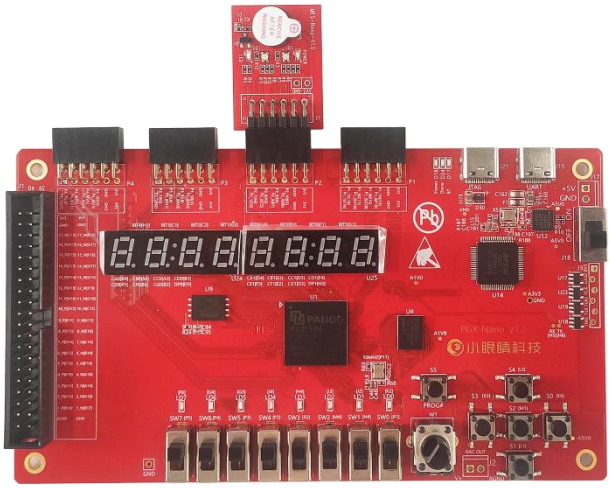
26. 蜂鸣器简易电子琴实验说明

26.1. 实验简介

实验使用“小眼睛科技”FPGA 开发板、PMOD 蜂鸣器模块完成简易电子琴设计。



26.2. 实验要求

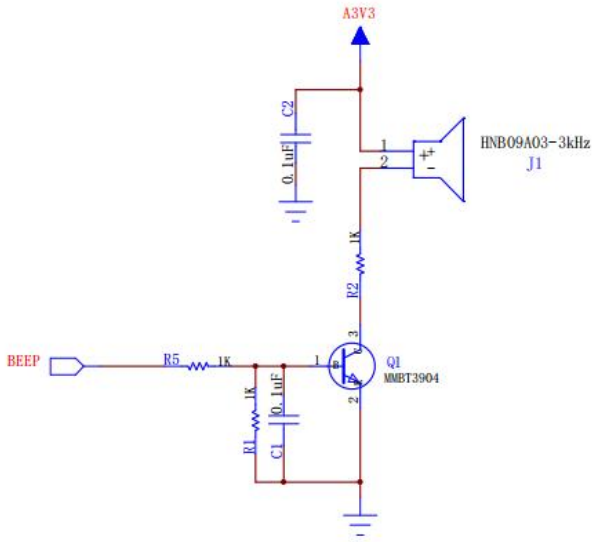
将 SW5 拨码开关**拨上**（此时保持 SW1、SW2、SW3 拨码开关为拨上状态）。

S1 按键**按下**发出 Do 音、S2 按键**按下**发出 Re 音、S3 按键**按下**发出 Mi 音、S4 按键**按下**发出 Fa 音、SW0 拨码开关**拨下**发出 So 音、SW1 拨码开关**拨下**发出 La 音、SW2 拨码开关**拨下**发出 Si 音。

26.3. 实验使用模块介绍

实验使用“小眼睛科技”公司的 PMOD 蜂鸣器模块，其中蜂鸣器为无源蜂鸣器。

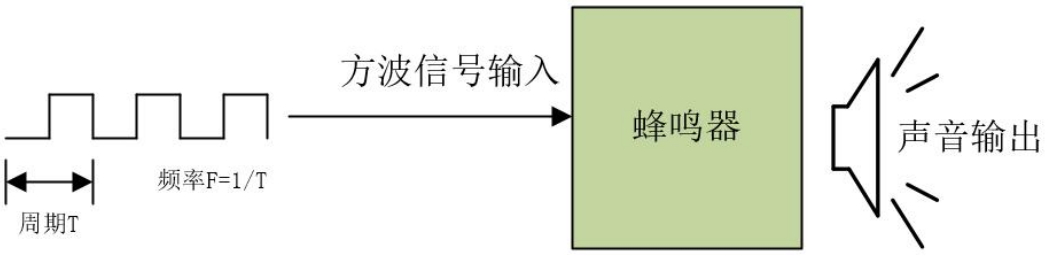
原理图如下图所示：震荡源通过三极管放大后输出给蜂鸣器，蜂鸣器接收到震荡源后，发出声响。



26.4. 实验原理

蜂鸣器是一种一体化结构的电子讯响器，分为有源蜂鸣器与无源蜂鸣器两种，其区别是有源蜂鸣器内置驱动电路即内置震荡源，通电即响，无源蜂鸣器使用外部驱动即外部震荡源，通电后需要给予震荡源才能发出响声，应注意的是这里的源指震荡源。

PMOD 蜂鸣器模块使用的是无源蜂鸣器，需要给予震荡源才能发出响声，实验中的震荡源即方波。



无源滤波器需要给予震荡源才能发出音响，当给予不同频率的音频源时，发出的音调也随之变化，如震荡源频率为 440Hz 时，蜂鸣器会发出 la 的音调；不同的音调对应的震荡源频率如下表所示：

C调各音符频率对照表		
音符	频率(Hz)	周期(μs)
低 1Do	262	3816
低 2Re	294	3401
低 3Mi	330	3030
低 4Fa	349	2865
低 5So	392	2551
低 6La	440	2272
低 7Si	494	2024
中 1Do	523	1912
中 2Re	587	1703
中 3Mi	659	1517
中 4Fa	698	1732
中 5So	784	1275
中 6La	880	1136
中 7Si	988	1012
高 1Do	1047	955
高 2Re	1175	851
高 3Mi	1319	758
高 4Fa	1397	751
高 5So	1568	637
高 6La	1760	568
高 7Si	1967	508

26.5. 实验源码设计

使用计数器对时钟进行分频，得到不同频率的方波，通过按键或拨码开关控制不同频率方波发送给蜂鸣器，使蜂鸣器发出不同的音调。

```
`timescale 1ns / 1ps
`define UD #1
module top_beep(
   input  wire clk ,
   input  wire rst_n ,
   input  [6:0]key ,
   input  [2:0]sw ,
   output wire  beep
   );
```

```
//////////////////////////////////////////////////////////////
//音符   低 1Do   低 2Re   低 3Mi   低 4Fa   低 5So   低 6La   低 7Si
//频率(Hz)262     294     330     349     392     440     494
//
//////////////////////////////////////////////////////////////

parameter clk_freq = 26'd50_000_000 ;

wire [10:0]voice_l[0:6] ;
wire [10:0]voice_m[0:6] ;
wire [10:0]voice_h[0:6] ;

reg [25:0]para_voice_l[0:6] ;
reg [25:0]para_voice_m[0:6] ;
reg [25:0]para_voice_h[0:6] ;

reg [25:0]cnt_voice_l[0:6] ;
reg [25:0]cnt_voice_m[0:6] ;
reg [25:0]cnt_voice_h[0:6] ;

reg freq_voice_l[0:6] ;
reg freq_voice_m[0:6] ;
reg freq_voice_h[0:6] ;

reg freq_voice[0:6] ;

assign voice_l[0] = 11'd262 ;
assign voice_l[1] = 11'd294 ;
assign voice_l[2] = 11'd330 ;
assign voice_l[3] = 11'd349 ;
assign voice_l[4] = 11'd392 ;
assign voice_l[5] = 11'd440 ;
assign voice_l[6] = 11'd494 ;
/*
assign voice_m[0] = 11'd523 ;
assign voice_m[1] = 11'd587 ;
assign voice_m[2] = 11'd659 ;
assign voice_m[3] = 11'd698 ;
assign voice_m[4] = 11'd784 ;
assign voice_m[5] = 11'd880 ;
assign voice_m[6] = 11'd988 ;

assign voice_h[0] = 11'd1047 ;
assign voice_h[1] = 11'd1175 ;
```

```
assign voice_h[2] = 11'd1319 ;
assign voice_h[3] = 11'd1397 ;
assign voice_h[4] = 11'd1568 ;
assign voice_h[5] = 11'd1760 ;
assign voice_h[6] = 11'd1967 ;
*/

reg [2:0]cnt_scan ;
reg beep_flag ;
wire key_and ;
reg lock_flag ;

wire [6:0]key_stable ;
wire [2:0]sw_stable ;

btn_deb#(
   .BTN_WIDTH (4'd7),
   .BTN_DELAY (20'hF423F)
)btn_deb1
(
   .clk              (clk)        ,
   .btn_in           (key)        ,

   .btn_deb          (key_stable)
);

btn_deb#(
   .BTN_WIDTH (4'd3),
   .BTN_DELAY (20'hF423F)
)btn_deb2
(
   .clk              (clk)     ,
   .btn_in           (sw)     ,

   .btn_deb          (sw_stable)
);

generate
   genvar i ;
      for ( i=0 ; i<=6 ; i=i+1 ) begin: voice_gen

         always @(posedge clk ) begin
               para_voice_l[i] <= (clk_freq / voice_l[i])>>1 ;
//               para_voice_m[i] <= (clk_freq / voice_m[i])>>1 ;
```

```
//              para_voice_h[i] <= (clk_freq / voice_h[i])>>1 ;
        end

        always @(posedge clk ) begin
           if (~rst_n) begin
              cnt_voice_l[i] <= 26'd0 ;
              freq_voice_l[i] <= 1'b0 ;
           end
           else if (cnt_voice_l[i] >= para_voice_l[i] - 1'b1) begin
              cnt_voice_l[i] <= 26'd0 ;
              freq_voice_l[i] <= ~freq_voice_l[i] ;
           end
           else begin
              cnt_voice_l[i] <= cnt_voice_l[i] + 1'b1 ;
              freq_voice_l[i] <= freq_voice_l[i] ;
           end
        end
/*
        always @(posedge clk ) begin
           if (~rst_n) begin
              cnt_voice_m[i] <= 26'd0 ;
              freq_voice_m[i] <= 1'b0 ;
           end
           else if (cnt_voice_m[i] >= para_voice_m[i] - 1'b1) begin
              cnt_voice_m[i] <= 26'd0 ;
              freq_voice_m[i] <= ~freq_voice_m[i] ;
           end
           else begin
              cnt_voice_m[i] <= cnt_voice_m[i] + 1'b1 ;
              freq_voice_m[i] <= freq_voice_m[i] ;
           end
        end

        always @(posedge clk ) begin
           if (~rst_n) begin
              cnt_voice_h[i] <= 26'd0 ;
              freq_voice_h[i] <= 1'b0 ;
           end
           else if (cnt_voice_h[i] >= para_voice_h[i] - 1'b1) begin
              cnt_voice_h[i] <= 26'd0 ;
              freq_voice_h[i] <= ~freq_voice_h[i] ;
           end
           else begin
              cnt_voice_h[i] <= cnt_voice_h[i] + 1'b1 ;
```

```
                freq_voice_h[i] <= ~freq_voice_h[i] ;
            end
        end
*/
        always @(posedge clk ) begin
          if (~rst_n)
              freq_voice[i] <= 1'b0 ;
           else if (sw_stable[0])
             freq_voice[i] <= freq_voice_l[i] ;
//            else if (sw_stable[1])
//              freq_voice[i] <= freq_voice_m[i] ;
//            else if (sw_stable[2])
//              freq_voice[i] <= freq_voice_h[i] ;
           else
            freq_voice[i] <= 1'b0 ;
        end

    end
endgenerate

always @(posedge clk ) begin
   if (~rst_n)
      cnt_scan <= 3'd0 ;
   else if (cnt_scan >= 3'd6)
      cnt_scan <= 3'd0 ;
   else
      cnt_scan <= cnt_scan + 1'b1 ;
end

assign key_and = &key_stable ;

always @(posedge clk ) begin
   if (~rst_n) begin
      beep_flag <= 1'b0 ;
//      lock_flag <= 1'b0 ;
    end
   else if (key_and) begin
      beep_flag <= 1'b0 ;
//      lock_flag <= 1'b0 ;&&(lock_flag == 1'b0 )
    end
   else if ((key_stable[cnt_scan] == 1'b0)) begin
      beep_flag <= freq_voice[cnt_scan] ;
//      lock_flag <= 1'b1 ;
    end
```

```
    else begin
        beep_flag <= beep_flag ;
//      lock_flag <= lock_flag ;
    end
end

assign beep = beep_flag ;

endmodule
```

26.6. 实验现象

将 SW5 拨码开关**拨上**，（此时保持 SW1、SW2、SW3 拨码开关为拨上状态）。

S1 按键**按下**发出 Do 音、S2 按键**按下**发出 Re 音、S3 按键**按下**发出 Mi 音、S4 按键**按下**发出 Fa 音、SW0 拨码开关**拨下**发出 So 音、SW1 拨码开关**拨下**发出 La 音、SW2 拨码开关**拨下**发出 Si 音。