[redacted]

数字逻辑课程综合实验报告

[redacted]

学　　号 [redacted]

姓　　名 [redacted]

专　　业 [redacted]

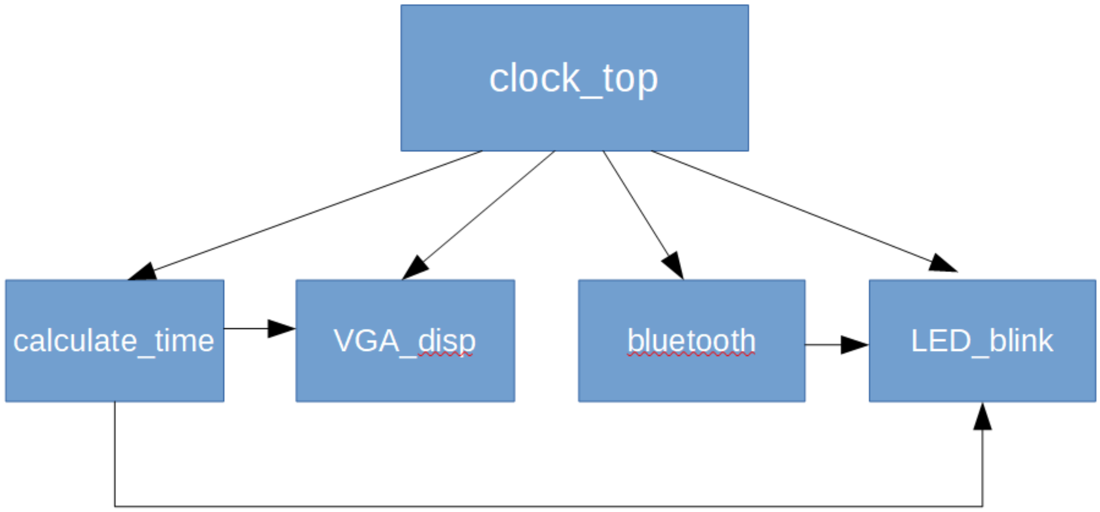
授课老师 [redacted]

一、实验内容

VGA 和蓝牙实现电子钟功能。

二、电子钟数字系统总框图

（按由顶向下方法进行子系统的划分，给出包含各子系统相互关系及控制信号的总框图，并对各子系统功能及实现进行概述。具体可参考教材 183 页的相关描述方法。）

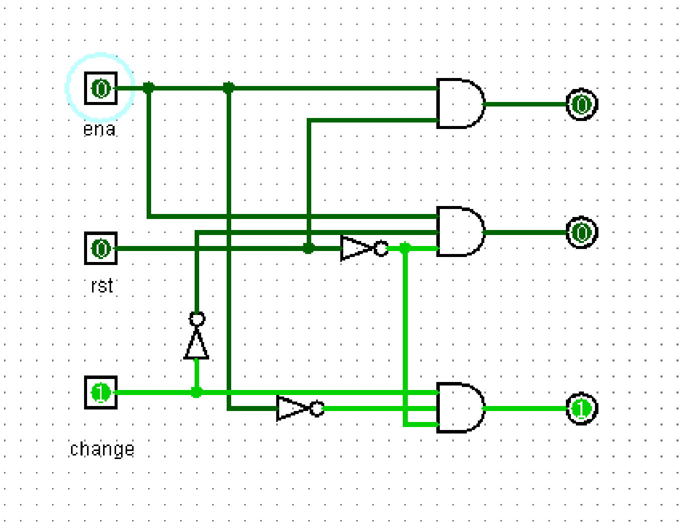


顶层模块（clock_top）：作为控制器，控制各个子系统的执行。
计时模块（calculate_time）：进行计时操作，有记录时间的功能，且可以切换计时状态和调整时间状态。
显示模块（VGA_disp）：用外设 VGA 进行显示，将时间显示在屏幕上。
蓝牙模块（bluetooth）：使用蓝牙外设进行闹钟的设定（整点）。
亮灯模块（LED_blink）：实现模拟闹钟的功能。当时间达到蓝牙设定的值，LED 开始亮一分钟。

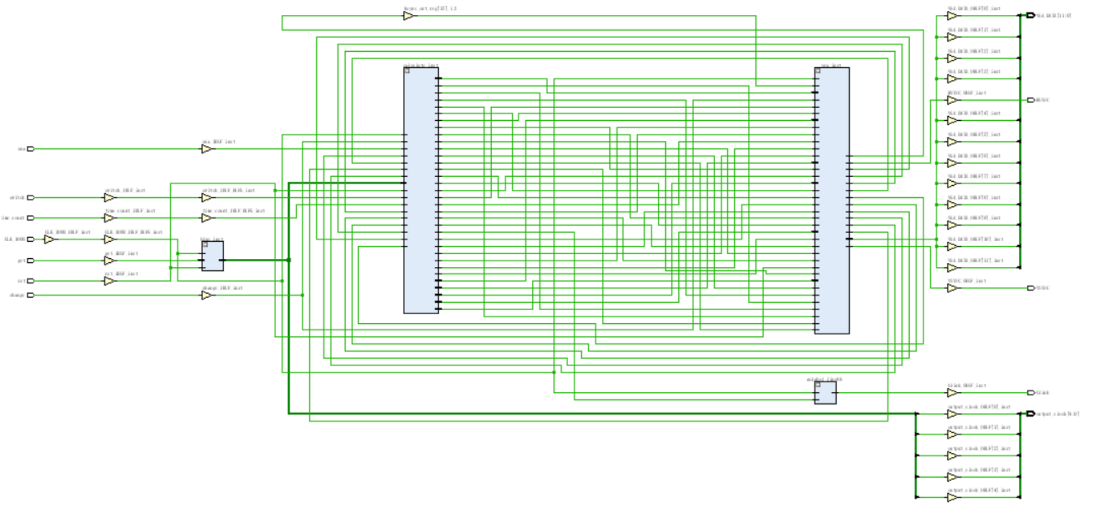
三、系统控制器设计

状态真值表：

现态（PS）	次态（NS）		
	X=00	X=01	X=10
休眠状态（00）	休眠状态（00）	自动计时+蓝牙接收（01）	手动调时+蓝牙接收（10）
自动计时+蓝牙接收（01）	休眠状态（00）	自动计时+蓝牙接收	手动调试+蓝牙接收（10）
手动调时+蓝牙接收（10）	休眠状态（00）	自动计时+蓝牙接收	手动调试+蓝牙接收（10）



RTL 总框图如图所示：



四、子系统模块建模

（该部分要求对实验中的所有子系统模块进行描述，给出各子系统的功能框图及接口信号定义，并列出各模块建模的 verilog 代码）

（一）计时模块（calculate_time）：进行计时操作，有记录时间的功能，且可以切换计时状态和调整时间状态。

输入接口信号定义：

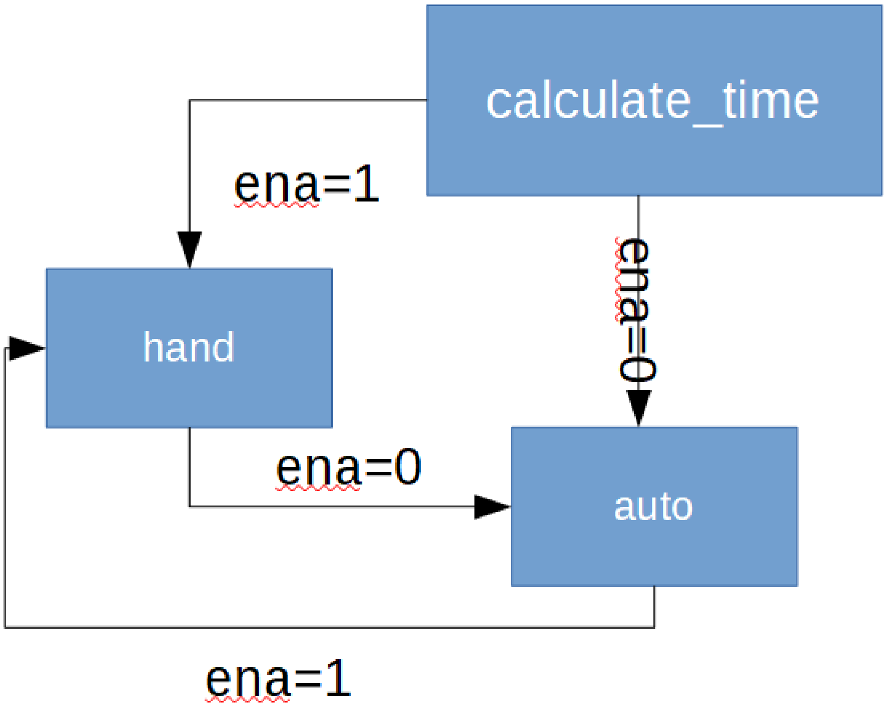
1、CLK_100M：输入时钟，使用 100MHz 的系统时钟作为输入；

2、ena：使能信号，高电平有效。使计时模块开始正常运作，低电平时所有值全部归零；

3、rst：复位信号，高电平有效。当高电平时整个子系统全部归零；

4、change：自动计时/手动调时信号，当高电平时切换为手动调时模式，自动计时功能暂停；当低电平时切换为自动计时模式，开始在手动调时结果的基础上自动计时；

5、switch：手动计时切换信号，上升沿有效。在时、分、秒的设置之间进行切换。
6、time_count：手动加一信号，上升沿有效。每到达一次上升沿，所设置的数字就进行一次加一操作。
输入接口信号定义：
1、hour_high：输出当前的小时高位；
2、hour_low：输出当前的小时低位；
3、minute_high：输出当前的分钟高位；
4、minute_low：输出当前的分钟低位；
5、second_high：输出当前的秒钟高位；
6、second_low：输出当前的秒钟低位；
功能框图如下图所示：



自动计时/手动调时部分代码：

```
always@(posedge CLK_1 or posedge rst)
    if(ena)
    begin
        if(rst)
            begin
                hour<=0;
                minute<=0;
                second<=0;
            end
        else if(change==0)
        begin
            if(second==6'd59)
            begin
                second<=0;
                minute<=minute+1;
                if(minute==6'd59)
                begin
                    minute<=0;
                    hour<=hour+1;
                    if(hour==5'd24)
                    begin
                        hour<=0;
                    end
                    else
                    ;
                end
                else
                ;
            end
            else
            second<=second+1;
        end
        else if(change==1)
        begin
        hour<=hour_hand;
        minute<=minute_hand;
        second<=second_hand;
        end
        end
    else
        begin
        hour<=0;
        minute<=0;
        second<=0;
        end
```

调时切换部分代码:

```
  reg [2:0] switch_reg;

    always@(posedge switch or posedge rst)
    if(rst)
        switch_reg<=0;
    else
    begin
        if(switch_reg==3'd3)
```

```
            switch_reg<=0;
        else
            switch_reg<=switch_reg+1;
    end

    always@(posedge time_count or posedge rst)
    if(rst)
    begin
        hour_hand<=0;
        minute_hand<=0;
        second_hand<=0;
    end
    else if(change==1)
    begin
        if(switch_reg==0)
        begin
            if(hour_hand==5'd23)
                hour_hand<=0;
            else
                hour_hand<=hour_hand+1;
        end
        else if(switch_reg==3'd1)
        begin
            if(minute_hand==6'd59)
                minute_hand<=0;
            else
                minute_hand<=minute_hand+1;
        end
        else if(switch_reg==3'd2)
        begin
            if(second_hand==6'd59)
                second_hand<=0;
            else
                second_hand<=second_hand+1;
        end
    end
    else
    ;
```

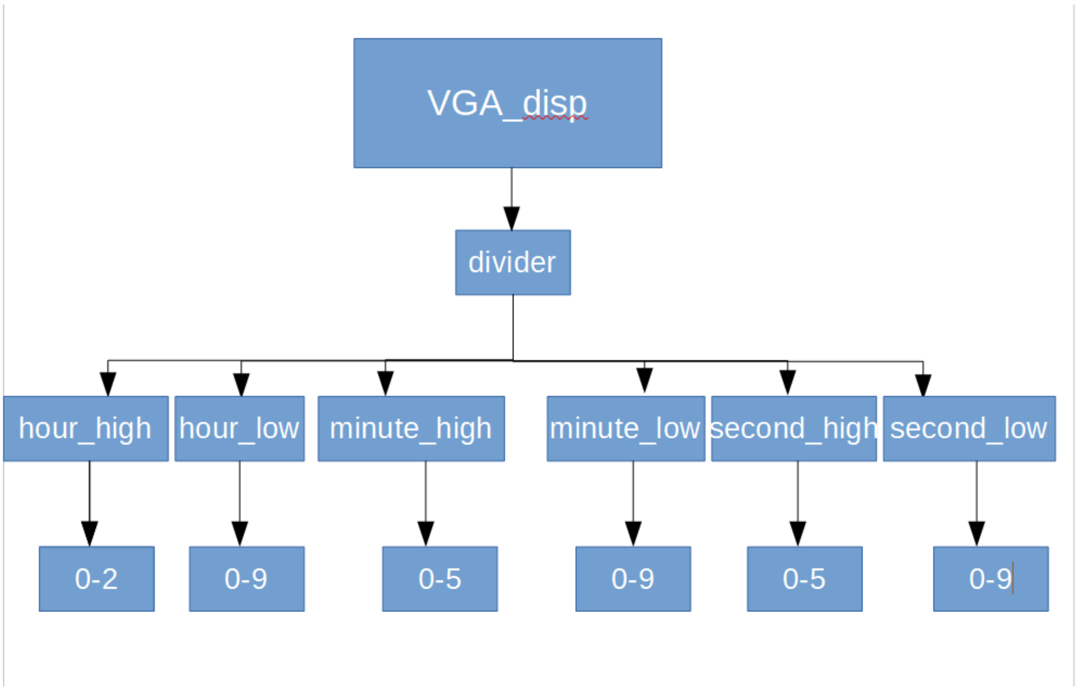
（二）显示模块（VGA_disp）：用外设 VGA 进行显示，将时间显示在屏幕上。

输入接口信号定义：

1、rst：复位信号，高电平有效。当高电平时整个子系统全部归零；

2、CLK_100M：输入时钟，使用 100MHz 的系统时钟作为输入（后续进行二分频）；

3、hour_high：输入当前的小时高位；
4、hour_low：输入当前的小时低位；
5、minute_high：输入当前的分钟高位；
6、minute_low：输入当前的分钟低位；
7、second_high：输入当前的秒钟高位；
8、second_low：输入当前的秒钟低位；
输出接口信号定义：
1、VSYNC：垂直扫描信号；
2、HSYNC：水平扫描信号；
功能框图如下图所示：



部分数字显示代码（由于篇幅限制，以小时低位为例）：

```
begin
                   if(hour_low==4'd0)//小时低位为0
                   begin
                        if(((hsync_cnt >= (`HSYNC_B+16'd175+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd175+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                        ||((hsync_cnt >= (`HSYNC_B+16'd175+16'd60)) && ( hsync_cnt <=(`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                        ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                        ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
```

```
                            VGA_DATA<=`WHITE;
                        else
                            VGA_DATA<=`BLACK;
                   end
                   else if(hour_low==4'd1)//小时低位为1
                   begin
                        if((hsync_cnt >= (`HSYNC_B+16'd175+16'd35)) &&
(hsync_cnt <= (`HSYNC_B+16'd175+16'd40))&&(vsync_cnt>=(`VSYNC_P+16'd225
+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                            VGA_DATA <= `WHITE;            //显示白色
                        else
                            VGA_DATA <= `BLACK;        //显示黑色
                   end
                   else if(hour_low==4'd2)//小时低位为2
                   begin
                       if(((hsync_cnt >= (`HSYNC_B+16'd175+16'd5)) && (
hsync_cnt <= (`HSYNC_B+16'd175+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+
16'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                       ||((hsync_cnt >= (`HSYNC_B+16'd175+16'd60)) && (
hsync_cnt <=(`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+1
6'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&& (hsy
nc_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'
d15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&& (hsy
nc_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'
d67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5)) && (hs
ync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16
'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                            VGA_DATA <= `WHITE;            //显示白色
                       else
                            VGA_DATA <= `BLACK;        //显示黑色
                   end
                   else if(hour_low==4'd3)//小时低位为3
                   begin
                       if(((hsync_cnt >= (`HSYNC_B+16'd175+16'd60)) &&
(hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225
+16'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                       ||((hsync_cnt >= (`HSYNC_B+16'd175+16'd60)) && (
hsync_cnt <=(`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+1
6'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
```

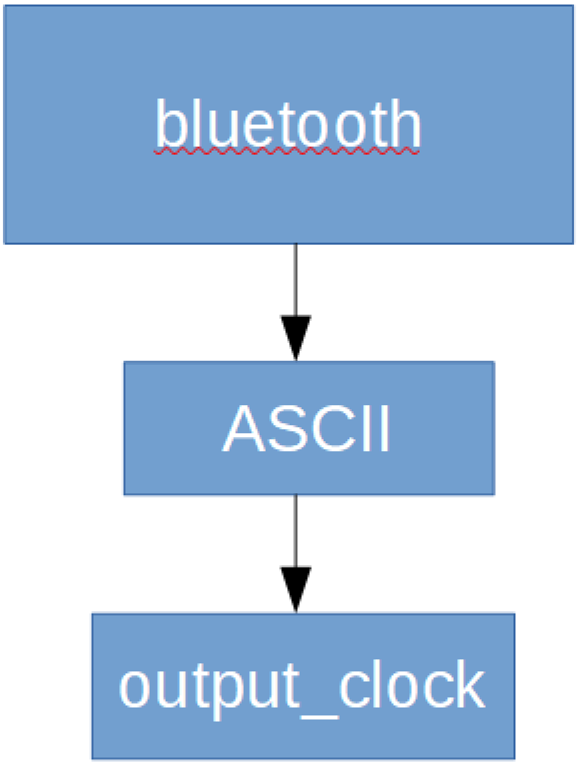
```
                       ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                            VGA_DATA <= `WHITE;            //显示白色
                       else
                            VGA_DATA <= `BLACK;        //显示黑色
                   end
                   else if(hour_low==4'd4)//小时低位为4
                   begin
                       if(((hsync_cnt >= (`HSYNC_B+16'd175+16'd60)) && (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                       ||((hsync_cnt >= (`HSYNC_B+16'd175+16'd60)) && (hsync_cnt <=(`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                       ||((hsync_cnt >= (`HSYNC_B+16'd175+16'd5)) && (hsync_cnt <=(`HSYNC_B+16'd175+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72))))
                            VGA_DATA <= `WHITE;            //显示白色
                       else
                            VGA_DATA <= `BLACK;        //显示黑色
                      end
                   else if(hour_low==4'd5)//小时低位为5
                   begin
                       if(((hsync_cnt >= (`HSYNC_B+16'd175+16'd60)) && (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                       ||((hsync_cnt >= (`HSYNC_B+16'd175+16'd5)) && (hsync_cnt <=(`HSYNC_B+16'd175+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&& (hsy
nc_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'
d67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5)) && (hs
ync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16
'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                            VGA_DATA <= `WHITE;            //显示白色
                       else
                            VGA_DATA <= `BLACK;        //显示黑色
                     end
                   else if(hour_low==4'd6)//小时低位为6
                    begin
                        if(((hsync_cnt >= (`HSYNC_B+16'd175+16'd60)) &&
 (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd22
5+16'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                        ||((hsync_cnt >= (`HSYNC_B+16'd175+16'd5)) && (
hsync_cnt <=(`HSYNC_B+16'd175+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+1
6'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                        ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&& (hs
ync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16
'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                        ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&& (hs
ync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16
'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                        ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5)) && (h
sync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-1
6'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                             VGA_DATA <= `WHITE;            //显示白色
                        else
                             VGA_DATA <= `BLACK;        //显示黑色
                      end
                   else if(hour_low==4'd7)//小时低位为7
                     begin
                          if(((hsync_cnt >= (`HSYNC_B+16'd175+16'd60))
&& (hsync_cnt <=(`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd2
25+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                         ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&& (h
sync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+1
6'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20))))
                              VGA_DATA<=`WHITE;
                          else
                              VGA_DATA<=`BLACK;
                     end
                   else if(hour_low==4'd8)//小时低位为8
```

```
                     begin
                         if(((hsync_cnt >= (`HSYNC_B+16'd175+16'd60)) &
& (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd2
25+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                         ||((hsync_cnt >= (`HSYNC_B+16'd175+16'd5)) &&
(hsync_cnt <=(`HSYNC_B+16'd175+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+
16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                         ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&& (h
sync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+1
6'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                         ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&& (h
sync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+1
6'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                         ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5)) && (
hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-
16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                              VGA_DATA <= `WHITE;            //显示白色
                         else
                              VGA_DATA <= `BLACK;        //显示黑色
                       end
                       else if(hour_low==4'd9)//小时低位为9
                        begin
                            if(((hsync_cnt >= (`HSYNC_B+16'd175+16'd60)
) && (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16
'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                            ||((hsync_cnt >= (`HSYNC_B+16'd175+16'd5))
&& (hsync_cnt <=(`HSYNC_B+16'd175+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd2
25+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                            ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&&
 (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd22
5+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                            ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&&
 (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd22
5+16'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                            ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5)) &
& (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd3
75-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                                 VGA_DATA <= `WHITE;            //显示白
色
                            else
                                 VGA_DATA <= `BLACK;        //显示黑色
                          end
                   else
                        VGA_DATA<=`BLACK;
```

```
                end
```

（三）蓝牙模块（bluetooth）：使用蓝牙外设进行闹钟的设定（整点）。

输入接口信号定义：
1、clk：输入时钟，使用 100MHz 的系统时钟作为输入；
2、rst：复位信号，高电平有效。当高电平时整个子系统全部归零；
3、get：接收蓝牙输入数据信号。
输出接口信号定义：
1、output_clock：蓝牙设定的闹钟数值（整数）。
功能框图如下图所示：



ASCII 转化整数代码：

```
always@(*)
        begin
        case(out)
            8'd48: output_clock<=5'd0;
            8'd49: output_clock<=5'd1;
            8'd50: output_clock<=5'd2;
            8'd51: output_clock<=5'd3;
            8'd52: output_clock<=5'd4;
            8'd53: output_clock<=5'd5;
            8'd54: output_clock<=5'd6;
            8'd55: output_clock<=5'd7;
            8'd56: output_clock<=5'd8;
            8'd57: output_clock<=5'd9;
            8'd65: output_clock<=5'd10;
            8'd66: output_clock<=5'd11;
            8'd67: output_clock<=5'd12;
            8'd68: output_clock<=5'd13;
            8'd69: output_clock<=5'd14;
            8'd70: output_clock<=5'd15;
            8'd71: output_clock<=5'd16;
            8'd72: output_clock<=5'd17;
            8'd73: output_clock<=5'd18;
            8'd74: output_clock<=5'd19;
            8'd75: output_clock<=5'd20;
            8'd76: output_clock<=5'd21;
            8'd77: output_clock<=5'd22;
            8'd78: output_clock<=5'd23;
            default: ;
        endcase
        end
```

（四）亮灯模块（LED_blink）：实现模拟闹钟的功能。当时间达到蓝牙设定的值，LED 开始亮一分钟。

输入接口信号定义：

1、rst：复位信号，高电平有效。当高电平时整个子系统全部归零；

2、CLK_100M：输入时钟，使用 100MHz 的系统时钟作为输入；

3、hour_high：输入当前的小时高位；

4、hour_low：输入当前的小时低位；

5、minute_high：输入当前的分钟高位；

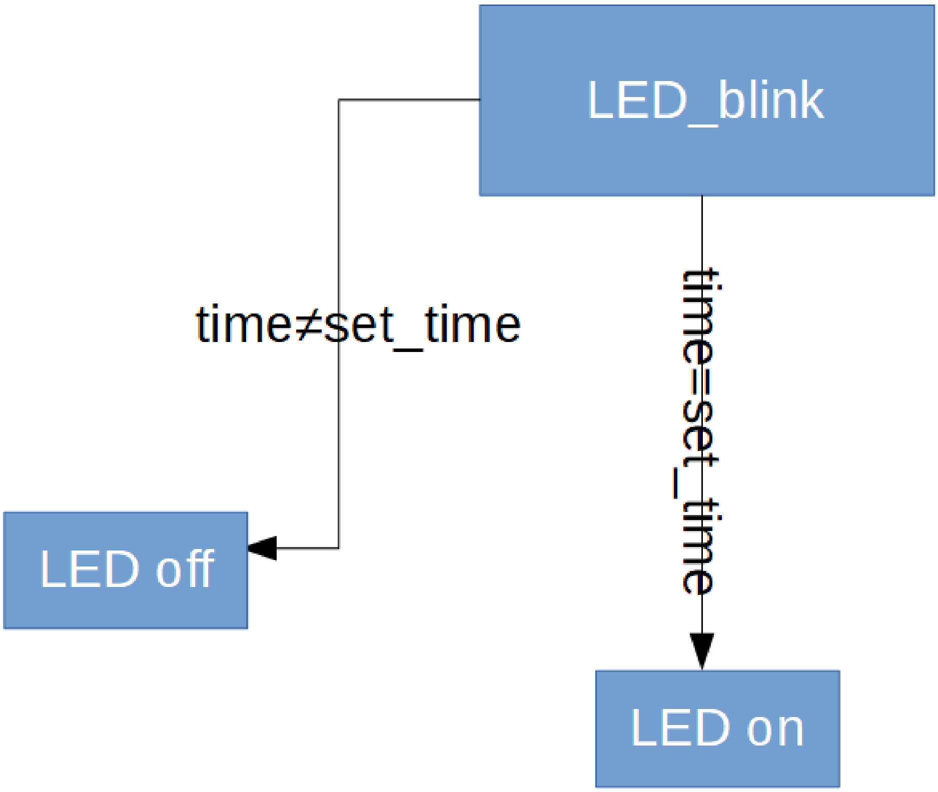
6、minute_low：输入当前的分钟低位；

7、alarm：蓝牙设定的闹钟。

输出接口信号定义：

1、blink：闹钟信号。当到达闹钟指定时间的时候变为高电平，一分钟后自动便问低电平。

功能框图如下图所示：



亮灯部分代码：

```
always@(posedge CLK_100M)
   if((hour_high*10+hour_low==alarm)&&minute_high==0&&minute_low==0)
   begin
      blink<=1;
   end
   else
   blink<=0;
```

五、测试模块建模

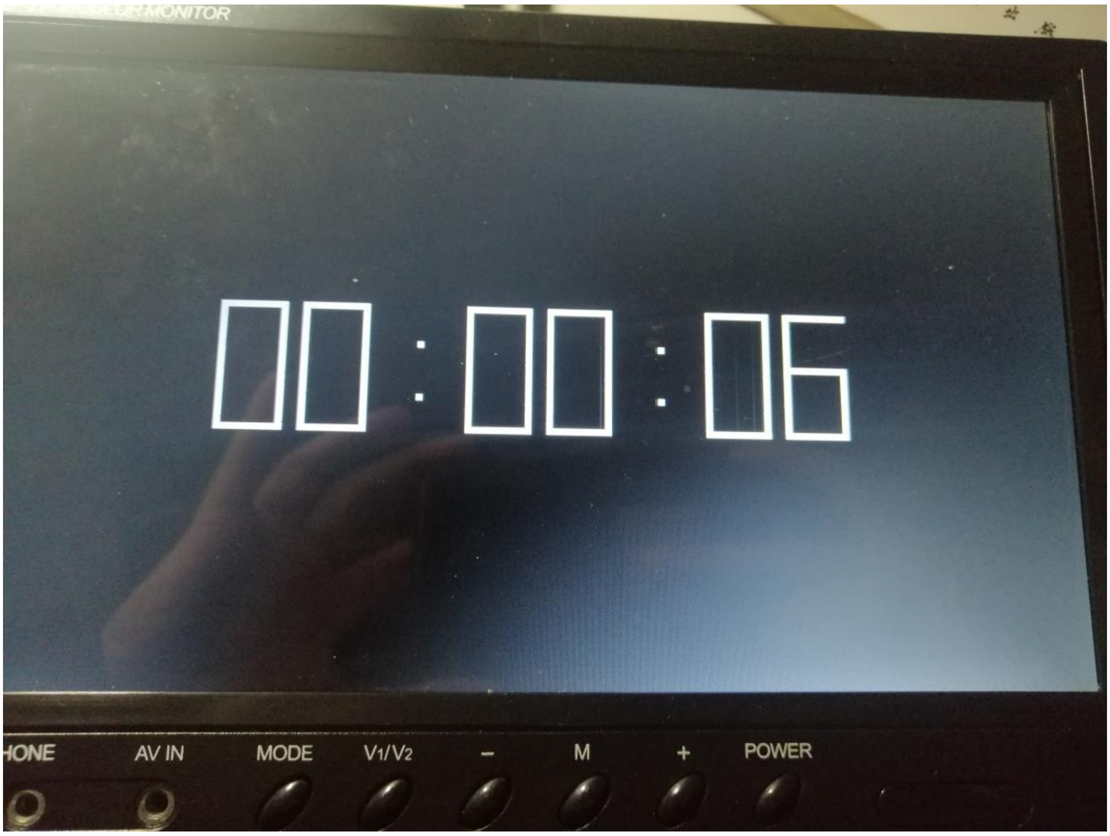
主要编写了时钟的自动计时/手动调时逻辑的验证

```
module Calculate_time_tb;
   reg clk;
   //reg ena;
   reg rst;
   reg select;
   reg time_cnt;
   reg change;

   wire [1:0] hour_high;
   wire [3:0] hour_low;
   wire [2:0] minute_high;
   wire [3:0] minute_low;
   wire [2:0] second_high;
   wire [3:0] second_low;
   wire select_time;

   calculate_time uut(clk,rst,select,time_cnt,change,second_low,second_high,minute_low,minute_high,hour_low,hour_high,select_time);
   integer i=0;
   integer j=0;
  /* initial
   begin
   ena=1;
   end*/

   initial
   begin
   rst=1;
   #20 rst=0;
   end

   initial
   begin
   clk=0;
   for(i=0;i<10000;i=i+1)
   begin
   #0.1 clk=1;
   #0.1 clk=0;
   end
   end

   initial
   begin
   change=0;
   #500 change=1;
   end

   initial
   begin
   time_cnt=0;
   #510 time_cnt=1;
   #20 time_cnt=0;
   #20 time_cnt=1;
   #20 time_cnt=0;
   #20 time_cnt=1;
   #20 time_cnt=0;
   end

   initial
   begin
   select=0;
   #520 select=1;
   #20 select=0;
   #20 select=1;
   end

endmodule
```

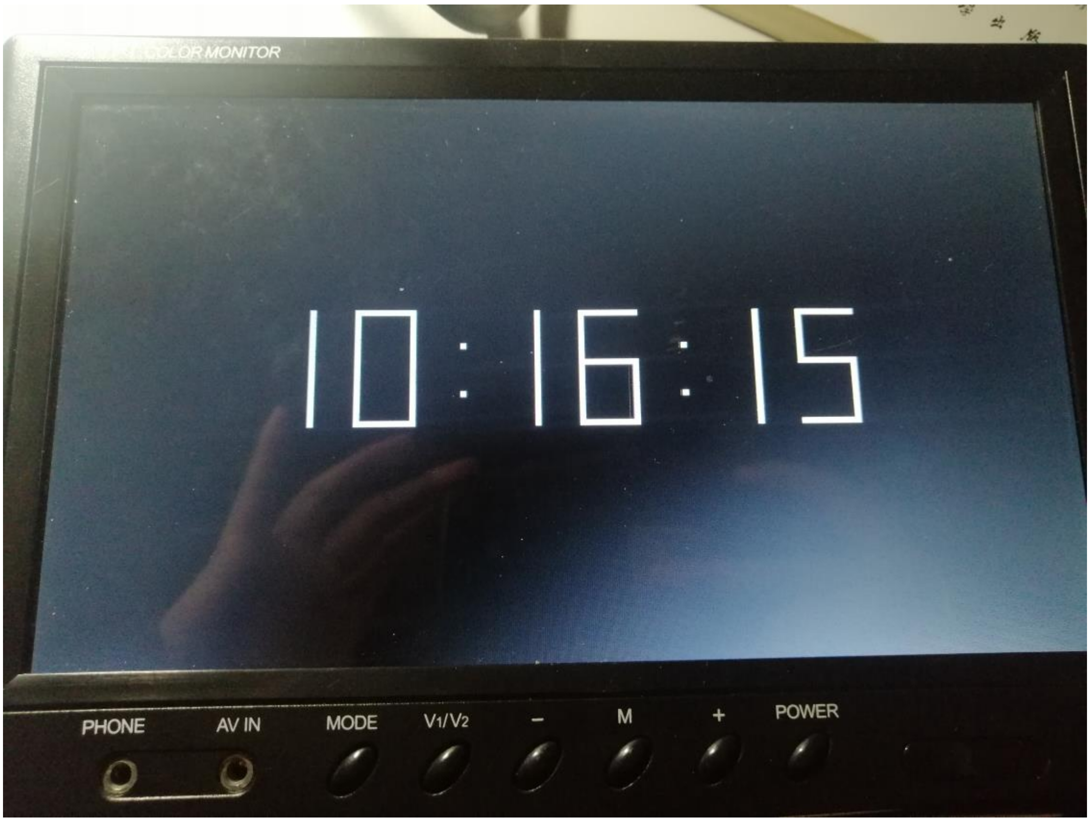
六、实验结果

（该部分可截图说明，可包含 logisim 逻辑验证图、modelsim 仿真波形图、以及下板后的实验结果贴图）

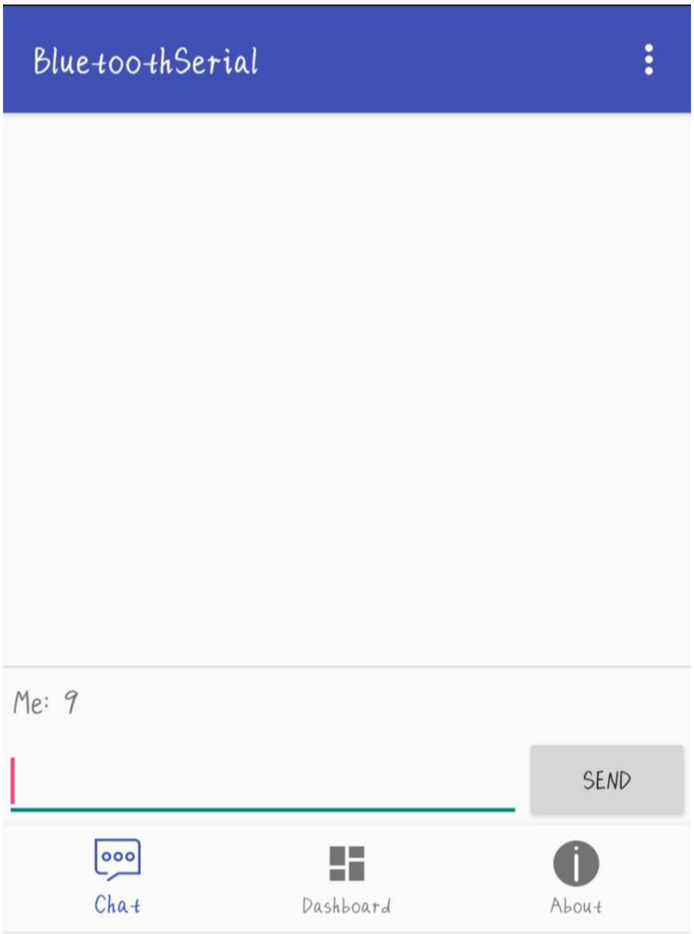
初始状态，小时、分钟和秒钟都默认为 0.当使能端输入为高电平时，开始计时。



当 change 对应为高电平时，停止计时，进入调时模式。

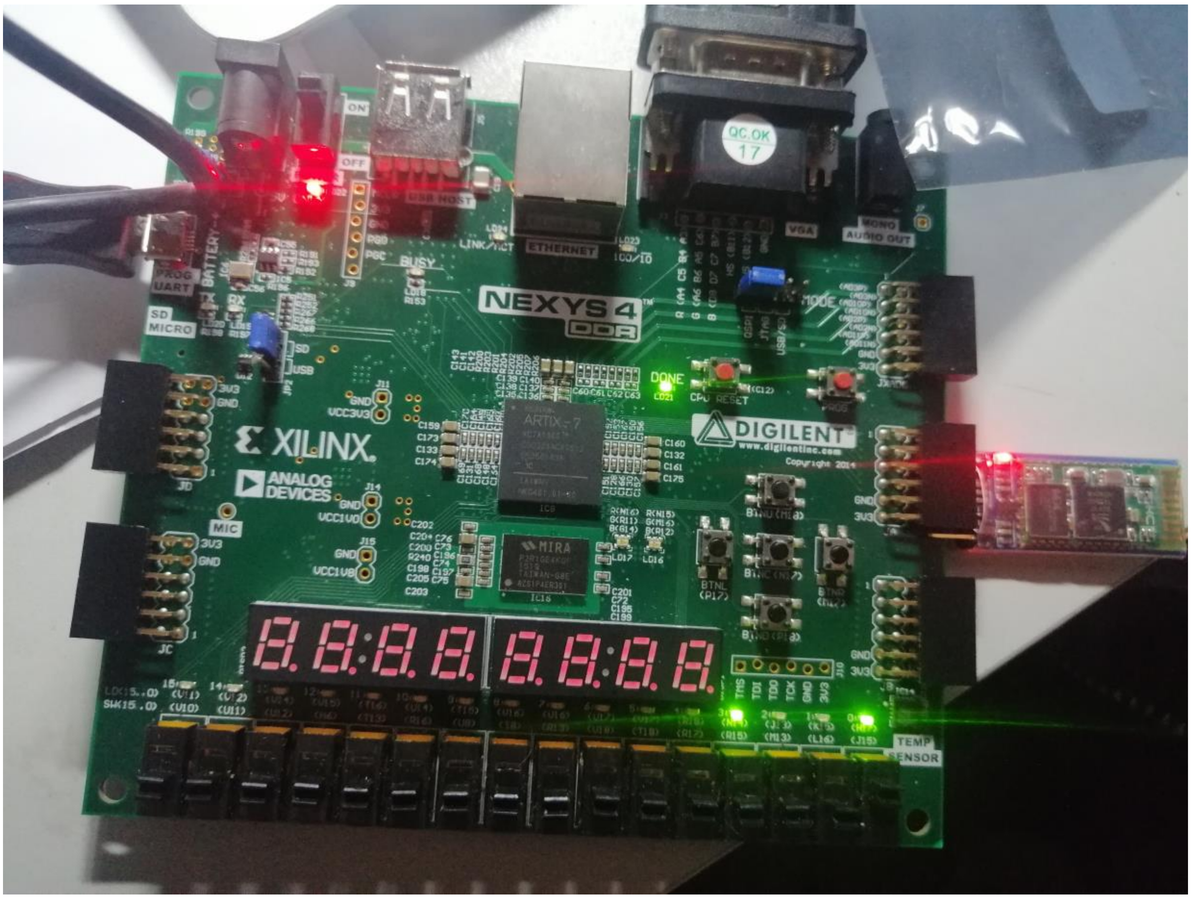


可以使用蓝牙串口助手对电子钟发送指令，比如输入 9，则订闹钟为 9 时。

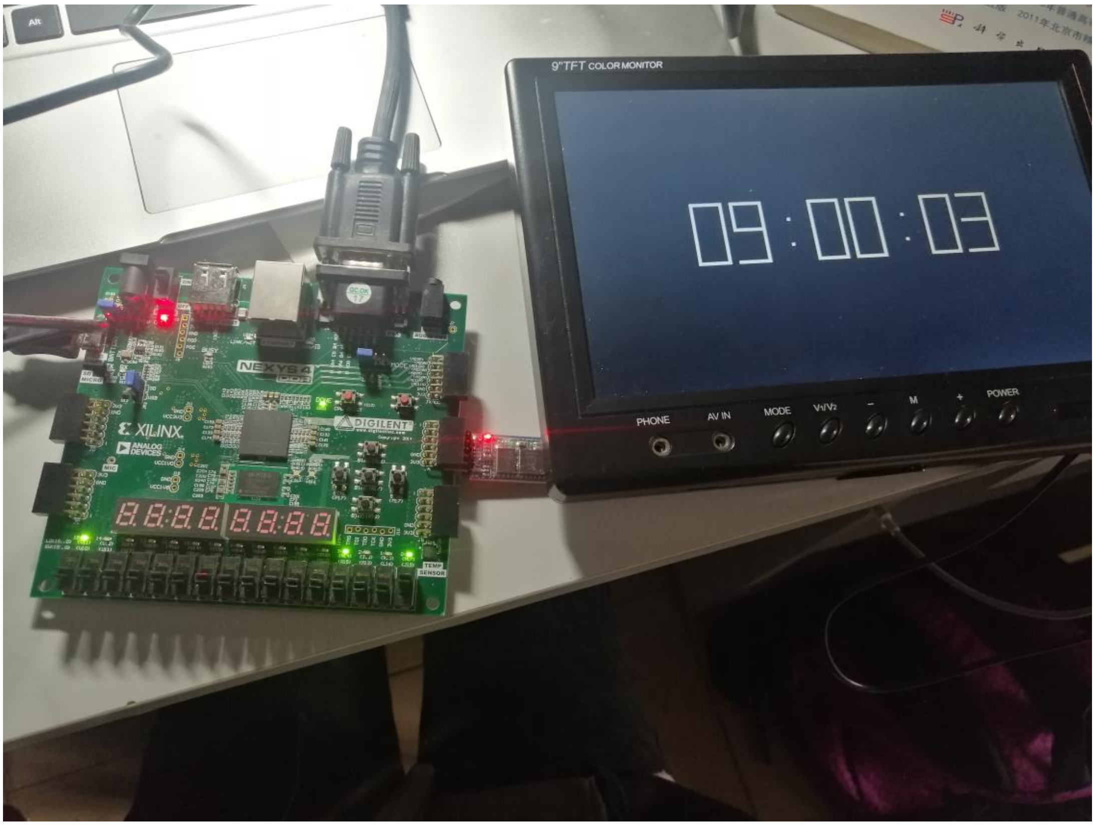


Me: 9

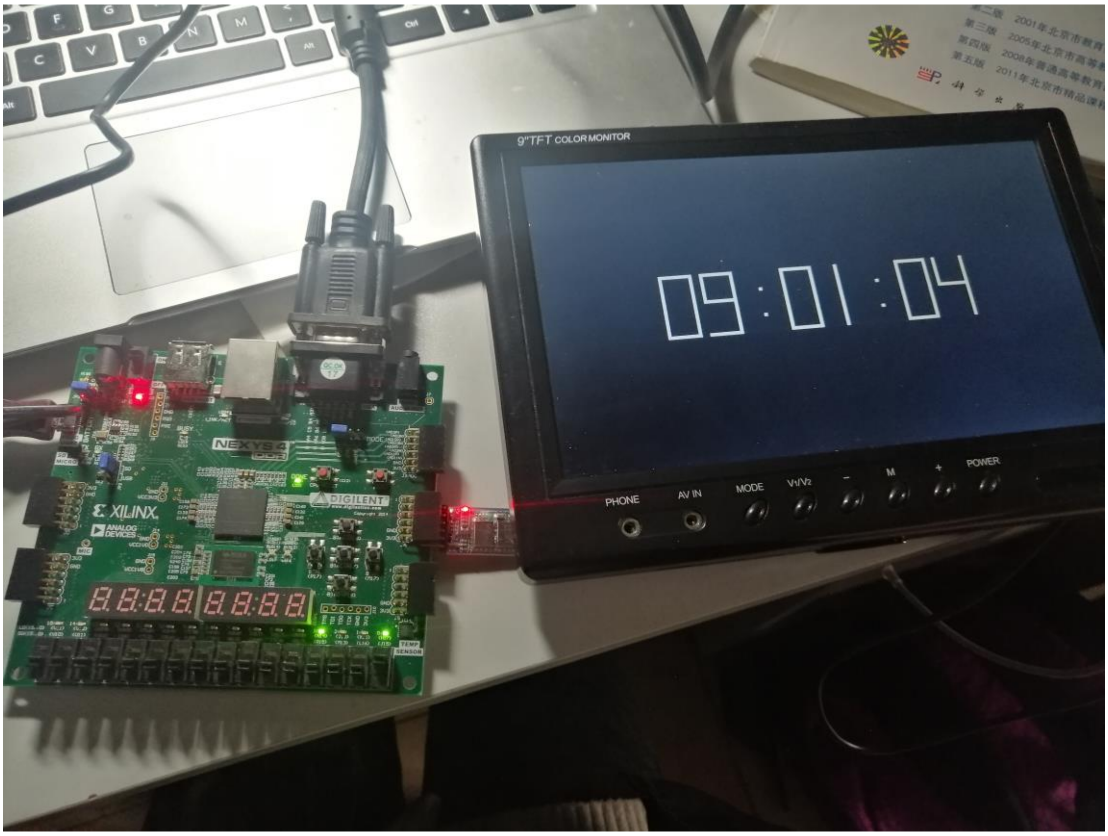
当输入 9 时输入成功后，开发板上 LED 灯显示的时 9 的二进制表示。



当电子钟上时间到达 9 点时，最左边的灯亮起。



当到 9：01 时，最左边的 LED 灯熄灭。



七、结论

经过多次下板实验，系统都表现良好，证明蓝牙和 VGA 显示器可以正常运作。这些可以说明：

1、蓝牙可以接收手机发送的信号，并通过 ASCII 码的形式传递给开发板。

2、VGA 可以在遵循行时序、列时序和场时序时对显示器的像素点进行扫描，并输出对应像素点的颜色。

八、心得体会及建议

在一开始着手做这个大作业的时候，我还是非常迷茫的。因为对这些外设不了解，以前并没有使用外设的经历，也不知道怎么把外设和我的开发板连接起来。因此，为了这项大作业，我搜查了很多资料，主要是关于外设的连接问题。在查找资料的过程中，我收获了很多，从一开始只能在显示器上显示彩条，到让显示器显示不同的数字，我在这期间有了不少进步和感悟。在连接蓝牙外设的时候，我也碰到了一些问题，通过不断调试和实践，我也终于有了很多成果。

对于数字逻辑课程来说，我主要建议是最好能完善外设的各种资料。有的外设的资料少之又少，让我在做大作业的过程中很难下手，有一些参数的设置也让我走了不少弯路，反而很多参数之类的信息我都是在网上找到的。如果能在外设的资料上提供得更到位一些的话，同学们就可以少花一些不必要的时间在搜索资料上了。

对于如今数字芯片的国内外现状，我有一些体会和感触。我国的芯片技术正在飞速发展，

但是相比一些西方国家而言依然有很大的差距。而芯片在国家科技实力方面所产生的影响越来越大，努力学好硬件是我们计算机系大学生义不容辞的责任。

九、附录代码

clock_top.v:

```
module clock_top(
//输入时钟、复位信号和使能信号
    input CLK_100M,
    input ena,
    input rst,

//时钟所需输入
    input change,
    input switch,//时、分、秒手动计时切换所需的信
    input time_count,//对于手动计时，需要对其进行加一操作
//输出VGA颜色分量
    output VSYNC,
    output HSYNC,
    output [11:0] VGA_DATA,
//蓝牙部分
    input get,
    output [4:0] output_clock,
//闹钟
    output blink
    );
    wire [1:0] hour_h;
    wire [3:0] hour_l;
    wire [2:0] minute_h;
    wire [3:0] minute_l;
    wire [2:0] second_h;
    wire [3:0] second_l;



    calculate_time calculate_inst(
                    .CLK_100M(CLK_100M),
                    .ena(ena),
                    .rst(rst),
                    .change(change),
                    .switch(switch),
                    .time_count(time_count),
                    .hour_high(hour_h),
                    .hour_low(hour_l),
                    .minute_high(minute_h),
                    .minute_low(minute_l),
                    .second_high(second_h),
                    .second_low(second_l)
                    );

      VGA_disp vga_inst(
                    .CLK_100M(CLK_100M),
                    .rst(rst),
                    .hour_high(hour_h),
                    .hour_low(hour_l),
                    .minute_high(minute_h),
                    .minute_low(minute_l),
                    .second_high(second_h),
                    .second_low(second_l),
                    .VSYNC(VSYNC),
                    .HSYNC(HSYNC),
                    .VGA_DATA(VGA_DATA)
                    );

       bluetooth blue_inst(
                    .clk(CLK_100M),
                    .rst(rst),
                    .get(get),
                    .output_clock(output_clock)
                    );

        LED_blink(
                    .CLK_100M(CLK_100M),
                    .rst(rst),
                    .hour_high(hour_h),
                    .hour_low(hour_l),
                    .minute_high(minute_h),
                    .minute_low(minute_l),
                    .alarm(output_clock),
                    .blink(blink)
                    );

endmodule
```

calculate_time.v:

```
module calculate_time(
    //输入时钟、复位信号和使能信号
    input CLK_100M,
    input ena,
    input rst,
    //输入电子钟状态变化所需要的输入信号
    input change,//手动计时
    input switch,//时、分、秒手动计时切换所需的信号
    input time_count,//对于手动计时，需要对其进行加一操作
    //输出时、分、秒的各位
    output [1:0] hour_high,
    output [3:0] hour_low,
    output [2:0] minute_high,
    output [3:0] minute_low,
    output [2:0] second_high,
    output [3:0] second_low

    );
    reg [4:0] hour;
    reg [5:0] minute;
    reg [5:0] second;
    parameter num_div=1_0000_0000;
    reg [31:0] count;
    reg CLK_1;

    reg [4:0] hour_hand;
    reg [5:0] minute_hand;
    reg [5:0] second_hand;

    always@(posedge CLK_100M)
    if(rst)
    begin
        count<=0;
        CLK_1<=0;
    end
    else if(count==num_div/2-1)
        begin
        count<=0;
        CLK_1<=~CLK_1;
        end
    else
        begin
        count<=count+1;
        end


    always@(posedge CLK_1 or posedge rst)
```

```
    if(ena)
    begin
        if(rst)
            begin
                hour<=0;
                minute<=0;
                second<=0;
            end
        else if(change==0)
        begin
            if(second==6'd59)
            begin
                second<=0;
                minute<=minute+1;
                if(minute==6'd59)
                begin
                    minute<=0;
                    hour<=hour+1;
                    if(hour==5'd24)
                    begin
                        hour<=0;
                    end
                    else
                    ;
                end
                else
                ;
            end
            else
            second<=second+1;
        end
        else if(change==1)
        begin
        hour<=hour_hand;
        minute<=minute_hand;
        second<=second_hand;
        end
        end
    else
        begin
        hour<=0;
        minute<=0;
        second<=0;
        end
```

```
reg [2:0] switch_reg;

always@(posedge switch or posedge rst)
if(rst)
    switch_reg<=0;
else
begin
    if(switch_reg==3'd3)
        switch_reg<=0;
    else
        switch_reg<=switch_reg+1;
end

always@(posedge time_count or posedge rst)
if(rst)
begin
    hour_hand<=0;
    minute_hand<=0;
    second_hand<=0;
end
else if(change==1)
begin
    if(switch_reg==0)
    begin
        if(hour_hand==5'd23)
            hour_hand<=0;
        else
            hour_hand<=hour_hand+1;
    end
    else if(switch_reg==3'd1)
    begin
        if(minute_hand==6'd59)
            minute_hand<=0;
        else
            minute_hand<=minute_hand+1;
    end
    else if(switch_reg==3'd2)
    begin
        if(second_hand==6'd59)
            second_hand<=0;
        else
            second_hand<=second_hand+1;
    end
    end
    else
    ;



    assign hour_high=change? hour_hand/5'd10:hour/5'd10;
    assign hour_low=change? hour_hand%5'd10:hour%5'd10;
    assign minute_high=change? minute_hand/6'd10: minute/6'd10;
    assign minute_low=change? minute_hand%6'd10:minute%6'd10;
    assign second_high=change? second_hand/6'd10:second/6'd10;
    assign second_low=change? second_hand%6'd10:second%6'd10;


endmodule
```

VGA_disp.v:

```
//行时序宏定义
`define HSYNC_A 16'd128
`define HSYNC_B 16'd216
`define HSYNC_C 16'd1016
`define HSYNC_D 16'd1056

//列时序宏定义
`define VSYNC_O 16'd4
`define VSYNC_P 16'd27
`define VSYNC_Q 16'd627
`define VSYNC_R 16'd628

//颜色定义
`define WHITE 12'Hfff
`define BLACK 12'H000

module VGA_disp(
    //输入：使能信号、复位信号和时钟
    input rst,//复位信号，高电平有效
    input CLK_100M,
    input [1:0] hour_high,
    input [3:0] hour_low,
    input [2:0] minute_high,
    input [3:0] minute_low,
    input [2:0] second_high,
    input [3:0] second_low,
    //输出：VGA颜色分量
    output reg VSYNC,
    output reg HSYNC,
    output reg [11:0] VGA_DATA
    );

    reg[15:0] hsync_cnt;        //水平扫描计数器
    reg[15:0] vsync_cnt;        //垂直扫描计数器
    reg CLK_50M;
    reg vga_data_valid;            //RGB数据信号有效区使能信号
    //水平扫描(扫描1056个点)
    always@(posedge(CLK_100M))
        begin
            CLK_50M <= ~CLK_50M;
        end
    always@(posedge CLK_50M or posedge rst)
    begin
        if(rst)
            hsync_cnt <= 16'd0;
        else if(hsync_cnt == `HSYNC_D)
            hsync_cnt <= 16'd0;
        else
            hsync_cnt <= hsync_cnt + 16'd1;
    end


    //垂直扫描(扫描628个点)
    always@(posedge CLK_50M or posedge rst)
    begin
        if(rst)
            vsync_cnt <= 16'd0;
        else if((vsync_cnt == `VSYNC_R) && (hsync_cnt == `HSYNC_D))
            vsync_cnt <= 16'd0;
        else if(hsync_cnt == `HSYNC_D)
            vsync_cnt <= vsync_cnt + 16'd1;
        else
            vsync_cnt <= vsync_cnt;
    end

    //行时序
    always@(posedge CLK_50M or posedge rst)
    begin
        if(rst)
```

```
            HSYNC <= 1'b0;
        else if(hsync_cnt < `HSYNC_A)    //a 域为0
            HSYNC <= 1'b0;
        else
            HSYNC <= 1'b1;                //其他域为1
    end

    //列时序
    always@(posedge CLK_50M or posedge rst)
    begin
        if(rst)
            VSYNC <= 1'b0;
        else if(vsync_cnt < `VSYNC_O)    //o 域为0
            VSYNC <= 1'b0;
        else
            VSYNC <= 1'b1;                //其他域为1
    end

    //提取显示有效区(q 域+c 域)
    always@(posedge CLK_50M or posedge rst)
    begin
        if(rst)
            vga_data_valid <= 1'b0;
        else if((hsync_cnt > `HSYNC_B && hsync_cnt < `HSYNC_C) && (vsyn
c_cnt >  `VSYNC_P && vsync_cnt < `VSYNC_Q))    //数据有效区
            vga_data_valid <= 1'b1;
        else
            vga_data_valid <= 1'b0;
    end

    always@(*)
    begin
        if(vga_data_valid)
        begin
            if(vsync_cnt >`VSYNC_P)//显示区
            begin
                if((hsync_cnt>(`HSYNC_B+16'd100))&&(hsync_cnt<(`HSYNC_B
+16'd175)))//小时高位显示
                begin
                    if(hour_high==2'd0)//小时高位为0
                    begin
                        if(((hsync_cnt >= (`HSYNC_B+16'd100+16'd5)) &&
(hsync_cnt <= (`HSYNC_B+16'd100+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225
+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
```

```
                        ||((hsync_cnt >= (`HSYNC_B+16'd100+16'd60)) && (hsync_cnt <=(`HSYNC_B+16'd100+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                        ||((hsync_cnt >=(`HSYNC_B+16'd100+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd100+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                        ||((hsync_cnt >=(`HSYNC_B+16'd100+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd100+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                            VGA_DATA<=`WHITE;
                        else
                            VGA_DATA<=`BLACK;
                    end
                    else if(hour_high==2'd1)//小时高位为1
                    begin
                        if((hsync_cnt >= (`HSYNC_B+16'd100+16'd35)) && (hsync_cnt <= (`HSYNC_B+16'd100+16'd40))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                            VGA_DATA <= `WHITE;            //显示白色
                        else
                            VGA_DATA <= `BLACK;        //显示黑色
                    end
                    else if(hour_high==2'd2)//小时高位为2
                    begin
                        if(((hsync_cnt >= (`HSYNC_B+16'd100+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd100+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                        ||((hsync_cnt >= (`HSYNC_B+16'd100+16'd60)) && (hsync_cnt <=(`HSYNC_B+16'd100+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                        ||((hsync_cnt >=(`HSYNC_B+16'd100+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd100+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                        ||((hsync_cnt >=(`HSYNC_B+16'd100+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd100+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                        ||((hsync_cnt >=(`HSYNC_B+16'd100+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd100+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                             VGA_DATA <= `WHITE;            //显示白色
                        else
                             VGA_DATA <= `BLACK;        //显示黑色
                    end
                    else
```

```
                        VGA_DATA<=`BLACK;
                end
                else if((hsync_cnt>(`HSYNC_B+16'd175))&&(hsync_cnt<(`HSYNC_B+16'd250)))//小时低位显示
                begin
                   if(hour_low==4'd0)//小时低位为0
                   begin
                        if(((hsync_cnt >= (`HSYNC_B+16'd175+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd175+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                       ||((hsync_cnt >= (`HSYNC_B+16'd175+16'd60)) && (hsync_cnt <=(`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                            VGA_DATA<=`WHITE;
                        else
                            VGA_DATA<=`BLACK;
                   end
                   else if(hour_low==4'd1)//小时低位为1
                   begin
                        if((hsync_cnt >= (`HSYNC_B+16'd175+16'd35)) && (hsync_cnt <= (`HSYNC_B+16'd175+16'd40))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                            VGA_DATA <= `WHITE;            //显示白色
                        else
                            VGA_DATA <= `BLACK;        //显示黑色
                   end
                   else if(hour_low==4'd2)//小时低位为2
                   begin
                       if(((hsync_cnt >= (`HSYNC_B+16'd175+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd175+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                       ||((hsync_cnt >= (`HSYNC_B+16'd175+16'd60)) && (hsync_cnt <=(`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                            VGA_DATA <= `WHITE;            //显示白色
                       else
                            VGA_DATA <= `BLACK;        //显示黑色
                   end
                   else if(hour_low==4'd3)//小时低位为3
                   begin
                       if(((hsync_cnt >= (`HSYNC_B+16'd175+16'd60)) && (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                       ||((hsync_cnt >= (`HSYNC_B+16'd175+16'd60)) && (hsync_cnt <=(`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                            VGA_DATA <= `WHITE;            //显示白色
                       else
                            VGA_DATA <= `BLACK;        //显示黑色
                   end
                   else if(hour_low==4'd4)//小时低位为4
                   begin
                       if(((hsync_cnt >= (`HSYNC_B+16'd175+16'd60)) && (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                       ||((hsync_cnt >= (`HSYNC_B+16'd175+16'd60)) && (hsync_cnt <=(`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                       ||((hsync_cnt >= (`HSYNC_B+16'd175+16'd5)) && (hsync_cnt <=(`HSYNC_B+16'd175+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72))))
                            VGA_DATA <= `WHITE;            //显示白色
                       else
                            VGA_DATA <= `BLACK;        //显示黑色
                      end
                   else if(hour_low==4'd5)//小时低位为5
                   begin
                       if(((hsync_cnt >= (`HSYNC_B+16'd175+16'd60)) && (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                       ||((hsync_cnt >= (`HSYNC_B+16'd175+16'd5)) && (hsync_cnt <=(`HSYNC_B+16'd175+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                            VGA_DATA <= `WHITE;            //显示白色
                       else
                            VGA_DATA <= `BLACK;        //显示黑色
                     end
                   else if(hour_low==4'd6)//小时低位为6
                    begin
                        if(((hsync_cnt >= (`HSYNC_B+16'd175+16'd60)) && (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                        ||((hsync_cnt >= (`HSYNC_B+16'd175+16'd5)) && (hsync_cnt <=(`HSYNC_B+16'd175+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                        ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                        ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                        ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                             VGA_DATA <= `WHITE;            //显示白色
                        else
                             VGA_DATA <= `BLACK;        //显示黑色
                      end
                   else if(hour_low==4'd7)//小时低位为7
                     begin
                          if(((hsync_cnt >= (`HSYNC_B+16'd175+16'd60)) && (hsync_cnt <=(`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                         ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20))))
                              VGA_DATA<=`WHITE;
                          else
                              VGA_DATA<=`BLACK;
                     end
                   else if(hour_low==4'd8)//小时低位为8
                     begin
                         if(((hsync_cnt >= (`HSYNC_B+16'd175+16'd60)) && (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                         ||((hsync_cnt >= (`HSYNC_B+16'd175+16'd5)) && (hsync_cnt <=(`HSYNC_B+16'd175+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                         ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                         ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                         ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                              VGA_DATA <= `WHITE;            //显示白色
                         else
                              VGA_DATA <= `BLACK;        //显示黑色
                       end
                       else if(hour_low==4'd9)//小时低位为9
                        begin
```

```
                            if(((hsync_cnt >= (`HSYNC_B+16'd175+16'd60)) && (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                            ||((hsync_cnt >= (`HSYNC_B+16'd175+16'd5)) && (hsync_cnt <=(`HSYNC_B+16'd175+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                            ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                            ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                            ||((hsync_cnt >=(`HSYNC_B+16'd175+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd175+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                                 VGA_DATA <= `WHITE;            //显示白色
                            else
                                 VGA_DATA <= `BLACK;        //显示黑色
                          end
                   else
                        VGA_DATA<=`BLACK;
                end
                else if((hsync_cnt>(`HSYNC_B+16'd250))&&(hsync_cnt<(`HSYNC_B+16'd325)))//小时与分钟之间的冒号显示
                begin
                    if(((hsync_cnt >= (`HSYNC_B+16'd250+16'd35)) && (hsync_cnt <= (`HSYNC_B+16'd250+16'd40))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd50))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd55)))
                    ||((hsync_cnt >= (`HSYNC_B+16'd250+16'd35)) && (hsync_cnt <= (`HSYNC_B+16'd250+16'd40))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd100))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd105))))
                        VGA_DATA<=`WHITE;       //显示白色
                    else
                        VGA_DATA<=`BLACK;          //显示黑色
                end
                else if((hsync_cnt>(`HSYNC_B+16'd325))&&(hsync_cnt<(`HSYNC_B+16'd400)))//分钟高位显示
                begin
                   if(minute_high==3'd0)//分钟高位为0
                   begin
                        if(((hsync_cnt >= (`HSYNC_B+16'd325+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd325+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                       ||((hsync_cnt >= (`HSYNC_B+16'd325+16'd60)) && (
hsync_cnt <=(`HSYNC_B+16'd325+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+1
6'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd325+16'd5))&& (hsy
nc_cnt <= (`HSYNC_B+16'd325+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'
d15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd325+16'd5)) && (hs
ync_cnt <= (`HSYNC_B+16'd325+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16
'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                            VGA_DATA<=`WHITE;
                        else
                            VGA_DATA<=`BLACK;
                   end
                   else if(minute_high==3'd1)//分钟高位为1
                   begin
                        if((hsync_cnt >= (`HSYNC_B+16'd325+16'd35)) &&
(hsync_cnt <= (`HSYNC_B+16'd325+16'd40))&&(vsync_cnt>=(`VSYNC_P+16'd225
+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                            VGA_DATA <= `WHITE;            //红
色        1110_0000
                        else
                            VGA_DATA <= `BLACK;        //黑
色        0000_0000
                   end
                   else if(minute_high==3'd2)//分钟高位为2
                  begin
                      if(((hsync_cnt >= (`HSYNC_B+16'd325+16'd5)) && (h
sync_cnt <= (`HSYNC_B+16'd325+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+1
6'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                      ||((hsync_cnt >= (`HSYNC_B+16'd325+16'd60)) && (h
sync_cnt <=(`HSYNC_B+16'd325+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16
'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                      ||((hsync_cnt >=(`HSYNC_B+16'd325+16'd5))&& (hsyn
c_cnt <= (`HSYNC_B+16'd325+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd
15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                      ||((hsync_cnt >=(`HSYNC_B+16'd325+16'd5))&& (hsyn
c_cnt <= (`HSYNC_B+16'd325+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd
67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                      ||((hsync_cnt >=(`HSYNC_B+16'd325+16'd5)) && (hsy
nc_cnt <= (`HSYNC_B+16'd325+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16'
d20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                           VGA_DATA <= `WHITE;            //显示白色
                      else
                           VGA_DATA <= `BLACK;        //显示黑色
                  end
                  else if(minute_high==3'd3)//分钟高位为3
                  begin
                      if(((hsync_cnt >= (`HSYNC_B+16'd325+16'd60)) && (hsync_cnt <= (`HSYNC_B+16'd325+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                      ||((hsync_cnt >= (`HSYNC_B+16'd325+16'd60)) && (hsync_cnt <=(`HSYNC_B+16'd325+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                      ||((hsync_cnt >=(`HSYNC_B+16'd325+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd325+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                      ||((hsync_cnt >=(`HSYNC_B+16'd325+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd325+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                      ||((hsync_cnt >=(`HSYNC_B+16'd325+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd325+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                           VGA_DATA <= `WHITE;            //显示白色
                      else
                           VGA_DATA <= `BLACK;        //显示黑色
                  end
                  else if(minute_high==3'd4)//分钟高位为4
                  begin
                      if(((hsync_cnt >= (`HSYNC_B+16'd325+16'd60)) && (hsync_cnt <= (`HSYNC_B+16'd325+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                      ||((hsync_cnt >= (`HSYNC_B+16'd325+16'd60)) && (hsync_cnt <=(`HSYNC_B+16'd325+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                      ||((hsync_cnt >= (`HSYNC_B+16'd325+16'd5)) && (hsync_cnt <=(`HSYNC_B+16'd325+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                      ||((hsync_cnt >=(`HSYNC_B+16'd325+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd325+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72))))
                           VGA_DATA <= `WHITE;            //显示白色
                      else
                           VGA_DATA <= `BLACK;        //显示黑色
                     end
                  else if(minute_high==4'd5)//分钟高位为5
                  begin
```

```
                      if(((hsync_cnt >= (`HSYNC_B+16'd325+16'd60)) && (hsync_cnt <= (`HSYNC_B+16'd325+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                      ||((hsync_cnt >= (`HSYNC_B+16'd325+16'd5)) && (hsync_cnt <=(`HSYNC_B+16'd325+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                      ||((hsync_cnt >=(`HSYNC_B+16'd325+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd325+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                      ||((hsync_cnt >=(`HSYNC_B+16'd325+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd325+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                      ||((hsync_cnt >=(`HSYNC_B+16'd325+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd325+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                           VGA_DATA <= `WHITE;            //显示白色
                      else
                           VGA_DATA <= `BLACK;        //显示黑色
                    end
                   else
                        VGA_DATA<=`BLACK;
                end
                else if((hsync_cnt>(`HSYNC_B+16'd400))&&(hsync_cnt<(`HSYNC_B+16'd475)))//分钟低位显示
                begin
                   if(minute_low==0)//分钟低位为0
                   begin
                        if(((hsync_cnt >= (`HSYNC_B+16'd400+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd400+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                       ||((hsync_cnt >= (`HSYNC_B+16'd400+16'd60)) && (hsync_cnt <=(`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd400+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd400+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                            VGA_DATA<=`WHITE;
                        else
                            VGA_DATA<=`BLACK;
                   end
                   else if(minute_low==4'd1)//分钟低位为1
                   begin
                        if((hsync_cnt >= (`HSYNC_B+16'd400+16'd35)) &&
(hsync_cnt <= (`HSYNC_B+16'd400+16'd40))&&(vsync_cnt>=(`VSYNC_P+16'd225
+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                            VGA_DATA <= `WHITE;           //红
色         1110_0000
                        else
                            VGA_DATA <= `BLACK;       //黑
色         0000_0000
                   end
                   else if(minute_low==4'd2)//分钟低位为2
                      begin
                          if(((hsync_cnt >= (`HSYNC_B+16'd400+16'd5)) &
& (hsync_cnt <= (`HSYNC_B+16'd400+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd2
25+16'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                          ||((hsync_cnt >= (`HSYNC_B+16'd400+16'd60)) &
& (hsync_cnt <=(`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd22
5+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                          ||((hsync_cnt >=(`HSYNC_B+16'd400+16'd5))&& (
hsync_cnt <= (`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+
16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                          ||((hsync_cnt >=(`HSYNC_B+16'd400+16'd5))&& (
hsync_cnt <= (`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+
16'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                          ||((hsync_cnt >=(`HSYNC_B+16'd400+16'd5)) &&
(hsync_cnt <= (`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375
-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                               VGA_DATA <= `WHITE;           //显示白色
                          else
                               VGA_DATA <= `BLACK;       //显示黑色
                      end
                      else if(minute_low==4'd3)//分钟低位为3
                      begin
                          if(((hsync_cnt >= (`HSYNC_B+16'd400+16'd60))
&& (hsync_cnt <= (`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd
225+16'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                          ||((hsync_cnt >= (`HSYNC_B+16'd400+16'd60)) &
& (hsync_cnt <=(`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd22
5+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                          ||((hsync_cnt >=(`HSYNC_B+16'd400+16'd5))&& (
hsync_cnt <= (`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+
16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
```

```
                          ||((hsync_cnt >=(`HSYNC_B+16'd400+16'd5))&& (
hsync_cnt <= (`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+
16'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                          ||((hsync_cnt >=(`HSYNC_B+16'd400+16'd5)) &&
(hsync_cnt <= (`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375
-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                               VGA_DATA <= `WHITE;             //显示白色
                          else
                               VGA_DATA <= `BLACK;         //显示黑色
                      end
                      else if(minute_low==4'd4)//分钟低位为4
                      begin
                          if(((hsync_cnt >= (`HSYNC_B+16'd400+16'd60))
&& (hsync_cnt <= (`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd
225+16'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                          ||((hsync_cnt >= (`HSYNC_B+16'd400+16'd60)) &
& (hsync_cnt <=(`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd22
5+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                          ||((hsync_cnt >= (`HSYNC_B+16'd400+16'd5)) &&
 (hsync_cnt <=(`HSYNC_B+16'd400+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225
+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                          ||((hsync_cnt >=(`HSYNC_B+16'd400+16'd5))&& (
hsync_cnt <= (`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+
16'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72))))
                               VGA_DATA <= `WHITE;             //显示白色
                          else
                               VGA_DATA <= `BLACK;         //显示黑色
                         end
                      else if(minute_low==4'd5)//分钟低位为5
                      begin
                          if(((hsync_cnt >= (`HSYNC_B+16'd400+16'd60))
&& (hsync_cnt <= (`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd
225+16'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                          ||((hsync_cnt >= (`HSYNC_B+16'd400+16'd5)) &&
 (hsync_cnt <=(`HSYNC_B+16'd400+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225
+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                          ||((hsync_cnt >=(`HSYNC_B+16'd400+16'd5))&& (
hsync_cnt <= (`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+
16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                          ||((hsync_cnt >=(`HSYNC_B+16'd400+16'd5))&& (
hsync_cnt <= (`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+
16'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
```

```
                          ||((hsync_cnt >=(`HSYNC_B+16'd400+16'd5)) &&
(hsync_cnt <= (`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375
-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                               VGA_DATA <= `WHITE;            //显示白色
                          else
                               VGA_DATA <= `BLACK;        //显示黑色
                        end
                      else if(minute_low==4'd6)//分钟低位为6
                       begin
                           if(((hsync_cnt >= (`HSYNC_B+16'd400+16'd60))
 && (hsync_cnt <= (`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'
d225+16'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                           ||((hsync_cnt >= (`HSYNC_B+16'd400+16'd5)) &
& (hsync_cnt <=(`HSYNC_B+16'd400+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd22
5+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                           ||((hsync_cnt >=(`HSYNC_B+16'd400+16'd5))&&
(hsync_cnt <= (`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225
+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                           ||((hsync_cnt >=(`HSYNC_B+16'd400+16'd5))&&
(hsync_cnt <= (`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225
+16'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                           ||((hsync_cnt >=(`HSYNC_B+16'd400+16'd5)) &&
 (hsync_cnt <= (`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd37
5-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                                VGA_DATA <= `WHITE;            //显示白
色
                           else
                                VGA_DATA <= `BLACK;        //显示黑色
                         end
                      else if(minute_low==4'd7)//分钟低位为7
                        begin
                             if(((hsync_cnt >= (`HSYNC_B+16'd400+16'd60
)) && (hsync_cnt <=(`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16
'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                            ||((hsync_cnt >=(`HSYNC_B+16'd400+16'd5))&&
 (hsync_cnt <= (`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd22
5+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20))))
                                 VGA_DATA<=`WHITE;
                             else
                                 VGA_DATA<=`BLACK;
                        end
                      else if(minute_low==4'd8)//分钟低位为8
                        begin
```

```
              if(((hsync_cnt >= (`HSYNC_B+16'd400+16'd60)
) && (hsync_cnt <= (`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16
'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
              ||((hsync_cnt >= (`HSYNC_B+16'd400+16'd5))
&& (hsync_cnt <=(`HSYNC_B+16'd400+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd2
25+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
              ||((hsync_cnt >=(`HSYNC_B+16'd400+16'd5))&&
 (hsync_cnt <= (`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd22
5+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
              ||((hsync_cnt >=(`HSYNC_B+16'd400+16'd5))&&
 (hsync_cnt <= (`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd22
5+16'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
              ||((hsync_cnt >=(`HSYNC_B+16'd400+16'd5)) &
& (hsync_cnt <= (`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd3
75-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                VGA_DATA <= `WHITE;           //显示白
色
              else
                VGA_DATA <= `BLACK;         //显示黑色
             end
             else if(minute_low==4'd9)//分钟低位为9
              begin
                if(((hsync_cnt >= (`HSYNC_B+16'd400+16'd
60)) && (hsync_cnt <= (`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P
+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                ||((hsync_cnt >= (`HSYNC_B+16'd400+16'd5
)) && (hsync_cnt <=(`HSYNC_B+16'd400+16'd10))&&(vsync_cnt>=(`VSYNC_P+16
'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                ||((hsync_cnt >=(`HSYNC_B+16'd400+16'd5)
)&& (hsync_cnt <= (`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'
d225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                ||((hsync_cnt >=(`HSYNC_B+16'd400+16'd5)
)&& (hsync_cnt <= (`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'
d225+16'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                ||((hsync_cnt >=(`HSYNC_B+16'd400+16'd5)
) && (hsync_cnt <= (`HSYNC_B+16'd400+16'd65))&&(vsync_cnt>=(`VSYNC_P+16
'd375-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                  VGA_DATA <= `WHITE;           //显
示白色
                else
                  VGA_DATA <= `BLACK;         //显示黑
色
               end
         else
```

```
            VGA_DATA<=`BLACK;
        end
        else if((hsync_cnt>(`HSYNC_B+16'd475))&&(hsync_cnt<(`HSYNC_B+16'd550)))//分钟和秒钟之间的冒号
        begin
          if(((hsync_cnt >= (`HSYNC_B+16'd475+16'd35)) && (hsync_cnt <= (`HSYNC_B+16'd475+16'd40))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd50))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd55)))
          ||((hsync_cnt >= (`HSYNC_B+16'd475+16'd35)) && (hsync_cnt <= (`HSYNC_B+16'd475+16'd40))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd100))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd105))))
            VGA_DATA<=`WHITE;
          else
            VGA_DATA<=`BLACK;
        end
        else if((hsync_cnt>(`HSYNC_B+16'd550))&&(hsync_cnt<(`HSYNC_B+16'd625)))//秒钟高位显示
        begin
         if(second_high==0)//秒钟高位为0
         begin
            if(((hsync_cnt >= (`HSYNC_B+16'd550+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd550+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
            ||((hsync_cnt >= (`HSYNC_B+16'd550+16'd60)) && (hsync_cnt <=(`HSYNC_B+16'd550+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
            ||((hsync_cnt >=(`HSYNC_B+16'd550+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd550+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
            ||((hsync_cnt >=(`HSYNC_B+16'd550+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd550+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
              VGA_DATA<=`WHITE;
            else
              VGA_DATA<=`BLACK;
         end
         else if(second_high==3'd1)//秒钟高位为1
         begin
            if((hsync_cnt >= (`HSYNC_B+16'd550+16'd35)) && (hsync_cnt <= (`HSYNC_B+16'd550+16'd40))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
              VGA_DATA <= `WHITE;            //红色         1110_0000
            else
```

```
                            VGA_DATA <= `BLACK;        //黑色         0000_0000
                   end
                   else if(second_high==3'd2)//秒钟高位为2
                    begin
                     if(((hsync_cnt >= (`HSYNC_B+16'd550+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd550+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                     ||((hsync_cnt >= (`HSYNC_B+16'd550+16'd60)) && (hsync_cnt <=(`HSYNC_B+16'd550+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                     ||((hsync_cnt >=(`HSYNC_B+16'd550+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd550+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                     ||((hsync_cnt >=(`HSYNC_B+16'd550+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd550+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                     ||((hsync_cnt >=(`HSYNC_B+16'd550+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd550+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                          VGA_DATA <= `WHITE;           //显示白色
                     else
                          VGA_DATA <= `BLACK;         //显示黑色
                 end
                 else if(second_high==3'd3)//秒钟高位为3
                 begin
                     if(((hsync_cnt >= (`HSYNC_B+16'd550+16'd60)) && (hsync_cnt <= (`HSYNC_B+16'd550+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                     ||((hsync_cnt >= (`HSYNC_B+16'd550+16'd60)) && (hsync_cnt <=(`HSYNC_B+16'd550+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                     ||((hsync_cnt >=(`HSYNC_B+16'd550+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd550+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                     ||((hsync_cnt >=(`HSYNC_B+16'd550+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd550+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                     ||((hsync_cnt >=(`HSYNC_B+16'd550+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd550+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                          VGA_DATA <= `WHITE;           //显示白色
                     else
                          VGA_DATA <= `BLACK;         //显示黑色
                 end
                 else if(second_high==3'd4)//秒钟高位为4
                 begin
                     if(((hsync_cnt >= (`HSYNC_B+16'd550+16'd60)) && (h
sync_cnt <= (`HSYNC_B+16'd550+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+1
6'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                     ||((hsync_cnt >= (`HSYNC_B+16'd550+16'd60)) && (hs
ync_cnt <=(`HSYNC_B+16'd550+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'
d15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                     ||((hsync_cnt >= (`HSYNC_B+16'd550+16'd5)) && (hsy
nc_cnt <=(`HSYNC_B+16'd550+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd
15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                     ||((hsync_cnt >=(`HSYNC_B+16'd550+16'd5))&& (hsync
_cnt <= (`HSYNC_B+16'd550+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd6
7))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72))))
                          VGA_DATA <= `WHITE;            //显示白色
                     else
                          VGA_DATA <= `BLACK;        //显示黑色
                    end
                 else if(second_high==4'd5)//秒钟高位为5
                 begin
                     if(((hsync_cnt >= (`HSYNC_B+16'd550+16'd60)) && (h
sync_cnt <= (`HSYNC_B+16'd550+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+1
6'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                     ||((hsync_cnt >= (`HSYNC_B+16'd550+16'd5)) && (hsy
nc_cnt <=(`HSYNC_B+16'd550+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd
15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                     ||((hsync_cnt >=(`HSYNC_B+16'd550+16'd5))&& (hsync
_cnt <= (`HSYNC_B+16'd550+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd1
5))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                     ||((hsync_cnt >=(`HSYNC_B+16'd550+16'd5))&& (hsync
_cnt <= (`HSYNC_B+16'd550+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd6
7))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                     ||((hsync_cnt >=(`HSYNC_B+16'd550+16'd5)) && (hsyn
c_cnt <= (`HSYNC_B+16'd550+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16'd
20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                          VGA_DATA <= `WHITE;            //显示白色
                     else
                          VGA_DATA <= `BLACK;        //显示黑色
                   end
                   else
                        VGA_DATA<=`BLACK;

                end
                else if((hsync_cnt>(`HSYNC_B+16'd625))&&(hsync_cnt<(`HS
YNC_B+16'd700)))//秒钟低位显示
                begin
                   if(second_low==0)
                   begin
                        if(((hsync_cnt >= (`HSYNC_B+16'd625+16'd5)) &&
(hsync_cnt <= (`HSYNC_B+16'd625+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225
+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                       ||((hsync_cnt >= (`HSYNC_B+16'd625+16'd60)) && (
hsync_cnt <=(`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+1
6'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd625+16'd5))&& (hsy
nc_cnt <= (`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'
d15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd625+16'd5)) && (hs
ync_cnt <= (`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16
'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                            VGA_DATA<=`WHITE;
                        else
                            VGA_DATA<=`BLACK;
                   end
                   else if(second_low==4'd1)
                   begin
                        if((hsync_cnt >= (`HSYNC_B+16'd625+16'd35)) &&
(hsync_cnt <= (`HSYNC_B+16'd625+16'd40))&&(vsync_cnt>=(`VSYNC_P+16'd225
+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                            VGA_DATA <= `WHITE;            //红
色        1110_0000
                        else
                            VGA_DATA <= `BLACK;        //黑
色        0000_0000
                   end
                   else if(second_low==4'd2)//秒钟低位为2
                 begin
                     if(((hsync_cnt >= (`HSYNC_B+16'd625+16'd5)) && (hs
ync_cnt <= (`HSYNC_B+16'd625+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+16
'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                     ||((hsync_cnt >= (`HSYNC_B+16'd625+16'd60)) && (hs
ync_cnt <=(`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'
d15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                     ||((hsync_cnt >=(`HSYNC_B+16'd625+16'd5))&& (hsync
_cnt <= (`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd1
5))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
```

```
                     ||((hsync_cnt >=(`HSYNC_B+16'd625+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                     ||((hsync_cnt >=(`HSYNC_B+16'd625+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                          VGA_DATA <= `WHITE;            //显示白色
                     else
                          VGA_DATA <= `BLACK;        //显示黑色
                 end
                 else if(second_low==4'd3)//秒钟低位为3
                 begin
                     if(((hsync_cnt >= (`HSYNC_B+16'd625+16'd60)) && (hsync_cnt <= (`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                     ||((hsync_cnt >= (`HSYNC_B+16'd625+16'd60)) && (hsync_cnt <=(`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                     ||((hsync_cnt >=(`HSYNC_B+16'd625+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                     ||((hsync_cnt >=(`HSYNC_B+16'd625+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                     ||((hsync_cnt >=(`HSYNC_B+16'd625+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                          VGA_DATA <= `WHITE;            //显示白色
                     else
                          VGA_DATA <= `BLACK;        //显示黑色
                 end
                 else if(second_low==4'd4)//秒钟低位为4
                 begin
                     if(((hsync_cnt >= (`HSYNC_B+16'd625+16'd60)) && (hsync_cnt <= (`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                     ||((hsync_cnt >= (`HSYNC_B+16'd625+16'd60)) && (hsync_cnt <=(`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                     ||((hsync_cnt >= (`HSYNC_B+16'd625+16'd5)) && (hsync_cnt <=(`HSYNC_B+16'd625+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                     ||((hsync_cnt >=(`HSYNC_B+16'd625+16'd5))&& (hsync
_cnt <= (`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd6
7))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72))))
                          VGA_DATA <= `WHITE;            //显示白色
                     else
                          VGA_DATA <= `BLACK;        //显示黑色
                    end
                 else if(second_low==4'd5)//秒钟低位为5
                 begin
                     if(((hsync_cnt >= (`HSYNC_B+16'd625+16'd60)) && (h
sync_cnt <= (`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+1
6'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                     ||((hsync_cnt >= (`HSYNC_B+16'd625+16'd5)) && (hsy
nc_cnt <=(`HSYNC_B+16'd625+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd
15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                     ||((hsync_cnt >=(`HSYNC_B+16'd625+16'd5))&& (hsync
_cnt <= (`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd1
5))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                     ||((hsync_cnt >=(`HSYNC_B+16'd625+16'd5))&& (hsync
_cnt <= (`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd6
7))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                     ||((hsync_cnt >=(`HSYNC_B+16'd625+16'd5)) && (hsyn
c_cnt <= (`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16'd
20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                          VGA_DATA <= `WHITE;            //显示白色
                     else
                          VGA_DATA <= `BLACK;        //显示黑色
                   end
                 else if(second_low==4'd6)//秒钟低位为6
                  begin
                      if(((hsync_cnt >= (`HSYNC_B+16'd625+16'd60)) && (
hsync_cnt <= (`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+
16'd72))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                      ||((hsync_cnt >= (`HSYNC_B+16'd625+16'd5)) && (hs
ync_cnt <=(`HSYNC_B+16'd625+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'
d15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                      ||((hsync_cnt >=(`HSYNC_B+16'd625+16'd5))&& (hsyn
c_cnt <= (`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd
15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                      ||((hsync_cnt >=(`HSYNC_B+16'd625+16'd5))&& (hsyn
c_cnt <= (`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd
67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
```

```
                      ||((hsync_cnt >=(`HSYNC_B+16'd625+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                           VGA_DATA <= `WHITE;            //显示白色
                      else
                           VGA_DATA <= `BLACK;        //显示黑色
                    end
                 else if(second_low==4'd7)//秒钟低位为7
                   begin
                        if(((hsync_cnt >= (`HSYNC_B+16'd625+16'd60)) && (hsync_cnt <=(`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd625+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20))))
                            VGA_DATA<=`WHITE;
                        else
                            VGA_DATA<=`BLACK;
                   end
                 else if(second_low==4'd8)//秒钟低位为8
                   begin
                       if(((hsync_cnt >= (`HSYNC_B+16'd625+16'd60)) && (hsync_cnt <= (`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                       ||((hsync_cnt >= (`HSYNC_B+16'd625+16'd5)) && (hsync_cnt <=(`HSYNC_B+16'd625+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd625+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd625+16'd5))&& (hsync_cnt <= (`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                       ||((hsync_cnt >=(`HSYNC_B+16'd625+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                            VGA_DATA <= `WHITE;            //显示白色
                       else
                            VGA_DATA <= `BLACK;        //显示黑色
                     end
                     else if(second_low==4'd9)//秒钟低位为9
                      begin
```

```
                          if(((hsync_cnt >= (`HSYNC_B+16'd625+16'd60)) && (hsync_cnt <= (`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15)))
                          ||((hsync_cnt >= (`HSYNC_B+16'd625+16'd5)) && (hsync_cnt <=(`HSYNC_B+16'd625+16'd10))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                          ||((hsync_cnt >=(`HSYNC_B+16'd625+16'd5))&& ( hsync_cnt <= (`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd15))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd20)))
                          ||((hsync_cnt >=(`HSYNC_B+16'd625+16'd5))&& ( hsync_cnt <= (`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd225+16'd67))&&(vsync_cnt<=(`VSYNC_P+16'd225+16'd72)))
                          ||((hsync_cnt >=(`HSYNC_B+16'd625+16'd5)) && (hsync_cnt <= (`HSYNC_B+16'd625+16'd65))&&(vsync_cnt>=(`VSYNC_P+16'd375-16'd20))&&(vsync_cnt<=(`VSYNC_P+16'd375-16'd15))))
                               VGA_DATA <= `WHITE;            //显示白色
                          else
                               VGA_DATA <= `BLACK;        //显示黑色
                        end
                   else
                        VGA_DATA<=`BLACK;
                end
                else
                VGA_DATA<=`BLACK;
            end
            else
            VGA_DATA <= `BLACK;            //黑色
        end
       else
       VGA_DATA <= `BLACK;                //黑色
    end



endmodule
```

bluetooth.v:

```
module bluetooth(
    input clk,
    input rst,
    input get,
    output reg [4:0] output_clock
);
    reg [7:0] out;
    parameter bps=10417;
    reg [14:0] count_1;//每一位中的计数器
    reg [3:0] count_2;//每一组数据的计数器
    reg filter_0,filter_1,filter_2;//除去滤波
    wire filter_en;//检测到边沿
    reg add_en;//加法使能信号

    always @ (posedge clk)
    begin
        if(rst)
        begin
            filter_0<=1;
            filter_1<=1;
            filter_2<=1;
        end
        else
        begin
            filter_0<=get;
            filter_1<=filter_0;
            filter_2<=filter_1;
        end
    end

    assign filter_en=filter_2&~filter_1;

    always @ (posedge clk)
    begin
        if(rst)
        begin
            count_1<=0;
        end
        else if(add_en)
        begin
            if(count_1==bps-1)
            begin
                count_1<=0;
            end
            else
            begin
                count_1<=count_1+1;
            end
        end
    end

    always @ (posedge clk)
    begin
        if(rst)
        begin
            count_2<=0;
        end
        else if(add_en&&count_1==bps-1)//如果每一位加
        begin
            if(count_2==8)
            begin
                count_2<=0;
            end
            else
            begin
                count_2<=count_2+1;
            end
        end
    end

    always @ (posedge clk)
    begin
        if(rst)
        begin
            add_en<=0;
        end
        else if(filter_en)
        begin
            add_en<=1;
        end
        else if(add_en&&count_2==8&&count_1==bps-1)
        begin
            add_en<=0;
        end
    end

    always @ (posedge clk)
    begin
        if(rst)
        begin
            out<=0;
        end
        else if(add_en&&count_1==bps/2-1&&count_2!=0)
        begin
            out[count_2-1]<=get;
        end
    end

    always@(*)
        begin
        case(out)
            8'd48: output_clock<=5'd0;
            8'd49: output_clock<=5'd1;
            8'd50: output_clock<=5'd2;
            8'd51: output_clock<=5'd3;
            8'd52: output_clock<=5'd4;
            8'd53: output_clock<=5'd5;
            8'd54: output_clock<=5'd6;
            8'd55: output_clock<=5'd7;
            8'd56: output_clock<=5'd8;
            8'd57: output_clock<=5'd9;
            8'd65: output_clock<=5'd10;
            8'd66: output_clock<=5'd11;
            8'd67: output_clock<=5'd12;
            8'd68: output_clock<=5'd13;
            8'd69: output_clock<=5'd14;
            8'd70: output_clock<=5'd15;
            8'd71: output_clock<=5'd16;
            8'd72: output_clock<=5'd17;
            8'd73: output_clock<=5'd18;
            8'd74: output_clock<=5'd19;
            8'd75: output_clock<=5'd20;
            8'd76: output_clock<=5'd21;
            8'd77: output_clock<=5'd22;
            8'd78: output_clock<=5'd23;
            default: ;
        endcase
        end

endmodule
```