# FlappyBat VHDL Gamer Input Project

**Alim Kerem ERDOĞMUŞ - 151220132027**
**Burak Can FAZLA - 151220144035**

## Contents

# 1. INTRODUCTION

## 1.1. Project Details

This project aims to design a game on a VGA display with FPGA to an Xilinx Spartan 3E kit. In this report, first the basic and optional features of the game are explained, then the emergence stages of the game are detailed. While designing the game, technical details such as working principles of the circuit parts to be coded and input/output information are given. In the design section, the system architecture and the working systems of different components are explained. Finally, in the test section, the functionality of the game features will be tested, and at this stage how the game is stabilized.

## 1.2. About FlappyBat Game

FlappyBat is designed as a different version of the game known as Flappy Bird on the market. The flappy bird is replaced by a bat. One-touch gameplay in normal play is arranged as up / down movement in this game. In addition to the overcoming obstacles mission, insects have been added to allow points to be won when defeated. Other game rules are the same as Flappy Bird. The score is reset when the obstacles are multiplied, and each obstacle passes by one point. The obstacles come from right to left, over the bat in random positions, and we are expected to overcome these obstacles with the bat under our control. The main goal of the game is to earn as high points as you can with these obstacle overcoming and insect eating actions. The score earned is displayed on the kit's 8-bit LEDs.

# 2. METHODS

This section describes the basic and optional features of the game and the FlappyBat's gameplay elements.

## 2.1. Game Properties

This section lists the features that will be included in the game. These features include:

- Essential features,
- Optional features.

### 2.1.1. Essential Features

The basic dynamics of the game are:

- Display the game's main menu
- Two controls (up-down movement), one restart, one reset button
- Display of bat figure, boxes, insects and score
- Shift of boxes and insects in a defined order (determination of coordinates, positions and transition areas, etc.)
- Detection of contact between bat and boxes and insects
- Switch to different screen when the game is over:
    - Score indicator
    - Restart the game or return to the main menu

### 2.1.2. Optional Features

In addition to the basic features, additional features that are thought to improve the quality of the game are:

- Keeping record of record points
- Insects move randomly around themselves
- Improved graphics and colors
- Cave background and arrangement of boxes as cave stalactites
- Random size of boxes
- Difficulty levels
    - Increased sliding speed

- Change in spaces between boxes (tend to be less)
- Game pause
- Easter Egg

## 2.2. Start Game

In this section, we will start the game with the bat model designed for FlappyBat.

### 2.2.1. Controlling the Game

The control buttons to interact with the game must be defined. FlappyBat will have three different physical buttons.

- Up button to start the game and move upwards (V4),
- Down button moves down (K17),
- The right button will reset the game (RESET) (H13).

Figure 1 shows the buttons and button codes to be used in the FlappyBat game on the Spartan 3E kit.

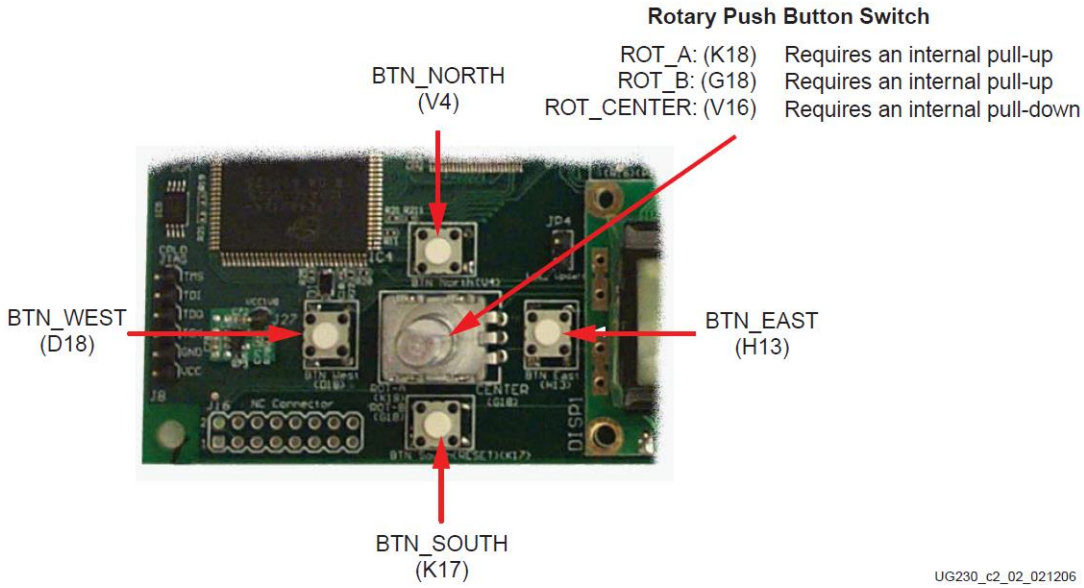


*Figure 1:Spartan 3E Button Config*

### 2.2.2. In Game

When the game is started, the bat we control will be fixed in the middle left of the game screen, while no obstacles (boxes) will appear on the screen. The second press of the V4 button starts the game by starting to slide obstacles from the right side of the screen.

### 2.2.3. Playing Phase

The buttons V4 and K17 (ie the up and down keys) will allow the bat to move up and down. With these maneuvers, the player will try to pass through the spaces without hitting the boxes that are sliding from right to left. It will try to eat insects in the areas where there are no boxes.

When the bat touches one of the boxes, the score is reset. The game will continue by touching the box from a different point.

## 3. GAME DESIGN

By defining the functioning of the game, the game architecture is more or less visible. Different components were planned for the healthy operation of the game and their designs were separated and the architectural design was realized. The components of the circuit that make up the game and their technical description are listed below.

## 3.1. Components

In this section, the components that make up the game are listed and explained in detail. Then the circuit diagram will be formed with these components.

### 3.1.1. Display

**Functions**

This component is the VGA controller and is the module that displays the display components. Its function is to scan the display lines and subtract the counters (horizontal / vertical coordinate values) that represent the scan. It is also the controller that controls the arrival of boxes and insects on the screen.

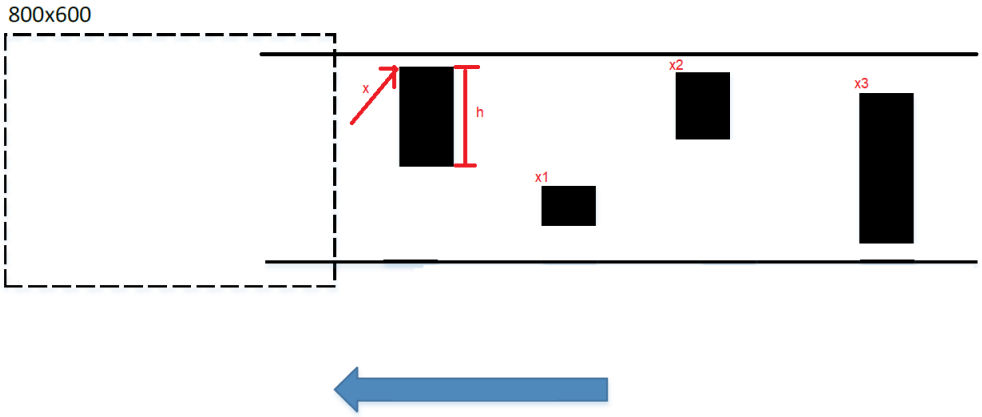


*Figure 2:Boxes screen and display modüle working princible (x: coord, h: height)*

For the current game, bats, boxes and insects are displayed on the screen. The y coordinate and status are taken for the bat. The size of the character and the x value are fixed.

For boxes, the Display component receives the x coordinate from the upper left corner of the boxes and the x coordinate corresponding to the height information of the boxes. The width of the boxes is fixed (100 and 150 px).


*Figure 3:Gameplay*

### 3.1.2. Btn_sync (Button Syncronization)

**Functions**

Tüm input düğmelerini clock sinyali ile senkronize eder.

### 3.1.3. BAT (pointer_move)

**Functions**

This component calculates the position of the bat in the game. This position information includes the x and y coordinates of the bat as well as the size and width values.

The x coordinate of the bat in the game is fixed (about 280 on the 800x640 screen) and the width / height values are fixed (about 40 pixels). Only the y coordinate changes.

When starting the game, the bat stands in the middle of the screen (y = 300 px). The bat will move up / down on the y-axis according to the command the player gives with the control keys. If this change is expressed by the variable n, the instantaneous position can be shown as "± ±. The algorithm required to calculate the y coordinate with this motion can be as follows.

```
if((y>0) and (y<640)){
        if (( btn_up == 1) and (btn_down == 0)) {
                y++;

         }else if((btn_up ==0) and ( btn_down == 1)){
                y--;
          }
}
```

*Figure 4:FlappyBat Character Models*

## 3.2.Top Level Component

**Functions**

The Top Level Component is a system that connects and connects all the components and components of the game. The component diagram is attached.

## 3.3. Internal Packages / Signals Created for Constant Values

There are fixed values that the components in the system will use among themselves. These constants to be defined are used by each component as needed and the system remains connected to each other.

The constants to be included in the system are listed below:

- Constants that define the VGA driver (display width / height data),
- Bat position and dimensions (Bat x position, first y position, size, etc.),
- Score calculation,
- The position, number and order of arrival of boxes and insects,
- Crash test components,
- Visual elements on the screen (graphic elements, image images of characters, etc.),

A second set of constants to be given as a matrix will also be defined in order to systematically reflect images on the screen and avoid errors. This package is intended to include the following items:

- Score text and record score (optional),
- Digit numbers (0-9),
- "Start Game" screen,
- Different image visuals including bat flapping movements.

## 4. TESTING

When designing the code system of the project, the basic and optional features described in the above sections are as follows:

- Display the score screen,
- More characteristic movement of insects,
- Game stages,
- All the main objectives for the Bat character, except features such as animated images and add-on targets,
- Among the optional targets, only random box arrival and different box sizes were realized.

## 5. ADDITIONS

In this section, the block diagram of the project is given.

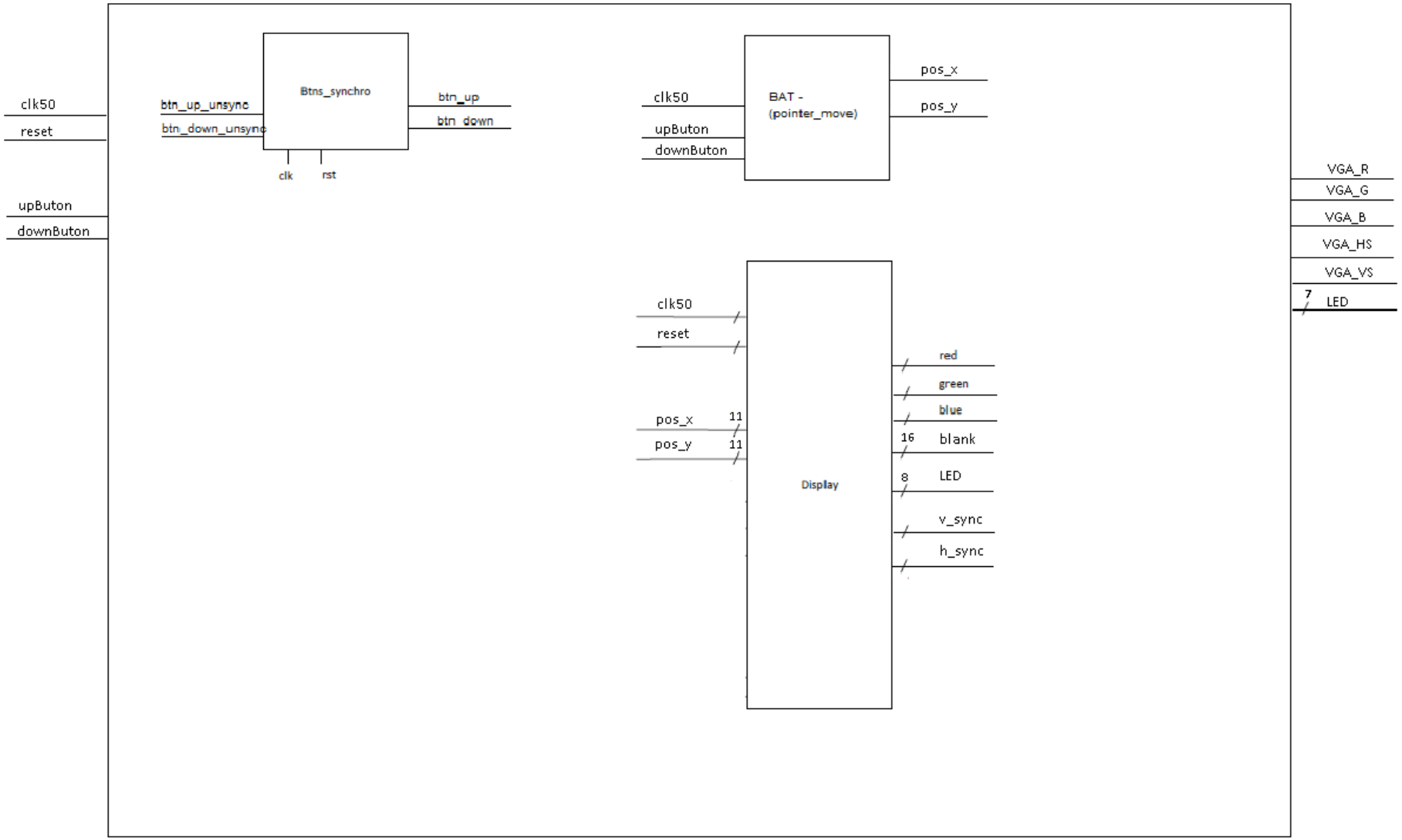


*Figure 5:FlappyBat Top Level Component Schematic*

## 6. RESULTS

In the first reported phase of the FlappyBat FPGA game design project, research and ideas related to the design architecture, component structures, and working logic of the game are explained. In line with this design draft, the second and final stage of the project was aimed at applying and testing the ideas to the Spartan 3E kit, and then completing the project by ensuring the stable operation of the game. In line with these goals, most of the main objectives of the initial reporting have been achieved and a stable game system has been established. However, even if the project can be further improved, especially with visual improvements, this will not be effective for the Spartan 3E.