Snake
lab 4
m152a
Group 3

William Randall
Christian Loanzon
Austin Keil

Overview:

Our project will be a recreation of the game “Snake” on an FPGA. The board’s push buttons will be used to control the player’s snake on the screen, while the state of the game itself will be displayed on a monitor through VGA output from the board. The circuit will also feature a reset button to return the board to its initial state. The player’s score will also be displayed in binary on the LEDs of the board.

Game Mode:

The player will control a dot (also denoted as the ‘head’ of the snake) on a square plane with bordering walls. This dot will constantly be traveling in a certain direction based on the player’s push-button inputs (upon initialization, the dot will stay still until the player’s first input). By using the board’s push buttons in the four main directions (up, down, left, and right), the dot will change its path to follow the same direction assuming that the inputted direction is not already along the dot’s current path (e.g. an input of up or down when the dot is already traveling downward will not change anything, but a button input of left will cause the dot to change its direction to left).

When the dot moves, a trail will be left behind with a certain initial length, resembling a moving snake. If the head of the player’s ‘snake’ (the moving dot) contacts this trail or the bordering walls of the plane, the player loses the game, resetting the game to its initial state (of a static dot with no direction). We will display one dot at a time and when the snake’s head hits the dot we will then increment the player’s score and display another dot at a random free point on the board.

To add complexity to this game, the overall goal of *Snake* is to collect smaller dots that gradually spawn into the plane. Upon collecting a dot by contacting it with the head of the snake, the player’s score will increment by 1 and the length of the trail will then extend (thus making the snake longer). This current score will be reflected on the LEDs on the board. The player ‘wins’ the game by collecting enough dots for the snake to be long enough to fill up the entire plane with its body (e.g. there will be no blank spaces in the square plane).

Extras:

Beyond the actual mechanics of the game, we will be adding graphics and sound to the game. We will output the state of the game, drawing the snake as a sequence of dots, while playing noises upon changing the snake’s direction, collecting a dot, and dying.

Components:

VGA, Speaker, 4 Push buttons, Leds

Grading Rubric:

- 40% VGA functionality
    - 10% Snake Movement
        - The snake will be able to move up, down, left, and right. It should not be able to do a full 180 degree turn in one clock tick.
    - 10% Dot pickup
        - The snake should be able to pick up dots and increase its score.
    - 10% End of game when touching wall
        - If the snake touches the edges of the display then the snake will die and the game will restart.
    - 10% End of game when touching snake itself
        - If the snake touches itself (goes in a loop) the game should end and then restart.
- 20% Sound functionality
    - Every time the game begins a certain begin sound will play. Every time a dot is picked up a sound will play. Every time the snake dies a sound will play.
- 15% Button Debouncing and controls
    - We need to implement the debouncing to the 4 controls and link the controls to the movement of the snake.
- 5% Reset functionality
    - When the reset button is pressed, the game will restart and wait for user input.
- 20% Scoring functionality / display on board in binary
    - When the snake goes over the dots the score should increase and that score should display in binary on the LEDs on the board.