

贪吃蛇游戏 FPGA 实现

郑佳培

17300750021

微电子科学与工程

摘要

项目的主要内容是利用 Verilog 语言编写贪吃蛇游戏，并且通过软件 Quartus II 在 Cyclone III 系的 EP3C25Q240C8 芯片上实现。

为了能够更加深入的了解硬件电路，本次课程作业最终决定实现贪吃蛇游戏，因为它是一个逻辑复杂同时又易于显示的游戏，因此我们可以更加强调逻辑，从而更加强调硬件实现部分。

贪吃蛇游戏的玩法就是操控一条虚拟的蛇吃下苹果，每吃下一个苹果就会获得一分，同时蛇的长度也会增加一格。一旦蛇碰到了边界或者自己的身体，游戏结束。

游戏的复位 (rst)、开始 (start) 以及控制蛇运动的上下左右 (up、down、left、right) 由开发板上的独立按键产生，游戏界面显示在 16*16 的 LED 点阵上，游戏分数显示在两位的七段数码管上。

顶层设计

贪吃蛇游戏采用了模块化的设计，在对游戏机制充分了解的基础上简化了硬件设计。

首先，对顶层模块来说，包括六个输入信号：复位 (rst)、开始 (start) 以及上下左右 (up、down、left、right)，这些信号控制着游戏的进程。包括 16+16 的三十二位 LED 扫描显示信号，其中 16 位信号控制 LED 点阵的行，16 位信号控制 LED 点阵的列。另外 7+7 的十四位的两位数码管显示信号。

顶层模块之下，主要包括分频模块、按键消抖模块，游戏模块、显示模块，接下来将详细介绍各个模块。

分频模块

输入 clk，输出由于 LED 点阵扫描显示的 800Hz 信号 clk_800 和用于控制蛇运动快慢的信号 clk_5。

实现方式为：用 cnt 信号进行计时，当 cnt1=30000 时，clk_800 输出 1；当 cnt2=4800000 时，clk_5 输出 1。

按键消抖模块

按键消抖电路，输入按键信号，输出消抖后的按键信号。

其中消抖的方式是在信号跳转后延迟 20ms 再采集。若此时信号与跳转前相同，则认为没有发生按下的操作，若信号与跳转后相同，则认为信号确实发生改变。

电路由六个状态的状态机构成，分别为低电平态，计数延迟态，判断态，高电平态，计

数延迟态，判断态。

注意：为了与一般的逻辑相对应，我们进行了额外进行了取反操作，即按下时输入为 0，经滤波后输出为 1。

其中计数延迟由计数器来控制，当 en=1 时开始计数，当计数值达到 500000 时，因为晶振为 24MHz，所以即 20.8ms 时，输出溢出信号，flag=1。

有限状态机的状态转移表如下图所示：

现态	次态			输出	
	Key_in=0	Key_in=1	Flag=1	En	Keyout
S0	S1			0	0
S1			S2	1	0
S2	S3			0	0
S3		S4		0	1
S4			S5	1	1
S5		S0		0	1

表 1. 按键消抖状态转移表

游戏模块

此模块为贪吃蛇游戏的主体，主要实现控制蛇的运动，苹果的产生和被吃，死亡的判断。为了实现这些功能，此模块下又分为很多小模块。

1. 按键信号持续模块

因为只要按一次方向键，蛇就会一直朝着此方向运动，直到下一个方向键被按下，所以此模块的作用就是将脉冲型的按键信号转换为连续信号。

需要注意的是，当按键方向与蛇当前运动方向相反时，按键输入信号无效，蛇会按照之前的运动方向继续运动下去。

另外，人为规定：初始运动方向为向右运动。

通过 if-else 的级联来实现此部分模块。如：

```

.....
else if (up==1'b1 && d_down == 1'b0) begin
    d_up =1; d_down=0; d_left=0; d_right =0;
end
else if (down==1'b1 && d_up== 1'b0) begin
    d_up =0; d_down=1; d_left=0; d_right =0;
end
else if
.....
    
```

2. 蛇头控制模块

对于蛇的运动，设计的思路是只控制蛇头的运动，剩余蛇身的运动跟随前一节的运动即可。

蛇头运动的控制信号由按键信号持续模块产生，当蛇头控制模块接收到信号后，在相应的方向进行一位的移位操作即可。

其实现的部分代码如下：

```

.....
else if(d_down)begin
    x[0]<=x[0];
    y[0]<=y[0]>>1;
end
else if(d_left)begin
    x[0]<=x[0]<<1;
    y[0]<=y[0];
end
else if
.....

```

3. 蛇身跟随模块

此模块的功能是当蛇头在运动时，控制蛇身跟随蛇头进行运动。

具体的实现过程就是在每个 clk_5 上升沿的时候，如果此时蛇在运动，即 d_up、d_down、d_left、d_right 不全为 0，则将上一段蛇的横坐标和纵坐标赋值给下一段。此外，蛇身的坐标保持原样。

其实现的部分代码如下：

```

.....
if(d_up||d_down||d_left||d_right)begin
    x[1]<=x[0];y[1]<=y[0];x[2]<=x[1];y[2]<=y[1];x[3]<=x[2];y[3]<=y[2];
    x[4]<=x[3];y[4]<=y[3];x[5]<=x[4];y[5]<=y[4];
.....

```

4. 苹果与得分模块

此模块主要判断苹果有没有被吃，以及控制新苹果的产生。

苹果有没有被吃的判断非常简单，只要比较蛇头坐标与苹果坐标即可。若蛇头坐标与苹果坐标完全一致，则说明苹果被吃了，产生 eat=1 的信号，分数+1，否则，说明苹果没有被吃，产生 eat=0 的信号，分数不变。其中，eat 信号用于控制新苹果产生模块的状态转移。详细情况将在稍后介绍。

其实现的部分代码如下：

```

.....
if(xx[0]==apple_x&&yy[0]==apple_y) begin
    score<=score+1;
    eat<=1;
end
.....

```

新苹果的产生从游戏角度来说应该是随机的，然而在逻辑电路中无法做到真正的随机，因此，我们写了一个“随机数发生器”。

“随机数发生器”的主体就是一个接着高速时钟的计数器，计数的上限、下限以及步长由传入的参数决定。当我们在某个时刻读取“随机数发生器”时，就会得到一个近似随机的数。我们就将这个数作为新苹果的坐标。

其实现的部分代码如下：

```
always@(posedge clock)
begin
    if(counter == max_number)
        counter <= min_number;
    else
        counter <= counter + step;
end
```

注意到，新苹果不能与已有的蛇身坐标相重叠，所以需要描述一个状态机，用于判断新产生的苹果是否与蛇身相重叠，如果重叠，则再产生一个新的坐标，如果不重叠，则将这个新苹果的坐标赋值给苹果。

有限状态机的状态转移表如下图所示：

现态	次态		
	eat=0	eat=1	不与蛇身重叠
S0		S1	
S1			S2
S2	S0		

表 2. 新苹果坐标判断状态转移表

5.蛇身长度调控模块

在游戏中，蛇身的长度会随着吃掉的苹果而逐渐增长，此模块的功能就是控制蛇身的长度。

我们规定，初始状态的蛇身长度为 4，此外，每吃下一个苹果，蛇身长度就+1。

为了实现这样的效果，我们例化了一组新的蛇身坐标寄存器，它们的初始值全部为 0，并且只将蛇身坐标寄存器的索引比长度值小的寄存器赋上坐标值。

其实现的部分代码如下：

```
if(size>3)begin
    xx[0]<=x[0];yy[0]<=y[0];xx[1]<=x[1];yy[1]<=y[1];
    xx[2]<=x[2];yy[2]<=y[2];xx[3]<=x[3];yy[3]<=y[3];
end
if(size>4) begin xx[4]<=x[4];yy[4]<=y[4]; end
if(size>5) begin xx[5]<=x[5];yy[5]<=y[5]; end
if(size>6) begin xx[6]<=x[6];yy[6]<=y[6]; end
.....
```

6.死亡模块

此模块的功能就是判断蛇有没有死亡，即判断蛇头是否与边界重合或者与自身重合，若重合则输出死亡信号 DIE=1，否则，输出死亡信号 DIE=0。

其实现的部分代码如下：

```
if(xx[0]==16'b0|| yy[0]==16'b0)
    DIE<=1;
else if (
```

```

xx[0]=xx[1]&&yy[0]=yy[1]||xx[0]=xx[2]&&yy[0]=yy[2]||xx[0]=xx[3]&&
yy[0]=yy[3]||xx[0]=xx[4]&&yy[0]=yy[4]||xx[0]=xx[5]&&yy[0]=yy[5]||……||xx[0]=
xx[99]&&yy[0]=yy[99])
    D/E<=1;
else
    D/E<=0;

```

显示模块

此模块主要负责将蛇和苹果显示在 LED 点阵上，以及将游戏分数显示在数码管上。

LED 点阵显示需要进行扫描，以 800Hz 的频率进行扫描，每一行的扫描频率为 50Hz，因为人眼的视觉暂留效应和 LED 的余晖效应，就能够做到 LED 点阵的同时显示。

当扫描到某一行时，其对应的列信号又对应的蛇的列坐标和苹果的列坐标相或而成。

另外，注意到此 LED 点阵是低电平点亮，所以最后输出时进行取反操作。

分数的显示就更简单了，首先对输入的分数进行二进制到 BCD 码的译码转换，接着进入 BCD 码-七段译码电路，就得到相应的七段码。

另外，注意到数码管是低电平点亮，所以最后输出时进行取反操作。

项目成果

硬件描述文件经过分析、综合、实现、位流生成能顺利在开发板上运行，其资源消耗情况入下表。

Total logic elements	11,745 / 24,624 (48 %)
Total combinational functions	7,101 / 24,624 (29 %)
Dedicated logic registers	6,672 / 24,624 (27 %)
Total pins	53 / 149 (36 %)

表 3. 器件逻辑资源消耗情况

游戏运行过程。

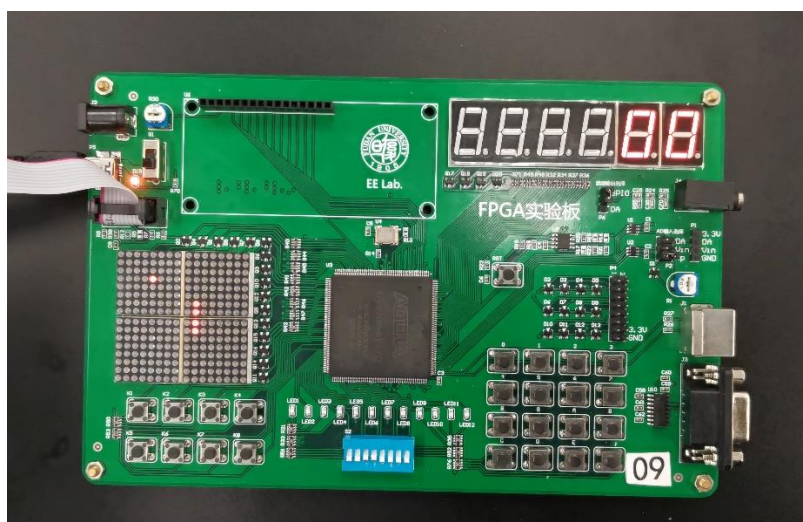


图 1. 初始态

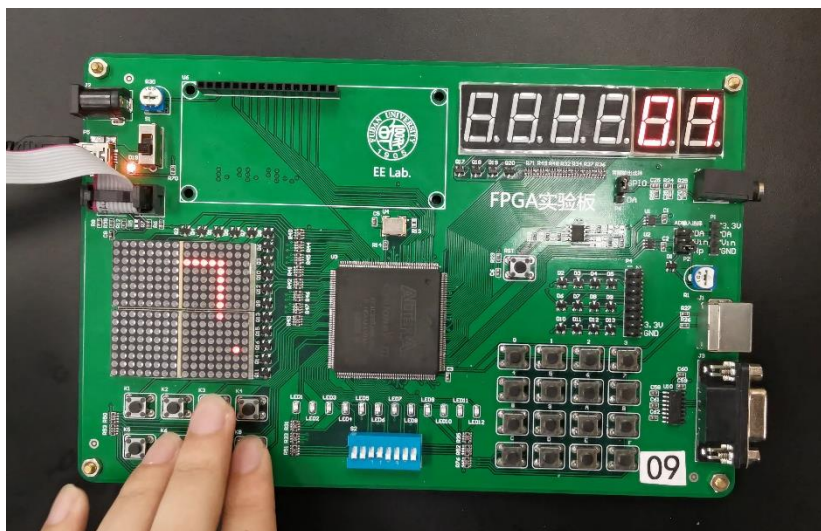


图 2. 游戏中

总结

经过几周的学习和实践，顺利地完成了预定的贪吃蛇游戏项目。通过项目，对 Verilog 语言有了更深的了解，对代码语言和硬件电路之间的关系也有了了解。更好地掌握了仿真文件的编写，并且能够通过仿真波形对代码进行 debug。

这些都为以后参与更加复杂、更加全面和系统的项目夯实了基础。

最后要感谢老师和助教一学期的指导，这门课对我的帮助很大，万分感谢！