Project 设计原理

朱稼乐 3120000346

一、源代码文件列表：

1.bomb_mask.v //炸弹颜色蒙版
2.crack_mask.v //火焰颜色蒙版
3.man1mask.v //人物 1 颜色蒙版
4.man2mask.v //人物 2 颜色蒙版
5.wallmask.v //墙壁图案颜色蒙版
6.winmap.v //WIN 界面显示
7.move_man1.v //绿色人物移动
8.move_man2.v //蓝色任务移动
9.pbdebounce.v //按键去抖动以及 1ms 计时器
10.put_bomb1.v //绿色人物放炸弹
11.put_bomb2.v //蓝色人物放炸弹
12.crack_signal.v //判定火焰信号是否显示
13.endflag.v //判定游戏是否结束
14.vga_sync.v //vga 显示信号计数器
15.TOP.v //顶层文件
16.bomb.ucf

二、图形显示

vga_sync.v 代码起的是一个计数器作用，代码取自“*FPGA prototyping by Verilog examples*”一书的第十三章。通过 vga_sync 模块传出的 px py 信号确定了目前扫描到的像素方位。

```
    always @(posedge clk , posedge reset) begin
            if (reset)
                rgb_reg <= BLACK;
            else if (flag[2]) begin
                if (end_on) rgb_reg <= GREEN;
                else rgb_reg <= BLACK;
            end
            else if (flag[1]) begin
                if (end_on) rgb_reg <= BLUE;
                else rgb_reg <= BLACK;
            end
            else if (wall_on)
                rgb_reg <= wall_col;
            else if (bomb_on)
                rgb_reg <= bomb_col;
            else if (player1_on)
                rgb_reg <= man1_col;
            else if (player2_on)
                rgb_reg <= man2_col;
            else if (crack_on)
                rgb_reg <= crack_col;
            else
                rgb_reg <= BLACK;
    end
// output
    assign rgb = rgb_reg;
```

上图是 TOP.v 中的显示图像的主要部分，当 wall_on\bomb_on\player1_on 等信号为 1 时，rgb 信号就会被赋值相应的颜色信号。其中各个信号的赋值如下定义：

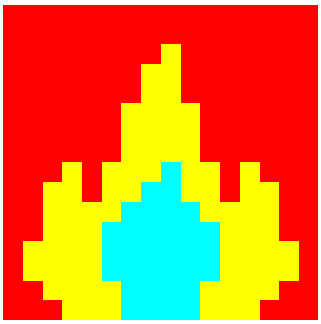
```
assign player1_on = (px/block_width==x1)&&(py/block_width==y1);
assign player2_on = (px/block_width==x2)&&(py/block_width==y2);
```

```
move_man1 m1(clk, btn1, bomb_x, bomb_y, x2, y2, x1, y1);
move_man2 m2(clk, btn2, bomb_x, bomb_y, x1, y1, x2, y2);
```

move_man 模块输出 x1, y1, x2, y2 等 40*30 的人物定位，便于 player_on 信号确定在当前坐标下是否应该显示人物颜色（man1_col 和 man2_col）。

```
bombmask m7 (clk, px, py, bomb_col);
man1mask m9 (clk, px, py, man1_col);
man2mask m10(clk, px, py, man2_col);
wallmask m11(clk, px, py, wall_col);
crackmask m13(clk, px, py, crack_col);
```

从上图可以看见 bomb_col(炸弹颜色显示)、man1_col 和 man2_col(人物颜色显示)、crack_col(火焰颜色显示)等都是通过其他模块输出来决定的，这些模块都存在于各个对应的 mask 文件中。以火焰颜色显示为例：
首先在画图中画出火焰形状（16*16 像素 bmp）：



我写了一个 C 程序对该 bmp 图像文件进行转化，通过程序输出 RGB 三色的蒙版，在 crack_mask.v 中可以看到上图对应的绿色蒙版（上下颠倒）：

```
green[15] = 16'b0001111111111000;
green[14] = 16'b0011111111111100;
green[13] = 16'b0111111111111110;
green[12] = 16'b0111111111111110;
green[11] = 16'b0011111111111100;
green[10] = 16'b0011111111111100;
green[9] = 16'b0011011111101100;
green[8] = 16'b0001011111101000;
green[7] = 16'b0000001111000000;
green[6] = 16'b0000001111000000;
green[5] = 16'b0000001111000000;
green[4] = 16'b0000000110000000;
green[3] = 16'b0000000110000000;
green[2] = 16'b0000000010000000;
green[1] = 16'b0000000000000000;
green[0] = 16'b0000000000000000;
```

通过 RGB 三色叠加（或运算）即可形成要显示的图像。

三、人物操作及其它

从（一）中的注解可以大致了解各部分文件的功用，此处不进行赘述，具体可见源代码文件夹中的代码（附带注释）。