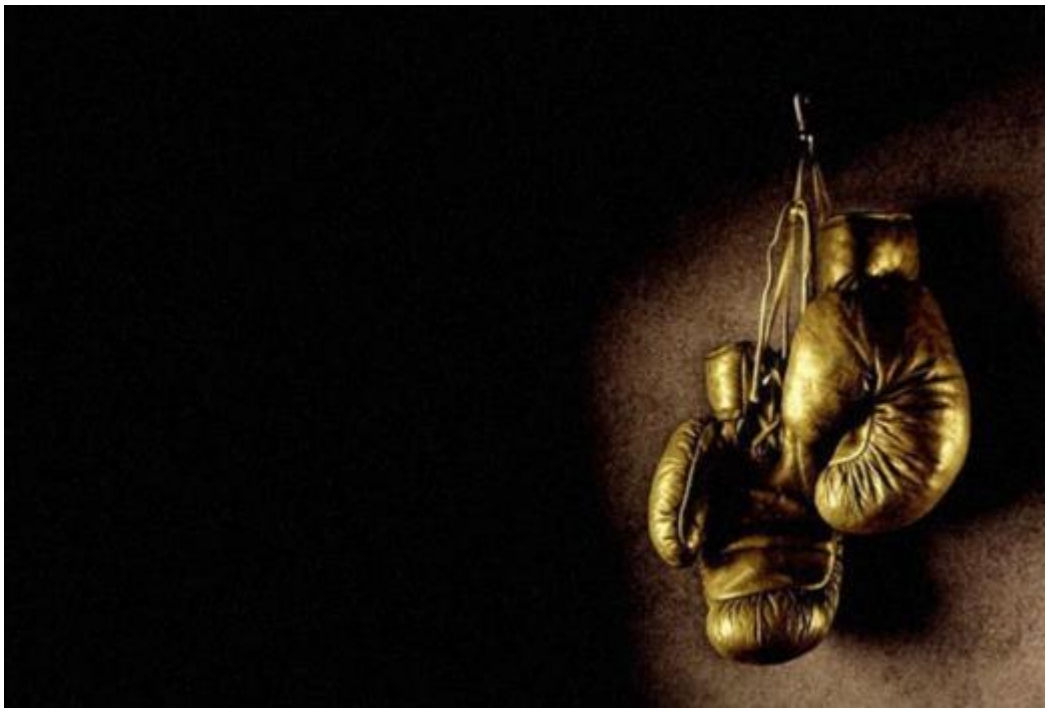
CSEE 4840 Embedded System Project Design

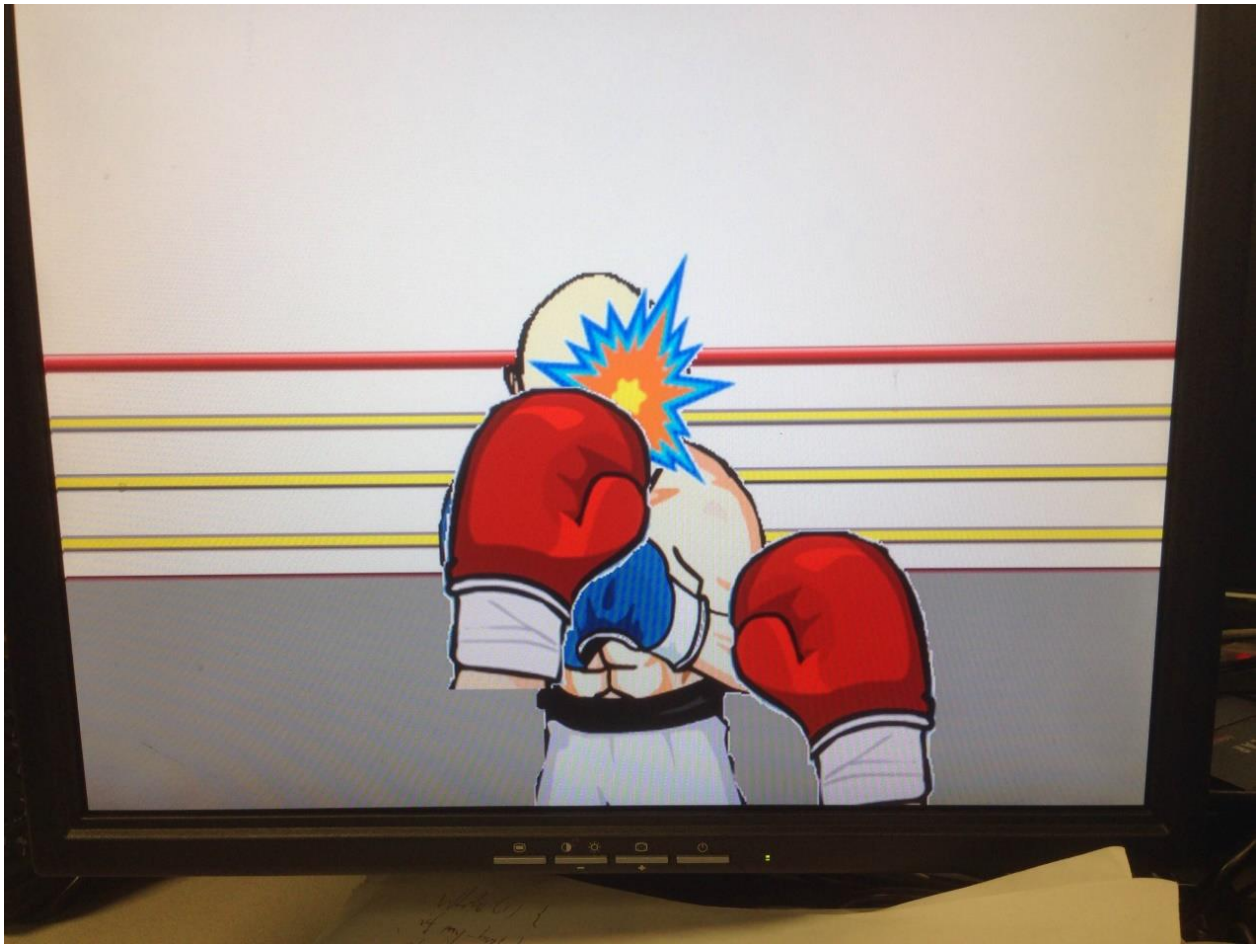
Video Game: Real Boxing

Jiaqi Guo jg3639

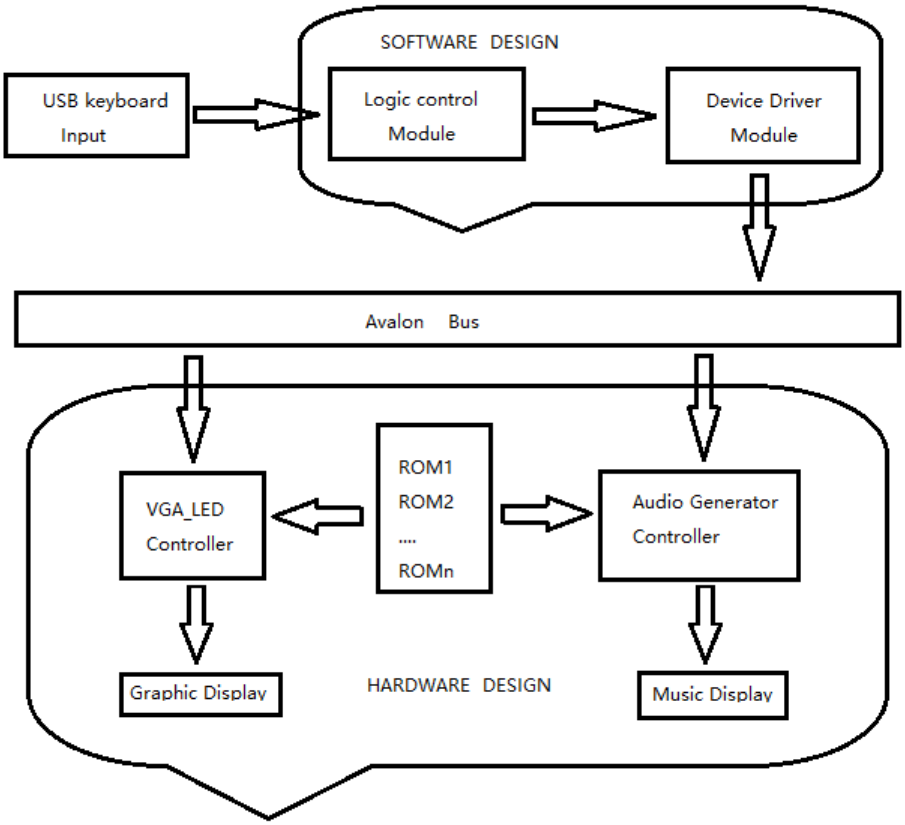


1. Introduction

For this project, I aim to implement a boxing game via FPGA Cyclone board and several peripherals such as VGA monitor Audio generator and USB keyboard. This simple game could be a good practice for my embedded system knowledge since it combines both hardware and software design. In the game, players could control their boxers through USB keyboard and interact with AI boxer through VGA monitor. Players could do left, right attack and defend. Once a boxer's health bar goes down to the bottom, the game is ended and his opponent wins.



The four basic parts of the video game are logic control module, peripheral driver module, VGA image display module and Audio generator module. The logic control module reads in signals from keyboard, makes logical judgement, and gives output to peripheral driver. Peripheral driver receives data from logic control module, encodes the data and sends them to VGA monitor and Audio generator. After the peripherals receive data from their driver, they decode the data and display the images or music. The overall flowchart can be seen as follows:

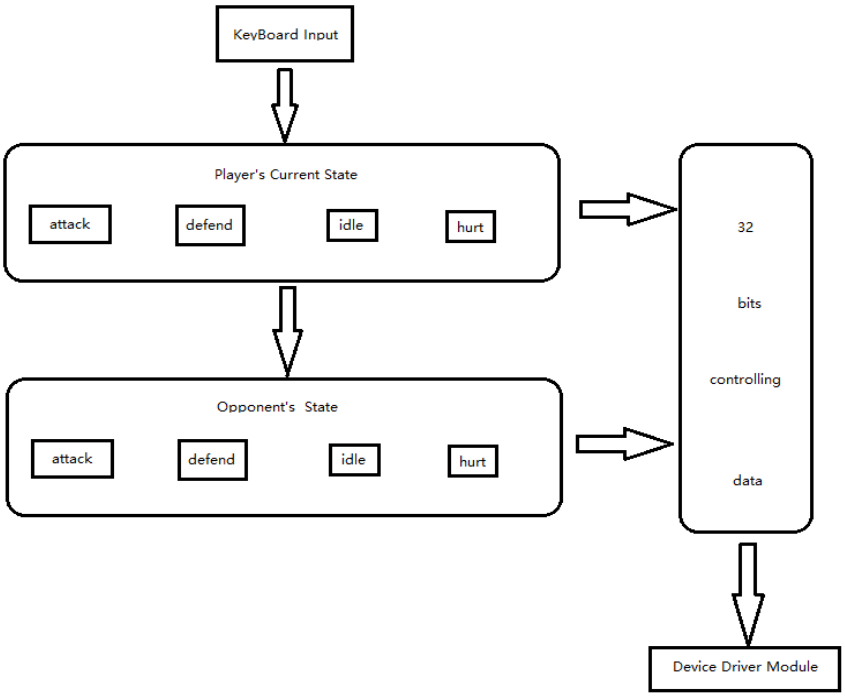


2. Logic Control Module

For Logic Control Module, I intend to run the main part of the program in a while loop. In each iteration, control module first checks the keyboard input and decides the current player's state through the keyboard input. Using the function RAN(), Logic Control Module would decide what AI boxer will act according to player's current state. Both player and AI boxer would have four states: attack, idle, defend and hurt. Once the states of player and AI are decided, Logic Control Module would generate a 32 bits controlling data and send this data to hardware interface through Device Driver Module.

AI boxer would do idle, attack or defend according to player's idle or defend states, defend or hurt according to player's attack state. If AI boxer chooses to attack, control unit would set player's state to hurt. And when one side's state becomes defend or hurt and the other side's state is attack, the display condition is met and control unit would send data to device driver, after that both player's and AI boxer's states are reset to idle.

This is shown in the following logic control flowchart.



3. VGA Image Display Module

This part mainly designs the hardware which generates signals to VGA monitor and displays image on the 640x480 pixels screen. Each pixel needs a 24 bits signal that is composed of three 8 bits RGB values. In order to display the images in several different layers, I will use Sprite Graphics discussed in the lecture. At each pixel, the VGA controller first finds out how many sprites have values at this point, and each sprite reads in data from its ROM. Then through a priority controller, VGA module finally decides which sprite to display at the pixel. The sprites are background_slice, fist_atk, fist_def, opponent, opponent_fist, and spark. And a few sprites are listed below.

Figure 1. Sprite for AI opponent

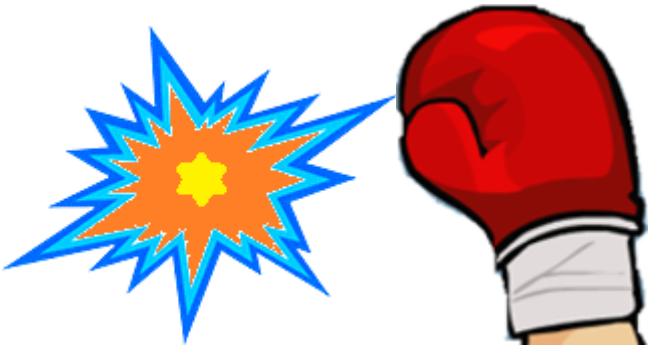


Figure 2. Sprites for attacking fist and hit effect

As we can see from these examples, most sprites are not restricted in a rectangle. In order to get rid of the margin part, I add a signal NBLANK to each of these sprites to show the pixel value is not white and use this signal in priority controller to help decide which sprite to display.

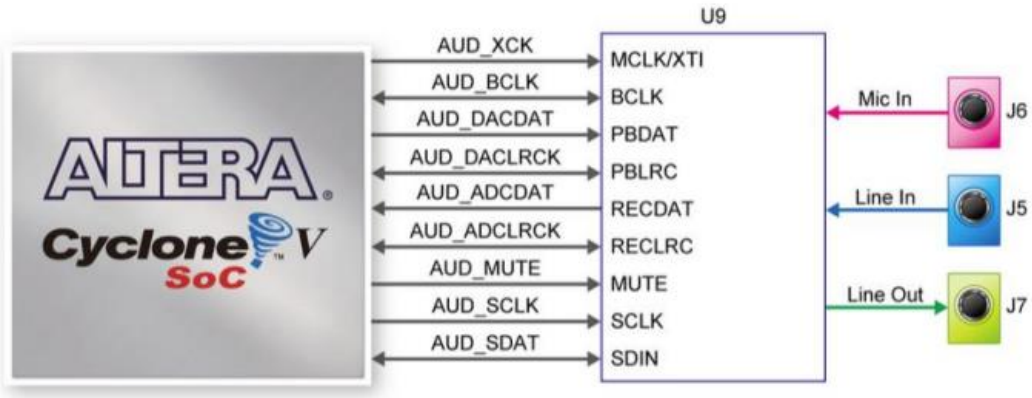
Another trick I played to save memory is image transformation. As we can see, there are a lot of symmetric images in the game like left and right fist. So I use just one ROM when I deal with these symmetric images. Besides, when I need to display an image and its amplified version, I also use one ROM to store the smaller image data and use image interpolation to get the larger one. This might give us poor resolution in the larger image but does save me a lot of memory.

The following table contains the current sprites I used in the game, and in the future I will add health bar, game start symbol and a few other details.

Sprite	Amount	Pixel Size	Total ROM
Fist_atk	1	170*125	63750
Fist_def	1	170*125	63750
Opponent	1	300*200	180000
Spark	1	120*150	54000
Opponent_fist	1	115*115	39675
Background_slice	1	125*1	375
Total	6	117650	352950

4. Audio Generator Module

In this project, I will use analog devices chip SSM2603 audio CODEC provided by the Cyclone V Sockit board for Audio encoder and decoder. SSM2603 supports sampling rates ranging from 8 kHz to 96 kHz. According to Nyquist sampling theory and human’s acceptable audio frequency range, I choose the sampling rate to be 44.1 kHz.



At the beginning of the game, the software extracts the background music sample previously stored in the SDRAM and loads it into the circular queue in the Audio Controller Module, which will continuously send signals to the SSM2603 Audio CODEC to play the sounds until the player exit the game. During the boxing game, when logic control module decides the player's and AI boxer's states, it sends audio controlling signals as well as image controlling signals. It will instruct the DRAM to load the corresponding special sound effect into FIFO, and instruct SSM2603 to play the special sound.