**多功能数字钟的设计和实现**

**——数字逻辑课程设计**

**专 业**： 计算机科学与技术

**班 级**： 计算机1302

**学 号**： 3130602050

**姓 名**： 曾 彪

**指导老师：** 丁 伟

**成 绩**：

**2015年1月14日**

[一．设计目的及任务要求 1](#_Toc409040490)

[1．设计目的 1](#_Toc409040491)

[2．任务要求 1](#_Toc409040492)

[二．设计方案 2](#_Toc409040493)

[三．设计过程及原理叙述 3](#_Toc409040494)

[四．模块原理图描述 5](#_Toc409040495)

[1.计时校时模块 5](#_Toc409040496)

[2.闹钟模块 6](#_Toc409040497)

[3．倒计时模块 7](#_Toc409040498)

[4．秒表模块 8](#_Toc409040499)

[5．模式选择模块 8](#_Toc409040500)

[6．动态显示模块 9](#_Toc409040501)

[7.分频模块 9](#_Toc409040502)

[五．主要VHDL程序清单 10](#_Toc409040503)

[1．分频器 10](#_Toc409040504)

[2．24进制加减计数器 11](#_Toc409040505)

[3．60进制加减计数器 12](#_Toc409040506)

[4．100进制计数器 14](#_Toc409040507)

[5．模式分配 14](#_Toc409040508)

[6．显示选择 15](#_Toc409040509)

[7．动态显示 16](#_Toc409040510)

[8.闹钟判断 17](#_Toc409040511)

[六．设计结论 18](#_Toc409040512)

[七．经验总结 19](#_Toc409040513)

# 一．设计目的及任务要求

## 1．设计目的

(1).学会应用数字系统设计方法进行电路设计；

(2).进一步提高quartus软件的开发应用能力；

(3).提高VHDL进行综合设计的能力；

(4).培养学生书写综合实验报告的能力。

## 2．任务要求

（1）能进行正常的时、分、秒计时，分别用6个七段数码管动态扫描显示时，分、秒的十位和个位。

（2）利用按键开关快速调整时间：时、分。

（3）通过按键开关设定闹铃时间，到了设定时间发出提示音，提示音长度为1分钟。

（4）通过按键开关设定倒计时的时间，通过开关启动倒计时，倒计时为0时发出提示音，提示音长度为1分钟。

（5）通过开关控制秒表的开始和结束。

# 二．设计方案

根据任务要求可将电路设计成四个模式，分别为计时调时模式，闹钟模式，倒计时模式和秒表模式，分别通过两个开关的组合来选择不同的模式，两个开关可有四种状态，每种状态对应一个模式，其中调时可在计时的基础上进行，共用一个模式，四种状态种除了秒表模式外都需要对时间进行设置，因此可以使用三个按键来调整时间，按键根据具体的模式来修改相应状态下的时分秒，不影响其他模式的时间，电路可以先使用VHDL语言写出各个模块的逻辑语言，然后生成器件，再一一连接起来，电路中用到的一些基本的模块有计时模块（24进制计数器、60进制计数器等）、动态显示模块、分频模块、多路选择器、多路分配器、闹钟模块、倒计时模块、秒表模块等。

电路设计时先将电路的大体框架构建起来，分模块一步步地设计，然后再根据具体的要求和实际情况一点点的细化优化，不断的调试修改，将各个电路组装起来。

# 三．设计过程及原理叙述

**计时模块**：使用一片24进制计数器和两片六十进制计数器以进位输出实现级联，通过来自基准分频器的1Hz计秒时钟，该模块完成最基本的计时、计分、计秒功能。六十进制计数器实现秒的计数，计数至59，进位进1，进位作为分计数器计时的脉冲信号，用六十进制实现分计数器，分计数至59，进位进1，分的进位作为时计数器的脉冲信号，以二十四进制实现时计数器。

**校时模块**：该模块主要要求实现校时、校分、校秒及时、分、秒清零的功能: 按下校时键，小时位计数器依次递增以调至所需要的数值；按下校分键，分位计数器依次递增以调至所需要的数值；按下校秒键，秒位计数器清零；拨上清零开关，对应数字均清零。

**分频模块**：实验中以1khz为基准频率，但系统中报时、按键消抖、计时用的频率均不相同。而所有的这些脉冲信号可以通过一个基准频率分频器生成。基准频率分频器就是一个进制很大的计数器，利用计数器的分频功能，从不同的输出位得到所需的脉冲信号1hz、100hz、1000hz等。

**动态显示模块**：时间的显示需要用到6个数码管，由于实验板上的所有数码管均对应于同一组七段码，因此，需要采用动态扫描的方式实现时间显示。在动态方式下，所有的数码管对应同一组七段码，一个数码管由一个选择段控制点亮或熄灭，如果点亮，则显示七段码所对应的的数字或字符形式。

**闹钟模块**：为了避免干扰正常计时，所设定的闹钟时间应使用新的计数器进行存储（设定闹钟的小时位和分位，秒位默认为0）；设置一个闹钟开始的开关，置0时设定闹钟时间，显示闹钟时间，置一时闹钟开始运行并显示计时时间；应实现一个比较模块，当计时到闹钟时间相对等时，驱动扬声器鸣叫；闹钟响声应限定在一分钟，，且在这段时间内应随时可以通过停鸣开关取消。

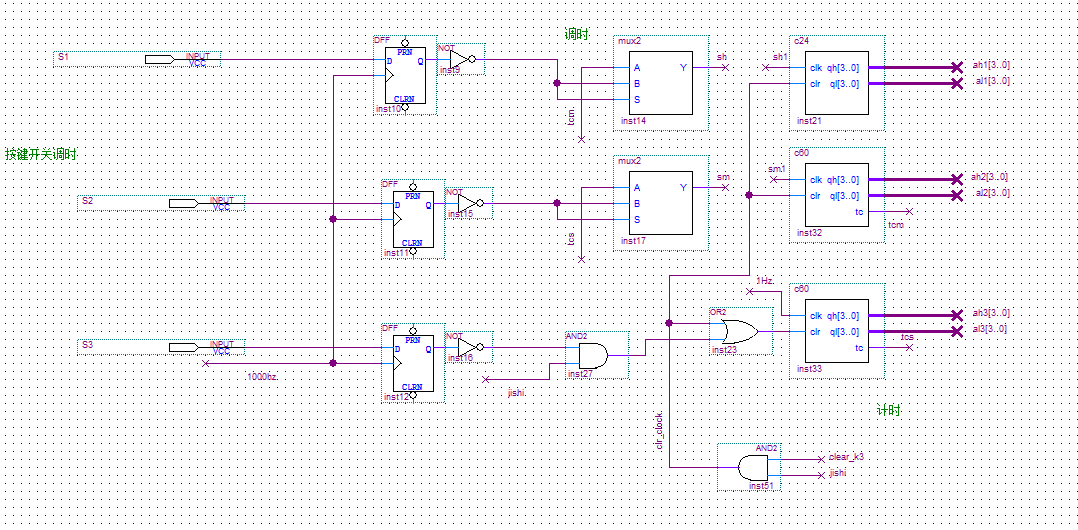
**秒表模块**：此模块不需要小时显示，需要两个60进制计数器来分别实现分、秒的计时，一个100进制计数器表示毫秒的前两位计时。

**倒计时模块**：通过组合开关选择倒计时模式，然后在倒计时开关为0的时候可以通过按键来调整倒计时的时间，调好了后通过开关启动倒计时，当时间减至0的时候发出一分钟的提示音，可以通过倒计时开关手动停止。

# 四．模块原理图描述

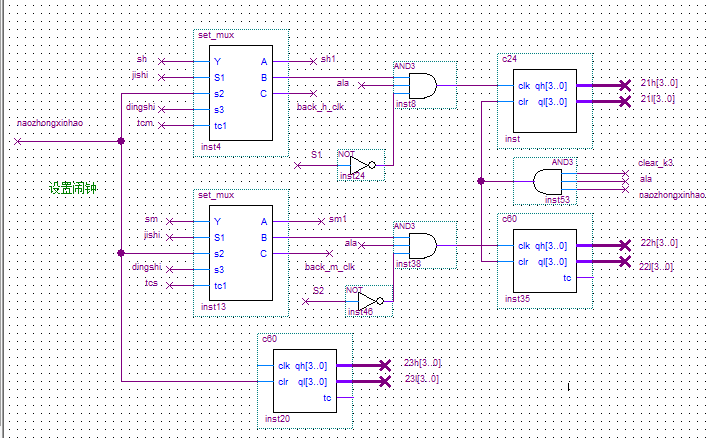
## 1.计时校时模块

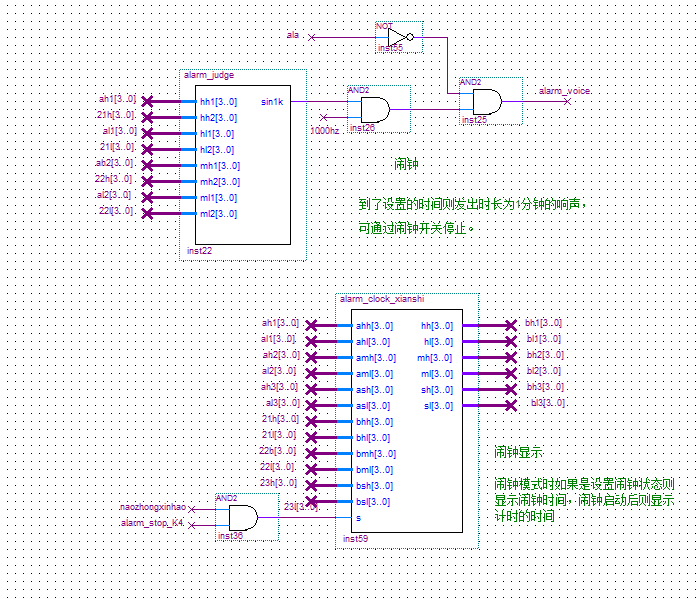
当选择这个模式后1hz的时钟信号使时钟开始工作，时间通过动态显示模块在数码管上显示出来，此时按下S1、S2、S3将在计时的基础上分别对时分秒的时间进行调整，当选择其他模式的时候，按键开关S1、S2还可以调其他模式的时间。



## 2.闹钟模块

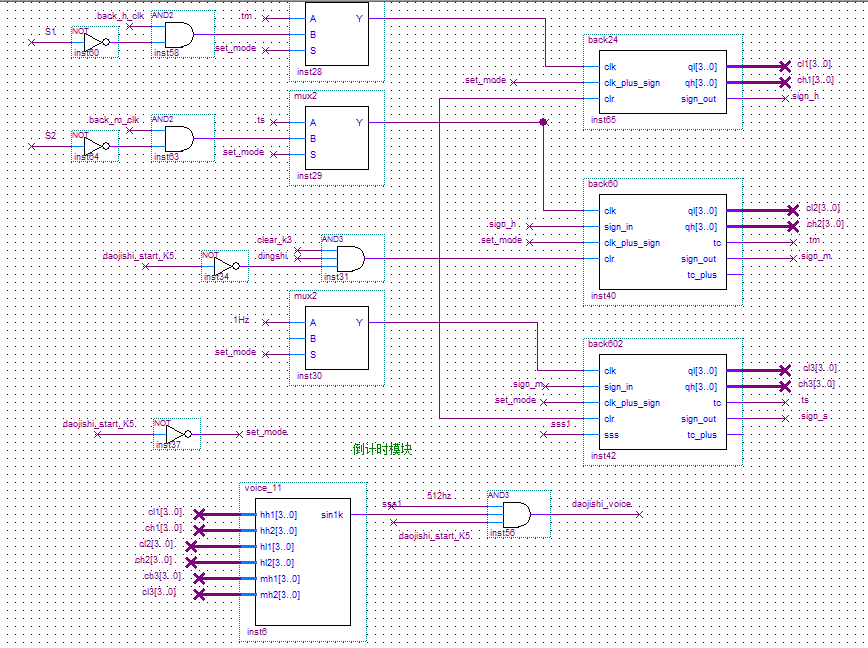
处于闹钟模式时先通过按键开关S1、S2调整时分，秒位为0，此时显示闹钟时间，然后启动闹钟，此时显示计时时间，当时间到时发出提示音，可以通过闹钟的启动开关停止。





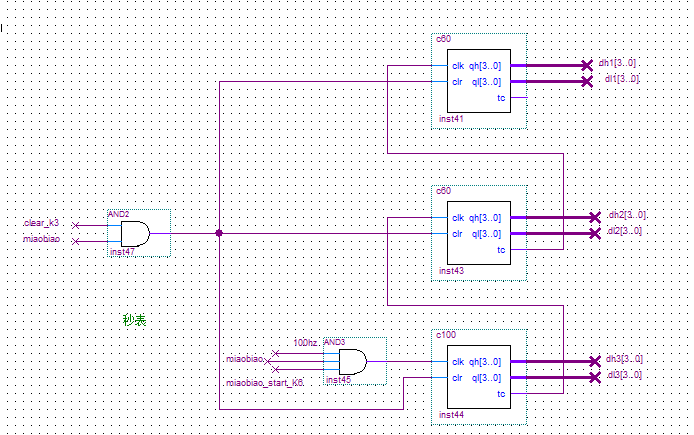
## 3．倒计时模块

在倒计时模式时，当倒计时开关为0时处于设置状态，此时可以通过开关S1、S2设置倒计时器的时间，秒位为0，设置好时间后通过开关启动倒计时，当时间到了的时候发出一分钟的提示音，可以通过启动开关关闭。模块中有三个二路选择器，设置状态时将相应的值送到计数器，倒计时状态时将时钟或进位送到计数器，计数器在设置状态时可以向上加，在倒计时状态时则向下减。



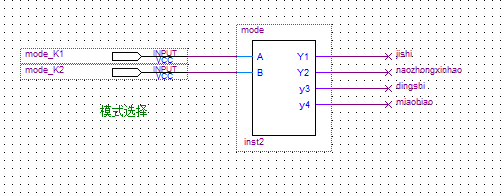
## 4．秒表模块

在秒表模式时秒表初始化为0，当启动秒表开关后，秒表向上计时，模块用到两个60进制计数器和一个100进制计数器，分别计分、秒、毫秒的前两位。可以通过启动开关暂停，清零开关清零。



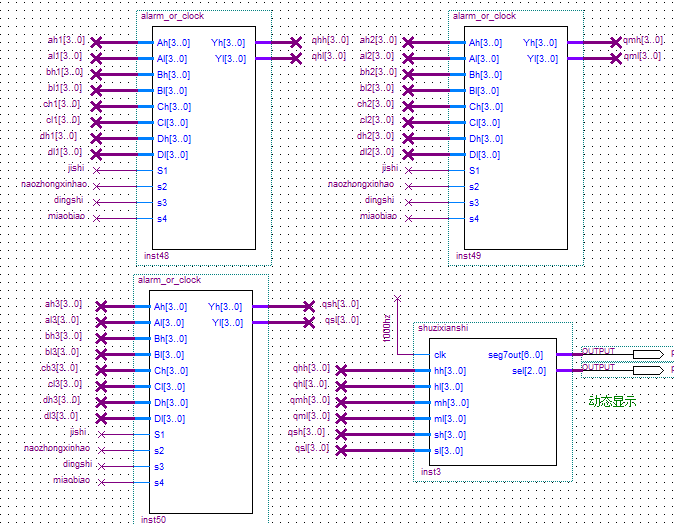
## 5．模式选择模块

此模块可以通过开关K1、K2组合成四种状态，对应四种状态。00为计时校时模式，01闹钟模式，10倒计时模式，11秒表模式。

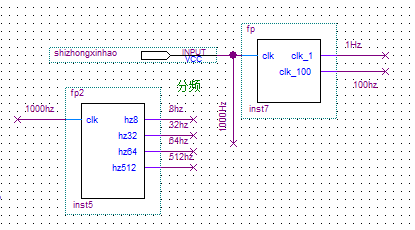


## 6．动态显示模块

模块通过相应的模式信号输出相应数字信号，在六个数码管上依次显示，由于输入频率用了1000hz所以肉眼看起来就像在同时显示。



## 7.分频模块



# 五．主要VHDL程序清单

## 1．分频器

library ieee ;

use ieee.std\_logic\_1164.all ;

use ieee.std\_logic\_unsigned.all ;

entity fp is

port ( clk : in std\_logic ;

clk\_1,clk\_100 : out std\_logic ) ;

end fp ;

architecture bhv of fp is

begin

process ( clk )

variable count1,count100 : integer range 0 to 999 := 0 ;

begin

if clk'event and clk = '1' then

if count100 = 9 then

count100 := 0 ;

clk\_100 <= '1' ;

else

count100 := count100 + 1 ;

clk\_100 <= '0' ;

end if ;

if count1 = 999 then

count1 := 0 ;

clk\_1 <= '1' ;

else

count1 := count1 + 1 ;

clk\_1 <= '0' ;

end if ;

end if ;

end process ;

end bhv ;

## 2．24进制加减计数器

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity back24 is

port (clk:in std\_logic;

clk\_plus\_sign:in std\_logic;

clr:in std\_logic;

ql,qh:buffer std\_logic\_vector(3 downto 0);

sign\_out:out std\_logic);

end back24;

architecture beh of back24 is

begin

process(clk,clr,clk\_plus\_sign)

begin

if (clr='0')then

if clk'event and clk='1' and clk\_plus\_sign='0' then

if ql="0000"then

if qh="0000"then

qh<="0000";

ql<="0000";

sign\_out<='0';

else

qh<=qh-1;

ql<="1001";

sign\_out<='1';

end if;

else

ql<=ql-1;

sign\_out<='1';

end if;

end if;

if clk'event and clk='1' and clk\_plus\_sign='1' then

ql<=ql+1;

sign\_out<='1';

if ql="1010"then

qh<=qh+1;

ql<="0000";

end if;

if qh="0010"and ql="0100"then

qh<="0000";

ql<="0000";

end if;

end if;

else

qh<="0000";

ql<="0000";

end if;

end process;

end beh;

## 3．60进制加减计数器

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity back60 is

port (clk,sign\_in:in std\_logic;

clk\_plus\_sign:in std\_logic;

clr:in std\_logic;

ql,qh:buffer std\_logic\_vector(3 downto 0);

tc,sign\_out:out std\_logic;

tc\_plus:out std\_logic);

end back60;

architecture beh of back60 is

begin

process(clk,clr,clk\_plus\_sign)

begin

if clr='0'then

if clk'event and clk='1' and clk\_plus\_sign='0' then

if qh="0000" and ql="0001" then sign\_out<='0';

end if;

if ql="0000"then

if qh="0000"then

if sign\_in='0' then

qh<="0000";

ql<="0000";

tc<='0';

sign\_out<='0';

else

qh<="0101";

ql<="1001";

tc<='1';

sign\_out<='1';

end if;

else

qh<=qh-1;

ql<="1001";

tc<='0';

sign\_out<='1';

end if;

else

ql<=ql-1;

tc<='0';

sign\_out<='1';

end if;

end if;

if clk'event and clk='1' and clk\_plus\_sign='1' then

ql<=ql+1;

sign\_out<='1';

if ql="1010" then

qh<=qh+1;

ql<="0000";

if qh="0110"then

qh<="0000";

tc\_plus<='1';

else tc\_plus<='0';

end if;

else tc\_plus<='0';

end if;

end if;

else

qh<="0000";

ql<="0000";

end if;

end process;

end beh;

## 4．100进制计数器

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity c100 is

port(clk,clr:in std\_logic;

qh,ql:buffer std\_logic\_vector(3 downto 0);

tc:out std\_logic);

end c100;

architecture beh of c100 is

begin

process(clk,clr)

begin

if clr='1' then

qh<="0000";

ql<="0000";

elsif clk'event and clk='1'then

ql<=ql+1;

if ql="1001" then

qh<=qh+1;

ql<="0000";

if qh="1010"then

qh<="0000";

tc<='1';

else tc<='0';

end if;

end if;

end if;

end process;

end beh;

## 5．模式分配

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity mode is

port(A,B:in std\_logic;

Y1,Y2,y3,y4:out std\_logic);

end mode;

architecture beh of mode is

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

begin

process(a,b)

variable temp:std\_logic\_vector(1 downto 0);

begin

temp:=a&b;

case temp is

when"00"=>y<="1000";

when"01"=>y<="0100";

when"10"=>y<="0010";

when"11"=>y<="0001";

end case;

y1<=y(3);

y2<=y(2);

y3<=y(1);

y4<=y(0);

end process;

end beh;

## 6．显示选择

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity alarm\_or\_clock is

port(Ah:in std\_logic\_vector(3 downto 0);

Al:in std\_logic\_vector(3 downto 0);

Bh:in std\_logic\_vector(3 downto 0);

Bl:in std\_logic\_vector(3 downto 0);

Ch:in std\_logic\_vector(3 downto 0);

Cl:in std\_logic\_vector(3 downto 0);

Dh:in std\_logic\_vector(3 downto 0);

Dl:in std\_logic\_vector(3 downto 0);

S1,s2,s3,s4:in std\_logic;

Yh,Yl:out std\_logic\_vector(3 downto 0));

end alarm\_or\_clock;

architecture beh of alarm\_or\_clock is

begin

process(S1,s2,s3,s4)

begin

if(S1='1')then Yh<=Ah;Yl<=Al;

elsif(S2='1')then Yh<=Bh;Yl<=Bl;

elsif(S3='1') then Yh<=Ch;Yl<=Cl;

else Yh<=Dh;Yl<=Dl;

end if;

end process;

end beh;

## 7．动态显示

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY shuzixianshi is

port(

clk:in std\_logic;

hh:in std\_logic\_vector(3 downto 0);

hl:in std\_logic\_vector(3 downto 0);

mh:in std\_logic\_vector(3 downto 0);

ml:in std\_logic\_vector(3 downto 0);

sh:in std\_logic\_vector(3 downto 0);

sl:in std\_logic\_vector(3 downto 0);

seg7out:out std\_logic\_vector(6 downto 0);

sel:buffer std\_logic\_vector(2 downto 0) );

END shuzixianshi;

ARCHITECTURE beha of shuzixianshi is

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

BEGIN

PROCESS(clk)

variable dount:std\_logic\_vector(2 downto 0):="000";

BEGIN

IF(rising\_edge(clk))then

IF dount="111" then

dount:="000";

ELSE dount:=dount+1;

END IF;

END IF;

sel<=dount;

end process;

PROCESS(sel)

BEGIN

CASE sel IS

when "000"=>key<=hh;

when "001"=>key<=hl;

when "010"=>key<="1010";

when "011"=>key<=mh;

when "100"=>key<=ml;

when "101"=>key<="1010";

when "110"=>key<=sh;

when "111"=>key<=sl;

when others=>null;

END CASE;

END PROCESS;

PROCESS (key)

BEGIN

case key is

when"0000"=>seg7out<="0111111";

when"0001"=>seg7out<="0000110";

when"0010"=>seg7out<="1011011";

when"0011"=>seg7out<="1001111";

when"0100"=>seg7out<="1100110";

when"0101"=>seg7out<="1101101";

when"0110"=>seg7out<="1111101";

when"0111"=>seg7out<="0000111";

when"1000"=>seg7out<="1111111";

when"1001"=>seg7out<="1101111";

when"1010"=>seg7out<="1000000";

when others=>null;

END CASE;

END PROCESS;

END beha;

## 8.闹钟判断

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity alarm\_judge is

port(hh1,hh2,hl1,hl2,mh1,mh2,ml1,ml2:in std\_logic\_vector(3 downto 0);

sin1k:out std\_logic);

end alarm\_judge;

architecture judge of alarm\_judge is

begin

sin1k<='1'when(hh1=hh2 and hl1=hl2 and mh1=mh2 and ml1=ml2)else'0';

end judge;

# 六．设计结论

本次实验终于完成了既定的实验要求，设计完成了计时、校时、闹钟、秒表、倒计时等各个功能，时间能正常显示，也能对时间进行修改、清零，另外还根据自己的理解以及实际情况对一些细节的地方进行了设计，比如说整个电路的调时按键都是共用S1、S2、S3，清零开关也是共用的，但是却互不影响，各个开关仅在相应的模式下才有效果。另外调闹钟和倒计时的时候他们的秒位都是置为0，默认只能调时位和分位。可以通过停鸣开关随时取消闹铃（提示音），否则闹铃（提示音）可持续时间一分钟，运行结果与设计预期结果一致。电路通过开关S1和S2组合后进入相应的模式，在具体的模式下另有开关控制它们的工作状态（设置状态、开始状态、暂停状态）。对于倒计时功能，设计了一个既可以加又可以减的24进制计数器和一个60进制计数器，当设置倒计时的时间时进行加，倒计时启动后进行减。

# 七．经验总结

**在本次课程设计中，从无到有，学到了很多自己以前实验中不知道的东西，如空间数组命名、由代码生成器件等等。实验中遇到了很多问题，有的时候是思路不清晰，刚开始的时候还不太会使用软件，不过这些问题后来都通过仔细思考、询问同学解决了，而且还学到了一些新的思路，在所有逻辑思考后，画图也同样考验逻辑思维，不允许出现一点差错，一定要仔细看清线与线之间的关系，经过多次修改，不断发现错误并解决问题，在这个过程中，对QuartusⅡ工具软件有了更深的了解和体会。只有掌握好理论知识，并懂得应用到实践中去，才会有一定的收获和体会。希望以后自己可以做的更好，积累更多的经验。有的时候遇到问题就和同学讨论，这也是一个学习的过程。在实验中也产生了好多好多错误，不过后来经过反复的思考、修改，最终解决了问题，而且还学到了一些新方法，同时对电路的理解也更加清晰了。**