

CS223 Laboratory Project

Game of Nim

Groups: Each student will do the project individually. Group size = 1

Important dates:

Demo presentation day and final report: 7/5/2018 during lab times of each section

The Ancient and Honorable Game of Nim

Description:

Nim is an ancient game with several variations. Nim starts with a group of heaps, each of which contains one or more matchsticks. Two players takes turns removing matchsticks from a heap.

In this version of Nim the heap of matchsticks will be sequenced within 4 rows and the rows will contain 1, 3, 5 and 7 matchsticks respectively.

On each turn, the player must remove **at least one** matchstick. Once a player starts removing the matchstick from a row, that player can only continue to remove matchsticks through the **same row**.

The player who is forced to remove **the last matchstick loses**.

8x8 RGB LED Display will be used in order to represent the matchsticks. Two LEDs one over the other will simulate a matchstick. By using 8x8 RGB LED matrixes 4 rows of matchsticks will be simulated. In order to distinguish the rows red and blue colored LEDs will be used respectively. The initial matchstick placement and colors will be as follows.

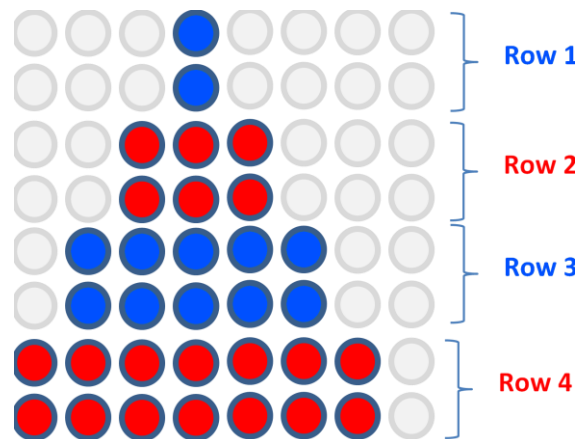


Figure 1: Initial matchstick placement

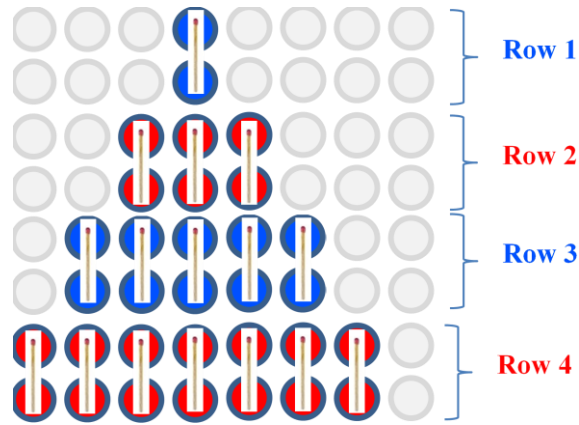


Figure 2: Initial placement with matchstick simulation

Starting with a total number of 4 rows, red and blue colors will be used to separate the lines. Each matchstick to be used in the game will be represented by two LEDs which will be driven in the vertical axis (Figure 2).

Initial state of the score board will show 0 for both players. (Figure 3)



Figure 3: Initial value of score table

The game starts from the player on the left side. As the initial position stepper motor will show the left side of the card with an arrow to be stuck on it. Stepper motor will show the active player throughout the whole game. (Figure 4)

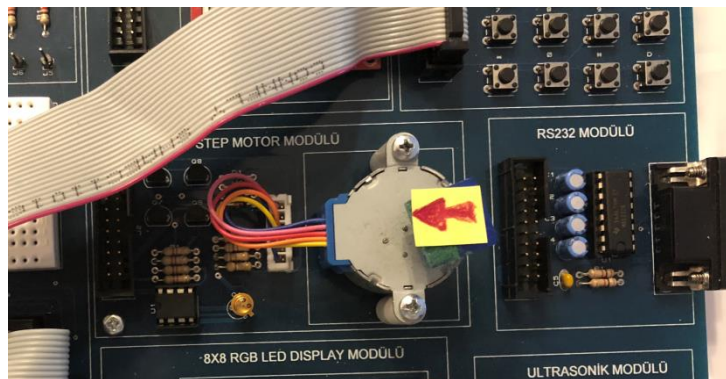


Figure 4: Initial position of stepper motor

Game will be played by using the Basys3's keys (switches and pushbuttons).

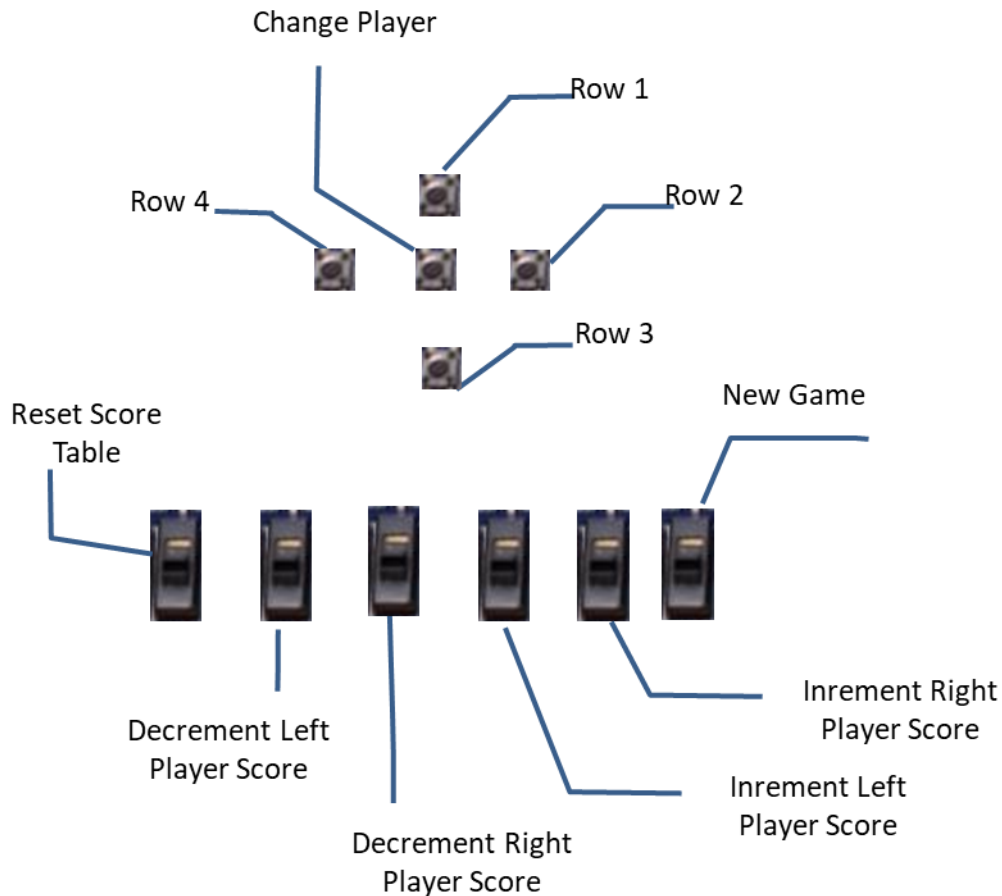


Figure 5: Functions of the buttons-slide switches

Button/Slide Switch Function Definitions:

Change Player: Changes the active player from left to right or right to left. When pressed stepper motor position will change.

Row[1..4]: Will be used to collect matchsticks one by one on the corresponding row.

New Game: Initiates matchstick placement on 8x8 RGB LED matrix, initiates stepper motor position to show left player. Level change of the slide switch from HIGH to LOW will initiate this function.

Reset Score Table: Will reset the score table to 0 for both players. Level change of the slide switch from HIGH to LOW will initiate this function.

Increment left player score: Will increment left player score displaying on the left most two digits of the seven segment display. The score will roll to 00 after 99. Level change of the slide switch from HIGH to LOW will initiate this function.

Increment right player score: Will increment right player score displaying on the right most two digits of the seven segment display. The score will roll to 00 after 99. Level change of the slide switch from HIGH to LOW will initiate this function.

Decrement left player score: Will decrement left player score displaying on the left most two digits of the seven segment display. The score will roll to 99 after 00. Level change of the slide switch from HIGH to LOW will initiate this function.

Decrement right player score: Will decrement right player score displaying on the right most two digits of the seven segment display. The score will roll to 99 after 00. Level change of the slide switch from HIGH to LOW will initiate this function.

Game Starts:

When the player presses one of the row pushbuttons (1..4), the matchstick placed on the right most side of the corresponding row will be taken off (both LEDs representing a matchstick will be OFF). If the player starts to remove a matchstick from a row and finishes all the matchsticks within that row, turn will switch to the other player and stepper motor will show the other player. Player can only use the row button **after the stepper motor finishes turning**. When the player who is playing is finished with taking the matchsticks, he must press the "Change Player" button to switch to next player. The player who is forced to **remove the last matchstick loses**.

The player who wins the game will be indicated by the stepper motor, it will show the side of the winning player. Winner of the game will be shown both with the stepper motor and the 8x8 RGB LED matrix as given in Figure 6 and 7.

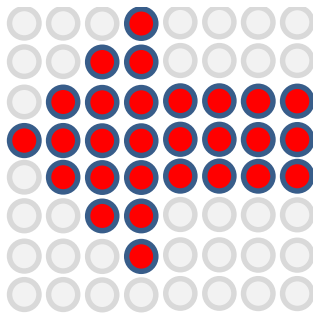


Figure 6: Winner is left player

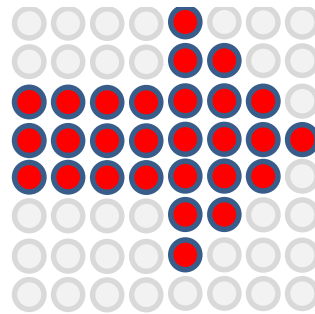


Figure 7: Winner is right player

The seven segment display on the BASYS3 board will be used to show the score. The left two segments of the display will be used for "left player" score and right two segments will be used for right player score. Score of the winning player will be increased by 1 automatically.

Regardless of the state of the game, score table can be modified, each player's scores can be incremented or decremented. The score will circulate to 00 after 99, and will circulate to 99 in case decrement switch changes level from HIGH to LOW.

Regardless of the state of the game, if "new game" switch changes level from HIGH to LOW, 8x8 LED matrix and stepper motor will be taken to initial positions.

For doing the project, some of the modules that you need are already given to you and you do not need to implement or even modify them. It is enough to just use them as black-box in your design and connect them to other part of the design by writing your own SystemVerilog codes.

References: https://www.archimedes-lab.org/game_nim/play_nim_game.html

Notes:

1. You have been given some ready modules. You should use them directly without editing them. It is better to test them first to be sure that your Beti board has no problem.

2. Study Basys3 and Beti board documents carefully to avoid connecting different wires to each other and damaging your Basys3. Specifically, read `Basys3_converter_board.pdf` file on Unilica. Do not connect two modules at the same time using parallel cables. Use cable for one Module (like 8x8 RGB display) and for others use wires. You can also use two unused ports on the left-hand side of your Basys3.