简易电子琴的设计

小组成员：靳婷婷
刘颖
韦飞燕

背景：

- 巩固和运用所学课程，理论联系实际，提高分析、解决实际问题的独立工作能力。

  通过对一个简易的八音符电子琴的设计，加深对VHDL数字系统设计方面的了解，熟悉VHDL数字系统设计制作与调试的方法。

- 小组成员比较喜欢音乐，本学期的单片机和VHDL的学习提供给我们这样一个能自己动手制作简易电子琴的机会。

主要设计内容：

- 设计一个简易的八音符电子琴，
  它可通过**按键输入**来**控制音响**。
- 演奏时可以选择是**手动演奏**（由键盘输入）
  还是**自动演奏**已存入的乐曲。

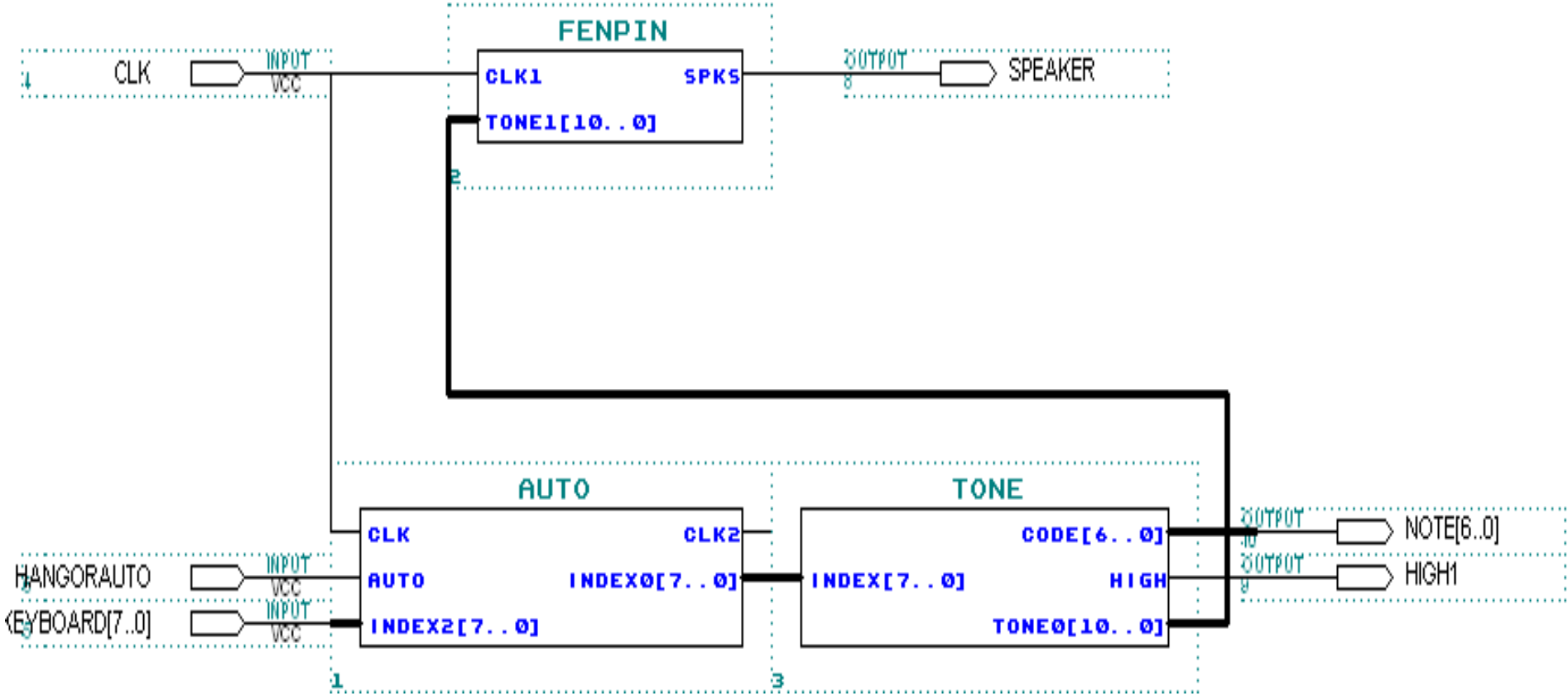
前车之鉴：

电子琴的设计包括七个模块：

- 弹奏模块
- 分频模块
- 自动演奏模块
- 查表及显示模块
- 存储模块
- 七段数码管显示模块
- 点阵的显示模块

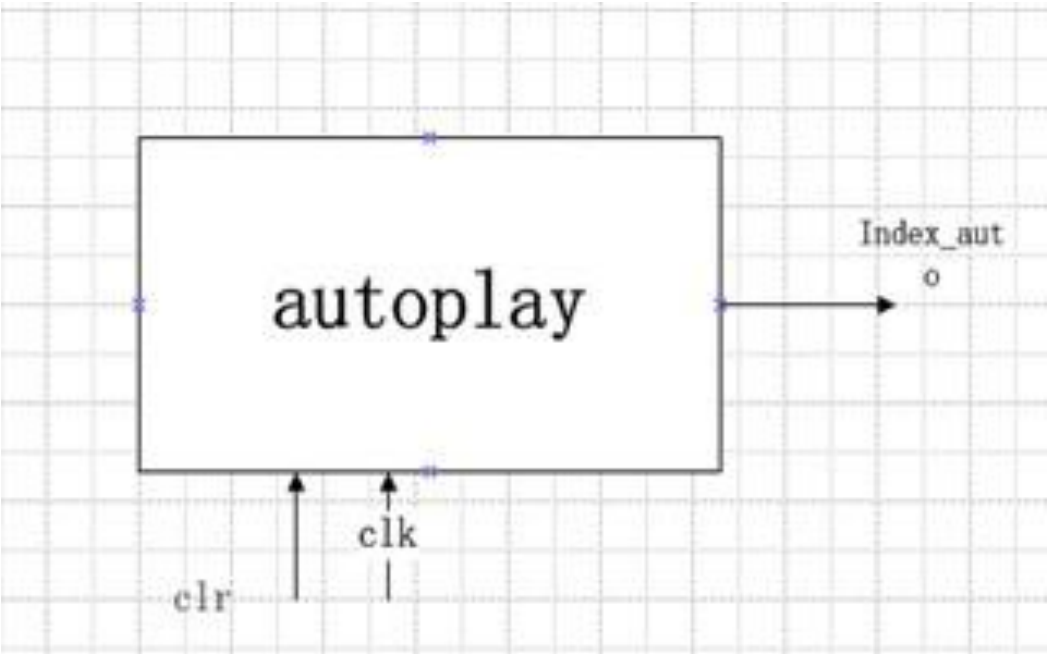
设计规划:

- 根据设计要求，采用**自顶向下**的设计方法，系统的整体组装设计原理图如图所示，它由乐曲**自动演奏模块**、**音调发生模块**和**数控分频模块**三部分组成。



自动演奏模块：

- 接收32MHz的时钟信号，产生8位发声控制输入信号
- 输出index_auto。作为节拍。将要自动演奏的歌曲预先写为index_auto的格式,将index_auto输出。这里需要用到计数器，此计数器的长度由演奏的歌曲长度而定。



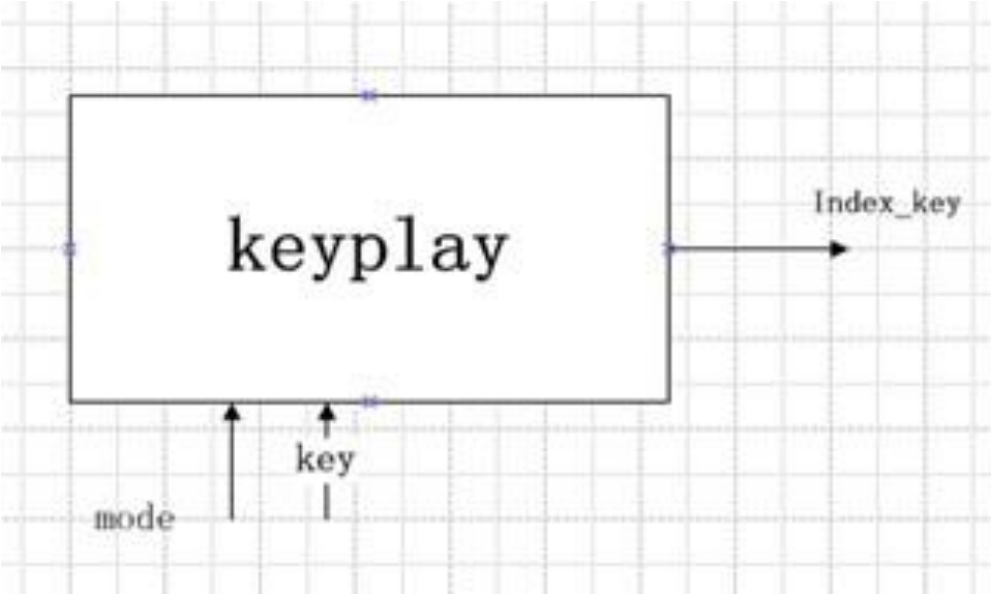
自动演奏模块

```
ENTITY AUTO IS
PORT ( CLK :      IN STD_LOGIC;
       AUTO :     IN STD_LOGIC;
       CLK2 :     BUFFER STD_LOGIC;
       INDEX2 :   IN STD_LOGIC_VECTOR(7   DOWNTO 0);
       INDEX0 :   OUT STD_LOGIC_VECTOR(7 DOWNTO 0));
END AUTO;
ARCHITECTURE BEHAVIORAL OF AUTO IS
SIGNAL COUNT0: INTEGER RANGE 0 TO 31;
BEGIN
PULSE0 :PROCESS(CLK,AUTO)
VARIABLE COUNT :INTEGER RANGE 0 TO 8;
BEGIN
IF AUTO ='1' THEN
COUNT = 0;CLK2<='0';
ELSIF(CLK'EVENT AND CLK ='1')THEN
COUNT =COUNT +1;
IF COUNT =4 THEN
CLK2 <='1';
ELSIF COUNT =8 THEN
CLK2<='0'; COUNT:=0;
```

```
END IF ;
END IF ;
END PROCESS;
MUSIC:PROCESS(CLK2)
BEGIN
IF (CLK2'EVENT AND CLK2='1')THEN
IF (COUNT0=31)THEN
COUNT0<=0;
ELSE
COUNT0<=COUNT0+1;
END IF ;
END IF ;
END PROCESS;
COM1:PROCESS(COUNT0,AUTO,INDEX2)
BEGIN
IF AUTO ='0' THEN
CASE COUNT0 IS
WHEN 0=>INDEX0<="00000100";  --3
WHEN 1=>INDEX0<="00000100";  --3
      ……
WHEN 30=>INDEX0<="00000010";  --2
WHEN 31=>INDEX0<="00000010";  --2
WHEN OTHERS =>NULL;
END CASE;
ELSE INDEX0<=INDEX2;
END IF;
END PROCESS;
END BEHAVIORAL;
```

弹奏模块设计：

- 弹奏模块：根据按键动作index，
  相应产生指示音调的高低音选项以及音调code。



弹奏模块:

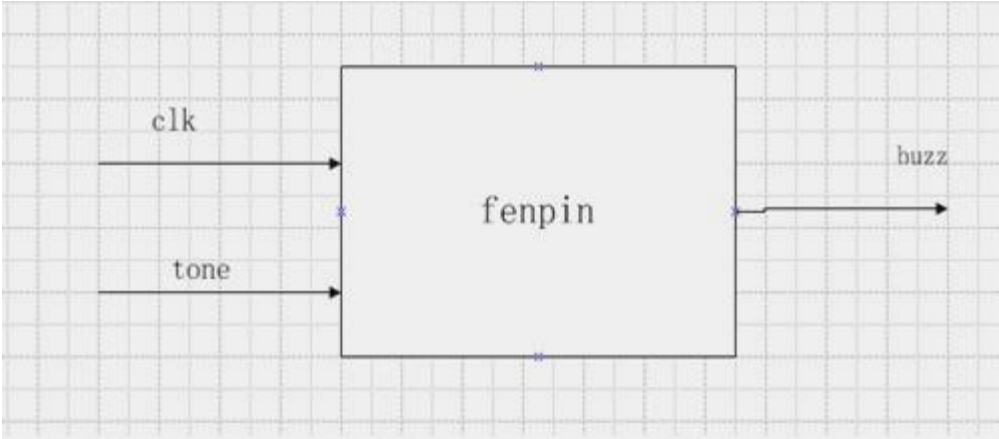
表 1 简谱中的音名与频率的关系

音名	频率（Hz）	音名	频率（Hz）	音名	频率（Hz）
低音 1	261.63	中音 1	523.25	高音 1	1046.50
低音 2	293.67	中音 2	587.33	高音 2	1174.66
低音 3	329.63	中音 3	659.25	高音 3	1318.51
低音 4	349.23	中音 4	698.46	高音 4	1396.92
低音 5	391.99	中音 5	783.99	高音 5	1567.98
低音 6	440	中音 6	880	高音 6	1760
低音 7	493.88	中音 7	987.76	高音 7	1975.52

```
ENTITY TONE IS
  PORT (INDEX:    IN STD_LOGIC_VECTOR(7 DOWNTO 0);
        CODE:     OUT STD_LOGIC_VECTOR(6 DOWNTO 0
        HIGH:     OUT STD_LOGIC;
        TONE0:    OUT INTEGER RANGE 0 TO 2047);
END TONE;
ARCHITECTURE ART OF TONE IS
BEGIN
SEARCH : PROCESS(INDEX)
BEGIN
CASE INDEX IS
WHEN "00000001"=>TONE0 <=773;CODE<="1001111";HIGH<='0';
WHEN "00000010"=>TONE0 <=912;CODE<="0010010";HIGH<='0';
WHEN "00000100"=>TONE0 <=1036;CODE<="0000110";HIGH<='0';
WHEN "00001000"=>TONE0 <=1116;CODE<="1001100";HIGH<='0';
WHEN "00010000"=>TONE0 <=1197;CODE<="0100100";HIGH<='1';
WHEN "00100000"=>TONE0 <=1290;CODE<="0100000";HIGH<='1';
WHEN "01000000"=>TONE0 <=1372;CODE<="0001111";HIGH<='1';
WHEN "10000000"=>TONE0 <=1410;CODE<="0000000";HIGH<='1';
WHEN OTHERS =>TONE0<=2047;CODE<="0000001";HIGH<='1';
END CASE;
END PROCESS;
END ART;
```

分频模块设计：

- 接收分频系数 tone，并据此分频，将对应频率的信号输出给扬声器供其发声。
- 输入分频系数 tone。
- (1)设置内部信号count用于计数，clk_data作为分频结果。,每次clk上升沿检测count是否等于tone，相等则把count清零，并使clk_data翻转，否则count自增1。
- (2)把clk_data赋给输出信号，由该信号驱动扬声器发声。



分频模块：

```
ENTITY FENPIN IS
 PORT(CLK1: IN STD_LOGIC;
          TONE1: IN INTEGER RANGE 0 TO 2047;
           SPKS:    OUT STD_LOGIC);
END ENTITY FENPIN;
ARCHITECTURE ART OF FENPIN IS
SIGNAL   PRECLK:STD_LOGIC;
SIGNAL   FULLSPKS:STD_LOGIC;
BEGIN
PROCESS(CLK1)
VARIABLE COUNT:INTEGER RANGE 0 TO 8;
BEGIN
IF (CLK1'EVENT AND CLK1='1')THEN
COUNT=COUNT +1;
IF COUNT=2 THEN
PRECLK<='1';
ELSIF COUNT =4 THEN
PRECLK<='0';COUNT=0;
END IF ;
END IF ;
END PROCESS;
PROCESS(PRECLK,TONE1)
VARIABLE COUNT11:INTEGER RANGE 0 TO 2047;
```

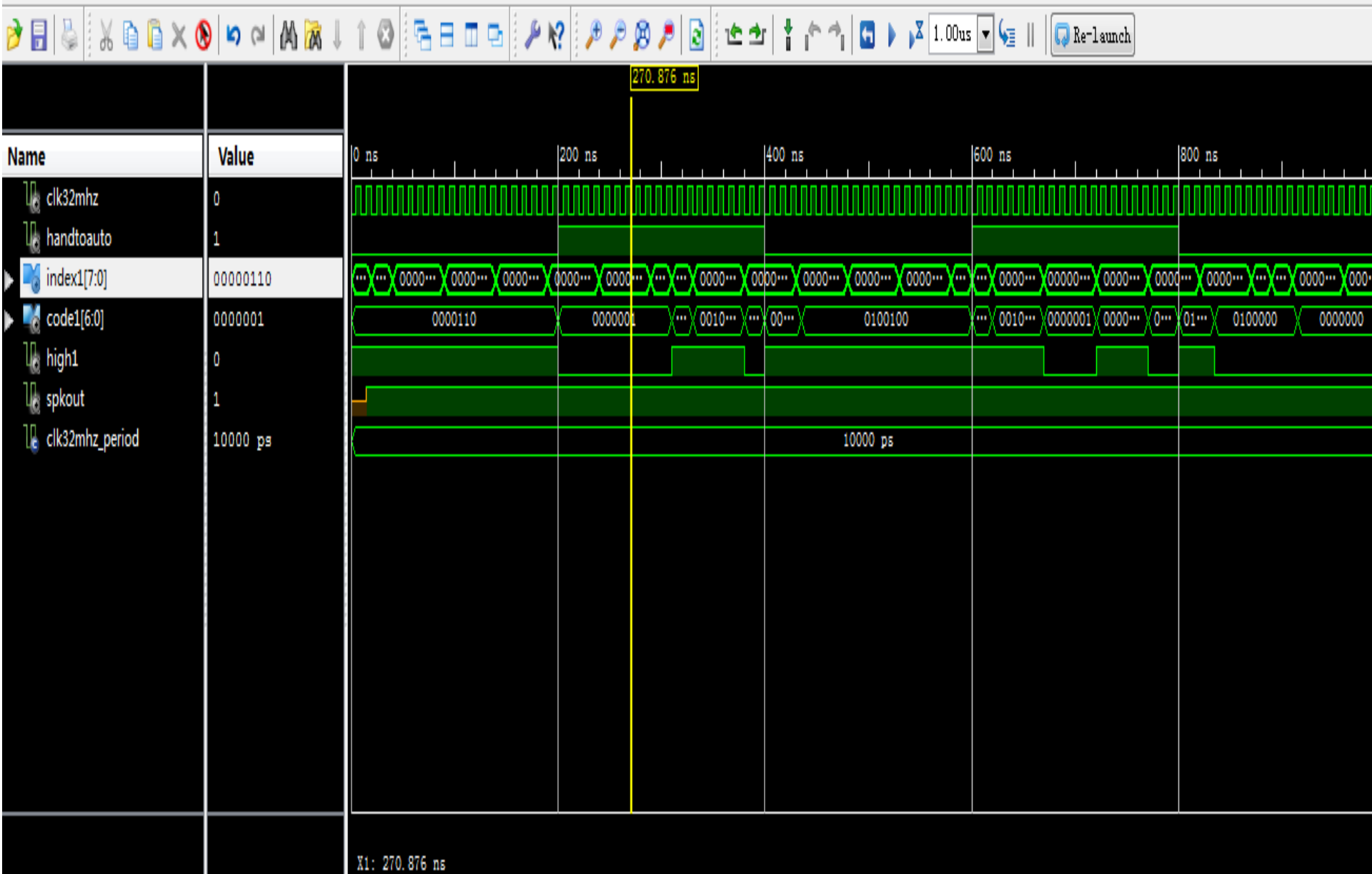
```
BEGIN
IF (PRECLK'EVENT AND PRECLK='1')THEN
IF COUNT11<TONE1 THEN
COUNT11=COUNT11+1;FULLSPKS<='1';
ELSE
COUNT11=0;FULLSPKS<='0';
END IF ;
END IF ;
END PROCESS;
PROCESS(FULLSPKS)
VARIABLE  COUNT2 :STD_LOGIC:='0';
BEGIN
IF (FULLSPKS'EVENT AND FULLSPKS='1')THEN
COUNT2=NOT COUNT2;
IF COUNT2='1'THEN
SPKS<='1';
ELSE
SPKS<='0';
END IF ;
END IF;
END PROCESS;
END ART;
```

顶层结构（例化）：

```
ENTITY DIANZIQIN IS
PORT(CLK32MHZ:  IN STD_LOGIC;
        HANDTOAUTO:IN STD_LOGIC;
        CODE1: OUT STD_LOGIC_VECTOR(6 DOWNTO 0);--音符信号
        INDEX1: IN STD_LOGIC_VECTOR(7 DOWNTO 0);--键盘输入信号
        HIGH1: OUT STD_LOGIC;--高低音节信号
        SPKOUT: OUT STD_LOGIC);--音频信号
END;
ARCHITECTURE ART OF DIANZIQIN IS
COMPONENT AUTO
PORT(CLK:   IN STD_LOGIC;
         AUTO:   IN STD_LOGIC;
         INDEX2:   IN STD_LOGIC_VECTOR(7 DOWNTO 0);
         INDEX0: OUT STD_LOGIC_VECTOR(7 DOWNTO 0));
END COMPONENT;
```

```
COMPONENT TONE
PORT(INDEX:  IN STD_LOGIC_VECTOR(7 DOWNTO 0);
        CODE:   OUT STD_LOGIC_VECTOR(6 DOWNTO 0);
        HIGH: OUT STD_LOGIC;
       TONE0: OUT INTEGER RANGE 0 TO 2047);
END COMPONENT;
COMPONENT FENPIN
PORT(CLK1: IN STD_LOGIC;
        TONE1:IN INTEGER RANGE 0 TO 2047;
        SPKS: OUT STD_LOGIC);
END COMPONENT;
SIGNAL TONE2:INTEGER RANGE 0 TO 2047;
SIGNAL INDX:STD_LOGIC_VECTOR(7 DOWNTO 0);
BEGIN
U0:AUTO PORT MAP(CLK=>CLK32MHZ,INDEX2=>INDEX1,
   INDEX0=>INDX,AUTO=>HANDTOAUTO);
U1:TONEPORTMAP(INDEX=>INDX,TONE0=>TONE2,CODE=>CODE1,   HIGH=>HIGH1);
U2:PENPIN PORT MAP(CLK1=>CLK32MHZ,TONE1=>TONE2,SPKS=>SPKOUT);
END ART;
```

仿真结果：


1.00us
Re-launch
270.876 ns
Name
Value
clk32mhz
0
handtoauto
1
index1[7:0]
00000110
code1[6:0]
0000001
high1
0
spkout
1
clk32mhz_period
10000 ps
0 ns
200 ns
400 ns
600 ns
800 ns
0000110
0000001
0100100
0100000
0000000
10000 ps
X1: 270.876 ns

TIPS

- 很多从网上找来的源代码有较多的错误，找错时一定要认真仔细，尤其是空格问题 。
- 进行波形仿真时，我们得自己赋值，刚开始时赋完值进行波形仿真一直无法出现 ，找了很久才发现是未赋值时的波形仿真图没有关掉。
- 在波形仿真程序编写时，要对程序整体解读。
- 对vector型赋值时要注意。

File Edit View Project Source Process Tools Window Layout Help

Design

View: Implementation Simulation

Behavioral

Hierarchy

- Xiangmu1
- xc3s500e-4fg320
  - xiangmu12 - behavior (xiangmu
    - uut - DIANZIQIN - ART (xian
      - U0 - AUTO - BEHAVIORA
      - U1 - TONE - ART (xiangn
      - U2 - FENPIN - ART (xiang

Run Aborted: Simulation

Processes: xiangmu12 - behavior

- ISim Simulator
  - Behavioral Check Syntax
  - Simulate Behavioral Model

Start Design Files

```
97        -- insert stimulus here
98
99        wait;
100     end process;
101        handtoauto_process :process
102     begin
103        handtoauto <= '0';
104        wait for 200 ns;
105        handtoauto <= '1';
106        wait for 200 ns;
107     end process;
108
109    index1_process :process
110     begin
111
112        index1<="0000_0000";
113        wait for 20 ns;
114        index1<=B"0000_0001";
115        wait for 20 ns;
116        index1<=B"0000_0010";
117        wait for 50 ns;
118        index1<=B"0000_0011";
119        wait for 50 ns;
120        index1<=B"0000_0100";
121        wait for 50 ns;
122        index1<=B"0000_0101";
123        wait for 50 ns;
124        index1<=B"0000_0110";
```

Design Summary | xiangmu1.vhd

Errors

ERROR:HDLCompiler:304 - "F:/Xiangmu1/xiangmu12.vhd" Line 112: Character '_' is not in element type std_logic

ERROR:HDLCompiler:854 - "F:/Xiangmu1/xiangmu12.vhd" Line 38: Unit <behavior> ignored due to previous errors

File Edit View Project Source Process Tools Window Layout Help

Hierarchy

- Xiangmu1
- xc3s500e-4fg320
  - xiangmu12 - behavior (xiangmu
  - xiangmu22 - behavior (xiangmu

Run Aborted: Simulation

Processes: xiangmu22 - behavior

- ISim Simulator
  - Behavioral Check Syntax
  - Simulate Behavioral Model

```
95        wait for CLK32MHZ_period*10;
96
97        -- insert stimulus here
98
99        wait;
100     end process;
101     handtoauto_process :process
102     begin
103        handtoauto <= '0';
104        wait for 200 ns;
105        handtoauto <= '1';
106        wait for 200 ns;
107    wait for 10000 ns;
108   end process;
109   index1_process :process
110      begin
111        index1 <= B"0000_0000";
112        wait for 50 ns;
113        index1 <= B"0000_0001";
114        wait for 50 ns;
115        index1 <= B"0000_0010";
116        wait for 50 ns;
117        index1 <= B"0000_0011";
118        wait for 50 ns;
119      wait for 10000 ns;
120  end process;
121  end;
```

Start | Design | Files | xiangmu1.vhd | Design Summary (Synthesized) | xiangmu22.vhd

arnings

WARNING:HDLCompiler:1369 - "F:/Xiangmu1/xiangmu22.vhd" Line 109: Possible infinite loop; process does not have a wait statement

总结：

- 这个设计的基本是接触一门新的语言并加以应用，入手的速度比我的预料快，在以前编程的基础上，从接触到开始动手编程的时间得到了很大的缩短。
- VHDL的编程与其他语言的编程有着本质的不同，然而以往形成的旧编程习惯在VHDL编程中依然起着很大的作用。
- VHDL的设计关键是电路逻辑设计，而一个程序的关键是总体设计。
- 通过这个程序设计让我学会一种新的语言，对数字系统结构也有了更进一步的了解和认识，对以后的学习有很大帮助。希望其他人再做类似设计时有所借鉴。

致谢：

- 感谢老师提供这样一次机会，让我们更加熟悉地掌握了ISE软件，同时提高了大家的团队合作精神；
- 感谢在平台上分享软件下载地址的同学；
- 感谢帮助我们的老师和同学。

小组成员分工：

- **靳婷婷**：下载ISE软件，熟悉软件，网上找代码，进行仿真操作，制作PPT
- **刘颖**：下载ISE软件，熟悉理解代码，仿真过程中检查错误以及纠正,完善PPT，讲解
- **韦飞燕**：下载ISE软件，网上找代码，进行仿真操作,后期论文准备

参考文献：

网上VHDL电子琴源代码

《VHDL数字系统设计》李欣，张海燕. 科学出版社

《VHDL语言100例详解》北京理工大学ASIC研究所. 清华大学出版社

《VHDL程序设计》曾繁泰等. 清华大学出版社

《VHDL简明教程》王小军 . 清华大学出版社