CS M152A/EE M116L: Extra Final Project
Georgi Baghdasaryan

SNAKE GAME

Overview:

I (Georgi) decided to dedicate some of my free time (since I am so passionate about digital design) to working on an extra final project for this class. I am going to create a Snake game that will be running on Spartan 3e FPGA board. To accomplish this task, I will utilize Xilinx ISE software for the game design (vhdl, schematic, user constraint and other files). Also, in order to allow users to play this game, I am going to utilize a display via FPGA's VGA port. This will be accomplished to provide players with a full visual representation of the game. More details about this game can be found below, in *Gameplay* and *Specifications* sections.

Gameplay:

The snake game works as follows: a player controls a long, thin creature, resembling a snake, which roams around on a bordered plane, picking up food (an apple), and trying to avoid hitting the "walls" that surround the playing area. Each time the snake eats a piece of food, its tail grows longer and the speed increases, making the game increasingly difficult. The user controls the direction of the snake's head (up, down, left, or right), and the snake's body follows. Player cannot stop the snake from moving, while the game is in progress, without losing (resetting the game).

Specifications:

In this project, I implemented the snake as a green rectangle, capable of moving in horizontal and vertical directions. After the program is loaded this snake in the middle of the screen combined with borders on screen edges is what a user can see. The game starts when a user switches the leftmost switch to location '1'. This causes the initial score to become 00 and snake to start moving right at a speed of 2 pixels. User is supposed to use A, W, S, and D keys of the controller (keyboard) to navigate the snake towards a red square representing an apple. After acquiring an apple, snake's speed increases by 1 pixel, which makes it both go fast as well as seem to get longer. Also, this causes the score counter (2 rightmost 7-segment LEDs) to increase by 1. The interesting part about apples is that each of them is generated at a random location (using a clock based random number generator written in VHDL for x and y positions) either right after the game starts or after one was eaten (by the snake). The game continues until a user either acquires 20 apples or hits a border of the screen (represented by blue color).

There are three ways for ending the game: hit a border (loose), reach score of 20 (win), and switch to reset mode (using the switch). If the user resets the game then all the counters are set to zero and the initial screen (snake and borders) is shown. Otherwise, when the game ends, the user sees a black screen with corresponding text (either "Game Over!" or "You Won!") and hears either a losing or a winning sound. Whenever a game reaches its end (not via reset mode), a corresponding sound can be heard from the speaker as well as the score displays can be seen blinking with a final score that the player achieved.

Since no one was yet able to achieve the highest score in this game, the rightmost switch was programmed to provide the player with a similar screen as the one when the game is passed (not the same though). In other words, turning on this switch will produce a winning sound as well as show a winning screen.

Details about input/output that the system can receive/provide are given in the tables below.

Board Inputs	**Details**
50MHz clock (pin T9)	Will be used for snake movement, and system updates
Keyboard 'A' key	Used for changing snake's movement direction to LEFT
Keyboard 'D' key	Used for changing snake's movement direction to RIGHT
Keyboard 'W' key	Used for changing snake's movement direction to UP
Keyboard 'S' key	Used for changing snake's movement direction to DOWN
Leftmost switch	On(1)/Off(0) (used for reset)
Rightmost switch	Show winning screen (in case if you want to "cheat")

Table 1: Board inputs and their details.

Board Outputs	**Details**
Monitor (external)	Used for displaying game to the user (red -> apples, green -> snake, blue -> borders)
7-Segment LED display (onboard)	Used for displaying the score board (number of eaten apples, blinking means end of game)
C10 port on the board's right side	Used for producing sound output

Table 2: Board outputs and their details.

Implementation:

Different methods and files were used to implement this system. Schematic files were used to implement most of the clocks, score counters, top and other modules. Meanwhile VHDL code was used to correctly light up 7-segment LEDs based on the bitwise input from the counters, display the game, and to implement game logic. The input 50MHz clock was used to drive almost all the modules, including the random number generator, display synchronization, counters and others. More implementation details can be obtained from the project files submitted via CourseWeb.

Problems:

Most of the problems I had during the implementation of this game were concerning either with software or with hardware. Specifically, it took me an extremely long time to get the project done, compiling the program via Xilinx ISE used to take 5 minutes each time. Another problem I faced was with the FPGA board hardware, which was giving me an error saying that the board does not have enough components for my design. This was caused by an assignment of snake component at ***i*** to snake component at ***i-1*** (used for snake movement). This problem was fixed by decreasing the length of the snake.

Overall, I can say that this was a very interesting and a fun project. I was able to learn a lot about digital design, FPGAs, hardware in general, and VHDL. I am sure that the knowledge acquired in this lab will enhance my engineering qualifications.