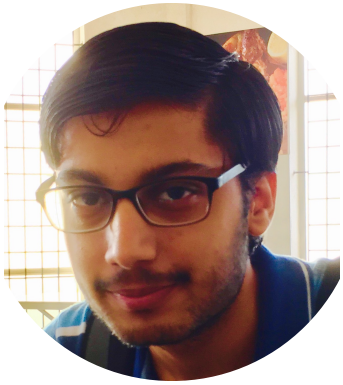
Tic Tac Toe

Meet the TEAM



Satvik Choudhary
(111601021)

Contribution

1. Implemented the AI functionality using C++ code.
2. Helped in adding to the content of the report.

Vishal Choudhari
(121601027)

Contribution

1. Designed and minimised the breadboard circuit.
2. Wrote the minimum viable Verilog code.
3. Made the abstract.
4. Designed the final report.

Sourabh Aggarwal
(111601025)

Contribution

1. Wrote the Test Bench.
2. Helped in writing the Verilog code.

Table of Contents

Tic Tac Toe (Game)

AIM:

1. To implement a Tic Tac Toe game machine with computer mode.

MATERIALS REQUIRED:

- Breadboard
- LEDs (green and red - 10 each)
- Connecting wires
- 330Ω resistors (20 numbers)
- Xilinx Zybo - 7000 SoC Trainer Board.

SOFTWARE REQUIRED:

Xilinx Vivado Design Suite.

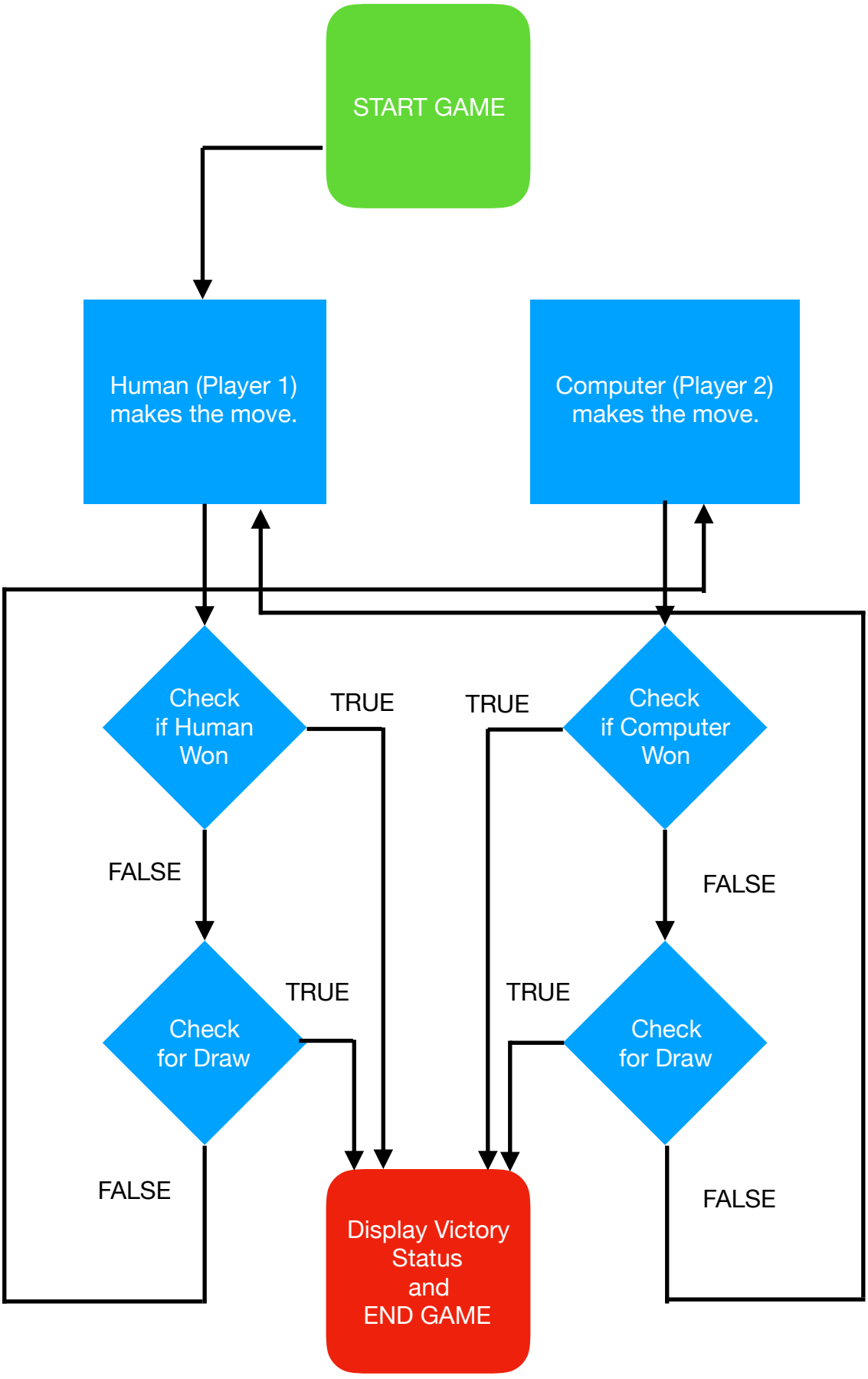
ABSTRACT

Tic Tac Toe:

Tic Tac Toe is a two player paper and pencil game. It involves a 3 x 3 grid, in which the each of the player marks either an X or O, taking alternate turns. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row wins the game. Players soon discover that best play from both parties leads to a draw.

Because of the simplicity of tic-tac-toe, it is often used as a pedagogical tool for teaching the concepts of good sportsmanship and the branch of artificial intelligence that deals with the searching of game trees.

BLOCK DIAGRAM



STATE DIAGRAM

Due to the sheer complexity of the game, it is impossible to specify all the states of the game on a sheet of paper. However, to give an idea of the complexity, a demo state diagram for a specific gameplay has been included.

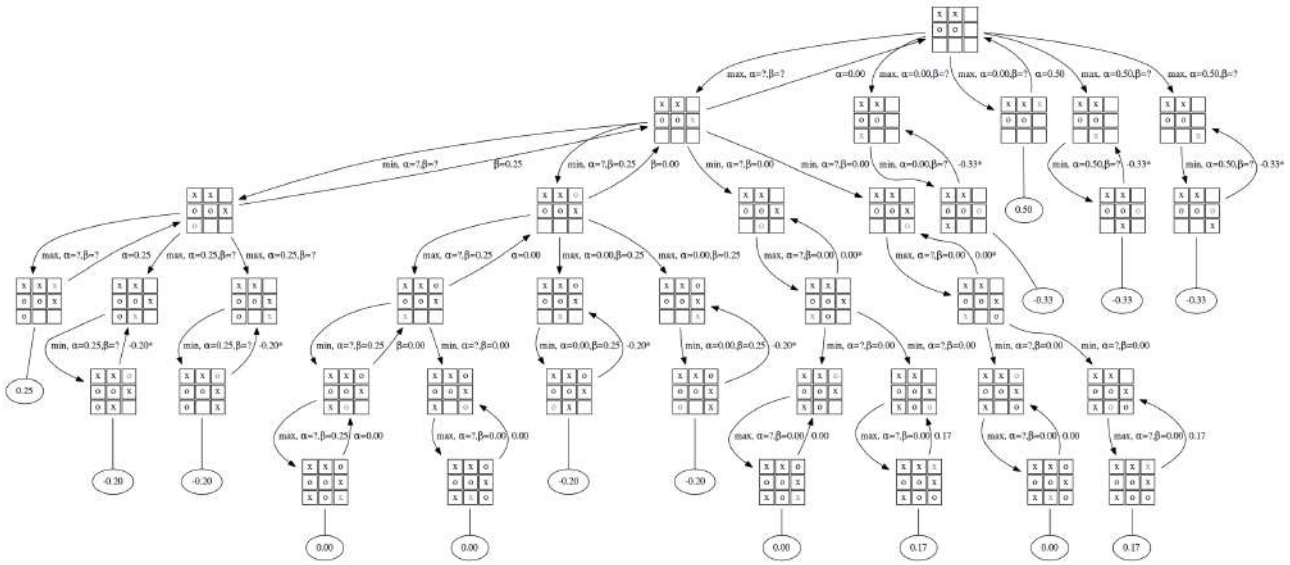
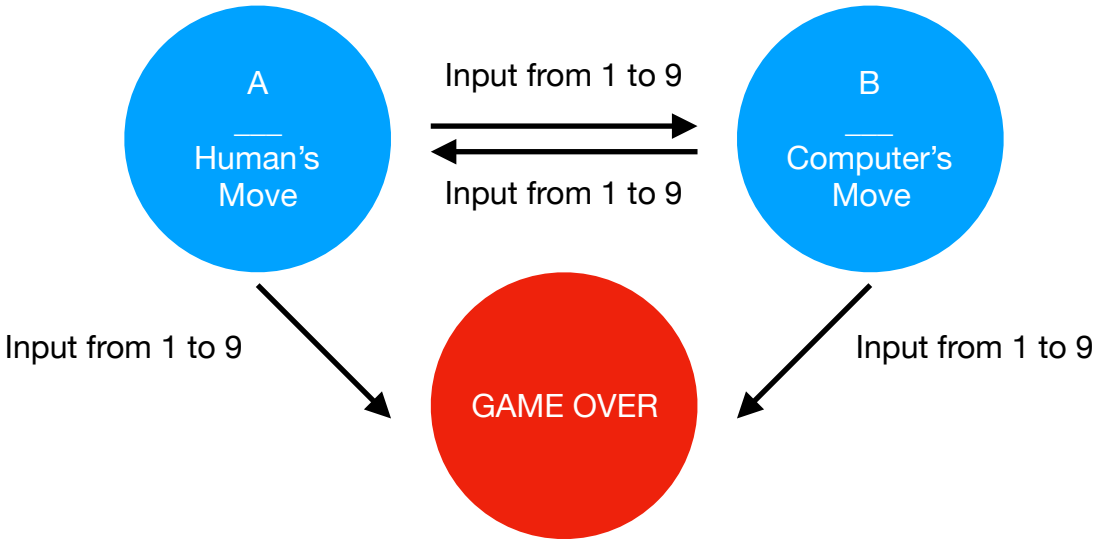


Fig 1: **Complex State Diagram of the Tic Tac Toe Game.**

A non-standard state diagram can be constructed as follows:



TIMING DIAGRAM

Human's Moves (in red ovals)

Computer's Moves (in green ovals)

Winner Identified!

It can be observed that the human and the computer play the moves alternatively, until there is a winner or the game is drawn.

Implementing Tic Tac Toe (with computer mode) using FPGA

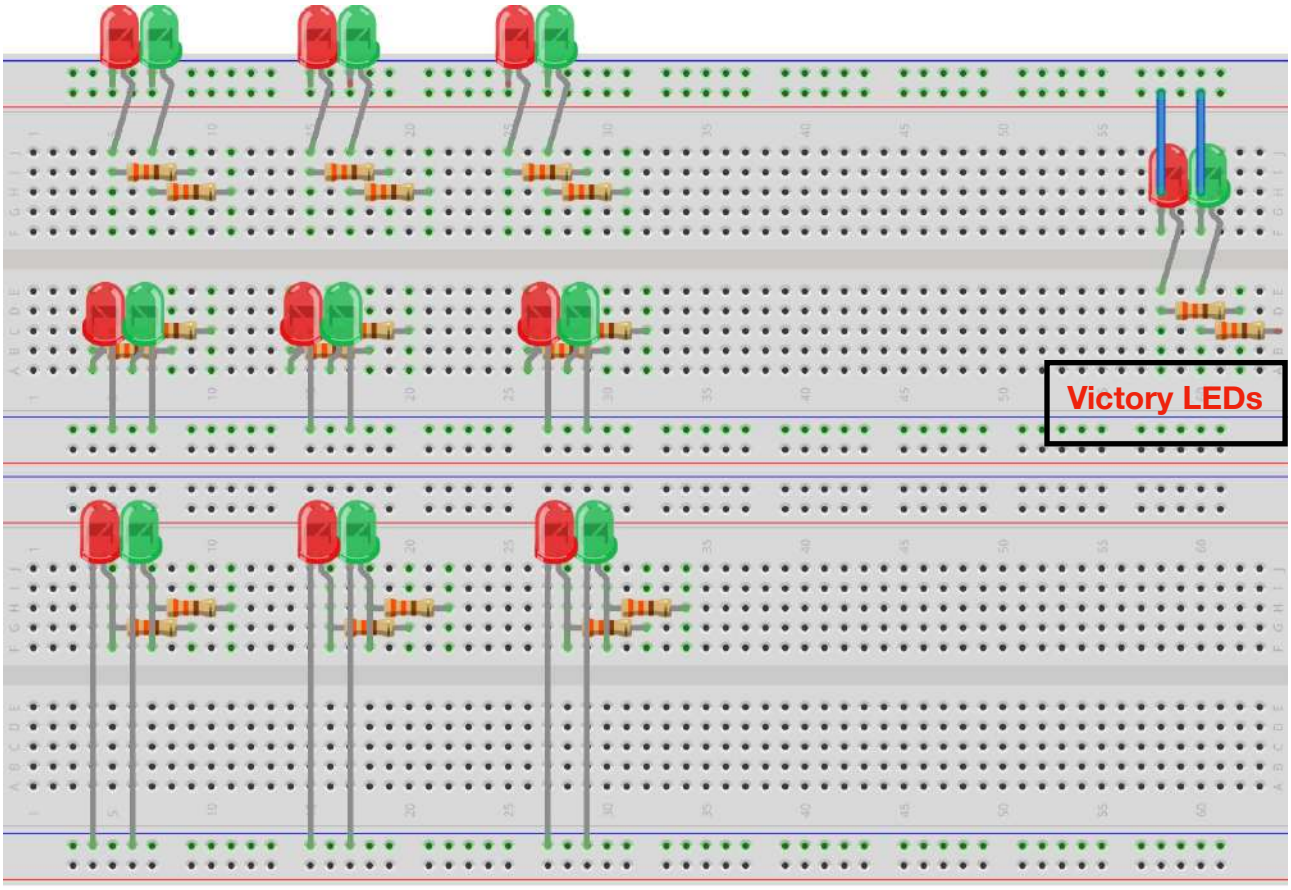


Fig 2: **Implementation of the game on the breadboard.**

Let us assign each of the players a colour. Say:

- Player 1 - Red
- Player 2 (computer) - Green

On the breadboard there will be **20 LEDs**:

- Grid LEDs

    1. 9 Red LEDs for indicating Player 1's marks.
    2. 9 Green LEDs for indicating Player 2's (computer's) marks.

- Winner Display LEDs

    1. 1 Red LED to denote Player 1's Victory.
    2. 1 Green LED to denote Player 2's (computer's) Victory.
    3. When both LEDs turn on, it indicates a draw.

Rules to Play the game on FPGA:

1. When the FPGA boots up, the Red player can make the first move.

2. To make a move, the player marks his position by first choosing it through the switches (in BCD) and then confirms it by pressing the first enter button.

3. Once the enter button is pressed, the Red LED corresponding to the marked position lights up. Its now the Green player's (computer's) turn to play.

4. The Green player's (computer's) move is activated pressing the second enter button.

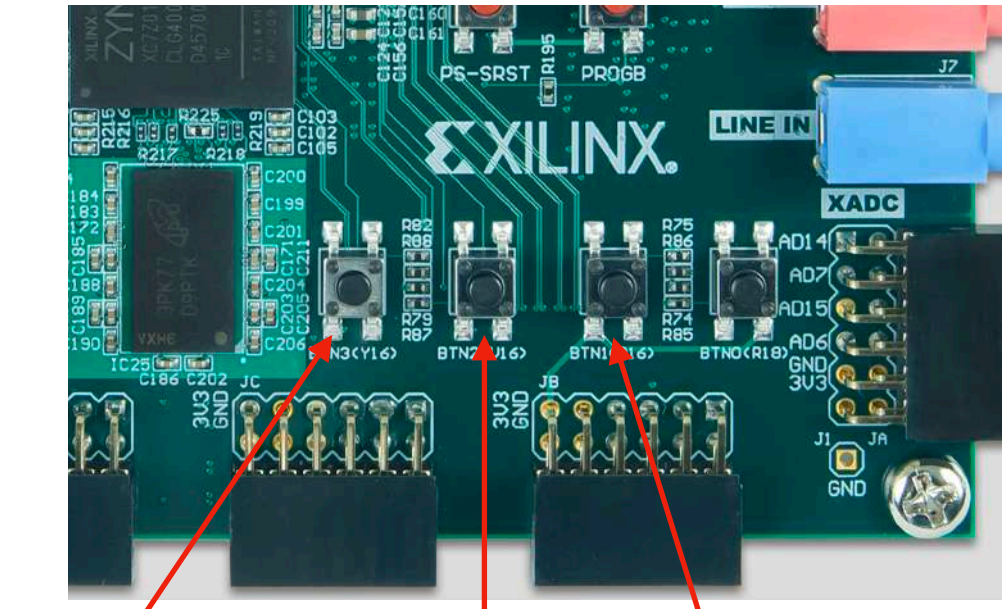
5. The players take alternative turns until one of them emerges as a winner or when the game ends in a draw.

6. If there is a winner, the victory LED corresponding to the player's colour lights up. If there is a draw, both the victory LEDs light up.

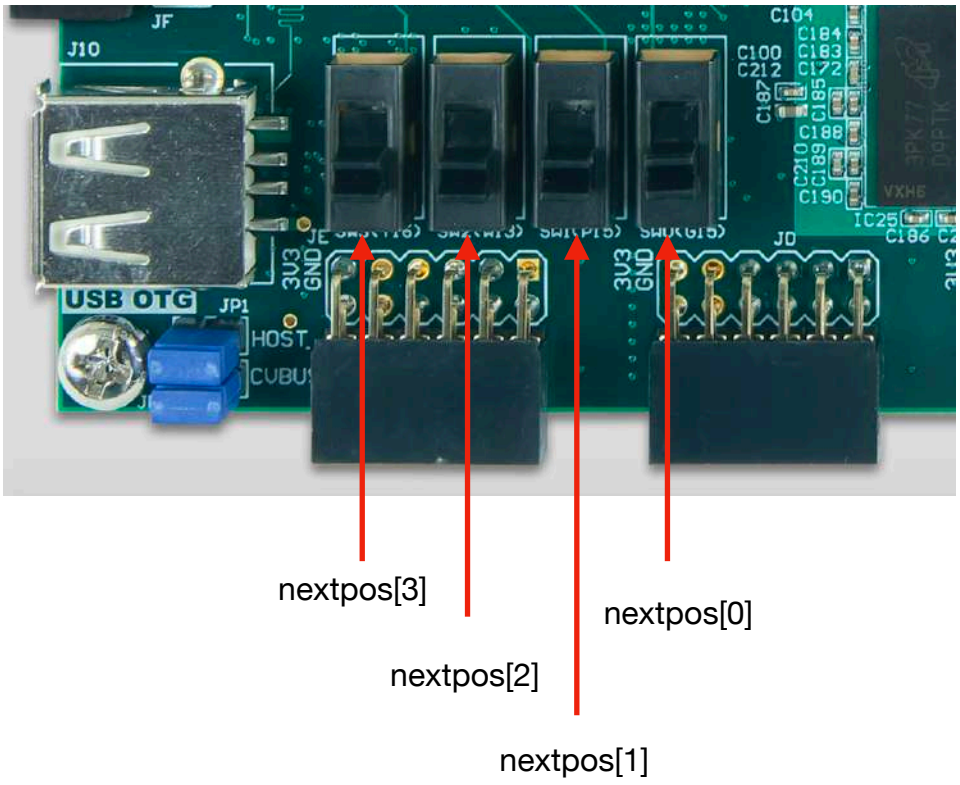
1 (0001)	2 (0010)	3 (0011)
4 (0100)	5 (0101)	6 (0110)
7 (0111)	8 (1000)	9 (1001)

Fig 3: **Chart indicating the BCD values corresponding to the squares.**

Implementing **Reset** and **Enter** Functionality:



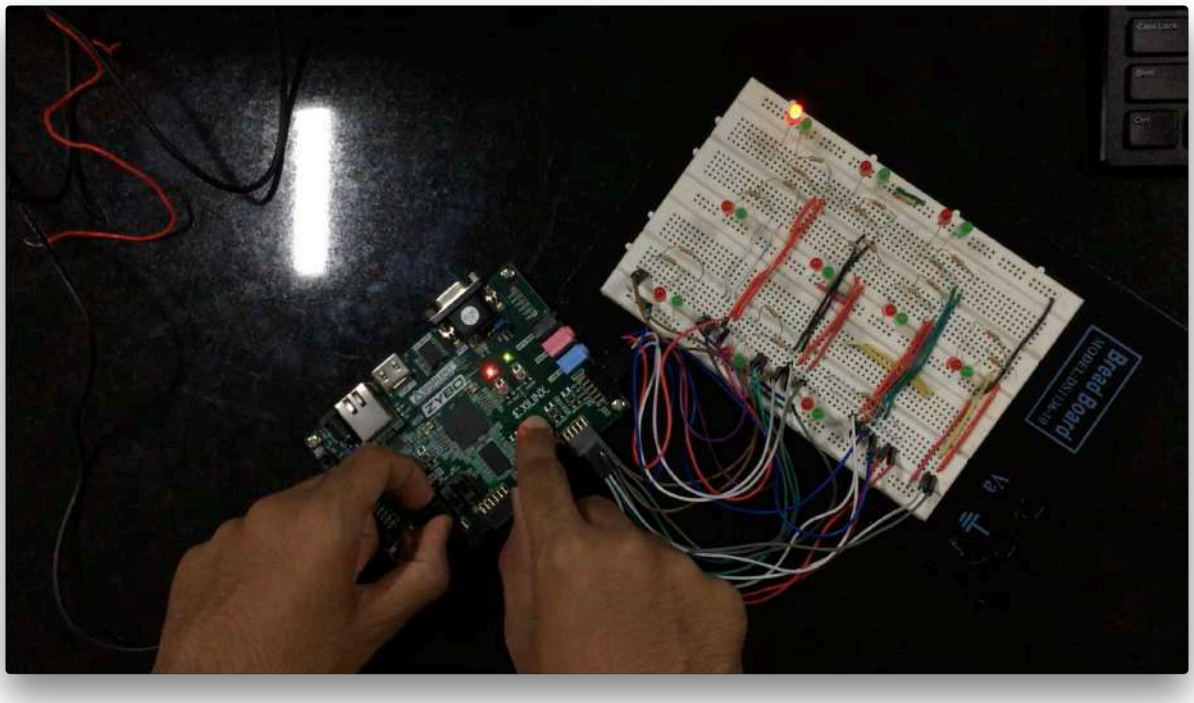
Inputs for the player to **choose** his **next move**:



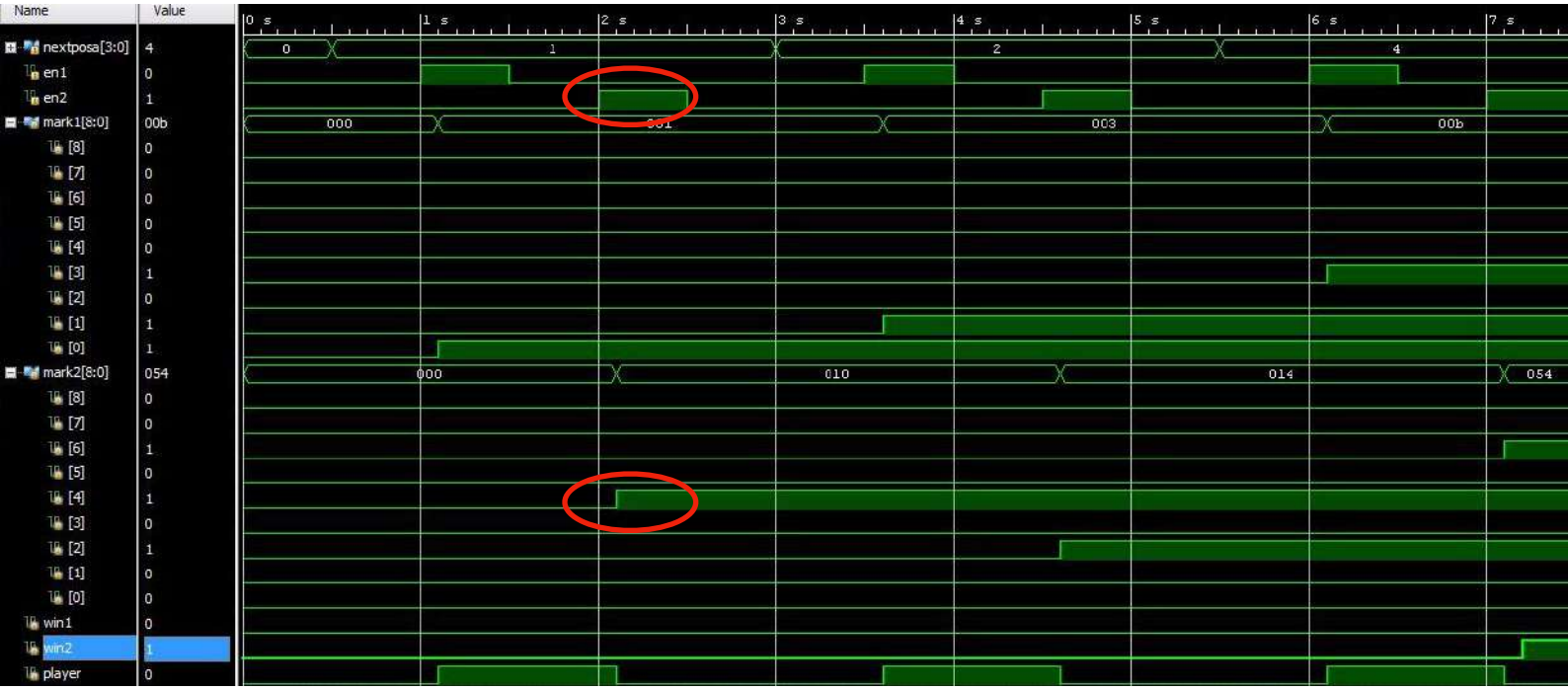
SIMULATIONS

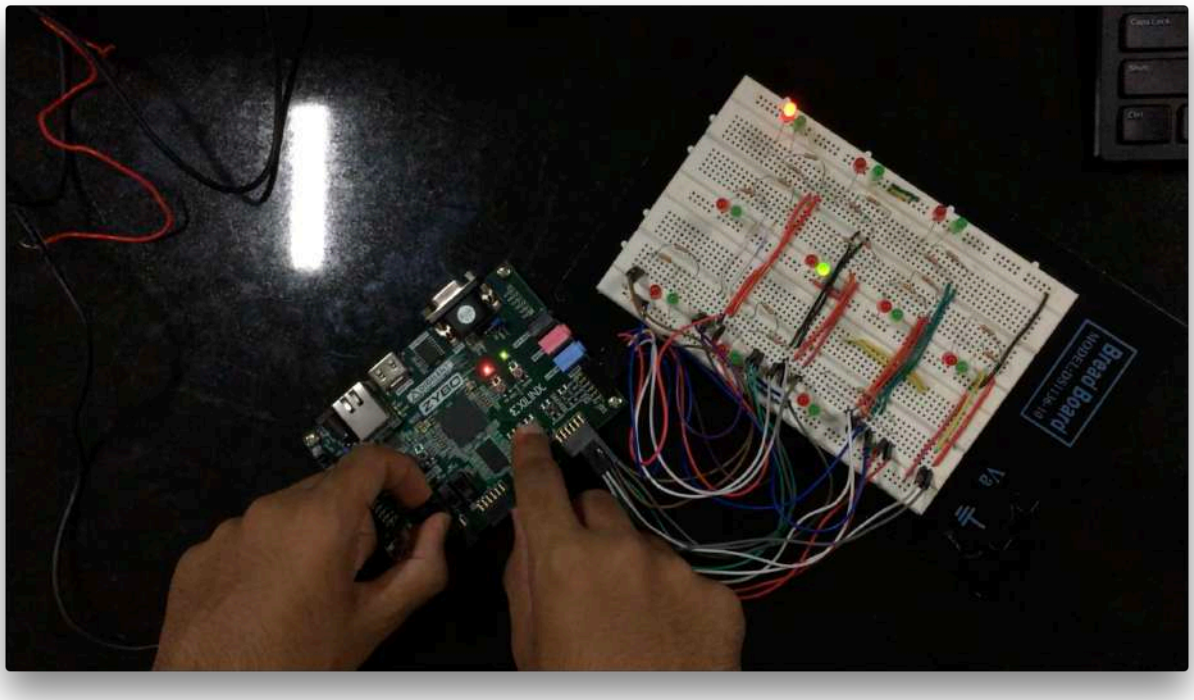
Simulation 1.) Human Loses, Computer Wins:

Move 1 A.) Human makes a mark on the 1st square.

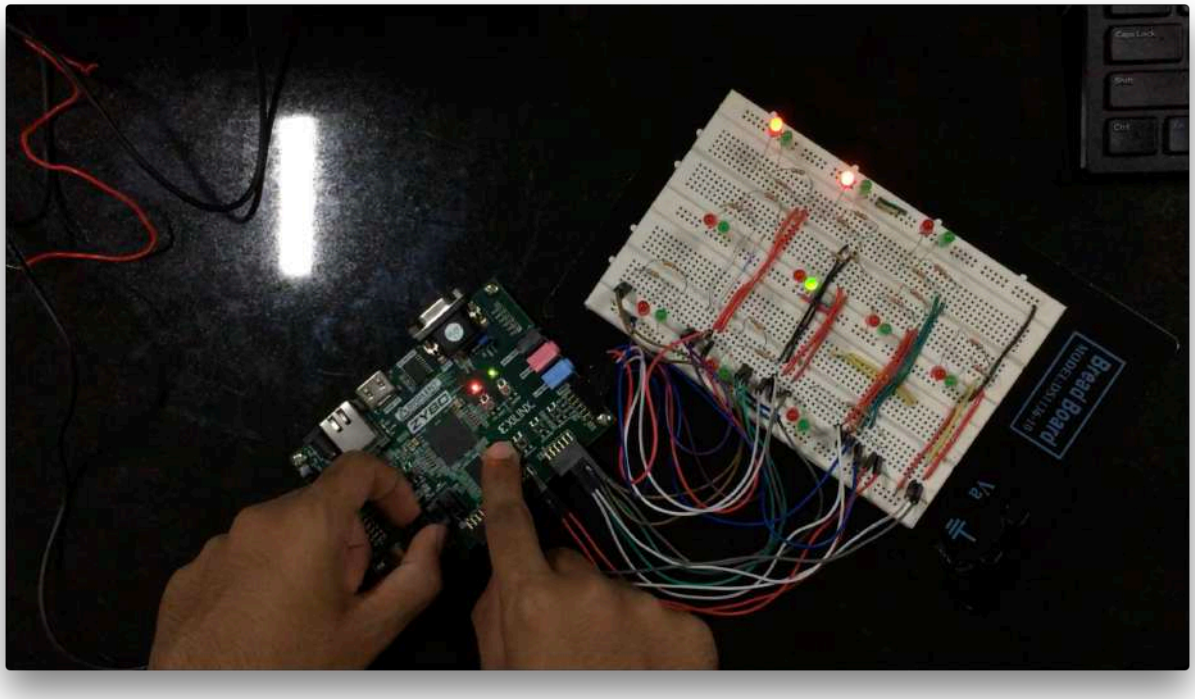


Move 1 B.) Computer makes a mark on the 5th square.

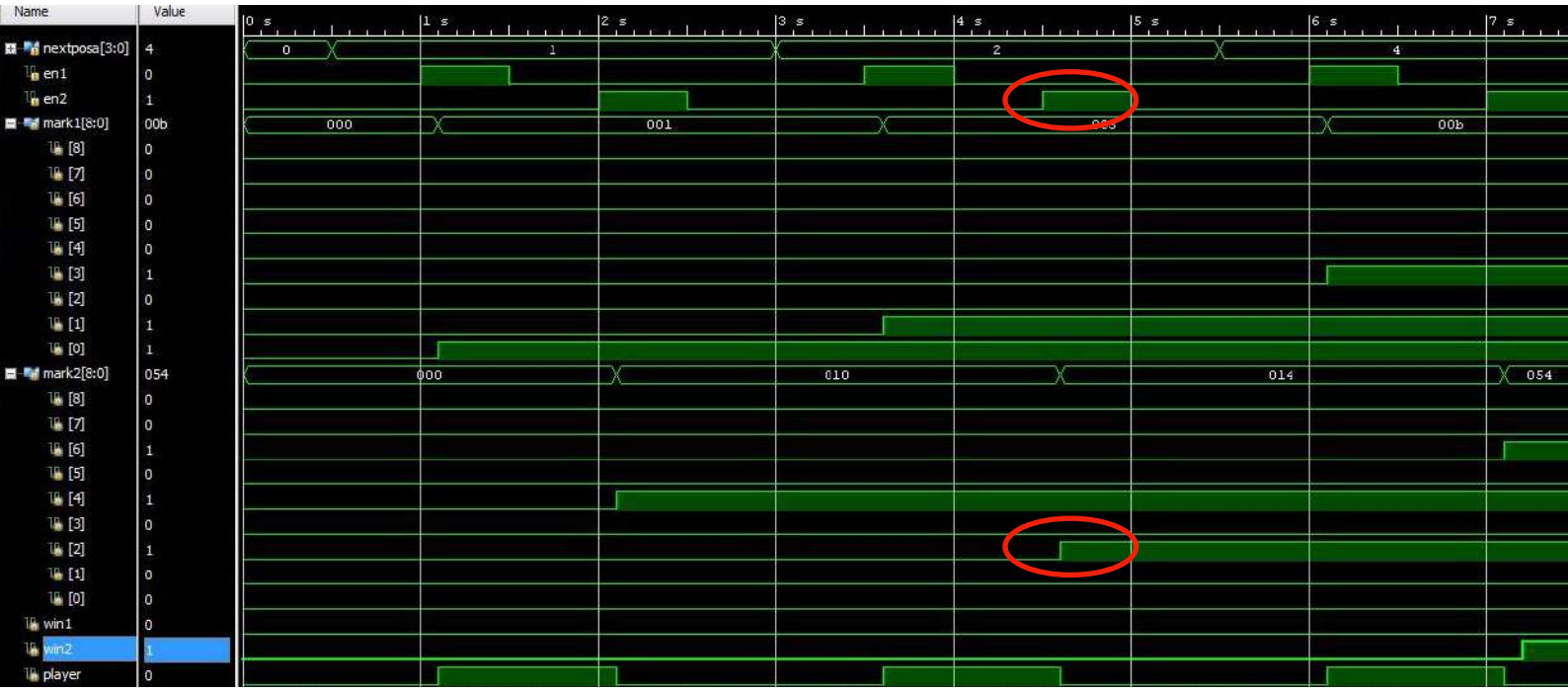


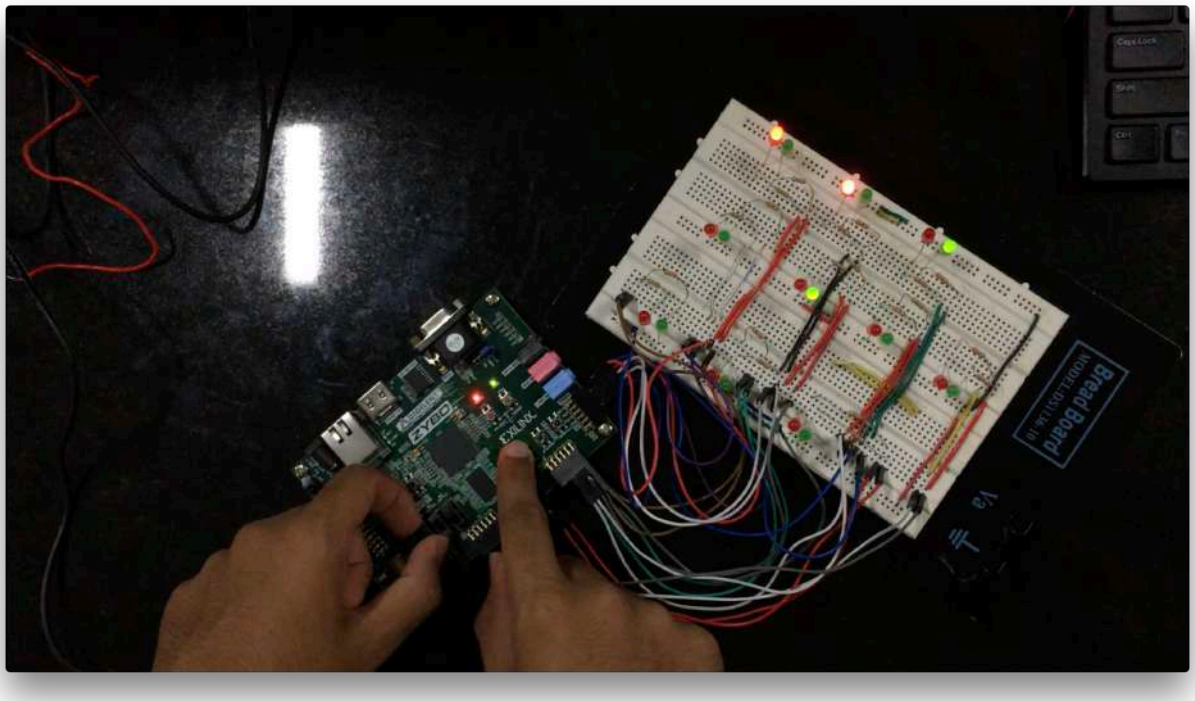


Move 2 A.) Human makes a mark on the 2nd square.

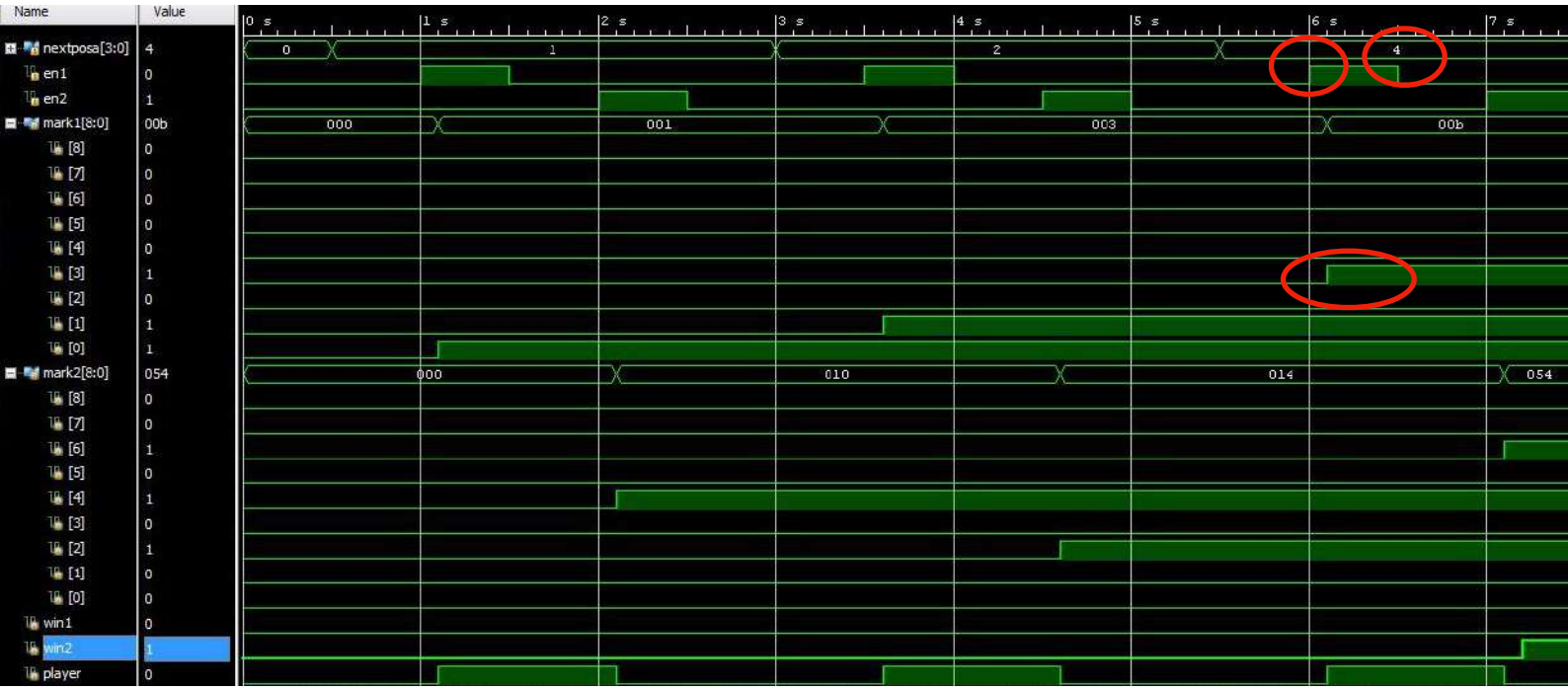


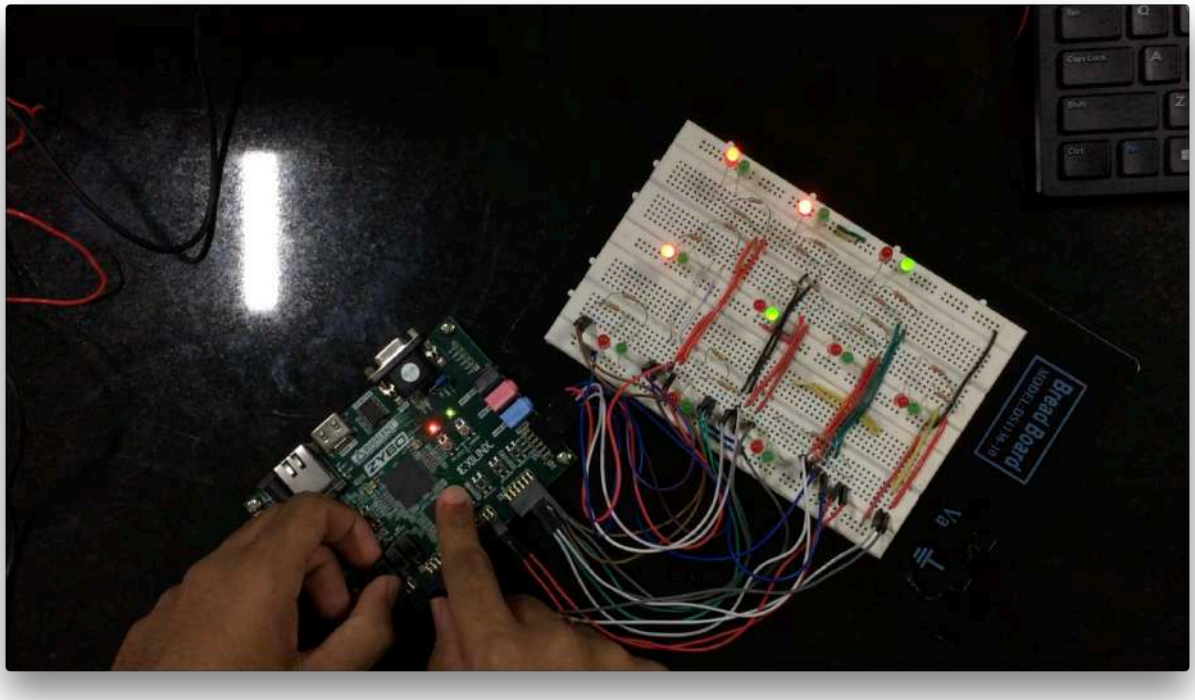
Move 2 B.) Computer makes a mark on the 3rd square.



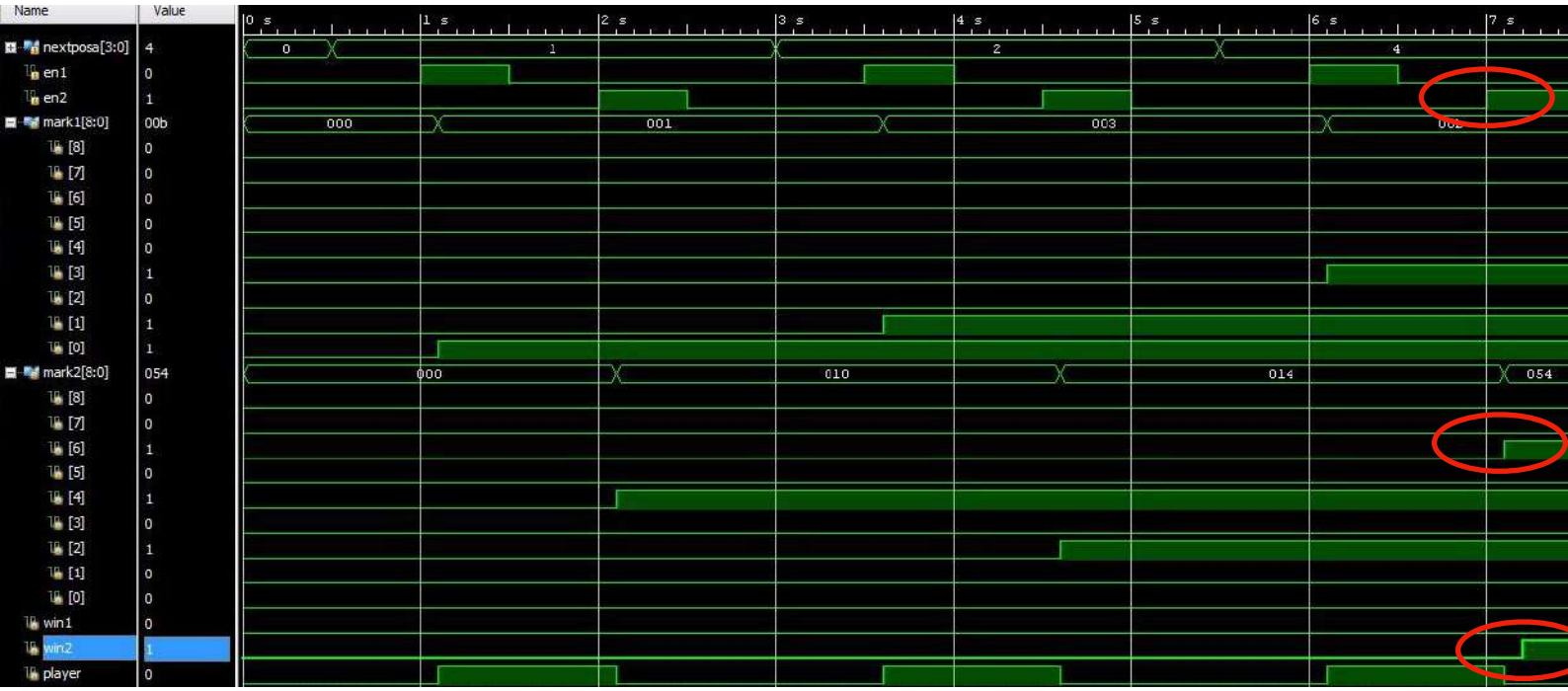


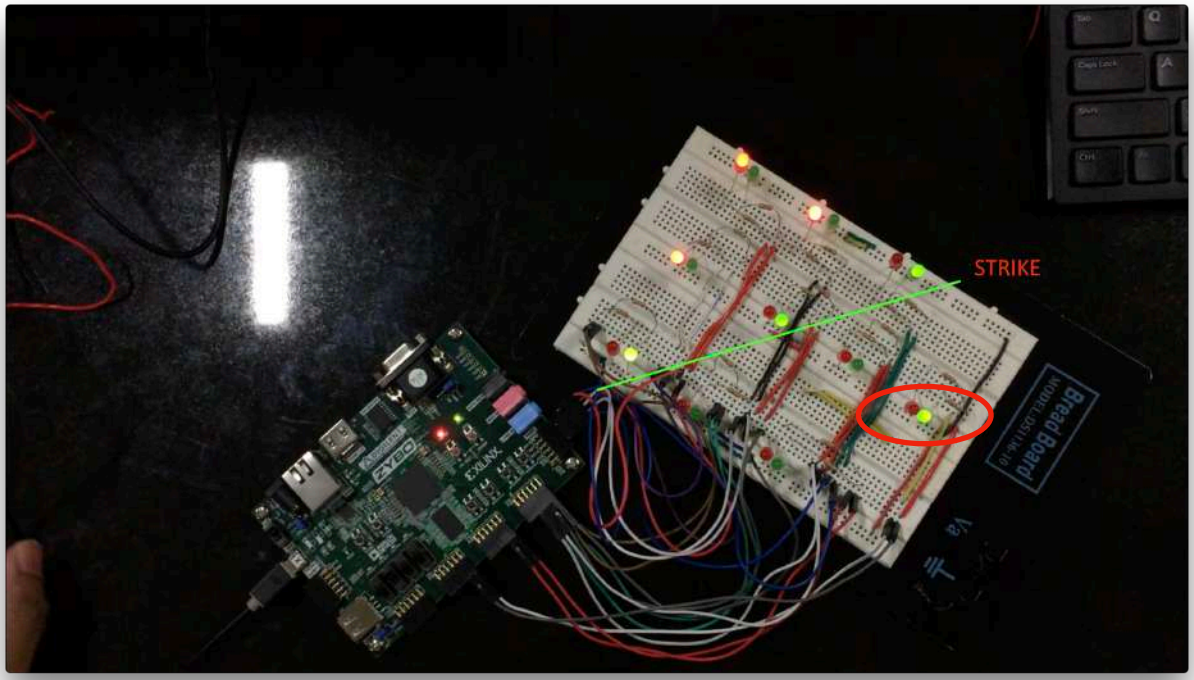
Move 3 A.) Human makes a mark on the 4th square.





Move 3 B.) Computer makes a mark on the 7th square and Wins!





Green Victory LED glowing (highlighted) indicates the victory of the computer.

RTL SCHEMATIC

1.) For 2 Player Mode:

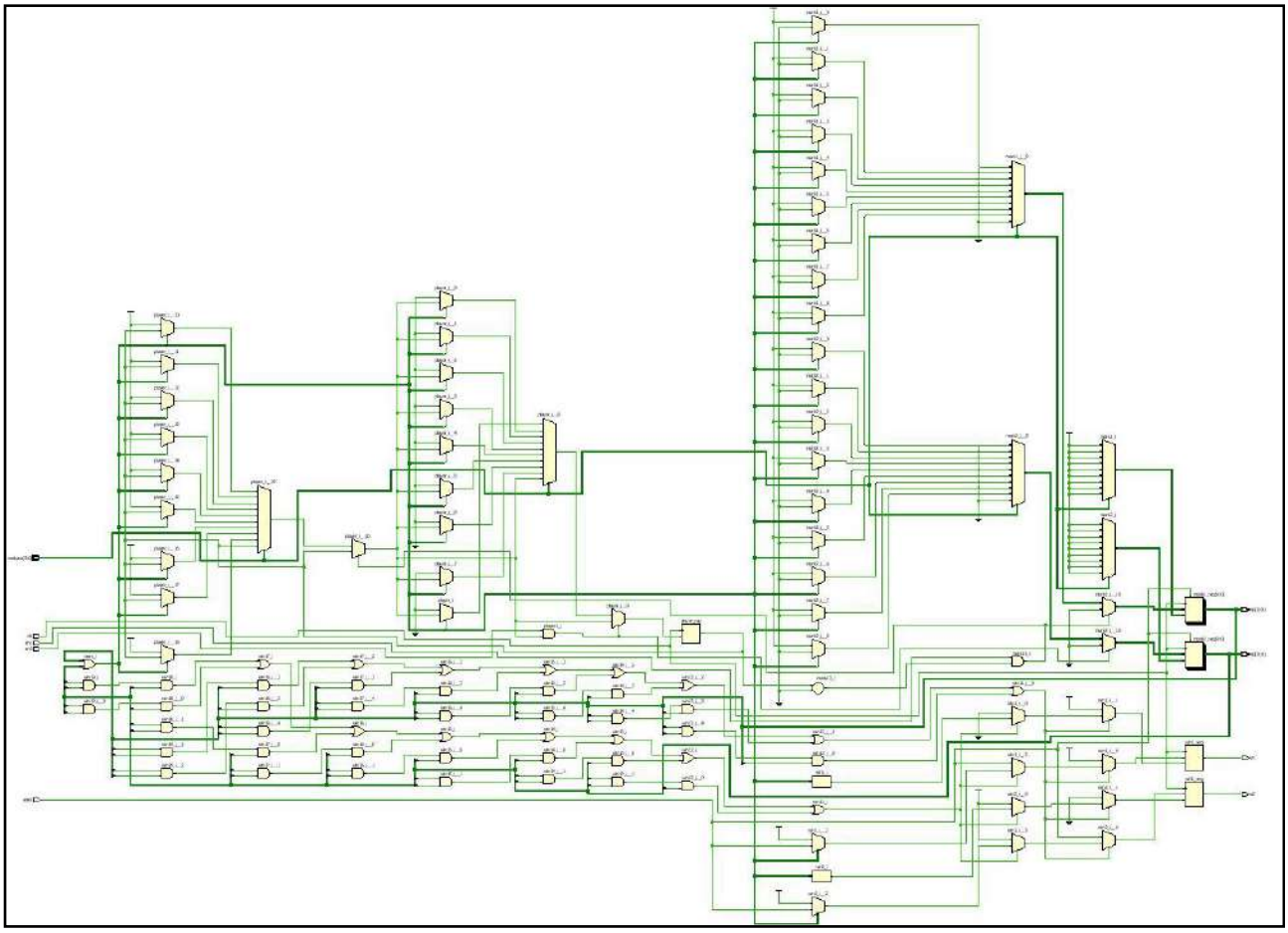
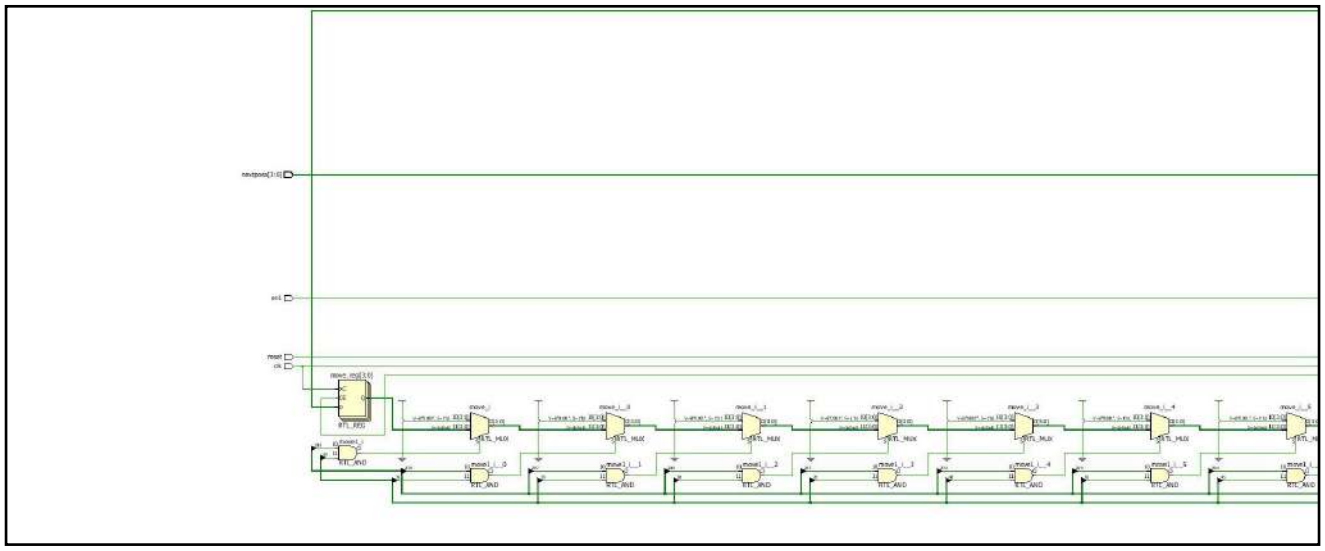
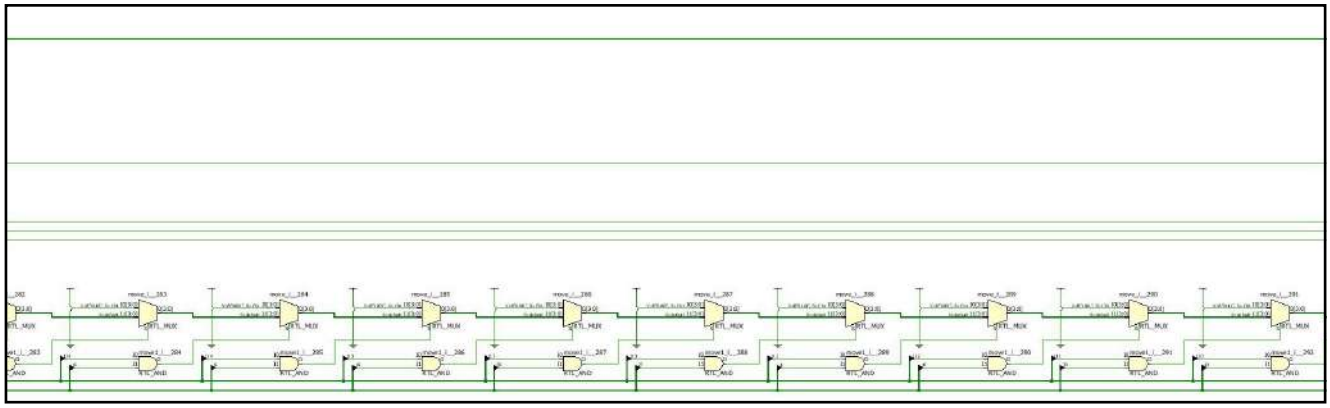
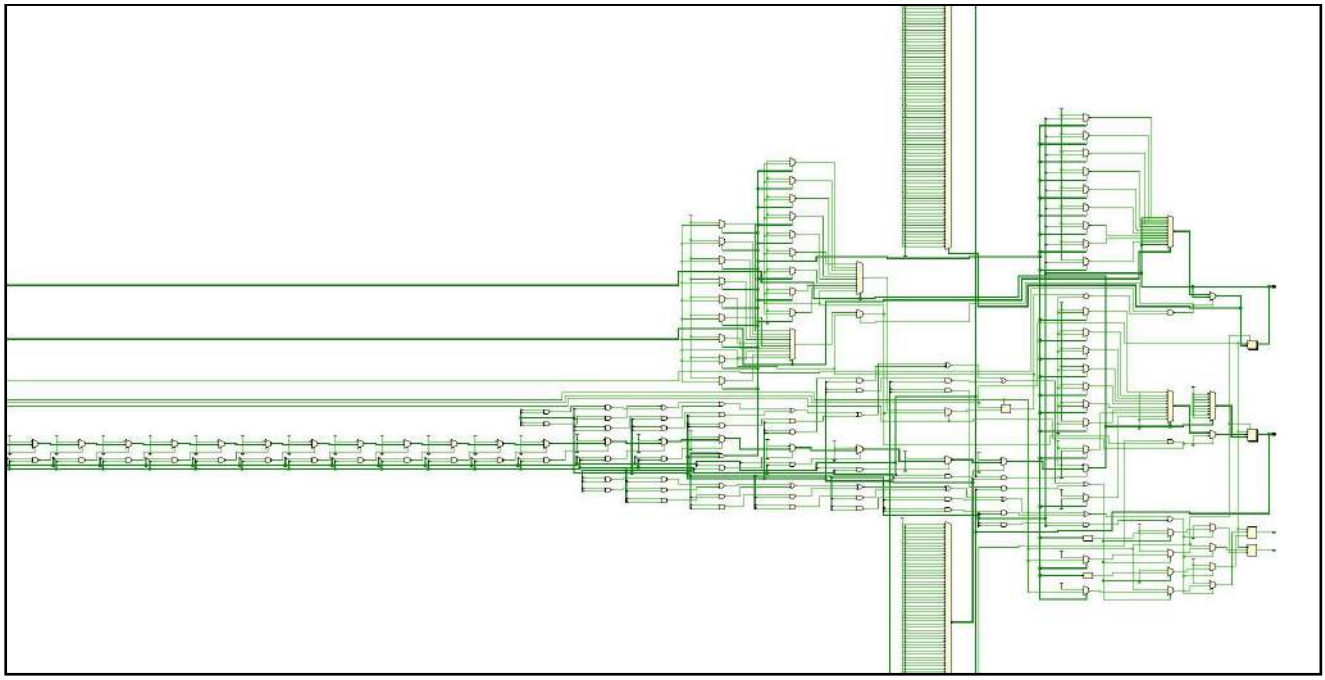


Fig. 4: **RTL Schematic for TWO PLAYER mode.**

2.) For Human v/s Computer Mode:







Due to heavy implementation of AI (Artificial Intelligence), we can see a lengthy cascaded chain of MUXes in the second diagram.

VERILOG CODE:

```
module tictactoe(
    input [3:0] nextposa,
    input en1,
    input en2,
    input clk,                          // Declaring the input and output ports
    input reset,
    output [8:0] m1,m2,
    output w1,w2
    );

    reg [3:0] nextpos;
    reg [3:0] move;
    reg win1;
    reg win2;                           // Declaring the registers and wires
    reg [8:0] mark1;
    reg [8:0] mark2;
    reg player;
    wire [8:0] mark;

    assign mark = mark1 | mark2;
    assign m1 = mark1;
    assign m2 = mark2;                  // assigning the output ports to registers
    assign w1 = win1;
    assign w2 = win2;

    initial
    begin
    move <= 4'b0;
    nextpos <= 4'b0;
    mark1 <= 9'b0;
    win1 <= 1'b0;                       // setting up initial values when the FPGA boots
    win2 <= 1'b0;
    mark2 <= 9'b0;
    player <= 0;
    end
```

```
always @(posedge clk)
begin
   if (en1 == 1 & en2 == 0 & player == 0)
   begin
      nextpos = nextposa;
      case (nextpos)
         4'b0001: if (mark[0]==0) begin mark1[0]=1; player=1; end
         4'b0010: if (mark[1]==0) begin mark1[1]=1; player=1; end
         4'b0011: if (mark[2]==0) begin mark1[2]=1; player=1; end

         4'b0100: if (mark[3]==0) begin mark1[3]=1; player=1; end
         4'b0101: if (mark[4]==0) begin mark1[4]=1; player=1; end     // HUMAN's moves
         4'b0110: if (mark[5]==0) begin mark1[5]=1; player=1; end

         4'b0111: if (mark[6]==0) begin mark1[6]=1; player=1; end
         4'b1000: if (mark[7]==0) begin mark1[7]=1; player=1; end
         4'b1001: if (mark[8]==0) begin mark1[8]=1; player=1; end
      endcase
   end

   if (en2 == 1 & en1 == 0 & player == 1)
   begin
        if(mark1 == 1 && mark2 == 0) move = 5;
   else if(mark1 == 2 && mark2 == 0) move = 1;
   else if(mark1 == 3 && mark2 == 16) move = 3;
   else if(mark1 == 4 && mark2 == 0) move = 5;
   else if(mark1 == 5 && mark2 == 16) move = 2;
   else if(mark1 == 6 && mark2 == 1) move = 4;
   else if(mark1 == 6 && mark2 == 16) move = 1;
   else if(mark1 == 8 && mark2 == 0) move = 1;
   else if(mark1 == 9 && mark2 == 16) move = 7;
   else if(mark1 == 10 && mark2 == 1) move = 5;
   else if(mark1 == 11 && mark2 == 20) move = 7;            // Computer's Move: AI
   else if(mark1 == 11 && mark2 == 80) move = 3;
   else if(mark1 == 12 && mark2 == 1) move = 5;
   else if(mark1 == 12 && mark2 == 16) move = 1;
   else if(mark1 == 13 && mark2 == 18) move = 8;
   else if(mark1 == 13 && mark2 == 80) move = 2;
   else if(mark1 == 14 && mark2 == 17) move = 9;
   else if(mark1 == 16 && mark2 == 0) move = 1;
   else if(mark1 == 18 && mark2 == 1) move = 8;
   else if(mark1 == 20 && mark2 == 1) move = 7;
```

```
else if(mark1 == 22 && mark2 == 9) move = 7;
else if(mark1 == 22 && mark2 == 65) move = 4;
else if(mark1 == 22 && mark2 == 129) move = 7;
else if(mark1 == 24 && mark2 == 1) move = 6;
else if(mark1 == 26 && mark2 == 33) move = 8;
else if(mark1 == 26 && mark2 == 129) move = 6;
else if(mark1 == 28 && mark2 == 33) move = 7;
else if(mark1 == 28 && mark2 == 65) move = 6;
else if(mark1 == 30 && mark2 == 97) move = 8;
else if(mark1 == 30 && mark2 == 161) move = 7;
else if(mark1 == 30 && mark2 == 193) move = 9;
else if(mark1 == 32 && mark2 == 0) move = 3;
else if(mark1 == 33 && mark2 == 4) move = 4;
else if(mark1 == 33 && mark2 == 16) move = 2;
else if(mark1 == 34 && mark2 == 1) move = 7;
else if(mark1 == 34 && mark2 == 4) move = 4;
else if(mark1 == 35 && mark2 == 12) move = 5;
else if(mark1 == 35 && mark2 == 20) move = 7;
else if(mark1 == 36 && mark2 == 16) move = 9;
else if(mark1 == 37 && mark2 == 18) move = 8;
else if(mark1 == 37 && mark2 == 272) move = 2;          // Computer's Move: AI
else if(mark1 == 38 && mark2 == 9) move = 7;
else if(mark1 == 38 && mark2 == 17) move = 9;
else if(mark1 == 38 && mark2 == 65) move = 4;
else if(mark1 == 38 && mark2 == 272) move = 1;
else if(mark1 == 40 && mark2 == 1) move = 5;
else if(mark1 == 40 && mark2 == 4) move = 5;
else if(mark1 == 41 && mark2 == 18) move = 7;
else if(mark1 == 41 && mark2 == 20) move = 7;
else if(mark1 == 41 && mark2 == 80) move = 2;
else if(mark1 == 42 && mark2 == 17) move = 3;
else if(mark1 == 42 && mark2 == 20) move = 1;
else if(mark1 == 42 && mark2 == 65) move = 5;
else if(mark1 == 43 && mark2 == 84) move = 8;
else if(mark1 == 44 && mark2 == 17) move = 9;
else if(mark1 == 44 && mark2 == 272) move = 1;
else if(mark1 == 45 && mark2 == 82) move = 8;
else if(mark1 == 45 && mark2 == 146) move = 7;
else if(mark1 == 45 && mark2 == 274) move = 8;
else if(mark1 == 46 && mark2 == 81) move = 9;
else if(mark1 == 46 && mark2 == 273) move = 7;
else if(mark1 == 48 && mark2 == 1) move = 4;
else if(mark1 == 48 && mark2 == 4) move = 4;
```

```
else if(mark1 == 49 && mark2 == 12) move = 9;
else if(mark1 == 50 && mark2 == 9) move = 7;
else if(mark1 == 50 && mark2 == 12) move = 8;
else if(mark1 == 50 && mark2 == 65) move = 4;
else if(mark1 == 50 && mark2 == 129) move = 4;
else if(mark1 == 51 && mark2 == 140) move = 9;
else if(mark1 == 51 && mark2 == 268) move = 8;
else if(mark1 == 52 && mark2 == 9) move = 7;
else if(mark1 == 52 && mark2 == 65) move = 4;
else if(mark1 == 54 && mark2 == 73) move = 8;
else if(mark1 == 54 && mark2 == 137) move = 7;
else if(mark1 == 54 && mark2 == 193) move = 4;
else if(mark1 == 64 && mark2 == 0) move = 5;
else if(mark1 == 65 && mark2 == 16) move = 4;
else if(mark1 == 66 && mark2 == 1) move = 5;
else if(mark1 == 66 && mark2 == 16) move = 1;
else if(mark1 == 67 && mark2 == 20) move = 4;
else if(mark1 == 67 && mark2 == 24) move = 6;
else if(mark1 == 68 && mark2 == 16) move = 2;
else if(mark1 == 69 && mark2 == 18) move = 4;
else if(mark1 == 69 && mark2 == 24) move = 2;   // Computer's Move: AI
else if(mark1 == 70 && mark2 == 9) move = 5;
else if(mark1 == 70 && mark2 == 17) move = 4;
else if(mark1 == 71 && mark2 == 56) move = 8;
else if(mark1 == 72 && mark2 == 1) move = 2;
else if(mark1 == 72 && mark2 == 16) move = 1;
else if(mark1 == 74 && mark2 == 17) move = 9;
else if(mark1 == 76 && mark2 == 3) move = 5;
else if(mark1 == 76 && mark2 == 17) move = 2;
else if(mark1 == 76 && mark2 == 18) move = 1;
else if(mark1 == 77 && mark2 == 146) move = 6;
else if(mark1 == 78 && mark2 == 273) move = 6;
else if(mark1 == 80 && mark2 == 1) move = 3;
else if(mark1 == 82 && mark2 == 5) move = 8;
else if(mark1 == 82 && mark2 == 129) move = 3;
else if(mark1 == 88 && mark2 == 3) move = 3;
else if(mark1 == 88 && mark2 == 5) move = 2;
else if(mark1 == 88 && mark2 == 33) move = 3;
else if(mark1 == 90 && mark2 == 37) move = 9;
else if(mark1 == 90 && mark2 == 133) move = 6;
else if(mark1 == 90 && mark2 == 161) move = 3;
else if(mark1 == 96 && mark2 == 4) move = 1;
else if(mark1 == 96 && mark2 == 16) move = 2;
```

```
else if(mark1 == 97 && mark2 == 12) move = 5;
else if(mark1 == 97 && mark2 == 18) move = 8;
else if(mark1 == 97 && mark2 == 24) move = 2;
else if(mark1 == 98 && mark2 == 5) move = 5;
else if(mark1 == 98 && mark2 == 12) move = 5;
else if(mark1 == 98 && mark2 == 17) move = 9;
else if(mark1 == 99 && mark2 == 28) move = 8;
else if(mark1 == 100 && mark2 == 18) move = 8;
else if(mark1 == 100 && mark2 == 272) move = 1;
else if(mark1 == 101 && mark2 == 26) move = 8;
else if(mark1 == 101 && mark2 == 146) move = 4;
else if(mark1 == 101 && mark2 == 274) move = 8;
else if(mark1 == 102 && mark2 == 25) move = 9;
else if(mark1 == 102 && mark2 == 273) move = 4;
else if(mark1 == 104 && mark2 == 3) move = 3;
else if(mark1 == 104 && mark2 == 5) move = 2;
else if(mark1 == 104 && mark2 == 17) move = 2;
else if(mark1 == 104 && mark2 == 18) move = 1;
else if(mark1 == 104 && mark2 == 20) move = 1;          // Computer's Move: AI
else if(mark1 == 105 && mark2 == 146) move = 3;
else if(mark1 == 106 && mark2 == 21) move = 9;
else if(mark1 == 106 && mark2 == 273) move = 3;
else if(mark1 == 108 && mark2 == 19) move = 8;
else if(mark1 == 108 && mark2 == 146) move = 1;
else if(mark1 == 108 && mark2 == 273) move = 2;
else if(mark1 == 112 && mark2 == 5) move = 2;
else if(mark1 == 112 && mark2 == 9) move = 3;
else if(mark1 == 112 && mark2 == 12) move = 1;
else if(mark1 == 113 && mark2 == 268) move = 2;
else if(mark1 == 114 && mark2 == 13) move = 8;
else if(mark1 == 114 && mark2 == 133) move = 4;
else if(mark1 == 114 && mark2 == 137) move = 3;
else if(mark1 == 114 && mark2 == 140) move = 1;
else if(mark1 == 120 && mark2 == 7) move = 8;
else if(mark1 == 128 && mark2 == 0) move = 2;
else if(mark1 == 129 && mark2 == 2) move = 7;
else if(mark1 == 129 && mark2 == 16) move = 4;
else if(mark1 == 130 && mark2 == 1) move = 5;
else if(mark1 == 131 && mark2 == 20) move = 4;
else if(mark1 == 131 && mark2 == 24) move = 3;
else if(mark1 == 132 && mark2 == 2) move = 7;
else if(mark1 == 132 && mark2 == 16) move = 4;
else if(mark1 == 133 && mark2 == 18) move = 4;
```

```
else if(mark1 == 133 && mark2 == 24) move = 6;
else if(mark1 == 133 && mark2 == 66) move = 5;
else if(mark1 == 134 && mark2 == 9) move = 5;
else if(mark1 == 134 && mark2 == 17) move = 4;
else if(mark1 == 134 && mark2 == 24) move = 1;
else if(mark1 == 135 && mark2 == 56) move = 7;
else if(mark1 == 136 && mark2 == 1) move = 3;
else if(mark1 == 136 && mark2 == 2) move = 7;
else if(mark1 == 137 && mark2 == 66) move = 5;
else if(mark1 == 137 && mark2 == 80) move = 3;
else if(mark1 == 138 && mark2 == 5) move = 5;
else if(mark1 == 138 && mark2 == 17) move = 3;
else if(mark1 == 139 && mark2 == 84) move = 6;
else if(mark1 == 140 && mark2 == 17) move = 9;
else if(mark1 == 140 && mark2 == 66) move = 5;
else if(mark1 == 141 && mark2 == 82) move = 6;
else if(mark1 == 142 && mark2 == 273) move = 6;
else if(mark1 == 144 && mark2 == 1) move = 2;
else if(mark1 == 144 && mark2 == 2) move = 1;
else if(mark1 == 145 && mark2 == 66) move = 9;
else if(mark1 == 148 && mark2 == 3) move = 7;
else if(mark1 == 148 && mark2 == 65) move = 4;
else if(mark1 == 148 && mark2 == 66) move = 1;
else if(mark1 == 149 && mark2 == 322) move = 4;
else if(mark1 == 150 && mark2 == 73) move = 6;
else if(mark1 == 152 && mark2 == 3) move = 3;
else if(mark1 == 152 && mark2 == 5) move = 2;          // Computer's Move: AI
else if(mark1 == 152 && mark2 == 33) move = 2;
else if(mark1 == 152 && mark2 == 66) move = 6;
else if(mark1 == 153 && mark2 == 98) move = 9;
else if(mark1 == 153 && mark2 == 322) move = 6;
else if(mark1 == 156 && mark2 == 35) move = 7;
else if(mark1 == 156 && mark2 == 67) move = 6;
else if(mark1 == 156 && mark2 == 97) move = 2;
else if(mark1 == 156 && mark2 == 98) move = 1;
else if(mark1 == 160 && mark2 == 2) move = 7;
else if(mark1 == 160 && mark2 == 4) move = 1;
else if(mark1 == 161 && mark2 == 12) move = 5;
else if(mark1 == 161 && mark2 == 18) move = 7;
else if(mark1 == 161 && mark2 == 24) move = 3;
else if(mark1 == 161 && mark2 == 66) move = 5;
else if(mark1 == 162 && mark2 == 5) move = 5;
else if(mark1 == 162 && mark2 == 12) move = 5;
```

```
else if(mark1 == 162 && mark2 == 17) move = 3;
else if(mark1 == 162 && mark2 == 65) move = 4;
else if(mark1 == 163 && mark2 == 28) move = 7;
else if(mark1 == 163 && mark2 == 84) move = 4;
else if(mark1 == 164 && mark2 == 24) move = 9;
else if(mark1 == 164 && mark2 == 66) move = 9;
else if(mark1 == 164 && mark2 == 272) move = 1;
else if(mark1 == 165 && mark2 == 26) move = 9;
else if(mark1 == 165 && mark2 == 82) move = 9;
else if(mark1 == 165 && mark2 == 274) move = 4;
else if(mark1 == 165 && mark2 == 280) move = 2;
else if(mark1 == 165 && mark2 == 322) move = 4;
else if(mark1 == 166 && mark2 == 25) move = 7;
else if(mark1 == 166 && mark2 == 73) move = 5;
else if(mark1 == 166 && mark2 == 273) move = 4;
else if(mark1 == 166 && mark2 == 280) move = 1;
else if(mark1 == 168 && mark2 == 5) move = 2;
else if(mark1 == 168 && mark2 == 17) move = 2;
else if(mark1 == 168 && mark2 == 20) move = 1;
else if(mark1 == 168 && mark2 == 66) move = 5;          // Computer's Move: AI
else if(mark1 == 169 && mark2 == 82) move = 3;
else if(mark1 == 169 && mark2 == 84) move = 2;
else if(mark1 == 170 && mark2 == 21) move = 7;
else if(mark1 == 170 && mark2 == 81) move = 3;
else if(mark1 == 172 && mark2 == 19) move = 9;
else if(mark1 == 172 && mark2 == 82) move = 9;
else if(mark1 == 172 && mark2 == 273) move = 2;
else if(mark1 == 172 && mark2 == 322) move = 5;
else if(mark1 == 176 && mark2 == 3) move = 3;
else if(mark1 == 176 && mark2 == 5) move = 2;
else if(mark1 == 176 && mark2 == 9) move = 2;
else if(mark1 == 176 && mark2 == 12) move = 2;
else if(mark1 == 176 && mark2 == 66) move = 4;
else if(mark1 == 177 && mark2 == 14) move = 9;
else if(mark1 == 177 && mark2 == 74) move = 9;
else if(mark1 == 177 && mark2 == 268) move = 2;
else if(mark1 == 177 && mark2 == 322) move = 4;
else if(mark1 == 178 && mark2 == 73) move = 3;
else if(mark1 == 180 && mark2 == 11) move = 7;
else if(mark1 == 180 && mark2 == 67) move = 4;
else if(mark1 == 180 && mark2 == 73) move = 2;
else if(mark1 == 180 && mark2 == 74) move = 1;
else if(mark1 == 180 && mark2 == 322) move = 4;
```

```
else if(mark1 == 184 && mark2 == 7) move = 7;
else if(mark1 == 192 && mark2 == 2) move = 9;
else if(mark1 == 192 && mark2 == 16) move = 9;
else if(mark1 == 193 && mark2 == 24) move = 6;
else if(mark1 == 193 && mark2 == 258) move = 4;
else if(mark1 == 193 && mark2 == 272) move = 4;
else if(mark1 == 194 && mark2 == 17) move = 9;
else if(mark1 == 194 && mark2 == 272) move = 1;
else if(mark1 == 195 && mark2 == 28) move = 6;
else if(mark1 == 195 && mark2 == 56) move = 3;
else if(mark1 == 195 && mark2 == 280) move = 6;
else if(mark1 == 196 && mark2 == 18) move = 9;
else if(mark1 == 196 && mark2 == 24) move = 6;
else if(mark1 == 196 && mark2 == 258) move = 5;
else if(mark1 == 196 && mark2 == 272) move = 1;
else if(mark1 == 197 && mark2 == 26) move = 6;
else if(mark1 == 197 && mark2 == 56) move = 2;
else if(mark1 == 197 && mark2 == 266) move = 5;        // Computer's Move: AI
else if(mark1 == 197 && mark2 == 274) move = 4;
else if(mark1 == 197 && mark2 == 280) move = 6;
else if(mark1 == 198 && mark2 == 25) move = 6;
else if(mark1 == 198 && mark2 == 56) move = 1;
else if(mark1 == 198 && mark2 == 273) move = 4;
else if(mark1 == 200 && mark2 == 3) move = 3;
else if(mark1 == 200 && mark2 == 5) move = 2;
else if(mark1 == 200 && mark2 == 17) move = 9;
else if(mark1 == 200 && mark2 == 258) move = 1;
else if(mark1 == 200 && mark2 == 272) move = 1;
else if(mark1 == 202 && mark2 == 21) move = 9;
else if(mark1 == 202 && mark2 == 273) move = 3;
else if(mark1 == 204 && mark2 == 19) move = 9;
else if(mark1 == 204 && mark2 == 259) move = 5;
else if(mark1 == 204 && mark2 == 273) move = 2;
else if(mark1 == 204 && mark2 == 274) move = 1;
else if(mark1 == 208 && mark2 == 3) move = 3;
else if(mark1 == 208 && mark2 == 5) move = 2;
else if(mark1 == 208 && mark2 == 258) move = 3;
else if(mark1 == 209 && mark2 == 262) move = 6;
else if(mark1 == 209 && mark2 == 266) move = 3;
else if(mark1 == 216 && mark2 == 7) move = 6;
else if(mark1 == 216 && mark2 == 35) move = 3;
else if(mark1 == 216 && mark2 == 37) move = 2;
else if(mark1 == 216 && mark2 == 259) move = 3;
```

```
else if(mark1 == 216 && mark2 == 262) move = 1;
else if(mark1 == 224 && mark2 == 5) move = 2;
else if(mark1 == 224 && mark2 == 18) move = 9;
else if(mark1 == 224 && mark2 == 258) move = 1;
else if(mark1 == 224 && mark2 == 272) move = 1;
else if(mark1 == 225 && mark2 == 26) move = 9;
else if(mark1 == 225 && mark2 == 28) move = 9;
else if(mark1 == 225 && mark2 == 266) move = 3;
else if(mark1 == 225 && mark2 == 274) move = 4;
else if(mark1 == 225 && mark2 == 280) move = 2;
else if(mark1 == 226 && mark2 == 21) move = 9;
else if(mark1 == 226 && mark2 == 28) move = 9;
else if(mark1 == 226 && mark2 == 273) move = 3;
else if(mark1 == 228 && mark2 == 259) move = 5;
else if(mark1 == 228 && mark2 == 273) move = 2;
else if(mark1 == 228 && mark2 == 274) move = 1;
else if(mark1 == 228 && mark2 == 280) move = 1;        // Computer's Move: AI
else if(mark1 == 232 && mark2 == 7) move = 5;
else if(mark1 == 232 && mark2 == 19) move = 3;
else if(mark1 == 232 && mark2 == 21) move = 2;
else if(mark1 == 232 && mark2 == 259) move = 3;
else if(mark1 == 232 && mark2 == 273) move = 2;
else if(mark1 == 232 && mark2 == 274) move = 1;
else if(mark1 == 240 && mark2 == 7) move = 4;
else if(mark1 == 240 && mark2 == 11) move = 3;
else if(mark1 == 240 && mark2 == 13) move = 2;
else if(mark1 == 240 && mark2 == 14) move = 1;
else if(mark1 == 240 && mark2 == 259) move = 3;
else if(mark1 == 240 && mark2 == 262) move = 1;
else if(mark1 == 256 && mark2 == 0) move = 5;
else if(mark1 == 257 && mark2 == 16) move = 2;
else if(mark1 == 258 && mark2 == 1) move = 5;
else if(mark1 == 258 && mark2 == 16) move = 1;
else if(mark1 == 259 && mark2 == 20) move = 4;
else if(mark1 == 260 && mark2 == 16) move = 6;
else if(mark1 == 261 && mark2 == 18) move = 6;
else if(mark1 == 261 && mark2 == 48) move = 2;
else if(mark1 == 262 && mark2 == 9) move = 6;
else if(mark1 == 262 && mark2 == 17) move = 6;
else if(mark1 == 262 && mark2 == 48) move = 4;
else if(mark1 == 263 && mark2 == 56) move = 7;
else if(mark1 == 264 && mark2 == 1) move = 3;
else if(mark1 == 264 && mark2 == 16) move = 1;
```

```
else if(mark1 == 265 && mark2 == 18) move = 7;
else if(mark1 == 265 && mark2 == 80) move = 2;
else if(mark1 == 266 && mark2 == 5) move = 5;
else if(mark1 == 266 && mark2 == 17) move = 3;
else if(mark1 == 267 && mark2 == 84) move = 6;
else if(mark1 == 268 && mark2 == 17) move = 6;
else if(mark1 == 268 && mark2 == 48) move = 1;
else if(mark1 == 269 && mark2 == 50) move = 8;
else if(mark1 == 269 && mark2 == 82) move = 8;
else if(mark1 == 269 && mark2 == 146) move = 6;
else if(mark1 == 270 && mark2 == 49) move = 7;
else if(mark1 == 272 && mark2 == 1) move = 3;
else if(mark1 == 274 && mark2 == 5) move = 8;
else if(mark1 == 274 && mark2 == 129) move = 3;
else if(mark1 == 276 && mark2 == 65) move = 4;
else if(mark1 == 278 && mark2 == 41) move = 7;
else if(mark1 == 278 && mark2 == 73) move = 6;
else if(mark1 == 278 && mark2 == 193) move = 4;
else if(mark1 == 280 && mark2 == 5) move = 2;
else if(mark1 == 280 && mark2 == 33) move = 2;
else if(mark1 == 282 && mark2 == 133) move = 6;
else if(mark1 == 282 && mark2 == 161) move = 3;
else if(mark1 == 284 && mark2 == 35) move = 7;
else if(mark1 == 284 && mark2 == 97) move = 2;
else if(mark1 == 288 && mark2 == 4) move = 1;
else if(mark1 == 288 && mark2 == 16) move = 3;          // Computer's Move: AI
else if(mark1 == 289 && mark2 == 12) move = 5;
else if(mark1 == 289 && mark2 == 18) move = 3;
else if(mark1 == 289 && mark2 == 20) move = 2;
else if(mark1 == 290 && mark2 == 5) move = 7;
else if(mark1 == 290 && mark2 == 12) move = 7;
else if(mark1 == 290 && mark2 == 17) move = 3;
else if(mark1 == 290 && mark2 == 20) move = 7;
else if(mark1 == 290 && mark2 == 65) move = 3;
else if(mark1 == 291 && mark2 == 28) move = 7;
else if(mark1 == 291 && mark2 == 76) move = 5;
else if(mark1 == 291 && mark2 == 84) move = 4;
else if(mark1 == 293 && mark2 == 146) move = 4;
else if(mark1 == 294 && mark2 == 73) move = 5;
else if(mark1 == 296 && mark2 == 5) move = 2;
else if(mark1 == 296 && mark2 == 17) move = 3;
else if(mark1 == 296 && mark2 == 20) move = 1;
else if(mark1 == 297 && mark2 == 22) move = 7;
```

```
else if(mark1 == 297 && mark2 == 82) move = 3;
else if(mark1 == 297 && mark2 == 84) move = 2;
else if(mark1 == 298 && mark2 == 21) move = 7;
else if(mark1 == 298 && mark2 == 69) move = 5;
else if(mark1 == 298 && mark2 == 81) move = 3;
else if(mark1 == 298 && mark2 == 84) move = 1;
else if(mark1 == 304 && mark2 == 5) move = 2;
else if(mark1 == 304 && mark2 == 9) move = 3;
else if(mark1 == 304 && mark2 == 12) move = 1;
else if(mark1 == 306 && mark2 == 13) move = 7;
else if(mark1 == 306 && mark2 == 69) move = 4;
else if(mark1 == 306 && mark2 == 73) move = 3;
else if(mark1 == 306 && mark2 == 76) move = 1;
else if(mark1 == 306 && mark2 == 133) move = 4;
else if(mark1 == 306 && mark2 == 137) move = 7;
else if(mark1 == 306 && mark2 == 140) move = 1;
else if(mark1 == 308 && mark2 == 73) move = 2;          // Computer's Move: AI
else if(mark1 == 312 && mark2 == 7) move = 7;
else if(mark1 == 320 && mark2 == 16) move = 8;
else if(mark1 == 321 && mark2 == 18) move = 8;
else if(mark1 == 321 && mark2 == 24) move = 6;
else if(mark1 == 321 && mark2 == 144) move = 2;
else if(mark1 == 322 && mark2 == 17) move = 8;
else if(mark1 == 322 && mark2 == 144) move = 1;
else if(mark1 == 323 && mark2 == 28) move = 6;
else if(mark1 == 323 && mark2 == 56) move = 3;
else if(mark1 == 324 && mark2 == 18) move = 8;
else if(mark1 == 324 && mark2 == 48) move = 4;
else if(mark1 == 324 && mark2 == 144) move = 2;
else if(mark1 == 325 && mark2 == 26) move = 6;
else if(mark1 == 325 && mark2 == 50) move = 4;
else if(mark1 == 325 && mark2 == 56) move = 2;
else if(mark1 == 325 && mark2 == 146) move = 4;
else if(mark1 == 326 && mark2 == 25) move = 6;
else if(mark1 == 326 && mark2 == 41) move = 5;
else if(mark1 == 326 && mark2 == 49) move = 4;
else if(mark1 == 326 && mark2 == 56) move = 1;
else if(mark1 == 326 && mark2 == 145) move = 6;
else if(mark1 == 328 && mark2 == 3) move = 3;
else if(mark1 == 328 && mark2 == 5) move = 2;
else if(mark1 == 328 && mark2 == 17) move = 8;
else if(mark1 == 328 && mark2 == 144) move = 2;
else if(mark1 == 329 && mark2 == 146) move = 3;
```

```
else if(mark1 == 330 && mark2 == 21) move = 8;
else if(mark1 == 330 && mark2 == 145) move = 3;
else if(mark1 == 332 && mark2 == 19) move = 8;
else if(mark1 == 332 && mark2 == 49) move = 8;
else if(mark1 == 332 && mark2 == 145) move = 2;
else if(mark1 == 332 && mark2 == 146) move = 1;
else if(mark1 == 336 && mark2 == 5) move = 2;
else if(mark1 == 338 && mark2 == 133) move = 4;
else if(mark1 == 344 && mark2 == 7) move = 6;
else if(mark1 == 344 && mark2 == 35) move = 3;
else if(mark1 == 344 && mark2 == 37) move = 2;
else if(mark1 == 352 && mark2 == 5) move = 2;
else if(mark1 == 352 && mark2 == 18) move = 8;
else if(mark1 == 352 && mark2 == 20) move = 8;
else if(mark1 == 352 && mark2 == 144) move = 2;
else if(mark1 == 353 && mark2 == 22) move = 8;
else if(mark1 == 353 && mark2 == 26) move = 8;
else if(mark1 == 353 && mark2 == 28) move = 8;
else if(mark1 == 353 && mark2 == 146) move = 3;
else if(mark1 == 353 && mark2 == 148) move = 2;
else if(mark1 == 354 && mark2 == 21) move = 8;
else if(mark1 == 354 && mark2 == 28) move = 8;
else if(mark1 == 354 && mark2 == 145) move = 3;        // Computer's Move: AI
else if(mark1 == 354 && mark2 == 148) move = 1;
else if(mark1 == 356 && mark2 == 146) move = 1;
else if(mark1 == 360 && mark2 == 7) move = 5;
else if(mark1 == 360 && mark2 == 19) move = 3;
else if(mark1 == 360 && mark2 == 21) move = 2;
else if(mark1 == 360 && mark2 == 145) move = 2;
else if(mark1 == 360 && mark2 == 146) move = 1;
else if(mark1 == 360 && mark2 == 148) move = 2;
else if(mark1 == 368 && mark2 == 7) move = 4;
else if(mark1 == 368 && mark2 == 13) move = 2;
else if(mark1 == 384 && mark2 == 2) move = 7;
else if(mark1 == 384 && mark2 == 16) move = 7;
else if(mark1 == 385 && mark2 == 18) move = 7;
else if(mark1 == 385 && mark2 == 24) move = 6;
else if(mark1 == 385 && mark2 == 66) move = 5;
else if(mark1 == 385 && mark2 == 80) move = 3;
else if(mark1 == 386 && mark2 == 17) move = 7;
else if(mark1 == 386 && mark2 == 80) move = 1;
else if(mark1 == 387 && mark2 == 28) move = 6;
else if(mark1 == 387 && mark2 == 56) move = 3;
```

```
else if(mark1 == 387 && mark2 == 84) move = 4;
else if(mark1 == 388 && mark2 == 24) move = 6;
else if(mark1 == 388 && mark2 == 48) move = 4;
else if(mark1 == 388 && mark2 == 66) move = 6;
else if(mark1 == 388 && mark2 == 80) move = 6;
else if(mark1 == 389 && mark2 == 26) move = 6;
else if(mark1 == 389 && mark2 == 50) move = 4;
else if(mark1 == 389 && mark2 == 56) move = 2;
else if(mark1 == 389 && mark2 == 82) move = 6;
else if(mark1 == 389 && mark2 == 98) move = 5;
else if(mark1 == 389 && mark2 == 112) move = 4;
else if(mark1 == 390 && mark2 == 25) move = 6;
else if(mark1 == 390 && mark2 == 41) move = 5;
else if(mark1 == 390 && mark2 == 49) move = 4;
else if(mark1 == 390 && mark2 == 56) move = 1;
else if(mark1 == 390 && mark2 == 81) move = 4;
else if(mark1 == 390 && mark2 == 112) move = 4;
else if(mark1 == 392 && mark2 == 5) move = 2;
else if(mark1 == 392 && mark2 == 17) move = 7;
else if(mark1 == 392 && mark2 == 66) move = 3;
else if(mark1 == 392 && mark2 == 80) move = 3;          // Computer's Move: AI
else if(mark1 == 393 && mark2 == 70) move = 5;
else if(mark1 == 393 && mark2 == 82) move = 3;
else if(mark1 == 393 && mark2 == 84) move = 2;
else if(mark1 == 394 && mark2 == 21) move = 7;
else if(mark1 == 394 && mark2 == 81) move = 3;
else if(mark1 == 394 && mark2 == 84) move = 1;
else if(mark1 == 396 && mark2 == 49) move = 7;
else if(mark1 == 396 && mark2 == 81) move = 6;
else if(mark1 == 396 && mark2 == 82) move = 6;
else if(mark1 == 396 && mark2 == 98) move = 1;
else if(mark1 == 396 && mark2 == 112) move = 1;
else if(mark1 == 400 && mark2 == 3) move = 3;
else if(mark1 == 400 && mark2 == 5) move = 2;
else if(mark1 == 400 && mark2 == 66) move = 1;
else if(mark1 == 404 && mark2 == 67) move = 4;
else if(mark1 == 404 && mark2 == 73) move = 2;
else if(mark1 == 404 && mark2 == 98) move = 1;
else if(mark1 == 408 && mark2 == 7) move = 6;
else if(mark1 == 408 && mark2 == 35) move = 3;
else if(mark1 == 408 && mark2 == 67) move = 3;
else if(mark1 == 408 && mark2 == 70) move = 1;
else if(mark1 == 408 && mark2 == 98) move = 1;
```

```
else if(mark1 == 416 && mark2 == 5) move = 2;
else if(mark1 == 416 && mark2 == 20) move = 7;
else if(mark1 == 416 && mark2 == 66) move = 3;
else if(mark1 == 416 && mark2 == 80) move = 3;
else if(mark1 == 417 && mark2 == 22) move = 7;
else if(mark1 == 417 && mark2 == 28) move = 7;
else if(mark1 == 417 && mark2 == 70) move = 5;
else if(mark1 == 417 && mark2 == 82) move = 3;
else if(mark1 == 417 && mark2 == 84) move = 2;
else if(mark1 == 418 && mark2 == 21) move = 7;
else if(mark1 == 418 && mark2 == 28) move = 7;
else if(mark1 == 418 && mark2 == 69) move = 4;
else if(mark1 == 418 && mark2 == 73) move = 3;
else if(mark1 == 418 && mark2 == 76) move = 1;
else if(mark1 == 418 && mark2 == 81) move = 3;          // Computer's Move: AI
else if(mark1 == 418 && mark2 == 84) move = 1;
else if(mark1 == 424 && mark2 == 7) move = 5;
else if(mark1 == 424 && mark2 == 19) move = 3;
else if(mark1 == 424 && mark2 == 21) move = 2;
else if(mark1 == 424 && mark2 == 70) move = 1;
else if(mark1 == 424 && mark2 == 81) move = 3;
else if(mark1 == 424 && mark2 == 82) move = 3;
else if(mark1 == 424 && mark2 == 84) move = 1;
else if(mark1 == 432 && mark2 == 7) move = 4;
else if(mark1 == 432 && mark2 == 11) move = 3;
else if(mark1 == 432 && mark2 == 13) move = 2;
else if(mark1 == 432 && mark2 == 14) move = 1;
else if(mark1 == 432 && mark2 == 67) move = 3;
else if(mark1 == 432 && mark2 == 70) move = 1;
else if(mark1 == 432 && mark2 == 74) move = 1;
else if(mark1 == 449 && mark2 == 56) move = 2;
else if(mark1 == 452 && mark2 == 56) move = 1;
else if(mark1 == 456 && mark2 == 7) move = 5;
else if(mark1 == 464 && mark2 == 7) move = 4;
else if(mark1 == 480 && mark2 == 7) move = 4;
nextpos = move;
```

```
        case (nextpos)
            4'b0001: if (mark[0]==0) begin mark2[0]=1; player=0; end
            4'b0010: if (mark[1]==0) begin mark2[1]=1; player=0; end
            4'b0011: if (mark[2]==0) begin mark2[2]=1; player=0; end

            4'b0100: if (mark[3]==0) begin mark2[3]=1; player=0; end
            4'b0101: if (mark[4]==0) begin mark2[4]=1; player=0; end
            4'b0110: if (mark[5]==0) begin mark2[5]=1; player=0; end

            4'b0111: if (mark[6]==0) begin mark2[6]=1; player=0; end
            4'b1000: if (mark[7]==0) begin mark2[7]=1; player=0; end
            4'b1001: if (mark[8]==0) begin mark2[8]=1; player=0; end
        endcase
    end
    if (reset == 1)
    begin
        mark1=0;
        mark2=0;
        win1 = 0;
        win2 = 0;
        player = 0;
    end

// CHECKING for DRAW and WIN.

    if ((m1[0]& m1[1]& m1[2]) | (m1[3]& m1[4]& m1[5]) | (m1[6]& m1[7]& m1[8]) | (m1[0]&
m1[3]& m1[6]) | (m1[1]& m1[4]& m1[7]) | (m1[2]& m1[5]& m1[8]) | (m1[0]& m1[4]& m1[8]) |
(m1[2]& m1[4]& m1[6]))
    win1 = 1;

    else if ((m2[0]& m2[1]& m2[2]) | (m2[3]& m2[4]& m2[5]) | (m2[6]& m2[7]& m2[8]) |
(m2[0]& m2[3]& m2[6]) | (m2[1]& m2[4]& m2[7]) | (m2[2]& m2[5]& m2[8]) | (m2[0]& m2[4]&
m2[8]) | (m2[2]& m2[4]& m2[6]))
    win2 = 1;

    else if (mark == 9'b111111111)
    begin
    win1=1;
    win2=1;
    end
  end
endmodule
```

CONSTRAINT FILE DEFINITION:

```
set_property -dict { PACKAGE_PIN L16   IOSTANDARD LVCMOS33 } [get_ports {clk}];

set_property -dict { PACKAGE_PIN G15   IOSTANDARD LVCMOS33 } [get_ports {nextposa[0]}];
set_property -dict { PACKAGE_PIN P15   IOSTANDARD LVCMOS33 } [get_ports {nextposa[1]}];
set_property -dict { PACKAGE_PIN W13   IOSTANDARD LVCMOS33 } [get_ports {nextposa[2]}];
set_property -dict { PACKAGE_PIN T16   IOSTANDARD LVCMOS33 } [get_ports {nextposa[3]}];


set_property -dict { PACKAGE_PIN P16   IOSTANDARD LVCMOS33 } [get_ports {reset}];
set_property -dict { PACKAGE_PIN V16   IOSTANDARD LVCMOS33 } [get_ports {en2}];
set_property -dict { PACKAGE_PIN Y16   IOSTANDARD LVCMOS33 } [get_ports {en1}];


##Pmod Header JA (XADC)
set_property -dict { PACKAGE_PIN N15   IOSTANDARD LVCMOS33 } [get_ports {m1[0]}];
set_property -dict { PACKAGE_PIN L14   IOSTANDARD LVCMOS33 } [get_ports {m2[0]}];
set_property -dict { PACKAGE_PIN K16   IOSTANDARD LVCMOS33 } [get_ports {m1[1]}];
set_property -dict { PACKAGE_PIN K14   IOSTANDARD LVCMOS33 } [get_ports {m2[1]}];
set_property -dict { PACKAGE_PIN N16   IOSTANDARD LVCMOS33 } [get_ports {m1[2]}];
set_property -dict { PACKAGE_PIN L15   IOSTANDARD LVCMOS33 } [get_ports {m2[2]}];
set_property -dict { PACKAGE_PIN J16   IOSTANDARD LVCMOS33 } [get_ports {m1[3]}];
set_property -dict { PACKAGE_PIN J14   IOSTANDARD LVCMOS33 } [get_ports {m2[3]}];

##Pmod Header JB
set_property -dict { PACKAGE_PIN T20   IOSTANDARD LVCMOS33 } [get_ports {m1[4]}];
set_property -dict { PACKAGE_PIN U20   IOSTANDARD LVCMOS33 } [get_ports {m2[4]}];
set_property -dict { PACKAGE_PIN V20   IOSTANDARD LVCMOS33 } [get_ports {m1[5]}];
set_property -dict { PACKAGE_PIN W20   IOSTANDARD LVCMOS33 } [get_ports {m2[5]}];
set_property -dict { PACKAGE_PIN Y18   IOSTANDARD LVCMOS33 } [get_ports {m1[6]}];
set_property -dict { PACKAGE_PIN Y19   IOSTANDARD LVCMOS33 } [get_ports {m2[6]}];
set_property -dict { PACKAGE_PIN W18   IOSTANDARD LVCMOS33 } [get_ports {m1[7]}];
set_property -dict { PACKAGE_PIN W19   IOSTANDARD LVCMOS33 } [get_ports {m2[7]}];

##Pmod Header JC
set_property -dict { PACKAGE_PIN V15   IOSTANDARD LVCMOS33 } [get_ports {m1[8]}];
set_property -dict { PACKAGE_PIN W15   IOSTANDARD LVCMOS33 } [get_ports {m2[8]}];

##Pmod Header JD
set_property -dict { PACKAGE_PIN T14   IOSTANDARD LVCMOS33 } [get_ports {w1}];
set_property -dict { PACKAGE_PIN T15   IOSTANDARD LVCMOS33 } [get_ports {w2}];
```

CODE EXPLANATION

Basic Input and Output:

This part of the code checks whether the inputs given by the user or move played by the computer is valid and then assigns the corresponding squares their new values, thereby preventing illegal moves.

If the corresponding square is not empty, it does not accept the input and the same player has to play the next move, otherwise if the bit is empty, then the input is accepted and registered as 1.

Implementation of AI:

This part of the Verilog code is made up of if and else statements. This is based on the Minimax algorithm that we were able to implement in C++ programming language.

The if statements have 2 operands - the human's and computer's 9 bit states. They assign the next best 4-bit register move, the value that given both the states what an ideal smart player would play to win.

Winner Evaluation:

This part of the code validates which player is the winner. It also checks if the game ended in a draw and assigns the winner corresponding to the 2 LEDs.

The evaluation part checks if the 3 rows or 3 columns or the 2 diagonals have all states as 1 corresponding to either of the player states. If yes, then the win state corresponding to that particular player is assigned 1. Also if there is no winner and if all the positions have been filled, then a draw is declared.

TEST BENCH

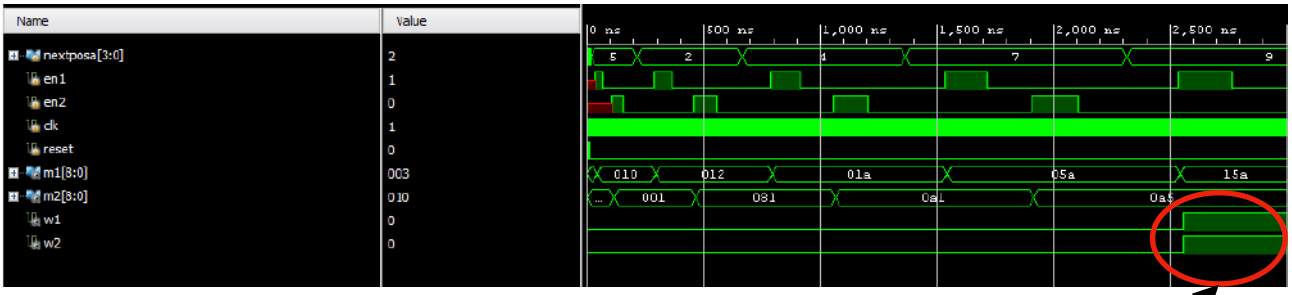
```
module testBench();
    reg [3:0] nextposa;
    reg en1, en2, clk, reset;
    wire [8:0] m1, m2;
    wire w1, w2;
    success check(
        .nextposa(nextposa),
        .en1(en1),
        .en2(en2),
        .clk(clk),
        .reset(reset),
        .m1(m1),
        .m2(m2),
        .w1(w1),
        .w2(w2)
    );

    initial
    begin
        clk = 1'b0;
        #1 reset = 1'b1;                //Reseting the game
        #5 reset = 1'b0;                // TEST BENCH FOR A DRAW GAME
        #10 nextposa = 4'b0101;     //Move 5, 1
        #20 en1 = 1'b1;
        #30 en1 = 1'b0;
        #40 en2 = 1'b1;
        #50 en2 = 1'b0;
        #60 nextposa = 4'b0010;     //Move 2, 8
        #70 en1 = 1'b1;
        #80 en1 = 1'b0;
        #90 en2 = 1'b1;
        #100 en2 = 1'b0;
        #110 nextposa = 4'b0100;    //Move 4, 6
        #120 en1 = 1'b1;
        #130 en1 = 1'b0;
        #140 en2 = 1'b1;
        #150 en2 = 1'b0;
        #160 nextposa = 4'b0111;    //Move 7, 3
        #170 en1 = 1'b1;
        #180 en1 = 1'b0;
```

```
        #190 en2 = 1'b1;
        #200 en2 = 1'b0;
        #210 nextposa = 4'b1001;    //Move 9 - Results in Draw (Last Move)
        #220 en1 = 1'b1;
        #230 en1 = 1'b0;
        #250 reset = 1'b1;          //Reseting the game again.
        #260 reset = 1'b0;          // TEST BENCH FOR A COMPUTER WIN GAME
        #270 nextposa = 4'b0001;    //Move 1, 5
        #280 en1 = 1'b1;
        #290 en1 = 1'b0;
        #300 en2 = 1'b1;
        #310 en2 = 1'b0;
        #320 nextposa = 4'b0010;    //Move 2, 3
        #330 en1 = 1'b1;
        #340 en1 = 1'b0;
        #350 en2 = 1'b1;
        #360 en2 = 1'b0;
        #370 nextposa = 4'b0100;    //Move 4, 7 - Results in Computer winning the game.
        #380 en1 = 1'b1;
        #390 en1 = 1'b0;
        #400 en2 = 1'b1;
        #410 en2 = 1'b0;
        #420 $finish;               //End the Test Bench.
    end
    always
        #5 clk = !clk;
endmodule
```

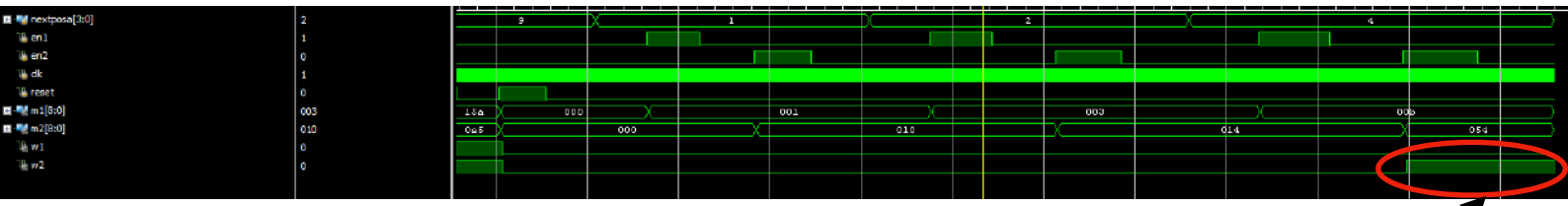
TESTBENCH TIMING DIAGRAMS

1.) **DRAW:**



Both the LEDs glow indicating a draw.

2.) **COMPUTER WINNING:**



LED corresponding to the computer glows, indicating its victory.

CONCLUSION

A Tic Tac Toe game machine with computer mode was successfully implemented and played with.

We learned advanced techniques like test benches to be considered while making projects on an FPGA. A clever use of another programming language executed using processors can simplify the Verilog code that we write.

It is thereby evident that the possibilities of an FPGA is **limitless**.