

## 基于 FPGA 的数码管计数和贪吃蛇动画

### 一、设计思路

首先是最基础的部分，在八个数码管上显示全 1。因为之前自学过 51 的单片机，我的那个开发板是通过 3-8 译码器，可以直接计数来选中八个数码管中的一个，但是并不能够同时选中多个数码管，而是通过余晖效应来实现所有数码管同时选中的。所以我一开始是直接写了一个通过移位扫描每个数码管来显示此时段码的，之后的部分也可以通过这种方式来实现，这样我就分成了位选和段选两个部分，都是通过 always 来实现的。然而，后来我发现了 FPGA 厉害的地方，我其实可以直接把与数码管位对应的一个寄存器全部配置成高电平，这样我就能全部选中，而且此时因为全显示 1 并且数码管的对应的段都是连通的，我只要而配置一组段选值为 1 的就可以了。这一部分我又把它称为静态部分。

静态之后自是动态，我通过设置一个开关作为模式选择，在静态动态间进行切换，接着就是以分频后 5Hz 的频率计数并转换为 bcd 码然后在数码管上显示。可以很清晰地发现这是三步走，我先设计一个分频计数模块，然后再将分频计数的值传给 16 进制转 bcd 码的模块，最后再将 bcd 码低四位对应的段值在数码管显示模块中作为个位位选时的段码值，以此类推中间四位作为百位，最高两位再和两个零进行拼接凑够四位，以便在数字段码值函数中找到段码。

基本思路后再说一下详细设计，分频首先是分频为 5Hz，这一部分，我尝试和上次不一样的方法，通过设定时钟计数上限为 20000000，每到上限，我的计数值就加一，也就是我计数的周期是 0.2s，从而实现 5Hz；bcd 码转

换部分，参考老师提供出的代码；数码管显示部分，我通过 FPGA 自带的时钟进行扫描位选，然后设计了三个状态的状态机，因为最终要显示的数的上限是 3 位，那么我只要扫描三位就行，也就设计三个状态。每一个状态选中一位，并将该位的段码值赋给段码锁存器。然后状态计数加一，在最后一个状态爱又将状态清零以实现状态的循环。然而，当我觉得天衣无缝的时候，生成比特流后，硬件的反应给了我当头一棒。

## 二、遇到问题以及调试

静态显示没有问题，但是一到动态，就显示得非常奇怪，字影残缺，字体很暗，而且七段数码管的中间一横总是不显示的，一开始我觉得可能是管脚约束错了，因为我静态时是显示 1，不存在 g 段的影响，然而再三检查后似乎没有错，我决定试一下，于是我将静态的值显示为 2。下载后发现，静态显示丝毫没有问题。很好，那么我就排除了是约束的问题，现在问题就只在动态扫描了。我仔细检查了代码，难道是 bcd 转换的地方出了问题？为了检查是不是这个原因，我觉得动态扫描输出一个我直接给定的规定值，结果这个问题仍然存在，那么也不是 bcd 转换的问题；于是我就开始怀疑可能是我阻塞或非阻塞赋值，以及我把段选和位选写了两个 always 导致的中间的逻辑错误。于是我反复查了好几遍，把两个 always 也精简成了一个 always，而且非常简洁和清晰的结构。然后即使就是这样，字体很暗，显示残缺的问题还在，于是我就想是不是扫描太快了没有消隐，导致前一个值对现在有影响，但是我又想，我现在给的固定值不存在要消隐的问题，我仔细观察数码管的现象，我非常怀疑是上一个位的段选值对当前位的有影响导致，但是我记得 FPGA 书并行执行的，不应该会这样，难道还是我的时序问题？

几番修改仍然无果后，在确保我的逻辑没问题下，我开始大胆猜测，可能是扫描太快，FPGA 尽管是并行执行，但是还是存在一定延迟，再加上人眼的延迟，最后就变成这样，因为我在残影时，关闭模式选择开关——相当于暂停，数码管上可以清晰显示我要输出的数。于是我给数码管的扫描又加上了一个分频时钟大概是 10000 翻转一次。

然后奇迹就出现了!!! 我的数码管们开始清晰正常地开始显示数了!!! 真是又开心又气人啊! 所以这次我是明白了，延迟是不可以忽略的!拿到硬件就要从实际出发，不能理想化了。

### 三、创意部分

这个完成后就是提高部分了，提高部分我想到了在我调试时意外亮起来的数码管的小数点，于是我灵机一动，不如做个贪吃蛇吧。在动态扫描的基础上，我将贪吃蛇动画的每一帧像是动画一样描绘出来，我们知道在贪吃蛇吃掉当前事物后，贪吃蛇的身体就要加一段，下一个食物就要出现。这其实就是一个每一帧都是一个状态的状态机。我将每一帧的段值也编写一个类似数字段值的函数。最初设想是通过按键控制贪吃蛇前进，不过由于时间有限我只实现了两个数码管。

### 四、代码部分

我分为三个模块和一个顶层例化模块

首先是顶层文件：

```
module top(  
    input clk,  
    input rst,  
    //计数模式开关  
    input sel,  
    //贪吃蛇开关
```

```

input advance,
output [6:0] seg,
output [7:0] bit_154,
output ball
);
wire [7:0] cnt_154;
wire [9:0] p;
//5hz 分频，输出计数值
counter counter(
.clk(clk),
.rst(rst),
.sel(sel),
.cnt_154(cnt_154)
);
//数码管显示模块
segdisplay segdisplay(
.clk(clk),
.rst(rst),
.p(p),
.sel(sel),
.bit_154(bit_154),
.seg(seg),
.ball(ball),
.advance(advance)
);
//将计数值转化成 bcd 码
bcd bcd(
.b(cnt_154),
.p(p)
);
Endmodule

```

5Hz 计数模块:

```

module counter(
input clk,
input rst,
//计数模式开关
input sel,
output reg [7:0] cnt_154
);
reg [31:0] clk_cnt;
//实现 5hz
always@(posedge clk)
begin

```

```

if(rst)
begin
clk_cnt<=0;
cnt_154<=2;
end
else
if(clk_cnt==20000000)
begin
clk_cnt <= 0;
if(cnt_154==255)
cnt_154<=2;
//判断是否为计数模式
else if(sel)
//每 0.2s 计数一次
cnt_154=cnt_154+1;
end
else
clk_cnt<=clk_cnt+1;
end
endmodule

```

bcd 转换模块:

```

module bcd(
input wire [7:0] b,
output reg [9:0] p
);
reg [17:0] z;
integer i;
always@(*)
begin
for(i=0;i<=17;i=i+1)
z[i]=0;
z[10:3]=b;
repeat(5)
begin
if(z[11:8]>4)
z[11:8]=z[11:8]+3;
if(z[15:12]>4)
z[15:12]=z[15:12]+3;
z[17:1]=z[16:0];
end
p=z[17:8];
end
endmodule

```

## 数码管显示模块:

```
module segdisplay(
input clk,
input rst,
input [9:0] p,
input sel,
input advance,
output reg [7:0] bit_154,
output reg [6:0] seg,
output reg ball
);
    reg [2:0] cnt;
    reg [31:0] counter_154;
    reg [31:0] counter2_154;
    reg div_clk_154;
    reg [3:0] cnt2;
//七段数码管段选表
function [6:0] seg_154;
input [3:0] p_buf;
begin
case (p_buf)
0: seg_154 = 7'b0000001;
1: seg_154 = 7'b1001111;
2: seg_154 = 7'b0010010;
3: seg_154 = 7'b0000110;
4: seg_154 = 7'b1001100;
5: seg_154 = 7'b0100100;
6: seg_154=7 'b0100000;
7: seg_154=7'b0001111;
8: seg_154 = 7'b0000000;
9: seg_154 = 7'b0000100;
default: seg_154 = 7'b0000001;
endcase
end
endfunction
//贪吃蛇
function [6:0] seg2_154;
input [3:0] p2_buf;
begin
case(p2_buf)
0:seg2_154 = 7'b1111101;
1:seg2_154 = 7'b0111111;
2:seg2_154 = 7'b1011111;
```

```

3:seg2_154 = 7'b1101111;
4:seg2_154 = 7'b1100111;
5:seg2_154 = 7'b1110011;
6:seg2_154 = 7'b1111001;
7:seg2_154 = 7'b1111101;
8:seg2_154 = 7'b1001111;
9:seg2_154 = 7'b1000111;
10:seg2_154 = 7'b1100011;
11:seg2_154 = 7'b1110001;
12:seg2_154 = 7'b1111001;
13:seg2_154 = 7'b1111101;
14:seg2_154 = 7'b1000111;
endcase
end
endfunction
//扫描时钟
always@(posedge clk)
begin
if(rst)
    counter_154<=32'h00000000;
else
    if(counter_154 == 10000)
        begin
            counter_154 <=32'h00000000;
            div_clk_154<=~div_clk_154;
        end
    else
        counter_154 <=counter_154 +1;
end
//贪吃蛇时钟
always@(posedge clk)
begin
if(rst)
begin
    counter2_154<=32'h00000000;
    cnt2<=0;
end
else
    if(counter2_154 == 30000000)
        begin
            counter2_154 <=32'h00000000;
            if(cnt2==15)
                cnt2<=0;
        end
    else

```

```

        cnt2<=cnt2+1;
    end
    else
        counter2_154 <=counter2_154 +1;
    end
    //循环移位
    always@(posedge div_clk_154 or posedge rst)
    begin
        if(rst)
        begin
            bit_154 <= 8'b00000001;
            cnt <= 3'b000;
            ball <= 1;
        end
        else
        begin
            //判断模式
            if(sel)
            begin
                case(cnt)
                0: begin
                    bit_154 <= 8'b00000001;
                    seg <= seg_154(p[3:0]);
                    cnt <= cnt+1;
                end
                1:
                begin
                    bit_154 <= 8'b00000010;
                    seg <= seg_154(p[7:4]);
                    cnt <= cnt+1;
                end
                2: begin
                    bit_154 <= 8'b00000100;
                    seg <= seg_154({2'b0,p[9:8]});
                    cnt <= 0;
                end
                default:seg<=seg_154(4'b0000);
            endcase
        end
        //提高部分——贪吃蛇
        else if(advance)
        begin
            case(cnt)
            0: begin

```



```

bit_154 <= 8'b00000001;
if(cnt2<=3)
ball = 0;
else if(cnt2>=4)
ball = 1;
if(cnt2<=7)
seg <= seg2_154(cnt2);
else
seg <= 7'b1111111;
cnt <= cnt+1;
end
1:
begin
bit_154 <= 8'b00000010;
if(cnt2 >= 4 && cnt2 <=8)
ball <= 0;
else
ball <= 1;
if(cnt2==7)
seg <= 7'b1011111;
else if(cnt2>=8&&cnt2<=12)
seg <= seg2_154(cnt2);
else
seg <= 7'b1111111;
cnt <= cnt+1;
end
2: begin
bit_154 <= 8'b00000100;
seg <= 7'b1111111;
ball = 1;
cnt <= 0;
end
default:seg<=seg_154(4'b0000);
endcase
end
//基础部分
else
begin
bit_154 <= 8'b11111111;
seg <= seg_154(4'b0001);
end
end
end
endmodule

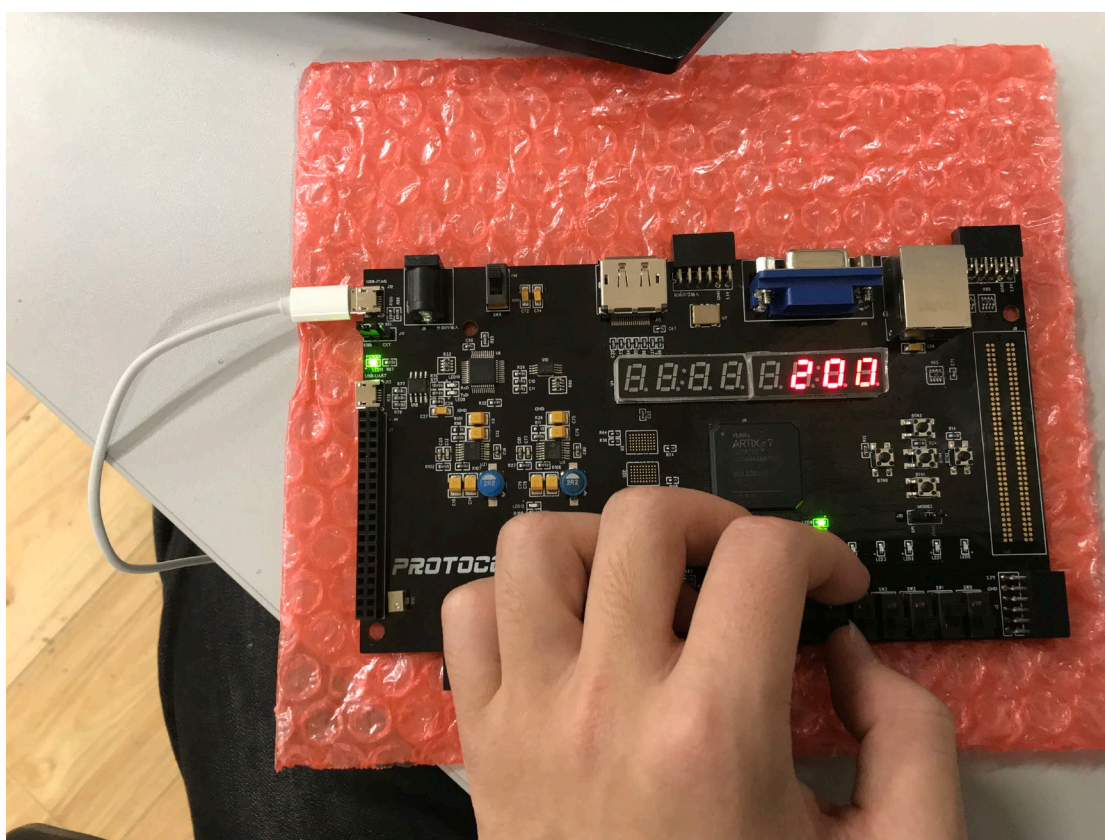
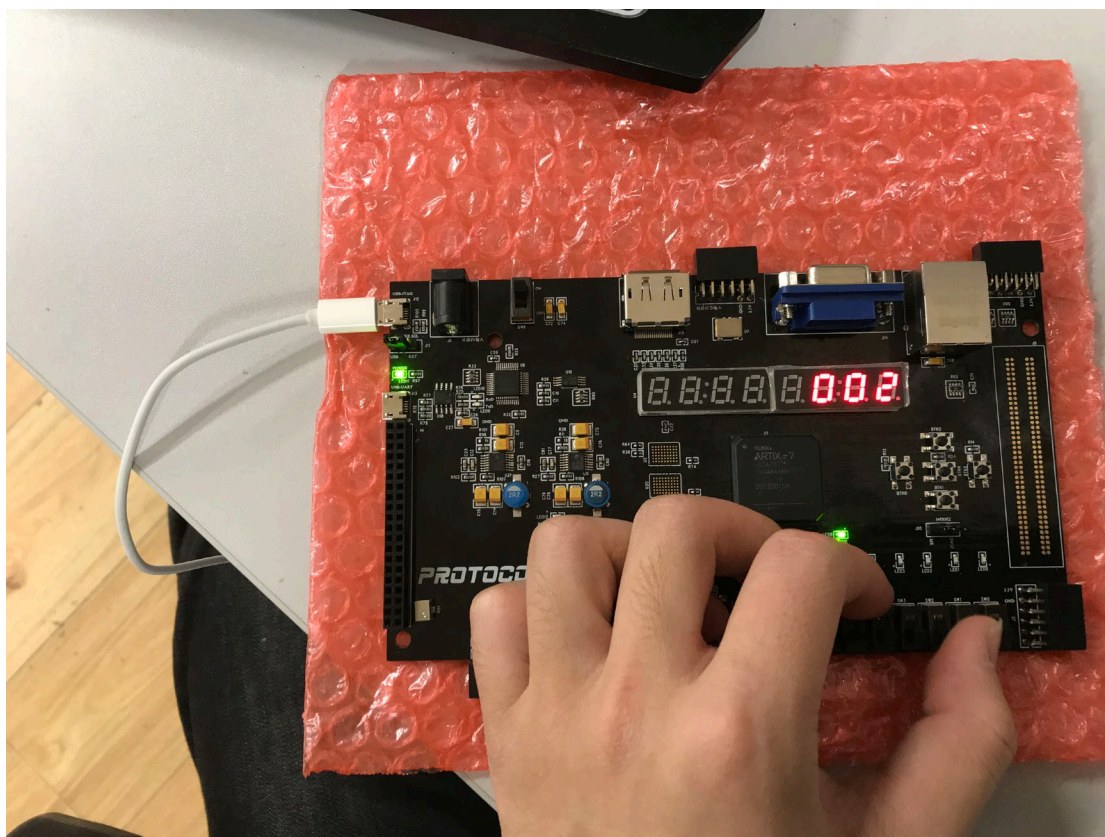
```

## 五、效果展示

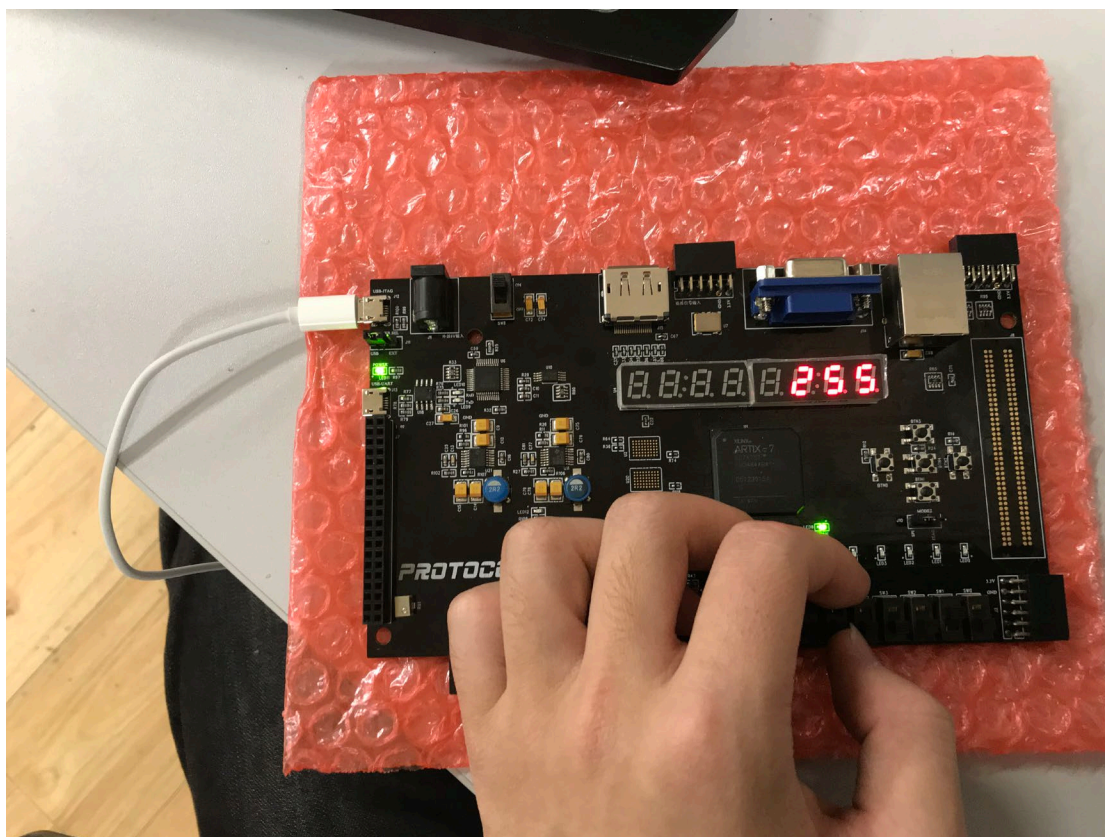
显示全 1:



计数显示:







贪吃蛇

