Tic-tac-toe

Mini project report

Name: Kshitij M. Ghodake (112101025) and Likith Anaparty(112101026)
Team: 2
Course no: EE2170

Introduction:

The classic game of tic-tac-toe was implemented on the breadboard using Zynq-7000 board and Verilog to code its implementational logic. The design was able to successfully register the moves of both players and detect who won the game by detecting the winning sequence which was hard-coded in the code. Four switches SW1, SW2, SW3, and SW4 of Zybo were used to access the respective block that needed to be modified and two push-buttons were used to define X or O on that respective block.

0001	0010	0011
0100	0101	0110
0111	1000	1001

To differentiate the move played by player one and the other player using only 9 LEDs, the concept of lighting-blinking was used i.e. if it is player one's move, the LED will constantly be ON otherwise for the second player the LED will constantly blink. This method halved the number of LEDs required to implement the project. There will be 2 LEDs that light up for the corresponding winner. The numbering started from 1 because 0 and any push-button will trigger the RESET for the design.

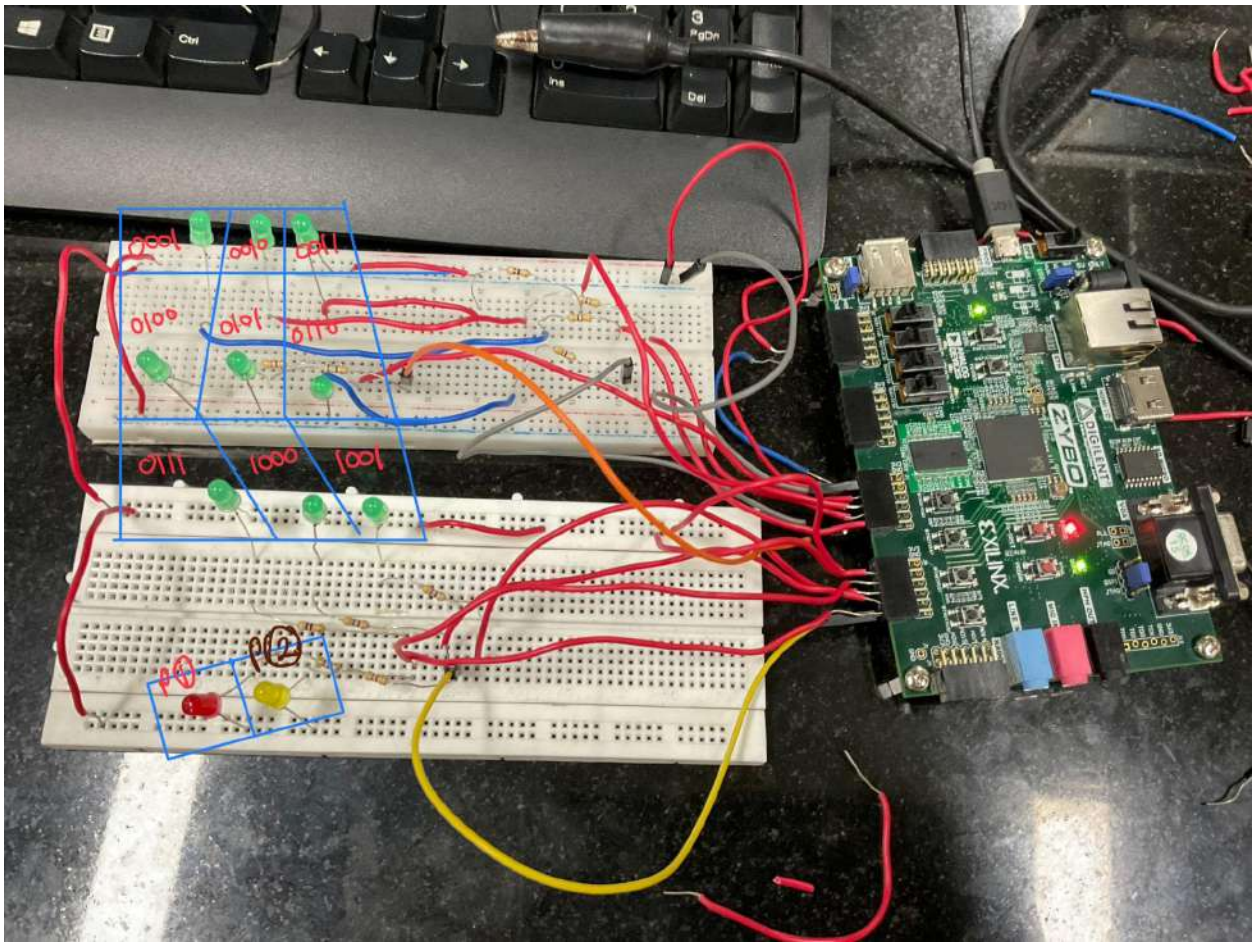
Materials required: Breadboard, Vivado, Zybo-7000, connecting wires, 9 LEDs for the grid and 2 LEDs for declaring the winner, and DC power supply.

Working mechanism:

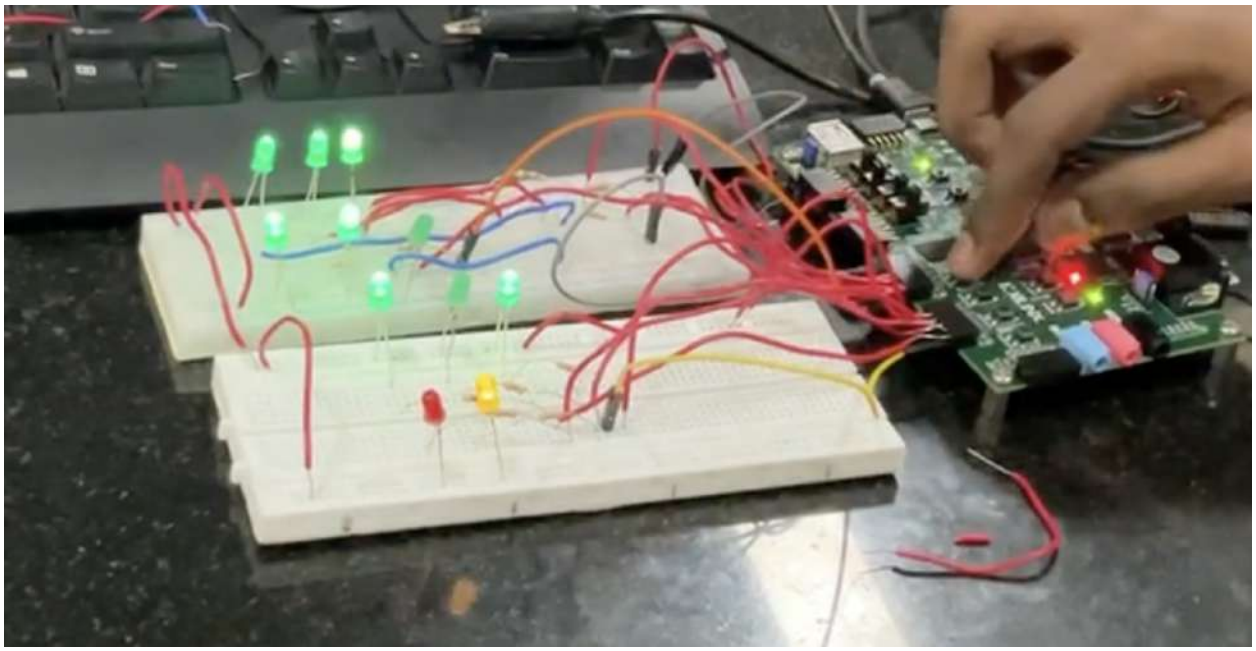
As tic-tac-toe has a 3x3 grid, we labeled each square as 1, 2, 3,... starting from left to right and the square that needed to be updated was given in binary using the 4 switches. E.g. if we want to mark X or O on the bottom-right square, an input of 1001 has to be given first and then the respective X or O. The entries are stored in an integer matrix which will be constantly detecting certain combinations that are the winning combinations, e.g. 1-2-3, 1-5-9,...(there are 8 of them) it will check if the winning combination is achieved by player 1 or 2, and then will light the corresponding LED for the winner. After the game is

over, the input of 0000 and of any push button should be given to reset the game. The Verilog code and the images/videos of the working design are given below.

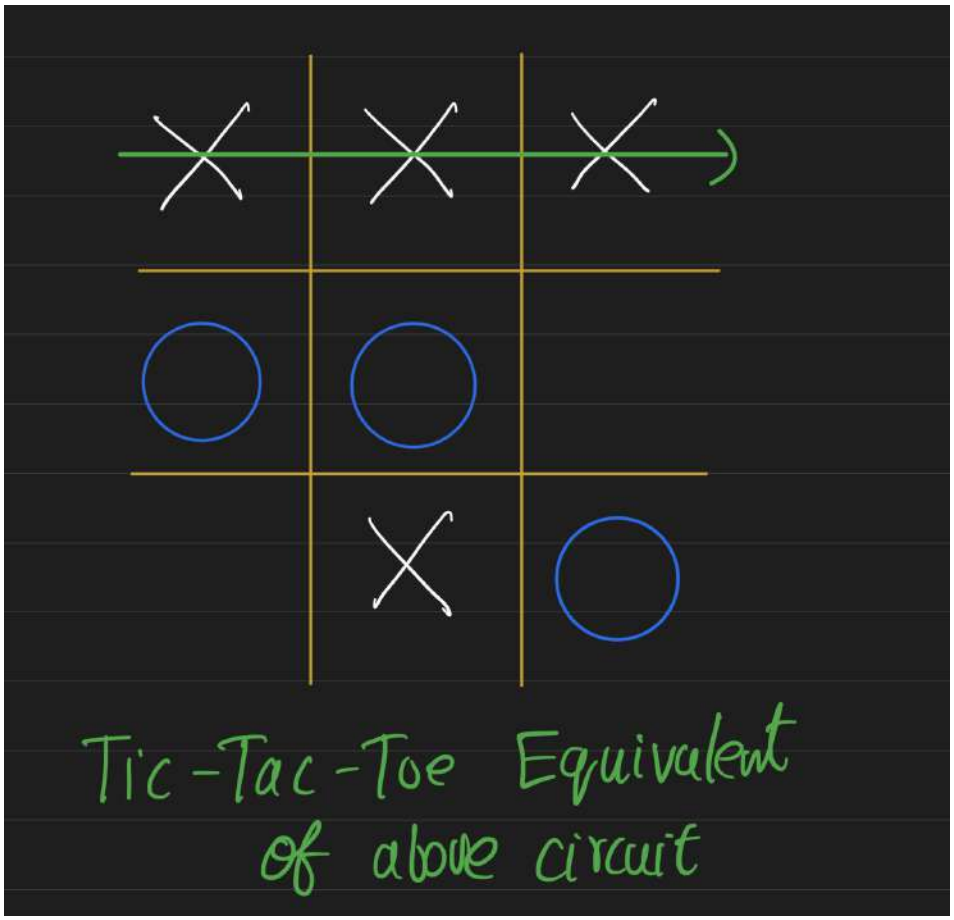
Working design:



Initial state of the design



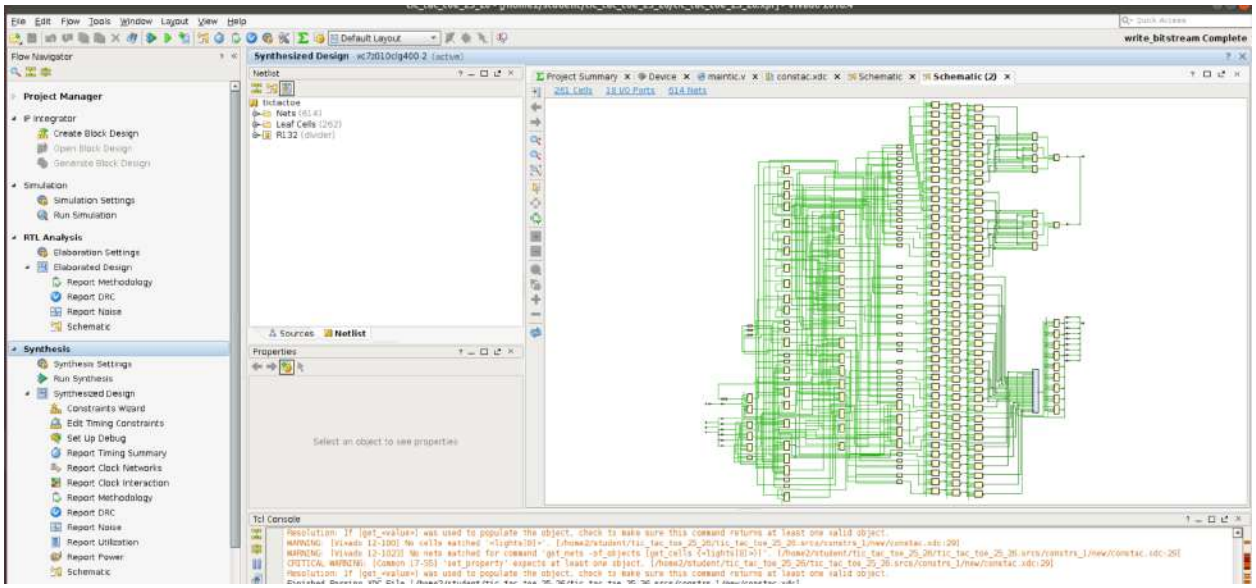
Working state of the game (the blinking and flashing are not visible in the image, here's the video link)



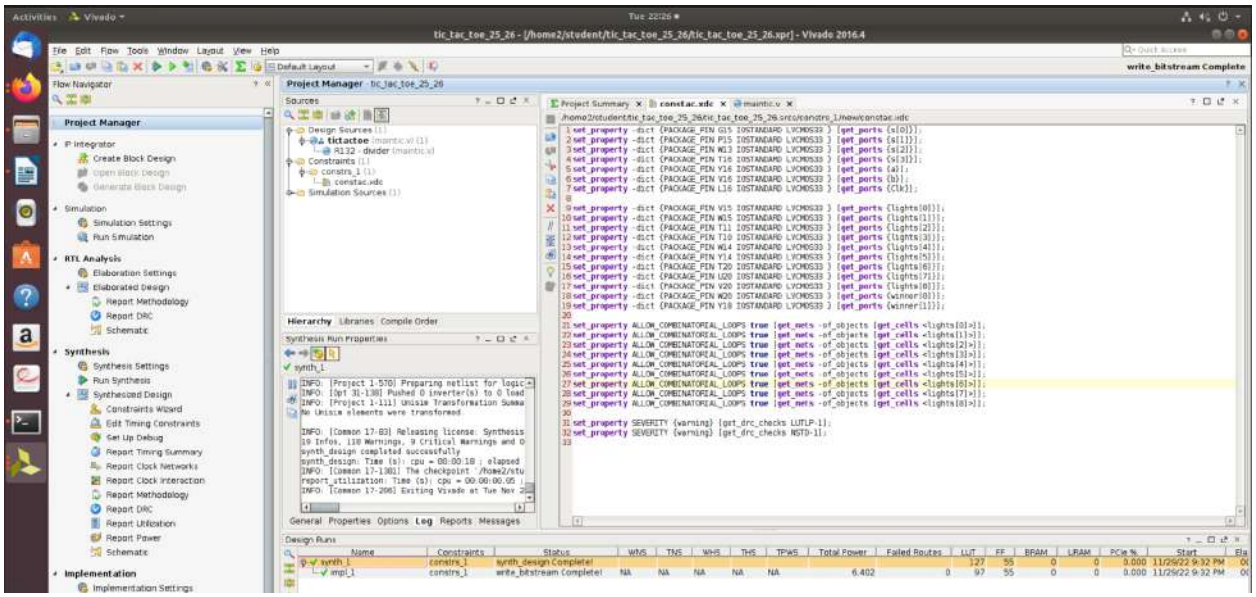
Explanation: Yellow LED lit up because player 2 won the game, the switches were used to access the block which had to be modified and the push buttons to plant the X and O's, in the end, the design detected the 1-2-3 winning sequence in the grid and hence the light lit up.

Verilog code explanation: Firstly, the frequency of the Zybo's clock is 1.25GHz. A 1-second clock was made using our Divider function. , it has a variable named Clk which negates the value of itself after 625000 (half the frequency of the Zybo's Clock) cycles of Zybo's Clock. After accessing the particular square. Inside the always block, we made a case statement such that whatever modulus of input is given via the switches. We can access the (Modulus of input) the input array value. If the player marks his symbol via the first push button. Then (Modulus of Switch input) the array value will become integer zero. Similarly, becomes an integer if it's the 2nd player. Depending on the value of that particular nth array term. We assigned the lights to either glow or blink or be turned off. Two reg outputs (one for each player) one for each player, and all the winning conditions were hard-core in the design. Whenever the Verilog detects a winning sequence. This output reg value of that particular player changes from 0 to 1 and the winning light glows. Lastly, In 0000 input mode. If we give a push button input. Then all the values of the integer array will be re-assigned to the value 2, again, i.e. resetting the game

RTL design/schematic:



Constraints file:



Conclusion: The theory of sequential circuit designs, and the implementation of them using Verilog by using parallel execution of Verilog by using the minimal number of hardware was carried out. This project made our understanding of the subject clear and gave us the ability to appreciate the world of circuits more.