**VHDL课程设计报告**

**基于FPGA的电子密码锁**

专业：通信工程

班级：6A班

姓名：何尔恒

学号：20172333090

**一、设计内容**

设计一电子密码锁，用VHDL语言描述，用QuartusII工具编译和综合，并在实验板上调试实现所要求功能和技术指标，撰写实验报告，最后提交验收并答辩。

电子密码锁（4位密码），如果输入的密码与锁内预存的密码一致，锁被打开；若多次输入错误密码，发出报警信号。

实验板上的数码管和按钮对应操作界面如下图所示。

LED

SET

↑

↓

←

CLR

该设计实现以下功能：

1. 若锁处于上锁的状态，短按一次SET，开始输入开锁密码的第一个数字位，此时最右边数码管显示“0”（其余三位数码管不显示或显示“-”），按“↑”或“↓”分别加1或减1。按下“←”按键确定并将此数字左移1位数码管，最右边数码管填“0”（其余两位数码管不显示或显示“-”）……如此完成4位密码的输入。若输入的密码与预设的密码相同，输出开锁信号（点亮led灯指示，或数码管显示开锁提示）。
2. 若输入密码错误达三次，报警，led灯闪烁。
3. 若锁处于开锁的状态，长按SET（如3秒），进入修改密码的状态，输入新密码的方式类似输入开锁密码。完成4位新密码的输入后，再次长按SET，新密码替换原密码。

其它说明：长按CLR用于模拟锁门，即按下此按键密码锁由开的状态转为锁上的状态。短按CLR用于清零已输入的部分密码，以重新输入。只有当锁是处于开锁的状态时，才允许修改密码。

**二、设计环境及工具**

Quartus II 13.0编程环境、教学实验板一套（Cyclone II EP2C5T144C8N芯片）

**三、设计原理**

实验板上的晶振产生50MHz的时钟信号，利用分频器输出消抖信号作为按键短按输入和按键长按3秒的输入。利用五个按键，依次定为set、↑、↓、←、clr按键。SET和←用于控制顶层变量number+1，number不同的值代表数码管不同的显示模式和加减位所能控制的不同显示变量(number0, number1, number2, number3)的值。按“↑”或“↓”分别使当前显示变量加1或减1。利用例子里面的消抖的思路，可得到短按和长按两种按键信号。当在一些特定情况下出现短按或者长按信号时，赋予顶层变量number特定的值使系统切换到某特定的模式。

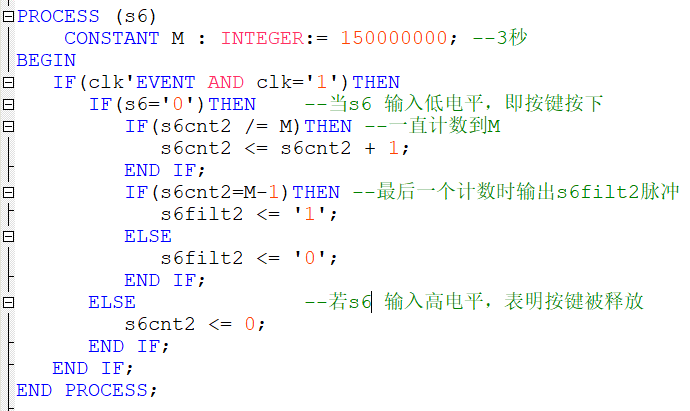
**四、系统分析**

本设计题目系统整体性较为明显，所以个人认为可在一个大的系统文件里面进行完成，只需在其中进行适当细分即可。

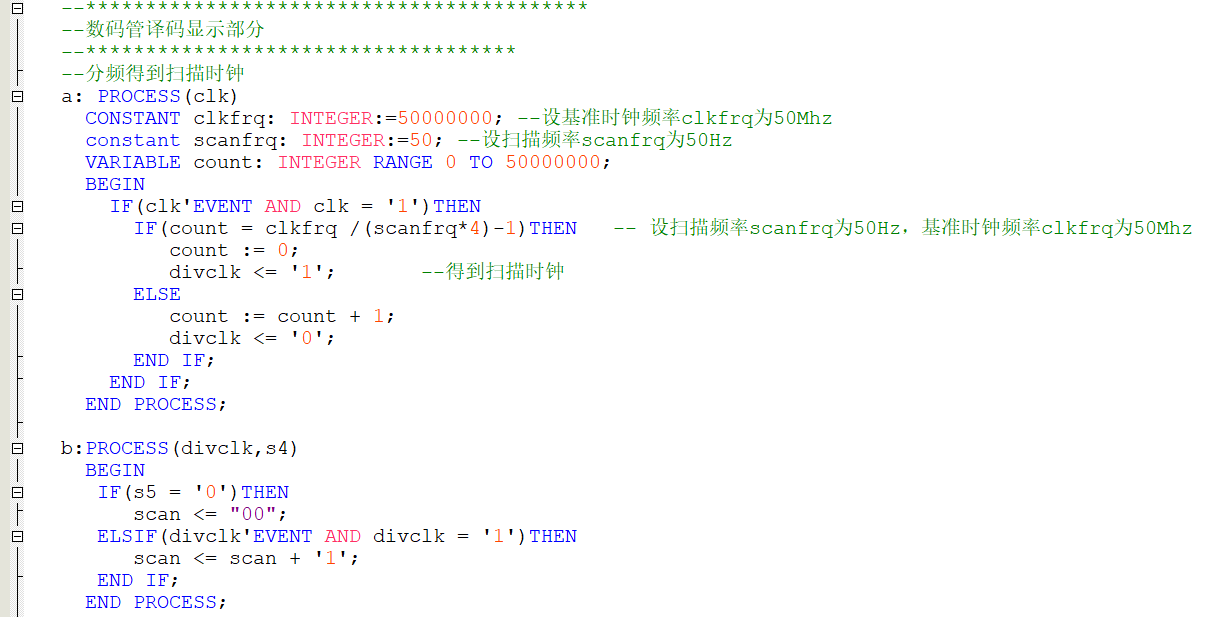
其中包含六个模块（部分举例）：

按键短按消抖模块

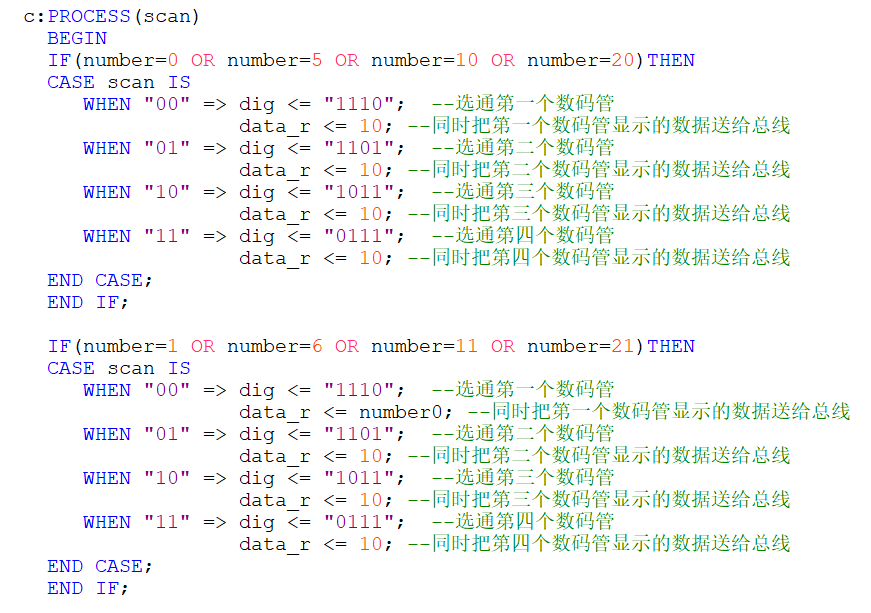
按键长按消抖及产生模块



数码管分频扫描模块



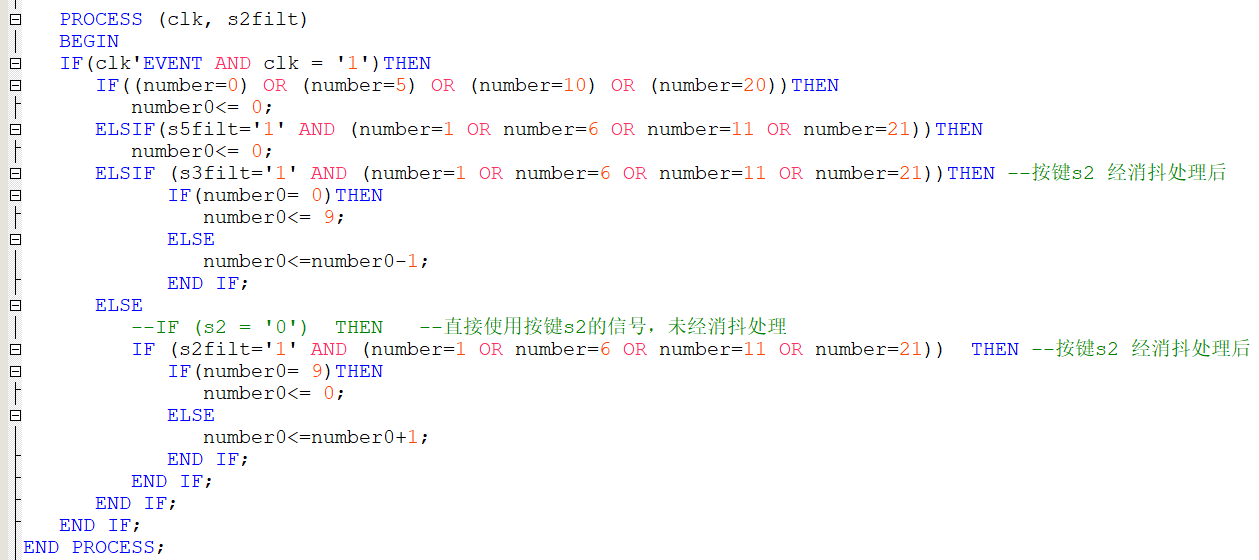
数码管显示模块（部分）



模式选择模块（包含密码对比）（部分）



控制加减模块



**五、实验结果**

****

开机显示----

**图片包含 电子产品

描述已自动生成**

按set开始输入第一位密码

**图片包含 电子产品

描述已自动生成**

按↑可以使数字加1

**图片包含 电子产品

描述已自动生成**

按↓可以使数字减一

**图片包含 电子产品

描述已自动生成**

按←使密码左移同时输入下一位密码

****

当四位密码输入完成按set

**图片包含 电子产品

描述已自动生成**

密码正确显示OPEN

**图片包含 电子产品

描述已自动生成**

若输入三次密码错误显示FFFF

**图片包含 电子产品

描述已自动生成**

密码正确长按set修改密码

**图片包含 电子产品

描述已自动生成**

输入新密码之后长按进行设置

**图片包含 电子产品

描述已自动生成**

长按clr上锁

**图片包含 电子产品

描述已自动生成**

短按clr清零当前输入的数字

**六、使用说明**

五个按键从左到右依次为：set、↑、↓、←、clr

图片包含 屏幕截图

描述已自动生成

**七、结论**

功能123和其他要求都能实现，但是功能2里的led闪烁不能成功。所有功能的实现是基于对老师给的例子的程序理解，本综合设计的程序是在例子的程序基础上进行修改完善得出。通过本次课程设计，本人对基于VHDL的设计有了更深刻的理解，同时在设计时不断修改错误和完善程序以及查找资料的过程中，本人的查找修改bug能力得到了很大提升。

**八、程序设计**

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY Lock\_Combined IS

PORT( clk: IN STD\_LOGIC;

s6, s2, s3, s4, s5: IN STD\_LOGIC;

led: OUT STD\_LOGIC; --'1'点亮，'0'熄灭

seg: OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0); --数码管的段数据，包括小数点共8段LED，低电平点亮

dig: OUT STD\_LOGIC\_VECTOR(3 DOWNTO 0) --数码管的选通信号,共4个数码管，低电平选通

);

END Lock\_Combined;

ARCHITECTURE behav OF Lock\_Combined IS

--数码管扫描显示用到的信号

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

SIGNAL divclk: STD\_LOGIC; --由基准时钟分频得到的扫描时钟

SIGNAL data\_r: INTEGER RANGE 0 TO 20;

SIGNAL number: INTEGER RANGE 0 TO 30; --按键s6消抖及计数用的信号，

SIGNAL s6filt: STD\_LOGIC; --按键s6消抖后得到的信号

SIGNAL s6filt2: STD\_LOGIC; --按键s6消抖后得到的信号

SIGNAL s6cnt: INTEGER RANGE 0 TO 50000000; --用于对s6按键输入有效时间进行计数

SIGNAL s6cnt2: INTEGER RANGE 0 TO 500000000; --用于对s6按键输入有效时间进行计数

SIGNAL number0: INTEGER RANGE 0 TO 9; --按键消抖及计数用的信号，数码管0显示的数字

SIGNAL number1: INTEGER RANGE 0 TO 9; --按键消抖及计数用的信号，数码管1显示的数字

SIGNAL number2: INTEGER RANGE 0 TO 9; --按键消抖及计数用的信号，数码管2显示的数字

SIGNAL number3: INTEGER RANGE 0 TO 9; --按键消抖及计数用的信号，数码管3显示的数字

SIGNAL s2filt: STD\_LOGIC; --按键s2消抖后得到的信号

SIGNAL s2cnt: INTEGER RANGE 0 TO 50000000; --用于对s2按键输入有效时间进行计数

SIGNAL s3filt: STD\_LOGIC; --按键s2消抖后得到的信号

SIGNAL s3cnt: INTEGER RANGE 0 TO 50000000; --用于对s2按键输入有效时间进行计数

SIGNAL s4filt: STD\_LOGIC; --按键s2消抖后得到的信号

SIGNAL s4cnt: INTEGER RANGE 0 TO 50000000; --用于对s2按键输入有效时间进行计数

SIGNAL s5filt: STD\_LOGIC; --按键s2消抖后得到的信号

SIGNAL s5filt2: STD\_LOGIC; --按键s2消抖后得到的信号

SIGNAL s5cnt: INTEGER RANGE 0 TO 50000000; --用于对s2按键输入有效时间进行计数

SIGNAL s5cnt2: INTEGER RANGE 0 TO 500000000; --用于对s2按键输入有效时间进行计数

SIGNAL keyword0: INTEGER RANGE 0 TO 9:=1;

SIGNAL keyword1: INTEGER RANGE 0 TO 9:=1;

SIGNAL keyword2: INTEGER RANGE 0 TO 9:=1;

SIGNAL keyword3: INTEGER RANGE 0 TO 9:=1;

BEGIN

--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--数码管译码显示部分

--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--分频得到扫描时钟

a: PROCESS(clk)

CONSTANT clkfrq: INTEGER:=50000000; --设基准时钟频率clkfrq为50Mhz

constant scanfrq: INTEGER:=50; --设扫描频率scanfrq为50Hz

VARIABLE count: INTEGER RANGE 0 TO 50000000;

BEGIN

IF(clk'EVENT AND clk = '1')THEN

IF(count = clkfrq /(scanfrq\*4)-1)THEN -- 设扫描频率scanfrq为50Hz，基准时钟频率clkfrq为50Mhz

count := 0;

divclk <= '1'; --得到扫描时钟

ELSE

count := count + 1;

divclk <= '0';

END IF;

END IF;

END PROCESS;

b:PROCESS(divclk,s4)

BEGIN

IF(s5 = '0')THEN

scan <= "00";

ELSIF(divclk'EVENT AND divclk = '1')THEN

scan <= scan + '1';

END IF;

END PROCESS;

c:PROCESS(scan)

BEGIN

IF(number=0 OR number=5 OR number=10 OR number=20)THEN

CASE scan IS

WHEN "00" => dig <= "1110"; --选通第一个数码管

data\_r <= 10; --同时把第一个数码管显示的数据送给总线

WHEN "01" => dig <= "1101"; --选通第二个数码管

data\_r <= 10; --同时把第二个数码管显示的数据送给总线

WHEN "10" => dig <= "1011"; --选通第三个数码管

data\_r <= 10; --同时把第三个数码管显示的数据送给总线

WHEN "11" => dig <= "0111"; --选通第四个数码管

data\_r <= 10; --同时把第四个数码管显示的数据送给总线

END CASE;

END IF;

IF(number=1 OR number=6 OR number=11 OR number=21)THEN

CASE scan IS

WHEN "00" => dig <= "1110"; --选通第一个数码管

data\_r <= number0; --同时把第一个数码管显示的数据送给总线

WHEN "01" => dig <= "1101"; --选通第二个数码管

data\_r <= 10; --同时把第二个数码管显示的数据送给总线

WHEN "10" => dig <= "1011"; --选通第三个数码管

data\_r <= 10; --同时把第三个数码管显示的数据送给总线

WHEN "11" => dig <= "0111"; --选通第四个数码管

data\_r <= 10; --同时把第四个数码管显示的数据送给总线

END CASE;

END IF;

IF(number=2 OR number=7 OR number=12 OR number=22)THEN

CASE scan IS

WHEN "00" => dig <= "1110"; --选通第一个数码管

data\_r <= number1; --同时把第一个数码管显示的数据送给总线

WHEN "01" => dig <= "1101"; --选通第二个数码管

data\_r <= number0; --同时把第二个数码管显示的数据送给总线

WHEN "10" => dig <= "1011"; --选通第三个数码管

data\_r <= 10; --同时把第三个数码管显示的数据送给总线

WHEN "11" => dig <= "0111"; --选通第四个数码管

data\_r <= 10; --同时把第四个数码管显示的数据送给总线

END CASE;

END IF;

IF(number=3 OR number=8 OR number=13 OR number=23)THEN

CASE scan IS

WHEN "00" => dig <= "1110"; --选通第一个数码管

data\_r <= number2; --同时把第一个数码管显示的数据送给总线

WHEN "01" => dig <= "1101"; --选通第二个数码管

data\_r <= number1; --同时把第二个数码管显示的数据送给总线

WHEN "10" => dig <= "1011"; --选通第三个数码管

data\_r <= number0; --同时把第三个数码管显示的数据送给总线

WHEN "11" => dig <= "0111"; --选通第四个数码管

data\_r <= 10; --同时把第四个数码管显示的数据送给总线

END CASE;

END IF;

IF(number=4 OR number=9 OR number=14 OR number=24)THEN

CASE scan IS

WHEN "00" => dig <= "1110"; --选通第一个数码管

data\_r <= number3; --同时把第一个数码管显示的数据送给总线

WHEN "01" => dig <= "1101"; --选通第二个数码管

data\_r <= number2; --同时把第二个数码管显示的数据送给总线

WHEN "10" => dig <= "1011"; --选通第三个数码管

data\_r <= number1; --同时把第三个数码管显示的数据送给总线

WHEN "11" => dig <= "0111"; --选通第四个数码管

data\_r <= number0; --同时把第四个数码管显示的数据送给总线

END CASE;

END IF;

IF(number=18)THEN

CASE scan IS

WHEN "00" => dig <= "1110"; --选通第一个数码管

data\_r <= 13; --同时把第一个数码管显示的数据送给总线

WHEN "01" => dig <= "1101"; --选通第二个数码管

data\_r <= 12; --同时把第二个数码管显示的数据送给总线

WHEN "10" => dig <= "1011"; --选通第三个数码管

data\_r <= 11; --同时把第三个数码管显示的数据送给总线

WHEN "11" => dig <= "0111"; --选通第四个数码管

data\_r <= 0; --同时把第四个数码管显示的数据送给总线

END CASE;

END IF;

IF(number=19)THEN

CASE scan IS

WHEN "00" => dig <= "1110"; --选通第一个数码管

data\_r <= 14; --同时把第一个数码管显示的数据送给总线

WHEN "01" => dig <= "1101"; --选通第二个数码管

data\_r <= 14; --同时把第二个数码管显示的数据送给总线

WHEN "10" => dig <= "1011"; --选通第三个数码管

data\_r <= 14; --同时把第三个数码管显示的数据送给总线

WHEN "11" => dig <= "0111"; --选通第四个数码管

data\_r <= 14; --同时把第四个数码管显示的数据送给总线

END CASE;

END IF;

END PROCESS;

PROCESS(data\_r)

BEGIN

CASE data\_r IS -- hgfedcba,0亮

WHEN 0 => seg <= "11000000"; -- 0

WHEN 1 => seg <= "11111001"; -- 1

WHEN 2 => seg <= "10100100"; -- 2

WHEN 3 => seg <= "10110000"; -- 3

WHEN 4 => seg <= "10011001"; -- 4

WHEN 5 => seg <= "10010010"; -- 5

WHEN 6 => seg <= "10000010"; -- 6

WHEN 7 => seg <= "11111000"; -- 7

WHEN 8 => seg <= "10000000"; -- 8

WHEN 9 => seg <= "10010000"; -- 9

WHEN 10=> seg <= "10111111"; -- 显示-

WHEN 11=> seg <= "10001100"; -- P

WHEN 12=> seg <= "10000110"; -- E

WHEN 13=> seg <= "11001000"; -- N

WHEN 14=> seg <= "10001110"; -- F

WHEN OTHERS =>NULL;

END CASE;

END PROCESS;

PROCESS (clk, s6filt)

BEGIN

IF(clk'EVENT AND clk = '1')THEN --按键s6

IF(s5filt2 = '1')THEN

keyword0<=1; keyword1<=1; keyword2<=1; keyword3<=1;

number<=0;

ELSIF(s6filt2='1' AND number=18)THEN

number<=20;

ELSIF(s6filt2='1' AND number=24)THEN

number<=0;

ELSE

--IF (s2 = '0') THEN --直接使用按键s2的信号，未经消抖处理

IF (s6filt='1' OR s4filt='1') THEN --按键s6 经消抖处理后

IF(number= 4)THEN

IF((number0=keyword0) AND (number1=keyword1) AND (number2=keyword1) AND (number3=keyword1))THEN

number<=18;

ELSE

number<=number+1;

END IF;

ELSIF(number=9)THEN

IF((number0=keyword0) AND (number1=keyword1) AND (number2=keyword1) AND (number3=keyword1))THEN

number<=18;

ELSE

number<=number+1;

END IF;

ELSIF(number=14)THEN

IF((number0=keyword0) AND (number1=keyword1) AND (number2=keyword1) AND (number3=keyword1))THEN

number<=18;

ELSE

number<=19;

END IF;

ELSIF(number=18)THEN

number<=18;

ELSIF(number=19)THEN

number<=19;

ELSIF(number=24)THEN

keyword0<=number0; keyword1<=number1; keyword2<=number2; keyword3<=number3;

ELSE

number<=number+1;

END IF;

END IF;

END IF;

END IF;

END PROCESS;

PROCESS (clk, s2filt)

BEGIN

IF(clk'EVENT AND clk = '1')THEN

IF((number=0) OR (number=5) OR (number=10) OR (number=20))THEN

number0<= 0;

ELSIF(s5filt='1' AND (number=1 OR number=6 OR number=11 OR number=21))THEN

number0<= 0;

ELSIF (s3filt='1' AND (number=1 OR number=6 OR number=11 OR number=21))THEN --按键s2 经消抖处理后

IF(number0= 0)THEN

number0<= 9;

ELSE

number0<=number0-1;

END IF;

ELSE

--IF (s2 = '0') THEN --直接使用按键s2的信号，未经消抖处理

IF (s2filt='1' AND (number=1 OR number=6 OR number=11 OR number=21)) THEN --按键s2 经消抖处理后

IF(number0= 9)THEN

number0<= 0;

ELSE

number0<=number0+1;

END IF;

END IF;

END IF;

END IF;

END PROCESS;

PROCESS (clk, s2filt)

BEGIN

IF(clk'EVENT AND clk = '1')THEN

IF((number=0) OR (number=5) OR (number=10) OR (number=20))THEN

number1<= 0;

ELSIF(s5filt='1' AND (number=2 OR number=7 OR number=12 OR number=22))THEN

number1<= 0;

ELSIF (s3filt='1' AND (number=2 OR number=7 OR number=12 OR number=22)) THEN --按键s2 经消抖处理后

IF(number1= 0)THEN

number1<= 9;

ELSE

number1<=number1-1;

END IF;

ELSE

--IF (s2 = '0') THEN --直接使用按键s2的信号，未经消抖处理

IF (s2filt='1' AND (number=2 OR number=7 OR number=12 OR number=22)) THEN --按键s2 经消抖处理后

IF(number1= 9)THEN

number1<= 0;

ELSE

number1<=number1+1;

END IF;

END IF;

END IF;

END IF;

END PROCESS;

PROCESS (clk, s2filt)

BEGIN

IF(clk'EVENT AND clk = '1')THEN

IF((number=0) OR (number=5) OR (number=10) OR (number=20))THEN

number2<= 0;

ELSIF(s5filt='1' AND (number=3 OR number=8 OR number=13 OR number=23))THEN

number2<= 0;

ELSIF (s3filt='1' AND (number=3 OR number=8 OR number=13 OR number=23)) THEN --按键s2 经消抖处理后

IF(number2= 0)THEN

number2<= 9;

ELSE

number2<=number2-1;

END IF;

ELSE

--IF (s2 = '0') THEN --直接使用按键s2的信号，未经消抖处理

IF (s2filt='1' AND (number=3 OR number=8 OR number=13 OR number=23)) THEN --按键s2 经消抖处理后

IF(number2= 9)THEN

number2<= 0;

ELSE

number2<=number2+1;

END IF;

END IF;

END IF;

END IF;

END PROCESS;

PROCESS (clk, s2filt)

BEGIN

IF(clk'EVENT AND clk = '1')THEN

IF((number=0) OR (number=5) OR (number=10) OR (number=20))THEN

number3<= 0;

ELSIF(s5filt='1' AND (number=4 OR number=9 OR number=14 OR number=24))THEN

number3<= 0;

ELSIF (s3filt='1' AND (number=4 OR number=9 OR number=14 OR number=24)) THEN --按键s2 经消抖处理后

IF(number3= 0)THEN

number3<= 9;

ELSE

number3<=number3-1;

END IF;

ELSE

--IF (s2 = '0') THEN --直接使用按键s2的信号，未经消抖处理

IF (s2filt='1' AND (number=4 OR number=9 OR number=14 OR number=24)) THEN --按键s2 经消抖处理后

IF(number3= 9)THEN

number3<= 0;

ELSE

number3<=number3+1;

END IF;

END IF;

END IF;

END IF;

END PROCESS;

--按键消抖

--设计思路：当按键被按下时，输入低电平，开始计数，一直计数到N，期间一旦按键输入高电平，即清零计数值。

--最后计数值若达到预设值N，表明按键已稳定按下，发出一个脉冲。

PROCESS (s6)

CONSTANT N : INTEGER:= 5000000; --消抖时间，对于50Mhz的基准时钟，这相当于0.1S

BEGIN

IF(clk'EVENT AND clk='1')THEN

IF(s6='0')THEN --当s6 输入低电平，即按键按下

IF(s6cnt /= N)THEN --一直计数到N

s6cnt <= s6cnt + 1;

END IF;

IF(s6cnt=N-1)THEN --最后一个计数时输出s2filt脉冲

s6filt <= '1';

ELSE

s6filt <= '0';

END IF;

ELSE --若s2 输入高电平，表明按键被释放

s6cnt <= 0;

END IF;

END IF;

END PROCESS;

PROCESS (s6)

CONSTANT M : INTEGER:= 150000000; --3秒

BEGIN

IF(clk'EVENT AND clk='1')THEN

IF(s6='0')THEN --当s6 输入低电平，即按键按下

IF(s6cnt2 /= M)THEN --一直计数到M

s6cnt2 <= s6cnt2 + 1;

END IF;

IF(s6cnt2=M-1)THEN --最后一个计数时输出s6filt2脉冲

s6filt2 <= '1';

ELSE

s6filt2 <= '0';

END IF;

ELSE --若s6 输入高电平，表明按键被释放

s6cnt2 <= 0;

END IF;

END IF;

END PROCESS;

PROCESS (s2)

CONSTANT N : INTEGER:= 5000000; --消抖时间，对于50Mhz的基准时钟，这相当于0.1S

BEGIN

IF(clk'EVENT AND clk='1')THEN

IF(s2='0')THEN --当s2 输入低电平，即按键按下

IF(s2cnt /= N)THEN --一直计数到N

s2cnt <= s2cnt + 1;

END IF;

IF(s2cnt=N-1)THEN --最后一个计数时输出s2filt脉冲

s2filt <= '1';

ELSE

s2filt <= '0';

END IF;

ELSE --若s2 输入高电平，表明按键被释放

s2cnt <= 0;

END IF;

END IF;

END PROCESS;

PROCESS (s3)

CONSTANT N : INTEGER:= 5000000; --消抖时间，对于50Mhz的基准时钟，这相当于0.1S

BEGIN

IF(clk'EVENT AND clk='1')THEN

IF(s3='0')THEN --当s2 输入低电平，即按键按下

IF(s3cnt /= N)THEN --一直计数到N

s3cnt <= s3cnt + 1;

END IF;

IF(s3cnt=N-1)THEN --最后一个计数时输出s2filt脉冲

s3filt <= '1';

ELSE

s3filt <= '0';

END IF;

ELSE --若s2 输入高电平，表明按键被释放

s3cnt <= 0;

END IF;

END IF;

END PROCESS;

PROCESS (s4)

CONSTANT N : INTEGER:= 5000000; --消抖时间，对于50Mhz的基准时钟，这相当于0.1S

BEGIN

IF(clk'EVENT AND clk='1')THEN

IF(s4='0')THEN --当s4 输入低电平，即按键按下

IF(s4cnt /= N)THEN --一直计数到N

s4cnt <= s4cnt + 1;

END IF;

IF(s4cnt=N-1)THEN --最后一个计数时输出s2filt脉冲

s4filt <= '1';

ELSE

s4filt <= '0';

END IF;

ELSE --若s4 输入高电平，表明按键被释放

s4cnt <= 0;

END IF;

END IF;

END PROCESS;

PROCESS (s5)

CONSTANT N : INTEGER:= 5000000; --消抖时间，对于50Mhz的基准时钟，这相当于0.1S

BEGIN

IF(clk'EVENT AND clk='1')THEN

IF(s5='0')THEN --当s5 输入低电平，即按键按下

IF(s5cnt /= N)THEN --一直计数到N

s5cnt <= s5cnt + 1;

END IF;

IF(s5cnt=N-1)THEN --最后一个计数时输出s2filt脉冲

s5filt <= '1';

ELSE

s5filt <= '0';

END IF;

ELSE --若s5 输入高电平，表明按键被释放

s5cnt <= 0;

END IF;

END IF;

END PROCESS;

PROCESS (s5)

CONSTANT M : INTEGER:= 150000000; --3秒

BEGIN

IF(clk'EVENT AND clk='1')THEN

IF(s5='0')THEN --当s5 输入低电平，即按键按下

IF(s5cnt2 /= M)THEN --一直计数到N

s5cnt2 <= s5cnt2 + 1;

END IF;

IF(s5cnt2=M-1)THEN --最后一个计数时输出s2filt脉冲

s5filt2 <= '1';

ELSE

s5filt2 <= '0';

END IF;

ELSE --若s2 输入高电平，表明按键被释放

s5cnt2 <= 0;

END IF;

END IF;

END PROCESS;

END behav;