Project Proposal: Screaming Bird

Yiran Hu (yh3639), Yuesheng Ma (ym2976), Yang Li (yl5456), Chenyang Zhou(cz2791)

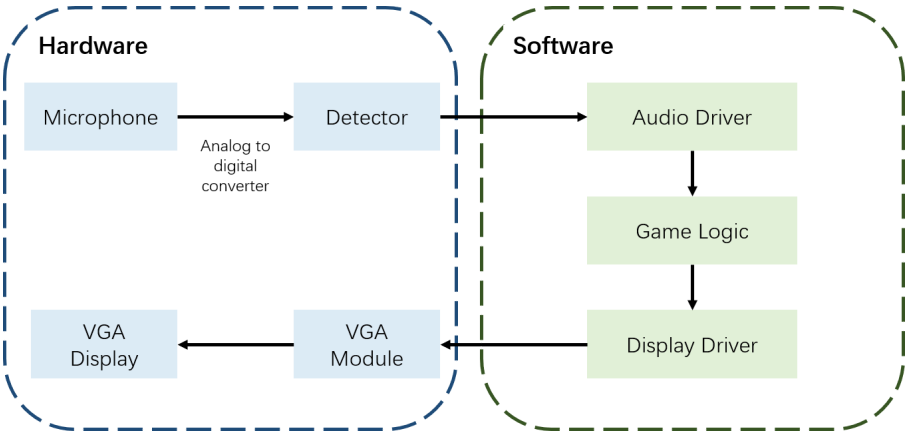
Overview

Our team intends to build a voice control game based on Flappy Bird, developed in 2013 by Dong Nguyen, an independent game developer from Vietnam, where players only need to use one finger to control the flight status of the bird, allowing it to avoid uneven pipes along the way. In our design, we will replace finger manipulation with sound and use sound to control the rise and fall of birds. When a sound exceeding the threshold is detected, the bird will rise, otherwise it will naturally fall.

Design Outline

The complete system takes in voice commands and keyboard inputs to control our game displayed on an VGA display. The system is divided into three components: audio module, game logic module, and display module.

- Audio Module: Includes detector and audio driver. Captures audio signals and converts them into flap commands for the bird using sound level threshold detection algorithm.
- Game Logic Module: Manages the core mechanics of the game, including bird movement, collision detection, and score management.
- Display Module: Includes display driver, VGA module, and VGA display. Renders the game's graphical output to a display.



Hardware

- FPGA Board
- Microphone: Captures voice commands from the user. An analog-to-digital converter is required to process analog signals.
- VGA monitor: Outputs the visual elements of the game.

Software

- Audio Driver: After the audio data is collected and detected by hardware, the Audio Driver will turn it into the command to control the bird.

- Game Logic: The game logic driver needs to perform the following functions: determine the vertical position of the flappy bird based on audio input, enforce game rules (e.g., determining if the game is over, calculating the number of pillars passed by the bird), and manage the generation of graphics.
- Display Driver: This driver collects all the necessary data for generating graphics, such as the coordinates of the bird and the pillars, as well as information about whether the game is ongoing. These coordinates are stored in memory and are updated based on the game logic. The graphic driver accesses this memory using addresses and then displays the required graphics on the screen.

Milestones

- Milestone 1
    - Implement audio interface on the FPGA board.
    - Implement display interface on the FPGA board.
- Milestone 2
    - Software driver interface with hardware
    - Implement the Game logic
- Milestone 3
    - Complete Graphics
    - Finish the project