

---

# 目 录

第一章	设计任务和要	
求 . . . . .		
. . . . .		
1.1	设计任务与要	
求 . . . . .		
. . . . .		
1.	2	设计分
析 . . . . .		
. . . . .		
第二章 电路工作原理及方案设计		
第三章	单元电路设计与仿	
真 . . . . .		
. . . . .		
3.1	软件原理	
图 . . . . .		
. . . . .		
3.2	各模块的原理及其程	
序 . . . . .		

---

.....

3.2.1 计数模块的设计及仿真  
图 .....  
.....

3.2.2 控制模块的设计及仿真  
图 .....  
.....

3.2.3 分位模块的设计及仿真  
图 .....  
.....

3.2.4 分频模块的设计及仿真  
图 .....  
.....

3.2.5 译码模块的设计及仿真  
图 .....  
.....

## 第四章

## 心得总

结 .....  
.....

**原创力文档**  
max.book118.com  
预览与源文档一致, 下载高清无水印

## 第一章 设计任务与要求

## 1.1 设计任务与要求

在十字路口的两个方向上各设一组红、绿、黄灯，显示顺序为其中一个方向为绿灯、黄灯、红灯。另一个方向为红灯、绿灯、黄灯。

设置一组数码管，以倒计时的方式显示可以通过的或者禁止通行的时间，其中绿灯、黄灯、红灯的持续时间分别为20S，5S，25S

当各条路上任意一条路上出现故障时，各方向都是红灯亮，倒计时停止，且显示数字在闪烁。当特殊情况结束后，控制器恢复正常。

## 1.2 设计分析

1, 主控制模块:

(1) 当没有故障出现时，灯的运行方法如下所示:

状态	A 路 (主干道)			B 路 (支干道)		
	红	黄	绿	红	黄	绿
S1			√	√		
S2		√		√		
S3	√					√
S4	√				√	

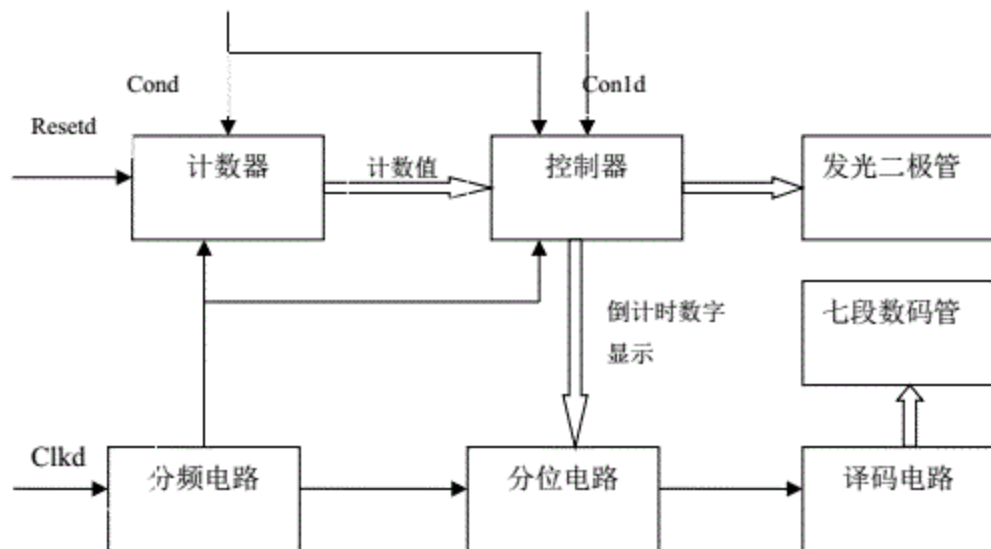
其中，s1,s2,s3,s4 分别是四种状态下的表现,程序我们用了状态机的方法，罗列了4种状态，利用 case 语句把4种状态分别表现出来；部分程序如下：

```
case state is
  when s1=>          -- "=" 等效于then
    reda<='1';      --a路红灯灭
    yellowa<='1';  --a路黄灯灭
    greena<='0';   --a路绿灯亮
    redb<='0';     --b路红灯亮
    yellowb<='1';  --b路黄等灭
    greenb<='1';   --b路黄等灭
    numa<=count;
    numb<=count+5;
```

可见，一路灯走完一个周期  $T=50s$ (红 25+黄 5+绿 20).

## 第二章 电路工作原理及方案设计

总体框图:



在 VHDL 设计描述中, 采用自顶向下的设计思路, 该思路, 首先要描述顶层的接口, 上面的描述已经规定了交通灯控制的输入输出信号: 输入信号: 复位开关信号 reset; 外部时钟信号 clk。LED 七段显示数码管的输出信号 count1(6 downto 0), count2(6 downto 0), count3(6 downto 0), count4(6 downto 0); 在自顶向下的 VHDL 设计描述中, 通常把整个设计的系统划分为几个模块, 然后采用结构描述方式对整个系统进行描述。根据实验设计的结构功能, 来确定使用哪些模块以及这些模块之间的关系。通过上面的分析, 不难得知可以把交通灯控制系统划分为 4 个模块: 时钟分频模块, 计数模块, 控制模块, 分位译码模块。

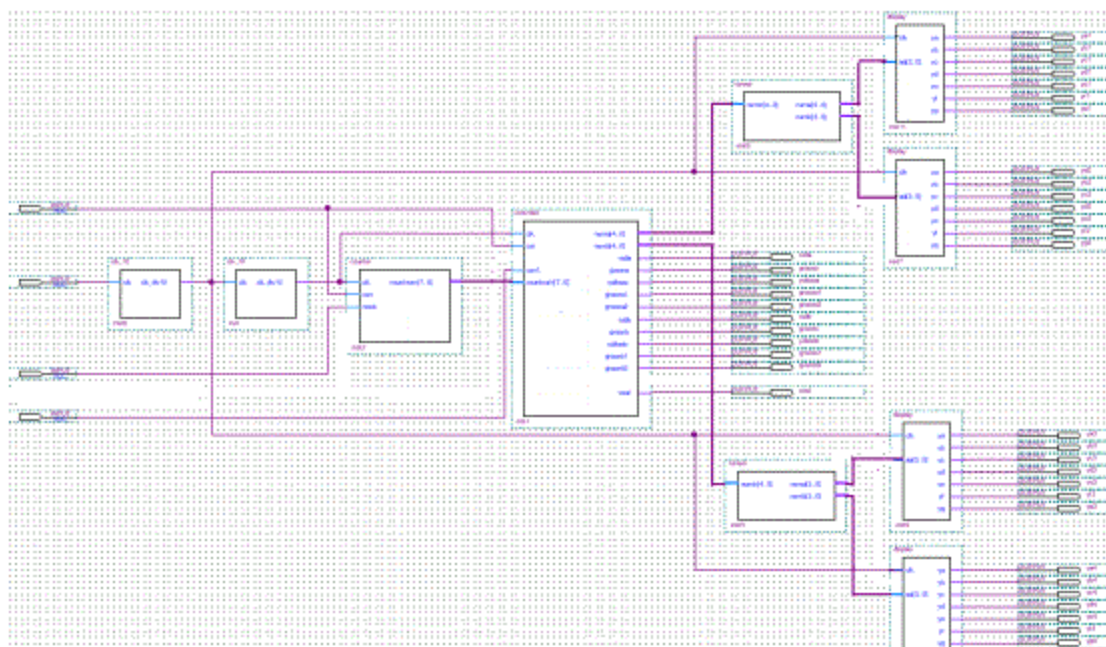
**控制器电路:** 根据计数器的计数值控制发光二极管的亮、灭, 以及输出倒计时数值给七段译码管的分位译码电路。当检测到手动控制信号 (Con='1') 时, 执行手动控制;

**计数器电路:** 这里需要的计数器的计数范围为 0~89。计到 89 后, 下一个时钟沿升为 1 时, 开始新一轮计数, 此外当系统复位信号 (Reset='1') 使计数器异步清 '0'。手动信号 (Con='1') 使系统清 '0'。

**分位译码电路:** 因为控制器输出的倒计时数值可能是 1 位或者 2 位十进制数, 所以在七段数码管的译码电路前要加上分位电路 (即将其分为 2 个 1 位的十进制数, 如 20 分为 2 和 0, 7 分为 0 和 7)。七段数码管的译码电路根据控制电路的控制信号, 驱动交通灯的显示, 通过输入二进制数值, 输出信号点亮二极管, 我们用的是共阳极数码管, 因此译码电路输出逻辑数值 '0' 点亮二极管, 译码电路输出逻辑数值 '1' 熄灭二极管。

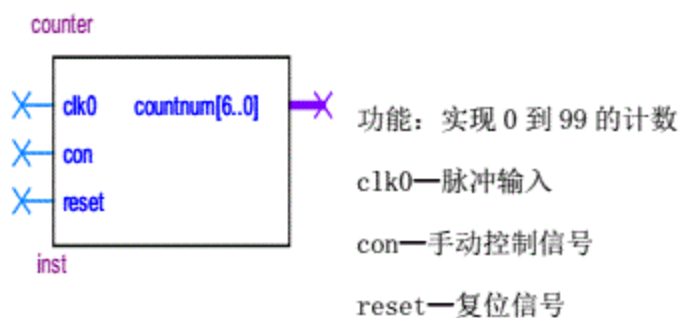
## 第三章 单元电路设计与仿真

### 3.1 软件原理图



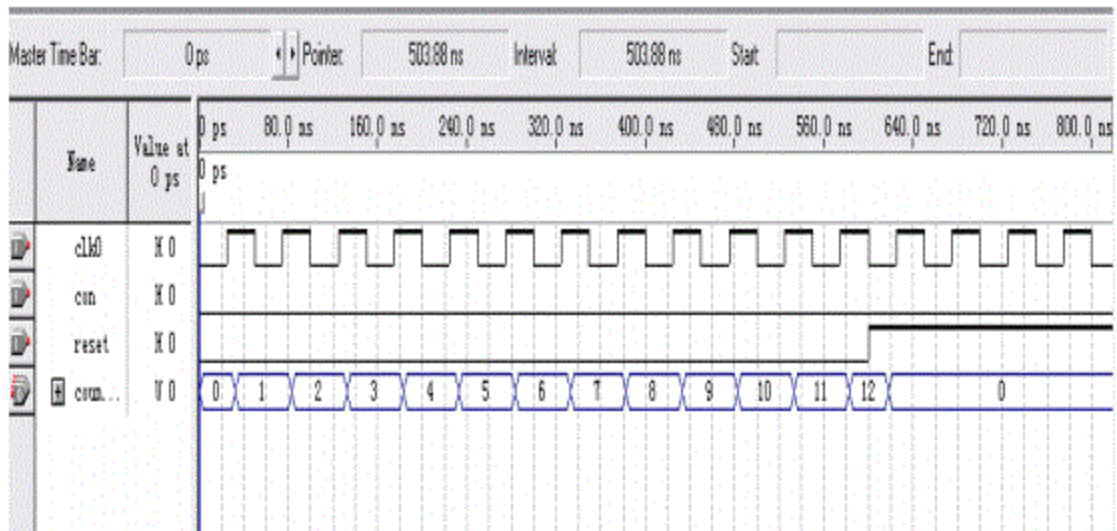
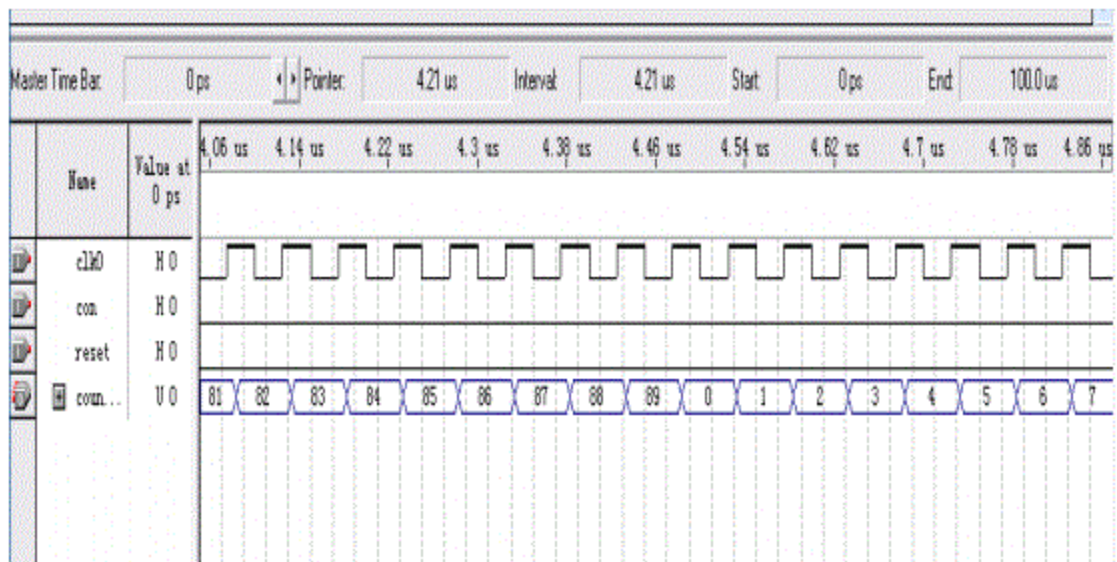
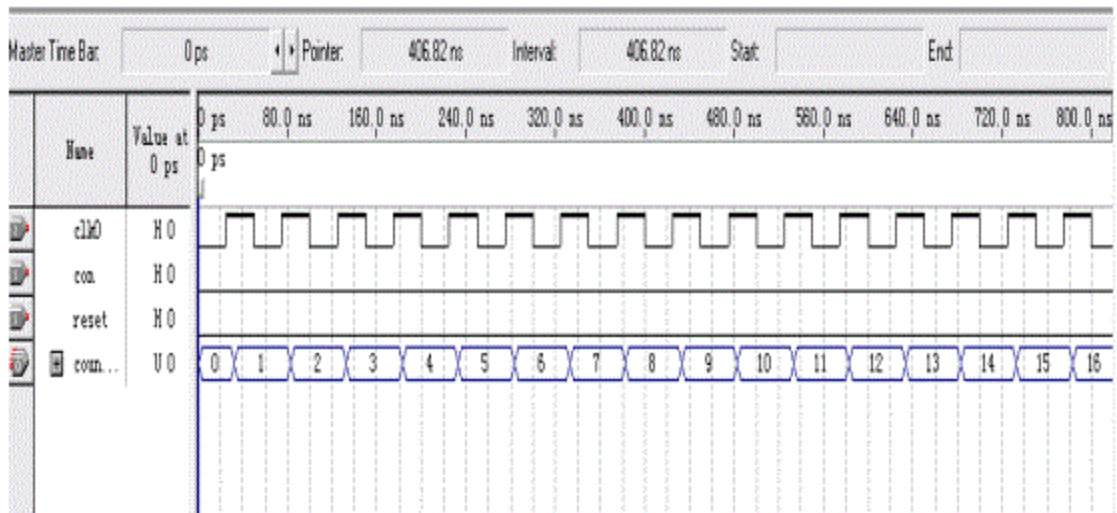
### 3.2 各模块的原理及其程序

#### 3.2.1 计数模块的设计及仿真图



countnum—计数输出

仿真波形如下：



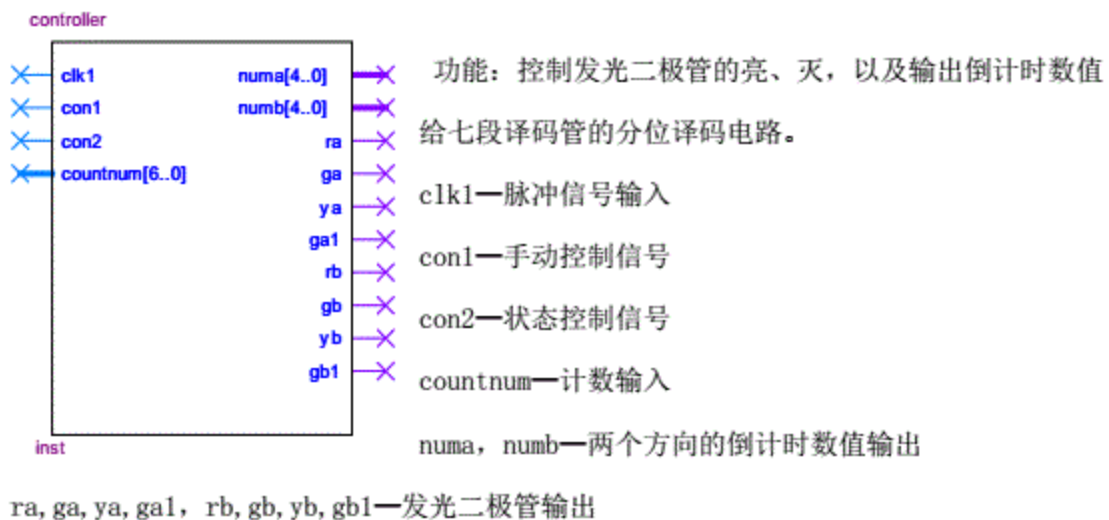
模块程序: library ieee;

```

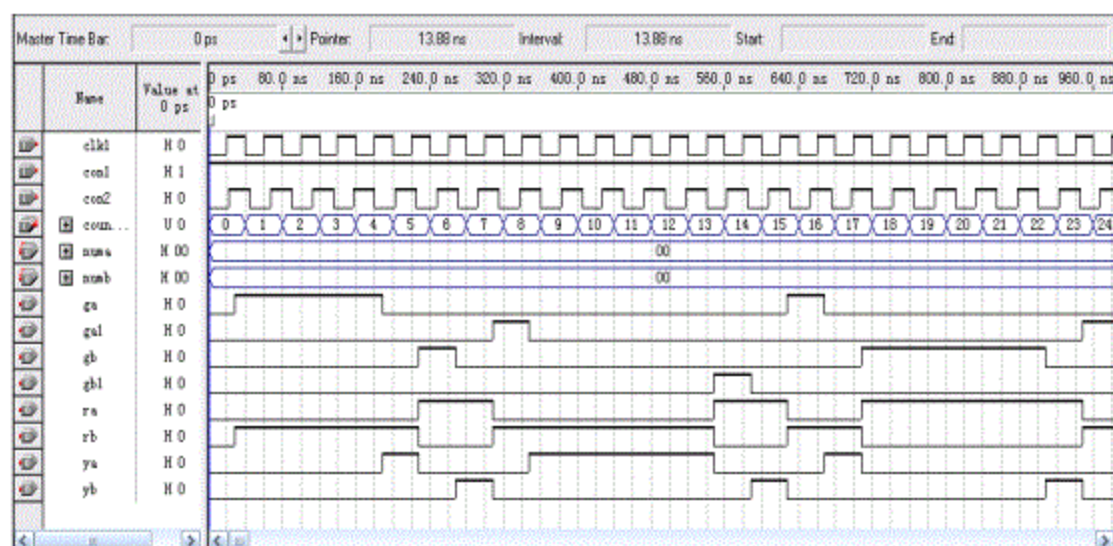
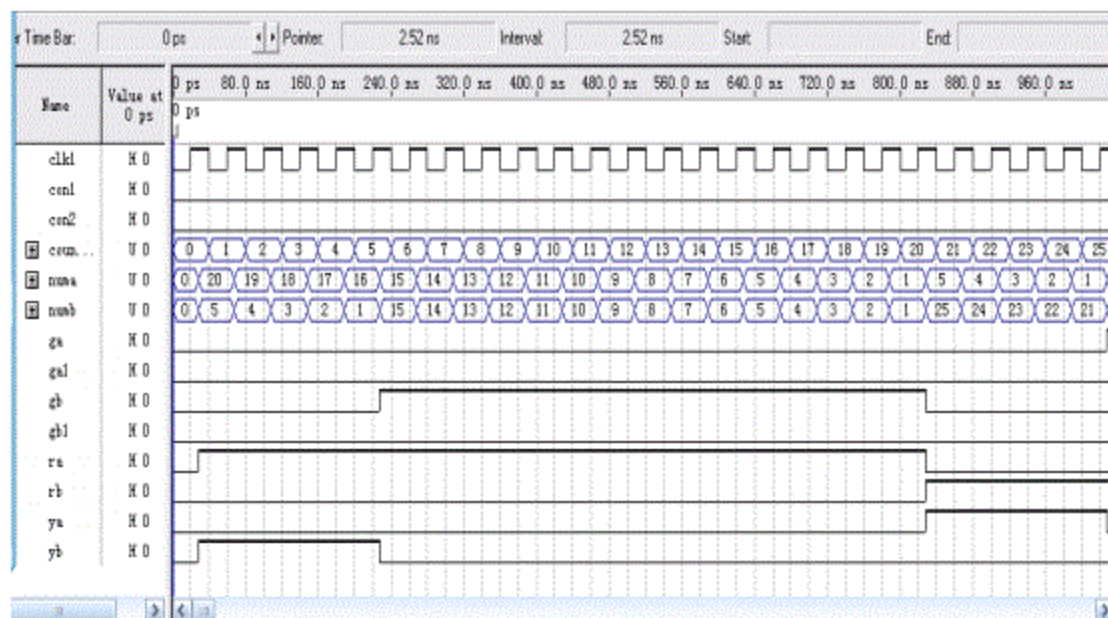
        use ieee.std_logic_1164.all;
        entity counter is
            port
            ( clk0:in std_logic;
              con:in std_logic;
              reset:in std_logic;
              countnum:buffer integer range 0 to 99);
        end counter;
        architecture one of counter is
        begin
            process (reset,clk0)
            begin
                if reset='1' then
                    countnum <= 0;
                elsif rising_edge( clk0 ) then
                    if con='1' then
                        countnum <= 0;
                    else
                        if countnum=99 then
                            countnum <= 0;
                        else
                            countnum <= countnum + 1;
                        end if;
                    end if;
                end if;
            end if;
        end process;
    end one;

```

### 3.2.2 控制模块的设计及仿真图



仿真波形如下：



模块程序：

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity controller is
port
    (
        clk1 : in std_logic;

```



---

```

    con1 : in std_logic;
    con2 : in std_logic;
    countnum : in integer range 0 to 89;
    numa,numb : out integer range 0 to 25;
    ra,ga,ya,ga1 : out std_logic;
    rb,gb,yb,gb1: out std_logic);
end controller;
architecture one of controller is
signal m : integer range 0 to 7;
begin
    process ( clk1,con1,con2, countnum )
        variable aa:std_logic_vector(7 downto 0);
        begin
            if (clk1'event and clk1 = '1' )then
if con1 ='0' then
            if countnum >= 65then
                numb<=90-countnum;
                numa <=90-countnum;
                if countnum >= 70 then
                    aa := "00011000";
                else
                    aa := "00101000";
                numa<= 70-countnum;
            end if;
elseif countnum >= 45 then
                numb <= 65-countnum;
numa <=65-countnum;
            if countnum >= 50 then
                aa := "10000100";
                else aa := "10000010";
                numb<= 50-countnum;
            end if;
elseif countnum >= 20 then
                numa <= 45-countnum;
numb <=45-countnum;
            if countnum >= 25 then
                aa := "01001000";
                else aa := "00101000";
                numa <= 25-countnum;

            end if;
            elsif countnum >= 0 then
                numa <= 20-countnum;
numb <= 20-countnum;
                if countnum >= 5 then

```

---

```
        aa := "10000100";
    else aa := "10000010";
        numb <= 5-countnum;
end if;
end if;
else
if con1= '1' then
numa <= 0;
numb <= 0;
if con2 = '1' then
if m = 7 then
m <= 0;
else
m <= m + 1;
end if;
end if;
if m = 0 then
    aa := "01001000";
elsif m= 1 then
    aa := "00101000";
    elsif m=2 then
        aa :="10000100";
    elsif m=3 then
        aa := "10000010";
    elsif m=4 then
        aa := "00011000";
    elsif m=5 then
        aa := "00101000";
    elsif m=6 then
        aa := "10000001";
    elsif m=7 then
        aa := "10000010";

end if;

end if;

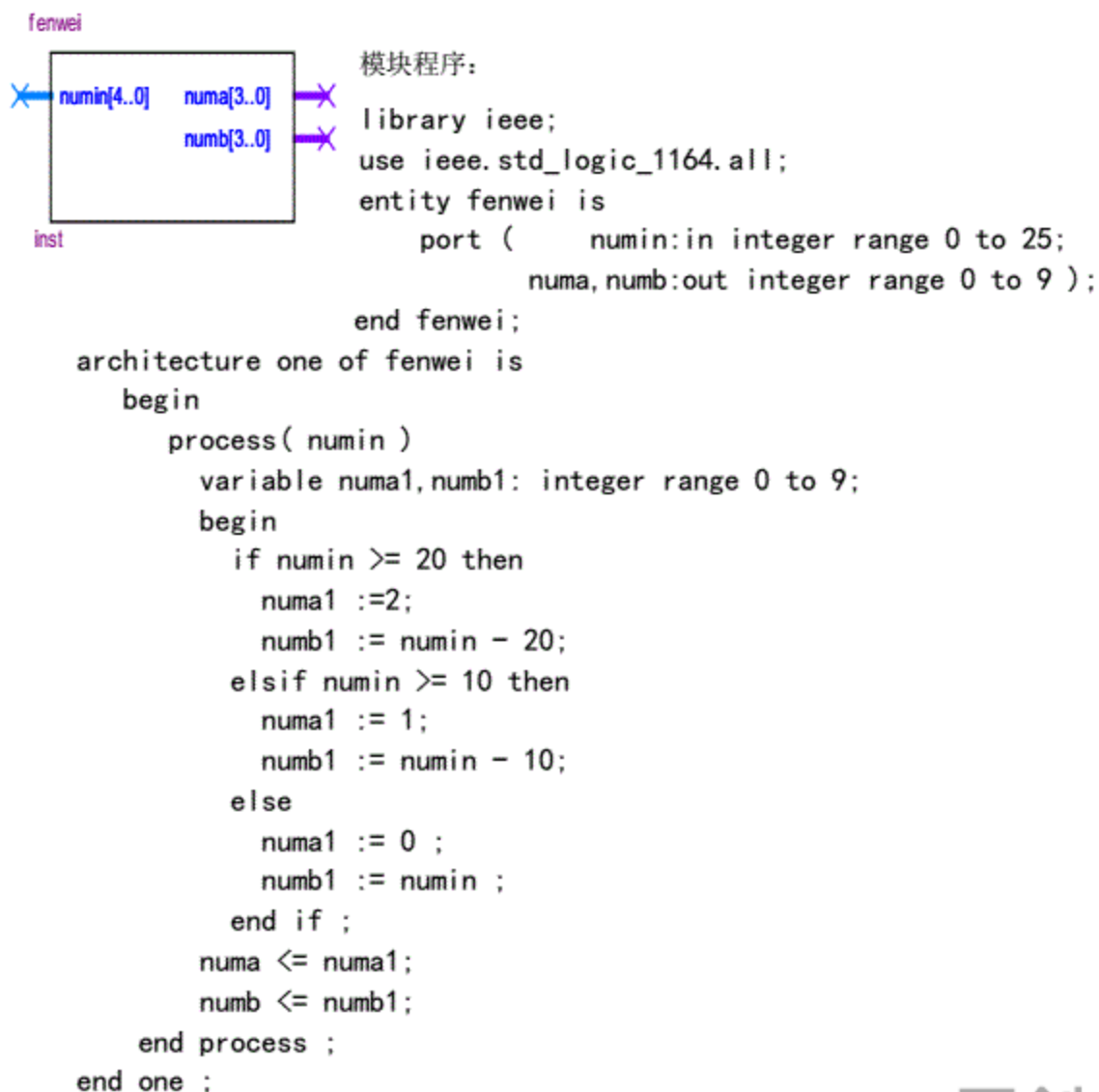
end if;
end if;
    ra <= aa(7);
    ga <= aa(6);
    ya <= aa(5);
    ga1 <= aa(4);
```

```

    rb <= aa(3);
    gb <= aa(2);
    yb <= aa(1);
    gb1 <= aa(0);
end process;
end one;

```

### 3.2.3 分位模块的设计与仿真图

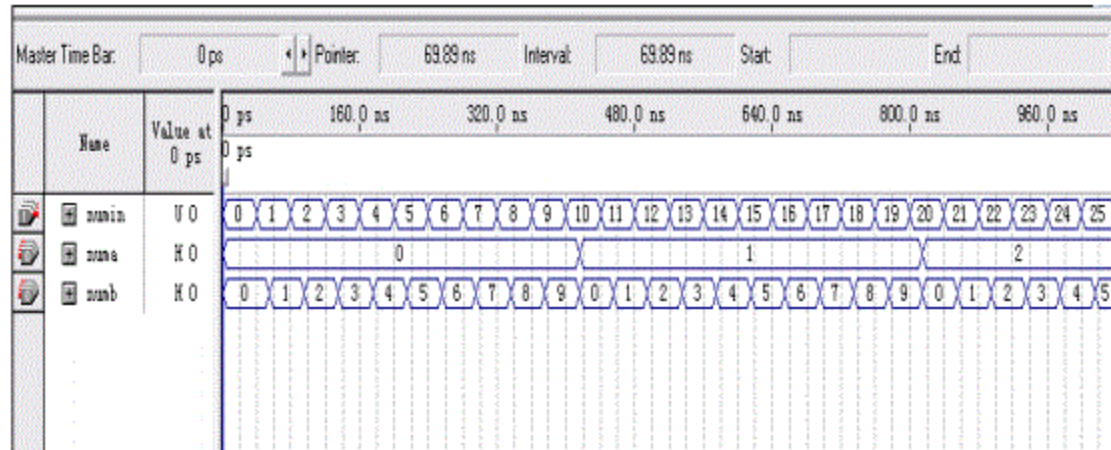


功能：把倒计时的数值分成 2 个 1 位的十进制数。

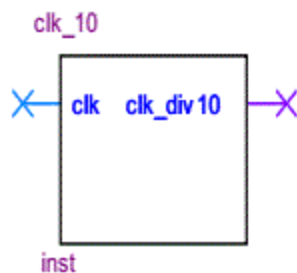
numin: 倒计时数值输入

numa, numb—将数值分为 2 个 1 位的十进制输出

仿真波形如下：



### 3.2.4 分频模块的设计及仿真



#### 模块程序

```
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity clk_10 is
port(
    clk : in std_logic;
    clk_div10 : out std_logic);
end clk_10;
architecture one of clk_10 is
signal clk_temp : std_logic;
begin
process(clk)
variable counter : std_logic_vector(2 downto 0);
constant md : std_logic_vector(2 downto 0) := "101";
begin
```

```

if(clk'event and clk='1') then
if(counter = md)then
counter := (others =>'0');
clk_temp <= not clk_temp;
end if;
counter := counter + 1 ;
end if;
end process;
clk_div10 <= clk_temp;
end one;

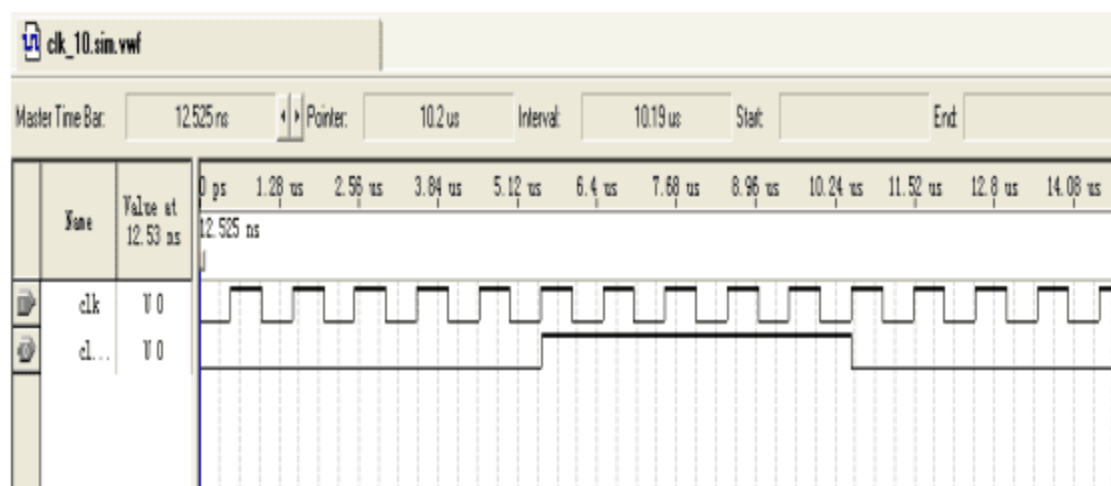
```

功能：实现分频

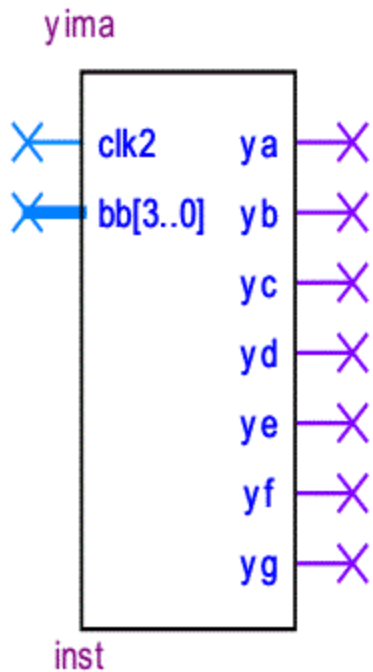
接口：clk—脉冲输入

Clk-div10—分频后脉冲输出

仿真结果如下：

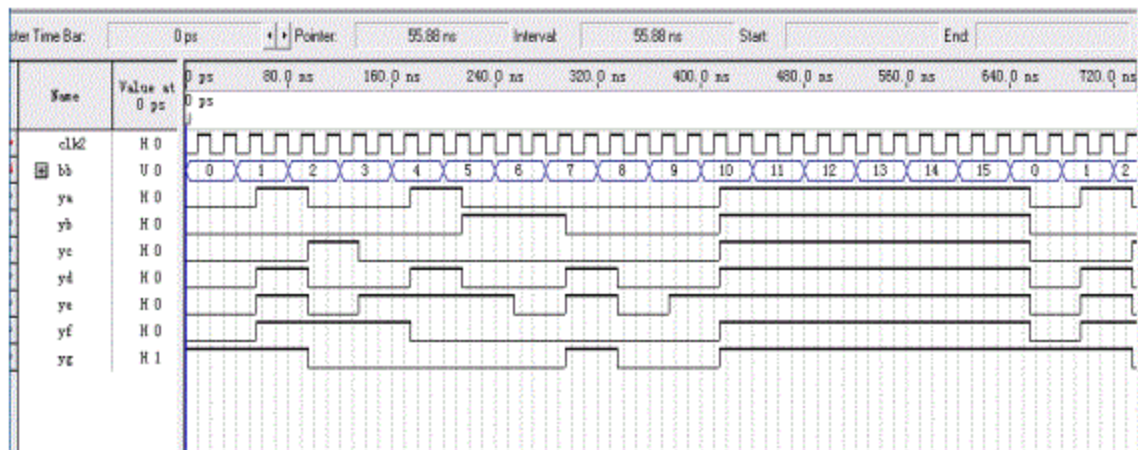


### 3.2.5 译码模块的设计及仿真图



功能: 根据控制电路的控制信号, 驱动交通灯的显示,  
通过输入二进制数值, 输出信号点亮二极管。  
bb[3..0]—BCD 码输入  
clk2—脉冲输入  
ya, yb, yc, yd, ye, yf, yg—七段数码管显示输出

仿真波形如下:



程序模块:

```
library ieee;  
use ieee.std_logic_1164.all;  
entity yima is  
    port(  
        clk2: in std_logic;  
        bb: in std_logic_vector(3 downto 0);  
        ya: out std_logic;  
        yb: out std_logic;  
        yc: out std_logic;
```

---

```
        yd:out std_logic;
        ye:out std_logic;
        yf:out std_logic;
        yg:out std_logic);
end display;
architecture one of yima is
    signal temp: std_logic_vector(6 downto 0);
begin
    process(bb)
    begin
        case bb is
        when "0000" => temp <= "1000000";
        when "0001" => temp <= "1111001";
        when "0010" => temp <= "0100100";
        when "0011" => temp <= "0110000";
        when "0100" => temp <= "0011001";
        when "0101" => temp <= "0010010";
        when "0110" => temp <= "0000010";
        when "0111" => temp <= "1011000";
        when "1000" => temp <= "0000000";
        when "1001" => temp <= "0010000";
        when others => temp <= "1111111";
        end case;
    end process;
    ya<=temp(0);
    yb<=temp(1);
    yc<=temp(2);
    yd<=temp(3);
    ye<=temp(4);
    yf<=temp(5);
    yg<=temp(6);
end one;
```

## 第四章 心得总结

通过实验设计能提高学生对所学知识的综合应用能力，能全面检查并掌握所学内容，本学期我们进行了 VHDL 实验设计，老师命题，六人一大组，分工合作进行设计（包括设计总体方案、软件原理、计数模块、控制模块、分位模块、分频模块、译码模块

---

等工作)。我们做的是交通灯控制器的设计。

确定题目后，我们首先进行了单元模块的设计，将每一个单元模块设计完成后再进行仿真，仿真成功后就可以进行顶层文件的编写了，在顶层文件的编写过程中遇到了一些问题，特别是各模块之间的连接，以及信号的定义，总是有错误。有的时候信号的定义容易出现混淆，在反复的修改过后，顶层文件终于能够编译成功了。在波形仿真的过程中，同样遇到了困难，有的时候，由于END TIME的时间修改的太大，会出现仿真时间过长的问题，这个时候应该要把END TIME的时间相应的改小，或是修改系统时钟的频率。在设计的过程中还应该多联系下实际情况，要了解实际情况下交通信号灯的工作情况，才能更好的完成此次的课程设计。在今后的工作和学习中，我们不能仅仅把目光停留在课本上，要多理论联系实际。有的时候，理论上是正确的东西放到现实中去，可能由于种种因素的制约，并不能达到实际的效果，还需要我们进行相应的修改才能完成要求。这次的课程设计使我巩固了以前学习到的知识，还使我掌握了以前没有掌握的知识，同时锻炼了自己的能力。

在这次实验设计中，在收获知识的同时，还收获了阅历，收获了成熟，在此过程中，我们通过查找大量资料，请教师兄，以及不懈的努力，不仅培养了独立思考、动手操作的能力，在各种其它能力上也都有了提高。更重要的是，我们学会了很多学习的方法。而这是日后最实用的，真的是受益匪浅。要面对社会的挑战，只有不断的学习、实践，再学习、再实践。

通过几天的课程设计，我对数据库软件 EDA 技术、VHDL、等系列知识都有了一定的了解。使用 EDA 技术开发页面的能力也有了很大提高。同时也掌握了做课程设计的一般流程，为以后的设计积累了一定的经验。做课程设计时，先查阅相关知识，把原理吃透，确定一个大的设计方向，在按照这个方向分模块的把要实现的功能用流程图的形式展示。最后参照每个模块把输入和输出引脚设定，运用我们所学的 VHDL 语言进行编程。总之，通过这次的设计，进一步了解了 EDA 技术，收获很大，对软件编程、排错调试、相关仪器设备的使用技能等方面得到较全面的锻炼和提高。

另外，我还学到了一个人的力量毕竟有限，但是团队的力量势不可挡，我们不能只靠自己，遇到不会的要多多求助同学，

这次设计虽然结束了，也留下了很多遗憾，因为由于期末，面临各科考试，时间的紧缺，并没有做到最好。相信以后我们会以更加积极地态度对待我们的学习、对待我们的生活。我们的激情永远不会结束，相反，我们会更加努力，努力的去弥补自己的缺点，发展自己的优点，去充实自己，只有在了解自己的长短之后，我们会更加珍惜拥有的，更加



努力的去完善它，增进它。

### 关于使用 VHDL 的问题：

①VHDL 的语法问题：由于初次接触 VHDL 硬件编程语言，容易犯一些小错误，像端口括号前分号不加，二进制变量与十进制变量加不加引号的问题等等。当然大部分语法问题编译的时候能够报错从而发现。但是，也有一些小规则，虽然语法通过了，但是功能却不能实现。本实验的代码调试中发现了大量的这样的问题。

②可能是由于编译器的一些原因，在使用 case.....when.....语句时,出现了一些不可预知的问题。由于要实现的是一个比较捕获触发器，所以只有在某几个 输入状态的时候需要改变输出的值，而在其他输入状态时输出量要保持，使用 case 语句，但在实验中发现，case 语句中没有使用的状态虽然用了 others 关键字进行设置，但实际运行中其输出并不稳定为我们希望的值。最终将所有可能输入情况的 case 状态均进行了设置，才解决了这个问题，看来以后 case when 语句要少用啊！！

### 参考书目

- |                   |         |
|-------------------|---------|
| (1) VHDL 数字控制系统设计 | 电子工业出版社 |
| (2) EDA 技术与 VHDL  | 清华大学出版社 |

### 附录 总程序

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity dingceng is
port (clkd, resetd, cond, con1d: in
std_logic;
count1:out std_logic_vector (6
downto 0);
count2:out std_logic_vector (6
downto 0);
count3:out std_logic_vector (6
downto 0);
count4:out std_logic_vector (6
downto 0);
reda, greena, yellowa, greena1: out
std_logic;
redb, greenb, yellowb, greenb1: out
std_logic);
end entity dingceng;
architecture one of dingceng is
component clk_10
port (clk: in std_logic;
clk_div10 : out
std_logic);
end component;
component counter
port
(clk0: in std_logic;
con: in std_logic;
reset: in std_logic;
```

原创力文档

www.lanxue.com  
预览与原文档一致 下载高清无水印

```

        countnum:buffer
integer range 0 to 89);
end component;
component controller
port
    ( clk1 : in std_logic;
      con1 : in std_logic;
      con2: in std_logic;
      countnum : in integer
range 0 to 89;
      numa,numb : out integer
range 0 to 25;
      ra, ga, ya, ga1: out
std_logic;
      rb, gb, yb, gb1 : out
std_logic);
end component;
component fenwei
port ( numin:in integer range 0
to 25;
      numa,numb:out integer
range 0 to 9 );
end component;
component yima
port(clk2:in std_logic;
      bb: in integer range
0 to 9;
      ya:out std_logic;
      yb:out std_logic;
      yc:out std_logic;
      yd:out std_logic;
      ye:out std_logic;
      yf:out std_logic;
      yg:out std_logic);
end component;
signal a, b, c:std_logic;
signal ww:integer range 0 to 89;
signal yy1,yy2:integer range 0
to 25;
signal tt1, tt2, tt3, tt4: integer
range 0 to 9;
begin
u0:          clk_10          port
map(clk=>clkd,clk_div10=>a);
u1:          clk_10          port
map(clk=>clkd,clk_div10=>b);
u2:          clk_10          port
map(clk=>b,clk_div10=>c);
u3:counter port
map(clk0=>c, con=>cond, reset=>res
etd, countnum=>ww);
u4:controller port map
(clk1=>c, con1=>cond, con2=>con1d,
countnum=>ww, numa=>yy1, numb=>yy2, r
a=>reda, ga=>greena, ya=>yellowa, g
a1=>greena1, rb=>redb, gb=>greenb,
yb=>yellowb, gb1=>greenb1);
u5:fenwei          port
map(numin=>yy1, numa=>tt1, numb=>t
t2);
u6:fenwei          port      map
(numin=>yy2, numa=>tt3, numb=>tt4);
u7:yima          port
map(clk2=>b, bb=>tt1, ya=>count1(0),
yb=>count1(1), yc=>count1(2), yd=>
count1(3), ye=>count1(4), yf=>coun
t1(5), yg=>count1(6));
u8:yima          port      map
(clk2=>b, bb=>tt2, ya=>count2(0), y
b=>count2(1), yc=>count2(2), yd=>c
ount2(3), ye=>count2(4), yf=>count
2(5), yg=>count2(6));
u9:yima          port
map(clk2=>b, bb=>tt3, ya=>count3(0),
yb=>count3(1), yc=>count3(2), yd=>
count3(3), ye=>count3(4), yf=>coun
t3(5), yg=>count3(6));
u10:yima          port
map(clk2=>b, bb=>tt4, ya=>count4(0),
yb=>count4(1), yc=>count4(2), yd=>
count4(3), ye=>count4(4), yf=>coun
t4(5), yg=>count4(6));
end architecture one;

library ieee;
use ieee.std_logic_1164.all;

```

```

use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity clk_10 is
port(      clk : in std_logic;
      clk_div10 : out std_logic);
end clk_10;
architecture one of clk_10 is
signal clk_temp : std_logic;
begin
process(clk)
variable      counter      :
std_logic_vector(2 downto 0);
constant md : std_logic_vector(2
downto 0) := "101";
begin
if(clk'event and clk='1') then
if(counter = md)then
counter := (others =>'0');
clk_temp <= not clk_temp;
end if;
counter := counter + 1 ;
end if;
end process;
clk_div10 <= clk_temp;
end one;

```

```

library ieee;
use ieee.std_logic_1164.all;
entity counter is
port
( clk0:in std_logic;
  con:in std_logic;
  reset:in std_logic;
  countnum:buffer integer range
0 to 89);
end counter;
architecture one of counter is
begin
process (reset,clk0)
begin
if reset='1' then
countnum <= 0;
elsif rising_edge( clk0 )
then

```

```

if con='1' then
countnum <= 0;
else
if countnum=89 then
countnum <= 0;
else
countnum <= countnum
+ 1;
end if;
end if;
end if;
end process;
end one;

```

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity controller is
port
(      clk1 : in std_logic;
      con1 : in std_logic;
      con2 : in std_logic;
      countnum : in integer range 0
to 89;
      numa,numb : out integer range 0
to 25;
      ra,ga,ya,ga1 : out std_logic;
      rb,gb,yb,gb1: out std_logic);
end controller;
architecture one of controller
is
signal m : integer range 0 to 7;
begin
process
(
clk1,con1,con2,countnum )
variable
aa:std_logic_vector(7 downto 0);
begin
if (clk1'event and clk1 = '1'
)then
if con1 ='0' then
if countnum >= 65then
numb<=90-countnum;
numa <=90-countnum;

```

---

```

    if countnum >= 70 then
        aa := "00011000";
    else
        aa := "00101000";
        numa <= 70-countnum;
    end if;
    elsif countnum >= 45 then
        numb <= 65-countnum;
        numa <= 65-countnum;
        if countnum >= 50 then
            aa := "10000100";
            else aa := "10000010";
            numb <= 50-countnum;
        end if;
    elsif countnum >= 20 then
        numa <= 45-countnum;
        numb <= 45-countnum;
        if countnum >= 25 then
            aa := "01001000";
            else aa := "00101000";
            numa <= 25-countnum;
        end if;
    elsif countnum >= 0 then
        numa <= 20-countnum;
        numb <= 20-countnum;
        if countnum >= 5 then
            aa := "10000100";
            else aa := "10000010";
            numb <= 5-countnum;
        end if;
    end if;
    else
        if con1= '1' then
            numa <= 0;
            numb <= 0;
            if con2 = '1' then
                if m = 7 then
                    m <= 0;
                else
                    m <= m + 1;
                end if;
            end if;
        end if;
        if m = 0 then
            aa := "01001000";

```

```

    elsif m= 1 then
        aa := "00101000";
    elsif m=2 then
        aa := "10000100";
    elsif m=3 then
        aa := "10000010";
    elsif m=4 then
        aa := "00011000";
    elsif m=5 then
        aa := "00101000";
    elsif m=6 then
        aa := "10000001";
    elsif m=7 then
        aa := "10000010";
    end if;
end if;

end if;

end if;
    ra <= aa(7);
    ga <= aa(6);
    ya <= aa(5);
    ga1 <= aa(4);
    rb <= aa(3);
    gb <= aa(2);
    yb <= aa(1);
    gb1 <= aa(0);
end process;
end one;

library ieee;
use ieee.std_logic_1164.all;
entity fenwei is
    port (
        numin:in integer
        range 0 to 25;
        numa,numb:out integer
        range 0 to 9 );
end fenwei;
architecture one of fenwei is
    begin
        process( numin )

```

---

```

        variable      numa1,numb1:
integer range 0 to 9;
    begin
        if numin >= 20 then
            numa1 :=2;
            numb1 := numin - 20;
        elsif numin >= 10 then
            numa1 := 1;
            numb1 := numin - 10;
        else
            numa1 := 0 ;
            numb1 := numin ;
        end if ;
        numa <= numa1;
        numb <= numb1;
    end process ;
end one ;

        when "0010" => temp <= "0100100";
        when "0011" => temp <= "0110000";
        when "0100" => temp <= "0011001";
        when "0101" => temp <= "0010010";
        when "0110" => temp <= "0000010";
        when "0111" => temp <= "1011000";
        when "1000" => temp <= "0000000";
        when "1001" => temp <= "0010000";
        when others => temp <= "1111111";
    end case;
end process;
ya<=temp(0);
yb<=temp(1);
yc<=temp(2);
yd<=temp(3);
ye<=temp(4);
yf<=temp(5);
yg<=temp(6);
end one;

library ieee;
use ieee.std_logic_1164.all;
entity display is
    port(
        clk2:in std_logic;
        bb:                                     in
std_logic_vector(3 downto 0);

        ya:out std_logic;
        yb:out std_logic;
        yc:out std_logic;
        yd:out std_logic;
        ye:out std_logic;
        yf:out std_logic;
        yg:out std_logic);
end display;
architecture one of display is
    signal temp: std_logic_vector(6
downto 0);
    begin
    process(bb)
    begin
    case bb is
    when "0000" => temp <= "1000000";
    when "0001" => temp <= "1111001";

```