PROJECT REPORT ON

‘CROSS AND NOUGHTS IMPLEMENTATION ON FPGA’

SUBMITTED BY

3844 Ojasvi Tummala

3850 Shraddha Raskar

3851 Preeti Raut

3853 Aparajita Sarkar

3860 Smridhi Dhingra

UNDER THE GUIDANCE OF

PROF. R. SURYAWANSHI

DEPARTMENT OF

ELECTRONICS AND TELECOMMUNICATION ENGINEERING

MKSSS’s

Cummins College of Engineering for Women, Pune

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

(2022-2023)

Table of contents :

Abstract

Cross and Noughts is a staple pastime for people to play with friends when they are bored. The project covers registers, decoders, multiplexers, addresses, and time with clocks. This design is of a 2-player Cross and Noughts game.

The general idea was for our group to recreate the board game using a FPGA board and indicate the winner with the use of LEDs. A decoder with enable is used to decide which register to access. Each register corresponds to one spot on the game board.

The game takes input from the switches present in the switch banks on the development board. There are nine switches, one for each position, per player. There are 2 LEDs, each indicating that one of the players has won. To start a new game, input for reset is also taken from the switch bank.

Introduction

Along with the development of electronic technology, especially the significant progress of the field programmable gate array (FPGA) speeds up the development of digital system. It describes each layer from top to bottom according to the system function requirement and with the help of computer aided design which reduces the cost and the size of the system, shortens the developing period so as to improve the efficiency comparing with the traditional "integrated modules with fixed function plus wiring" developing method. Verilog HDL is a hardware description language used in electronic design automation to describe digital and mixed-signal systems such as field-programmable gate arrays and integrated circuits. Verilog is a Dataflow language, unlike procedural computing languages such as BASIC, C, and assembly code, which all run sequentially, one instruction at a time. The key advantage of Verilog when used for systems design is that it allows the behaviour of the required system to be described (modelled) and verified (simulated) before synthesis tools translate the design into real hardware (gates and wires). Another benefit is that Verilog allows the description of a concurrent system (many parts, each with its own sub-behaviour, working together at the same time).

Cross and Noughts is a world-wide popular two-person game, also called Xs and Os or alternatively tic-tac-toe. It is traditionally a pencil-and-paper game for two players, who take turns marking the spaces in a 3×3 grid, usually X going first. The player who succeeds in placing three respective marks in a horizontal, vertical, or diagonal row wins the game. During online investigation, we found the game implemented in many different languages like C, C++, C#, Java and so on but hardly any designed using Verilog. To gain more exposure to the language as well as the FPGA, we have designed a game based on Xilinx FPGA using Verilog.

Literature Survey

No officially published papers exist on this topic. We found a few project reports by college undergraduates but they had used VHDL as their language, while we worked on this topic using Verilog. Though their work was not identical to what we have done, it definitely did guide us regarding the workflow. The 2 reports are listed under references [1] and [2].

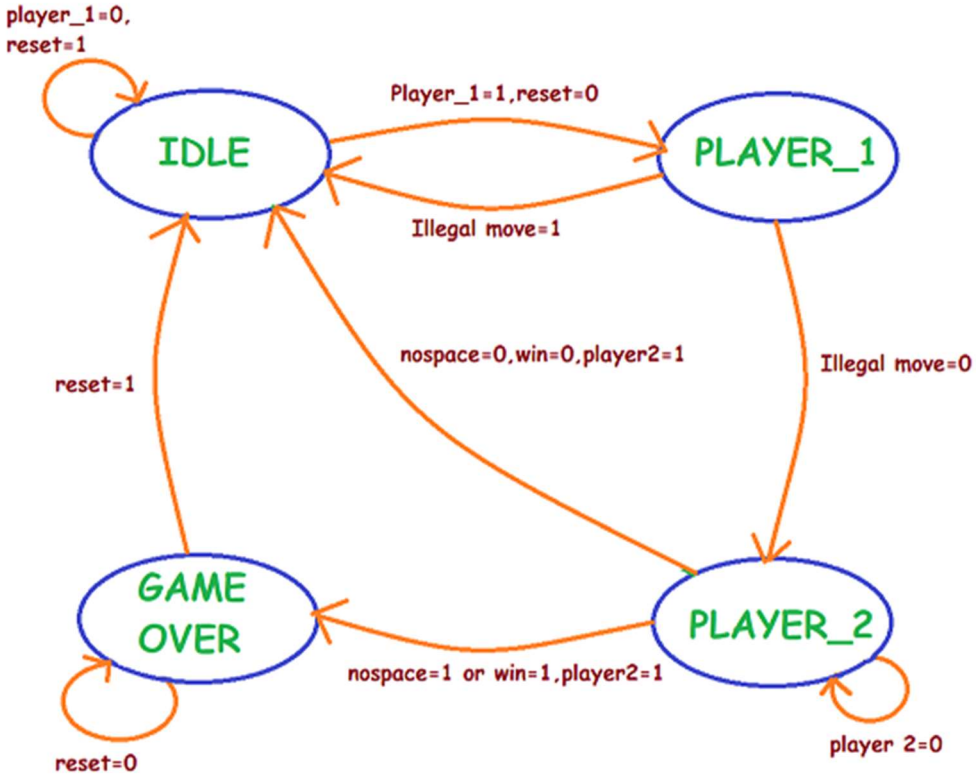
Methodology

1. We wrote the algorithm for the game on the basis of which we went on to write the code. The code was written in Verilog hardware description language.

2. We synthesized it using Xilinx synthesis tool and implemented it in Xilinx ISE (Integrated Synthesis Environment). Parallelly, we created the circuit design on Logisim using components like registers, decoders, multiplexers, addresses, and time with clocks.

3. The .bit file generated by is loaded onto XC 3S 250 E, a Field Programmable Gate Array by the manufacturer Xilinx. The game is operated completely using switches and all output signals are conveyed by the use of LEDs.

State Diagram



State Description

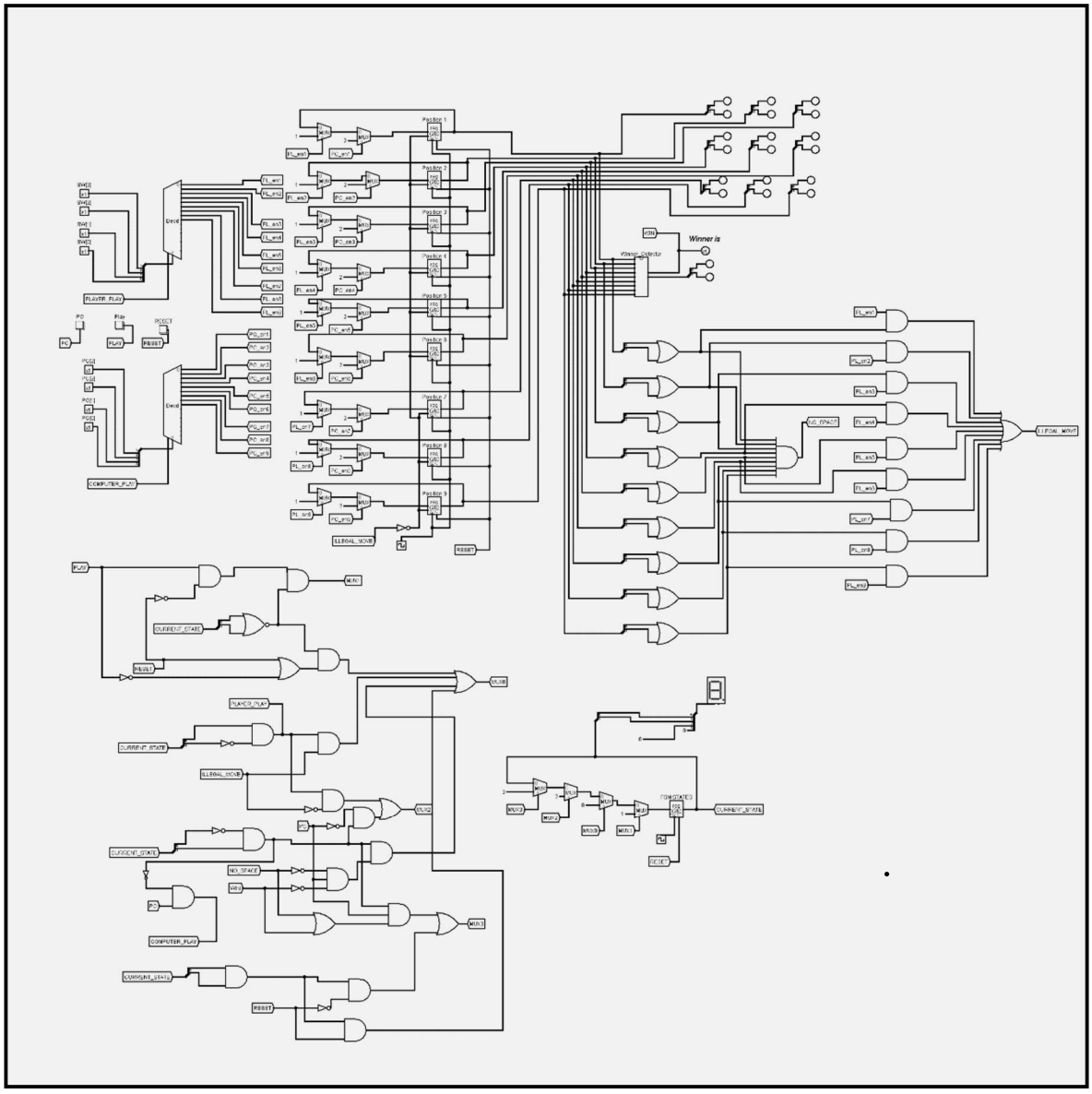
IDLE	**00**	When waiting for the player 1/ player 2 to play or when resetting the circuit, the FSM is at the IDLE state.
PLAYER_1	**01**	The player 1 turns to play and "01" to be stored into the decoded position.
PLAYER_2	**10**	The player 2 turns to play and "10" to be stored into the decoded position.
GAME OVER	**11**	The game is finished when there is a winner or no more space to play.

Reset	**0**	The game begins.
	1	Reset the game when in the GAME_OVER state.
Player 1	**0**	Stay in the IDLE state.
	1	When in the IDLE state, player 1 = 1 is to switch the controller to the PLAYER_1 state and the player 1 plays.
Player 2	**0**	stay in player 2 state
	1	When in PLAYER_2 state, player = 1 is to switch to the IDLE state and the player 2 plays.
Illegal Move	**0**	When in Player_ state, Illegal_move = 0 is to switch to Player_ state and let player plays when player 2= 1.
	1	Illegal moving from the player 1/player 2 and switch to the IDLE state.
No Space	**0**	still have space to play, continue the game.
	1	no more space to play, game over, and need to reset the game before playing again.
Win	**0**	Still waiting for the winner
	1	There is a winner, finish the game, and need to reset the game before playing again.

Position Matrix:-

1	2	3
4	5	6
7	8	9

Circuit Diagram



Cost

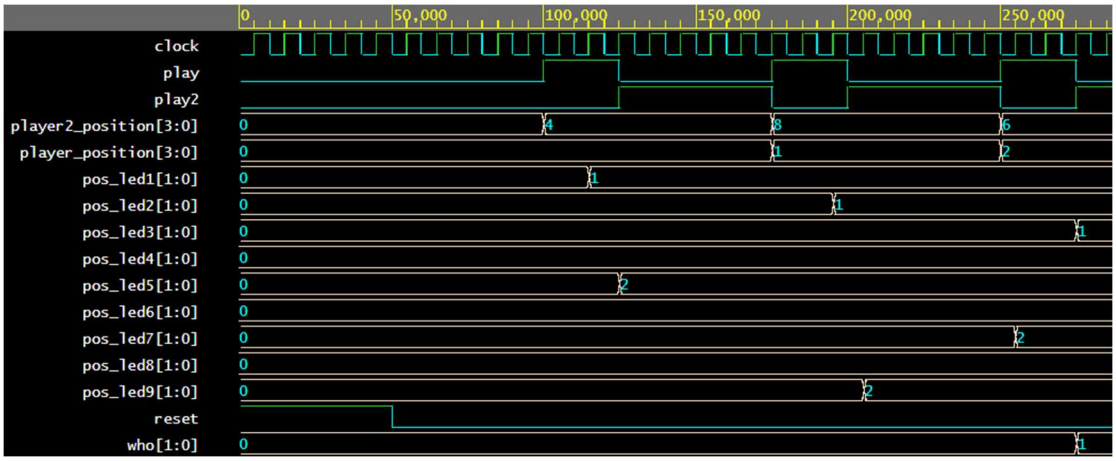
1. XC3S250E FPGA by itself is listed at ₹3300. For our application, we needed a development board consisting of switch banks, LEDs, etc in addition to the FPGA that would cost around ₹5500.

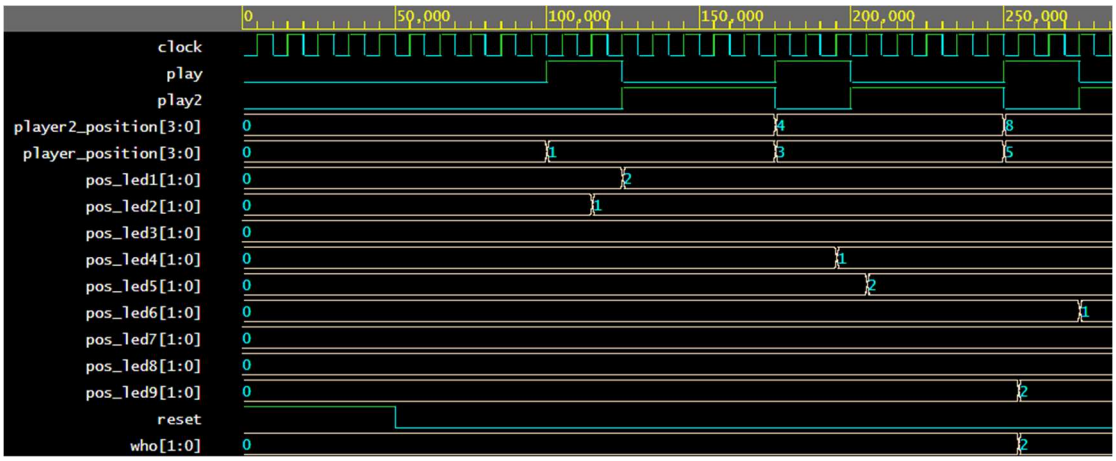
2. **Xilinx ISE (**Integrated Synthesis Environment**)** is a software tool produced by Xilinx for synthesis and analysis of HDL designs, which enables the developer to synthesize their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer. (The board as well as licensed version of the ISE was provided by our college labs, so we did not incur any cost for the same.)

Result/Output

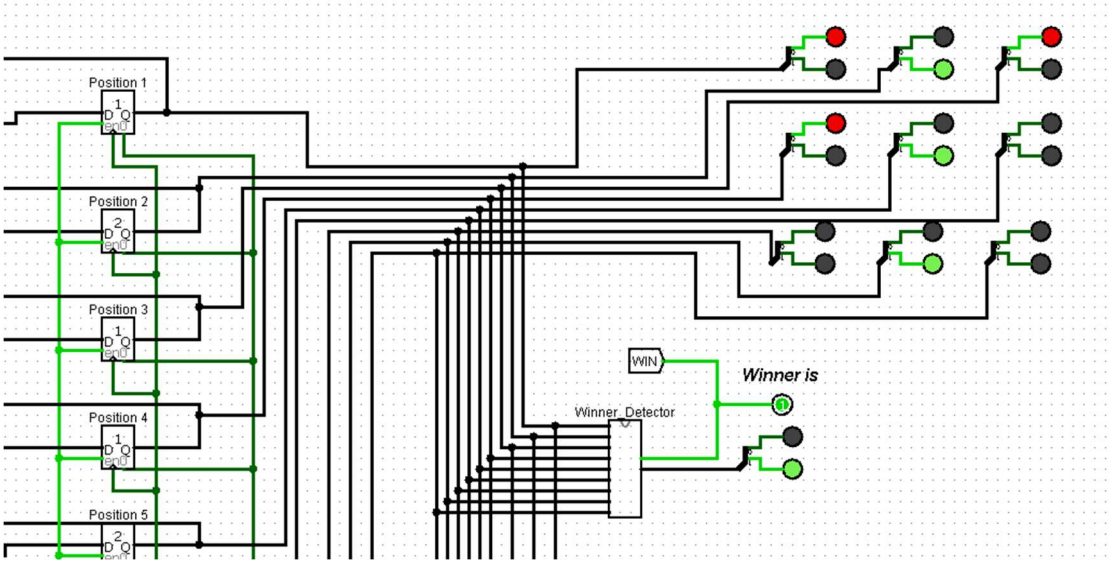
Software Simulation Output:

Verilog Code output:





Logisim output:



Hardware Result:

Observations

The game's behaviour replicated exactly what it intended to do. The outputs were same as what we had designed in the circuit that we built on Logisim circuit simulator.

The game-on LED stayed on as long a round was underway. As soon as all the positions were filled, or one of the players won, it went off. The 2 winner LEDs indicating which player one came on whenever respective player one.

Interpretation

This design defines human-human mode, if expanded to human-computer mode, it is related to Artificial Intelligence, it is rather complex, and might be beyond the computing ability of this FPGA. But it is possible to encode one fixed way of playing into FPGA. Like the first step of the player will have 9 possibilities, then encode the response to these possibilities, they are fixed, one to one, the computer will only follow a fixed way to play instead of making choice. There could be 8 choices corresponding to each 9 different first steps, but it will always choose a fixed way, thus, after the second step, there are still 9 situations. Then next step, it's 7 and so on. It's up to 9*7*5*3*1, i.e., 945 possibilities. In other words, it's like the human player is making decision and has at most up to 945 different choices. If we were to ignore the sequence of Xs and Os, and after eliminating symmetrical outcomes (i.e. rotations and/or reflections of other outcomes), there are only 138 unique outcomes. Assuming once again that X makes the first move every time:
91 unique outcomes are won by (X)
44 unique outcomes are won by (O)
3 unique outcomes are drawn

This means that there are138 different outcomes, the total amount of solutions would vary depending on the strategy used, but definitely it will be less than 138. So hard-coding all the possibilities may be an option to convert this design to 1-player rather than 2-player. However, this approach would be quite cumbersome in the coding area.

Conclusion

Using Verilog as our hardware description language and Xilinx synthesis tool and Integrated Synthesis Environment, we have successfully been able to create a 2-player ‘Cross and Noughts’ game.

The game takes input from the switches present in the switch banks on the development board. There are nine switches, one for each position, per player. There are 2 LEDs, each indicating that one of the players has won. To start a new game, input for reset is also taken from the switch bank.

The main take-away from creating this project was to apply the knowledge obtained from the course VLSI Design to a real-life project. The topics used that were learned from the course include decoders, registers, enable, Xilinx ISE with Verilog, FPGA, if statements, and multiplexers.

Future Scope

1. As explained in the inferences section, we could convert this design to 1-player rather than a 2-player game.

2. We can interface a GLCD to display the board as well as to declare the winner instead of using switches and LCDs. This would make the game much more user friendly.

3. We could also add a scoreboard to keep track of each player's wins across multiple rounds played.

References

1. Chi Zhang, School of Information Science, Computer and Electrical Engineering, Halmstad University - Technical report, June 2010 Tic-tac-toe game design based on Xilinx FPGA

2. John Akroush, Nicholas Wheelis, Tao Wang, Electrical and Computer Engineering Department School of Engineering and Computer Science Oakland University, Rochester, MI - FPGA Tic Tac Toe Game

3. http://www.cburch.com/logisim/docs/2.3.0/guide/subcirc/creating.html