实验七　梁祝音乐演奏实验

一、实验目的

1、 了解普通扬声器的工作原理。

2、 使用 FPGA 产生不同的音乐频率。

3、 进一步体验 FPGA 的灵活性。

二、硬件要求

1、375KHz 信号源。

2、FPGA　EP1K10TC100—3 主芯片。

3、扬声器。

三、实验原理

本实验是完成一小段音乐程序的开发，然后再用扬声器进行试听。下面主要介绍一下完成本实验的几个主要部分的工作原理。

音符的产生：音符的产生是利用计数器对输入的时钟信号进行分频，然后输出不同的频率来控制扬声器发不同的声音。计数器必须是模可变的计数器，也就是其初始计数值可变，这样便可以对其进行初始化，使其从不同的初始值开始计数，实现对输入时钟信号的不同分频。

节拍的产生：节拍也是利用计数器来实现，如果某一个音符需要维持的时间比较长，那么就可以在此计数器从计数值 A 到计数值 B 之间都维持该音符，很显然，A 和 B 之间的间隔越大，那么该音符维持的时间也就越长。

乐谱的存储：乐谱是一个固定的组合电路，根据不同的输入值，然后输出一个固定的值，该值就是音符产生计数器的分频的初始值。

适当的选择这些计数器和组合电路，便可完成不同的乐曲和不同节奏。

四、实验内容及步骤

本实验要完成的任务是设计一个驱动扬声器产生梁祝音乐的程序，设计步骤如下：

1、编写音乐输出的 VHDL 代码。

2、用 QuartusII 对其进行编译仿真。

3、在仿真确定无误后，选择芯片 ACEX1K EP1K10TC100－3。

4、给芯片进行管脚绑定，在此进行编译。

5、根据自己绑定的管脚，在实验箱上对扬声器接口和 FPGA 之间进行正确连线。

6、给目标板下载代码，观看实验结果。

五、实验连线

如果是调用的本书提供的 VHDL 代码，则实验连线如下：

Clk：时钟输入信号，接 375KHz 的时钟源。

Spk：输出，接扬声器部分的输入端。

六、实验 VHDL 部分代码

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
--------------------------------------------------------------------
entity music is
```

```
  port( clk      :  in      std_logic;        --Clock Signal
        spk      :  buffer  std_logic);       --speaker driver
end music;
-------------------------------------------------------------------
architecture behave of music is
  signal tone        :  std_logic_vector(10 downto 0);
  signal tone_count  :  std_logic_vector(10 downto 0);
  signal tone_index  :  integer range 0 to 15;
  signal clk10_count :  std_logic_vector(17 downto 0);
  signal time        :  integer range 0 to 150;
  signal clk10       :  std_logic;
  begin
  process(clk)  --generate 10hz clock signal
    begin
      if(clk'event and clk='1') then
        clk10_count<=clk10_count+1;
        if(clk10_count=16#3fff#) then
           clk10<=not clk10;
        end if;
      end if;
  end process;
  process(clk10)
    begin
      if(clk10'event and clk10='1') then
        if(time=150) then
           time<=0;
        else
           time<=time+1;
        end if;
      end if;
  end process;
  process(clk10)
    begin
      if(clk10'event and clk10='1') then
        case time is
          when 0=>tone_index<=3;
          when 1=>tone_index<=3;
          when 2=>tone_index<=3;
          when 3=>tone_index<=3;
          when 4=>tone_index<=3;
          when 5=>tone_index<=3;
          when 6=>tone_index<=3;
          when 7=>tone_index<=3;
```

```
          when 8=>tone_index<=3;
                 ……
          when 137=>tone_index<=5;
          when 138=>tone_index<=0;
          when 139=>tone_index<=0;--
          when others=>tone_index<=0;
        end case;
      end if;
  end process;
  process(tone_index)
    begin
      case tone_index is
         when 0=>tone<="11111111111";   --no output
         when 1=>tone<="01100000101";   --773
                        ……
         when 15=>tone<="11011000000";  --1728
         when others=>tone<="11111111111"; --others:no output
      end case;
  end process;
  process(clk) --control the frequence of the speaker
    begin
      if(clk'event and clk='1') then
         if(tone_count=16#7ff#) then
            tone_count<=tone;
            if(tone<2047) then
               spk<=not spk;
            end if;
         else
            tone_count<=tone_count+1;
         end if;
      end if;
  end process;
end behave;
```

七、实验报告

1、 了解乐曲节拍产生的过程，注意每一音符的节拍长短的变化是由什么控制的？

2、 改变时钟频率，看乐曲有什么改变？

3、 熟悉音乐编程的过程，填入新的乐曲。

4、 将两个或多个乐曲演奏电路合二为一，以一开关控制，可选择演奏不同的乐曲。