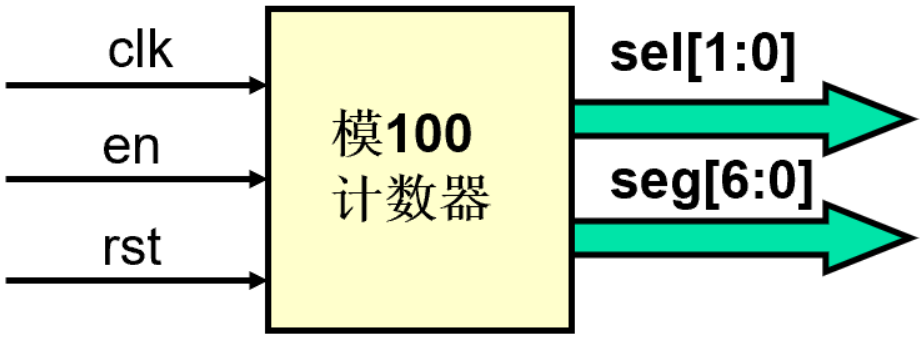
实验四 0-99 计数及扫描显示电路设计

一、 实验内容

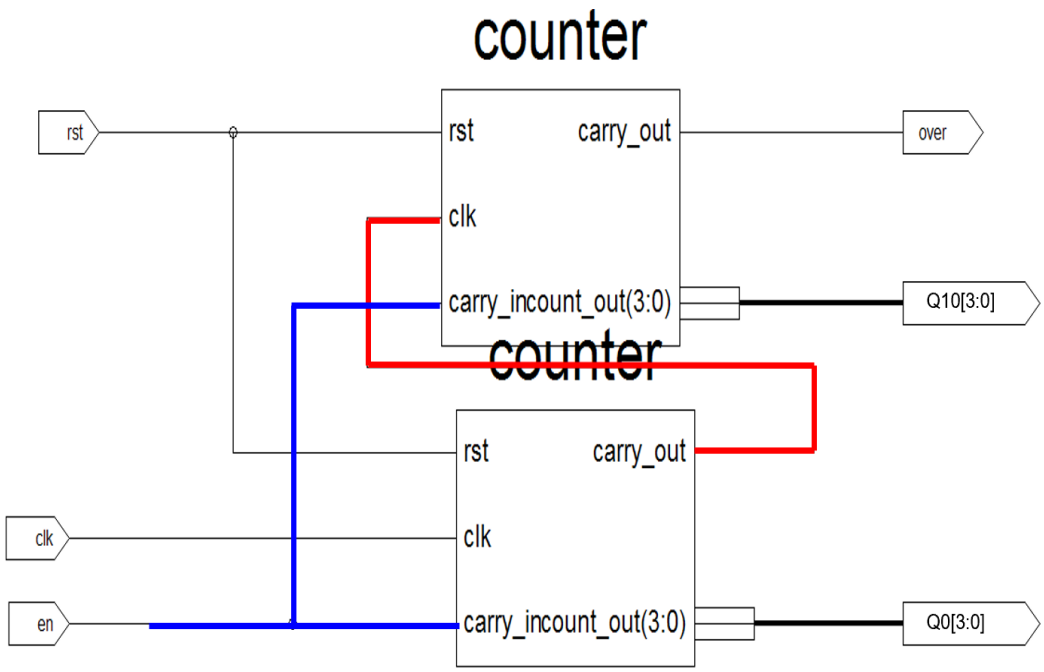
采用自顶向下的设计方法，完成 0-99 的计数及显示，输出用两个数码管。



二、 实验原理与方案

2.1 计数器模块

采用十进制异步复位，同步使能有进位输出的十进制计数器进行异步级联，形成 100 进制计数器，进行 0-99 计数显示。

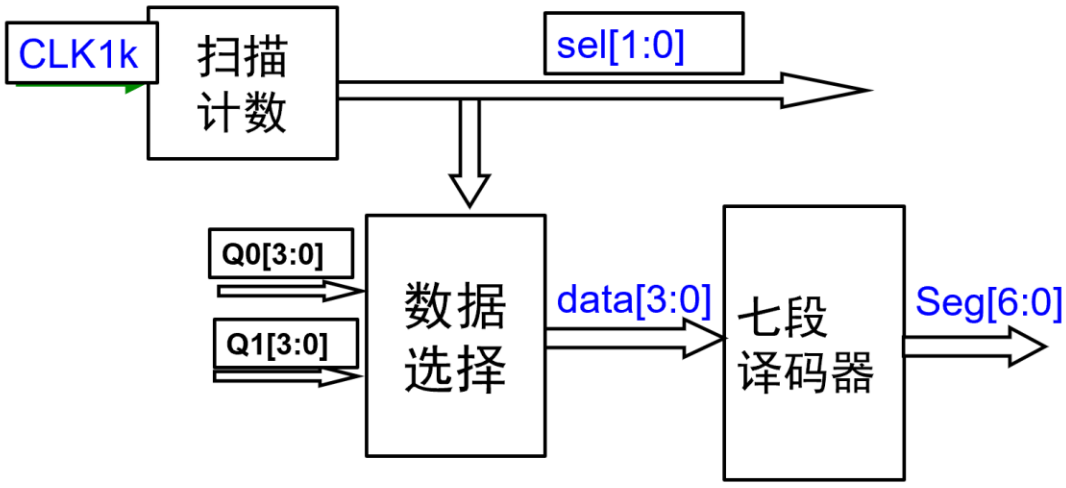


2.2 分频

100hz 分频给计数器计时使用，1khz 分频用于数码管动态显示。

2.3 数码管动态显示

每个数码管会收到同样的 7 段信号，我们可以通过控制数码管的位选端来实现数码管分时工作，当刷新频率达到一定程度时，人眼无法分辨从而实现数码管动态显示，频率太慢会导致人眼可以分辨，频率过快导致低速器件数码管系统功耗增加，显示效果变暗,1KHz 左右最好，当数码管数量过多时，可以将数码管分组实现动态显示。



三、 实验过程(源程序)

```
module top99(
clk_50mhz,
en,
rst,
sel,
seg
    );
  input clk_50mhz;
  input rst;
```

```
 input en;
 output reg[3:0] sel=0;
 output reg[6:0] seg=0;
 wire[3:0] q0,q1;
 wire carry_out;
 wire over;

 //100hz
 reg clk100hz=0;
 integer cnt2=1;
 always@(posedge clk_50mhz)
 begin
    if(cnt2==250000)//25000000
    begin
    cnt2=1;
    clk100hz=~clk100hz;
    end
    else
        cnt2=cnt2+1;
end

 counter10 u0(
    .clk100hz(clk100hz),
    .rst(rst),
    .carry_in(en),
    .cnt(q0),
    .carry_out(carry_out)
 );
counter10 u1(
.clk100hz(carry_out),
.rst(rst),
.carry_in(en),
.cnt(q1),
.carry_out(over)
);

//1khz
reg clk1khz=0;
 integer cnt3=1;
always@(posedge clk_50mhz)
begin
    if(cnt3==25000)//25000000
    begin
    cnt3=1;
```

```
    clk1khz=~clk1khz;
    end
    else
        cnt3=cnt3+1;
end
reg count=0;
always@(posedge clk1khz)
begin
count=~count;
end

always@(count)
begin
case(count)
0:sel=4'b1110;
1: sel=4'b1101;
default:sel=4'b1111;
endcase
end

reg[3:0] disp_data;
always@(count)
begin
    case(count)
        0:disp_data=q0;
        1:disp_data=q1;
        default: disp_data=0;
    endcase
end

always@(disp_data)
begin
  case(disp_data)
    0: seg<=7'b0000001;
    1:  seg<=7'b1001111;
    2: seg<=7'b0010010;
    3:  seg<=7'b0000110;
    4:  seg<=7'b1001100;
    5:  seg<=7'b0100100;
    6:  seg<=7'b0100000;
    7:  seg<=7'b0001111;
    8:  seg<=7'b0000000;
    9: seg<=7'b0001100;
    default: seg<=7'b1111111;
```

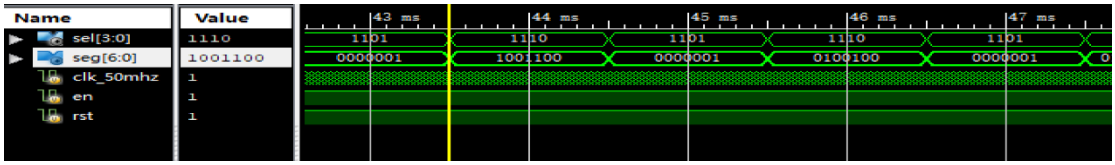
```
        endcase
     end
    endmodule
    //============10 进制计数器模块=====================
    module counter10(
    rst,
    clk100hz,
    carry_in,
    cnt,
    carry_out
        );
      input rst,clk100hz,carry_in;
      output reg carry_out;
      output reg[3:0] cnt=0;

      always@(posedge clk100hz or negedge rst)
      begin

      if(!rst)
      begin
      cnt=0;
      carry_out=0;
      end
      else if(carry_in)
      begin
      if(cnt==9)
         begin
         cnt=0;
         carry_out=1;
         end
         else
         begin
         cnt=cnt+1;
         carry_out=0;
         end
      end
      end
endmodule
```

四、 实验结果与分析(仿真程序与仿真图，硬件测试图)

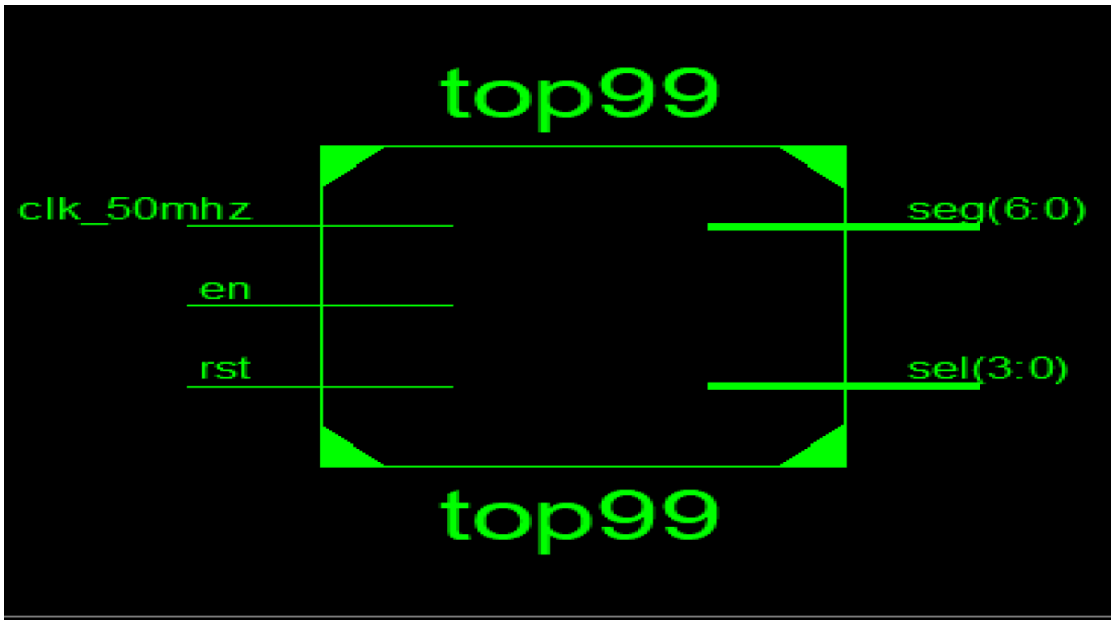
4.1 仿真程序

```
module top99_testbench;

    // Inputs
    reg clk_50mhz;
    reg en;
    reg rst;

    // Outputs
    wire [3:0] sel;
    wire [6:0] seg;

    // Instantiate the Unit Under Test (UUT)
    top99 uut (
        .clk_50mhz(clk_50mhz),
        .en(en),
        .rst(rst),
        .sel(sel),
        .seg(seg)
    );

    initial begin
        // Initialize Inputs
        clk_50mhz = 0;
        en = 0;
        rst = 0;

        // Wait 100 ns for global reset to finish
        #20;
        rst=1;
        en=1;


        // Add stimulus here
    end
    always
    begin
    #10;
    clk_50mhz=~clk_50mhz;
    end

endmodule
```

4.2 仿真图



4.3RTL



4.4 硬件测试图