

用 8*8 点阵实现贪吃蛇游戏

傅丽琴

徐楠楠

中国海洋大学

中国海洋大学

15726204330@163.com

15762264727@163.com

摘要

本实验以设计贪吃蛇游戏为课题，用 VHDL 语言实现。最终结果是以 8*8 点阵作为本游戏的显示界面，通过四个按键分别控制蛇的运动方向，分别为“上下左右”。蛇的长度为 7 个点阵，另外以一点表示老鼠，当蛇头吃到老鼠时，老鼠则在点阵上改变位置，也就是说通过一个伪随机函数在点阵上产生老鼠的位置，每当蛇头吃到老鼠时，就自动加一分。每隔 15 秒，在蛇没有吃到老鼠的前提下，老鼠就自动在点阵上改变位置，这样就增加了游戏的趣味性和难度性。

关键字：VHDL，8*8 点阵，贪吃蛇

第一章 引言

ISE 是 Xilinx 公司提供的一套集成开发工具, 它支持几乎所有的 Xilinx 公司的 FPGA/CPLD 主流产品, 并完成整个 FPGA/CPLD 的开发过程, 包括从设计输入到综合、布线、仿真、下载的全套解决方案, 并很方便的同其他 EDA 工具接口。ISE 将先进的技术与灵活性、易使用性的图形界面结合在一起, 不管使用者的经验如何, 都能在最短的时间, 以最少的努力, 达到最佳的设计。ISE 的集成工具主要包括设计输入工具、综合工具、仿真工具、实现工具和辅助设计工具等五类。

而 1995 年中国国家技术监督局组织编撰并出版《CAD 通用技术规范》, 推荐 VHDL 语言作为我国电子设计自动化硬件描述语言的国家标准。

将硬件描述语言 VHDL 于 ISE 软件上运用, 可以加深学生对硬件描述语言 VHDL 的理解, 而本项目所选课题“贪吃蛇”游戏, 不仅可以巩固学生课上所学知识, 而且该项目对 VHDL 初学者来说稍显复杂, 增加了学生于项目进行时的挑战性与兴趣性。

第二章 方法

2.1 总体框图

2.1.1 系统结构图

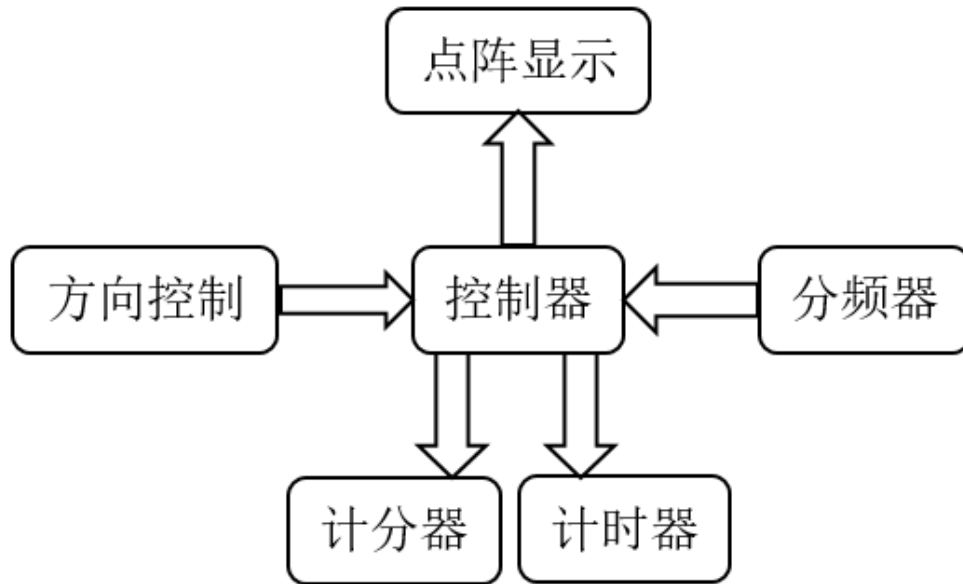


图 2.1

2.1.2 流程图

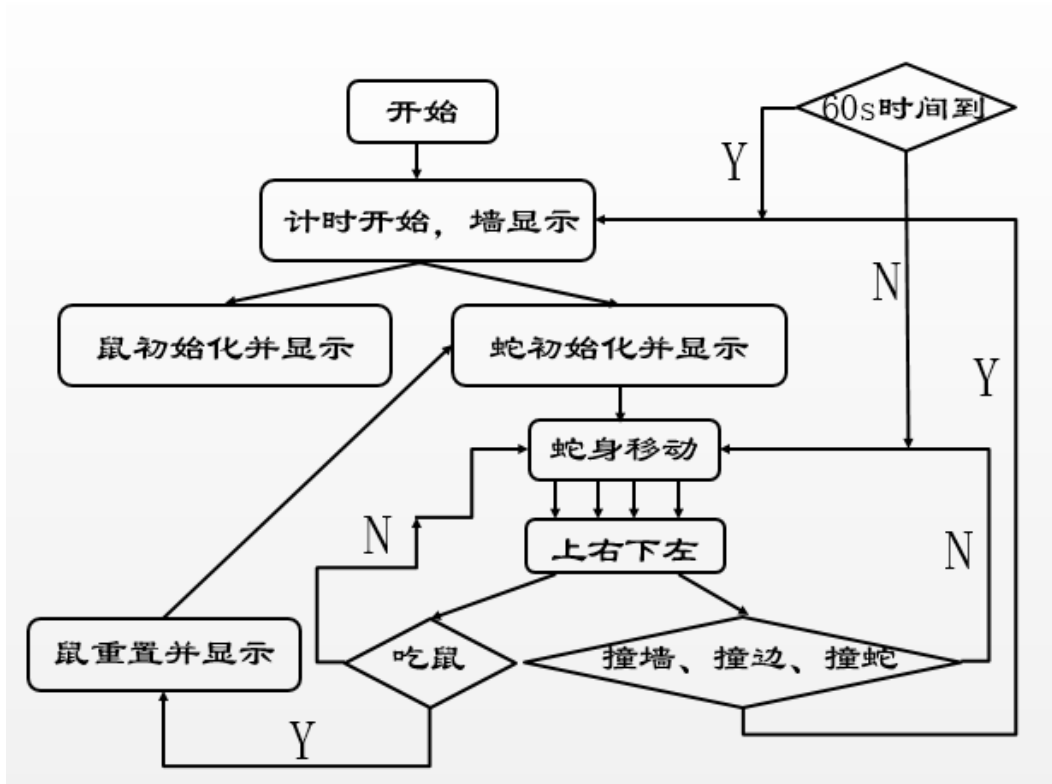


图 2.2

2.2 模块设计

2.2.1 分频模块

1. 下载板中的 LED 灯的显示和蛇移动需要 1Hz 的时钟, 因此将下载板中自带的 50MHz 时钟采用 5×10^6 分频, 产生频率为 1Hz 的 clk_1, 这样可以保证倒计时正常显示以及蛇的正常移动。
2. 下载板中的点阵需要显示, 并且所有状态不能同时显示, 故采用 10^5 分频, 产生频率为 5KHz 的 clk_5k, 用循环扫描, 以看到较为稳定的画面。

2.2.2 时间显示模块

倒计时从 59s 开始, 到 0 结束。以 clk_1 为时钟, 在 clk_1 上升沿到来时, 令时钟高位

T_HIGH 和时钟低位 T_LOW 进行相应变化，使总倒计时减一。

2.2.3 蛇身显示模块

蛇身各点用 8*8 点阵以 clk_5k 为时钟进行循环扫描进行显示。由于点阵上的点亮的条件是行 ROW 为高电平 '1'，列 COL 为低电平 '0' 时，则对应的点就被点亮。因此令 ROW 中与 TY 对应的行坐标赋值为 '1'，COL 中与 TX 对应的列坐标赋值为 '0'，其他的行列坐标应分别赋值 '0' 及 '1'。

2.2.4 控制模块

1. 控制蛇的运行或死亡状态。当 RESET=1 时：当蛇处于死亡状态，游戏进行初始化。
当蛇撞墙、撞边、撞蛇或总计时时间到，蛇转到死亡状态，游戏重新开始。
2. 控制随机鼠的放置。当蛇处于死亡状态、老鼠放置时间 15S 到或蛇把老鼠吃掉，老鼠自动重置。重置老鼠为伪随机函数，并且令鼠的横坐标不能 5，以保证鼠与墙不会重叠。
3. 控制蛇身移动方向。通过 4 个按键来控制蛇的移动方向，并且用 TURN 表示下一秒蛇的有效移动方向，即蛇头的坐标变化，其余点则将前一个点的值赋给后一个点。
4. 控制蛇身长度。当蛇吃到鼠，并且未死亡时，总分 SCORE 加一。通过总分的数值来确定蛇身长度。

第三章 实验分析

3.1 波形仿真及分析

3.1.1 点阵的显示

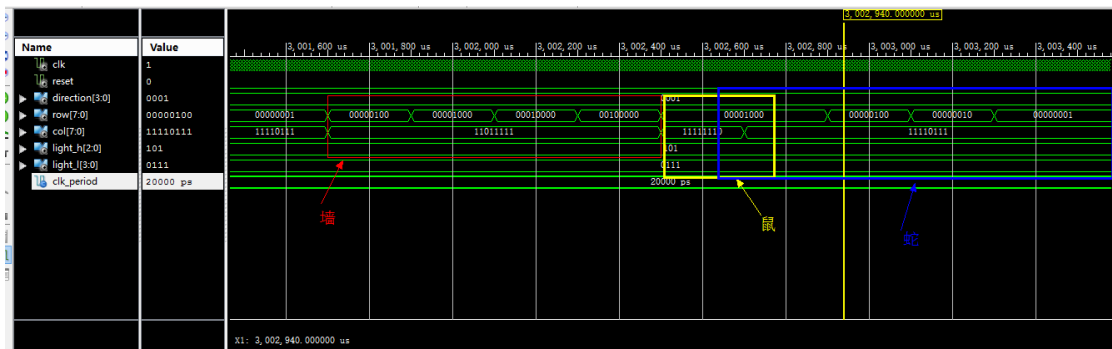


图 3.1

3.1.2 复位键 Reset 的作用

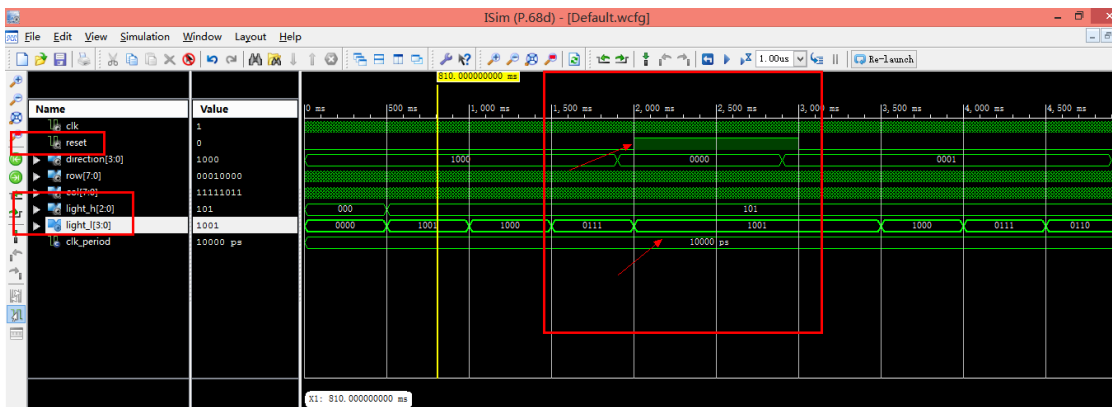


图 3.2

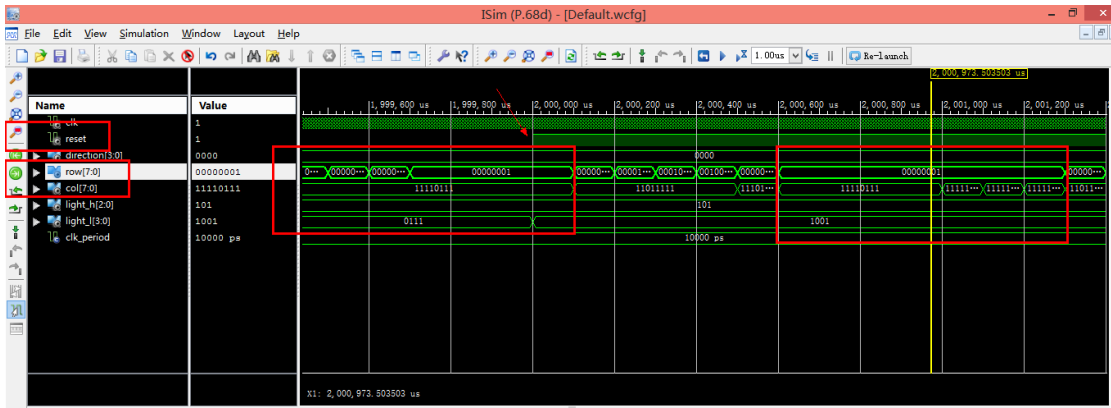


图 3.3

3.1.3 时间显示模块

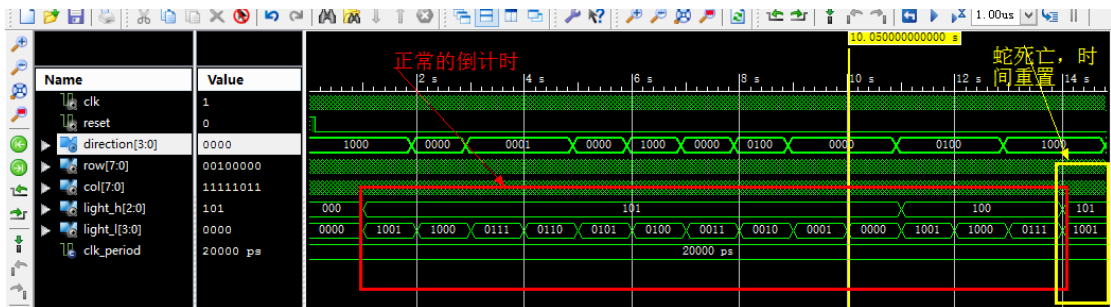


图 3.4

3.2 实验错误分析

在项目进行中，我们遇到了以下几个问题并在组员讨论和请教别人后成功解决。

1. 在方向控制最初的算法设计时，出现了蛇相向运动的状态，如在向右运动的时，若按向左键蛇会向左运动，蛇身变短。

解决方案：增加了一个 signal：TURN 来控制蛇的运动。只有当输入方向信号生效时才会改变 TURN，从而解决这一错误的发生。

2. 在鼠的随机坐标赋值时，出现了鼠的坐标与墙重合的问题。

解决方案：因为墙的 wx 固定为 5，因此设置 TX=4 时，让 TX 的下一个随机值为 0。

在其他情况下 $TX \leq TX+1$ 。

3. 仿真波形输出时在点阵显示了很多不应该出现的点,后来分析之后发现这是因为显示模块中老鼠、墙和蛇同时扫描,使点阵接通所有被赋值的行列坐标。最后我们使用分频扫描后,点阵显示正常。

第四章 总结

解决了以上问题之后,我们较顺利的完成了此次项目,我们也得到很多收获。首先,第一次尝试一个较复杂项目的设计,从中学习到了很多知识,加深了对 VHDL 这门课程的理解。

其次,虽然没有设计出成品,但是从开始到仿真的过程中也克服了许多困难,比如安装软件和不断改错修改调试等,我们深刻感受到静下心来完成一件事是件很有成就感的事,这也增强了自己对本专业的学习兴趣。

总之,与同学合作,可以取长补短,互相给予灵感共同完成项目,是个很好的体验,收获很大。

第五章 作者贡献

在前期准备中,本小组成员傅丽琴负责软件的安装、调试。徐楠楠同学负责资料的查找筛选。

在程序编写的过程中,傅丽琴同学负责控制模块和蛇身显示模块。徐楠楠同学负责分频模块和时间显示模块。在程序调试阶段,本项目两位同学共同进行波形仿真,并对其中出现问题进行讨论分析。

致谢

首先,感谢郑海永老师给我们的这次机会,在有限的时间内,完成一个小项目,大大提高了本组成员的学习效率与效果。其次,感谢王振宇同学在 piazza 平台上的解答,使 ise14.6 能够在 win8.1 的操作系统中运行。

参考文献

《ISE 集成开发环境下基于 FPGA 的数字设计》曾祥萍, 赵海全

《VHDL 与仿真工具的应用》王观凤 王礼平

附录

程序源代码

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY snack IS
PORT ( CLK:IN STD_LOGIC;
      RESET:IN STD_LOGIC;
      DIRECTION:IN STD_LOGIC_VECTOR(3 DOWNTO 0);
      ROW:OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
      COL:OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
      LIGHT_H:OUT STD_LOGIC_VECTOR(2 DOWNTO 0);
      LIGHT_L:OUT STD_LOGIC_VECTOR(3 DOWNTO 0)
);
END snack;

ARCHITECTURE Behavioral OF snack IS
```

```

SIGNAL CLK_1,CLK_5K:STD_LOGIC;                --1hz,5khz 时钟信号

SIGNAL DEATH:STD_LOGIC;                        --控制蛇死亡状态变量

SIGNAL NUM:INTEGER RANGE 0 TO 12;              --显示分频参数

SIGNAL T_HIGH:INTEGER RANGE 0 TO 5;            --倒计时高位

SIGNAL T_LOW:INTEGER RANGE 0 TO 9;             --倒计时低位

SIGNAL T_MOUTH:INTEGER RANGE 0 TO 15;          --鼠放置倒计时

SIGNAL SCORE:INTEGER RANGE 0 TO 3;             --比分

SIGNAL TURN:INTEGER RANGE 0 TO 3;             --方向控制

SIGNAL COUNT_1:INTEGER RANGE 0 TO 50000000;    --1hz 分频参数

SIGNAL COUNT_5K:INTEGER RANGE 0 TO 10000;      --5khz 分频参数

TYPE DIAN IS ARRAY(INTEGER RANGE<>) OF INTEGER RANGE 0 TO 7;

SIGNAL SX,SY:DIAN(6 DOWNT0 0);                --蛇坐标

CONSTANT WX:DIAN(3 DOWNT0 0):=(5,5,5,5);      --墙的横坐标

CONSTANT WY:DIAN(3 DOWNT0 0):=(5,4,3,2);      --墙的纵坐标

SIGNAL TX,TY,TXT,TYT:INTEGER RANGE 0 TO 7;    --鼠坐标

BEGIN

p1:PROCESS(CLK,COUNT_1,COUNT_5K)              --分频部分
    BEGIN
        IF CLK'EVENT AND CLK='1' THEN
            COUNT_1<=COUNT_1+1;
            COUNT_5K<=COUNT_5K+1;

            IF COUNT_1<=24999999 THEN          --1HZ 分频
                CLK_1<='1';
            
```

```

        ELSIF COUNT_1>24999999 AND COUNT_1<=49999999 THEN
            CLK_1<='0';
        ELSIF COUNT_1=50000000 THEN
            COUNT_1<=0;
        END IF;

        IF COUNT_5K<=4999 THEN                                --5KHZ 分频
            CLK_5K<='1';
        ELSIF COUNT_5K>4999 AND COUNT_5K<=9999 THEN
            CLK_5K<='0';
        ELSIF COUNT_5K=10000 THEN
            COUNT_5K<=0;
        END IF;
    END IF;
END PROCESS;

p2:PROCESS(CLK,RESET,DEATH,TX,TY,SX,SY,T_MOUTH)              --控制部分
BEGIN
    IF RESET='1' THEN                                        --复位后，蛇转到死亡状态，初始化鼠坐标
        DEATH<='1';
        TX<=4;TY<=1;
    ELSIF CLK'EVENT AND CLK='1' THEN
        IF T_LOW=9 AND T_HIGH=5 THEN
            --当倒计时为 59 秒时，蛇转到运行状态
            DEATH<='0';
        ELSIF T_HIGH=0 AND T_LOW=0 THEN
            --倒计时时间到，蛇转到死亡状态
            DEATH<='1';
        ELSIF SX(0)=WX(0) AND (SY(0)=WY(0) OR SY(0)=WY(1) OR SY(0)=WY(2) OR
SY(0)=WY(3)) THEN
            DEATH<='1';                                    --蛇撞墙后，转到死亡状态
        END IF;
        IF DEATH='1' OR T_MOUTH=0 OR (SX(0)=TX AND SY(0)=TY) THEN
            --随机鼠的设置

```

```

        TXT<=TX;
        TYT<=TY;
        IF TY=5 OR TY=6 OR TY=7 THEN
            TY<=TY-2;
        ELSE TY<=TY+2;

        END IF;                                --鼠不与墙重叠

        IF (TX=4 OR TX=7) THEN
            TX<=0;
        ELSE
            TX<=TX+1;
        END IF;
    END IF;
END IF;
END PROCESS;

p3:PROCESS(CLK_1,DEATH,SCORE,TURN,DIRECTION)    --控制蛇运行及倒计时部
                                                分

BEGIN

    IF DEATH='1' THEN                            --当蛇转为死亡状态，游戏重新开始，鼠放置时间
                                                为 15S

        T_MOUTH<=15;    --比分为 0 分，倒计时为 59S，蛇位置初始化

        SCORE<=0;
        T_HIGH<=5;
        T_LOW<=9;
        SX<=(0,1,2,3,3,3,3);
        SY<=(0,0,0,0,0,0,0);
    ELSIF CLK_1'EVENT AND CLK_1='1' THEN

        IF T_MOUTH=0 THEN                        --鼠放置时间倒计时，15 秒后重置

            T_MOUTH<=15;
        ELSE
            T_MOUTH<=T_MOUTH-1;
        END IF;

        IF T_LOW=0 AND T_HIGH/=0 THEN          --总时间倒计时

```

```
T_LOW<=9;
T_HIGH<=T_HIGH-1;
ELSE
T_LOW<=T_LOW-1;
T_HIGH<=T_HIGH;
END IF;

IF (SX(0)=TXT AND SY(0)=TYT) THEN    --当蛇吃鼠后，比分加一，鼠重置
    SCORE<=SCORE+1;
    T_MOUTH<=15;
END IF;

--DIRECTION:"1000"--上;"0100"--右;"0010"--下;"0001"--左;

--TURN:0--上;1--右;2--下;3--左;

IF (DIRECTION="1000" AND TURN/=2)THEN
    TURN<=0;                --C 当输入方向上且上状态不为下时输入
                             向上信号生效
ELSIF (DIRECTION="0100" AND TURN/=3) THEN
    TURN<=1;                --C 当输入方向右且上状态不为左时输入
                             向右信号生效
ELSIF (DIRECTION="0010" AND TURN/=0) THEN
    TURN<=2;                --C 当输入方向下且上状态不为上时输入
                             向下信号生效
ELSIF (DIRECTION="0001" AND TURN/=1) THEN
    TURN<=3;                --C 当输入方向左且上状态不为右时输入
                             向左信号生效

ELSE
    TURN<=TURN;
END IF;
CASE TURN IS
    WHEN 0 =>
```

```
        IF(SY(0)=7) THEN
            T_HIGH<=0;

            T_LOW<=0;           --当蛇向下运动时向上不可用

        ELSE
            SY(0)<=SY(0)+1;
        END IF;

    WHEN 1 =>                   --当蛇向右运动时向左不可用

        IF SX(0)=7 THEN
            T_HIGH<=0;
            T_LOW<=0;
        ELSE
            SX(0)<=SX(0)+1;
        END IF;

    WHEN 2 =>                   --当蛇向下运动时向上不可用

        IF SY(0)=0 THEN
            T_HIGH<=0;
            T_LOW<=0;
        ELSE
            SY(0)<=SY(0)-1;
        END IF;

    WHEN 3 =>                   --当蛇向左运动时向右不可用

        IF SX(0)=0 THEN
            T_HIGH<=0;
            T_LOW<=0;
        ELSE
            SX(0)<=SX(0)-1;
        END IF;

    WHEN OTHERS => TURN <= TURN;
END CASE;
CASE SCORE IS

    WHEN 0 =>                   --当比分为 0 时，蛇长为 4 点

        SX(4)<=SX(2);
        SY(4)<=SY(2);
        SX(5)<=SX(2);
        SY(5)<=SY(2);
```

```

        SX(6)<=SX(2);
        SY(6)<=SY(2);
    WHEN 1 =>
        SX(4)<=SX(3);           --当比分为 1 时，蛇长为 5 点
        SY(4)<=SY(3);
        SX(5)<=SX(3);
        SY(5)<=SY(3);
        SX(6)<=SX(3);
        SY(6)<=SY(3);

    WHEN 2 =>                   --当比分为 2 时，蛇长为 6 点
        SX(4)<=SX(3);
        SY(4)<=SY(3);
        SX(5)<=SX(4);
        SY(5)<=SY(4);
        SX(6)<=SX(4);
        SY(6)<=SY(4);

    WHEN 3 =>                   --当比分为 3 时，蛇长为 7 点
        SX(4)<=SX(3);
        SY(4)<=SY(3);
        SX(5)<=SX(4);
        SY(5)<=SY(4);
        SX(6)<=SX(5);
        SY(6)<=SY(5);
    WHEN OTHERS => SCORE <=SCORE;
END CASE;
SX(3)<=SX(2);
SY(3)<=SY(2);
SX(2)<=SX(1);
SY(2)<=SY(1);
SX(1)<=SX(0);
SY(1)<=SY(0);
END IF;
END PROCESS;

p4:PROCESS(CLK_5K)
    --显示部分

    BEGIN

```

```
IF CLK_5K'EVENT AND CLK_5K='1' THEN
```

```
--扫描显示时每来一个时钟就对整个 8*8 点阵全部扫描一次
```

```
--由于点阵上的点亮的条件是行为高电平'1',列为低电平'0'时,则对应的点就被点亮
```

```
COL<="11111111";
```

```
ROW<="00000000";
```

```
IF NUM=11 THEN
```

```
--分频显示参数
```

```
NUM<=0;
```

```
ELSE NUM<=NUM+1;
```

```
END IF;
```

```
CASE NUM IS
```

```
WHEN 0 =>
```

```
--扫描蛇坐标
```

```
COL(SX(0))<='0';
```

```
ROW(SY(0))<='1';
```

```
WHEN 1 =>
```

```
COL(SX(1))<='0';
```

```
ROW(SY(1))<='1';
```

```
WHEN 2=>
```

```
COL(SX(2))<='0';
```

```
ROW(SY(2))<='1';
```

```
WHEN 3=>
```

```
COL(SX(3))<='0';
```

```
ROW(SY(3))<='1';
```

```
WHEN 4=>
```

```
COL(SX(4))<='0';
```

```
ROW(SY(4))<='1';
```

```
WHEN 5=>
```

```
COL(SX(5))<='0';
```

```
ROW(SY(5))<='1';
```

```
WHEN 6=>
```

```
COL(SX(6))<='0';
```

```
ROW(SY(6))<='1';
```

```
WHEN 7=>
```

```
--扫描墙坐标
```

```
COL(WX(0))<='0';
```

```
ROW(WY(0))<='1';
```

```
WHEN 8=>
```

```

        COL(WX(1))<='0';
        ROW(WY(1))<='1';
    WHEN 9=>
        COL(WX(2))<='0';
        ROW(WY(2))<='1';
    WHEN 10=>
        COL(WX(3))<='0';
        ROW(WY(3))<='1';

    WHEN 11=>                                     --扫描鼠坐标

        COL(TX)<='0';
        ROW(TY)<='1';
    WHEN OTHERS=>
        COL <="11111111";
        ROW <="00000000";
    END CASE;
END IF;
END PROCESS;

```

P5:PROCESS(CLK_1,T_HIGH,T_LOW) --时间显示模块

```

BEGIN
    IF CLK_1'EVENT AND CLK_1='1' THEN
        CASE T_HIGH is
            WHEN 5 => LIGHT_H <= "101";
            WHEN 4 => LIGHT_H <= "100";
            WHEN 3 => LIGHT_H <= "011";
            WHEN 2 => LIGHT_H <= "010";
            WHEN 1 => LIGHT_H <= "001";
            WHEN 0 => LIGHT_H <= "000";
            WHEN others=>LIGHT_H<="000";
        END CASE;
        CASE T_LOW is
            WHEN 9 => LIGHT_L <= "1001";
            WHEN 8 => LIGHT_L <= "1000";
            WHEN 7 => LIGHT_L <= "0111";
            WHEN 6 => LIGHT_L <= "0110";
            WHEN 5 => LIGHT_L <= "0101";
            WHEN 4 => LIGHT_L <= "0100";
            WHEN 3 => LIGHT_L <= "0011";

```

```
        WHEN 2 => LIGHT_L <= "0010";  
        WHEN 1 => LIGHT_L <= "0001";  
        WHEN 0 => LIGHT_L <= "0000";  
        WHEN others=>LIGHT_L<="0000";  
    END CASE;  
END IF;  
END PROCESS;  
END Behavioral;
```

顶层模块图

