北京航空航天大學

Basic Practice on Electrical Technology: Integrated Design

Title：Design of voice-activated snake game based on FPGA

College Name Shenyuan Honors College

Student Name 李谨杰 雷彤彤 赵谦

Student ID 16231235 16231229 16231059

Instructor 黄亚玲

Abstract

FPGAs have certain advantages in game development due to their high speed and high precision features. This article designed a snake game based on the Storm IV_E10_FPGA development platform. In order to improve the interaction and fun of the game, this paper adds sound control based on the keyboard control to control the movement of the snake through the player's voice frequency. The sound control module is implemented by designing an external circuit to convert the sound frequency and then inputs the frequency signal into the FPGA. At the same time, the game incorporates a VGA image display module to enhance the visual experience of the game.

Key words: FPGA, snake game, voice control, VGA

Contents

1. Introduction

FPGA devices have the advantages of high parallelism, high integration, customizability, reconfigurability, and low power consumption. They have high predomination in digital circuit design and are widely used in various fields. Learning FPGA development technology and familiarizing with FPGA development process will enable us to deeply understand and master the relevant knowledge of FPGA and apply it practically.

At the same time, with the rapid development of the game industry, the traditional game development model has been difficult to meet the demand. New game development has higher requirements for the speed, human interaction and confidentiality of the game. The high parallelism of the FPGA can greatly improve the running speed of the game, and the FPGA can enhance the confidentiality of the game. Using FPGAs to develop game can also shorten the game development cycle. Based on the advantages of FPGA, this development model has been accepted by more and more game developers.

Based on the core development of the Snake game using FPGA, we added external circuits to realize the control by sound to the game. This way, using a sound instead of a keyboard to operate the game can greatly enhance the fun of the game. The FPGA selects between keyboard input and external circuit input to ensure keyboard operation compatibility.

2. Functional description

We designed a snake game based on the Storm IV_E10_FPGA development platform and VHDL language, which has the following functions:

1) There are two control modes, one is the keyboard input on the development board, and the other is the frequency meter input. When using the key input, the four keys represent clear, start, left, and right. The player can control the movement of the snake by pressing the buttons to eat the mouse. When using the digital cymometer input, the frequency of the person's sound can be converted into an electric signal by the amplification module and the frequency meter, and turning left, turning right directions of the snake motion direction can be controlled by different frequencies of the sound. For instance, when a player speaks in low-pitched voice, snake turns left; in high-pitched voice, snake turns right.
2) Display the output with VGA. The game is displayed on the screen through an external monitor connected to the VGA port. Among them, the snake head, the body, the mouse and the wall are represented by different colors. Due to the limitations of the hardware platform, our display can only display 8 colors.
3) We use the seven-segment digital tubes on the development board to display the

current score. The range of the score is from 0 to 20.

4) The snake moves on the screen. When the mouse is eaten, the snake becomes longer, and the next mouse is generated simultaneously. When the snake bites into itself, the game fails.
5) The full score of the game is 20 points. When the score reaches 20, the game passes.

3. Scheme design

There are two modes of input. One mode is to input directly from the button which give the left and right control signals. This part is relatively simple and will not be described again. Another mode is that the digital cymometer converts the person's voice signal into a binary number of frequency, then give this number to the FPGA. The digital cymometer input the frequency, and buttons input directions to the FPGA. The control program judges the input signal and control the movement of the snake. The control program controls the progress of the entire game, and the image is output to the monitor through the VGA interface, and the score is displayed on the digital tube.

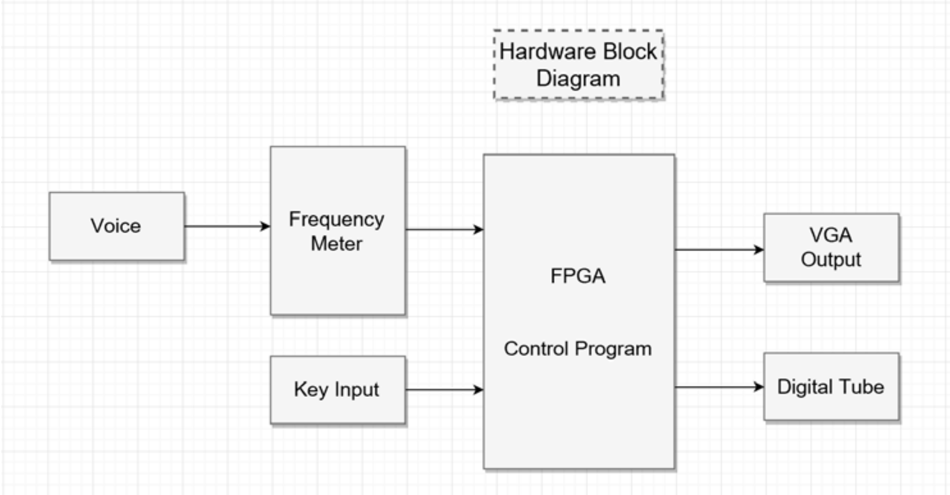


Figure 3.1 Hardware block diagram

3.1 Input part

3.1.1 General statement

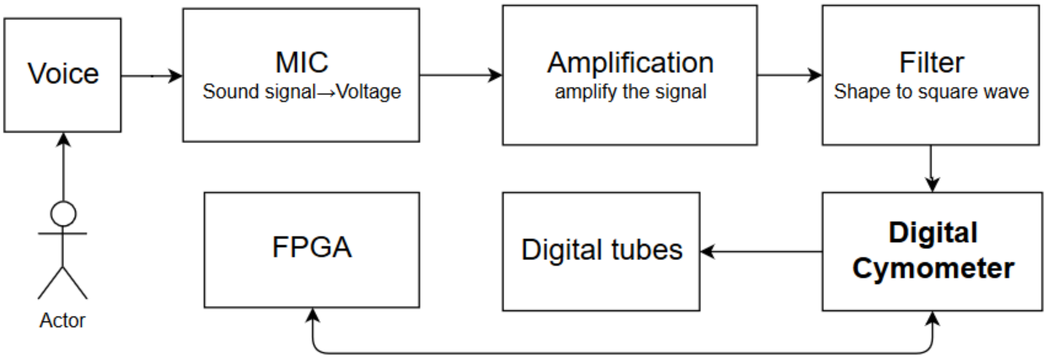


Figure 3.2 Input hardware design block diagram

The overall principle is shown in the figure 3.1.

The player makes a sound and converts it into a vibrating voltage signal through a microphone. The voltage signal is amplified to a magnitude of a few volts so that it can be transformed into a square wave by the shaping circuit. The square wave signal is send to the digital cymometer, and the statistical frequency is displayed on the digital tube and send to the FPGA. When the falling edge signal of the gate signal is detected, the FPGA reads number of frequency, controls the action of the snake according to the frequency, and then output clear signal. The clear signal will clear the counter, preparing for the next counting cycle, and continuously reciprocates. Finally, the entire circuit samples the frequency value every 0.5s and sends it to the FPGA.

3.1.2 MIC, amplification and filter part

1) MIC and amplification

The human voice consists of sound made by a human being using the vocal tract, such as talking, singing, laughing, crying, screaming, etc. The human voice frequency is specifically a part of human sound production in which the vocal folds (vocal cords) are the primary sound source. [1] The empirical value of the sound frequency is shown in the table 3.1

	Children	Male	Female
Bass	196~698.5Hz	24.5~349.2Hz	174.6~784Hz
High pitch	261.6~880Hz	110~523.3Hz	220~1046.5Hz

Table 3.1 Frequency of human voice [2]

MIC (microphone) can convert sound signals into electrical signals. MIC has various types to choose, in this project we choose electret condenser MIC.[3] In electret condenser MIC, the externally applied charge under condenser microphones is replaced by a permanent charge in an electret material. An electret is a ferroelectric material that has been permanently electrically charged or polarized. Electret condenser MIC has good performance and ease of manufacture, so, it is mostly used around the world, from every part of our daily life.

The MIC needs to drive the power supply, and its output voltage is about a few millivolts to a few dozen millivolts. The typical application circuit has been shown on the datasheet (figure 3.3). RL is 2.2kΩ, +Vs varies from 1V to 10V, and I choose 5V here.

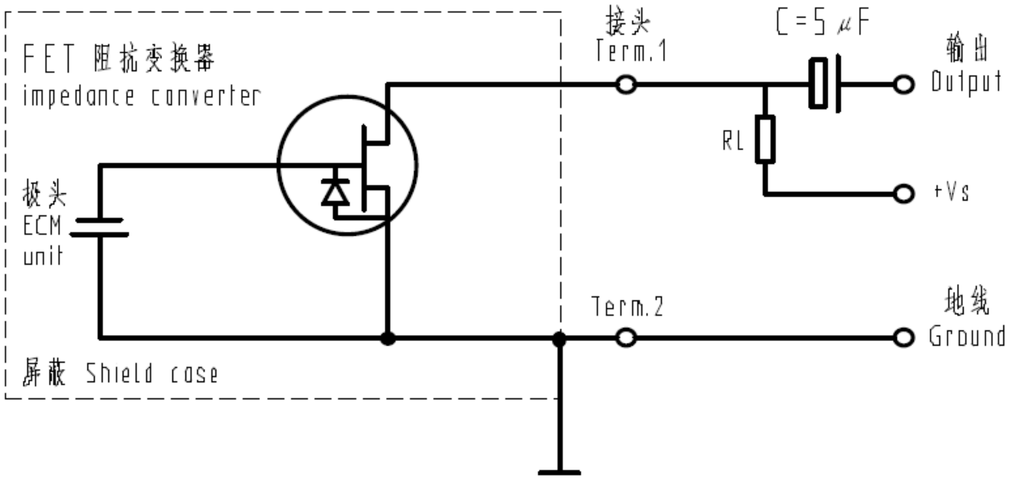


Figure 3.3 Typical application circuit [4]

The output voltage of this circuit is very small (a few mV to a dozen mV), and I need to amplify it to a few V, so that the digital cymometer can measure the diversification. The amplifier circuit I designed is shown in Figure 3.4.

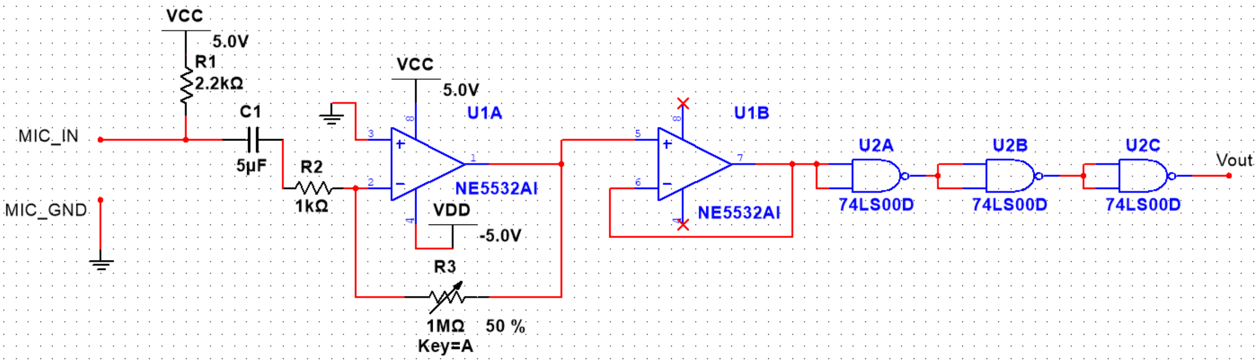


Figure 3.4 Amplification circuit

First operational amplifier amplifies input signal. And amplify ratio can be calculated as:

$$A = -\frac{R_3}{R_2}$$

Second op works as voltage follower.

2) Square wave shaping circuit

The collected voltage signal is shown in Figure 3.5.

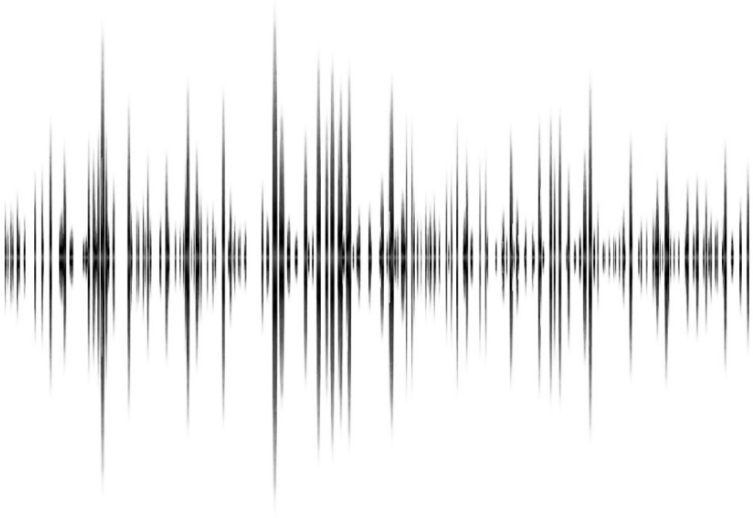


Figure 3.5 waveform of voice [5]

I use a 74LS00 to shape this wave into equal amplitude square wave, just like figure 3.6.

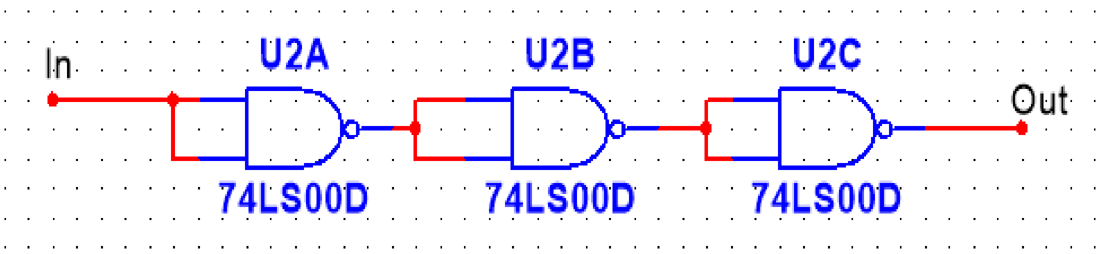


Figure 3.6

The range of voice frequency is from 60Hz to 800Hz, requires response time shorter than 1.25ms [6]. With response time of 15ns max, 74LS00 fulfils requirement.

3.1.2 digital cymometer

[7]The principle of digital cymometer has been taught in class. Figure 3.7 shows the principle block diagram of a simple digital cymometer. It comprises following four basic parts:

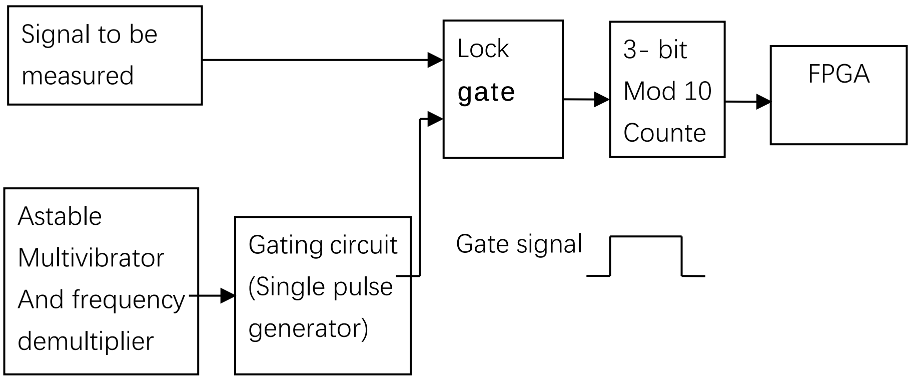

Signal to be measured
Lock gate
3- bit Mod 10 Counte
FPGA
Astable Multivibrator And frequency demultiplier
Gating circuit (Single pulse generator)
Gate signal

Figure 3.7 Principle Block Diagram of a Digital Cymometer

a) Lock gate

Lock gate consists of gate circuits with the pulses to be countered as one input and gate signal as another input, and the gate signal controls it to open or close. I choose 74LS00 and 74LS04 to construct a non-gate, combining signals from astable multivibrator and gating control circuit.

b) Astable multivibrator

Normally, crystal oscillator and frequency demultiplier can be used to get frequency known and steady oscillation whilst the latter dividing the signal frequency to change the width of gate signal. Here a ring oscillator with RC circuit is used as figure 3.8. Potentiometer R1 needs adjusting to make clock frequency(CLK1) equals 4Hz. Frequency can be calculated as the formula below:

$$f = \frac{1}{2.2RC}$$

74LS74 is dual-type positive edge-triggered flip-flop, it can divide 4Hz to 2Hz and output from CLK2.

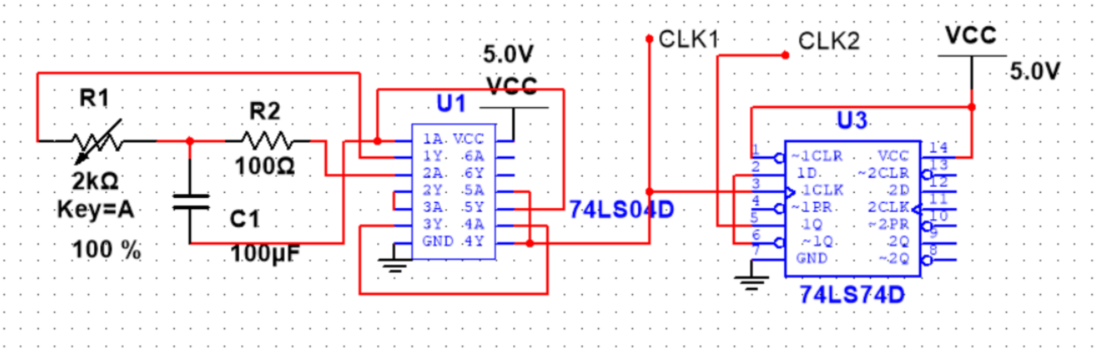


Figure 3.8 Astable multivibrator and frequency division

c) Gating control circuit

It is used to change the periodic signal from frequency demultiplier into single pulse signal called gate signal. In this circuit, CLK1 is used to control gate width (0.25s,4Hz), and CLK2 is used to trigger measurement (0.5s 2Hz). That means whole system would measure the frequency of voice per 0.5 second. Specific circuit is displayed as figure 3.9.

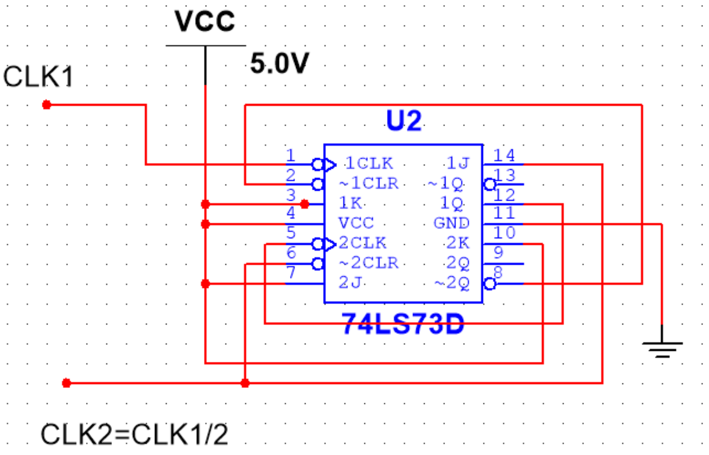


Figure 3.9 gating control circuit

d) Counter and circuit

It counts the pulses from lock gate, and then transfer count number to FPGA. Frequency defines as the cycle number of a periodic signal in 1 second. If the circulation number is N in 1 second, the frequency f = N.

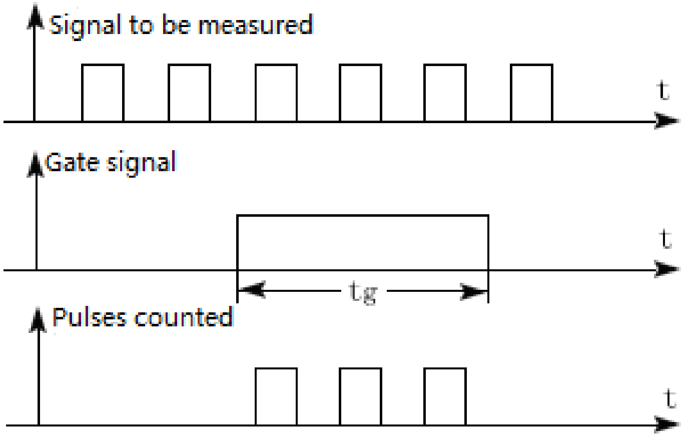


Figure 3.10 Sketch Map of Measuring Frequency Principle

Figure 3.10 illustrates the principle of measuring frequency. Only when gate signal is HIGH can the measured signal passes lock gate and count by the counter. When gate signal is LOW, the lock gate closes and counting stops. If time width of gate signal is 0.25s, the frequency to be measured is

$$f = \frac{N}{0.25\mathrm{s}} = 4N$$

Therein, N is the number counted by the counter.

Count circuit is designed as follow:

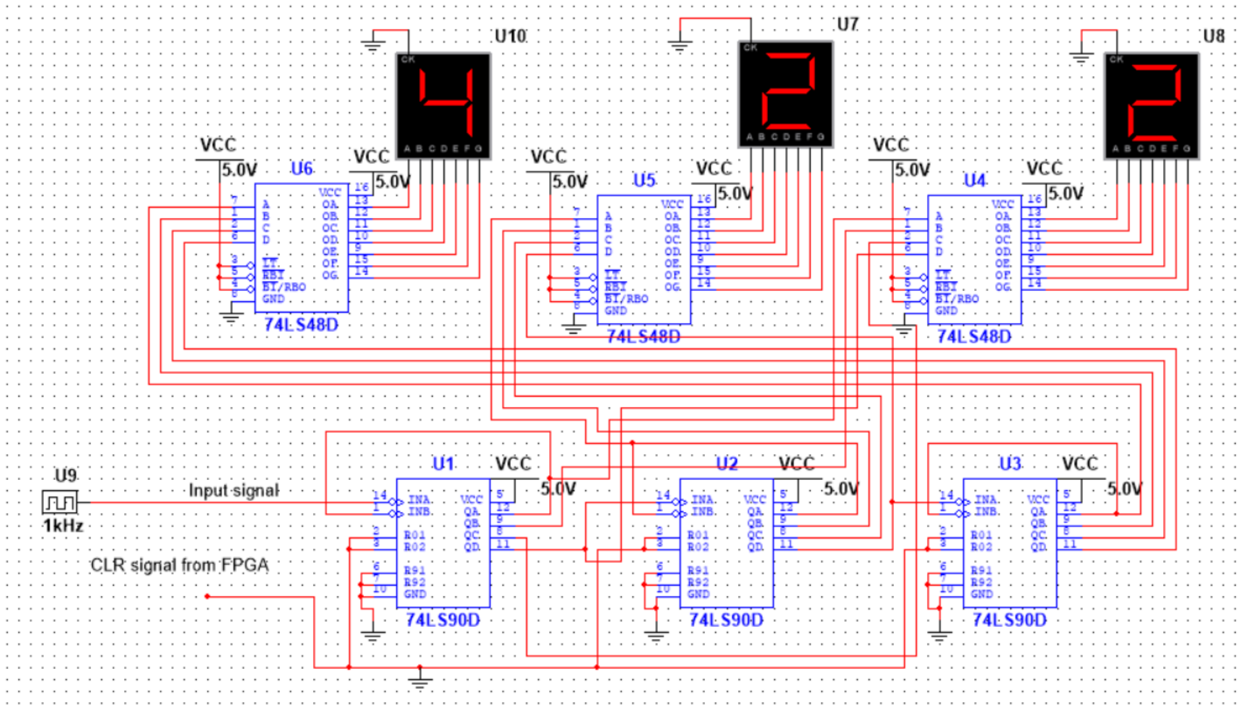


Figure 3.11 counting and display circuit

In this circuit, three IC74LS90s are connected to construct a 3 bits mod 10 counter. The number is recorded as BCD code, and can be read from 7-segments digital tubes. At the same time, when gate closes, the BCD code will be read by FPGA. The FPGA carries out next step based on frequency andSignal give clear signal before next gate opening period.

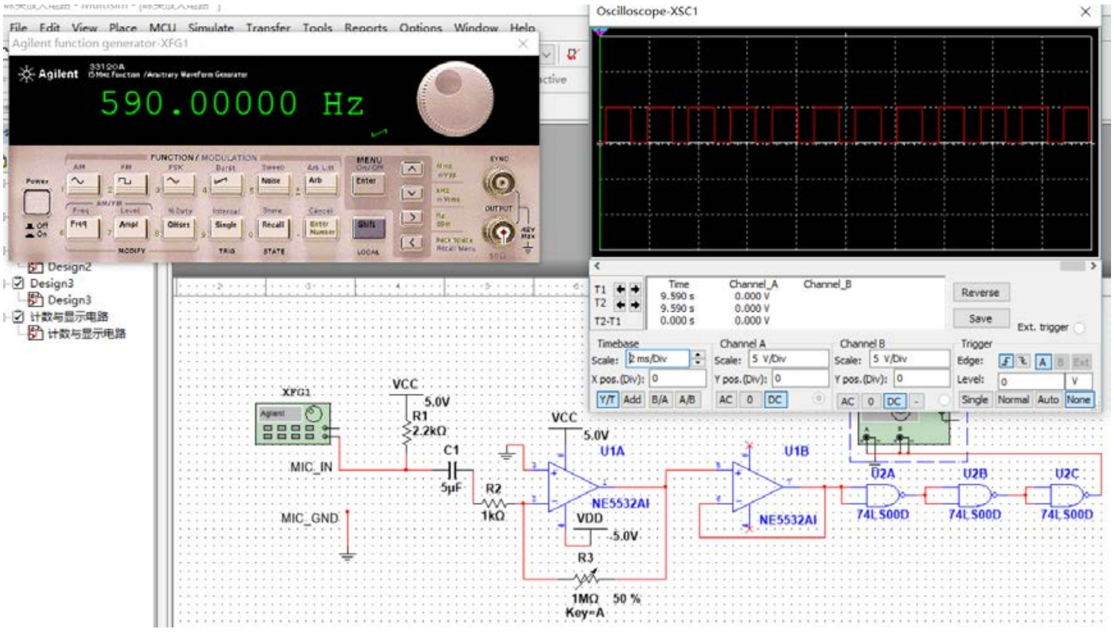
3.1.3 Points to be tested and expected values

1. Wire amplification part, when input a few of mV to a dozen of mV sin signal, the output waveform should be square wave and its frequency should be as same as the origin.
2. Build the astable oscillator by NOT gate, resistors and capacitors. Frequency should be adjusted to 4Hz.
3. Build the whole circuit, input sin signal in test1 and observe numbers on digital tubes. Count number should be f/4.

3.1.4 Test scheme

We don’t have time to print PCB boards, so I make some simulations instead. Multism always crashes when I simulate after connecting the whole circuit, so I separate them to test. And most of the designs have been proven by our experiment of simple digital cymometer, so, the major simulation is about amplification part and display part.

1) Simulation of amplification is record as figure 3.12.



Agilent function generator-XFG1

590.00000 Hz

Oscilloscope-XSC1

	Time	Channel_A	Channel_B
T1	9.590 s	0.000 V	
T2	9.590 s	0.000 V	
T2-T1	0.000 s	0.000 V	

Timebase Scale: 2 ms/Div X pos.(Div): 0
Channel A Scale: 5 V/Div Y pos.(Div): 0 DC
Channel B Scale: 5 V/Div Y pos.(Div): 0 DC
Trigger Level: 0 V None

Figure 3.12 simulation of amplification and result

The frequency of voice varies from 60Hz to 1000Hz, I choose 590Hz as the average. Actually, proper frequency should be adjusted based on actual circuit. The input amplitude is 50.00mVpp, which is the smallest value this function generation can give. I choose sawtooth wave, in order to simulate voice signal.
From simulation result, this circuit works well obviously.

2) Simulation of display circuit is recorded as figure 3.13:

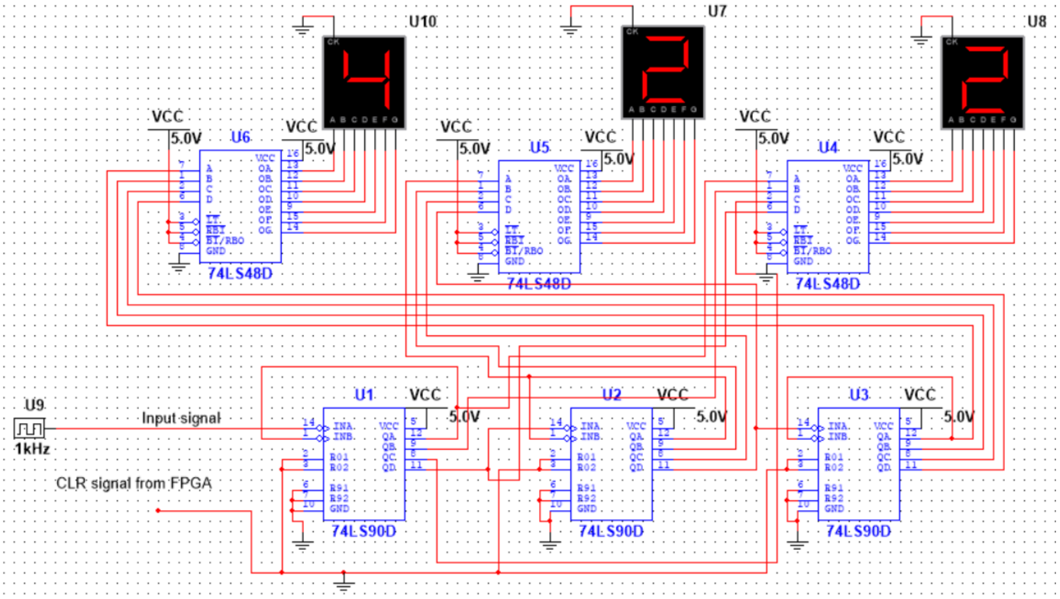


Figure 3.13 simulation of display circuit

When simulation begins, numbers jump one by one, from 0 to 9. So, this circuit satisfies design demand.
Next, we are going to make a PCB board and debug the actual circuit. The circuit design schematic and PCB diagram are as figure 3.14 and figure 3.15.

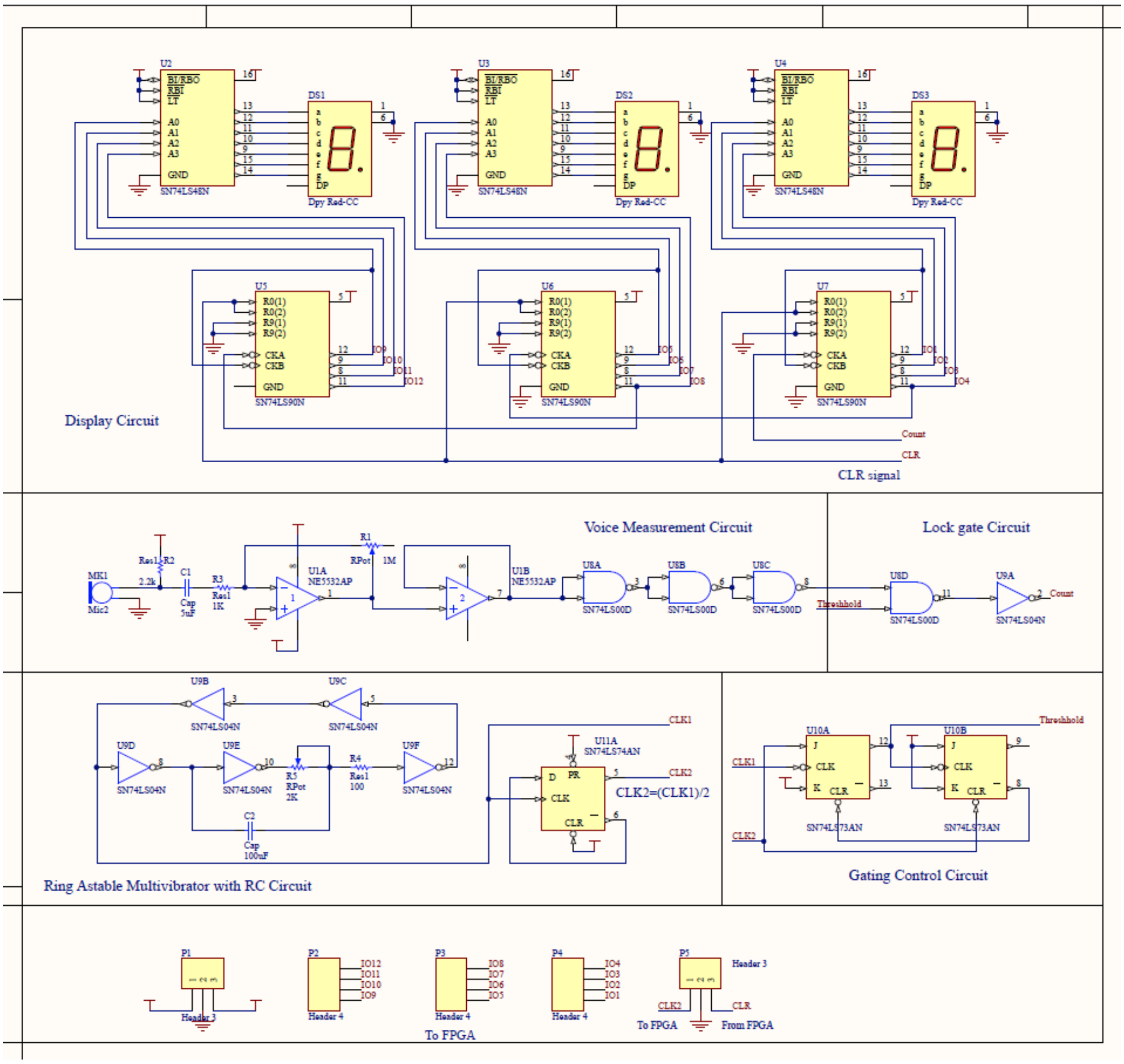


Figure3.14 Schematic

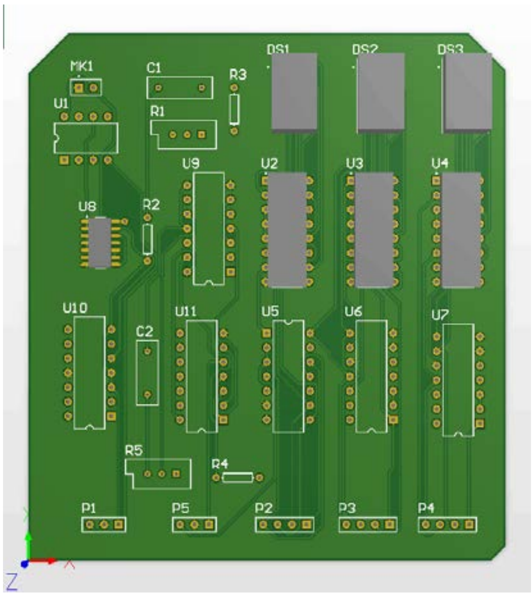


Figure 3.15 PCB Board

3.2 Control program

3.2.1 Direction control

The control direction can be judged by the signal given by buttons or the digital cymometer. Two buttons represent left and right and the frequency meter decides the direction according to the frequency of the sound emitted by the person. After receiving the control signal, the program will make a logical judgment on it. First judge the direction of motion of the snake relative to the plane, then calculate the direction of the snake after turning left or right.

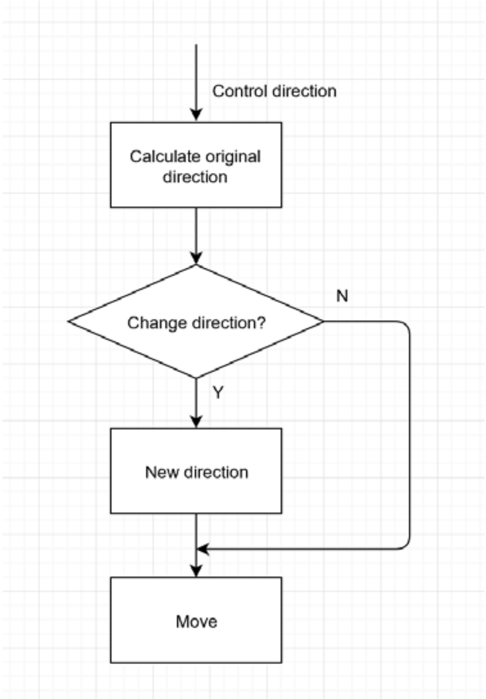


Figure 3.16 Logical block diagram of direction control

3.2.2 The movement and eating of the snake

After the data is read from the direction control module, the direction of movement of the snake will change accordingly. The snake is divided into the head and the body. The snake head keeps advancing along the direction of control. The coordinates of each part of the snake body become the coordinates of the previous part. A counter can be used to control how fast the snake moves. When the snake head coordinates coincide with the mouse coordinates, it means that the snake has eaten the mouse. At this time, the scorer addes one, and the body length of the snake is also increased by one unit. And in the next movement of the snake, the coordinates of the snake tail will not change. Next, judge whether the snake has eaten himself. If the coordinates of the snake head coincide with the coordinates of the body, it means that the snake bites into itself and the game is over. Otherwise, the snake continues to move.

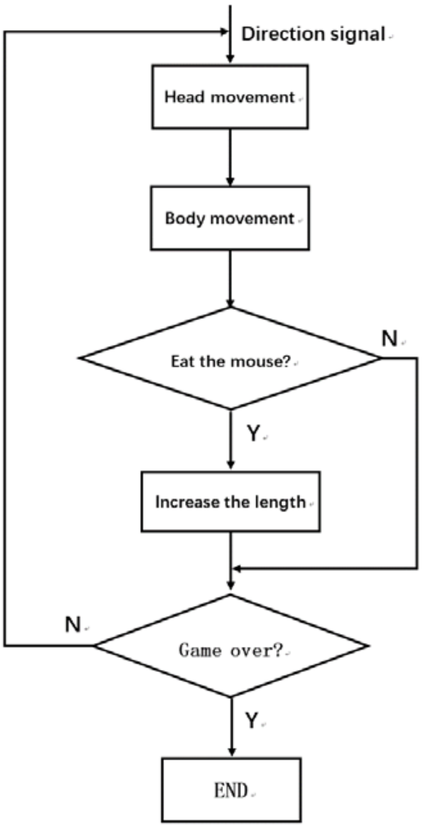


Figure 3.17 Logical block diagram of the movement and eating

3.2.3 Scorer

Every time the snake eats a mouse, the scorer adds one. And the score will be displayed on the digital tube. We have already done the experiment on displaying numbers on digital tubes, so I won't go into details here.

3.2.4 Generate the position of the mouse

After the snake has finished eating the mouse, a set of random numbers is generated as the coordinates of the next mouse by taking the reminder of the counter. Since the counter is constantly changing, the remainder of dividing the counter by 20 can be considered a random number. If the coordinate of the mouse don't coincide with the coordinates of the snake's body, then this coordinate is valid, otherwise another random set of coordinates is generated.

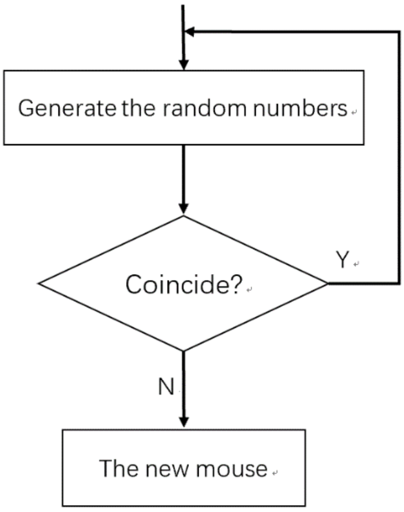


Figure 3.18 Logical block diagram

3.3 VGA module

In this game design process, we use the VGA module to display the game image.

3.3.1 Brief description of the VGA principle

VGA is a video transmission standard that can separate and transmit three primary color analog signals of red, green and blue, and realize the output of color images by superimposing and displaying the different intensity of the three primary color signals. The VGA hardware interface of the Storm IV_E10_FPGA development platform is shown in the figure. Among them, Hs (86) and Vs (87) are synchronous signal (ie, line scan and frame scan signal) inputs for controlling the timing display of the VGA. R, G, B (100, 99, 98) three interfaces are used to input three primary color analog signals. The platform only provides three RGB signal interfaces but it still meets the design requirements.

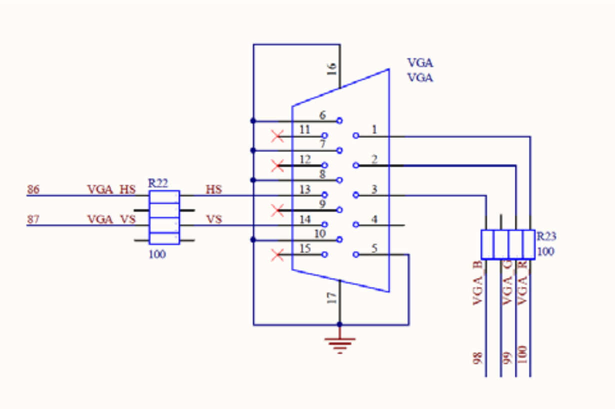


Figure 3.19 VGA hardware interface

The scanning mode of the VGA display is scanning by line, that is, on one display, the electron beam starts scanning from the upper left corner, scanning each pixel from left to right, and then returns to the starting position on the left side of the next line, scanning the next line. Until all the lines are scanned, the scan output of one frame of image is completed. The signal used to control the line scan is the line sync signal, and the signal used to control the frame scan is the field sync signal. The timing diagrams of the two signals are as follows。

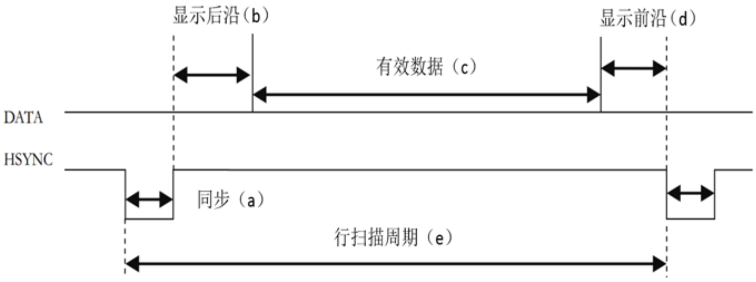


Figure 3.20 Line timing diagrams

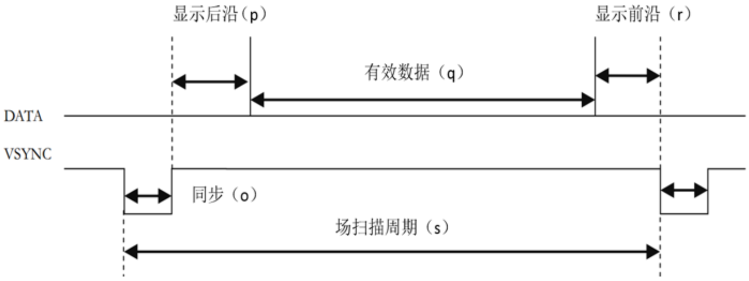


Figure 3.21 Field timing diagram

In a row scan cycle, the sync signal is first experienced. The sync signal is used to reset the scan address, ie to return to the beginning of a line. At this time, HSYNC is low. After a display trailing edge (for stabilizing the scan signal), the output of valid data is performed. Finally, after a display leading edge (preparing for the transfer of the scan address), the next round of scanning is started. The field scan period is the same. The line scan period is in units of pixel clocks, while the field scan period is in units of line scan periods.

3.3.2 VGA display design

In order to adapt to most monitors, in this game, we use 640*480@60 display mode, that is, the screen effectively displays 640 (rows) * 480 (columns) pixels, and the image is refreshed 60 times per second. The required drive clock is 25.175 MHz, and we approximate the 25 MHz clock signal.

Display mode	Clock (MHz)	Line timing (number of pixels)					Frame timing (number of lines)				
		a	b	c	d	e	o	p	q	r	s
640*480@60	25.175	96	48	640	16	800	2	33	480	10	525

Table 3.2 display mode

The VGA module design is mainly divided into the following parts:

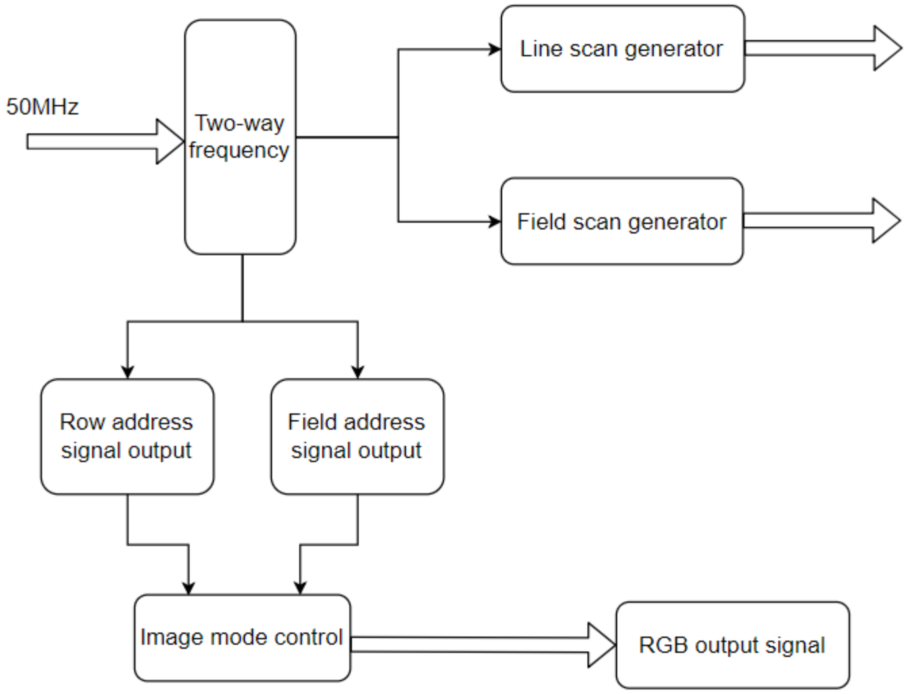


Figure 3.22 VGA module design

(1) Generate a scan drive clock for VGA display.
Since the development platform has a built-in 50MHz clock signal, it is only necessary to divide the signal by two to obtain a VGA drive clock signal.

(2) Generate the line timing signal and the field timing signal.
The number of pixel clocks required to scan each line is calculated according to the display standard, and is set as a line period. The control line timing signal sequentially scans the pixels of each line. One pixel is completed per scan, and the line counter is incremented by one. When in the synchronization phase, the HSYNC signal is controlled to be low, and the synchronization of the signals is completed. After scanning a line, the line signal is cleared to start the next line of scanning.

The field signal is scanned sequentially for each frame. The number of line periods required to scan each frame is calculated according to the display standard, and is set as the field period. The control field timing signal sequentially scans each line. One line is completed per scan, and the line counter is incremented by one. When in the synchronization phase, the VSYNC signal is controlled to be low, and the synchronization of the signals is completed. After scanning one frame, the field signal is cleared to start scanning for the next frame.

(3) Control the RGB number to display the wall, the snake body and the randomly generated red dot.(The red dot stands for a mouse)

The wall is the border of the game interface; the snake body is a user-controlled object that moves within the limits defined by the wall; red dots are randomly generated in the game area.

The VGA display mode is basically divided into three types. The first one is to directly operate the pixel point drawing, the second is to store the picture in the ROM and then call it. The last one is to first generate an image or character in the RAM and then call it. Since the features we need to display are very simple, we take a direct manipulation of the pixel plot to display.

The VGA display block diagram is below. We use the line counter and the field counter to determine which part of the wall, the snake body, the background, and the red dot that the currently scanning pixel belongs to. When the scan signal judges that the current scanning pixel is the wall, the RGB numbers are controlled to be 0,0,1, outputting blue; if judged as snake body, the RGB numbers are controlled to be 1,1,1, and the output is white; if it is judged as red dot, the RGB numbers are controlled to be 1,0,0, outputting red; when it is judged as background, the RGB numbers are controlled to be 0, 0, 0, that is, the black background is output.

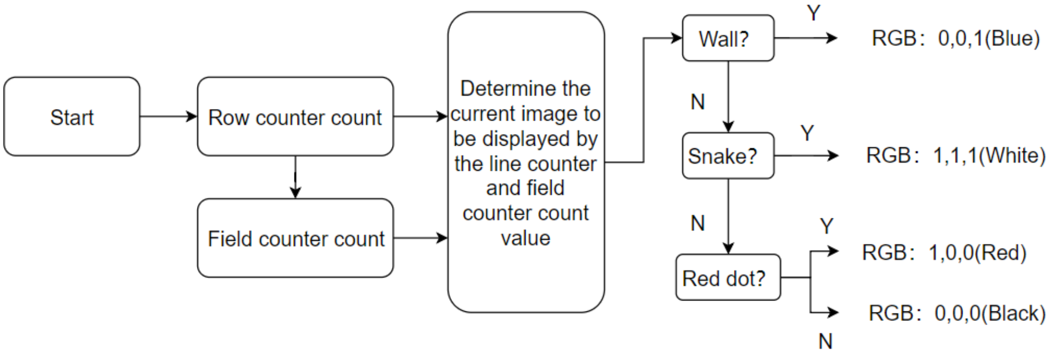


Figure 3.23 The VGA display block diagram

4. Project division

Li Jinjie is responsible for the sound control mode.
Zhao Qian is responsible for control program and FPGA debugging.
Lei Tongtong is responsible for VGA display output module and FPGA debugging.

5. Analysis and Conclusion

At present, the game has only completed the theoretical design, circuit construction and debugging are still in progress. There are still many shortcomings in the design. The game mode is too simple， more game levels and difficulty levels can be added in the future. Originally, we planned to use somatosensory input, but due to the failure of the program solution, and then we have switched to sound control. The implementation of somatosensory control can continue to be explored in future improvements. At the same time, due to hardware limitations, VGA color display is relatively simple.

We designed a voice-controlled snake game based on FPGA platform and VHDL language. The project combines software and hardware. We write the game control program through software, and realize voice input through hardware. We not only used the knowledge learned in previous FPGA experiments, and get familiarized with the VHDL language, but also combined with the knowledge of digital circuits to design the frequency meter circuit, which makes good use of many aspects of knowledge. This is not only a review of the knowledge that has already been learned, but also expands our knowledge and lays a good foundation for the future. In addition, we use VGA to display the output and extend the peripheral interface of the FPGA. Because the project is a game, it is very interesting, and incorporates practice into fun.

References

[1] Wikipedia human voice

[2] 周明军.合唱人声的拾取方法[J].音响技术,2007,(6):48-50. DOI:10.3969/j.issn.1008-1003.2007.06.016.

[3] wikipediahttps://en.wikipedia.org/wiki/Microphone#Liquid

[4] datasheet of TYPE：“XL”Electret Condenser Microphone

[5] 图虫创意 https://stock.tuchong.com/image?imageId=574118328765019O7&source=baiduimage

[6] datasheet of 74ls00

[7] BASIC PRACTICE On ELECTRICAL TECHNOLOGY (II)

[8] 艾兵. 基于 FPGA 的体感游戏[J]. 现代电子技术, 2014(6):47-50.

[9] 张亚平, 贺占庄. 基于 FPGA 的 VGA 显示模块设计[J]. 计算机技术与发展, 2007, 17(6):242-245.

[10] 黄柯, 刘昌华. 基于 FPGA 的入侵者游戏设计[J]. 信息通信, 2018(1):138-139.

[11] 杨金. 基于 FPGA 的 VGA 游戏设计[J]. 数字技术与应用, 2012(4):171-172.