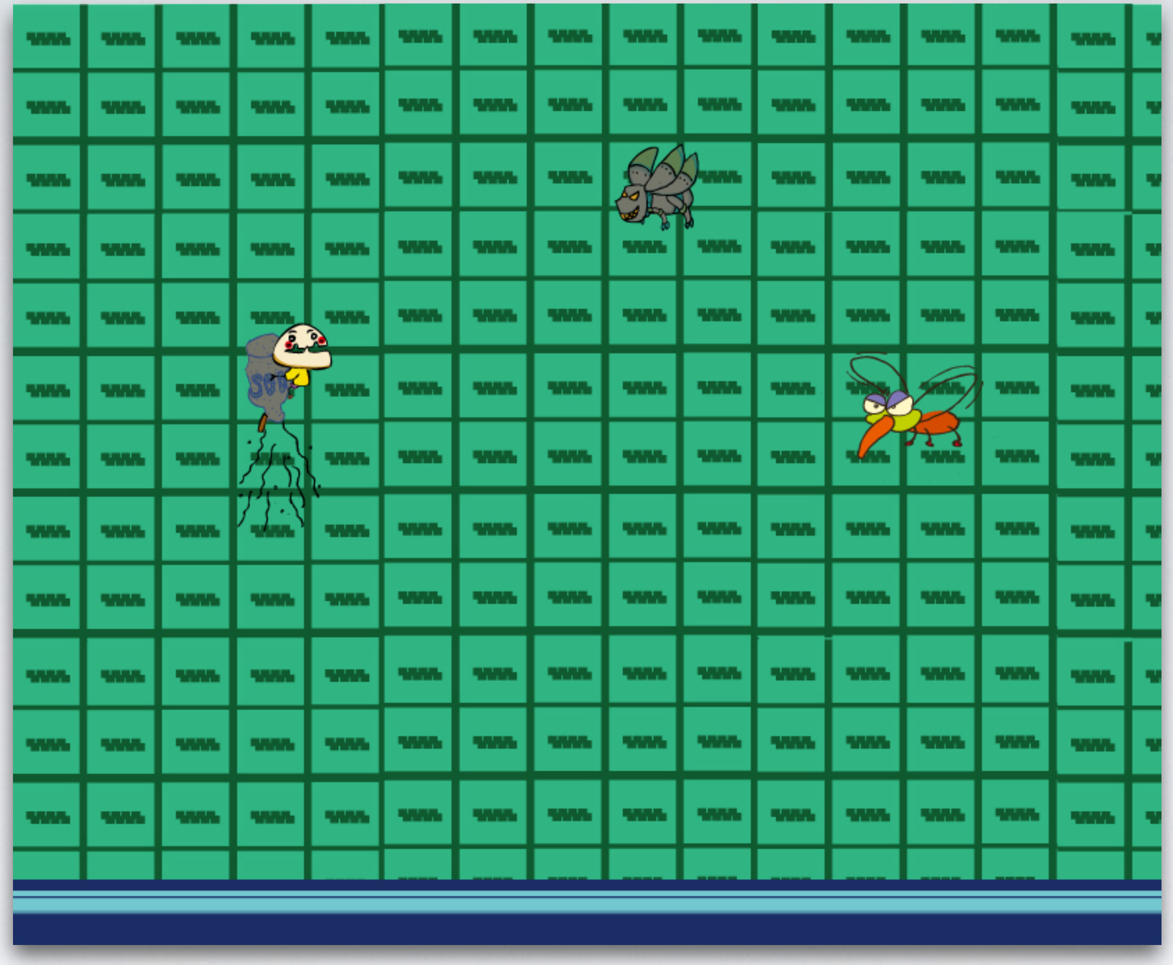
COLACAN

flight shooting game

CONTENT

- 1. Game Concept Introduction
- II. Architecture
- III. Hardware Implementation
- IX. Software Implementation
- X. Lessons Learned



GAME CONCEPT INTRODUCTION

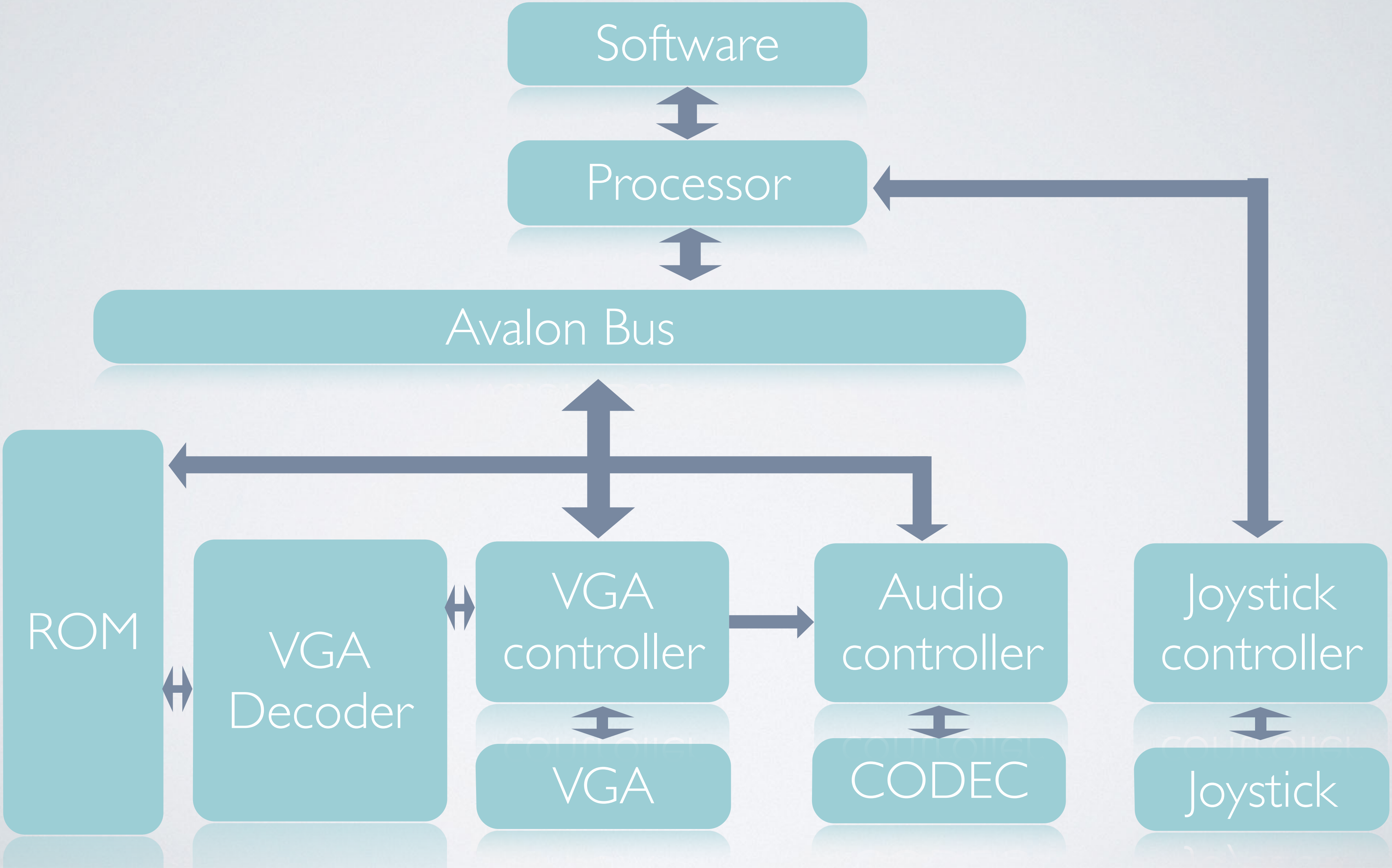
You control a goblin riding a ColaCan. Goblin can breathe flame blast to wipe out enemies.

GAME CONCEPT INTRODUCTION

GAME FEATURES

- 1. Main character, the Goblin has detailed animations.
- II. Flightstick control, more fun.
- III. Two dimensional moving range.
- IX. All graph patterns are originally hand-drawn.

ARCHITECTURE



HARDWARE IMPLEMENTATION

- **AUDIO**:

- Our game sound is implemented in Analog Devices SSM2603 audio CODEC(Encoder/Decoder).There are three clock cycles being applied: main clock cycles(50MHZ); master clock cycles(11.2896MHZ, generated by PLL); LRC(44.1kHZ).

- Only DAC model been used.

HARDWARE IMPLEMENTATION

- **VGA**:
- All sprites loaded from ROM and then access position information from software, then buffered on screen. Timing control is in implement because multiple items share the same figure which means they may access the same ROM address simultaneously.
- Vertical synchronize buffer is applied.

HARDWARE IMPLEMENTATION

- **VGA DISPLAY**:
- White determine. We try to avoid using white pattern while drawing figures, therefore makes each sprite fits the background.
- Vertical synchronize buffer is applied.

HARDWARE IMPLEMENTATION

- **JOYSTICK**:

- The USB joystick directly connect to usb port on the board. Then the signal would be access by software.

SOFTWARE IMPLEMENTATION

- **OVERVIEW**:
- 1. Character animation frames play.
- II. Items collision determination.
- III. Characters movements monitor and display.
- IX. Scores recording.
- X. Power Bar System.

SOFTWARE IMPLEMENTATION

- **Character animation frames play**:
- Each character(Goblin, enemies and fireball) has been set as a group of sprite. AS it moves, the detail animation plays via sprites changing.

SOFTWARE IMPLEMENTATION

- **Items collision determination**:
- If two items(goblin and enemy; fireball and enemy) are close enough, system will determine them as collision.

SOFTWARE IMPLEMENTATION

- **Characters movements monitor and display**:
- I. Goblin's position data is provide by joystick.
- II. Enemies are randomly generated and move via a fixed trace.
- III. Fireball which is generated by Goblin moves via a fixed trace.

LESSON LEARNED

- Use Quartus to construct hardware design.
- Processor, software, memory, VGA and other component in a embedded system.
- Inspiration in a video game design.