

Lab 4 Report

Daniel Chris Daniil - 504636684

Junhong Wang - 504941113

1. Introduction and requirements

Background Information

For our final project, we decided to develop a variation of the viral “Flappy Bird” game on the Nexys 3 board. The normal game features a bird that must not touch the pipes, which are moving towards it. In our variation, the pipes also move up and down, which results in the pipe gap to move. The game is displayed on the monitor via VGA connection, and the score is shown on the seven segment display on the board. The bird naturally falls down, due to gravity. To keep the bird flying, the user has to move the rightmost switch up (T10). The user also can control whether the pipes are moving up and down, via a switch (T9). Finally, the game can be reset through a button (D9).

Design requirements

- Moving Background: The background moves backward at a constant rate, and presents pipes of varying lengths, but with a fixed opening.
- Gravity: The bird moves (fall) downwards at a fixed rate when the “fly” button is not tapped.
- Jump Functionality: Turning “fly switch” on must make the bird move up, for a fixed amount.
- Collision Detection: When touching the ground or any of the pipes, the user loses.
- Pause after loss: After the user loses, the game enters a separate pause state
- Score Tracking: Every time the bird passes through a pipe opening, the score shown on the seven segment display is incremented by 1.
- Drawing: The graphics are drawn pixel by pixel, for better resolution.
- Pause Functionality : When the pause switch is turned on, the game enters pause state
- Moving Pipes: This is an unique feature we will add to our game. When the user turn on a switch, the pipes will keep moving in vertical direction.

2. Design description

Design Description

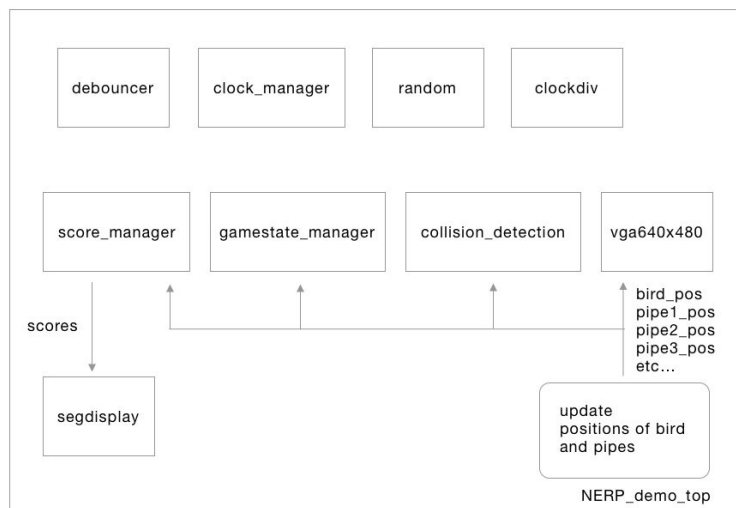
<i>Pause Switch</i>
The switch controls the operation mode. If off, the game flows naturally, and if on the game pauses.
<i>“Fly” Switch</i>
The button controls the height of the bird. When tapped, the bird will slightly moves up. (We decided to use switch instead to simulate the jump because we want the bird to continuously jump without pressing buttons many times.)
<i>Reset Button</i>
When tapped, resets the game after the user loses.
<i>Segment Display</i>
The seven segment display will show the score of the user at any point in time.
<i>VGA</i>
The VGA is used to display the game graphics

Modules

<i>NERP_demo_top</i>
Top module that updates the positions of bird and pipes
<i>clockdiv</i>
Clock divider that produces pixel clock and 7-segment clock
<i>segdisplay</i>
Display game score on 7-segment display
<i>vga640x480</i>

Handle rendering on screen
<i>debouncer</i>
Debouncer for jump button. We decided not to use switch for jump though.
<i>clock_manager</i>
Clock divider that produces clocks for bird and pipes and moving pieps.
<i>random</i>
A module that generates three random numbers.
<i>collusion_detection</i>
A module that checks if the bird collides with an object
<i>gamestate_manager</i>
A module that keeps track of the game state
<i>score_manager</i>
A module that updates the game score

Schematics



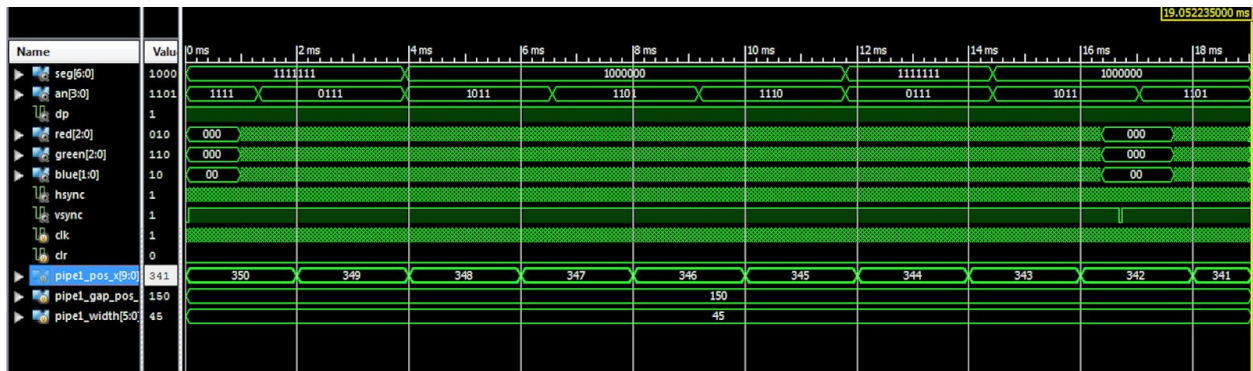
Simple schematic showing the hierarchies of modules

3. Simulation documentation

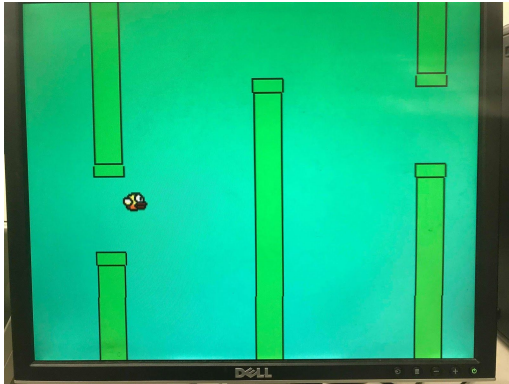
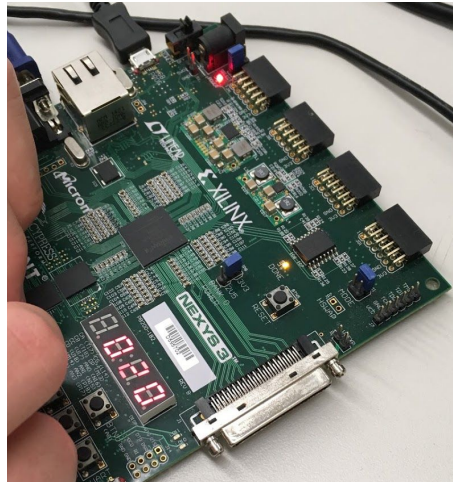
Requirement Tested / Test Cases

Test Description	Requirement Tested	Result
Display game on monitor	VGA connection	Passed
Start the game and do not touch "Fly Button"	Gravity and Pause	Passed
Make bird touch each pipe from every direction	Collision with all pipes and "Fly Switch"	Passed
Run bird between one pipe gap	Seven segment display and Score incrementation	Passed
Turn "Pause switch" on/off	"Pause switch"	Passed
Turn "moving pipes" switch on/off	Moving pipes	Passed

Simulation Waveforms



Check if the pipe is updating its position correctly. (Collision detection is disabled temporally)

VGA Example**Seven Segment Display****Graphics**

4. Conclusion

Summary of the Design

Our variation of “Flappy Bird”, involves the pipes moving up and down. Our implementation included 1 top module, 9 submodules and the UCF file. The top module manages the submodules, and each submodules each perform a different task. The game is displayed via VGA on the monitor, and the score is presented on the seven segment display. The “Fly Switch”, “Pause Switch” and “Pipe Switch” on the board control the game, and the reset button resets it.

Difficulties

Collision detection was the most difficult part of the lab. We decided to check if the four corners of the bird’s bounding box are overlapping with the pipes’ bounding boxes.

The graphics were definitely the most time consuming, as we had to draw the bird and pipes pixel by pixel.

Recommendations

More available lab office hours could help with the limited time we have to complete the project.