CSEE4840 Embedded Systems Design

Chenzhi Lu(cl4407), Shaokun Feng(sf3209), Zongwei Zhen(zz3083)

March 29, 2024

1 Introduction

In this project, our group will design a single-player game based on SoC (System on Chip) called Monster Casino. This game combines elements of casino slot machines with monster capture. The objective is to obtain monsters from the slot machine, nurture them, and finally use the monster to destroy the casino. Players control the start and stop of the slot machine with a USB controller, move the crosshair, and hit moving monsters to interact with the game. The FPGA is responsible for reading images, as well as driving the screen. The software communicates with the FPGA through Avalon bus to complete the design of the game logic.

It needs to be emphasized that the ultimate outcome of our game is to destroy the casino. This game can be understood as requiring the use of monsters within the game or other mechanisms to achieve the casino's downfall.

Figure 1: Figure of DE1-SoC

2 Design Overview

2.1 Peripherals Design Overview

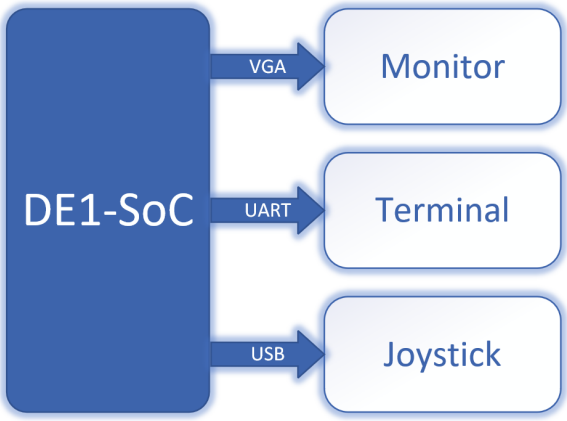


Figure 2: Peripherals Design Overview

2.2 Software Flow

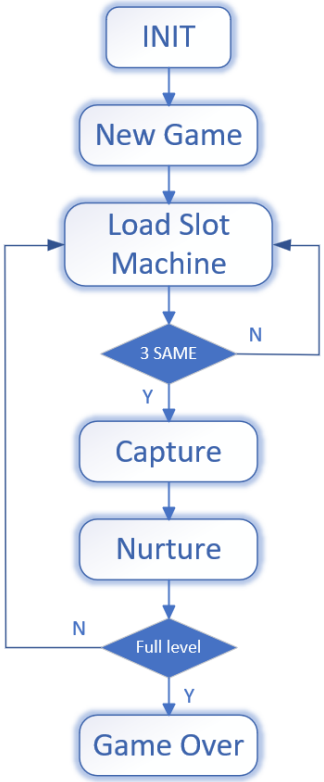


Figure 3: Software Flow

2.3 Global Design

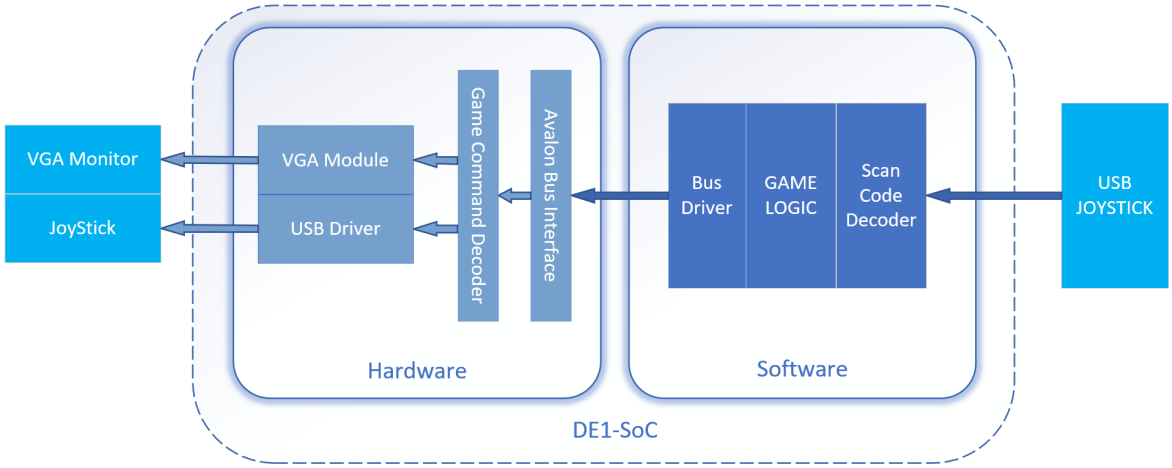


Figure 4: Global Design

3 System Design

This project contains hardware design and software design.

3.1 Hardware Design

The FPGA is responsible for file reading, peripheral interactions, and the design of a 32-bit bus.

Files include image files. Images will be read into the FPGA in txt format at compile time, and music files will be written into the FPGA in the same manner, preparing them for software calls.

Peripherals include a USB joystick, VGA screen output-related peripherals. The USB joystick uses the kiwitata brand controller. The VGA screen uses a screen provided by the laborator.

Figure 5: USB joystick will be used in this design

Due to the large number of images that will appear in the game, efficient communication is necessary to make effective use of the 32-bit bus. The 32-bit bus will transmit information such as the coordinates where the image appears, frame selection needed for animation, and other instructions. The FPGA has reserved interfaces that allow C language to call images.

3.2 Software Design

The software consists of three parts: slot machine design, capture and nurture system, and other designs.

The slot machine needs to accomplish a random number probability design, user interaction with the slot machine, and the calling and switching of images required by the slot machine. The random numbers will be directly generated using the rand function, and it is expected that only 4 patterns will be used in the slot machine part to reduce debugging difficulty, with 3 display windows completing the display of patterns. Game flow design will be based on the different patterns displayed. Users will use the joystick to start the slot machine and confirm the display of each window.

In the capture and nurture system, capture and nurture functions are implemented respectively. Capture will occur based on a small probability of monster rewards from slot machine results. When a monster reward appears, it will move differently on the screen based on difficulty. Players need to use the joystick's directional keys to track the monster and complete a certain number of clicks within a certain time. Clicks within a certain range of the cursor and the monster's center are deemed successful. After meeting the requirements, capture is completed and recorded. Nurture is based on obtaining score rewards from the results of the slot machine, or significantly enhancing the monster's abilities after recapturing the monster from the slot machine results. Once the monster's abilities exceed a threshold, it enters a rampage, destroys the slot machine, and the game ends.

Other designs include display design, i.e., the design of score display, slot machine, and page environment patterns. Difficulty algorithm design, i.e., how to maintain playability and increase interest. Finally, backdoor designs and special commands for feature debugging are included.

4 Workflow

4.1 Hardware Configuration

Setting up the FPGA to produce the appropriate signals for VGA display and establish the required hardware connections for VGA video output configuration.

Establishing the hardware connections to interpret joystick input through the USB port for configuring joystick controller input.

4.2 Data Communication

Design an interface between FPGA and software, creating an interface for transmitting image coordinates, image frame switching, and related commands.

4.3 Software Development

Design C language code to recognize joystick scan codes.

Design code to implement image calls, conducting tests for image positioning and frame switching.

Design the slot machine, monster capture, and nurturing functions separately.

4.4 Optimization and Verification

Optimize image display and certain algorithms to achieve better results.

5 Design Picture Element Examples

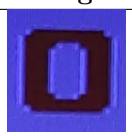
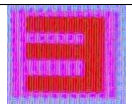
Category	Image	Size	Variants	Bits
Numbers and Letters		16 x 16	36	$(16 \times 16) \times 36 = 93,312$ bits
Figures		16 x 16	4	$(16 \times 16) \times 4 \times 3 = 3072$ bits
Crosshair		16 x 16	36	$(16 \times 16) \times 2 = 512$ bits
Rock Shields		100 x 100	4	$(100 \times 100) \times 4 \times 2 = 10,368$ bits

Table 1: Some examples of pictures.