

**数字系统设计课程**

**设计报告**

|  |  |
| --- | --- |
| **作品名称：** | **多功能万年历** |
| **学生姓名：** | **利俊标** |
| **学生专业：** | **14信息工程（电联班）** |
| **开课学期：** | **2016-2017（1）** |

目录

[一、要求 1](#_Toc476083857)

[二、设计思路 1](#_Toc476083858)

[2．1、设计思路 1](#_Toc476083859)

[2．2、原理框图 2](#_Toc476083860)

[三、模块实现 3](#_Toc476083861)

[3.1、秒分模块 3](#_Toc476083862)

[3.2、时模块 4](#_Toc476083863)

[3.3、日模块 4](#_Toc476083864)

[3.4、月模块 6](#_Toc476083865)

[3.5、年模块1 6](#_Toc476083866)

[3.6、年模块2 7](#_Toc476083867)

[3.7、闹钟模块 7](#_Toc476083868)

[3.8、控制模块 8](#_Toc476083869)

[3.8、切换显示模块 9](#_Toc476083870)

[3.9、合并模块 10](#_Toc476083871)

[3.10、四选一模块 10](#_Toc476083872)

[3.11、驱动模块 11](#_Toc476083873)

[3.12、其它模块 11](#_Toc476083874)

[四、测试 13](#_Toc476083875)

[4.1、整体电路图 13](#_Toc476083876)

[4.2、试验效果 14](#_Toc476083877)

[五、感想 15](#_Toc476083878)

[六、附件：源程序 16](#_Toc476083879)

# 一、要求

设计一个万年历系统，通过数码管显示。

1. 显示分钟、小时，秒通过一个LED来进行闪烁显示。
2. 可设置时间，通过按键进行设置，设置方式可采用增加或者减小来进行显示；
3. 显示年、月、日等，通过按键进行显示内容的切换；

发挥部分：

1. 显示星期；
2. 具备闹钟功能；
3. 具备跑马表功能。

# 二、设计思路

## 2．1、设计思路

根据题目的要求，可知本作品包含秒、分、时、日、月、年等模块。由于以上模块之间存在这进位关系，所以可以独立成六个模块。各个模块之间通过进位联系。

由于开发板上的时钟位50MHz，所以必须有分频模块，作为秒的时钟信号。为了显示信息，需要数码管驱动模块。数码管只有四位，所以需要切换显示模块。要实现设置时间，则可以通过按键，模仿进位，对时间进行调整，姑且称之位控制模块。除此之外，我们需要对按键进行消抖，以提高按键信息的稳定性，所以消抖模块。

实现了基本的功能，我们想更进一步，选择了意义重大的闹钟进行发挥。闹钟模块与时间模块类似，但需要与实际时间进行比对，同时，如果时间吻合，则输出方波，驱动蜂鸣器（用led代替）。

## 2．2、原理框图

闹钟

分频

时分秒

月日

年

时钟

设置模块

选择显示

按键

驱动

数码管

按键

图 1 原理框图

注：图1中省略一些次要模块。

# 三、模块实现

本系统采用模块化的方法，分别按照框图中的功能划分，自顶向下设计，一步一步完成相应的模块。

## 3.1、秒分模块

秒、分模块原理相似，都可以看作60进制的计数器。但两者的输出不同。秒模块的输入clk位分频器的输出的1Hz脉冲，输出为60s的进位。而分模块的输出除进位外，还有供显示的两个4位BCD码输出，其中q1为个位，q2为十位。

由于两个模块功能相似，只是输出不同，所以我们只对分模块进行仿真。有图4可以得出结论，模块正常工作。



图 2 秒模块



图 3 分模块

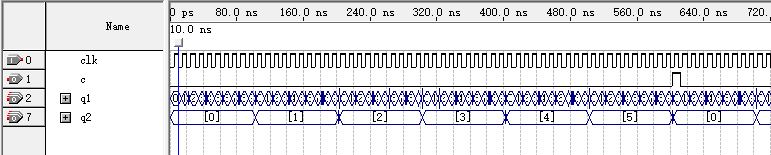


图 4 分模块仿真

## 3.2、时模块

时模块相当于24进制计数器，当计数达到24时便产生一个高电平脉冲。Clk接收来自分模块的进位。此外，还有供显示的两个4位BCD码输出，其中q1为个位，q2为十位。

由仿真图可以知道设计满足要求。



图 5 时模块

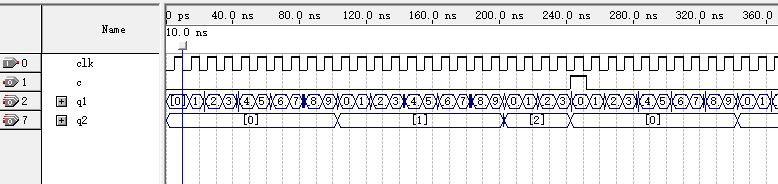


图 6 时模块的仿真

## 3.3、日模块

日模块相当于计数器，但每月的天数不同，根据month\_ifo来确定每月的天数，计数达到当月的天数时便产生一个高电平脉冲。Clk接收来自时模块的进位。此外，还有供显示的两个4位BCD码输出，其中q1为个位，q2为十位。

由仿真图可以知道设计满足要求。



图 7 月模块

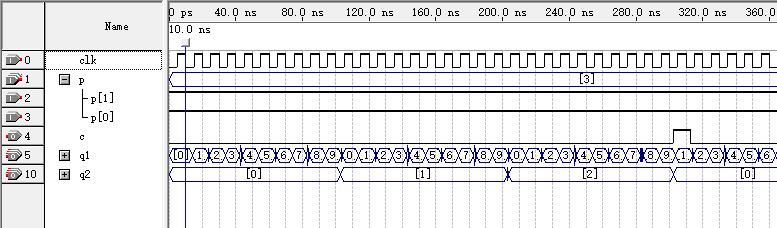
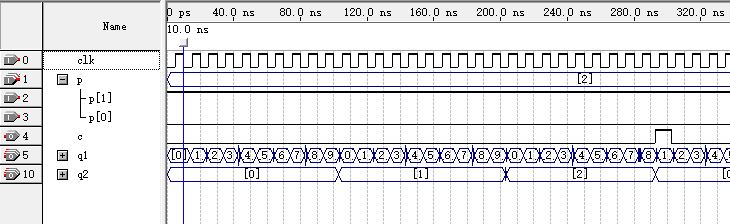
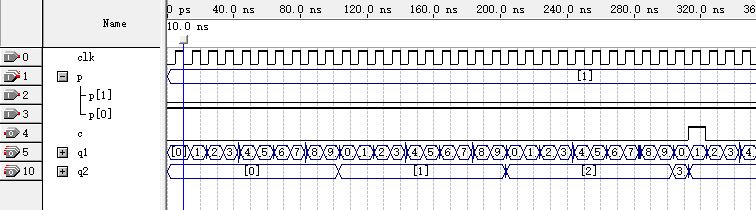
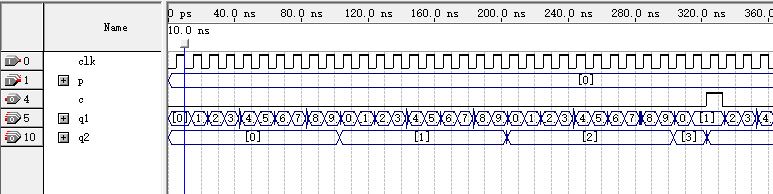


图 8 日模块仿真

## 3.4、月模块

月模块相当于12进制的带进位的计数器。Clk为天的进位，run为年是否为闰年的判断。输出month\_ifo为每月天数的判断信号，c为进位。此外，还有供显示的两个4位BCD码输出，其中q1为个位，q2为十位。



图 9 月模块

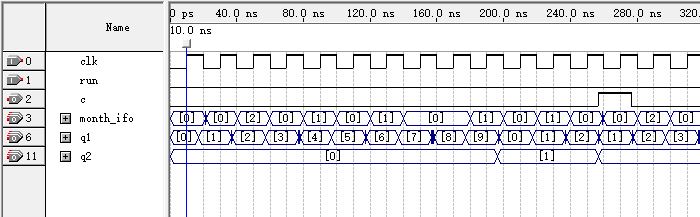


图 10 月模块仿真

## 3.5、年模块1

年低两位模块相当于100进制的带进位的计数器。Clk为月的进位，run\_400为百年是否为闰年的判断。输出run为年是否为闰年的判断，c为进位。此外，还有供显示的两个4位BCD码输出，其中y1为个位，y2为十位。



图 11 年模块1

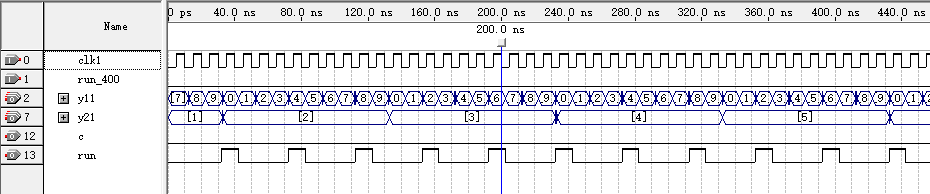


图 12 年模块仿真

## 3.6、年模块2

与3.5相似，其中run\_400为百年是否为闰年的判断。



图 13 年模块

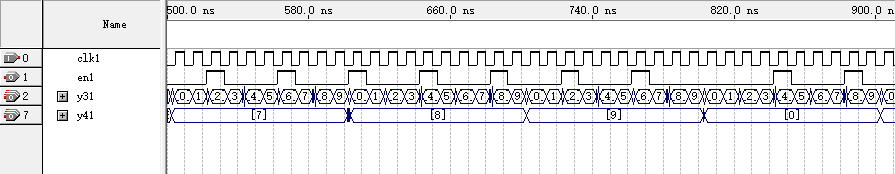


图 14 年模块的仿真

## 3.7、闹钟模块

通过min、hour输入与实际的时间对比，若吻合则连接clk\_1Hz1（占空比50%的1Hz频率）与ring，点亮led。En为分吻合时为高电平。输出还有供显示的四个4位BCD码输出。

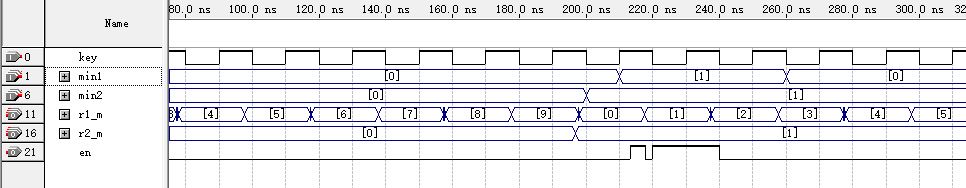
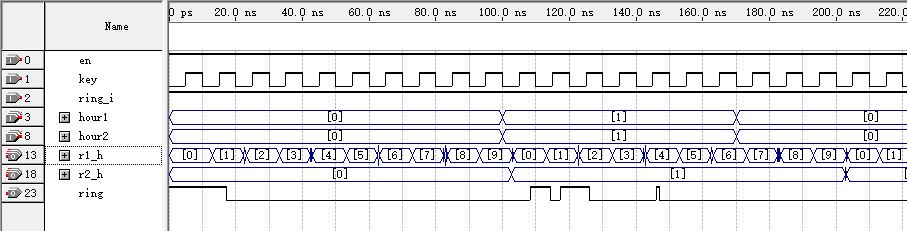


图 15 闹钟仿真

## 3.8、控制模块

s\_o,min\_o,h\_o,d\_o,M\_o,y\_o，各个模块的进位信号。

en\_time,en\_date,en\_year,en\_ring，显示信息。

key\_cs ，k1,k2 ，切换显示控制，调时模块的控制，调时信号输入

min\_i,h\_i,d\_i,M\_i,y1\_i,y2\_i,r1\_i,r2\_i，各计数器的输入时钟信号

通过模仿各个时间计数器的进位输出，来调整时间。K1调整修改的模块，k2短按加1，长按连续增加。



图 16 控制模块

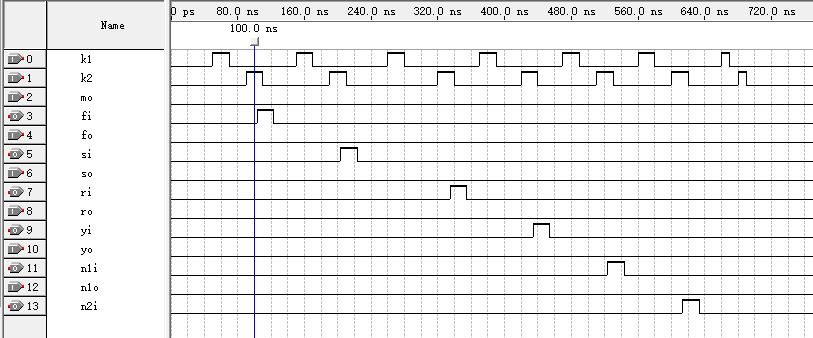


图 17 控制模块仿真

## 3.8、切换显示模块

Key\_input 为按键，按一次，四个输出依次输出高电平。由仿真可知，初始默认显示时间，拿下按键，输出依次高电平，符合设计要求。依次显示时分、日月、年份、闹钟。



图 18切换显示模块

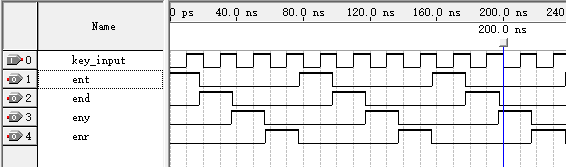


图 19 切换模块仿真

## 3.9、合并模块

将时和分、日和月、年份、闹钟的四个BCD的输出合成十六位输出，输出至四选一模块供选择。



图 20 合并模块

## 3.10、四选一模块

p\_time，p\_date，p\_year，p\_ring，分别位切换模块的输出。本模块配合切换模块，根据p\_time，p\_date，p\_year，p\_ring的值选择pick的相应输出，作为pick\_out的输出到驱动模块。由仿真结果可知，设计满足要求。



图 21 四选一模块

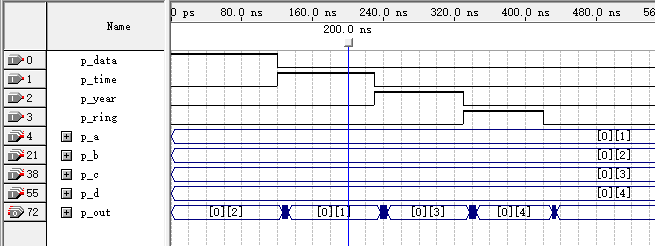


图 22 四选一模块仿真

## 3.11、驱动模块

驱动数码管



图 23 驱动模块

## 3.12、其它模块

**消抖模块1**

消抖，当按下达到5ms时，输出一个高电平脉冲。



图 24 消抖模块1

**消抖模块2**

消抖，当按下达到5ms时，输出一个高电平脉冲。当按下超过2s，则会输出5Hz（clk\_5提供）的时钟，用来快速设置时间、日期、年份和闹钟。



图 25 消抖模块2

**分频模块**

将输入的50MHz的时钟信号分频为1Hz（clk\_1Hz、clk\_1Hz1为1Hz，clk\_1Hz1占空比50%的1Hz），其中clk\_1Hz用来驱动系统工作，clk\_1Hz1来点亮led，显示秒。Clk\_5用来快速设计的5Hz。



图 26 分频模块

# 四、测试

## 4.1、整体电路图

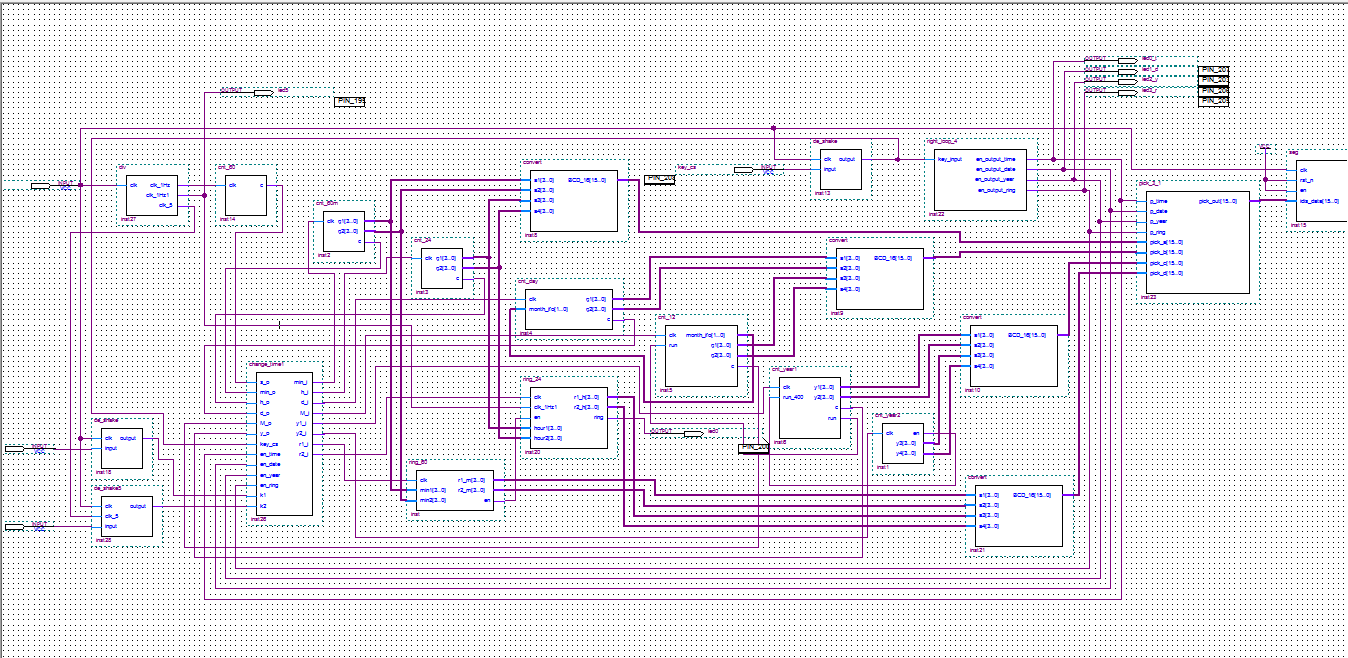


图 27 整体电路

## 4.2、试验效果

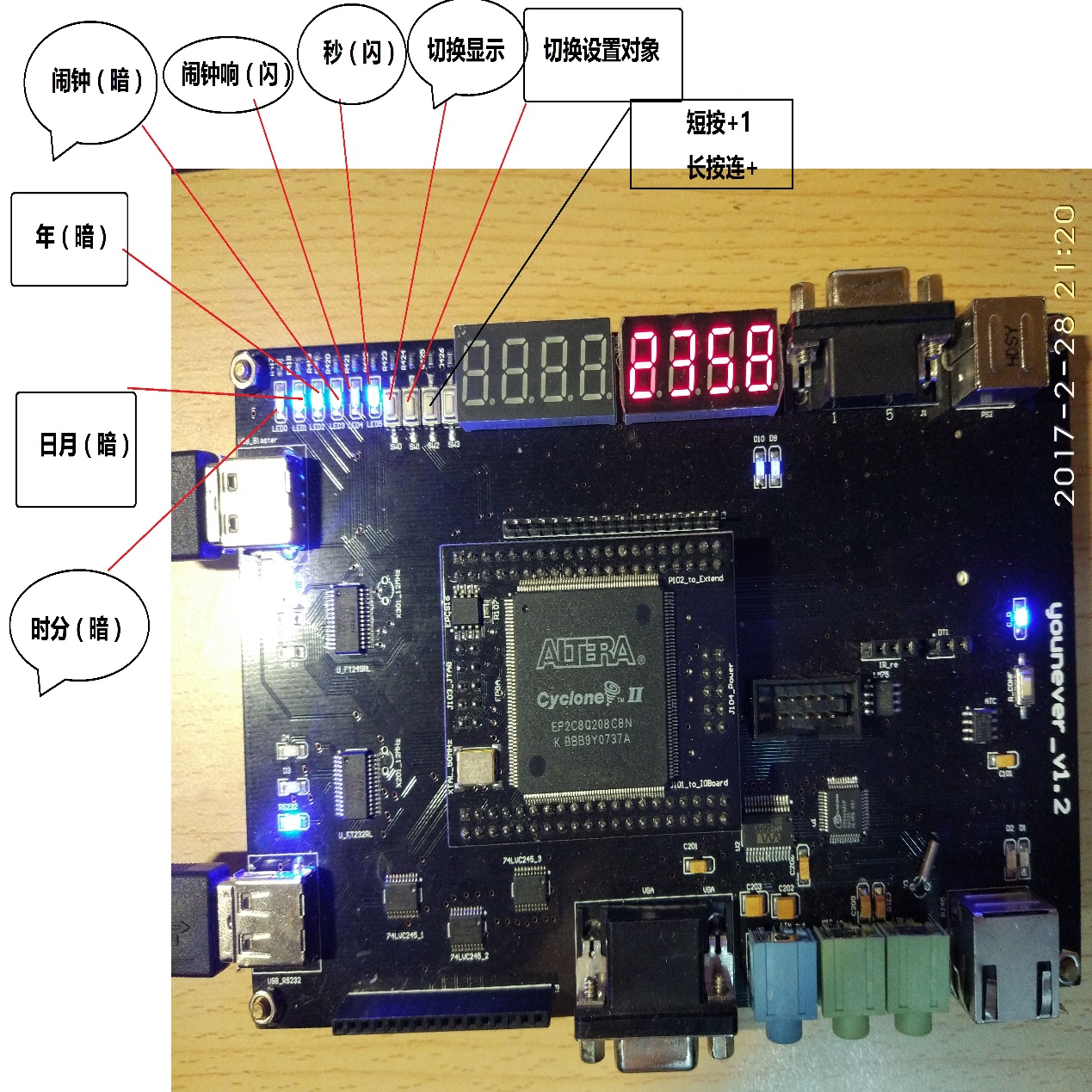


图 28 实物图

# 五、感想

这次课设能够将理论联系实际，通过不断查询资料，更改方案，终于成功完成。这个过程耗时接近一个星期，其中收获很多东西。

# 六、附件：源程序

秒

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY cnt\_60 IS

PORT(clk:IN STD\_LOGIC;

c:OUT STD\_LOGIC);

END ENTITY cnt\_60;

ARCHITECTURE rt\_1 OF cnt\_60 IS

SIGNAL q11,q22:STD\_LOGIC\_VECTOR(3 DOWNTO 0);

BEGIN

PROCESS(clk)

BEGIN

IF clk'EVENT AND clk='1' THEN

q11<=q11+'1';

IF q11=9 AND q22/=5 THEN

q11<="0000";

q22<=q22+'1';

END IF;

IF q22=5 AND q11=9 THEN

q22<="0000";

q11<="0000";

c<='1';

ELSE

c<='0';

END IF;

END IF;

END PROCESS;

END ARCHITECTURE rt\_1;

分:

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY cnt\_60m IS

PORT(clk:IN STD\_LOGIC;

q1,q2:OUT STD\_LOGIC\_VECTOR(3 DOWNTO 0);

c:OUT STD\_LOGIC);

END ENTITY cnt\_60m;

ARCHITECTURE rt\_1 OF cnt\_60m IS

SIGNAL q11:STD\_LOGIC\_VECTOR(3 DOWNTO 0):="0110";

SIGNAL q22:STD\_LOGIC\_VECTOR(3 DOWNTO 0):="0101";

BEGIN

PROCESS(clk)

BEGIN

IF clk'EVENT AND clk='1' THEN

q11<=q11+'1';

IF q11=9 AND q22/=5 THEN

q11<="0000";

q22<=q22+'1';

END IF;

IF q22=5 AND q11=9 THEN

q22<="0000";

q11<="0000";

c<='1';

ELSE

c<='0';

END IF;

END IF;

END PROCESS;

q1<=q11;q2<=q22;

END ARCHITECTURE rt\_1;

时:

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY cnt\_24 IS

PORT(clk:IN STD\_LOGIC;

q1,q2:OUT STD\_LOGIC\_VECTOR(3 DOWNTO 0);

c:OUT STD\_LOGIC);

END ENTITY cnt\_24;

ARCHITECTURE rt\_1 OF cnt\_24 IS

SIGNAL q11:STD\_LOGIC\_VECTOR(3 DOWNTO 0):="0011";

SIGNAL q22:STD\_LOGIC\_VECTOR(3 DOWNTO 0):="0010";

BEGIN

PROCESS(clk)

BEGIN

IF clk'EVENT AND clk='1' THEN

q11<=q11+'1';

IF q11=9 THEN

q11<="0000";

q22<=q22+'1';

END IF;

IF q22=2 AND q11=3 THEN

q22<="0000";

q11<="0000";

c<='1';

ELSE

c<='0';

END IF;

END IF;

END PROCESS;

q1<=q11;q2<=q22;

END ARCHITECTURE rt\_1;

天:

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY cnt\_day IS

PORT(clk:IN STD\_LOGIC;

month\_ifo:IN STD\_LOGIC\_VECTOR(1 DOWNTO 0);

q1,q2:OUT STD\_LOGIC\_VECTOR(3 DOWNTO 0);

c:OUT STD\_LOGIC);

END ENTITY cnt\_day;

ARCHITECTURE rt\_1 OF cnt\_day IS

SIGNAL q11:STD\_LOGIC\_VECTOR(3 DOWNTO 0):="0011";

SIGNAL q22:STD\_LOGIC\_VECTOR(3 DOWNTO 0):="0010";

SIGNAL P:STD\_LOGIC\_VECTOR(1 DOWNTO 0);

BEGIN

PROCESS(clk,month\_ifo)

BEGIN

IF clk'EVENT AND clk='1' THEN

q11<=q11+'1';

IF q11=9 THEN

q11<="0000";

q22<=q22+'1';

END IF;

P<=month\_ifo;

CASE P IS

WHEN "00"=>--31

IF q11="0001"AND q22="0011"THEN

q11<="0001";q22<="0000";c<='1';

ELSE c<='0';

END IF;

WHEN "01"=>--30

IF q11="0000"AND q22="0011"THEN

q11<="0001";q22<="0000";c<='1';

ELSE c<='0';

END IF;

WHEN "10"=>--28

IF q11="1000"AND q22="0010"THEN

q11<="0001";q22<="0000";c<='1';

ELSE c<='0';

END IF;

WHEN "11"=>

IF q11="1001"AND q22="0010"THEN

q11<="0001";q22<="0000";c<='1';

ELSE c<='0';

END IF;

WHEN OTHERS=>NULL;

END CASE;

END IF;

END PROCESS;

q1<=q11;q2<=q22;

END ARCHITECTURE rt\_1;

月:

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY cnt\_12 IS

PORT(clk,run:IN STD\_LOGIC;

month\_ifo:OUT STD\_LOGIC\_VECTOR(1 DOWNTO 0);

q1,q2:OUT STD\_LOGIC\_VECTOR(3 DOWNTO 0);

c:OUT STD\_LOGIC);

END ENTITY cnt\_12;

ARCHITECTURE rt\_1 OF cnt\_12 IS

SIGNAL q11:STD\_LOGIC\_VECTOR(3 DOWNTO 0):="0001";

SIGNAL q22:STD\_LOGIC\_VECTOR(3 DOWNTO 0);

SIGNAL q33:STD\_LOGIC\_VECTOR(7 DOWNTO 0);

BEGIN

PROCESS(clk)

BEGIN

IF clk'EVENT AND clk='1' THEN

q11<=q11+'1';

IF q11=9 THEN

q11<="0000";

q22<=q22+'1';

END IF;

IF q22=1 AND q11=2 THEN

q11<="0001";

q22<="0000";

c<='1';

ELSE

c<='0';

END IF;

END IF;

q33<=q22&q11;

CASE q33 IS

WHEN "00000001"=>month\_ifo<="00";--1

WHEN "00000010"=>--2

IF run='1' THEN

month\_ifo<="11";--29

ELSE

month\_ifo<="10";--28

END IF;

WHEN "00000011"=>month\_ifo<="00";--3

WHEN "00000100"=>month\_ifo<="01";--4

WHEN "00000101"=>month\_ifo<="00";--5

WHEN "00000110"=>month\_ifo<="01";--6

WHEN "00000111"=>month\_ifo<="00";--7

WHEN "00001000"=>month\_ifo<="00";--8

WHEN "00001001"=>month\_ifo<="01";--9

WHEN "00010000"=>month\_ifo<="00";--10

WHEN "00010001"=>month\_ifo<="01";--11

WHEN "00010010"=>month\_ifo<="00";--12

WHEN OTHERS=>NULL;

END CASE;

q1<=q11;q2<=q22;

END PROCESS;

q1<=q11;q2<=q22;

END ARCHITECTURE rt\_1;

年:

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY cnt\_year1 IS

PORT(clk,run\_400:IN STD\_LOGIC;

y1,y2:OUT STD\_LOGIC\_VECTOR(3 DOWNTO 0);

c,run:OUT STD\_LOGIC);

END ENTITY cnt\_year1;

ARCHITECTURE rt\_1 OF cnt\_year1 IS

SIGNAL q11:STD\_LOGIC\_VECTOR(3 DOWNTO 0):="0111";

SIGNAL q22:STD\_LOGIC\_VECTOR(3 DOWNTO 0):="0001";

SIGNAL q33:STD\_LOGIC:='0';

BEGIN

PROCESS(clk)

BEGIN

IF clk'EVENT AND clk='1' THEN

q11<=q11+'1';

IF q11=9 THEN

q11<="0000";

q22<=q22+'1';

END IF;

IF q22=9 AND q11=9 THEN

q22<="0000";

q11<="0000";

c<='1';

ELSE

c<='0';

END IF;

END IF;

END PROCESS;

PROCESS(clk,run\_400)

BEGIN

IF clk'EVENT AND clk='1' THEN

if q22=9 AND q11=9 THEN

if run\_400='1'then

q33<='1';

else

q33<='0';

end if;

elsif q22(0)='1' then

if q11=1 or q11=5 or q11=9 then

q33<='1';

else q33<='0';

end if;

elsif q22(0)='0' then

if q11=3 or q11=7 then

q33<='1';

else q33<='0';

end if;

else q33<='0';

end if;

end if;

END PROCESS;

y1<=q11;y2<=q22;run<=q33;

END ARCHITECTURE rt\_1;

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY cnt\_year2 IS

PORT(clk:IN STD\_LOGIC;

en:out std\_logic;--整除400时为高点平

y3,y4:OUT STD\_LOGIC\_VECTOR(3 DOWNTO 0));

END ENTITY cnt\_year2;

ARCHITECTURE rt\_1 OF cnt\_year2 IS

signal en\_1:std\_logic:='0';

SIGNAL q11:STD\_LOGIC\_VECTOR(3 DOWNTO 0):="0000";

SIGNAL q22:STD\_LOGIC\_VECTOR(3 DOWNTO 0):="0010";

begin

PROCESS(clk)

BEGIN

IF clk'EVENT AND clk='1' THEN

q11<=q11+'1';

IF q11=9 THEN

q11<="0000";

q22<=q22+'1';

END IF;

IF q22=9 AND q11=9 THEN

q22<="0000";

q11<="0000";

END IF;

END IF;

END PROCESS;

PROCESS(clk)

BEGIN

IF clk'EVENT AND clk='1' THEN

if q11=0 and q22=0 then

en\_1<='0';

elsif q22(0)='1' then

if q11=1 or q11=5 or q11=9 then

en\_1<='1';

else en\_1<='0';

end if;

elsif q22(0)='0' then

if q11=3 or q11=7 then

en\_1<='1';

else en\_1<='0';

end if;

else en\_1<='0';

end if;

END IF;

END PROCESS;

y3<=q11;y4<=q22;en<=en\_1;

END ARCHITECTURE rt\_1;

闹钟

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY ring\_60 IS

PORT(clk:IN STD\_LOGIC;--调整信号

min1,min2:in STD\_LOGIC\_VECTOR(3 DOWNTO 0);--实际分钟

r1\_m,r2\_m:OUT STD\_LOGIC\_VECTOR(3 DOWNTO 0);

en:OUT STD\_LOGIC);--使能闹钟hour

END ENTITY ring\_60;

ARCHITECTURE rt\_1 OF ring\_60 IS

SIGNAL q11:STD\_LOGIC\_VECTOR(3 DOWNTO 0):="0111";

SIGNAL q22:STD\_LOGIC\_VECTOR(3 DOWNTO 0):="0101";

BEGIN

PROCESS(clk,min1,min2)

BEGIN

IF clk'EVENT AND clk='1' THEN

q11<=q11+'1';

IF q11=9 AND q22/=5 THEN

q11<="0000";

q22<=q22+'1';

END IF;

IF q22=5 AND q11=9 THEN

q22<="0000";

q11<="0000";

END IF;

END IF;

if q11=min1 and q22=min2 then

en<='1';

else

en<='0';

end if;

END PROCESS;

r1\_m<=q11;r2\_m<=q22;

END ARCHITECTURE rt\_1;

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY ring\_24 IS

PORT(clk,clk\_1Hz1:IN STD\_LOGIC;--调整信号

en:in STD\_LOGIC;--分吻合

hour1,hour2:in STD\_LOGIC\_VECTOR(3 DOWNTO 0);--实际小时

r1\_h,r2\_h:OUT STD\_LOGIC\_VECTOR(3 DOWNTO 0);

ring:OUT STD\_LOGIC);--响铃

END ENTITY ring\_24;

ARCHITECTURE rt\_1 OF ring\_24 IS

SIGNAL q11:STD\_LOGIC\_VECTOR(3 DOWNTO 0):="0011";

SIGNAL q22:STD\_LOGIC\_VECTOR(3 DOWNTO 0):="0010";

BEGIN

PROCESS(clk,hour1,hour2,en)

BEGIN

IF clk'EVENT AND clk='1' THEN

q11<=q11+'1';

IF q11=9 THEN

q11<="0000";

q22<=q22+'1';

END IF;

IF q22=2 AND q11=3 THEN

q22<="0000";

q11<="0000";

END IF;

END IF;

if en='1' then

if q11=hour1 and q22=hour2 then

ring<=clk\_1Hz1;

else

ring<='0';

end if;

end if;

END PROCESS;

r1\_h<=q11;r2\_h<=q22;

END ARCHITECTURE rt\_1;

控制模块

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164. ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY change\_time1 IS

PORT(s\_o,min\_o,h\_o,d\_o,M\_o,y\_o,key\_cs:IN STD\_LOGIC; --调时模块的输入信号

en\_time,en\_date,en\_year,en\_ring:in std\_logic;

k1,k2:IN STD\_LOGIC; --调时模块的控制，调时信号输入

min\_i,h\_i,d\_i,M\_i,y1\_i,y2\_i,r1\_i,r2\_i:OUT STD\_LOGIC); --各计数器的输入时钟信号

END ENTITY change\_time1;

ARCHITECTURE rt\_1 OF change\_time1 IS

signal a:STD\_LOGIC\_VECTOR(1 DOWNTO 0);--记录调整的位置

SIGNAL b:STD\_LOGIC\_VECTOR(3 DOWNTO 0);--记录显示的内容

BEGIN

process(k1,k2,en\_time,en\_date,en\_year,en\_ring,key\_cs)

BEGIN

IF k1'EVENT AND k1='1' THEN

a<=a+'1';

if a=2 then

a<="00";

end if;

END IF;

if key\_cs='1'then

a<="00";

end if;

b<=en\_ring&en\_year&en\_date&en\_time;

CASE b IS

WHEN"0001"=>--显示时间

if a="00"then --正常

min\_i<=s\_o;h\_i<=min\_o;d\_i<=h\_o;M\_i<=d\_o;y1\_i<=M\_o;y2\_i<=y\_o;r1\_i<='0';r2\_i<='0';

elsif a="01" then--调分

min\_i<=k2;h\_i<='0';d\_i<='0';M\_i<='0';y1\_i<='0';y2\_i<='0';r1\_i<='0';r2\_i<='0';

else--调时

min\_i<=s\_o;h\_i<=k2;d\_i<='0';M\_i<='0';y1\_i<='0';y2\_i<='0';r1\_i<='0';r2\_i<='0';

end if;

WHEN"0010"=>--显示日期

if a="00"then

min\_i<=s\_o;h\_i<=min\_o;d\_i<=h\_o;M\_i<=d\_o;y1\_i<=M\_o;y2\_i<=y\_o;r1\_i<='0';r2\_i<='0';

elsif a="01" then--日

min\_i<=s\_o;h\_i<=min\_o;d\_i<=k2;M\_i<='0';y1\_i<='0';y2\_i<='0';r1\_i<='0';r2\_i<='0';

else--月

min\_i<=s\_o;h\_i<=min\_o;d\_i<=h\_o;M\_i<=k2;y1\_i<='0';y2\_i<='0';r1\_i<='0';r2\_i<='0';

end if;

WHEN"0100"=>--显示年份

if a="00"then

min\_i<=s\_o;h\_i<=min\_o;d\_i<=h\_o;M\_i<=d\_o;y1\_i<=M\_o;y2\_i<=y\_o;r1\_i<='0';r2\_i<='0';

elsif a="01" then--第两位

min\_i<=s\_o;h\_i<=min\_o;d\_i<=h\_o;M\_i<=d\_o;y1\_i<=k2;y2\_i<='0';r1\_i<='0';r2\_i<='0';

else--高两位

min\_i<=s\_o;h\_i<=min\_o;d\_i<=h\_o;M\_i<=d\_o;y1\_i<=M\_o;y2\_i<=k2;r1\_i<='0';r2\_i<='0';

end if;

WHEN"1000"=>--显示ring

if a="00"then

min\_i<=s\_o;h\_i<=min\_o;d\_i<=h\_o;M\_i<=d\_o;y1\_i<=M\_o;y2\_i<=y\_o;r1\_i<='0';r2\_i<='0';

elsif a="01" then--第两位

min\_i<=s\_o;h\_i<=min\_o;d\_i<=h\_o;M\_i<=d\_o;y1\_i<=k2;y2\_i<=y\_o;r1\_i<=k2;r2\_i<='0';

else--高两位

min\_i<=s\_o;h\_i<=min\_o;d\_i<=h\_o;M\_i<=d\_o;y1\_i<=M\_o;y2\_i<=y\_o;r1\_i<='0';r2\_i<=k2;

end if;

WHEN OTHERS=>NULL;

END CASE;

END process;

END ARCHITECTURE rt\_1;

切换

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.std\_logic\_arith.all;

entity right\_loop\_4 is

--输入端有一个输入，输出端向右移动一位; 选通 (时间 日期 年) 的输出

port(key\_input:in std\_logic;

en\_output\_time,en\_output\_date,en\_output\_year,en\_output\_ring:out std\_logic);

end entity;

architecture bhv of right\_loop\_4 is

signal out\_pp:std\_logic\_vector(3 downto 0):="0001";

--0001是显示时间 0010显示日期 0100显示年份 1000\_ring

begin

process(key\_input)

begin

if key\_input'event and key\_input='1' then

out\_pp<=out\_pp(2 downto 0)&out\_pp(3);

end if;

end process;

en\_output\_time<=out\_pp(0);

en\_output\_date<=out\_pp(1);

en\_output\_year<=out\_pp(2);

en\_output\_ring<=out\_pp(3);

end architecture;

合并

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.std\_logic\_arith.all;

entity convert is --一个十位数转换成一个16位的bcd码

port(a1,a2,a3,a4:in std\_logic\_vector(3 downto 0);

BCD\_16:out std\_logic\_vector(15 downto 0));

end entity;

architecture bhv of convert is

begin

BCD\_16<=a4&a3&a2&a1;

end architecture;

四选一:

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.std\_logic\_arith.all;

entity pick\_3\_1 is --3选1选择器，用于输出端

port(p\_time,p\_date,p\_year,p\_ring:in std\_logic;

pick\_a,pick\_b,pick\_c,pick\_d:in std\_logic\_vector(15 downto 0);

pick\_out:out std\_logic\_vector(15 downto 0));

end entity;

architecture bhv of pick\_3\_1 is

signal picker:std\_logic\_vector(3 downto 0);

begin

picker<=p\_ring&p\_year&p\_date&p\_time;

process(picker)

begin

case picker is

when "0001"=> pick\_out<=pick\_a;

when "0010"=> pick\_out<=pick\_b;

when "0100"=> pick\_out<=pick\_c;

when "1000"=> pick\_out<=pick\_d;

when others=>pick\_out<="ZZZZZZZZZZZZZZZZ";

end case;

end process;

end architecture;

分频

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity div is --分频器 把50MHz 分1Hz 200Hz

port(clk:in std\_logic;

clk\_1Hz,clk\_1Hz1,clk\_5:out std\_logic);

end entity;

architecture bhv of div is

signal clk\_1,clk\_2,clk\_3:std\_logic;

begin

process(clk)

variable cnt: integer range 0 to 50000000;--50000000

begin

if(clk'event and clk ='1')then

cnt:=cnt+1;

if cnt=50000000 then

clk\_1<='1';

cnt:=0;

else

clk\_1<='0';

end if;

end if;

end process;

process(clk)

variable cnt: integer range 0 to 50000000;

begin

if(clk'event and clk ='1')then

cnt:=cnt+1;

if cnt<25000000 then--25000000

clk\_2<='0';

else

if cnt=50000000 then

cnt:=0;

end if;

clk\_2<='1';

end if;

end if;

end process;

process(clk)

variable cnt: integer range 0 to 50000000;--50000000

begin

if(clk'event and clk ='1')then

cnt:=cnt+1;

if cnt=10000000 then

clk\_3<='1';

cnt:=0;

else

clk\_3<='0';

end if;

end if;

end process;

clk\_1Hz<=clk\_1;

clk\_1Hz1<=clk\_2;

clk\_5<=clk\_3;

end architecture;

消抖

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.std\_logic\_arith.all;

entity de\_shake5 is --延迟消抖程序，输出产生一个脉冲 （按下为‘1’）

port(clk,clk\_5,input:in std\_logic;

output:out std\_logic);

end entity;

ARCHITECTURE asm3 OF de\_shake5 IS

BEGIN

--operation

op: PROCESS (clk,input)

VARIABLE COUNT1 :INTEGER RANGE 0 TO 300000;

VARIABLE COUNT2 :INTEGER RANGE 0 TO 150000000;

BEGIN

IF input='0' THEN

IF RISING\_EDGE(CLK) THEN

IF COUNT1<300000 THEN COUNT1:=COUNT1+1; --300000

ELSE COUNT1:=COUNT1;

END IF;

IF COUNT2<100000000 THEN COUNT2:=COUNT2+1; --150000000

ELSE COUNT2:=COUNT2;

END IF;

IF COUNT1=299999 and count2<100000000 THEN --299999

output<='1';

elsif count2=100000000 then

output<=clk\_5;

ELSE output<='0';

END IF;

END IF;

ELSE

COUNT1:=0;

count2:=0;

END IF;

END PROCESS ;

END asm3;

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.std\_logic\_arith.all;

entity de\_shake is --延迟消抖程序，输出产生一个脉冲 （按下为‘1’）

port(clk,input:in std\_logic;

output:out std\_logic);

end entity;

ARCHITECTURE asm3 OF de\_shake IS

BEGIN

--operation

op: PROCESS (clk,input)

VARIABLE COUNT1 :INTEGER RANGE 0 TO 300000;

BEGIN

IF input='0' THEN

IF RISING\_EDGE(CLK) THEN

IF COUNT1<300000 THEN COUNT1:=COUNT1+1;

ELSE COUNT1:=COUNT1;

END IF;

IF COUNT1=299999 THEN output<='1';

ELSE output<='0';

END IF;

END IF;

ELSE COUNT1:=0;

END IF;

END PROCESS ;

END asm3;