EDA 实验四 设计数字钟

计算机与信息工程学院 2012 级电子信息工程

学号： 2012221105200179

实验人： 曹繁 同组人： 李胜宇

实验日期：2014 年 11 月 23 日

一、数字钟设计要求

1、在 FPGA 上利用 8 个 LED 数码管正常显示时间，显示时间格式为：CFHHMMSS，其中 C、F 两位常亮，用以标识数字钟的开发型号，HH、MM、SS 分别为小时、分钟、秒的十位与个位。

2、利用拨动开关 S1 到 S5，开关拨动均为低电平有效。S1 开关调整当前显示时间的小时位，自增 1；S2 调整当前时间的分钟位，自增 1；S3 具有复位功能，有效时当前时间复位到 000000 状态；S4 具有计时模式调整功能，有效时，计时模式为 24 小时制，无效时为 12 小时制；S5 具有预置时间常数功能，便于整点报时功能的进行。此外，总体上 S1 到 S5 这五个开关的优先级（从高到低排列）顺序为：S5>S3>S1>S4>S2，这个优先级限制会在不同的时间段被打破，比如置数开关 S5 与调时开关 S1 可以并行作用，即 S4=S1 可以成立，同样的，S4=S1、S4=S2、S4=S3、S4=S5 也是可以成立的，但是有一点，调时开关 S1 会制约调分开关 S2，复位开关 S3 会制约 S1、S2，预置开关 S5 会制约 S3。这些开关的优先级灵活性便于实现用户对数字钟的多种操作，在设计上体现全面性。

3、整点报时功能，在时间的分钟位与秒钟位均为 59 时启动报时提醒。为使实验现象更加明显，在 59 分 50 秒就启动报时提醒，报时提醒操作为 LED 灯闪烁和蜂鸣器鸣叫。

二、数字钟设计原理

1、数字钟原理总述。FPGA 开发板上的时钟信号频率为 50 MHz，通过分频器元件提取出我们需要的计时时钟信号，为1Hz；LED 灯闪烁整点报时驱动信号，为 10Hz；蜂鸣器鸣叫整点报时驱动信号，为200Hz；LED 数码管扫描时钟信号，为1KHz。数字钟在设计上主要分为分频模块、调时模块、计时模块、整点报时模块、输出显示模块。

2、数字钟各模块实现原理

2.1 分频原理

利用分频公式：

$$F_{OUT}=\frac{F_{IN}}{2\times(N+1)},\text{其中}N\text{为分频值} \qquad \textbf{（公式 1）}$$

2.2 调时原理

调整时间模块在计时时钟信号（1Hz）下正常工作，调整时间模块中设置接收来自开关 S1~S5 的输入端口，根据调时控制开关 S1（调时）、S2（调分）、S3

（复位）、S4（调整计时模式）、S5（预置时间）是否有效进行相应的逻辑判断，逻辑判断的依据应根据设计要求上开关在一段时间内的优先级进行，例如在 P2 模块中，先对 S1、S2、S3 进行有效性检验，再对调整时间模块中 9 变为 0、59 变为 00、11 变为 0、23 变为 0 等逻辑判断与操作，这一步骤可用 IF...ELSIF...ELSE...END IF 语句实现。值得注意的是，调时与计时模块应用一个状态控制变量区分开来，调时模块工作时，计时模块应一直保持在当前状态（计时时间值不能改变）；调时模块不工作，计时模块才继续运行；调时模块结束，计时模块的当前值应变为调时后的时间值。S4、S5 主要是传递开关的状态，它们的作用在 P3 模块体现。

2.3 计时原理

计时模块在计时时钟信号（1*Hz*）下正常工作，每来一个上升沿，计时秒值的个位，当秒值为 59 时，下一状态切换为 00，且分钟值个位加 1；在分钟值刚为 59 的那个时钟周期过后，再经过 60 个时钟信号周期，分钟值应切换为 00，小时值个位加 1；小时值的切换应受开关 S4 控制，S4 有效时（即为低电平），为 24 小时制，在小时值刚为 23 的那个时钟周期过后，再经过 3600 个时钟信号周期，时钟值应切换为 00；同理，S4 无效是，小时值在 11 时过后再切换为 00。在 P3 整个过程中，计时模块 P3 是受调时模块 P2 控制的，S5（预置时间开关）具有最高优先权。

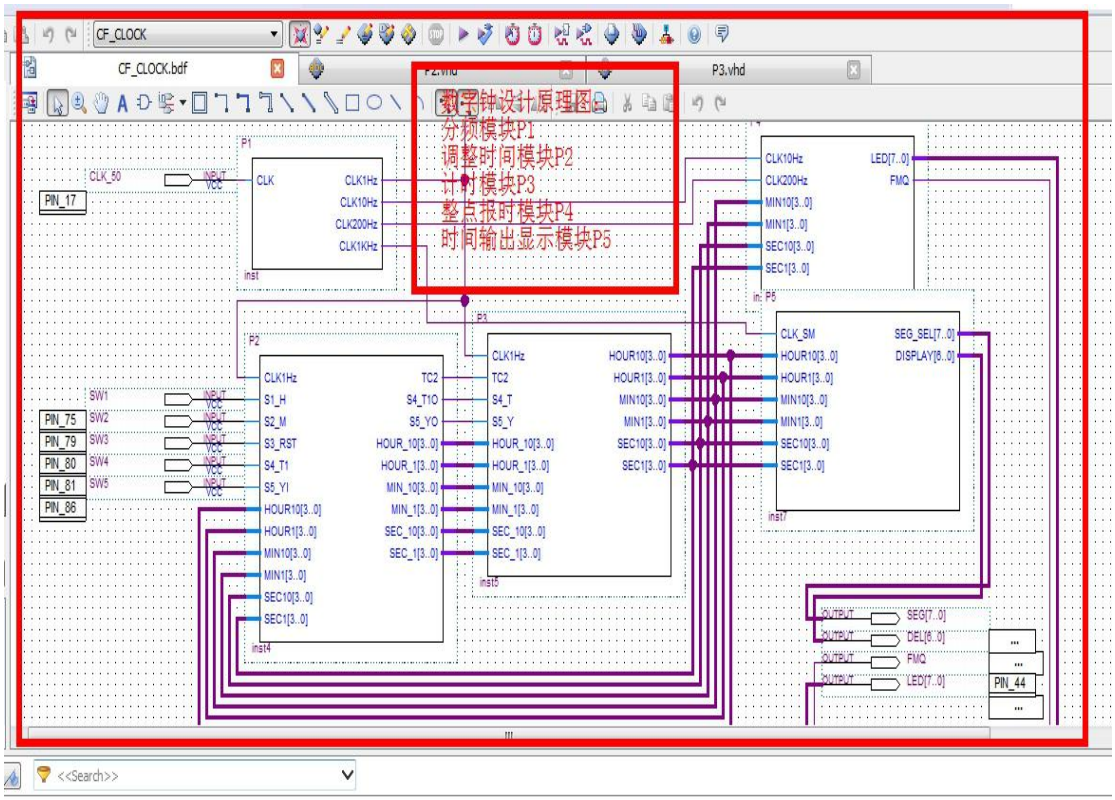
2.4 整点报时原理

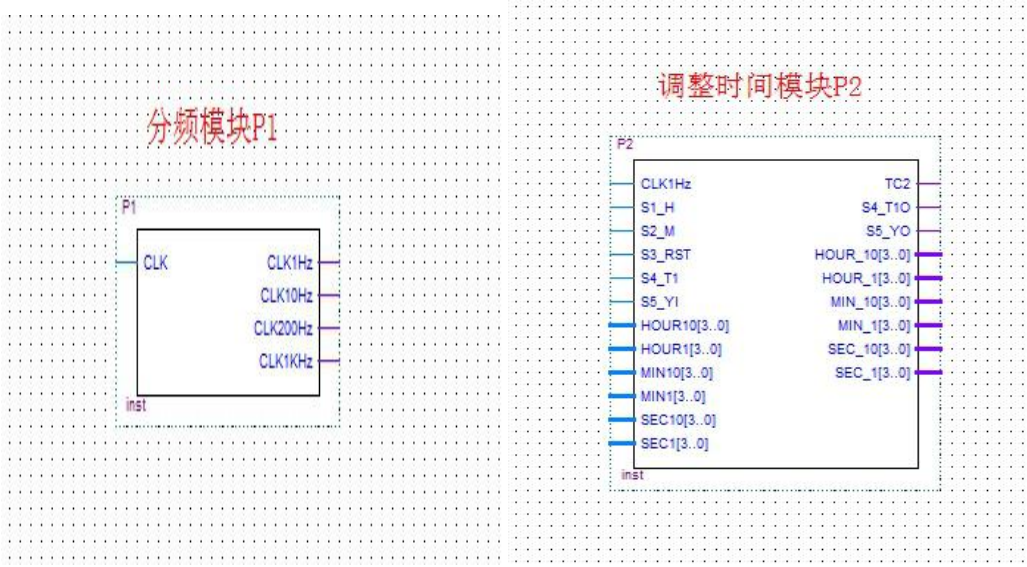
整点报时功能主要由 8 个 LED 灯闪烁和蜂鸣器鸣叫体现出来，LED 灯在 LED 灯闪烁驱动信号（10*Hz*）下正常工作，蜂鸣器在蜂鸣器驱动信号（200*Hz*）下正常工作。LED 灯和蜂鸣器正常进行整点报时工作在分钟值为 59，秒钟值为 50 开始，进行 10 秒的报时工作，即结束时间为 HH5959，在这 10 秒内，应该出现 8 个 LED 灯闪烁循环闪烁，蜂鸣器长鸣的现象。

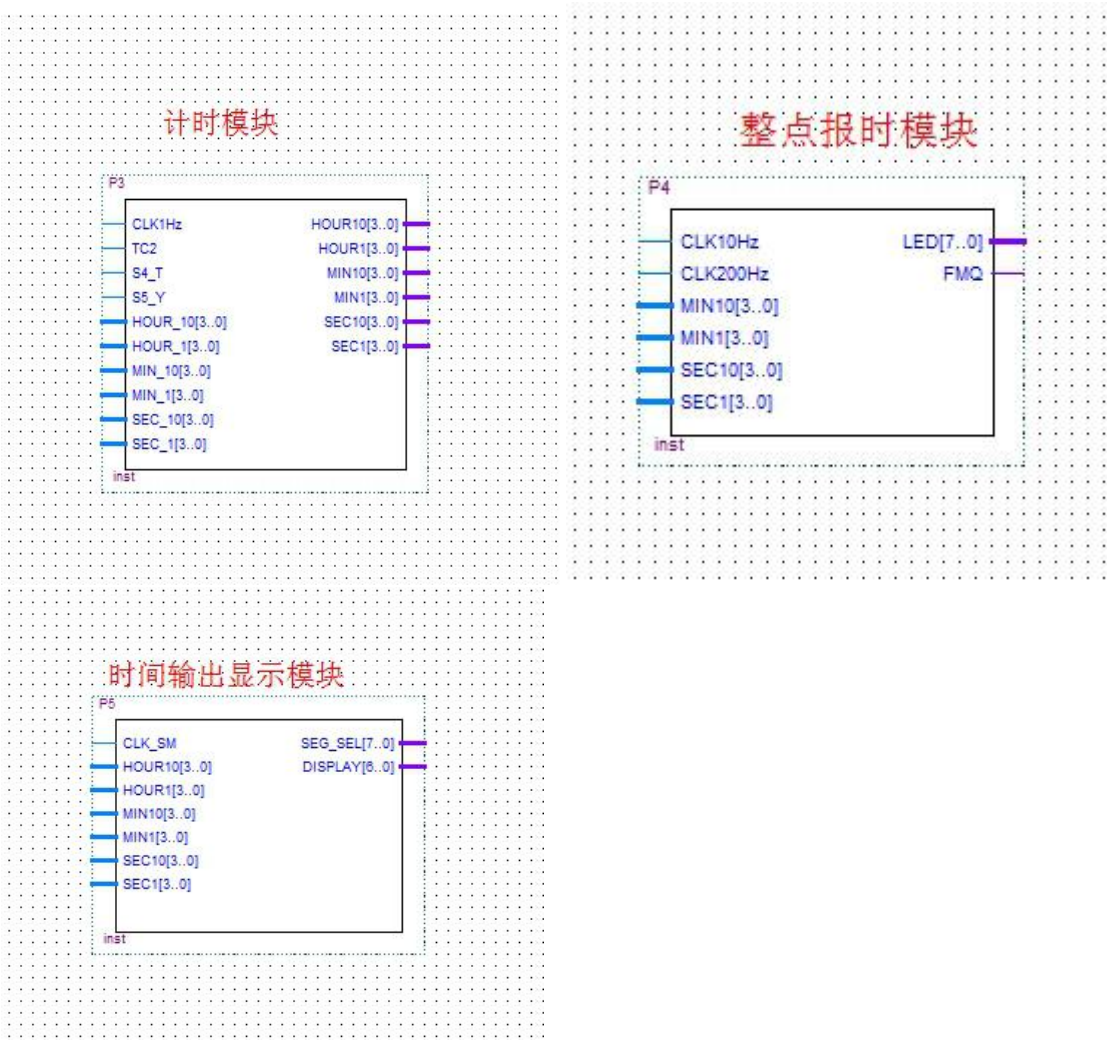
2.5 时间输出显示原理

时间输出显示模块在计时时钟信号（1*Hz*）和数码管扫描时钟信号共同作用下正常工作，其时间信号由计时模块 P3 传入，LED 数码管位选工作采用**移位运算**进行，段选工作采用 **CASE 分支语句**进行，先选中 LED1 数码管，输出在该数码管上要显示的数字 C，再选中 LED2 数码管，输出 F，再依次选中 LED3 到 LED8 数码管，分别按顺序输出“HHMMSS”（时间值示意）。由于数码管扫描脉冲是计时脉冲的 1000 倍，在间隔时间 1 秒内，8 个 LED 数码管的显示频率为 125 左右，人眼看起来，LED 数码管上的数字是一起显示的，从而实现了时间输出显示功能。

三、数字钟程序设计


CF_CLOCK
CF_CLOCK.bdf
P2.vhd
P3.vhd
数字钟设计原理图
分频模块P1
调整时间模块P2
计时模块P3
整点报时模块P4
时间输出显示模块P5
P1
CLK
CLK1Hz
CLK10Hz
CLK200Hz
CLK1KHz
inst
P2
P3
P5
inst4
inst5
inst7
CLK_50
PIN_17
PIN_75
PIN_79
PIN_80
PIN_81
PIN_86
PIN_44
SW1
SW2
SW3
SW4
SW5
VCC
S1_H
S2_M
S3_RST
S4_T1
S5_YI
TC2
S4_T1O
S5_YO
S4_T
S5_Y
HOUR10[3..0]
HOUR1[3..0]
MIN10[3..0]
MIN1[3..0]
SEC10[3..0]
SEC1[3..0]
HOUR_10[3..0]
HOUR_1[3..0]
MIN_10[3..0]
MIN_1[3..0]
SEC_10[3..0]
SEC_1[3..0]
LED[7..0]
FMQ
CLK_SM
SEG_SEL[7..0]
DISPLAY[8..0]
OUTPUT
SEG[7..0]
DEL[6..0]
<<Search>>


分频模块P1
P1
CLK
CLK1Hz
CLK10Hz
CLK200Hz
CLK1KHz
inst
调整时间模块P2
P2
CLK1Hz
S1_H
S2_M
S3_RST
S4_T1
S5_YI
HOUR10[3..0]
HOUR1[3..0]
MIN10[3..0]
MIN1[3..0]
SEC10[3..0]
SEC1[3..0]
TC2
S4_T1O
S5_YO
HOUR_10[3..0]
HOUR_1[3..0]
MIN_10[3..0]
MIN_1[3..0]
SEC_10[3..0]
SEC_1[3..0]
inst



以上截图为数字钟设计总原理图和各模块器件设计图。程序代码见附录。

四、数字钟程序调试

1、在 S1、S4 开关均有效，计时模式 24 小时制变为 12 小时制时，如果时间值大于 11，那么就出现数字钟“跑飞”现象，时间值一直加 1，逢 12 置 0 约束条件失效。在分析调整时间模块 P2 内的逻辑判断代码后，发现正是由于判断条件的不完善导致的，改进代码，在 S1、S4 开关有效条件下，先对时钟值进行判断，是否大于 11，如果大于 11，就将其转换为 12 小时制下的时钟值，再将时钟个位值加 1，其算法为：

$$HOUR\_10 = (HOUR\_10 \times 10 + HOUR\_1) \div 10 \quad (\text{取商})$$
$$HOUR\_1 = (HOUR\_10 \times 10 + HOUR\_1) \ \ MOD \ \ 10 \quad (\text{取余数}) \qquad \text{（公式 2）}$$

2、LED 灯闪烁循环闪烁时有一位总是先灭后亮，破坏了循环的连贯性（在驱动信号频率较低情况下），检查模块 P4 代码，在对 LED 灯进行位选时忽略了极值判断条件，使得 LED 灯控制字出现了“FFH”，即全灭现象，因而在硬件试验中出现循环显示被破坏的感觉。修正极值判断条件，计数变量 CNT_LED 先增

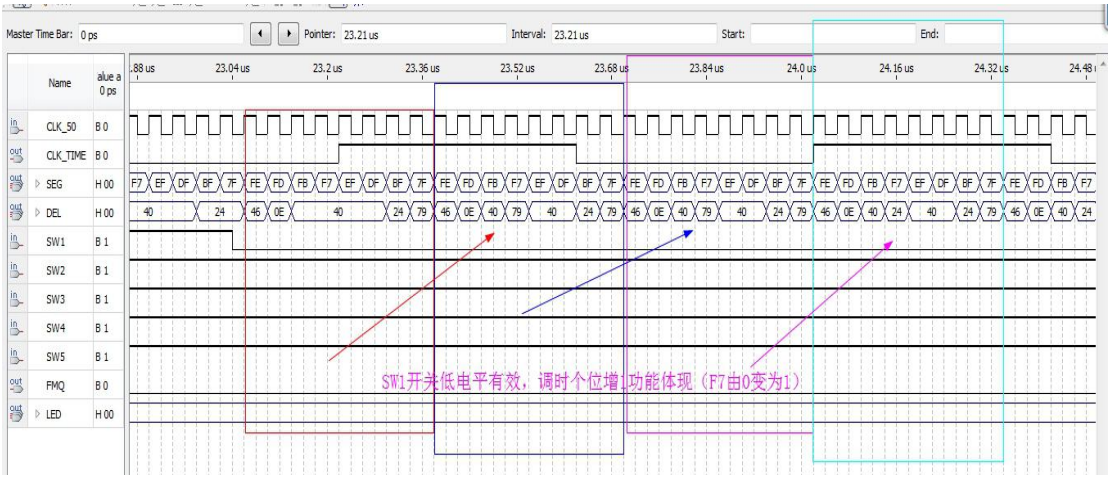
1，再判断是否为 7，使得所有情况限定为 0~7 共 8 个状态，解决了之前的 LED 灯循环显示 BUG。

五、数字钟功能仿真与分析

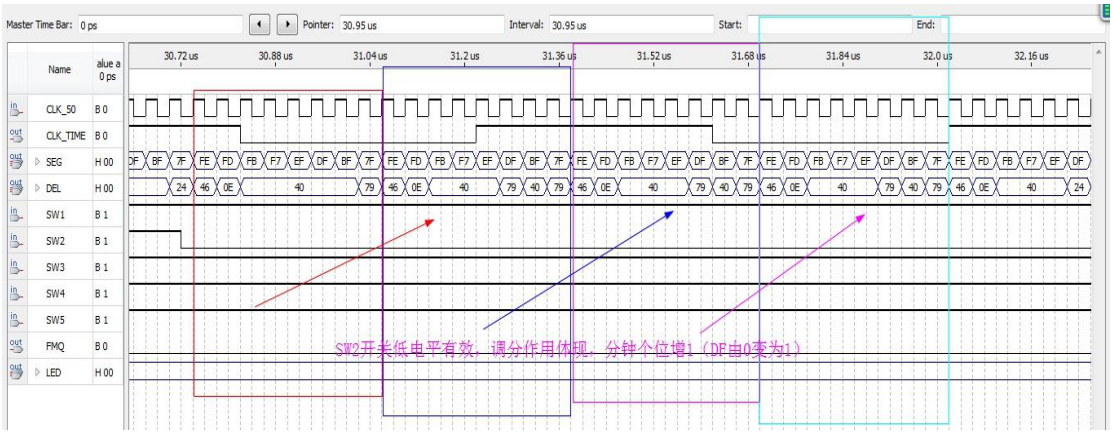
仿真说明：由于 QUARTUS II 13.0 版本对仿真时间的限制，最长为 100 μs ，因而本次进行数字钟功能仿真时将时钟信号频率设置为 25MHz ，计时脉冲信号频率为1.25MHz ，扫描脉冲、LED 报时脉冲、蜂鸣器报时脉冲频率均为 25MHz 。计时数码管 LED1~LED8 显示时段选和位选均为低电平有效，单独控制一个 LED 数码管对应的控制字节、在 LED1~LED8 上显示数字 0~9、C、F 对应的段选控制字列表如下：

数码管编号	**LED1**	**LED2**	**LED3**	**LED4**	**LED5**	**LED6**	**LED7**	**LED8**
位选控制字	**FEH**	**FDH**	**FBH**	**F7H**	**EFH**	**DFH**	**BFH**	**7FH**
显示内容	**“C”**	**“F”**	**小时十位**	**小时个位**	**分钟十位**	**分钟个位**	**秒十位**	**秒个位**

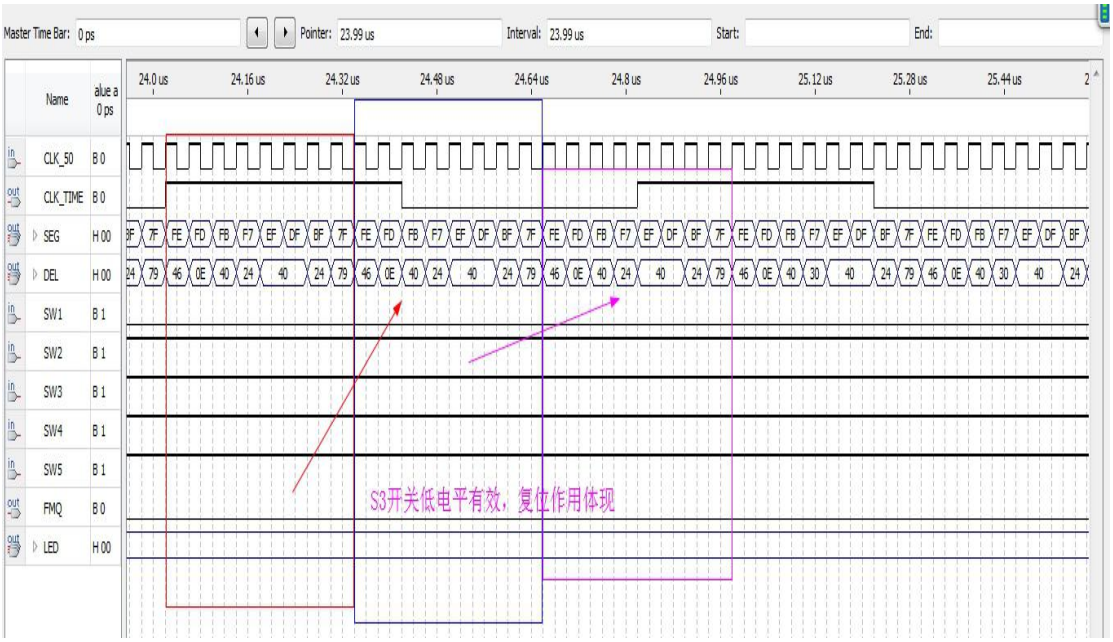
显示数字	0	1	2	3	4	5	6	7	8	9	C	F
段选控制字	40H	79H	24H	30H	19H	12H	02H	78H	00H	10H	46H	0EH



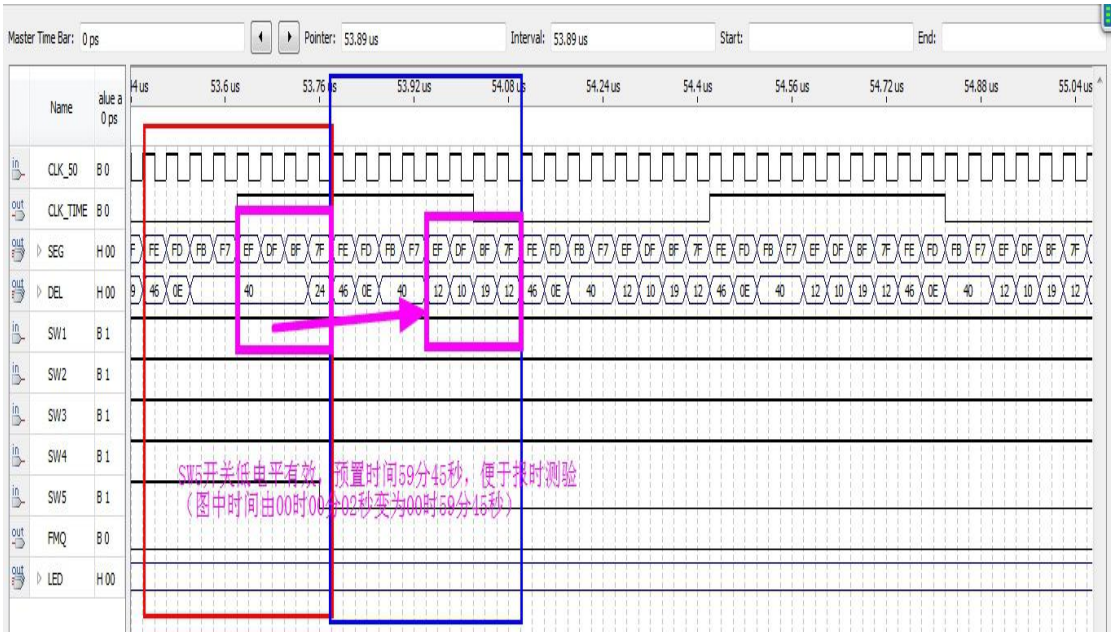
开关 S1 单独工作且有效时能正确实现时钟值个位加 1。



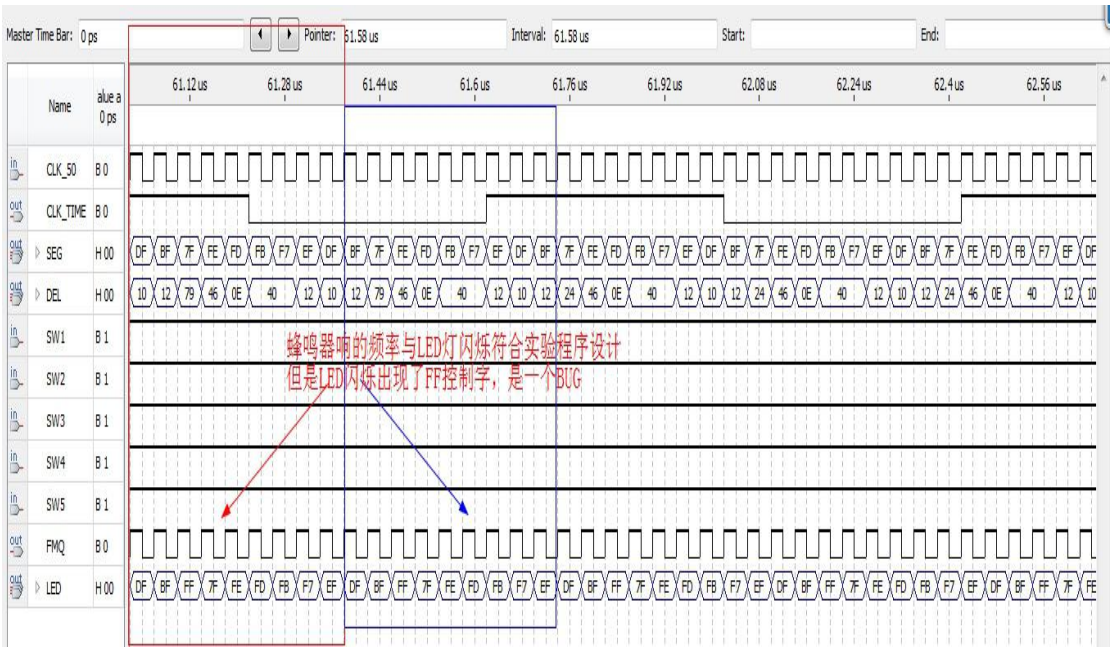
开关 S2 单独工作且有效时能正确实现分钟值个位加 1。



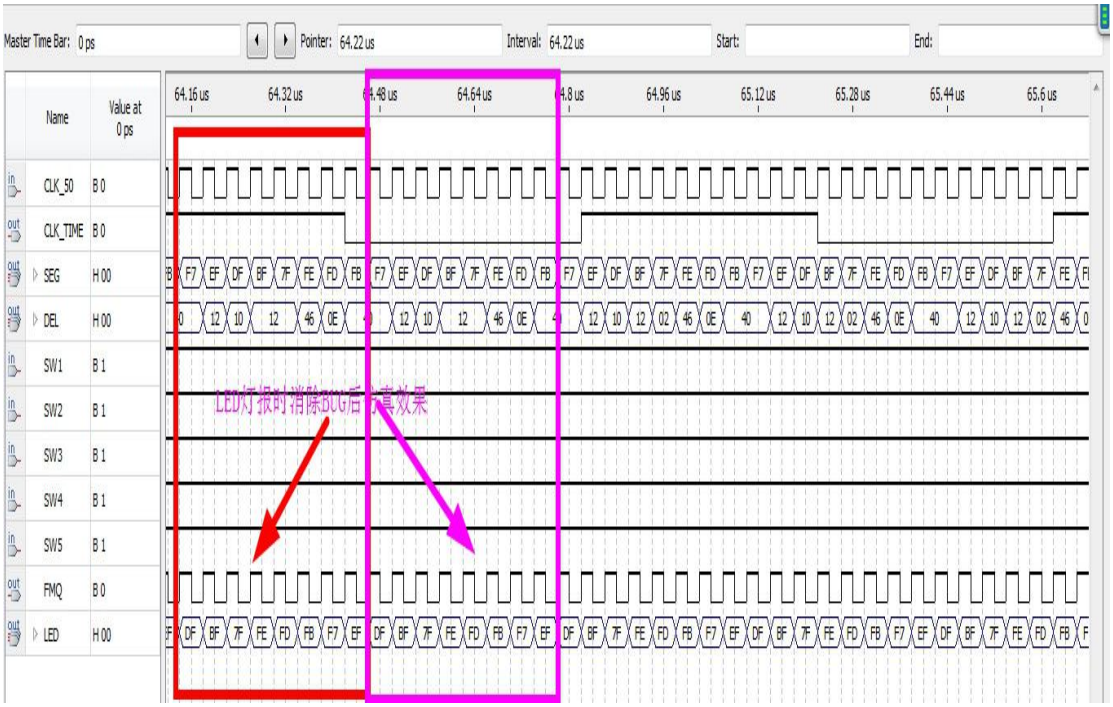
开关 S3 单独工作且有效时能实现数字钟复位功能。



开关 S5 有效时能实现数字钟预置时间功能。

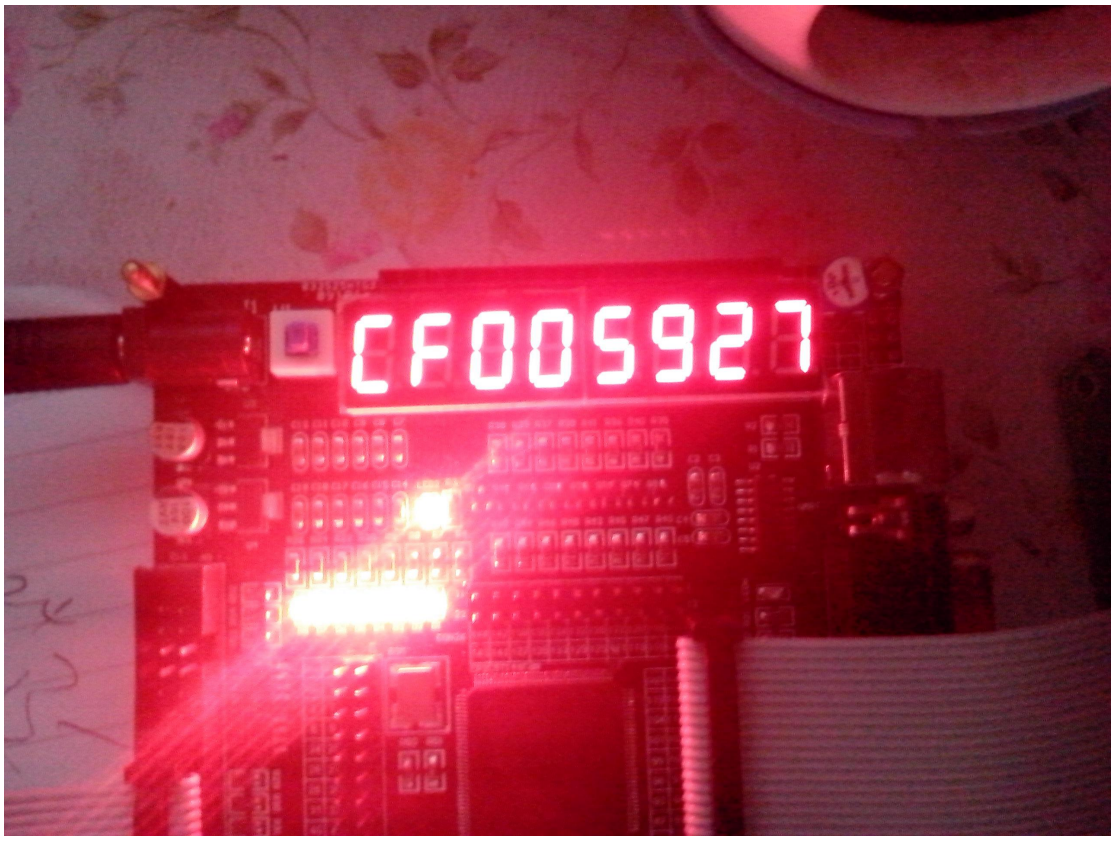


整点报时功能进行时 LED 灯与蜂鸣器能工作，但是存在小 BUG。



修正报时 LED 灯小 BUG 后报时功能完全能正常进行。

六、数字钟实验现象及分析



在 FPGA 上数字钟显示效果，图中整点报时采用的是 8 个 LED 灯一起闪烁进行的。（其他功能展示通过图片难以区分出功能，所以不列出，在实验检验块会直观反映出来的）

七、数字钟设计总结

通过本次数字钟的全程设计，从查找资料到设计分析，再到代码调试，功能仿真，其间波折的犯错与纠错过程是我在本次设计中最大的收获。通过整体与细节的分析，从模块上进行功能划分，在代码上进行细致的逻辑判断与检验，一步步达到设计要求，在这个过程中，不断的认识，熟悉，理解 VHDL 语言的特点，也深化了 VHDL 语言与其他软件语言的重大区别：VHDL 语言是以电路状态相关的，因而语句对应实际的电路，所以在程序设计上应尽可能避免乘法、除法和循环操作，用移位操作更好。

在功能仿真中通过细致分析发现了程序中存在的 BUG，再通过代码的逻辑分析修正了 BUG，这个过程也让我学到了不少语法认识，比如 IF...ELSIF...ELSE...END IF、CASE 选择分支语句、循环移位操作等等。

整体上，本次试验设计是成功的，完成了数字钟要求的基本功能，计时、调试、复位、报时、显示等。报时检测中开关 S4 的预置数方便实验进行检测。总之，这次实验是一次很好的学习与提升体验！

```
-- 进程 P1
-- 实现功能为分频。分别为计数频率和调时频率:
-- CLK 为输入 50MHzHz 频率，CLK1Hz、CLK10Hz、CLK200Hz、CLK1KHz 为分频后输出频率
-- 输入时钟 CLK，输出分频时钟 CLK1Hz、CLK10Hz、CLK200Hz、CLK1KHz
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY P1 IS
    PORT( CLK      :  IN  STD_LOGIC;
          CLK1Hz   :  OUT STD_LOGIC;
          CLK10Hz  :  OUT STD_LOGIC;
          CLK200Hz :  OUT STD_LOGIC;
          CLK1KHz  :  OUT STD_LOGIC
       );
END P1;
------------------------------------------------------
ARCHITECTURE BEHAVE OF P1 IS
BEGIN
--仿真使用代码，唯一进程
--PROCESS(CLK)        --产生 1Hz 信号，计时脉冲
--  VARIABLE CNT : INTEGER RANGE 0 TO 19;
--BEGIN
--  IF CLK='1' AND CLK'EVENT THEN
--      IF CNT=19 THEN
--         CNT:=0;
--      ELSE
--          IF CNT<10 THEN
--              CLK1Hz<='1';
--         ELSE CLK1Hz<='0';
--         END IF;
--      CNT:=CNT+1;
--      END IF;
--  END IF;
--END  PROCESS;
--CLK10Hz<=CLK;
--CLK200Hz<=CLK;
--CLK1KHz<=CLK;

PROCESS(CLK)        --产生 1Hz 信号，计时脉冲
    VARIABLE CNT : INTEGER RANGE 0 TO 49999999;
BEGIN
    IF CLK='1' AND CLK'EVENT THEN
        IF CNT=49999999 THEN
           CNT:=0;
```

```
      ELSE
            IF CNT<25000000 THEN
                CLK1Hz<='1';
         ELSE CLK1Hz<='0';
         END IF;
      CNT:=CNT+1;
      END IF;
    END IF;
END  PROCESS;

PROCESS(CLK)        --产生 10Hz 信号，LED 报时脉冲
    VARIABLE CNT : INTEGER RANGE 0 TO 4999999;
BEGIN
    IF CLK='1' AND CLK'EVENT THEN
        IF CNT=4999999 THEN
           CNT:=0;
      ELSE
            IF CNT<2500000 THEN
                CLK10Hz<='1';
         ELSE CLK10Hz<='0';
         END IF;
      CNT:=CNT+1;
      END IF;
    END IF;
END  PROCESS;

PROCESS(CLK)        --产生 200Hz 信号，蜂鸣器报时脉冲
    VARIABLE CNT : INTEGER RANGE 0 TO 249999;
BEGIN
    IF CLK='1' AND CLK'EVENT THEN
        IF CNT=249999 THEN
           CNT:=0;
      ELSE
            IF CNT<125000 THEN
                CLK200Hz<='1';
         ELSE CLK200Hz<='0';
         END IF;
      CNT:=CNT+1;
      END IF;
    END IF;
END  PROCESS;

PROCESS(CLK)        --产生 1KHz 信号，扫描脉冲
    VARIABLE CNT : INTEGER RANGE 0 TO 49999;
```

```
BEGIN
    IF CLK='1' AND CLK'EVENT THEN
        IF CNT=49999 THEN
            CNT:=0;
      ELSE
            IF CNT<25000 THEN
                CLK1KHz<='1';
         ELSE CLK1KHz<='0';
         END IF;
      CNT:=CNT+1;
      END IF;
    END IF;
END  PROCESS;
END BEHAVE;
```

```
--  进程 P2
--  实现功能为复位、调时
--  输入分频时钟 CLK1Hz，复位键 RST=0 时，复位，调时脉冲 S1，调分脉冲 S2。
--  HOUR10~SEC1 为计数当前状态输入
--  HOUR_10~SEC_1 为调时输出状态；作为计数模块的输入；
--  K2 为状态控制位，当 K2=1 时，说明调时模块工作，此时计数模块停止计时，K2=0 时，
调时模块不工作，此时计数模块照常工作。
--  K1 为计时模制控制位，当 K1=1 时，一天为 12 小时，K1=0 时，一天为 24 小时。
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY P2 IS
    PORT(  CLK1Hz                                    :  IN     STD_LOGIC;
           S1_H,S2_M,S3_RST,S4_T1,S5_YI              :  IN     STD_LOGIC;
           TC2                                       :  BUFFER STD_LOGIC;
           S4_T10                                    :  OUT    STD_LOGIC;
           S5_YO                                     :  OUT    STD_LOGIC;
           HOUR10,HOUR1,MIN10,MIN1,SEC10,SEC1        :  IN     INTEGER RANGE 0 TO 9;
           HOUR_10,HOUR_1,MIN_10,MIN_1,SEC_10,SEC_1:  BUFFER INTEGER RANGE 0 TO 9
           );
END P2;
-----------------------------------------------
ARCHITECTURE BEHAVE OF P2 IS
SIGNAL  THOUR_10,THOUR_1     :  INTEGER  RANGE 0 TO 9;
BEGIN
    PROCESS(CLK1Hz,S3_RST,S1_H,S2_M,S4_T1)
   BEGIN
   IF(S3_RST='0' OR S1_H='0' OR S2_M='0' ) THEN   -- 当调时模块中 RST，S1，S2 有
一位有效，则计数模块停止，调时模块的输出作为计数模块的输入。
```

```
    TC2<='1';       --K2为高电平，计数模块停止工作
ELSIF(S4_T1='1' AND HOUR_10*10+HOUR_1>11) THEN
    TC2<='1';
 ELSE
    TC2<='0';       --K2为低电平，计数模块正常工作
END IF;

 IF(CLK1Hz'EVENT AND CLK1Hz='1') THEN
    IF(S3_RST='0') THEN
     SEC_1<=0;
     SEC_10<=0;
     MIN_1<=0;
     MIN_10<=0;
     HOUR_1<=0;
     HOUR_10<=0;
  ELSIF(S1_H='0') THEN   --  调节小时。
     IF(HOUR_1=9) THEN
           HOUR_1<=0;
        HOUR_10<=HOUR_10+1;
     ELSIF(S4_T1='0' AND HOUR_10=2 AND HOUR_1=3) THEN
        HOUR_1<=0;
        HOUR_10<=0;
     ELSIF(S4_T1='1'AND HOUR_10=1 AND HOUR_1=1) THEN
        HOUR_1<=0;
        HOUR_10<=0;
        ELSIF(S4_T1='1' AND HOUR_10*10+HOUR_1>11) THEN
            IF(HOUR_10*10+HOUR_1=23) THEN
                HOUR_10<=0;
                HOUR_1<=0;
            END IF;
     ELSE
        HOUR_1<=HOUR_1+1;
     END IF;
  ELSIF(S2_M='0') THEN  --调节分钟
        IF(MIN_1=9) THEN
            MIN_1<=0;
            IF(MIN_10=5) THEN
                MIN_10<=0;
            ELSE
                MIN_10<=MIN_10+1;
            END IF;
     ELSE
           MIN_1<=MIN_1+1;
     END IF;
```

```
      ELSE
            HOUR_10<=HOUR10;
         HOUR_1<=HOUR1;
         MIN_10<=MIN10;
         MIN_1<=MIN1;
         SEC_10<=SEC10;
         SEC_1<=SEC1;
      END IF;
    END IF;
   END PROCESS;
    S4_T10<=S4_T1;
   S5_YO<=S5_YI;
END BEHAVE;
```

```
--  进程 P3
--  实现功能为计时
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY P3 IS
    PORT(CLK1Hz          :  IN  STD_LOGIC;  --时钟输入信号
         TC2             :  IN  STD_LOGIC;  --状态标志，当 RST、S1、S2 都 1 时为
0：调时进行、计时停止；否则为 1：调时结束、计时继续
        S4_T             :  IN  STD_LOGIC;  --时模可变输入信号，低电平为 24 小时
制高电平为 12 小时制
         S5_Y            :  IN  STD_LOGIC;  --报时检验预置数开关 S5,低电平有效
        HOUR_10,HOUR_1   :  IN  INTEGER RANGE 0 TO 9;
        MIN_10,MIN_1     :  IN  INTEGER RANGE 0 TO 9;
        SEC_10,SEC_1     :  IN  INTEGER RANGE 0 TO 9 ;
        HOUR10,HOUR1     :  BUFFER  INTEGER RANGE 0 TO 9;
        MIN10,MIN1       :  BUFFER  INTEGER RANGE 0 TO 9;
        SEC10,SEC1       :  BUFFER  INTEGER RANGE 0 TO 9
       );
END P3;
------------------------------------------------------------
ARCHITECTURE BEHAVE OF P3 IS
BEGIN
    PROCESS(CLK1Hz)
   BEGIN
        IF(S5_Y='0') THEN
            SEC1<=5;
            SEC10<=4;
            MIN1<=9;
```

```
            MIN10<=5;
        ELSIF(TC2='1')THEN       --如果模式控制状态位 TC2 为高电平，正常计数输出
            IF(S4_T='1' AND HOUR_10*10+HOUR_1>11) THEN
                SEC1<=SEC_1;
                SEC10<=SEC_10;
                MIN1<=MIN_1;
                MIN10<=MIN_10;
                HOUR10<=(HOUR_10*10+HOUR_1-12)/10;
                HOUR1<=((HOUR_10*10+HOUR_1-12) MOD 10);
                --24 小时计时模式向 12 小时计时模式转换
            ELSE
                SEC1<=SEC_1;
                SEC10<=SEC_10;
                MIN1<=MIN_1;
                MIN10<=MIN_10;
                HOUR1<=HOUR_1;
                HOUR10<=HOUR_10;
            END IF;
        ELSE
            IF(CLK1Hz'EVENT AND CLK1Hz='1') THEN
                IF(SEC1=9) THEN
                    SEC1<=0;
                IF(SEC10=5) THEN
                        SEC10<=0;
                            IF(MIN1=9) THEN
                                MIN1<=0;
                                IF(MIN10=5) THEN      -- 分为 60 进制
                                    MIN10<=0;
                                        IF(HOUR1=9) THEN
                                        HOUR1<=0;
                                        HOUR10<=HOUR10+1;
                                        ELSIF(S4_T='0' AND HOUR10=2 AND HOUR1=3)
THEN  -- 当 K1=0 时，一天 24 小时
                                            HOUR1<=0;
                                            HOUR10<=0;
                                        ELSIF(S4_T='1'    AND    HOUR10=1    AND
HOUR1=1)THEN   --当 K1=1 时，一天 12 小时
                                            HOUR1<=0;
                                            HOUR10<=0;
                                        ELSE
                                            HOUR1<=HOUR1+1;
                                        END IF;
                                ELSE
                                    MIN10<=MIN10+1;
```

```
                        END IF;
                    ELSE
                        MIN1<=MIN1+1;
                    END IF;
                ELSE
                    SEC10<=SEC10+1;
                END IF;
            ELSE
                SEC1<=SEC1+1;
            END IF;
        END IF;
      END IF;
        END IF;
    END PROCESS;
END BEHAVE;
```

```
--  进程 P4
--  实现功能为整点报时
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY P4 IS
    PORT(CLK10Hz      :  IN  STD_LOGIC;              -- 输入时钟信号
         CLK200Hz     :  IN  STD_LOGIC;
         LED  :  OUT STD_LOGIC_VECTOR(7 DOWNTO 0);  -- LED 灯控制信号，作报时用
         FMQ          :  OUT STD_LOGIC;              -- 蜂鸣器响，作报时用
         MIN10,MIN1,SEC10,SEC1  :  IN  INTEGER RANGE 0 TO 9     -- 实现整点报时，
59 分 55 秒开始
       );
END P4;
-------------------------------------------------------------
ARCHITECTURE BEHAVE OF P4 IS
SIGNAL CNT_LED  :  INTEGER  RANGE 0 TO 8;
BEGIN
    PROCESS(CLK10Hz)
        BEGIN
        IF(CLK10Hz'EVENT AND CLK10Hz='1') THEN
            IF(MIN10=5 AND MIN1=9 AND SEC10=5 AND (SEC1>=0 AND SEC1<=9)) THEN     --
在 59 分 50 秒开始提示
                IF CNT_LED=7 THEN
                    CNT_LED<=0;
                ELSE
                    CNT_LED<=CNT_LED+1;
                END IF;
```

```
                CASE CNT_LED IS
                    WHEN 0 => LED<="01111111";
                    WHEN 7 => LED<="10111111";
                    WHEN 6 => LED<="11011111";
                    WHEN 5 => LED<="11101111";
                    WHEN 4 => LED<="11110111";
                    WHEN 3 => LED<="11111011";
                    WHEN 2 => LED<="11111101";
                    WHEN 1 => LED<="11111110";
                    WHEN OTHERS =>LED<="11111111";
                END CASE;
            ELSE
                LED<="11111111";
            END IF;
      END IF;

--      IF(MIN10=5 AND MIN1=9 AND SEC10=5 AND (SEC1>=0 AND SEC1<=9)) THEN
--          LED<=(OTHERS=>NOT(CLK10HZ));
--      ELSE
--          LED<="11111111";
--      END IF;
    END PROCESS;

    PROCESS(SEC1)
    BEGIN
        IF (MIN10=5 AND MIN1=9 AND SEC10=5 AND (SEC1>=0 AND SEC1<=9)) THEN
            FMQ<=CLK200Hz;
        ELSE
            FMQ<='0';
        END IF;
    END PROCESS;
END BEHAVE;
```

```
--  进程 P5
--  实现功能为数码管动态扫描显示
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY P5 IS
    PORT(CLK_SM      :  IN  STD_LOGIC;   -- 输入时钟脉冲
        SEG_SEL      :  OUT STD_LOGIC_VECTOR(7 DOWNTO 0);  -- 数码管位选信号
        HOUR10,HOUR1,MIN10,MIN1,SEC10,SEC1  :  IN  INTEGER RANGE 0 TO 9;
        DISPLAY      :  OUT STD_LOGIC_VECTOR(6 DOWNTO 0)   -- 七段显示码
        );
```

```
END P5;
------------------------------------------------------------
ARCHITECTURE BEHAVE OF P5 IS
   SIGNAL DISP_TEMP     : INTEGER RANGE 0 TO 15;
    SIGNAL SEG_SEL_TEMP  : STD_LOGIC_VECTOR(7 DOWNTO 0);
   SIGNAL DISP_DECODE   : STD_LOGIC_VECTOR(6 DOWNTO 0);
   BEGIN

   PROCESS(CLK_SM)
        VARIABLE  COUNT:  INTEGER  RANGE 0 TO 7;
      BEGIN
           IF(CLK_SM'EVENT AND CLK_SM='1') THEN      --扫描累加
               IF COUNT=7 THEN
                   COUNT:=0;
              ELSE
                 COUNT:=COUNT+1;
              END IF;
               SEG_SEL_TEMP<="11111111";
               SEG_SEL_TEMP(COUNT)<='0';
               SEG_SEL<=SEG_SEL_TEMP;
               DISPLAY<=DISP_DECODE;
        END IF;
    END PROCESS;

    PROCESS(SEG_SEL_TEMP)
   BEGIN
        CASE (SEG_SEL_TEMP) IS
            WHEN "01111111"=>DISP_TEMP<=SEC1;
            WHEN "10111111"=>DISP_TEMP<=SEC10;
         WHEN "11011111"=>DISP_TEMP<=MIN1;
         WHEN "11101111"=>DISP_TEMP<=MIN10;
         WHEN "11110111"=>DISP_TEMP<=HOUR1;
         WHEN "11111011"=>DISP_TEMP<=HOUR10;
         WHEN "11111101"=>DISP_TEMP<=15;
         WHEN "11111110"=>DISP_TEMP<=12;
            WHEN OTHERS=>DISP_TEMP<=10;
      END CASE;
    END PROCESS;

    PROCESS(DISP_TEMP)       --数码管显示转换
        BEGIN
            CASE DISP_TEMP IS
                WHEN 0=>DISP_DECODE<="1000000";   --0
                WHEN 1=>DISP_DECODE<="1111001";   --1
```

```
                WHEN 2=>DISP_DECODE<="0100100";   --2
                WHEN 3=>DISP_DECODE<="0110000";   --3
                WHEN 4=>DISP_DECODE<="0011001";   --4
                WHEN 5=>DISP_DECODE<="0010010";   --5
                WHEN 6=>DISP_DECODE<="0000010";   --6
                WHEN 7=>DISP_DECODE<="1111000";   --7
                WHEN 8=>DISP_DECODE<="0000000";   --8
                WHEN 9=>DISP_DECODE<="0010000";   --9
                WHEN 15=>DISP_DECODE<="0001110";  --F
                WHEN 12=>DISP_DECODE<="1000110";  --C
                WHEN OTHERS=>DISP_DECODE<="1111111";   --全灭
            END CASE;
    END PROCESS;
END BEHAVE;
```