

# **PAC-XON**

**CSEE 4840 Embedded System Design**

Dongwei Ge (dg2563)

Bo Liang (bl2369)

Jie Cai (jc3480)

# PAC-XON Game Design

## Project Introduction

Our project is to design a video game that consists of a combination of custom designed hardware and software, which will take everything we have learned so far in the past 2 months. In the hardware part, we will design a VGA controller, a CPU, a RAM controller and a interface between the hardware and the software.

To implement our project, VHDL and C programming language are both must for hardware part and software part respectively. With them we can handle inputs from the keyboard to the video display output on the screen.

We play to use the VGA output of the Altera board to present the game's graphics. Images will be initialized in hardware and ghosts' movement done in software. The software will also control game logic. Player will play the game with the "⇐ ⇨ ↑ ↓" keys on the PS2 keyboard.

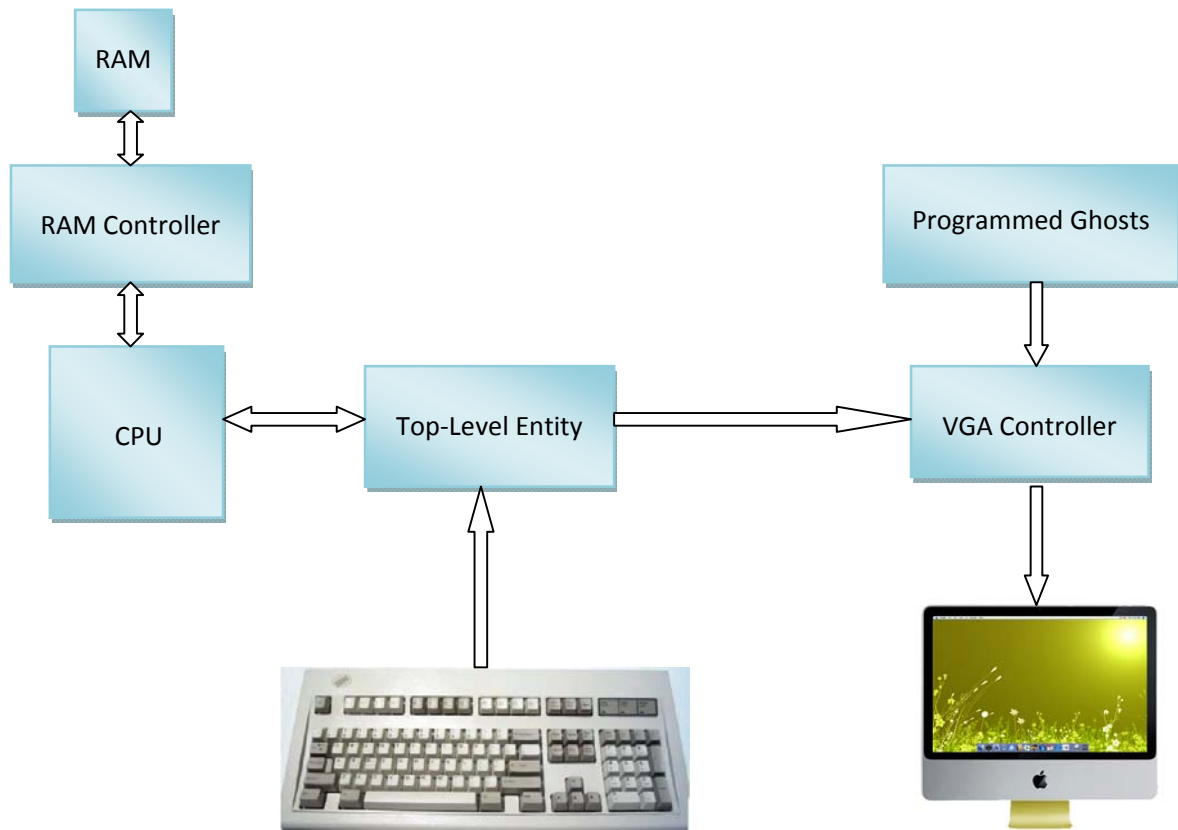


Fig.1 The block diagram of our project

## Game Design:

### 1. Game Logic:

The game is similar to the original PAC-XON game: Each player controls a pac-man with four lives at the beginning of the game. When the game starts, pac-man can move freely from the top-left. Player must fill empty space and capture ghosts by building wall. As soon as the player fills 80% or more empty space he will go to the next level. Players must also beware of the two ghosts. If one touches the wall the pac man is building or catches the pac man, he will lose a life.



Fig.2 Original PAC-XON level-start shot



Fig.2 Original PAC-XON level-up shot

## 2. Game Graphics:

In the lab3, we have learned to implement a live video display. We were able to store our character sets and graphics in the RAM.

We will generate our graphics in an array representing 8x8 pixels, each coded in hexadecimal. Each 8x8 pixels array on the screen will be represented by two strings of nine hexadecimal values. Each group of nine values will represent a mapping in one color. The first term in the string will represent the value for the color of the following pixel map. We will then proceed to read the second string of values to overlap the same 8x8 pixels space on our screen. The second string will represent the second color, with no pixels overlapping the first layer. This approach allows us to create an 8x8 pixels array with two colors. For convenience, the game engine will read 18 values at a time.

### 2.1 Pac-man Graphics

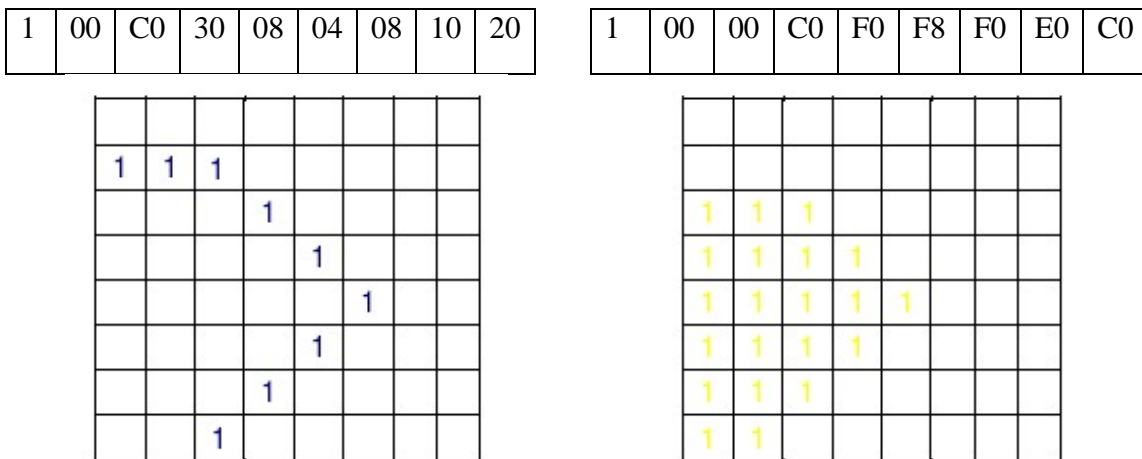


Fig.4 Example of a 8x8 block. The edge of the pac-man is blue and the inside in filling in yellow.

We chose to create pac-man as a 16x16 pixel character. Making our character 8x8 pixels would take away from our graphics capabilities, therefore we enlarged pac-man by giving him a 2x2 square of 8x8 pixel maps. Keeping in mind that pac-man will be facing different directions on each path he takes, graphics must be created for each direction he may take. By splitting pac-man up into a 2x2 square, we now have more of a fluid motion in his journey.

Since the pac-man is not always open its mouth, we design two different graphics for its movement, one is when it open its mouth, the other is when it close its mouth. And when we program them, we will make them change between the two states in a particular frequency.

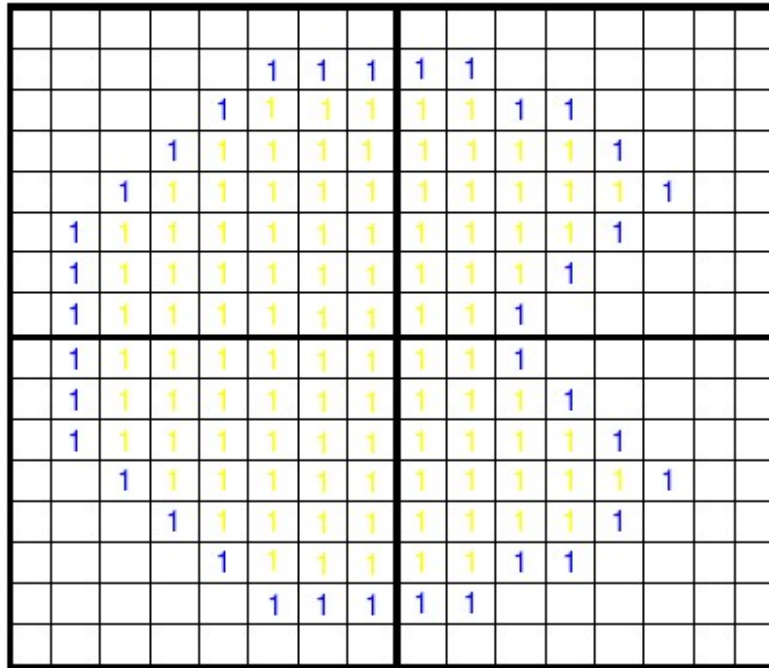


Fig.5 Graphic of pac-man when it open mouth.

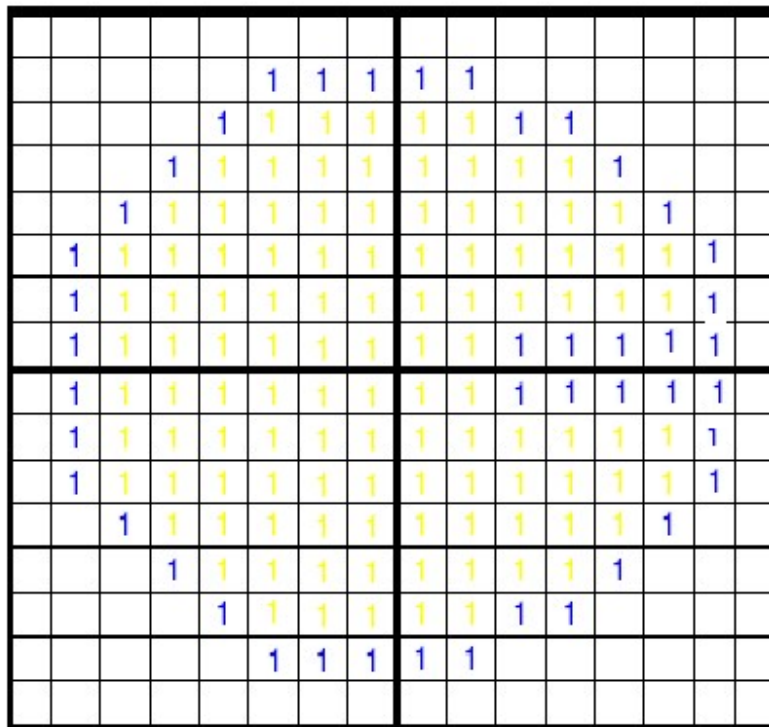


Fig.6 Graphic of pac-man when it close mouth.

## 2.2 Ghost Graphics

As the same method to set the pixels array, we chose to create pac-man as an 18x18 pixel character.

It is the same for the ghosts, besides bouncing to a in a particular direction somewhat like the bounce video ball we programmed in lab, it has two states as they move, one is when it wave its feet to the left, the other to the left. And when we program them, we will make them change between the two states in a particular frequency.

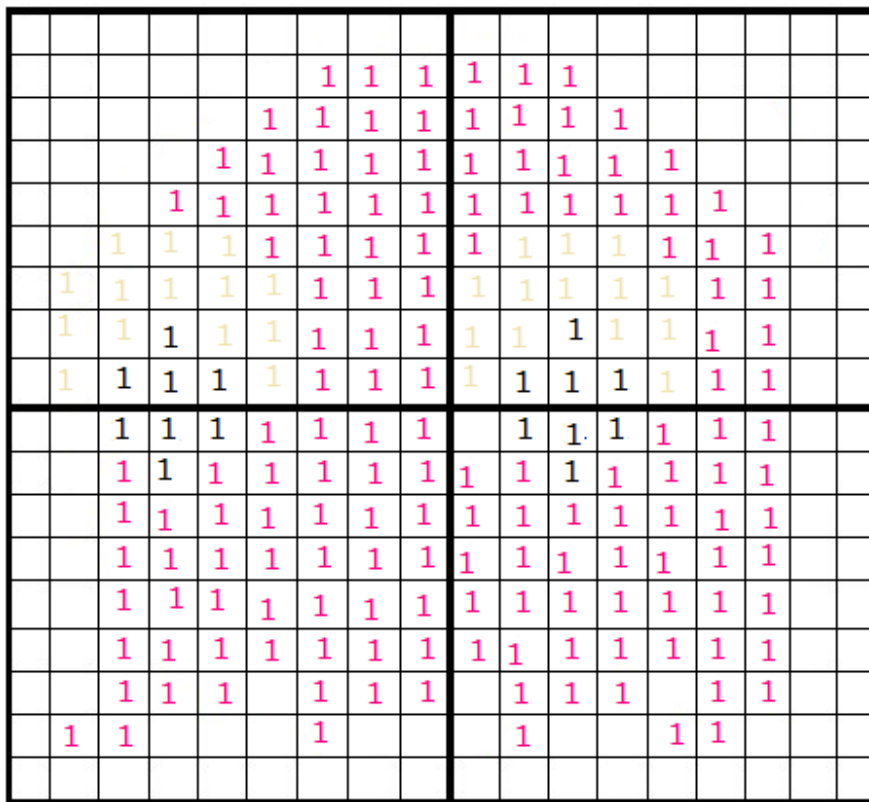


Fig.7 Graphic of ghost when it wave its feet to the left.

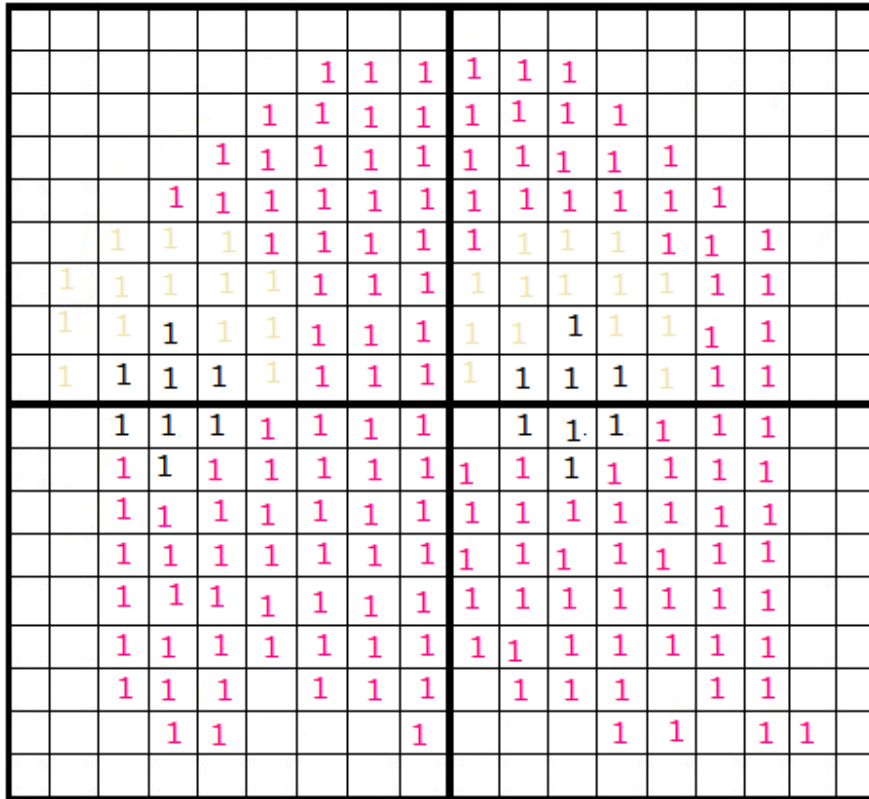


Fig.8 Graphic of ghost when it wave its feet to the right.

### 2.3 Background Graphics

The game background consists of two layers, first with the permanent the walls, and the second with the walls built by pac-man which the pac-man can move freely on while the ghosts can't. Instead of clearing the screen by setting all of the pixels to black, we will clear the screen by redrawing the permanent walls.

### 3. Game Engine

Input will be taken from the keyboard, and we will allow only four keys to do everything in the game. After each clock tick, the most recently suppressed key will determine the direction pac-man will move to. For example, if multiple keys were hit within the duration of one clock cycle only the mostly recently pressed key will register. Holding down a key will not be necessary, as pac-man will continue to move in the same direction, as he was the previous clock tick if no other key was pressed.