

EECS3216 Project Proposal

Allen Kaplan (215494925)
Stefan Sion (215597701)
Elie Frungorts (215659501)
Adam Silverman (215415318)

Email: allenkap@my.yorku.ca
Email: stef99@my.yorku.ca
Email: eliefr@my.yorku.ca
Email: ajs0429@my.yorku.ca

Project Problem

The problem that we are looking to create a solution for will be creating an implementation of *tic tac toe*. The game must be able to support all rules and operations of *tic tac toe*. To ensure the game is playable, the project must be able to handle user inputs of the column and row which users wish to place their move, the project must handle two users (blue and red) and the project must be able to display the current state of the game.

Use of Combinational and Sequential Logic

Combinatorial

User Input Function - Create function to store memory of user input on board (3x3 matrix)

Winning Algorithm - Create algorithm to detect when either user 1 or 2 wins

Sequential

Clock - create clock system to display using VGA interface for HSYNC and VSYNC

Looping of winning algorithm - dynamic checking of game outcome

User Input

Number Pad

Use 8-bit numpad to accept user position:

1-9 for 9 possible placements, enter pound for confirmation of position

Requirements

1. Handle user input to select location to drop piece

User Input will use either switches or a 10-key number pad to select where to place their move.

2. Change turn and display the current user which has their turn {blue, red}

With every user input, the turn will move to the 2nd user input. user's turn is currently in action on the seven segment display

3. Maintaining state of the board

Use registers to keep track of where pieces are and which user claimed {none, blue, red} → We will need 9 registers to represent 3 columns x 3 rows.

4. Detect and handle if win conditions met

Project will require to check if a user has won; if a user has won, we will keep track of score

5. Output current state to display

Display the board on a screen (VGA output): Create 2 nested loops for VSYNC and HSYNC, define monitor resolution and split the screen up into square segments that change with user input.