Tic Tac Toe Game

INFINITY

22-06-2022

1 Team members

Md Limon Apu
Syed Muhammad Abis Rizvi Pritish
Sanjay Samnat

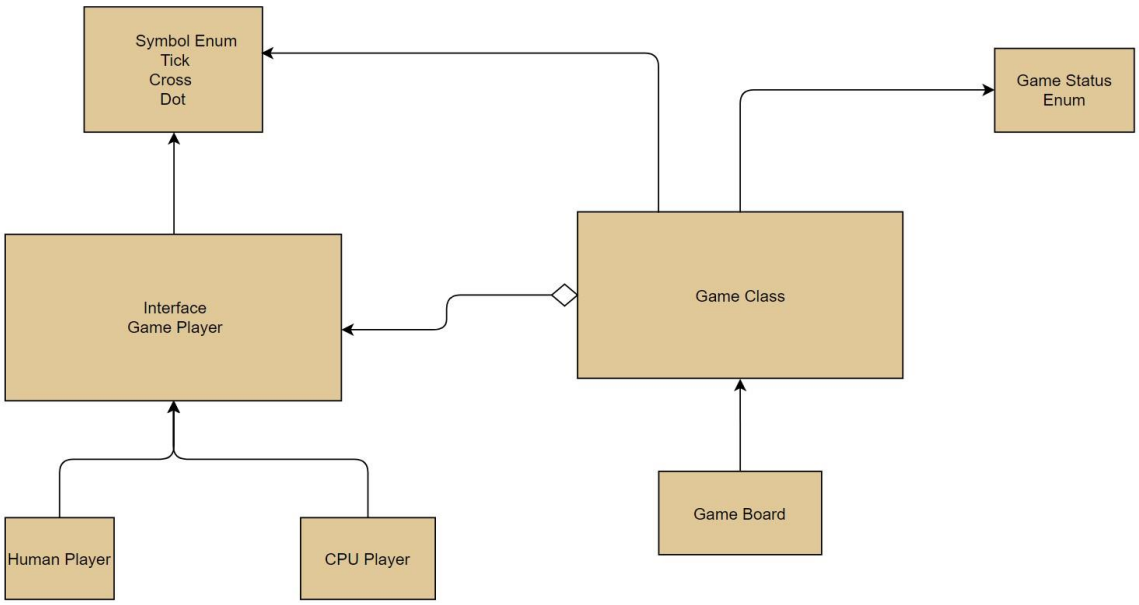
2 Introduction

Tic Tac Toe is a two-player, 3x3 grid, paper and pencil game. The winner is decided by the person who makes his or her first three marks in a diagonal, vertical, or horizontal row.

FPGAs are ideal for prototyping ASICs and processors. An FPGA can be reprogrammed until the ASIC or processor design is finalized and bug-free, at which point the final ASIC can be manufactured. FPGAs are used by Intel to prototype new chips. VHDL enables complicated electronic circuit behavior to be captured in a design system for automatic circuit synthesis or system simulation.

3 Concept description

Block diagram of our project



The diagram represents following things

- There is a Symbol Enum which represents different symbols used on the board. The symbol could be a Tick, Cross, or Dot. The Dot represents an empty block in the board.

- There is an Interface Game Player that represents the player in the game.
- Then there below shows that player can be a human player or a computer player.
- There is a board class that only captures the details of the board.
- There is a Game Status Enum class that defines the different starts of the game. A game can be either InProgress, Draw, FirstPlayerWin, or SecondPlayerWin state.
- There is a game class that controls the execution of the game

What is the main application for your prototype?

Tic Tac Toe is a game that supports one player and two player game play, so you can play against another human or against your mobile phone. The AI (Artificial Intelligence) for one player mode includes different difficulty levels, so you can play against a computer player that matches your skill level.Main application of our prototype is TIC Tac Toe game.

4 Project/Team management

Which project methods you used in your project?

VHDL - **Abis, Limon**,**Pritish**
Testbench - **Abis**,**Pritish**
PCB Desgin Eagle - **Pritish, Limon**,**Abis**
Xilix - **Abis**,**Pritish**
Documentation- **Abis**,**Pritish, Limon**
Presentation- **Abis,Pritish, Limon**
.

5 Technologies

- *We use VHDL in modelsim and use Xilinx as a simulator.*
- *We use Eagle and use it for schematic and layout design.*
- *For FPGA we use our modelsim code into Xilinx.*

6 VHDL Implementation

For our VHDL code we use modelsim and test and simulate the result in the ide. *Here is some snippet of our main code file into board and player one winning-*

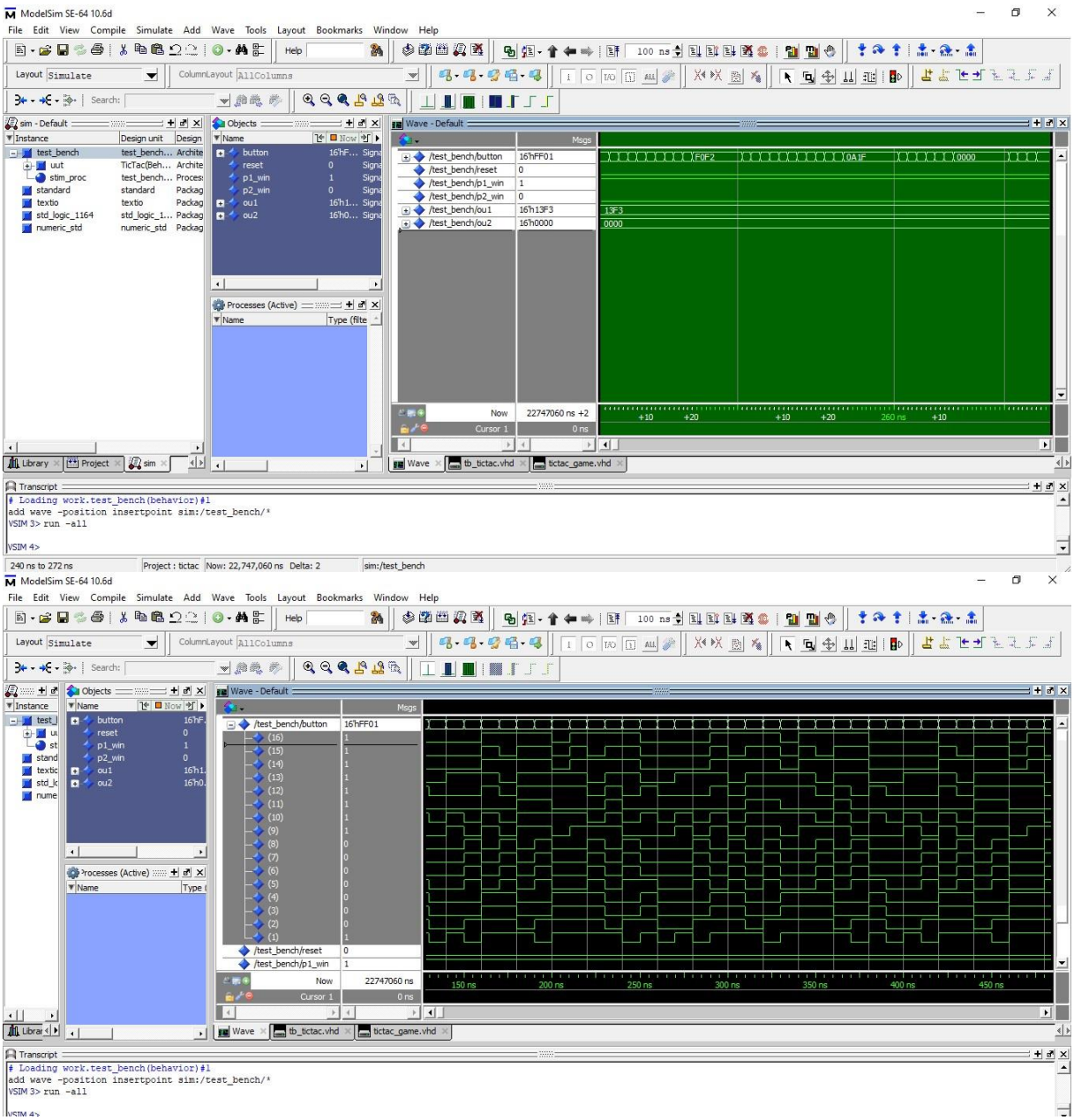
```
gen_spots : for i in 1 to 16 generate --3 flip flops share a clock (button) for every space on the board
    process(button(i), reset)
    begin
        if(reset = '1') then
            o(i)  <= '0';
            o1(i) <= '0';
            o2(i) <= '0';
        elsif(button(i)'event and button(i)='1' and o(i)='0' and win='0') then
            o(i) <= '1';
            if (p = '0') then
                o1(i) <= '1';
            else
                o2(i) <= '1';
            end if;
        end if;
    end process;
end generate gen_spots;
```

```
process(o1) --checks if player 1 wins
begin
    win1 <= '0'; --only happens if none of the win1 <= '1' statements occur
    for i in 0 to 3 loop
        if (o1(1+i*4)='1' and o1(2+i*4)='1' and o1(3+i*4)='1' and o1(4+i*4)='1') then --rows
            win1 <= '1';
        end if;
        if (o1(1+i)='1' and o1(5+i)='1' and o1(9+i)='1' and o1(13+i)='1') then --columns
            win1 <= '1';
        end if;
    end loop;
    if (o1(1)='1' and o1(6)='1' and o1(11)='1' and o1(16)='1') or (o1(4)='1' and o1(7)='1' and o1(10)='1' and o1(13)='1') then --diagonals
        win1 <= '1';
    end if;
end process;
```

And for the testbench we use various elements that contains an instantiation of a design entity. Here is some example of our testbench for the game-

```
uut: TicTac PORT MAP (
        button => button,----button is linked with button
        reset => reset,---reset is linked with reset
        p_to_play => p_to_play,----p_to_play is linked with p_to_play
        p1_win => p1_win,---------p1_win is linked with p1_win
        p2_win => p2_win,---p2_win is linked with p2 win
        ou1 => ou1, ---ou1 is linked with ou1
        ou2 => ou2---ou2 is linked with ou2
      );
```

Simulation of the code running in Modelsim-

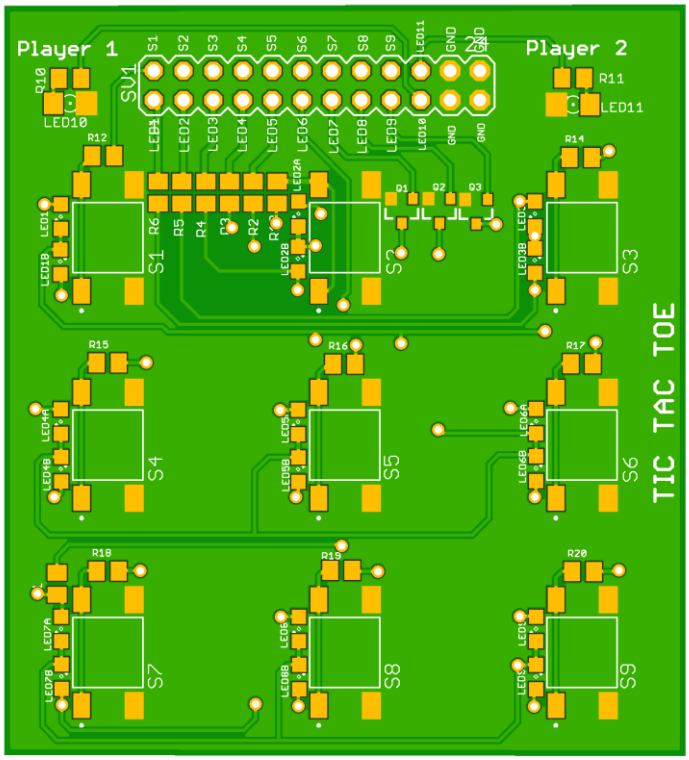


7 PCB Design

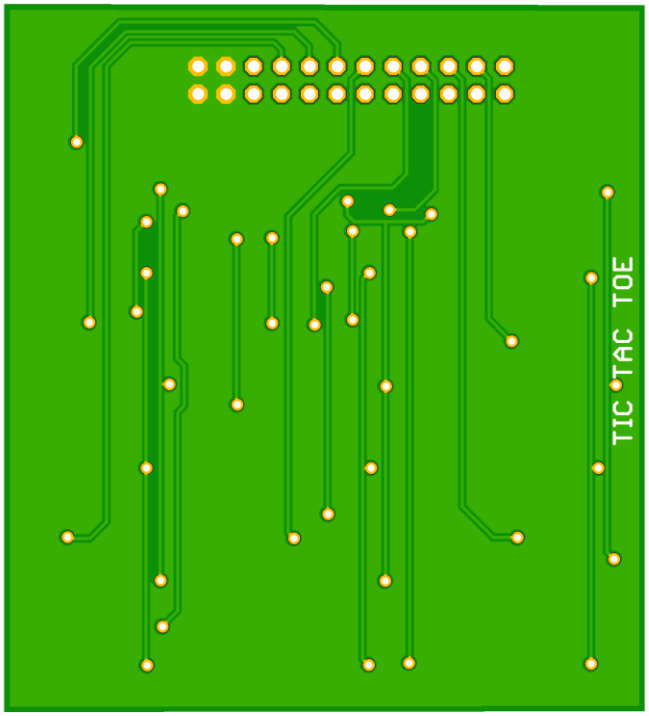
For our project we use Eagle for the layout and schematic of our pcb board design. We use normal logic gates, led, resistors etc. Building the tic-tac-toe game, we use more arbitrary structures with the same PCB, by assembling it in different ways. We made a slightly modified design to make it slightly more versatile for structures, and with the tic-tac-toe markings removed.

The PCB has a grid of 3x3 LEDs in the middle, a single resistor for the LEDs, 3-wide headers (GND-VCC-GND, symmetrical, so they can be plugged in either way) in the middle on each side for board tiles to connect, and a set of vertical headers in each corner to plug the piece tiles into.

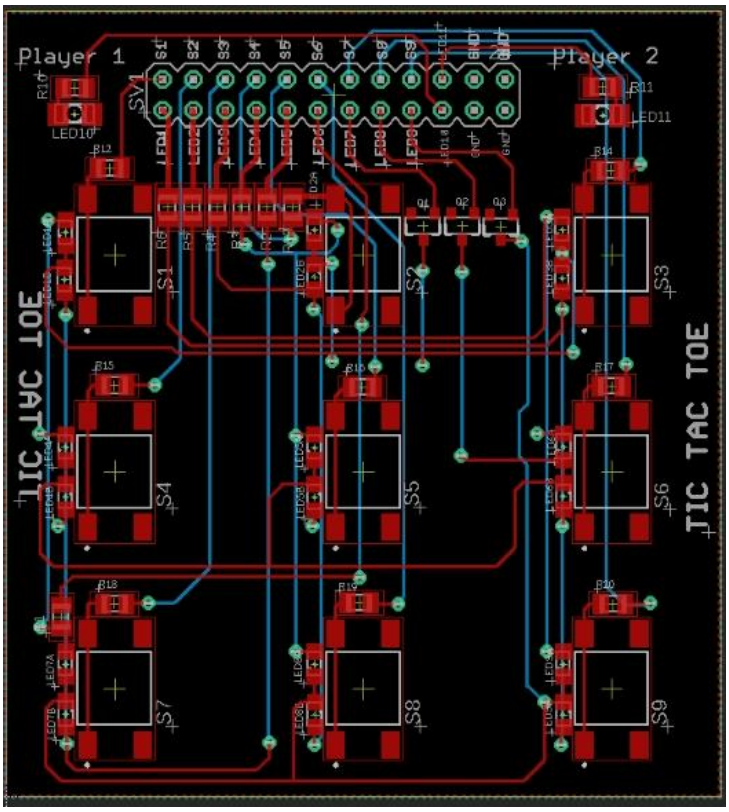
Here is a top view of our pcb design –



Also, the bottom view of the layout –



Also, the schematic view of the board-



8 Sources/References

[1] https://www.fpga4student.com/2017/06/tic-tac-toe-game-in-verilog-and-logisim.html
[2] https://community.element14.com/technologies/fpga-group/b/blog/posts/tic-tac-toeon-fpga-platform
[3] https://digilent.com/blog/tic-tac-toe-with-fpgas/
[4] https://hackaday.io/project/175442-pcb-tic-tac-toe-light-sculptures
[5] https://www.instructables.com/PCB-Visiting-Card-With-Tic-Tac-Toe-Game/