LAB 11 VERIFICATION REPORT

ECE 260 | 07/12/2021

Done by: Abrisam Durrani (A9) | Michel Shane Keh (A6) |

Contents

Table of Figures

List of Tables

1. Introduction

In this lab 11, we were tasked to design a digital circuit on the FPGA board, based on the knowledge learnt during ECE260 modules.

In this lab, the project that we have chosen to undertake is the creation of the game, Flappy brick. This project will encompass the following knowledge learnt during the ECE module:

- VGA interface
- Displaying Images with VGA Interface

This report will encompass the steps taken to ensure the workability of our circuit.

The verification steps taken are as follows:

1) Actual Testing

2. Introduction to Flappy Brick

Flappy brick is modelled after the popular mobile phone game, Flappy bird, a game developed by Dong Nguyen, a Vietnamese video game artist and programmer [1].

In flappy brick, the user will control a flying red brick, named Reddy and will have to maneuver the brick through an equally sized gaps between a pair of long rectangular blocks placed at random heights. Reddy automatically descends towards the bottom of the screen and will only ascend upon input from the user. Points are scored as the user successfully pass through the pair of rectangular blocks called obstacle blocks. The game ends when Reddy collides into the rectangular blocks or reaches the bottom of the screen.

The required Signal and button input design for Flappy Brick is as shown in the table below:

Signal Name	Pin Name on Board	Function/Description
Up	BTNC	To control Reddy to move up once button is pressed and to start the game
rstn	BTNU	To restart the game

Start	SW[0]	Intro Screen
audio	SW[15]	Turn on/off the audio
r[2:0]	-	VGA red color signal
g[2:0]	-	VGA green color signal
b[1:0]	-	VGA blue color signal
hs	-	VGA horizontal sync signal
vs	-	VGA vertical sync signal
ubird	-	Upper bound for Reddy
dbird	-	Lower bound for Reddy
ublock	-	Upper bound for obstacle blocks
lblock	-	Lower bound for obstacle blocks
rblock	-	Width bounds for obstacle pipes
usuiji	-	Randomized upper bound coordinate for obstacle block
videoen	-	Signal to enable the display of the game
Pass	-	Flag to increment score when flag is high
Num0[3:0] to Num3[3:0]	-	Counter for the score
Success	-	Flag set high every time Reddy passes through the obstacle. Once this flag is set low, the game automatically ends
Firstup	-	Flag to prevent continuous input if the BTNC is pressed and held

Table 1: Signal description and pin assignment

3. Testing

To ensure the workability of our design, several test scenarios were carried out. The test scenarios target the various functionalities of Flappy Brick. The test scenarios are as follows:

Test Scenarios	Purpose
VGA Display	To be able to display Reddy and the obstacle bricks
Button test	To ensure that Reddy can be controlled by user via BTNC input
Reddy Behavior Test	To ensure the workability of Reddy's default falling movement
Obstacle Bricks Test	To ensure that obstacles are able to move towards Reddy and are generated at random heights
Collision/Game Over test	To ensure the game ends whenever Reddy collides into an obstacle or reaches the bottom of the screen
Final Overall Game Testing	To ensure the full functionality of Flappy Bricks

Table 2: Test Scenarios and their respective purposes

3.1. VGA Display

This test scenario is the first test conducted and is to ensure that we are able to display Reddy and the obstacle bricks. The successful testing criteria would be to be able to display Reddy and the obstacle Bricks concurrently.

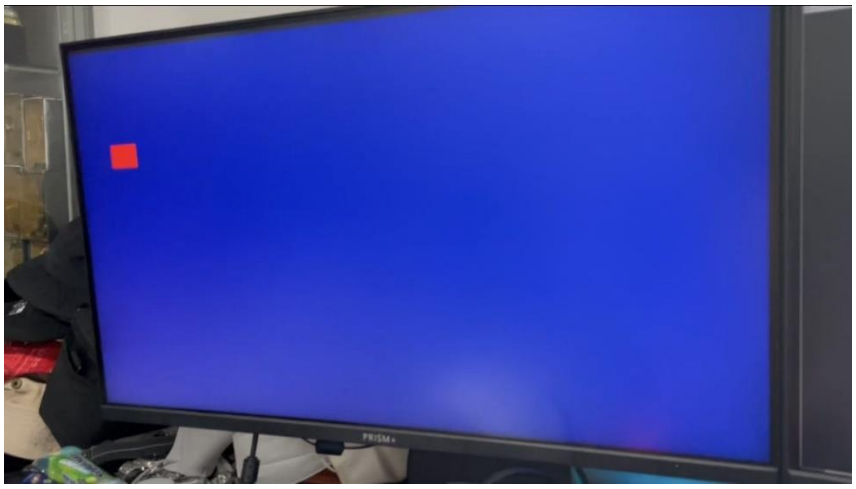

Figure 1: Testing the display of Reddy

Figure 1 above shows the successful testing of displaying Reddy on the screen. After which, we would test the ability to display Reddy and the obstacle bricks concurrently.

Figure 2: Testing of displaying Reddy and the obstacle bricks concurrently

Figure 2 above shows that we were able to concurrently display Reddy and the obstacle bricks.

3.2. Button Test

The purpose of this test is to ensure that the user input from BTNC is able to be detected and is able to control the UP movement of Reddy.

The success criteria of this test is to visually see that Reddy moves up upon input from the user.

We were able to successfully carry out this test concurrently with Obstacle Brick Test.

3.3. Obstacle Test

This test is to ensure that the obstacle is able to move towards Reddy, to simulate that Reddy is flying towards the obstacles.

The success criteria of this test are to be able to see visually that the obstacle blocks are moving. This Test scenario was carried out concurrently with the Button Test specified n section 3.2.

Figure 3: Initial state of Reddy and obstacle brick

Figure 4: Next state of Reddy and Obstacle Brick

Figure 5: Testing the generation of obstacle bricks of differing heights

Figure 3 and 4 above shows the transition of Reddy and obstacle bricks from its initial position to another differing position, thereby indicating that the button test and obstacle test was successful. Figure 5 above also shows the obstacle brick of a differing height from the obstacle bricks in figure 3 and 4, thereby showing the successful generation of obstacle bricks at differing heights at random.

3.4. Collision/Game Over Test

In this test, we will test whether a game over scenario can be detected and implemented correctly. The two failure scenarios are as follows:

1) Colliding into an obstacle brick
2) Reaching to the bottom of the screen

Once a failure scenario is reached, a red screen will be displayed, indicating the game is over.

The success criteria of this test are that once a failure scenario happens, the red screen will be displayed.

This test was successful, as shown in the demo video that was submitted together in the submission folder.

3.5. Final Overall Game Testing

This test is to ensure that Flappy Brick is functional.

The success criteria of this test are to ensure that the game is able to run correctly, the scores are counted and displayed correctly, and failure scenarios are able to end the game.

This test was carried out successfully and can be seen from the demo video that was submitted in the submission folder.

4. Conclusion

Verification is an important step in ensuring that our circuit design was functional. BY carrying out verification, we were also able to detect errors in our code and were able to carry out debugging activities.

In addition to this, we were also allowed to experience in practice various steps taken in the working industry when designing circuits. This project exposed us to crafting our own test scenarios and defining our own success requirements. This project has enabled us to develop important skills that will be required in the future, while also using the knowledge taught in ECE260.

Through verification, we were able to ensure that Flappy Brick was functional and correct, and helped in designing and creating a working circuit and Game.