课程设计

实验指导

- 药片装瓶系统设计
- 电子钟
- 选做

知识准备

- 复习

《数字逻辑》 VHDL内容

通知

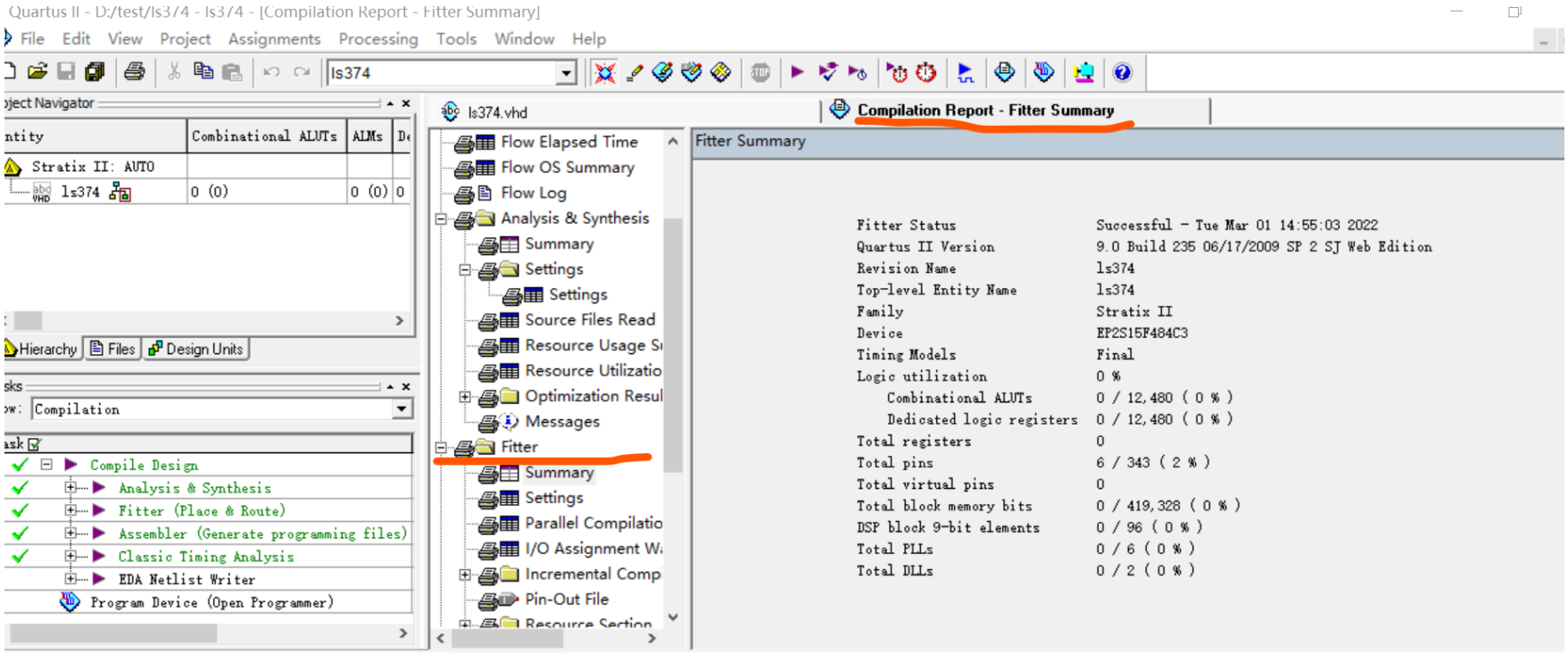
其他所有由各班实验老师安排。

实验内容

- 药片装瓶系统设计
- 电子钟；
- 自选实验由实验老师给定

资源占用

先设计、再仿真，看占用资源数，Compilation Report→Fitter 器件资源利用情况，引脚分配情况



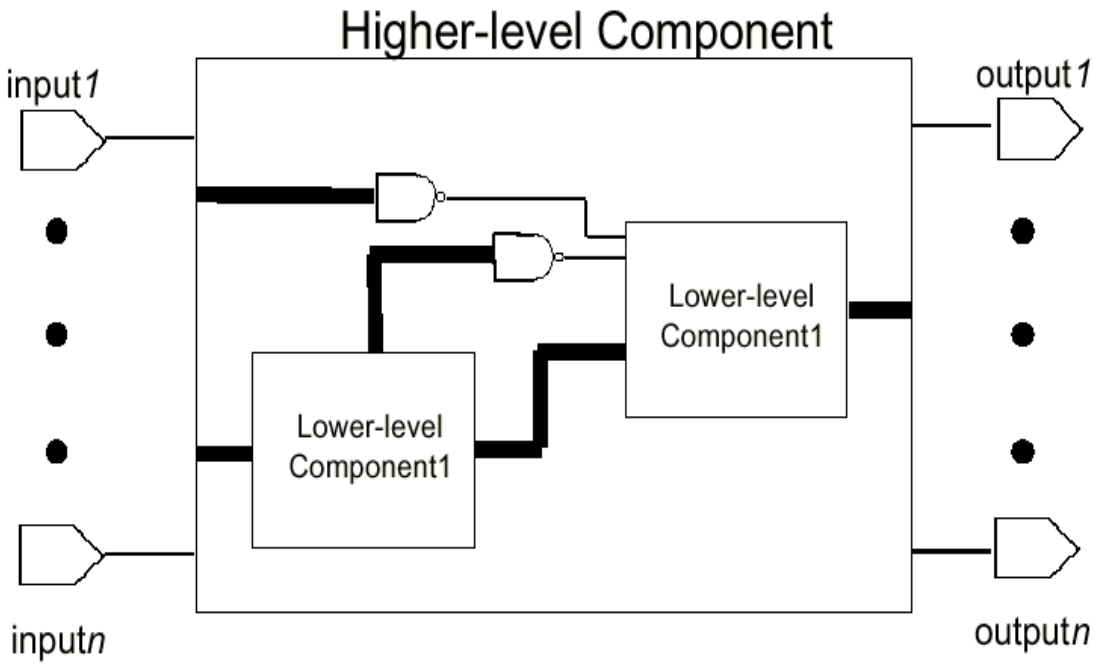
尽量少用多层嵌套的 if then else

编译之后，点击

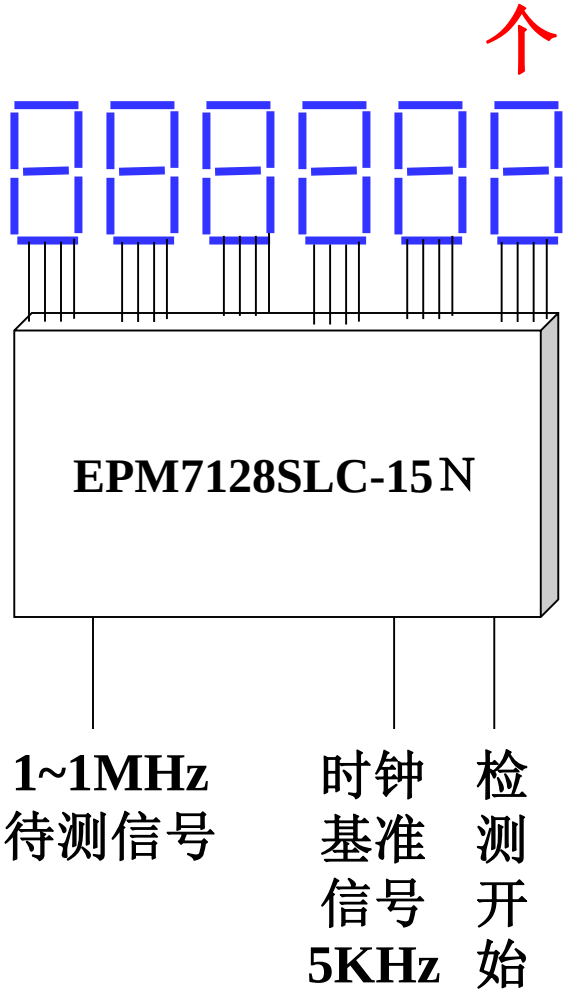
Tool>>Netlist Viewers>>RTL Viewers

设计思路

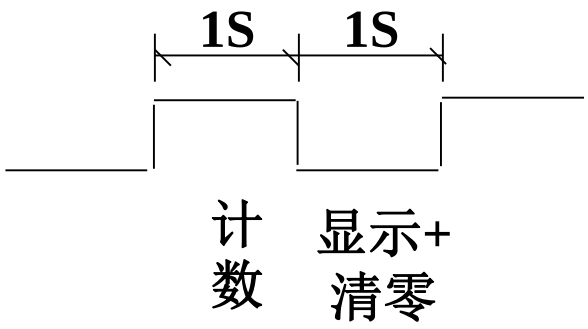
自顶向下，逐级细化
采用结构化（模块化）设计思想

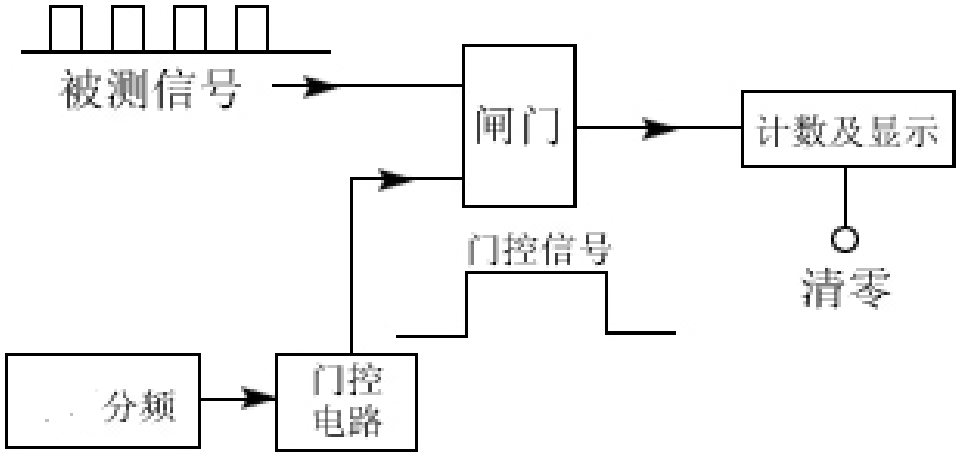


简易频率计



1S的闸门信号内计数的结果
就是被测信号的频率





1. 产生准确闸门信号（1s）。

5kHz时钟经5000分频（一次完成），再经2分频产生方波，1s用于计数，1s用于显示结果及清零）；

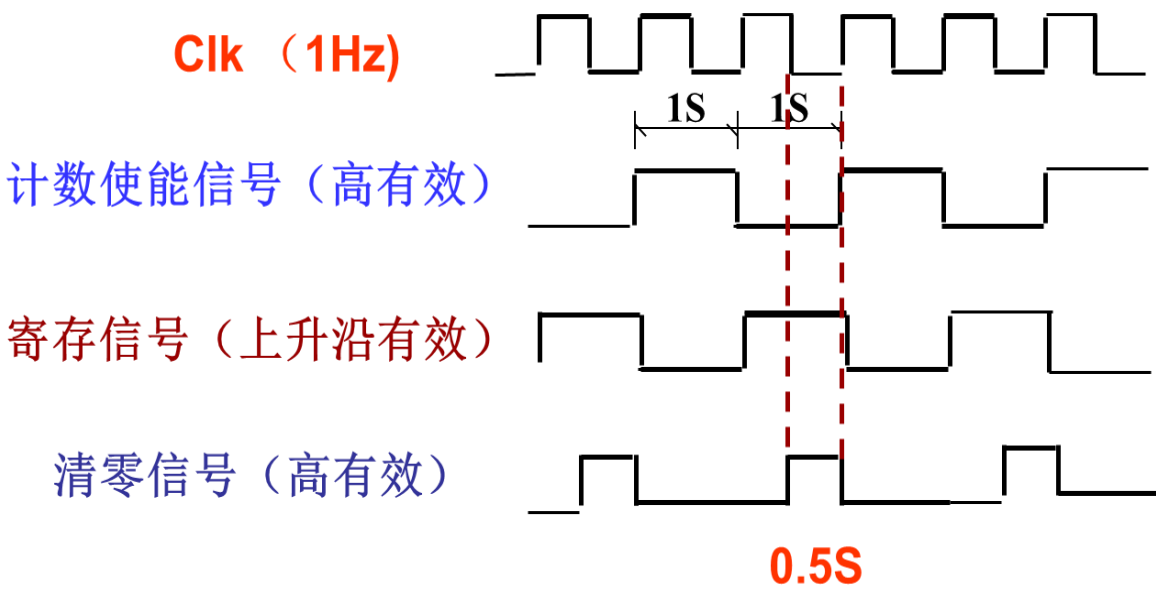
2. 使能控制计数，计数结果寄存；

3.每次对被测信号计数前，自动异步清零；

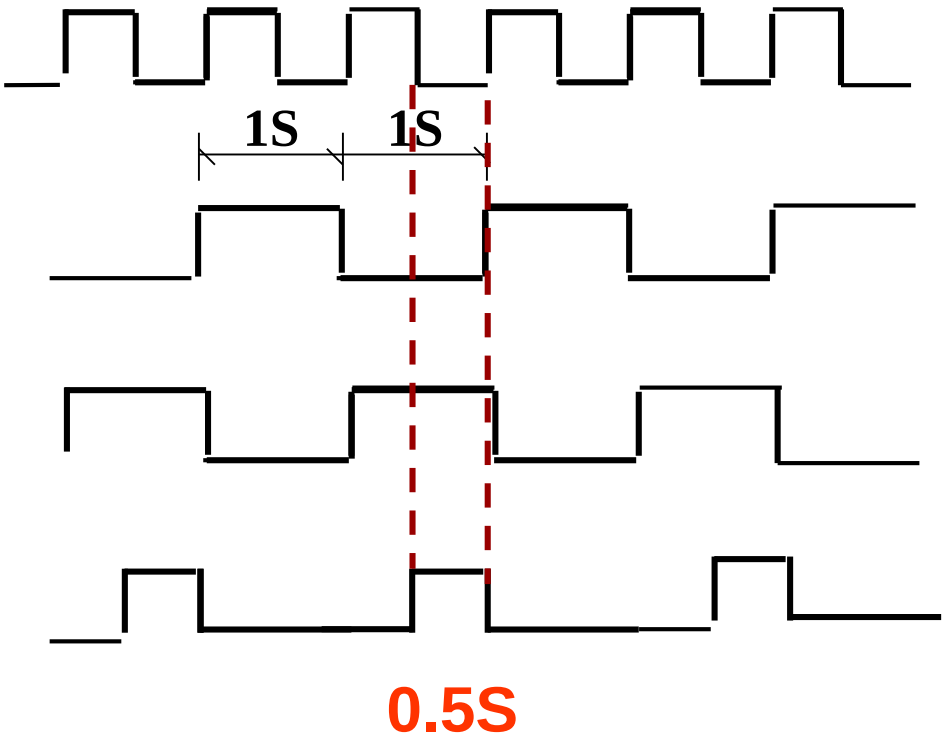
4. 计数器采用十进制；

5.显示器有1个需要写显示译码功能，驱动7段发光管，其他5个直接输出8421码给显示器

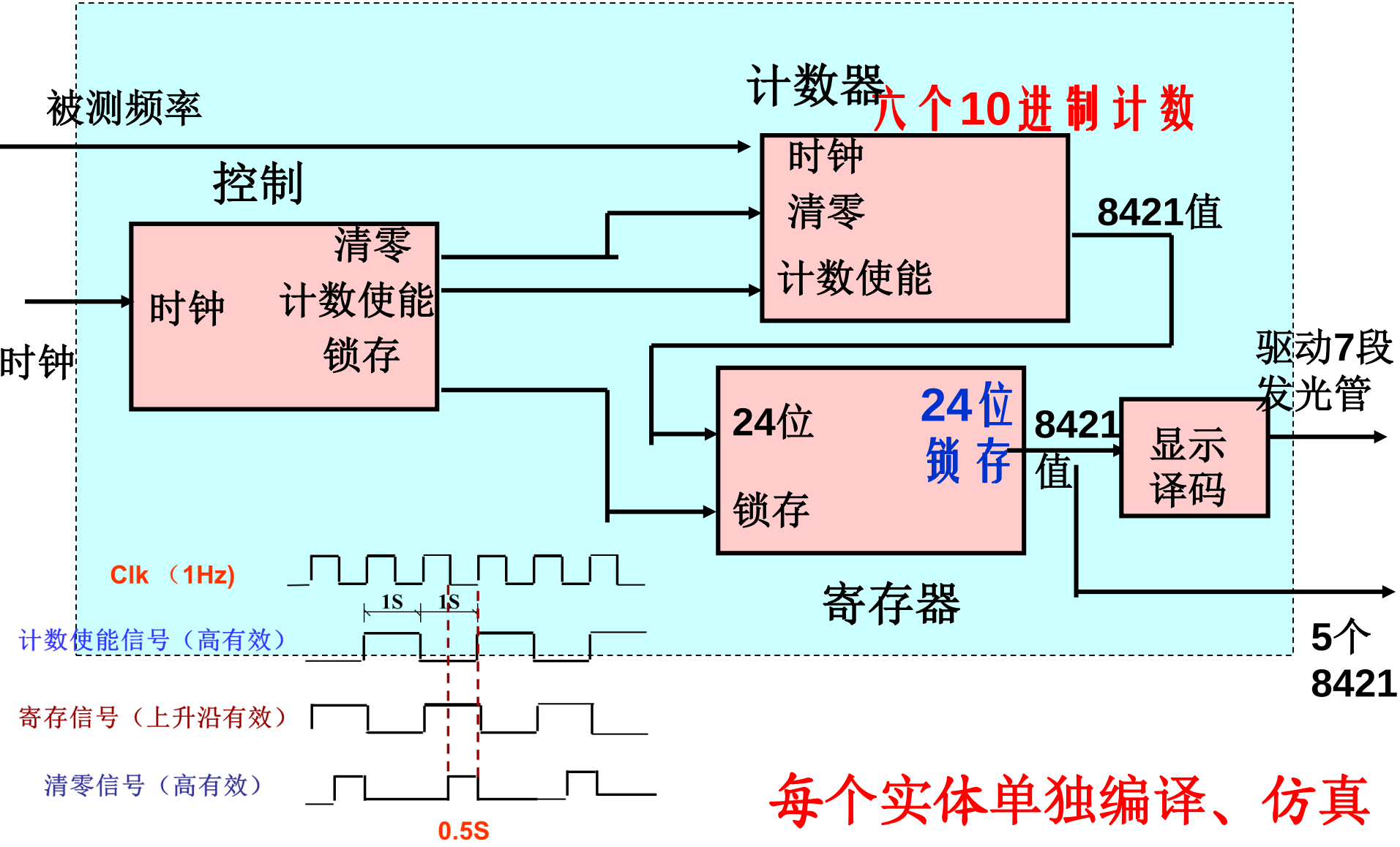
数码管高低位的接线。



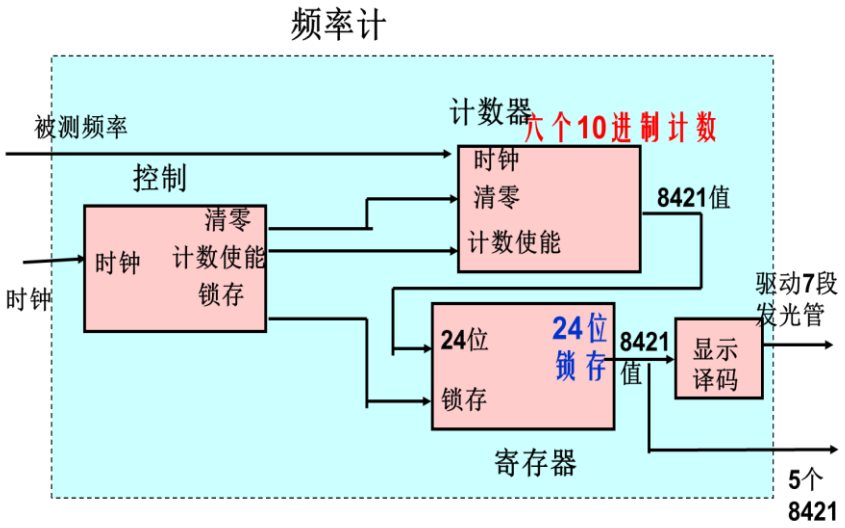
Clk （1Hz)



频率计

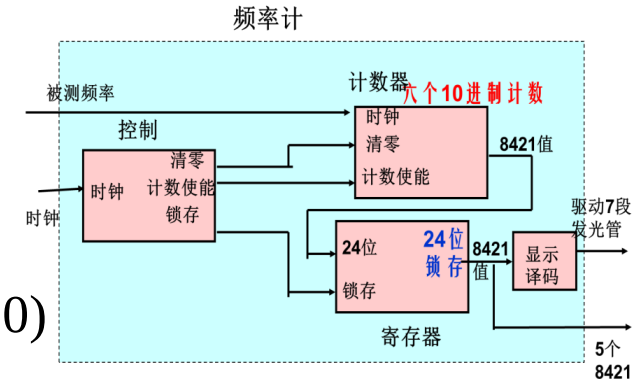


每个实体单独编译、仿真

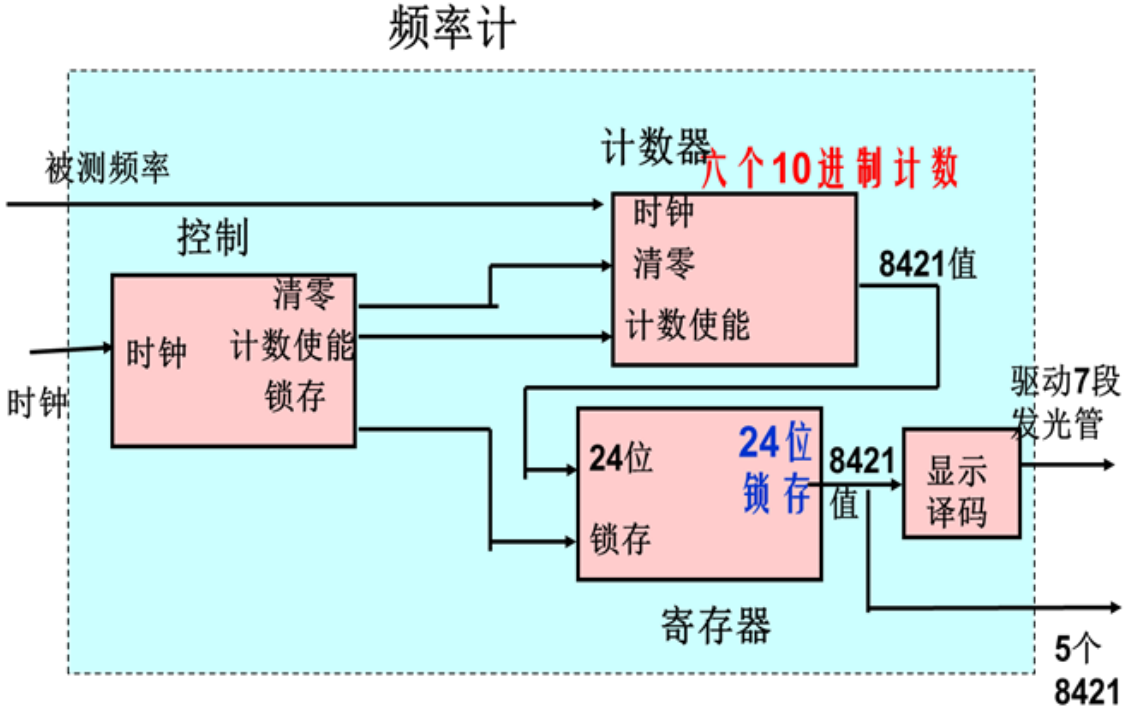



```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

ENTITY Data_fre_count IS
PORT(clk1:IN STD_LOGIC;  --待测
    CLK: IN STD_LOGIC;  --5KHz
    ledout1: OUT STD_LOGIC_VECTOR(6 DOWNTO 0);  --驱动7段发光管
    ledout2: OUT STD_LOGIC_VECTOR(19 DOWNTO 0)    -- 5个7段显示译码器
    );
END Data_fre_count;
ARCHITECTURE func OF Data_fre_count IS
 COMPONENT divide1
     PORT(cp_50:IN STD_LOGIC;
             cp1:OUT STD_LOGIC;   --使能
             cp2:OUT STD_LOGIC;   --寄存
             cp3:OUT STD_LOGIC    --清零
           );
 END COMPONENT;
```

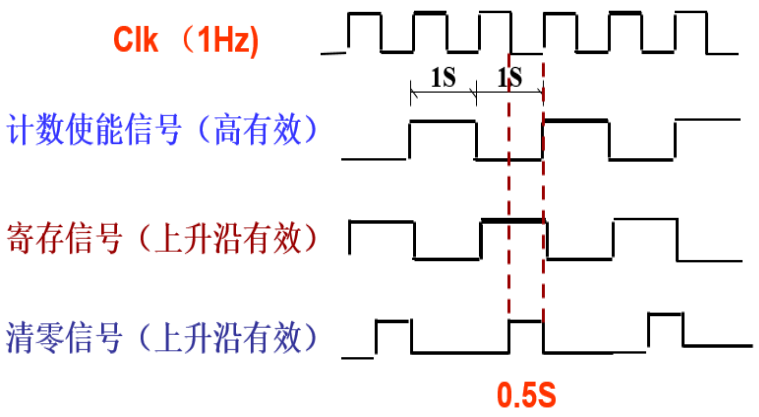
```
COMPONENT count10_6
    PORT(CLK:IN STD_LOGIC;
           EN: IN STD_LOGIC;
          CLR:IN STD_LOGIC;
        d:OUT STD_LOGIC_VECTOR(23 DOWNTO 0)
             );
 END COMPONENT;

 COMPONENT latch6
  PORT(lock:IN STD_LOGIC;
         qin: IN STD_LOGIC_VECTOR(23 DOWNTO 0);
         qout:OUT STD_LOGIC_VECTOR(23 DOWNTO 0)
      );
 END COMPONENT;

 COMPONENT display
         PORT(   pin:IN STD_LOGIC_VECTOR(3 DOWNTO 0);
                  pout:OUT STD_LOGIC_VECTOR(6 DOWNTO 0)
                );
 END COMPONENT;
```

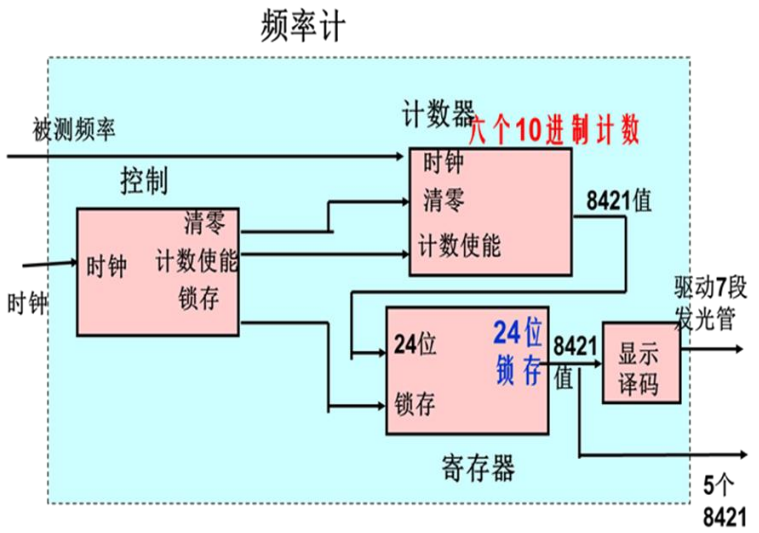


```
 SIGNAL clr_1:STD_LOGIC;
SIGNAL c_gate:STD_LOGIC;
SIGNAL lock_1:STD_LOGIC;
SIGNAL g,h:STD_LOGIC_VECTOR(23 DOWNTO 0);
  SIGNAL leds:STD_LOGIC_VECTOR(6 DOWNTO 0);  --7段
BEGIN
  u1:divide1 PORT MAP(CLK, c_gate, lock_1, clr_1);
  u2:count10_6 PORT MAP(clk1, c_gate, clr_1, g);
  u3:latch6 PORT MAP(lock_1, g, h);
  u4:display PORT MAP(h(23 DOWNTO 20), leds);
  ledout1<=leds;
  ledout2<=h(19 DOWNTO 0);
END func;
```

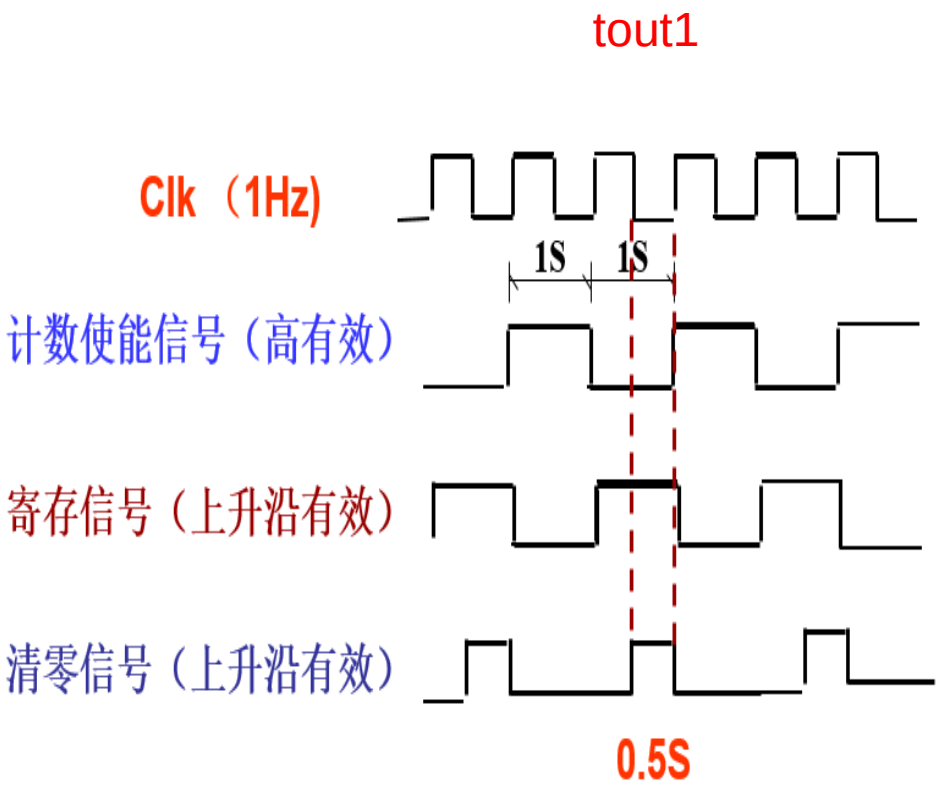


```
LIBRARY IEEE;   --分频及控制模块
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

ENTITY divide1 IS
  PORT(cp_50:IN STD_LOGIC;
       cp1:OUT STD_LOGIC;   -- 0.5Hz使能
       cp2:OUT STD_LOGIC;  --寄存
       cp3:OUT STD_LOGIC  --清零
       );
END divide1;

ARCHITECTURE func OF divide1 IS
  SIGNAL temp: INTEGER RANGE 0 TO 5000;
  SIGNAL tout1:STD_LOGIC;   --1HZ方波
  SIGNAL tout2:STD_LOGIC;
```

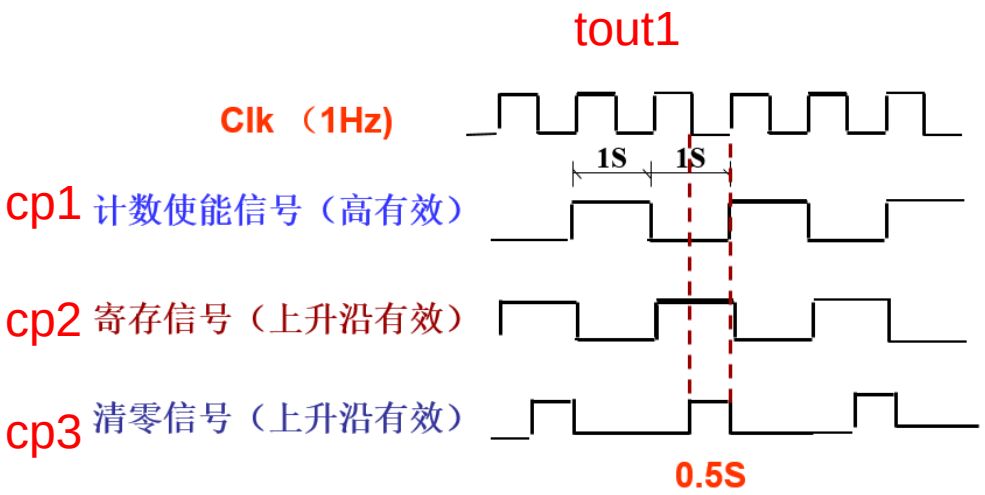




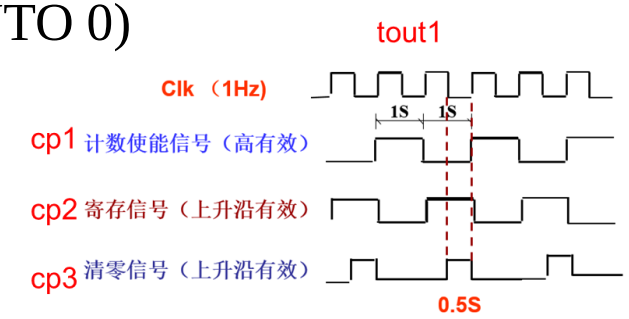
```
BEGIN
 p1:PROCESS(cp_50)                    --5000分频
    BEGIN
      IF(cp_50'event AND cp_50='1')THEN
        IF temp=4999 THEN
          temp <= 0;
        else
          temp <= temp+1;
        END IF;
      END IF;
END PROCESS;

 p2:PROCESS(cp_50,temp)
   BEGIN
      IF(cp_50'event AND cp_50='1')THEN
        if( temp<2500)    then
              tout1 <= '1' ; --1Hz方波
         else
              tout1 <= '0';
          end if;
      END IF;
   END PROCESS;
```

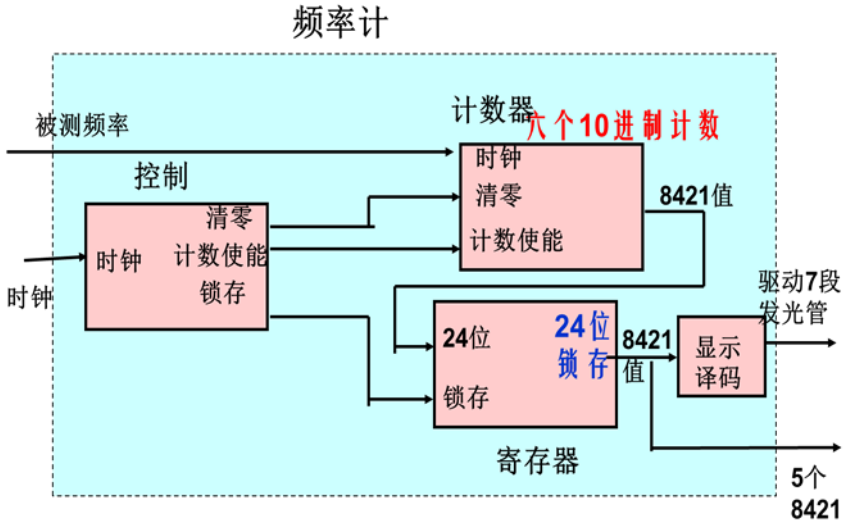


```
p3:PROCESS(tout1)    --2分频，0.5Hz方波
   VARIABLE t:STD_LOGIC;
  BEGIN
     IF(tout1‘event AND tout1=’1‘)THEN    --作为时钟信号、计数使能、打入脉冲，不能有竞争冒险
          t := NOT t;
       IF(t='1')THEN
         cp1<=‘1’;       --计数使能，高电平
         cp2<=‘0’;       --寄存，上升沿
       ELSE
         cp1<='0';
         cp2<='1';
       END IF;
     END IF;
  END PROCESS;
P4:PROCESS(tout1,cp1)
  BEGIN
    IF(tout1=‘0’AND cp1=‘0’)THEN
      cp3<=‘1’;    --清零
    ELSE
      cp3<='0';
    END IF;
  END PROCESS;    END func;
```



```
LIBRARY IEEE;   --24位锁存器模块
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY latch6 IS
PORT(  lock:IN STD_LOGIC;
            qin: IN STD_LOGIC_VECTOR(23 DOWNTO 0);
            qout:OUT STD_LOGIC_VECTOR(23 DOWNTO 0)
    );
END latch6;

ARCHITECTURE func OF latch6 IS
  SIGNAL temp:STD_LOGIC_VECTOR(23 DOWNTO 0);
BEGIN
   PROCESS(lock)
   BEGIN
            IF lock'event AND lock='1' THEN
                     temp<=qin;
            END IF;
            qout<=temp;
   END PROCESS;
END func;
```

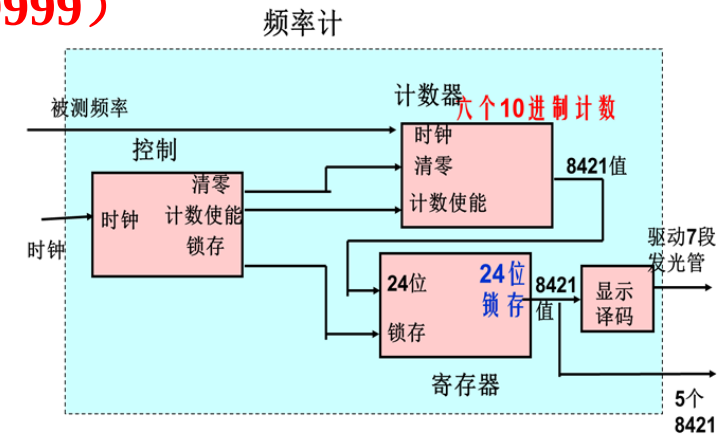




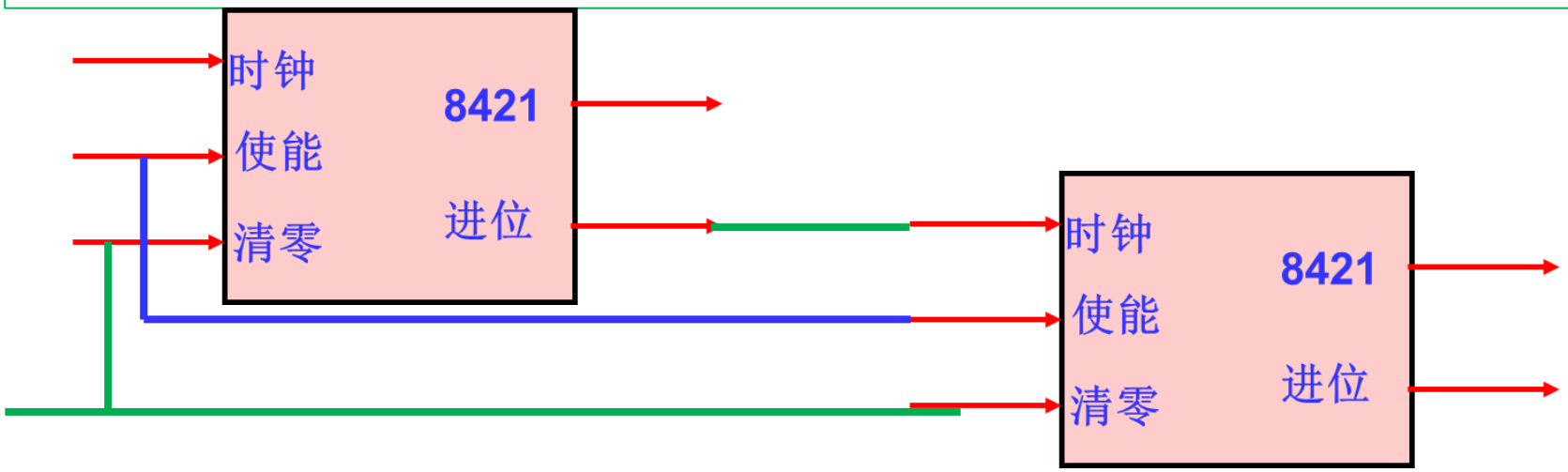
```
LIBRARY IEEE;  --10进制计数器 0--9
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

ENTITY count10 IS
  PORT(CLK:IN STD_LOGIC;
       EN: IN STD_LOGIC;
       CLR:IN STD_LOGIC;
       qout:OUT STD_LOGIC_VECTOR(3 DOWNTO 0);
       cout:OUT STD_LOGIC
        );
END count10

ARCHITECTURE  func OF count10 IS
  SIGNAL temp:STD_LOGIC_VECTOR(3 DOWNTO 0);
BEGIN
```

```
BEGIN
   PROCESS(CLK,EN,CLR)
   BEGIN
    IF(CLR='1')THEN
      temp<=“0000”;   --异步清零
    ELSIF(EN=‘1’)THEN   --计数使能
      IF (CLK'event AND CLK='1')THEN
          IF(temp="1001")THEN
              temp<="0000";
              cout<=‘1 ；  --寄存的进位输出，给高一级提供计数时钟
           ELSE
              temp<=temp+1;
              cout <= ‘0’;
           END IF;
        END IF;
    END IF;
   END PROCESS;
     qout<=temp;
END func;
```

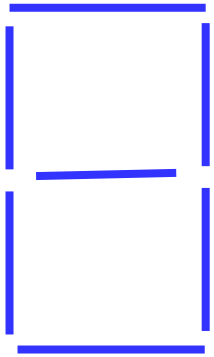
```
LIBRARY IEEE; --6个10进制计数器（000000~999999）
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

ENTITY count10_6 IS
  PORT(CLK:IN STD_LOGIC;
               EN: IN STD_LOGIC;
               CLR:IN STD_LOGIC;
               d:OUT STD_LOGIC_VECTOR(23 DOWNTO 0)
                 );
END count10_6;

ARCHITECTURE func OF count10_6 IS
  COMPONENT count10
           PORT(CLK:IN STD_LOGIC;
                        EN: IN STD_LOGIC;
                        CLR:IN STD_LOGIC;
                        qout:OUT STD_LOGIC_VECTOR(3 DOWNTO 0);
                        cout:OUT STD_LOGIC
                        );
  END COMPONENT;
```



```
SIGNAL c:STD_LOGIC_VECTOR(6 DOWNTO 0);
BEGIN
  u1:count10 PORT MAP(CLK, EN,CLR, d(3 DOWNTO 0),c(0));
  u2:count10 PORT MAP(c(0), EN,CLR, d(7 DOWNTO 4),c(1));
  u3:count10 PORT MAP(c(1), EN,CLR, d(11 DOWNTO 8),c(2));
  u4:count10 PORT MAP(c(2), EN,CLR, d(15 DOWNTO 12),c(3));
  u5:count10 PORT MAP(c(3), EN,CLR, d(19 DOWNTO 16),c(4));
  u6:count10 PORT MAP(c(4), EN,CLR, d(23 DOWNTO 20),c(5));
END func;
```

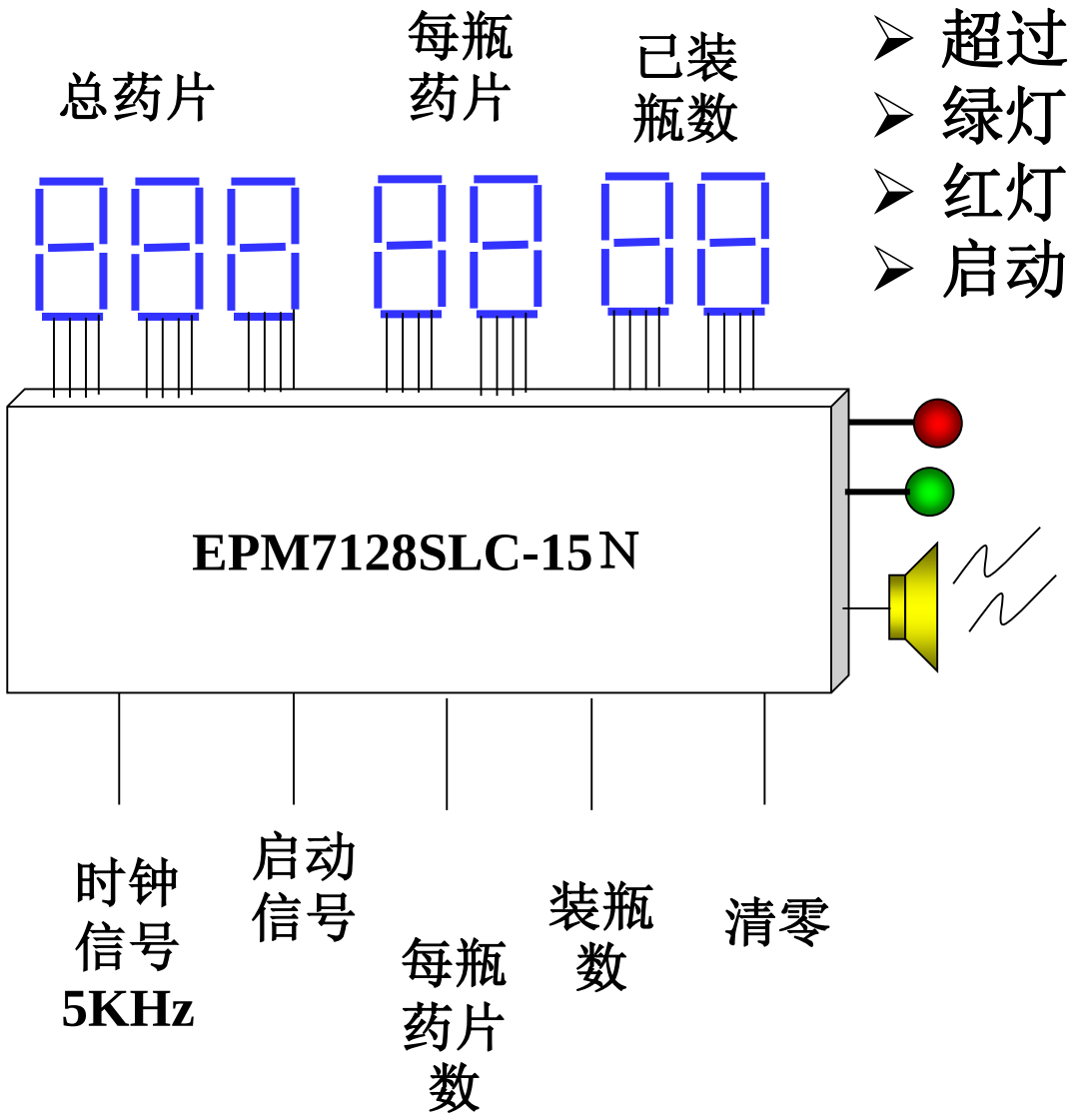


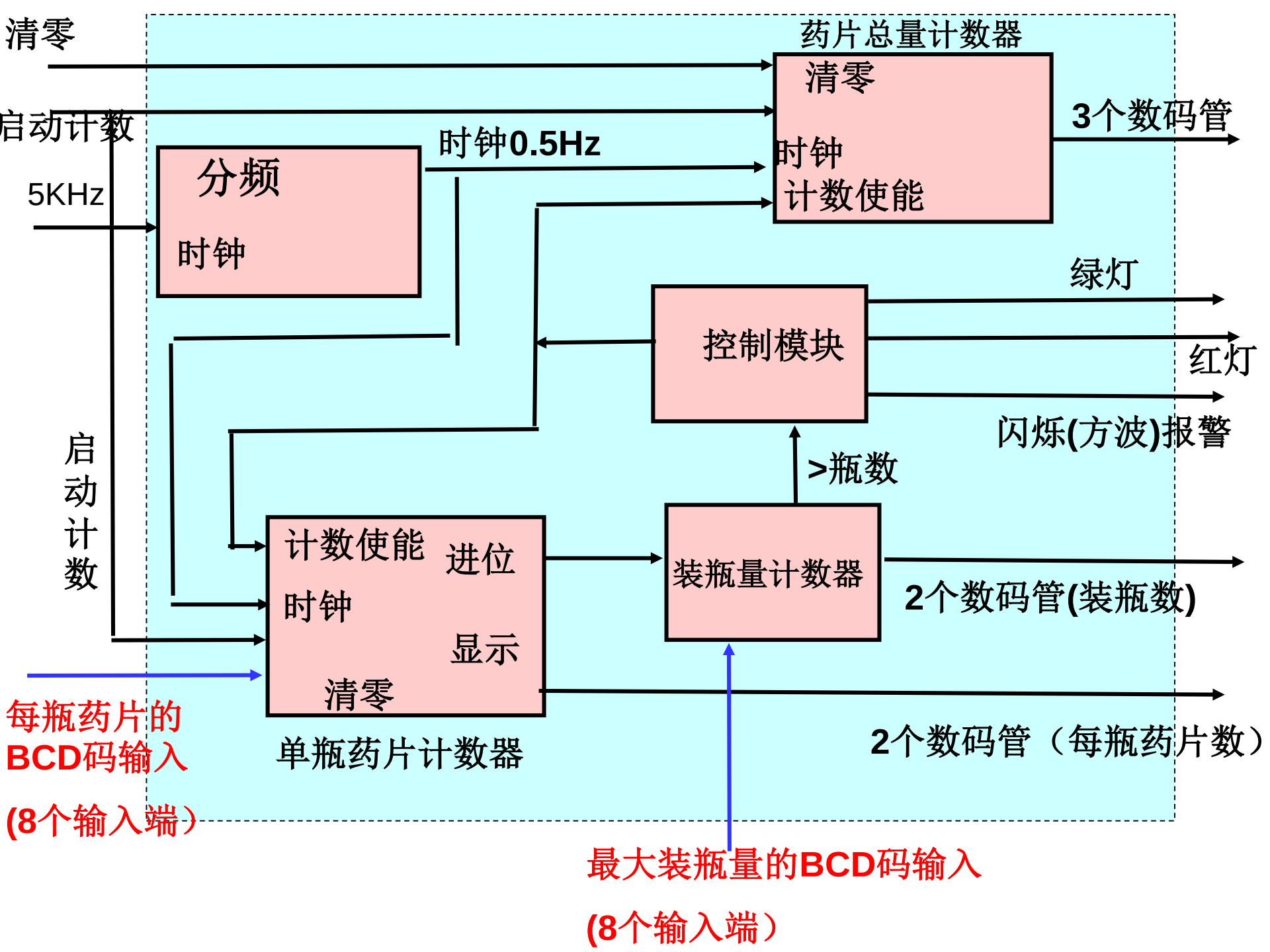
```
LIBRARY IEEE;   --7段显示译码
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

ENTITY display IS
PORT(pin:IN STD_LOGIC_VECTOR(3 DOWNTO 0);
    pout:OUT STD_LOGIC_VECTOR(6 DOWNTO 0)   );
END display;

ARCHITECTURE func OF display IS
BEGIN
        process(pin)
         begin
          case pin is
             when "0000"  => pout<="0111111" ;
             ween  "0001" => pout<= "0000110”;
             when "0010"  => pout<="1011011" ;
             when "0011"  => pout<="1001111" ;
             when "0100"  => pout<="1100110" ;
             when "0101"  => pout<="1101101" ;
             when "0110"  => pout<="1111101" ;
             when "0111"  => pout<="0000111" ;
             when “1000"  => pout<="1111111" ;
             when “1001"  => pout<="1101111" ;
          end case ;
       end process;
END func;
```



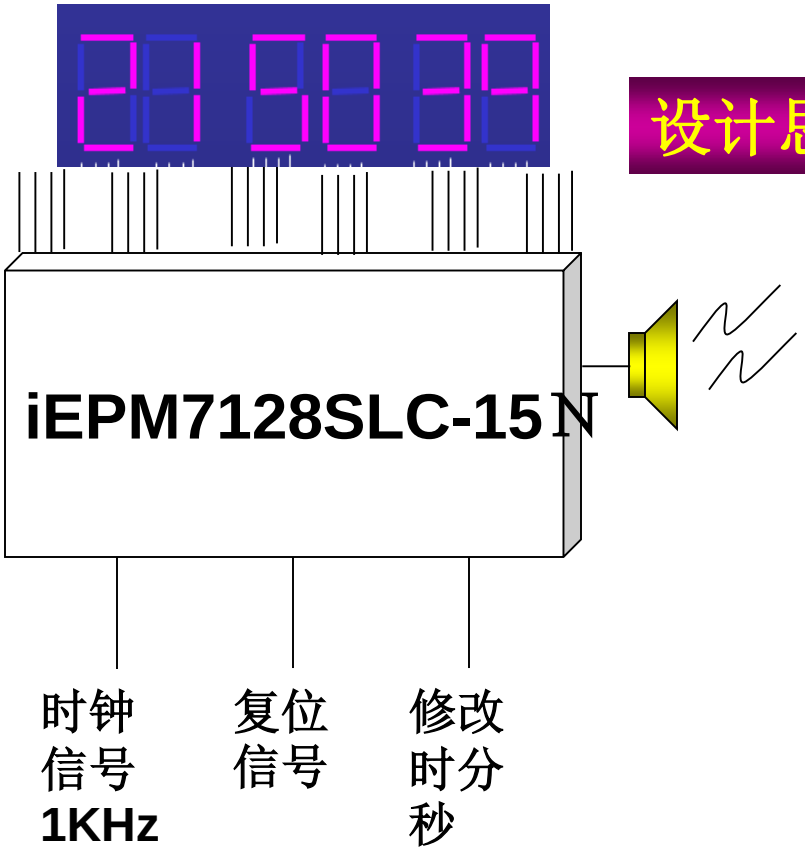
药片装瓶系统

- BCD码每瓶装药数输入，要求每瓶最大药片数50粒，最多装18瓶。
- 药片：0.5Hz方波
- 超过设定瓶数报警。
- 绿灯：正在装瓶。
- 红灯：装瓶停止。
- 启动：=0设置两个初值；=1装瓶




清零
药片总量计数器
清零
启动计数
3个数码管
时钟0.5Hz
分频
5KHz
时钟
时钟
计数使能
绿灯
控制模块
红灯
闪烁(方波)报警
启动计数
>瓶数
计数使能
进位
时钟
装瓶量计数器
2个数码管(装瓶数)
显示
清零
每瓶药片的BCD码输入
(8个输入端）
单瓶药片计数器
2个数码管（每瓶药片数）
最大装瓶量的BCD码输入
(8个输入端）

电子钟显示



设计思想

顶层模块(clock)

底层模块(clk_ring)　提供1Hz和整点响铃脉冲

底层模块(cnt60)　60进制计数器

底层模块(cnt24)　24进制计数器

底层模块(encode24)　2-4译码器

底层模块(ring)　整点响铃信号

（1）时钟各组成计数器的设计。最基本的为1s 计时器，通过对输入脉冲5KHZ时钟信号二次分频得到。其次由于时钟的分和秒为60 进制，小时为24 进制，故设置60 计数器counter_60和24 进制的计数器counter_24。

（2）元件连接模块。分频得到的1s计时器的时钟脉冲sec给控制秒的60进制计数器，控制秒的60进制计数器进位c0给控制分钟的60进制计数器，进而进位给24进制计数器，并将得到的相应的时、分、秒信号与对应的输出d1、d10相连（最高位为7段译码），最后通过数码管显示。

（3）控制信号。首先是清零信号，采用了电平的有效方式。其次是调表控制信号，采用信号change上升沿触发的方式改变修改模式，并用相同方式对时钟修改，模式mode采用了 2—4译码器：“00”为正常工作状态，“01”状态下对小时进行修改，“10”状态下对分钟进行修改，“11”状态下对秒进行修改。

（4）完成报时功能，通过分钟向小时的进位来产生响铃的控制信号，此信号持续一分钟，将响铃信号控制在四秒，前三秒为低音C后一秒为高音C。此外，为了达到到特定的响铃效果，响铃进程采用50KHZ的时钟信号控制响铃的音调，并将其二分频以提高响度。

```
PROCESS(clk,clr)    --24进制计数器
    BEGIN
     IF(clr='0')THEN
       temp1<="0000";
       temp10<=“0000";
     ELSIF(clk'event AND clk='1')THEN
       IF(temp1=3 AND temp10=2)THEN
         temp1<="0000";
         temp10<=“0000";
       ELSIF(temp1=9)THEN
         temp1<="0000";
         temp10<=temp10+1;
       ELSE
         temp1<=temp1+1;
       END IF;
     END IF;
    END PROCESS;
```

```
ROCESS(clk,clr)   --60进制计数器
  BEGIN
    IF(clr='0')THEN
       temp1<="0000";
       temp10<="0000";
    ELSIF(clk'event AND clk='1')THEN
      IF(temp1=9)THEN
        temp1<="0000";
        IF(temp10=5)THEN
          temp10<="0000";
          co_temp<=‘1’;      --给高一级的时钟
        ELSE
          temp10<=temp10+1;
          co_temp<='0';
        END IF;
      ELSE
        temp1<=temp1+1;
        co_temp<='0';
      END IF;
    END IF;
  END PROCESS;
```