“Ultimate Tic-Tac-Toe”

Group 6

Syeda Hoorain, Faheem , Adil, Zainab

December 12, 2024

Table of Contents

Abstract

This project focuses on the development of an enhanced version of the classic Tic-Tac-Toe game, known as Ultimate Tic-Tac-Toe, which integrates digital design concepts into a fully functional, interactive game system. The goal was to create a modular, scalable implementation that synchronizes game logic, player inputs, and visual rendering to provide an engaging user experience.

The game is played on a VGA display, with players interacting through physical buttons for navigation and move placement. The system features a cursor for intuitive player control, a dynamic scoring mechanism, and visual feedback displayed on both VGA display. A splash screen introduces the game, while a central control unit coordinates the game flow, validating moves, managing the board state, and checking for winning conditions.

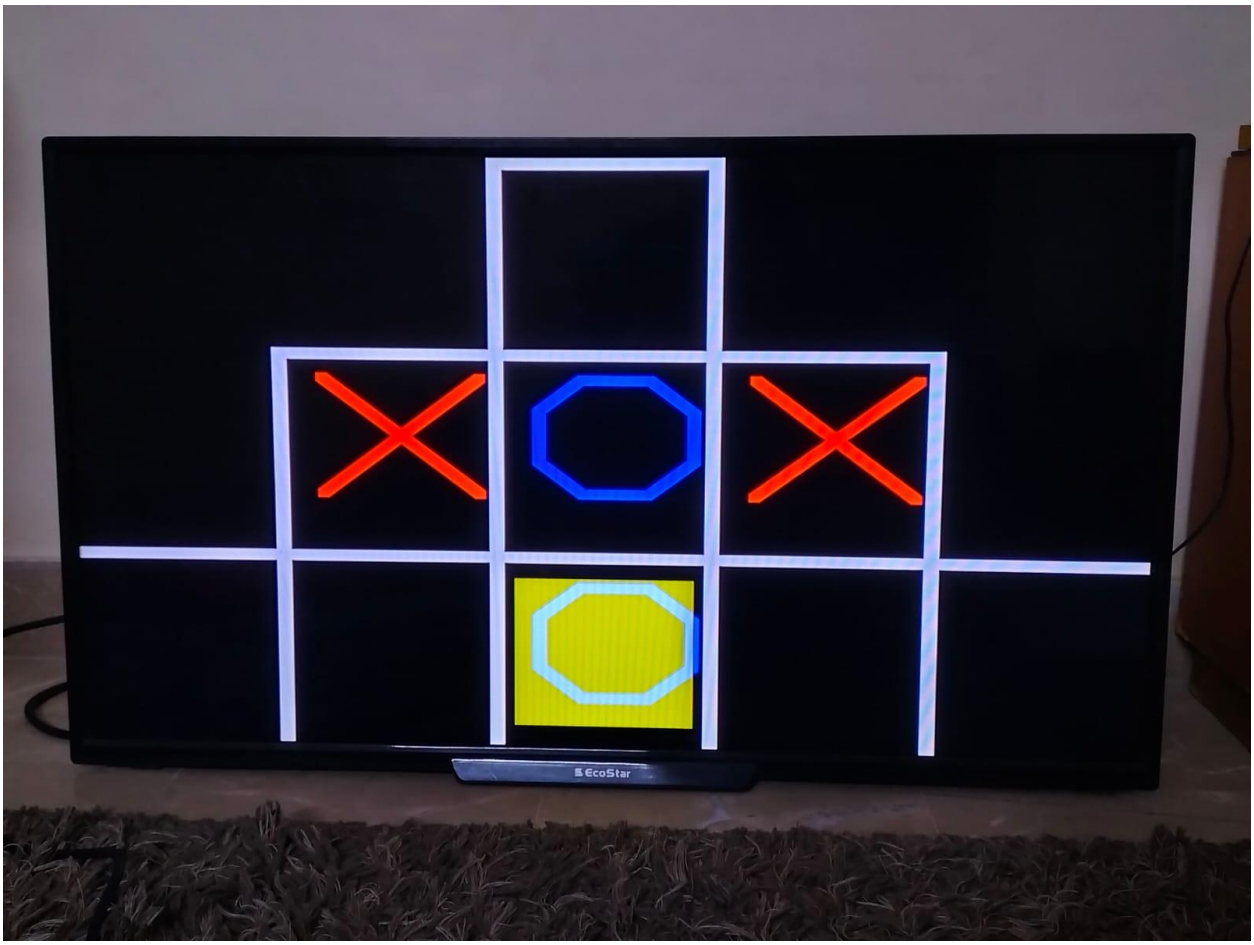
Advanced hardware features include precise input handling via button debouncing, a state-driven rendering system for board visuals and blinking cursor, and winner detection logic. The modular design enabled seamless integration of game components, resulting in a fluid and responsive gaming experience.

Introduction

The inspiration for this project stemmed from the timeless appeal of classic games and the desire to enhance the simplicity of Tic-Tac-Toe with added complexity and interactivity. Ultimate Tic-Tac-Toe combines strategic thinking with real-time digital design, creating an engaging experience for players of all ages.

This project is a virtual implementation of Ultimate Tic-Tac-Toe, designed for two players to compete in a dynamic and visually appealing environment. The game introduces a modern twist to the traditional Tic-Tac-Toe by incorporating advanced game mechanics, such as a Pyramid board structure integrated into a cohesive digital design framework.

The overall system comprises three main components: the user interface, the control unit, and the rendering block. The user interface provides intuitive controls for player interactions, the control unit manages game logic and decision-making, and the rendering block handles visual outputs on the VGA display.



User Interface

The user interface allows players to interact with the Ultimate Tic-Tac-Toe game seamlessly. It provides intuitive controls for navigating the game board and making moves. The interface is designed to ensure responsiveness and accuracy during gameplay.

The primary components of the user interface are the **PS/2 keyboard** and the **VGA display**. The keyboard now replaces the push buttons and switches, allowing players to use the arrow keys for navigating the board (up, down, left, and right) and the Enter key to confirm their moves. Additional keys may be mapped to reset the game or toggle other settings, enhancing

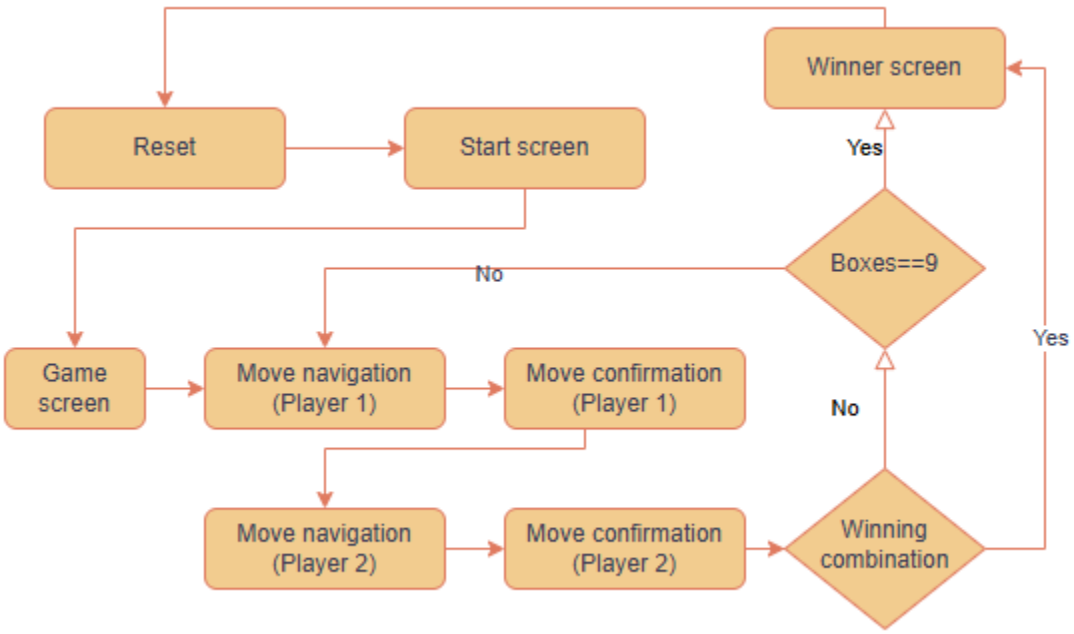
accessibility and efficiency for the players. The use of the keyboard eliminates the need for push buttons and switches, streamlining the hardware requirements.

To ensure smooth interaction, a **Keyboard Debouncer** has been implemented. This module processes keyboard inputs, filters out unintended signals, and guarantees accurate detection of keypress events. The debouncer also handles key hold and release events to prevent repetitive or noisy inputs.

The **VGA display** serves as a visual interface, presenting the game board, cursor position, and any additional visual feedback such as scoring or splash screens. The display dynamically updates to reflect the player's actions, ensuring clarity and engagement during gameplay.

Together, the keyboard and VGA display form an efficient, accessible, and user-friendly system for controlling the game. This updated interface makes the game more intuitive and easier to play while offering improved interaction capabilities compared to the previous FPGA push-button system.

User Interface Diagram



Playing Instructions

Player 1 and Player 2 can move in 4 directions using the designated keyboard keys.

1. **General Controls:**
    a. Space Bar (Reset) => GAME SCREEN to renewed empty GAME SCREEN (if there is an ongoing game or game is over)
    b. Space => START SCREEN to GAME SCREEN (when game is started in order to start the game.)
    c. Enter => Move confirmation. (Marks block in grid with players sign)
2. **Player Controls:**
    a. W key => Move up
    b. S key => Move down

c. A key => Move Left

d. D key => Move right

Control Unit

The Control Unit (CU) is the heart of the Ultimate Tic-Tac-Toe system, responsible for managing game logic, player inputs, and interactions between modules. The CU processes player moves, updates the game state, and communicates with the display to provide real-time feedback. It integrates inputs from the player controls and determines valid moves based on the game's rules, including the transition between regular Tic-Tac-Toe boards and the overarching game board.

The CU is a finite state machine (FSM) with distinct states for various game modes, including initialization, move selection, board updates, and victory checks. Initially, the CU begins in an idle state awaiting player input. Once a move is made, the CU validates it by checking the availability of the targeted cell and ensures compliance with Ultimate Tic-Tac-Toe rules, such as directing the next move based on the previous move's position. If valid, the CU updates the sub-board and overall board states and transitions back to the idle state for the next turn. In the case of a completed game (either at the sub-board or overall board level), the CU transitions to the victory state to determine the winner and reset the game as needed.

The CU integrates with several minor FSMs to handle specific tasks, such as starting the game, main game, and checking for wins. These FSMs streamline the system's efficiency, ensuring accurate and fast game logic processing.

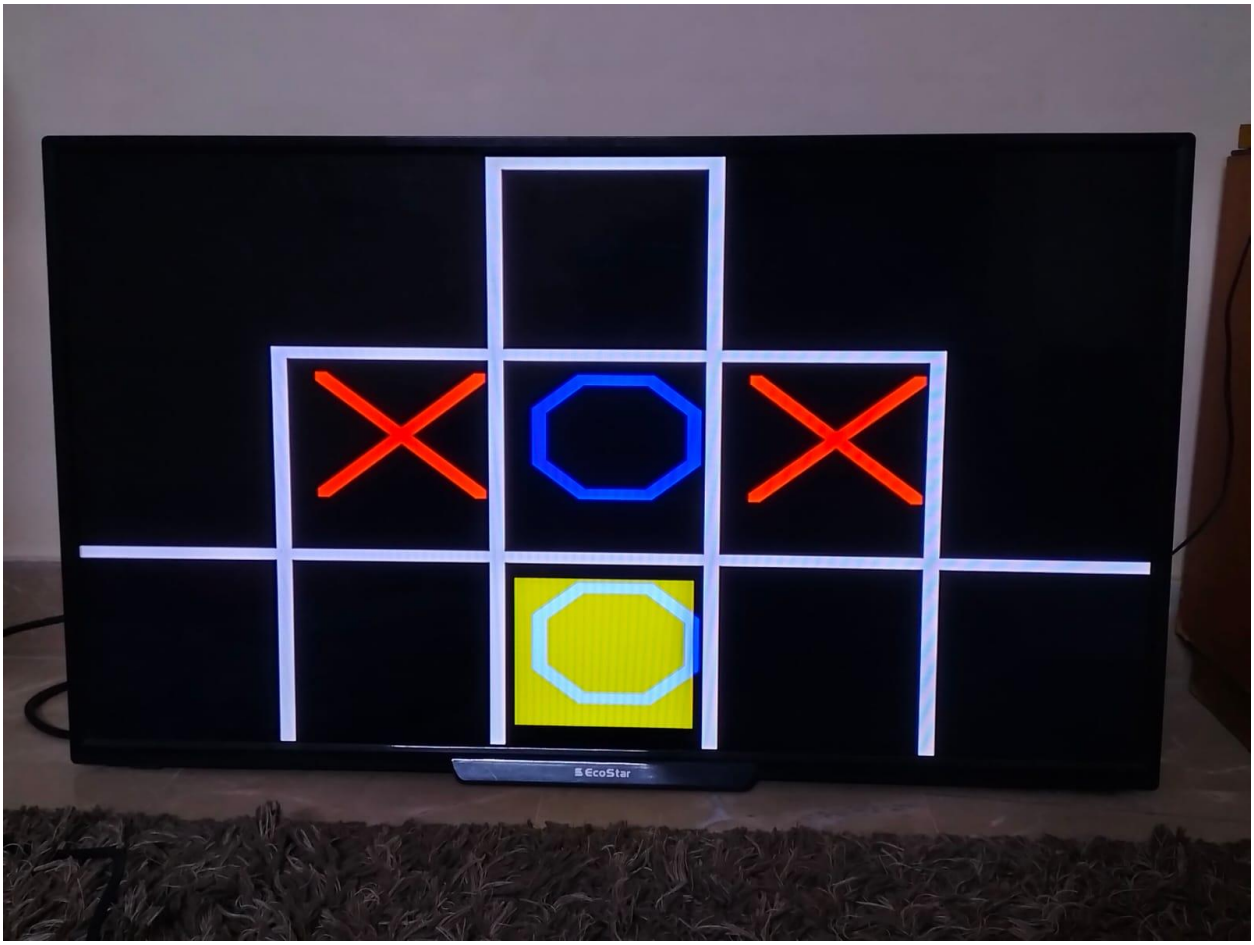
1. **Start Screen FSM**

The Start Screen FSM is activated at the start of the game and can also be considered as idle state. It indicates that the game is about to start. If the space bar is pressed, the FSM signals the CU to start the game board and display the game screen.

2. **Main Game**

The Main Game starts when the space bar is pressed which signals the CU to start the game board and display the game screen. It then allows navigation across the game board grid. It

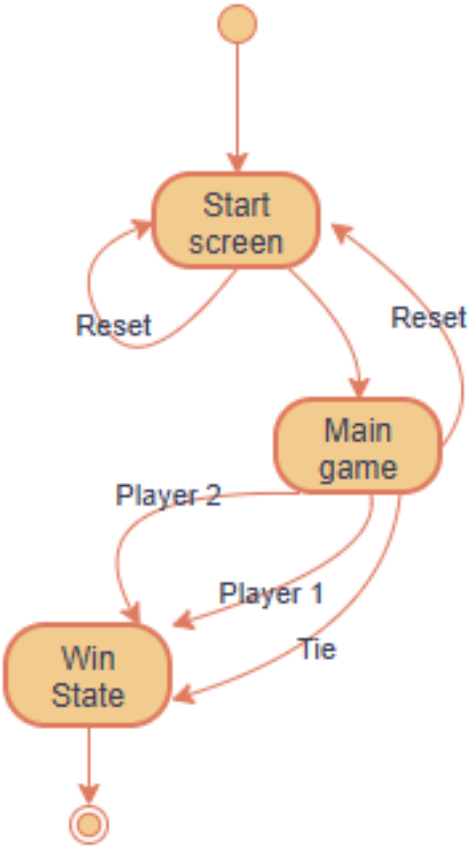
assigns the appropriate player's marker to the targeted cell and checks if a player has won.



3. Victory Check FSM

The Victory Check FSM is responsible for determining if either player has won the game. It checks for winning combinations on both the sub-board and overall board, including rows, columns, diagonals, and the entire grid. If a win is detected, the FSM signals the CU to transition to the victory state, displaying the winner and resetting the game for future rounds.

FSM Diagram



Inputs for Tic Tac Toe

1. **FPGA Keyboard Inputs (PS/2 Receiver and Keyboard Debouncer)**

   The updated input system utilizes a PS/2 keyboard interface to replace the previous FPGA push-button input. This change provides a more intuitive and accessible method for players to interact with the game. The PS/2 Receiver module is responsible for decoding the serial data stream from the keyboard, converting it into usable keycodes that correspond to specific game actions.

The Keyboard Debouncer ensures that only valid and stable keypress signals are passed to the game logic. It eliminates noise and prevents unintended multiple keypress detections by implementing a finite state machine (FSM) that checks for stable transitions in the input signal.

Key features of the updated modules include:

a. PS/2 Receiver:
    i. Captures and decodes serial input from the keyboard.
    ii. Converts keypress data into binary keycodes.
    iii. Maps keycodes to specific game actions, such as marking a grid position or restarting the game.
b. Keyboard Debouncer:
    i. Stabilizes the input signal from the PS/2 Receiver.
    ii. Implements a delay mechanism to filter out spurious signals.
    iii. Ensures that only one keypress is registered per action.

This new system enhances the user experience by allowing players to use a standard keyboard to play the game. It also simplifies the hardware setup by removing the dependency on FPGA push buttons.

Imaging for Tic Tac Toe

1. **VGA Module**

The imaging for Tic Tac Toe relies on a VGA module to drive the visual display. The VGA module is configured for a 640 x 480 resolution with a 60 Hz frame rate, ensuring smooth and clear visuals. This module generates the horizontal and vertical synchro

nization signals, along with the blank signals required for the display. These signals are sent to the ADV7125 Digital-to-Analog Converter (DAC), which produces the RGB outputs to illuminate the screen.

The VGA module tracks pixel and line counts, mapping them to specific on-screen coordinates. At the end of a frame (when the pixel count reaches 639 and the line count reaches 479), a frame pulse is generated to reset the counters. The module uses active video, front porch, sync, and back porch periods for both horizontal and vertical blanking intervals to ensure proper display timing.

The display operates on a clock signal, with pixel counts incrementing on each rising edge. When the pixel count or line count reaches predefined thresholds, the module transitions between active video and blanking intervals, maintaining synchronization. These precise timings ensure that the Tic Tac Toe grid, game markers, and other visual elements are displayed accurately and without flicker.

2. Display Field Module

The Display Field module is responsible for rendering the Tic Tac Toe grid, player markers (X and O), and other visual elements on the screen. It outputs 8-bit RGB signals to the DAC, which drive the display colors. The module receives inputs such as the pixel

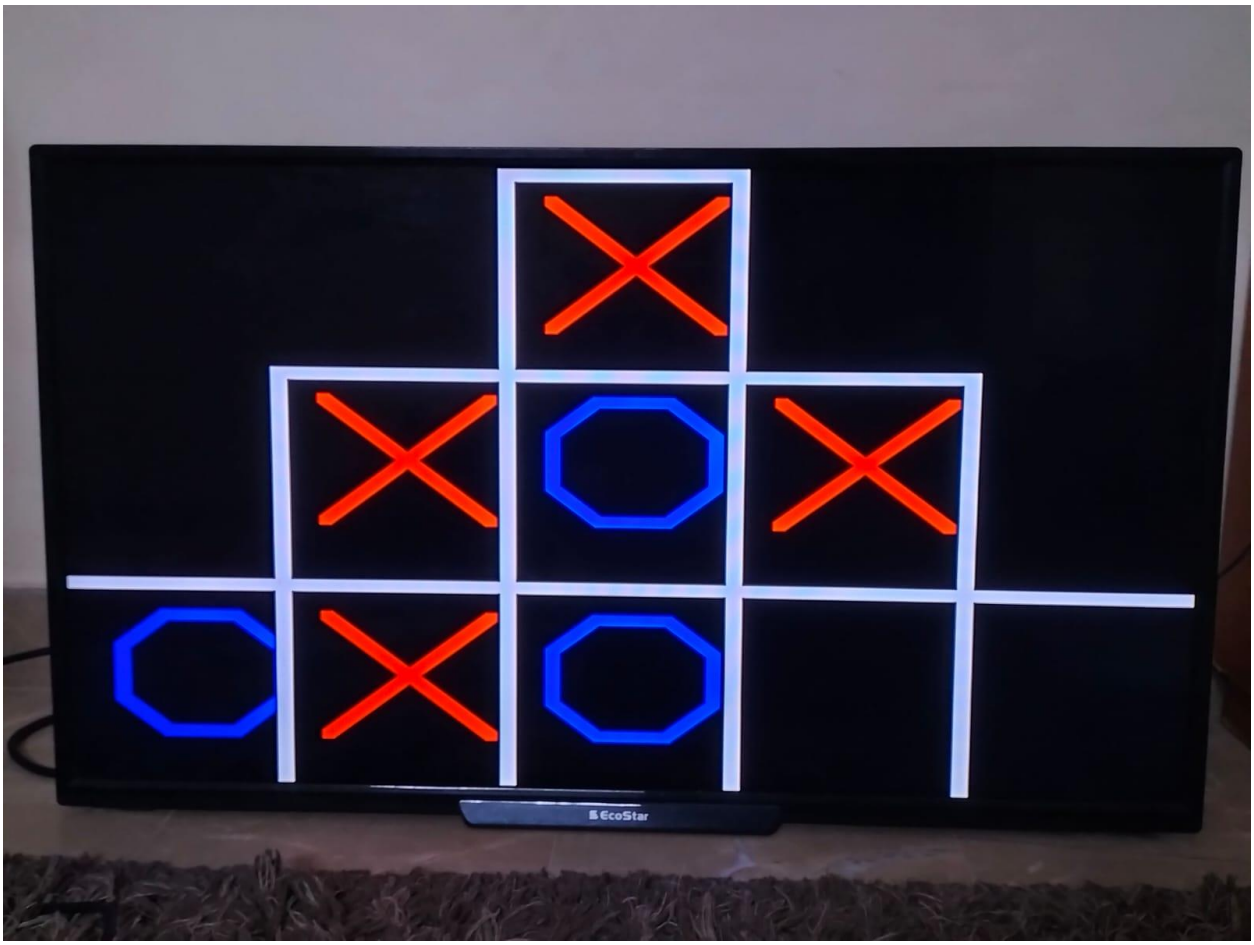
clock, frame pulse, pixel count, and line count from the VGA module, as well as game state data from the Control Unit.

The grid and markers are drawn using submodules, each corresponding to a specific game element. These submodules take the pixel and line counts as inputs and output the color values for the corresponding on-screen pixels. For example:

a. The grid submodule determines the boundaries of the 3x3 game grid and sets the pixel colors accordingly.
b. Marker submodules check the game state to determine whether to draw an X or an O in a specific grid cell.

The Display Field module determines which submodule outputs to use based on the current game state. During the game mode, it renders the grid and player markers. In other modes (e.g., start or victory screens), it outputs different visuals such as text or animations. By structuring the rendering logic into submodules, the system remains modular and easy to expand for additional features or visual enhancements.

This setup ensures that the Tic Tac Toe game is visually appealing, with a responsive and clear user interface. The modular design also allows for easy customization and scaling to different display resolutions or additional game features.



3. **Winning Screen**

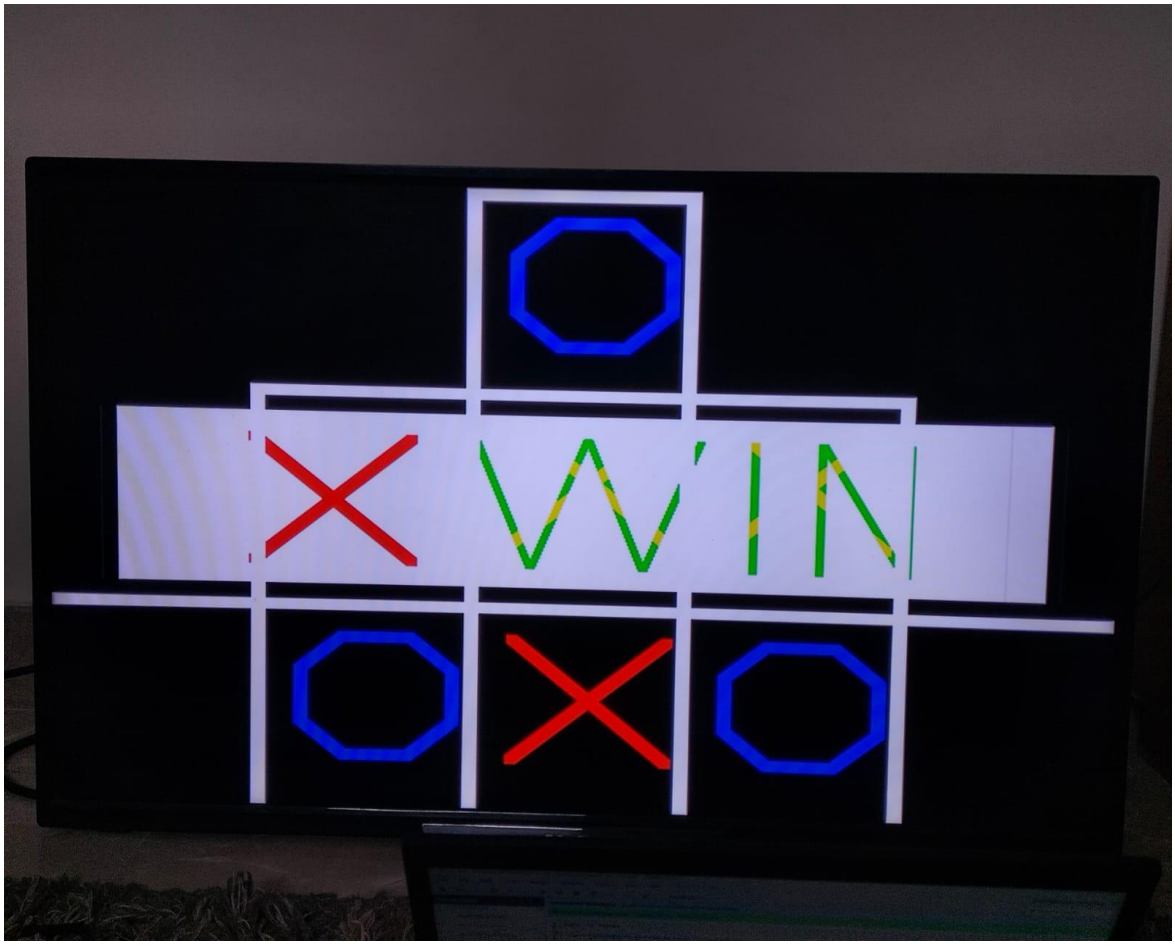
A dedicated winning screen is designed to display the victory of either player when the game concludes. Once the game logic determines a winner (Player X or Player O), the Control Unit triggers the winning screen module. This module overrides the regular game visuals, transitioning the display to a celebratory screen.

The winning screen includes the player's symbol (X or O) prominently displayed in the center of the screen, accompanied by a congratulatory message such as "X Wins!" or "O Wins!" The Display Field module adjusts the RGB output to include vibrant animations or effects, such as flashing colors or pulsing text, to highlight the victory. Additionally,

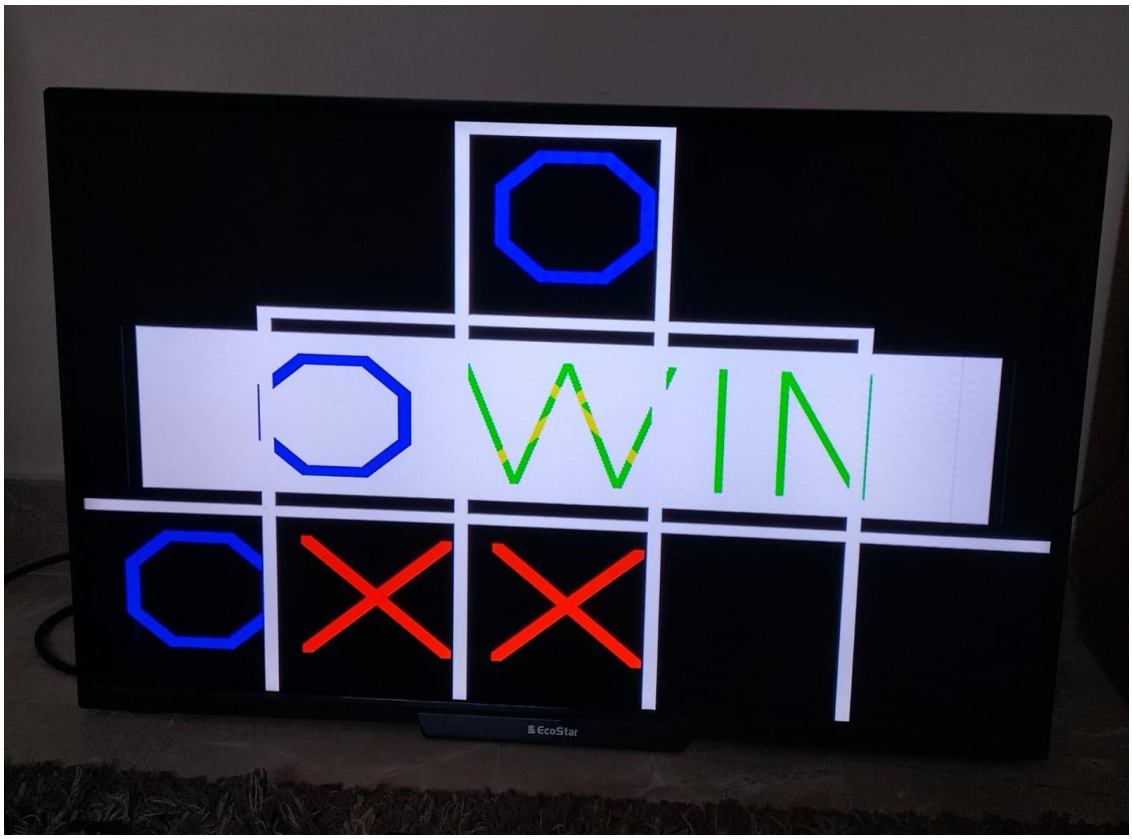
the screen includes an option to restart the game or return to the main menu, controlled via the input system.

The modular nature of the winning screen allows for further customization, such as adding player-specific themes, sound effects, or enhanced animations to make the victory more engaging.

a. X win screen



b. O win screen


WIN
EcoStar

c. Tie screen