一、任务要求

设计并实现一个简易迷宫游戏机。

（一）、基本要求：

（1）、用 8×8 点阵进行游戏显示。

（2）、迷宫游戏如图 1 所示，采用双色点阵显示，其中红色 LED 为迷宫墙壁，绿色 LED 表示人物。通过 BTN3～BTN0 四个按键控制迷宫中的人物进行上下左右移动，使人物从起始点出发，走到迷宫的出口，游戏结束。

（3）、普通计时模式：通过按键 BTN7 启动游戏，必须在 30 秒内找到出口，否则游戏失败，用两个数码管进行倒计时显示。游戏胜利或者失败均要在 8×8 点阵上有相应的画面出现。

（4）、迷宫中的人物在行走过程中，如果碰到墙壁，保持原地不动。

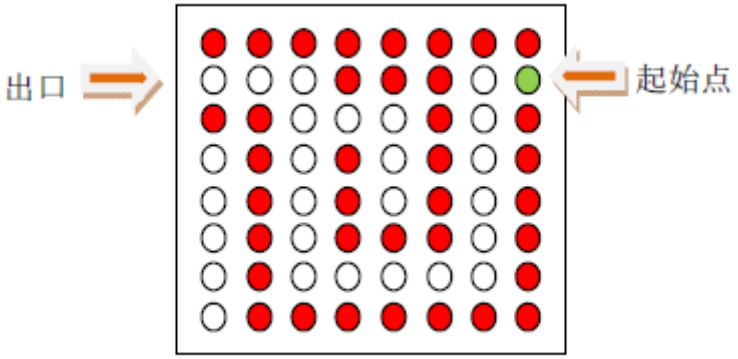


图 1 迷宫游戏示意图

（二）、提高要求

（1）、多种迷宫地图可以选择。

（2）、在计时的基础上增加计步的功能，每按一次控制按键步数加1，碰壁不计算步数，计步结果用数码管显示。

（3）、为游戏增加提示音乐，在不同时间段采用不同频率的信号控制蜂鸣器发声报警。

（4）、增加其他游戏模式。

（5）、自拟其它功能。

二、系统设计

（一）、设计思路

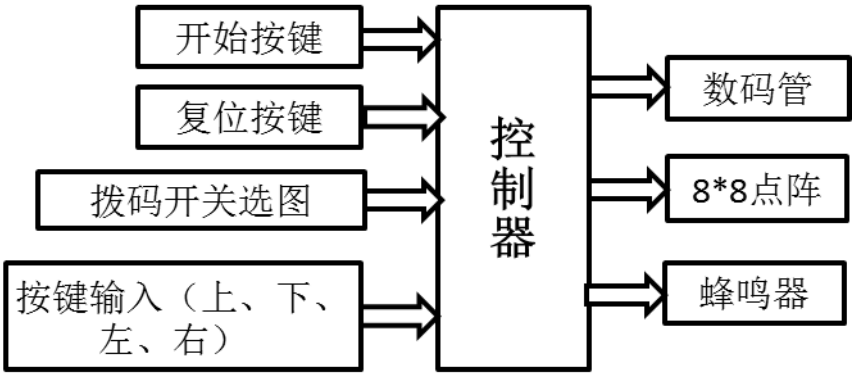
程序采用自顶向下和模块化设计的思路，系统由一个个模块组成，一个模块对应一个或多个进程，各模块相互协调工作。

首先系统会并发执行多个模块，包括各分频器模块，用以获得其他模块所需的工作时钟；状态控制器模块，用以控制系统的当前状态。然后执行个子模块，

包括拨码开关判断模块，点阵工作模块，计时器模块，绿点显示模块，数码管显示模块，蜂鸣器模块。

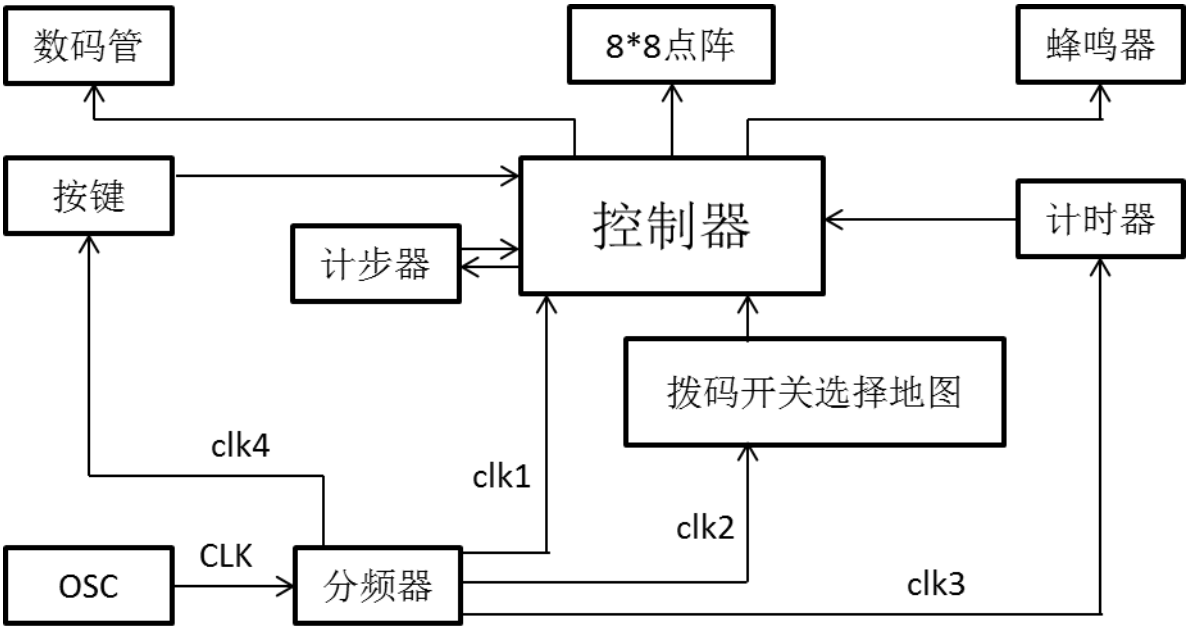
（二）、总体框图

1、总体结构框图



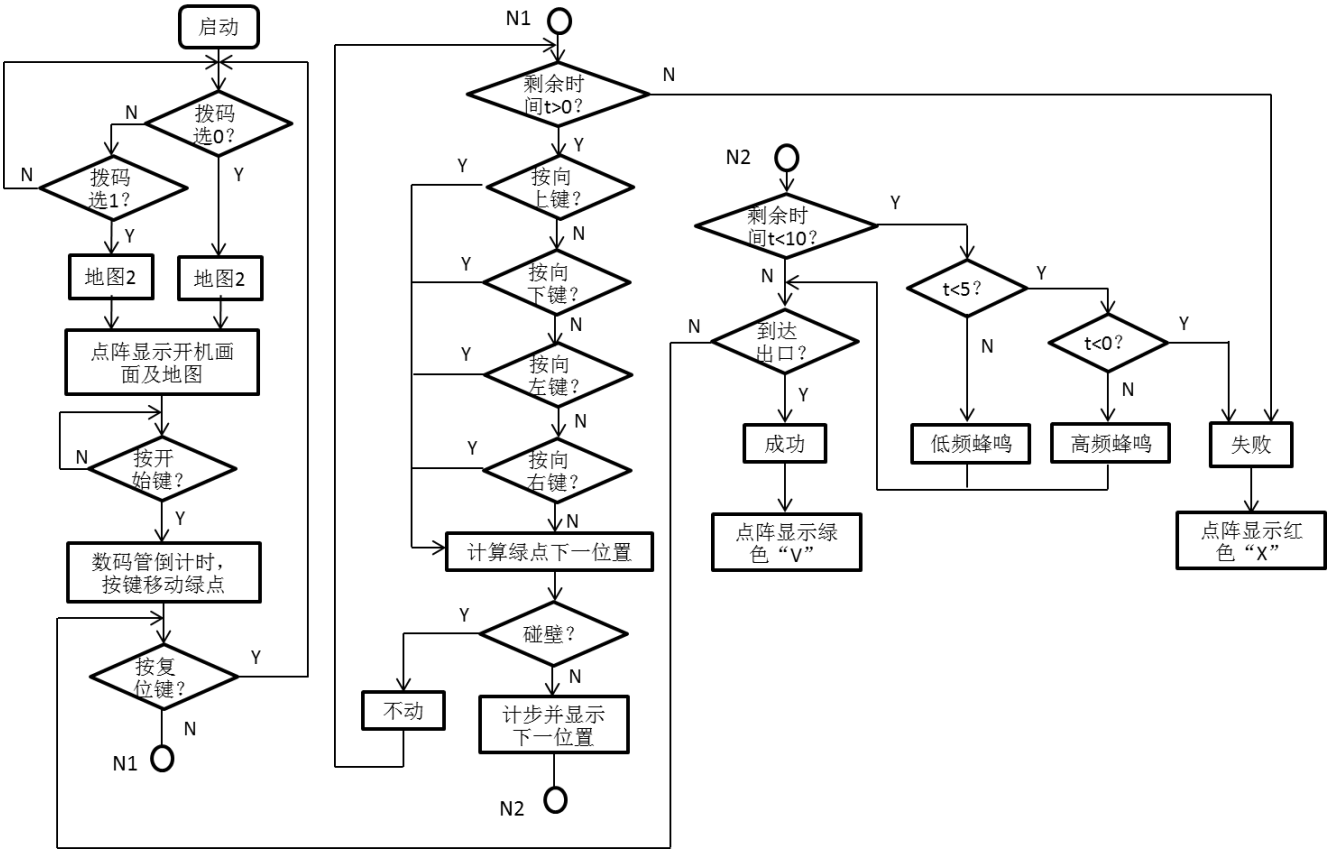
图一 总体结构框图

2、逻辑划分方框图



图二 逻辑划分方框图

3、ASM 图



图三 ASM 图

（三）、分块设计

1、分频器模块

共有四个分频器，分别为计时器分频（1000Hz），点阵分频（1Hz），按键控制判断分频（5Hz），蜂鸣器分频（10Hz）。分频器用于作产生各相关模块的工作时钟。

2、状态控制器模块

本模块用于控制开始键及复位键按下后、以及没有按开始键和复位键时，各相关模块状态值的变化。这样相关模块就可以通过判断自己的状态值来确定现在本模块的工作状态。

3、拨码开关判断模块

判断拨码开关 SW0 是选 0 还是选 1，从而确定游戏是使用地图 1 还是地图 2。

4、点阵工作模块

主工作模块，显示开机画面、地图以及游戏最终结果，并负责启动与关闭部分子模块。

5、计时器模块

游戏开始后负责计时工作，从 30s 开始以 1s 的步长进行倒计时，知道剩余总时间 tz 变为 0。

6、绿点显示模块

判断上、下、左、右按键是否按下，以及按下后绿点下一位置是否碰壁。如果碰壁，则绿点保持原地不动，计步器不计步；如果不碰壁，则绿点移动到下一位置，计步器加 1。

7、数码管显示模块

数码管高两位显示计步器的步数，低两位显示计数器的倒计时（以 1s 为步长进行显示），中间两位不显示。

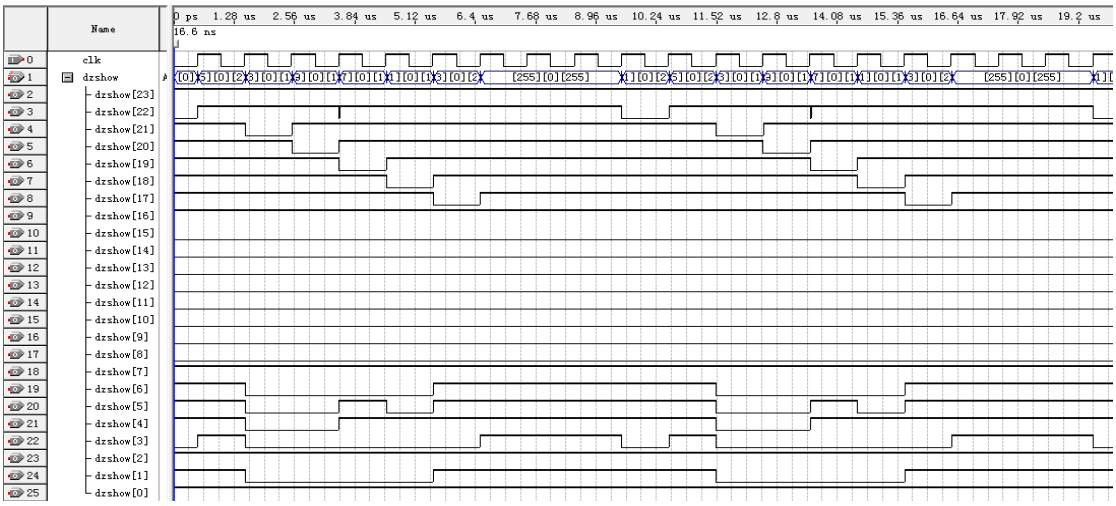
8、蜂鸣器模块

此模块为附加功能，当游戏剩余时间在 5~10s 内时，蜂鸣器以 1Hz 的频率进行鸣叫提示；当游戏剩余时间在 0~5s 内时，蜂鸣器以 5Hz 的频率进行鸣叫提示。

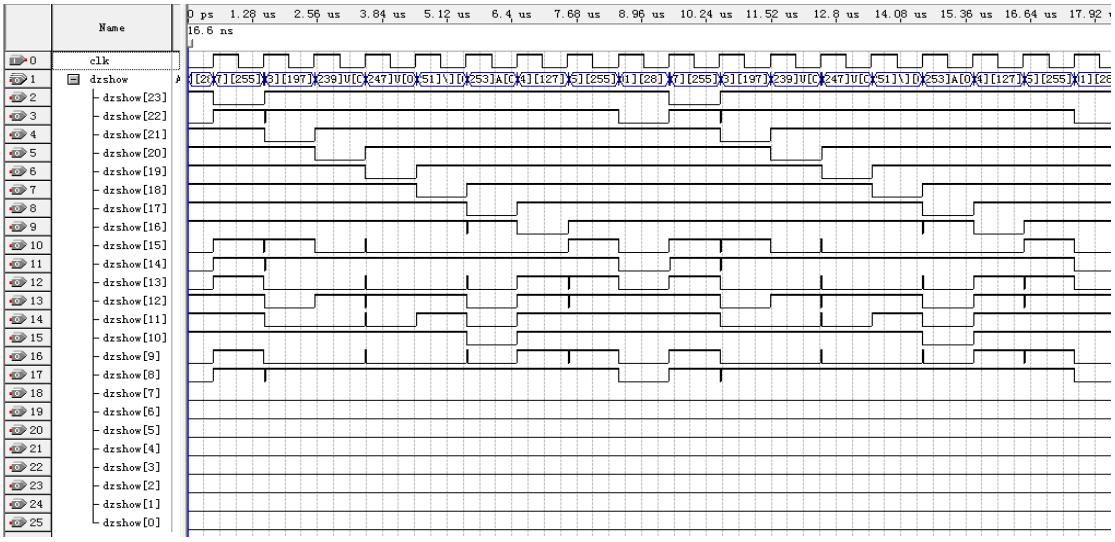
三、仿真波形及波形分析

1、点阵显示

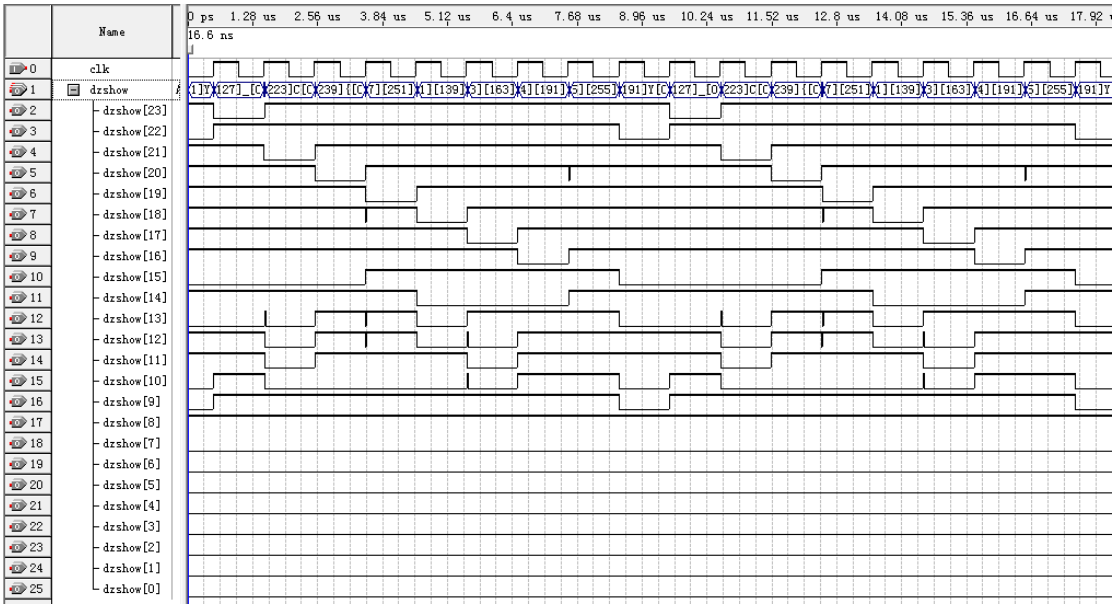
（1）点阵显示开机画面绿色“GO”



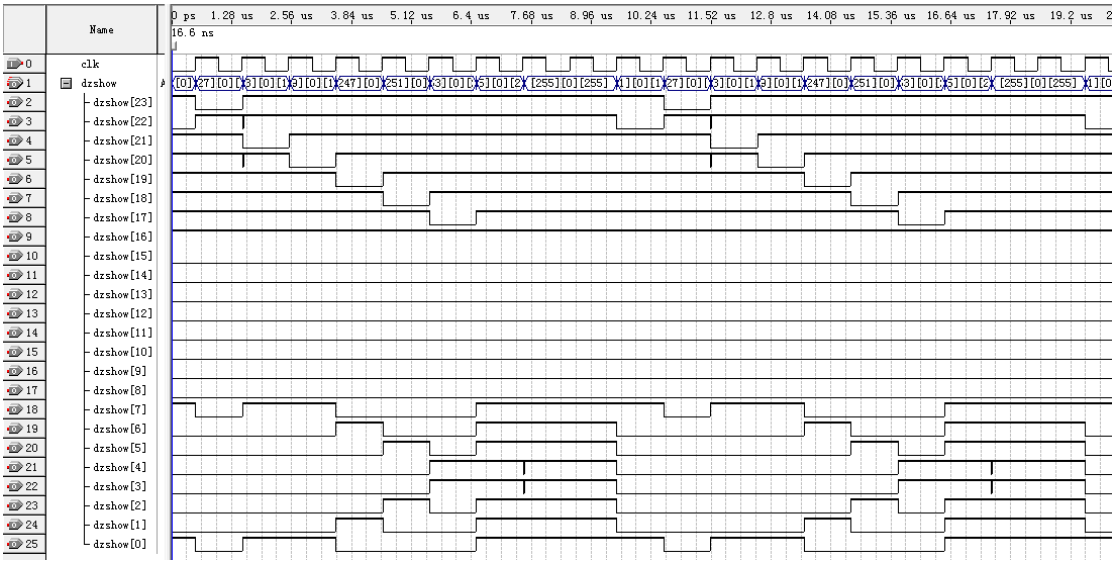
（2）点阵显示红色地图 1



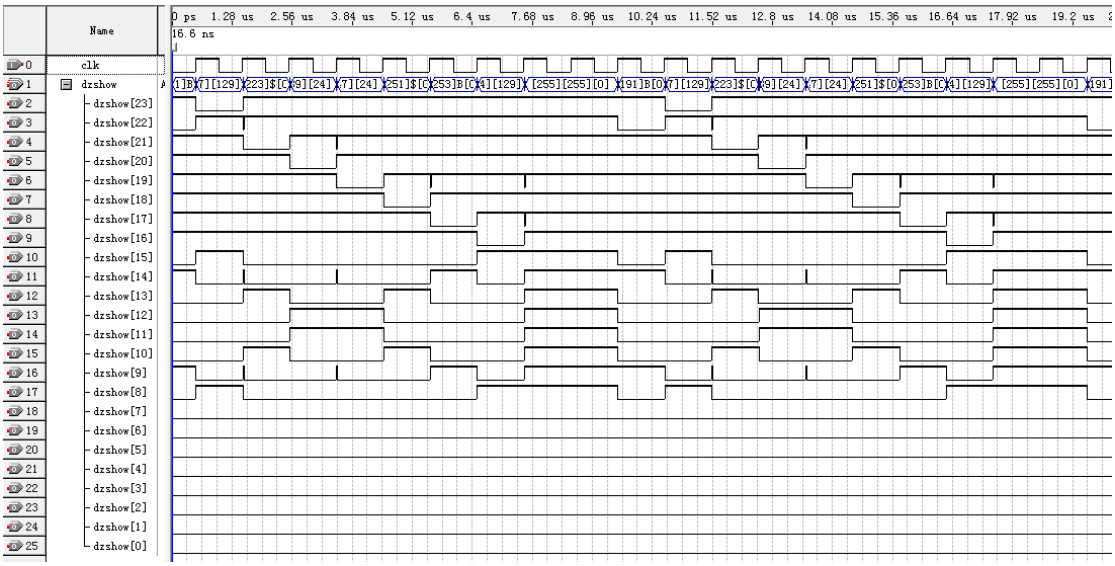
（3）点阵显示红色地图 2



（4）点阵显示绿色“V”

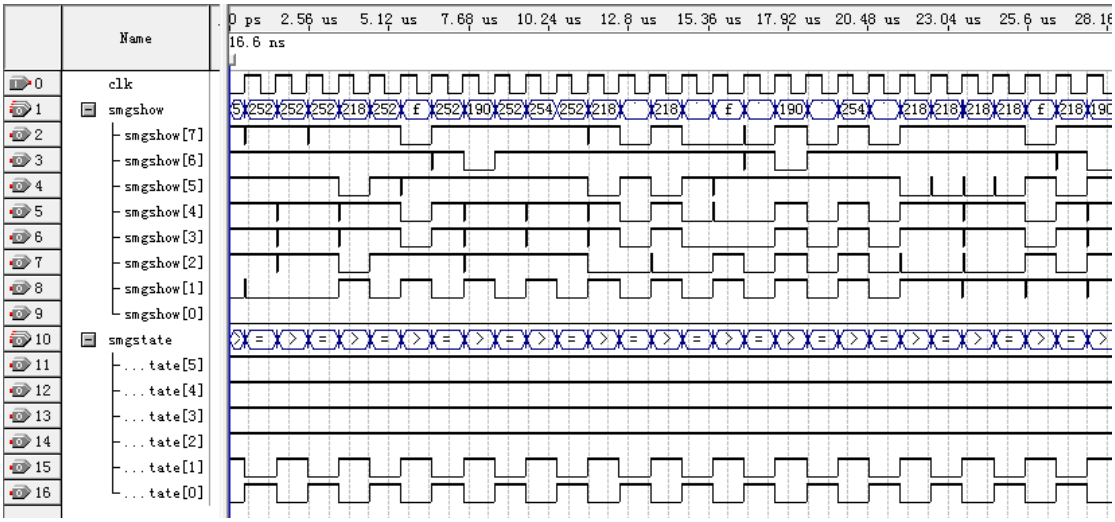


（5）点阵显示红色“X”



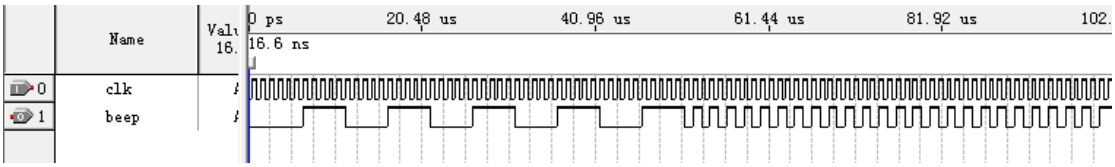
2、数码管显示

数码管显示倒计时。



3、蜂鸣器发出声音

先是以 1Hz 的频率响 5s，再以 5Hz 的频率响 5s。



四、源程序

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity maze is
    port(
        clk: in std_logic;              --PIN18
        start,reset: in std_logic;      --BTN7,BTN6
        u,d,l,r: in std_logic;          --上下左右,BTN3~BTN0
        switch: in std_logic;           --拨码选择地图,SW0
        beep: out std_logic;            --蜂鸣器
       dzshow: out std_logic_vector(23 downto 0);     --row7~row0,colr7~colr0,colg7~colg0
                                                      --点阵显示
        smgstate: out std_logic_vector(5 downto 0);   --cat5~cat0,数码管状态
        smgshow: out std_logic_vector(7 downto 0)     --abcdefgp,数码管显示
        );
end maze;

architecture main of maze is
    signal cnt1: integer range 0 to 24999;          --用于计时器分频
    signal cnt2: integer range 0 to 24999999;       --用于点阵分频
    signal cnt3: integer range 0 to 4999999;        --用于按键控制判断分频
    signal cnt4: integer range 0 to 4999999;        --用于蜂鸣器分频

    signal clk_time: std_logic;     --计时器时钟,T1=1ms
    signal clk_dz: std_logic;       --点阵时钟,T2=1s
    signal clk_key: std_logic;      --按键控制判断时钟,T3=0.2s
    signal clk_beep: std_logic;     --蜂鸣器时钟,T4=0.1s

    signal row: std_logic_vector(7 downto 0);    --用于显示绿点
    signal col: std_logic_vector(15 downto 0);   --用于显示绿点

    signal tostart: std_logic;      --系统是否开始工作的信号
    signal greenreset: std_logic; --绿点显示判断模块重置
    signal treset: std_logic;       --计时器模块重置
    signal dzreset: std_logic;      --点阵模块重置
    signal smgreset: std_logic;     --数码管模块重置
    signal beepreset: std_logic;    --蜂鸣器时钟模块重置

    signal dzwork: std_logic;       --点阵工作
```

```
signal twork: std_logic;        --计时器工作
signal smgwork: std_logic;      --数码管工作
signal greenwork: std_logic; --绿点显示模块工作
signal beepwork: std_logic;   --蜂鸣器工作
signal worked: std_logic;       --开机画面是否已经显示的信号

signal vic: std_logic;          --成功
signal fal: std_logic;          --失败

signal beeps:std_logic;         --点阵地图扫描
signal sw:std_logic;            --开关选图信号

type smgst is array(0 to 9)of std_logic_vector(7 downto 0);

signal smgshowed: smgst;     --存放数码管数字显示编码

shared variable tend: integer range 0 to 1;      --开机初始状态
                                                 --默认等同于按下 reset 键后的状态
shared variable smgscan:integer range 0 to 3; --数码管扫描
shared variable dzkjshow:integer range 0 to 2; --控制点阵开机画面显示时间
shared variable dzresult:integer range 0 to 3;  --控制点阵显示成功和失败画面的时间
shared variable dzscan:integer range 0 to 9;    --点阵扫描

shared variable x:integer range -1 to 8;         --绿点横坐标
shared variable y:integer range -1 to 8;         --绿点纵坐标

shared variable tz:integer range -1 to 30;       --总倒计时
shared variable ts:integer range 0 to 3;         --倒计时秒数十位
shared variable tg:integer range 0 to 9;         --倒计时秒数个位
shared variable tm:integer range 0 to 1000;    --倒计时内部精度,1ms

shared variable stepz: integer range 0 to 99;   --总移动步数
shared variable steps: integer range 0 to 9;     --移动步数的十位
shared variable stepg: integer range 0 to 9;     --移动步数的个位

shared variable cntbeepz:integer range 0 to 99;      --总蜂鸣器计数
shared variable cntbeeps:integer range 0 to 9;       --蜂鸣器计数十位
shared variable cntbeepg:integer range 0 to 9;       --蜂鸣器计数个位

--shared variable sw:integer range 0 to 1;

begin
```

```
smgshowed(0)<="11111100"; --0
smgshowed(1)<="01100000"; --1
smgshowed(2)<="11011010"; --2
smgshowed(3)<="11110010"; --3
smgshowed(4)<="01100110"; --4
smgshowed(5)<="10110110"; --5
smgshowed(6)<="10111110"; --6
smgshowed(7)<="11100000"; --7
smgshowed(8)<="11111110"; --8
smgshowed(9)<="11110110"; --9

timediv:process(clk)                    --计时器分频
begin
    if(clk'event and clk='1') then
        if(cnt1=24999) then
            cnt1<=0;
            clk_time<=not clk_time;     --5e4 分频,1000Hz,T1=1ms
        else
            cnt1<=cnt1+1;
        end if;
    end if;
end process;

dzdiv:process(clk)                      --点阵分频
begin
    if(clk'event and clk='1') then
        if(cnt2=24999999) then
            cnt2<=0;
            clk_dz<=not clk_dz;         --5e7 分频,1Hz,T2=1s
        else
            cnt2<=cnt2+1;
        end if;
    end if;
end process;

keydiv:process(clk)                     --按键控制判断分频
begin
    if(clk'event and clk='1') then
        if(cnt3=4999999) then
            cnt3<=0;
            clk_key<=not clk_key;  --1e7 分频,5Hz,T3=0.2s
        else
```

```
                cnt3<=cnt3+1;
            end if;
        end if;
end process;

beepdiv:process(clk)                        --蜂鸣器分频
begin
        if(clk'event and clk='1') then
            if(cnt4=2499999) then
                cnt4<=0;
                clk_beep<=not clk_beep;     --5e6 分频,10Hz,T4=0.1s
            else
                cnt4<=cnt4+1;
            end if;
        end if;
end process;

state:process(clk,start,reset)              --状态控制器
begin
        if(clk'event and clk='1') then
            if(start='1') then
                tostart<='1';
            end if;
            if(reset='1' or tend=1) then
                dzreset<='1';
                treset<='1';
                smgreset<='1';
                greenreset<='1';
                beepreset<='1';
                tostart<='0';
                dzwork<='0';
            else
                dzreset<='0';
                treset<='0';
                smgreset<='0';
                greenreset<='0';
                beepreset<='0';
                if(tostart='1') then
                    dzwork<='1';            --启动点阵模块
                end if;
            end if;
        end if;
```

```
end process;

dzkj:process(clk_dz,dzreset,dzwork,worked)   --控制点阵开机画面显示时间
begin
    if(dzreset='1') then
        dzkjshow:=0;                         --复位后从 0 开始
    else
        if(clk_dz'event and clk_dz='1' and dzwork='1' and worked='0') then
            if(dzkjshow=2) then              --8 位循环计数,1s 加 1
                dzkjshow:=0;
            else
                dzkjshow:=dzkjshow+1;
            end if;
        end if;
    end if;
end process;

dzjieguo:process(clk_dz,dzreset)             --控制点阵显示成功和失败画面的时间
begin
    if(dzreset='1') then
        dzresult:=0;                             --复位后从 0 开始
    else
        if(clk_dz'event and clk_dz='1') then
            if(vic='1' or fal='1') then          --控制绿 V 或红叉显示 2s
                if(dzresult<=2) then             --dzresult=3 后保持不变
                    dzresult:=dzresult+1;
                end if;
            end if;
        end if;
    end if;
end process;

dzsaomiao:process(clk_time,dzreset)              --点阵地图及结果扫描计数
begin
    if(dzreset='1') then
        dzscan:=0;                               --复位后从 0 开始
    else
        if(clk_time'event and clk_time='1') then
            if(dzscan=9) then                    --10 位循环计数,1ms 加 1
                dzscan:=0;
            else
                dzscan:=dzscan+1;
```

```
                end if;
            end if;
        end if;
end process;

swi:process(clk,switch)        --开关状态判断
begin
        if(clk'event and clk='1') then
            if(switch='0') then
                sw<='0';
            else
                sw<='1';
            end if;
        end if;
end process;

control:process(clk,dzreset,dzwork,sw)
begin
        if(dzreset='1') then
            dzshow<="000000000000000000000000";
            tend:=0;
            worked<='0';
            twork<='0';
            smgwork<='0';
            greenwork<='0';
            beepwork<='0';
        else
            if(clk'event and clk='1') then

                if(sw='0') then                    --拨码开关选 0
                    if(dzwork='1') then            --点阵开始工作
                        if(worked='0') then        --开机画面还未显示
                            case dzkjshow is
                                when 0=>  --显示单词 GO
                                    case dzscan is
                                        when 1=>dzshow<="111111110000000011111111";
                                        when 0=>dzshow<="101111110000000011110111";
                                        when 2=>dzshow<="110111110000000010000101";
                                        when 3=>dzshow<="111011110000000010000101";
                                        when 4=>dzshow<="111101110000000010110101";
                                        when 5=>dzshow<="111110110000000010010101";
                                        when 6=>dzshow<="111111010000000011110111";
```

```
                when 7=>dzshow<="11111111000000011111111";
                when 8=>dzshow<="11111111000000011111111";
                when 9=>dzshow<="11111111000000011111111";
            end case;
        when 1=>    --显示单词 GO
            case dzscan is
                when 1=>dzshow<="11111111000000011111111";
                when 0=>dzshow<="10111111000000011110111";
                when 2=>dzshow<="11011111000000010000101";
                when 3=>dzshow<="11101111000000010000101";
                when 4=>dzshow<="11110111000000010110101";
                when 5=>dzshow<="11111011000000010010101";
                when 6=>dzshow<="11111101000000011110111";
                when 7=>dzshow<="11111111000000011111111";
                when 8=>dzshow<="11111111000000011111111";
                when 9=>dzshow<="11111111000000011111111";
            end case;
        when 2=>
            worked<='1';     --开机画面已经显示
            twork<='1';      --计时器开始工作
            smgwork<='1';    --数码管开始工作
            greenwork<='1'; --绿点显示模块开始工作
            beepwork<='1';   --蜂鸣器模块开始工作
            if (reset='1') then
                twork<='0';
                smgwork<='0';
                greenwork<='0';
                beepwork<='0';
            end if;
    end case;
else                                --开机画面已经显示
    if(fal='1') then
        twork<='0';
        smgwork<='0';
        greenwork<='0';
        beepwork<='0';
        if(dzresult<=2) then
            case dzscan is    --显示红色叉
                when 1=>dzshow<="01111111100000100000000";
                when 0=>dzshow<="10111111010000100000000";
                when 2=>dzshow<="11011111001001000000000";
                when 3=>dzshow<="11101111000110000000000";
```

```
                when 4=>dzshow<="11110111000110000000000";
                when 5=>dzshow<="11111011001001000000000";
                when 6=>dzshow<="11111101010000010000000";
                when 7=>dzshow<="11111110100000010000000";
                when 8=>dzshow<="11111111111111110000000";
                when 9=>dzshow<="11111111111111110000000";
            end case;
        else                    --dzresult=3
            tend:=1;
        end if;
    elsif(vic='1') then
        twork<='0';
        smgwork<='0';
        greenwork<='0';
        beepwork<='0';
        if(dzresult<=2) then
            case dzscan is      --显示绿色 V
                when 1=>dzshow<="01111111000000000000000";
                when 0=>dzshow<="10111111000000010000001";
                when 2=>dzshow<="11011111000000010000001";
                when 3=>dzshow<="11101111000000010000001";
                when 4=>dzshow<="11110111000000001000010";
                when 5=>dzshow<="11111011000000000100100";
                when 6=>dzshow<="11111101000000000011000";
                when 7=>dzshow<="11111111000000011111111";
                when 8=>dzshow<="11111111000000011111111";
                when 9=>dzshow<="11111111000000011111111";
            end case;
        else                    --dzresult=3
            tend:=1;
        end if;
    else
        case dzscan is          --显示红色迷宫 1
            when 1=>dzshow<="01111111111111110000000";
            when 0=>dzshow<="10111111000111000000000";
            when 2=>dzshow<="11011111110001010000000";
            when 3=>dzshow<="11101111010101010000000";
            when 4=>dzshow<="11110111010101010000000";
            when 5=>dzshow<="11111011010111010000000";
            when 6=>dzshow<="11111101010000010000000";
            when 7=>dzshow<="11111110011111110000000";
            when 8=>dzshow<="11111111111111110000000";
```

```
                    when 9=>          --绿点显示(x,y)
                        case x is
                            when 0=>col<="0000000010000000";
                            when 1=>col<="0000000001000000";
                            when 2=>col<="0000000000100000";
                            when 3=>col<="0000000000010000";
                            when 4=>col<="0000000000001000";
                            when 5=>col<="0000000000000100";
                            when 6=>col<="0000000000000010";
                            when 7=>col<="0000000000000001";
                            when others=>x:=x;
                        end case;
                        case y is
                            when 0=>row<="11111110";
                            when 1=>row<="11111101";
                            when 2=>row<="11111011";
                            when 3=>row<="11110111";
                            when 4=>row<="11101111";
                            when 5=>row<="11011111";
                            when 6=>row<="10111111";
                            when 7=>row<="01111111";
                            when others=>y:=y;
                        end case;
                    dzshow<=row&col;
                end case;
            end if;
        end if;
    end if;

else                            --拨码开关选 1
    if(dzwork='1') then         --点阵开始工作
        if(worked='0') then     --开机画面还未显示
            case dzkjshow is
                when 0=>  --显示单词 GO
                    case dzscan is
                        when 1=>dzshow<="111111110000000011111111";
                        when 0=>dzshow<="101111110000000011110111";
                        when 2=>dzshow<="110111110000000010000101";
                        when 3=>dzshow<="111011110000000010000101";
                        when 4=>dzshow<="111101110000000010110101";
                        when 5=>dzshow<="111110110000000010010101";
                        when 6=>dzshow<="111111010000000011110111";
```

```
                when 7=>dzshow<="11111111000000011111111";
                when 8=>dzshow<="11111111000000011111111";
                when 9=>dzshow<="11111111000000011111111";
            end case;
        when 1=>    --显示单词 GO
            case dzscan is
                when 1=>dzshow<="11111111000000011111111";
                when 0=>dzshow<="10111111000000011110111";
                when 2=>dzshow<="11011111000000010000101";
                when 3=>dzshow<="11101111000000010000101";
                when 4=>dzshow<="11110111000000010110101";
                when 5=>dzshow<="11111011000000010010101";
                when 6=>dzshow<="11111101000000011110111";
                when 7=>dzshow<="11111111000000011111111";
                when 8=>dzshow<="11111111000000011111111";
                when 9=>dzshow<="11111111000000011111111";
            end case;
        when 2=>
            worked<='1';     --开机画面已经显示
            twork<='1';      --计时器开始工作
            smgwork<='1';    --数码管开始工作
            greenwork<='1'; --绿点显示模块开始工作
            beepwork<='1';   --蜂鸣器模块开始工作
            if (reset='1') then
                twork<='0';
                smgwork<='0';
                greenwork<='0';

                beepwork<='0';
            end if;
    end case;
else                            --开机画面已经显示
    if(fal='1') then
        twork<='0';
        smgwork<='0';
        greenwork<='0';
        beepwork<='0';
        if(dzresult<=2) then
            case dzscan is    --显示红色叉
                when 1=>dzshow<="01111111100000100000000";
                when 0=>dzshow<="10111111010000100000000";
                when 2=>dzshow<="11011111001001000000000";
```

```
                when 3=>dzshow<="11101111000110000000000";
                when 4=>dzshow<="11110111000110000000000";
                when 5=>dzshow<="11111011001001000000000";
                when 6=>dzshow<="11111101010000010000000";
                when 7=>dzshow<="11111110100000010000000";
                when 8=>dzshow<="11111111111111110000000";
                when 9=>dzshow<="11111111111111110000000";
            end case;
        else                    --dzresult=3
            tend:=1;
        end if;
    elsif(vic='1') then
        twork<='0';
        smgwork<='0';
        greenwork<='0';
        beepwork<='0';
        if(dzresult<=2) then
            case dzscan is      --显示绿色 V
                when 1=>dzshow<="01111111000000000000000";
                when 0=>dzshow<="10111111000000000100000001";
                when 2=>dzshow<="11011111000000001000000001";
                when 3=>dzshow<="11101111000000001000000001";
                when 4=>dzshow<="11110111000000000010000010";
                when 5=>dzshow<="11111011000000000000100100";
                when 6=>dzshow<="11111101000000000000011000";
                when 7=>dzshow<="11111111000000000111111111";
                when 8=>dzshow<="11111111000000000111111111";
                when 9=>dzshow<="11111111000000000111111111";
            end case;
        else                    --dzresult=3
            tend:=1;
        end if;
    else
        case dzscan is          --显示红色迷宫 2
            when 1=>dzshow<="01111111010111111000000000";
            when 0=>dzshow<="10111111010110010000000000";
            when 2=>dzshow<="11011111010000011000000000";
            when 3=>dzshow<="11101111011110110000000000";
            when 4=>dzshow<="11110111111110110000000000";
            when 5=>dzshow<="11111011100010110000000000";
            when 6=>dzshow<="11111101101000110000000000";
            when 7=>dzshow<="11111110101111111000000000";
```

```
                                        when 8=>dzshow<="111111111111111100000000";
                                        when 9=>        --绿点显示(x,y)
                                            case x is
                                                when 0=>col<="0000000010000000";
                                                when 1=>col<="0000000001000000";
                                                when 2=>col<="0000000000100000";
                                                when 3=>col<="0000000000010000";
                                                when 4=>col<="0000000000001000";
                                                when 5=>col<="0000000000000100";
                                                when 6=>col<="0000000000000010";
                                                when 7=>col<="0000000000000001";
                                                when others=>x:=x;
                                            end case;
                                            case y is
                                                when 0=>row<="11111110";
                                                when 1=>row<="11111101";
                                                when 2=>row<="11111011";
                                                when 3=>row<="11110111";
                                                when 4=>row<="11101111";
                                                when 5=>row<="11011111";
                                                when 6=>row<="10111111";
                                                when 7=>row<="01111111";
                                                when others=>y:=y;
                                            end case;
                                        dzshow<=row&col;
                                    end case;
                                end if;
                            end if;
                        end if;
                    end if;

            end if;
        end if;
end process;

t:process(clk_time,treset,twork)    --计时器模块
begin
        if(treset='1') then
            tz:=30;                 --倒计时初始值为 30s
            ts:=0;
            tg:=0;
            tm:=0;
```

```
        else
            if(twork='1') then
                if(clk_time'event and clk_time='1') then
                    tm:=tm+1;
                    if(tm>=1000) then       --满 1s
                        tm:=0;
                        tz:=tz-1;
                    end if;
                    if(tz<=0) then          --倒计时结束
                        tz:=0;
                    end if;
                    ts:=tz/10;
                    tg:=tz-10*ts;
                end if;
            end if;
        end if;
end process;

green:process(clk_key,greenreset,greenwork,u,d,l,r,sw)   --绿点显示模块
        variable derx :integer range -1 to 1;                    --x 的移动值
        variable dery :integer range -1 to 1;                    --y 的移动值
begin
        if(greenreset='1') then
            if(sw='0') then
                x:=7;
                y:=6;       --绿点初始位置
            else
                x:=1;
                y:=0;       --绿点初始位置
            end if;
                stepz:=0;   --移动步数为 0
                steps:=0;
                stepg:=0;
                vic<='0';
                fal<='0';
        else
            if(greenwork='1') then
                if(clk_key'event and clk_key='1') then
                    derx:=0;    --移动值初值为 0
                    dery:=0;    --移动值初值为 0
                    if(tz>0) then
                        if(l='1')then
```

```
        derx:=-1;
elsif(r='1') then
        derx:=1;
elsif(u='1')then
        dery:=1;
elsif(d='1')then
        dery:=-1;
else
end if;
x:=x+derx;
y:=y+dery; --按完按键后的下个位置
if(sw='0') then
        if( (x=7 and y=6)or
                (x=6 and y>=1 and y<=6)or
                (x>=2 and x<=5 and y=1)or
                (x=2 and y>=2 and y<=6)or
                (x>=0 and x<=1 and y=6)or
                (x=3 and y=5)or
                (x=4 and y>=3 and y<=5)) then      --如果下个位置在赛道上
                if(x=0 and y=6) then                --到达终点
                        vic<='1';
                elsif( not(derx=0 and dery=0)) then
                        stepz:=stepz+1;          --步数加 1，绿点移动到下一个位置
                        steps:=stepz/10;
                        stepg:=stepz-10*steps;
                end if;
        else                --下个位置不在赛道上
                x:=x-derx;
                y:=y-dery;  --绿点保持原来的位置
        end if;
else
        if( (x=1 and y>=0 and y<=2)or
                (x>=2 and x<=3 and y=2)or
                (x>=3 and x<=5 and y=1)or
                (x=5 and y>=2 and y<=5)or
                (x>=5 and x<=6 and y=6)or
                (x>=2 and x<=4 and y=5)or
                (x=2 and y>=6 and y<=7)) then      --如果下个位置在赛道上
                if(x=2 and y=7) then                --到达终点
                        vic<='1';
                elsif( not(derx=0 and dery=0)) then
                        stepz:=stepz+1;          --步数加 1，绿点移动到下一个位置
```

```
                                        steps:=stepz/10;
                                        stepg:=stepz-10*steps;
                                    end if;
                                else                --下个位置不在赛道上
                                    x:=x-derx;
                                    y:=y-dery;  --绿点保持原来的位置
                                end if;
                            end if;
                        else
                            fal<='1';
                        end if;
                    end if;
                end if;
            end if;
        end process;

        smg:process(clk_time)               --数码管显示模块
        begin
            if(clk_time'event and clk_time='1') then
                if(smgscan=3) then
                    smgscan:=0;
                else
                    smgscan:=smgscan+1;
                end if;
            end if;
        end process;

        process(clk_time,smgreset,smgwork)
        begin
            if(smgreset='1') then                   --数码管灭
                smgstate<="111111";
                smgshow<="00000000";
            else
                if(clk_time'event and clk_time='1') then
                    if(smgwork='1') then
                        case smgscan is
                            when 0=>  --数码管 cat0 显示倒计时秒数个位
                                smgstate<="111110";
                                smgshow<=smgshowed(tg);
                            when 1=>   --数码管 cat1 显示倒计时秒数十位
                                smgstate<="111101";
                                smgshow<=smgshowed(ts);
```

```
                    when 2=>  --数码管 cat4 显示移动步数个位
                        smgstate<="101111";
                        smgshow<=smgshowed(stepg);
                    when 3=>  --数码管 cat5 显示移动步数十位
                        smgstate<="011111";
                        smgshow<=smgshowed(steps);
                end case;
            else
                case smgscan is  --数码管不工作时各位显示 0
                    when 0=>
                        smgstate<="111110";
                        smgshow<=smgshowed(0);
                    when 1=>
                        smgstate<="111101";
                        smgshow<=smgshowed(0);
                    when 2=>
                        smgstate<="101111";
                        smgshow<=smgshowed(0);
                    when 3=>
                        smgstate<="011111";
                        smgshow<=smgshowed(0);
                end case;
            end if;
        end if;
    end if;
end process;

cntbeep:process(clk_beep)                --蜂鸣器频率计数模块
begin
    if(clk_beep'event and clk_beep='1') then
        if(tz>0 and tz<=10) then      --到倒计时剩余 10s 时开始计数
            if(cntbeepz=99) then   --频率计数范围为 0-99，对应 10s
                cntbeepz:=0;
            else
                cntbeepz:=cntbeepz+1;
            end if;
        else
            cntbeepz:=0;
        end if;
        cntbeeps:=cntbeepz/10;
        cntbeepg:=cntbeepz-10*cntbeeps;
    end if;
```

```
    end process;

    beep1:process(clk_beep,beepreset,beepwork,beeps)
    begin
        if(beepreset='1') then
            beeps<='0';
        else
            if(clk_beep'event and clk_beep='1') then
                if(beepwork='1') then
                    if(cntbeeps>=0 and cntbeeps<=5 and
                        cntbeepg>=0 and cntbeepg<=4) then    --倒计时 10s-6s 每秒的前半秒
                        beeps<='0';                          --蜂鸣器不响
                    elsif(cntbeeps>=0 and cntbeeps<=5 and
                        cntbeepg>=5 and cntbeepg<=9) then    --倒计时 10s-6s 每秒的后半秒
                        beeps<='1';                          --蜂鸣器响(频率为 1Hz)
                    elsif(cntbeeps>=6 and cntbeeps<=9 and
                        (cntbeepg=0 or cntbeepg=2 or
                        cntbeepg=4 or cntbeepg=6 or
                        cntbeepg=8) ) then                   --倒计时 5s-0s 每秒分成十段后的偶数时段
                        beeps<='0';                          --蜂鸣器不响
                    elsif(cntbeeps>=6 and cntbeeps<=9 and
                        (cntbeepg=1 or cntbeepg=3 or
                        cntbeepg=5 or cntbeepg=7 or
                        cntbeepg=9) ) then                   --倒计时 5s-0s 每秒分成十段后的奇数时段
                        beeps<='1';                          --蜂鸣器响(频率为 5Hz)
                    else
                        beeps<='0';
                    end if;
                else
                    beeps<='0';
                end if;
            end if;
        end if;
        beep<=beeps;
    end process;

end main;
```

五、功能说明及资源利用情况

（一）、功能说明

完成了基本要求与提高要求，并自拟了部分功能，具体分述如下。

（1）、开机后，先通过拨码开关 SW0 选择地图，本游戏有两种地图可选，SW0 选 0 为地图 1，SW0 选 1 为地图 2，然后通过按下按键 BTN7（开始键），就可以进入游戏。

（2）、进入游戏后，8*8 点阵先是显示开机画面（绿色单词“GO”）两秒钟，然后显示红色地图与绿点。显示地图的同时，计时器、计步器开始工作，六位数码管的低两位显示 30 秒倒计时，高两位显示当前已走步数。

（3）、通过按下按键 BTN3~BTN0（即上、下、左、右四个键）移动绿点，如果不碰壁，则绿点移动到下一位置，同时计步器步数加 1 并由数码管显示；如果碰壁，则绿点保持不动，计步器也不会计步。

（4）、如果游戏剩余时间在 10s~5s 之间，则蜂鸣器以 1Hz 的频率发出提示音；如果游戏剩余时间在 5s~0s 之间，则蜂鸣器以 5Hz 的频率发出提示音。

（5）、如果在 30s 内移动绿点到出口，则游戏胜利，点阵显示绿色“V”，否则游戏失败，点阵显示红色“X”。

（6）、自拟功能，游戏中的任意时刻都可以通过按下按键 BTN6（复位键）进行游戏复位，使游戏回到刚开机时的状态。

（二）、资源利用情况

Flow Status	Successful - Wed Nov 12 11:17:41 2014
Quartus II Version	9.1 Build 222 10/21/2009 SJ Full Version
Revision Name	maze
Top-level Entity Name	maze
Family	MAX II
Device	EPM1270T144C5
Timing Models	Final
Met timing requirements	No
Total logic elements	611 / 1,270 (48 %)
Total pins	47 / 116 (41 %)
Total virtual pins	0
UFM blocks	0 / 1 (0 %)

实验使用的硬件开发板是 MAXⅡ EPM1270T144C5 开发板，由上图可以看出，共使用了 611 个逻辑单元，占所有逻辑单元资源的 48%；使用了 47 个管脚，占所有管脚资源的 41%。

六、故障及问题分析

（1）、关于状态控制器中状态值的设定问题，本来打算只使用一种变量，即判断各模块是否工作的各 work 变量，但是如果这样编写程序，编译老是通过不了。后来在状态控制器里增加了判断各模块是否复位的各 reset 变量，然后就能编译通过了。

（2）、关于蜂鸣器的问题，在刚开始使用蜂鸣器时，蜂鸣器怎么也不会响，我以为是自己的程序有问题，但修改了程序后蜂鸣器仍然不响。后来请教了同学才知道，需要将开发板蜂鸣器旁边的按键按下去，蜂鸣器才能正常工作。我按下蜂鸣器旁边的按键后，蜂鸣器果然就能正常工作了。

（3）、本游戏系统的初期版本中，蜂鸣器只会在系统开机后的第一次游戏时，才会准确在倒计时剩余 10s 时响起提示音；如果接着进行第二次或第多次游戏，则蜂鸣器响起提示音的时间是随机的。

经过分析发现，原来是控制蜂鸣器响起的循环计数器（程序中通过判断计数器的值来确定送给蜂鸣器低电平还是高电平）在系统开机后便不停地循环计数，即使游戏结束后仍然计数。由于开始第二次和第多次的游戏时间是随机的，这就导致了循环计数器在第二次和第多次游戏开始时的值是随机的，从而导致了蜂鸣器响起提示音的时间是随机的。

解决办法是将计数器通过一个单独的进程完成，该进程在游戏结束后便停止工作，在游戏开始时又重新工作。这就保证每次游戏开始时，循环计数器都是由确定的初始值（即 0）开始计数的，从而蜂鸣器每次响起提示音的时间也是确定的。

（4）、由于我的整个游戏系统是由一个程序完成的，并没有通过一个个分立元件的连接来组成系统，这就导致如果进行波形仿真，则需要对整个系统进行仿真，那么仿真时的 End Time 就得设置为 30s，但显然这是不现实的，因为计算机要仿真完这 30s 需要花费的时间是极其漫长的。

但是不仿真又是不行的，于是我将程序中的各个模块单独拿出来，又新建了若干工程，然后单独对这些模块进行仿真，得到了最终的仿真结果。

（5）、地图 1 中的出发点与终点处于同一行，并且出发点在第 7 列，终点在第 0 列。在本游戏的初期版本中，当绿点处于出发点时，如果按下向右键，则绿点会移动到终点，并显示游戏胜利，这种情况显然是不允许存在的。

经过分析发现原来是控制绿点位置的变量 x（横坐标）与 y（纵坐标）值的范围是 0~7，由于程序中对 x 与 y 的操作，会使 x 和 y 的值可能变为-1 或 8，于是将 x 和 y 的取值范围改为-1~8，问题就解决了。