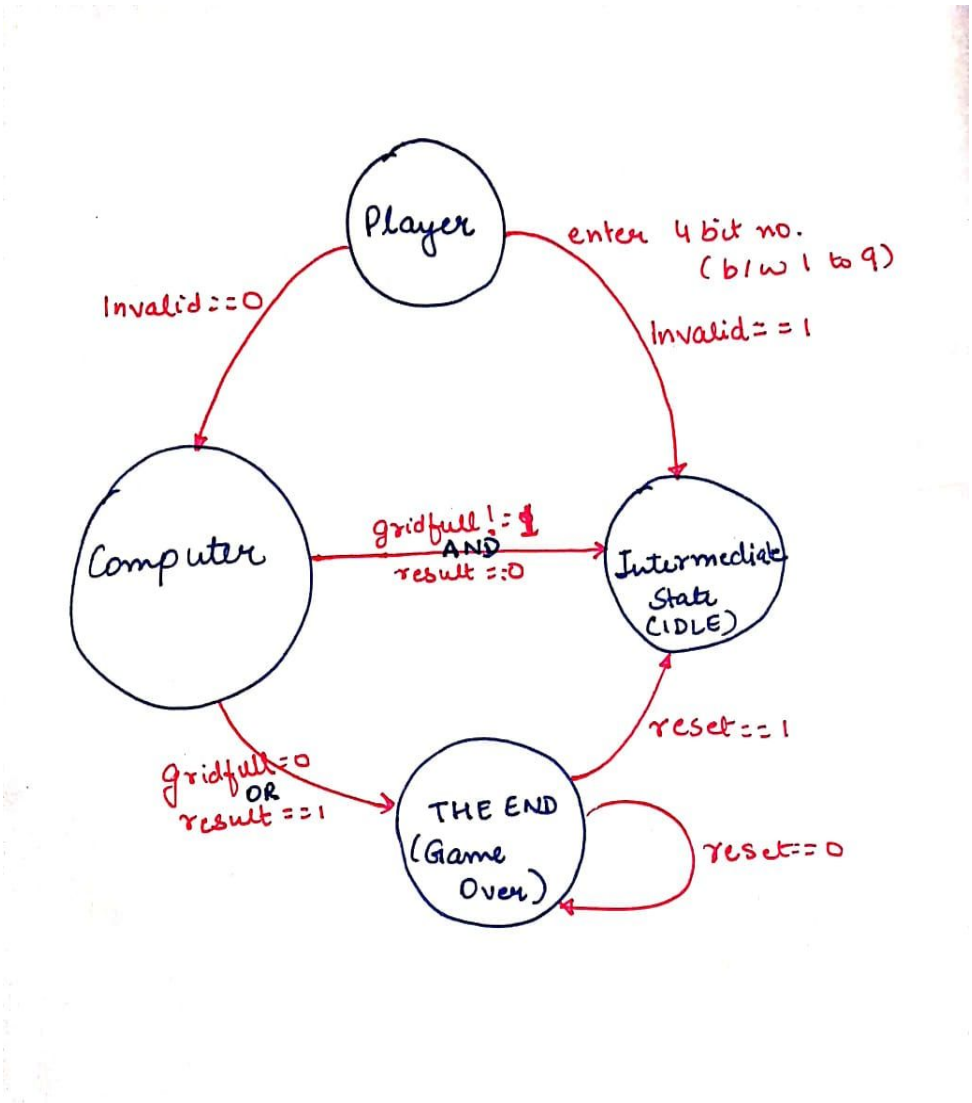
Its either a draw (all LEDs of both colours light up) or the Computer wins which is depicted by lighting up of all LEDs of Red Colour and the Second last LED of the FPGA.
The Last LED of FPGA will light up when the Player wins, represented by all green LEDs lighting up.

Modules Involved:

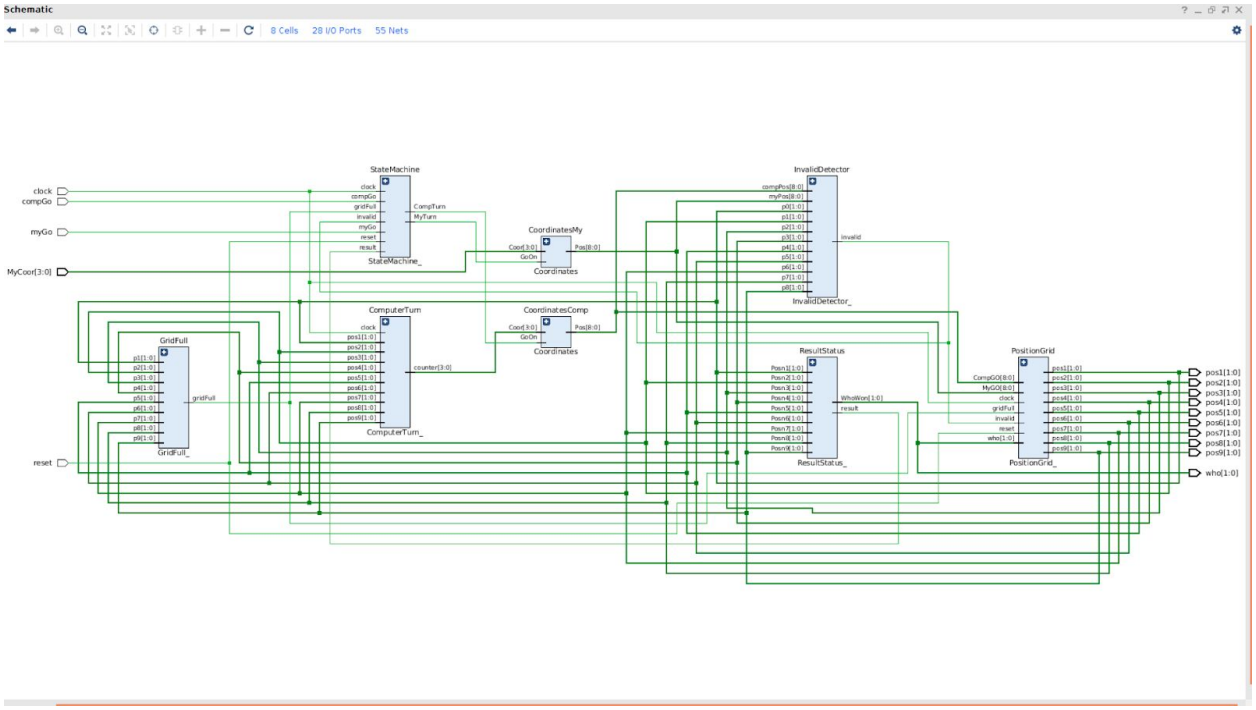
- **Coordinates** : Converts the 4 bit position entered to a single decimal digit
- **ComputerTurn_** : It decides the LED that has the lighted up as a response to the Player, based on logic mentioned above
- **GridFull_** : It shows when the Grid is full to declare a draw or winner
- **InvalidDetector_** : It renders the move invalid if the position entered is already occupied or does not exist
- **PositionGrid_** : It lights up the LED corresponding to the output of Coordinates
- **ResultStatus_** : It shows if someone has won the game

- **StateMachine_ :** Describes the Different States of the game, namely **INTERMEDIATE, COMPUTER, PLAYER and END_GAME**

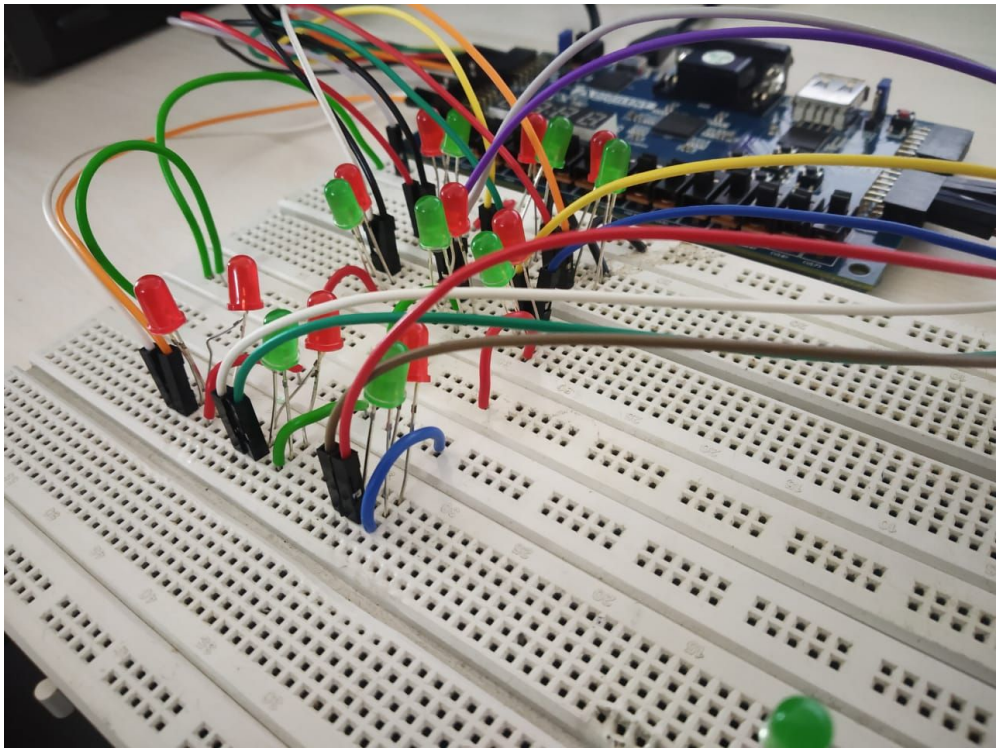
State Diagram:

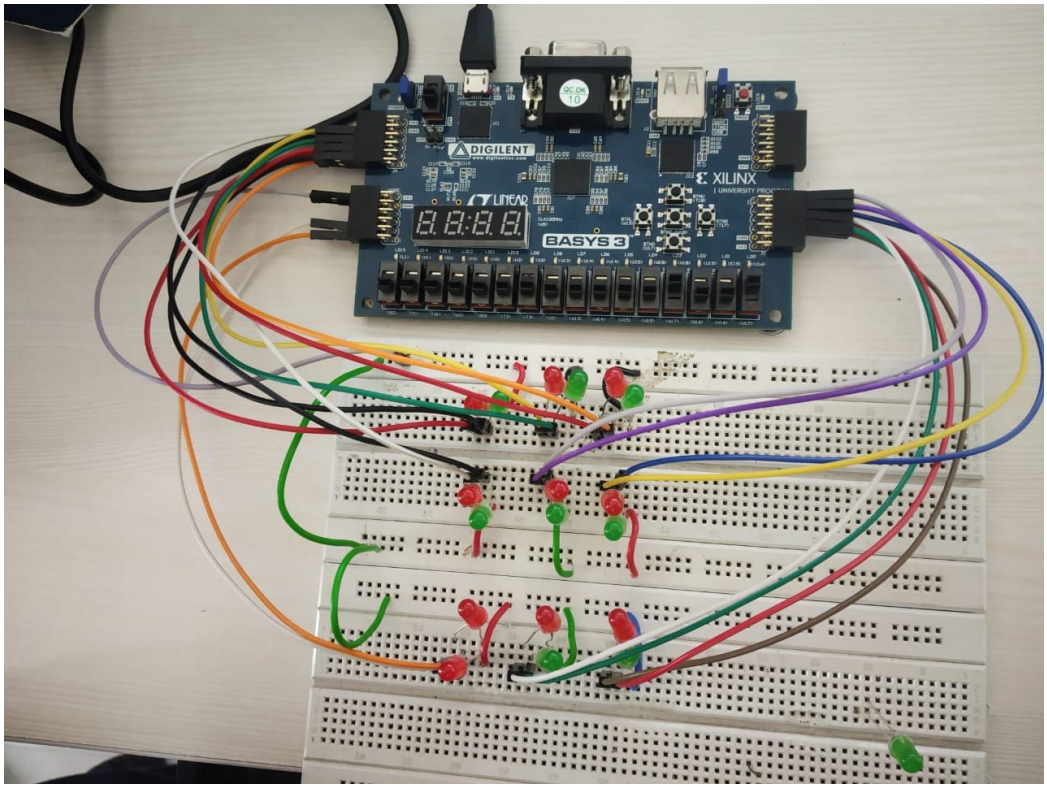


Elaborate Diagram:



Circuit :




DIGILENT
XILINX
BASYS 3