

实验九 基于 FPGA 的乐曲硬件演奏电路设计

一、实验目的

- 巩固 QuartusII 软件的相关操作，掌握数字电路设计的基本流程。
- 巩固 QuartusII 的软件，掌握基本的设计思路，软件环境参数配置，仿真，管脚分配，下载等基本操作
- 巩固数控分频器的设计方法。

二、实验内容

利用数控分频器设计硬件乐曲演奏电路，利用蜂鸣器播放《我和我的祖国》。

我和我的祖国

1=D $\frac{6}{8}$
J=150 庄重 热情地

张 黎词
秦咏诚曲

李 谷 一 演唱

($\dot{1}$ $\dot{2}$ $\dot{3}$ $\dot{2}$ $\dot{1}$ $\dot{6}$ | $\dot{7}$ $\dot{6}$ $\dot{3}$ $\dot{5}$ | $\dot{1}$ $\dot{2}$ $\dot{3}$ $\dot{2}$ $\dot{1}$ $\dot{6}$ | $\dot{7}$ $\dot{5}$ $\dot{3}$ $\dot{6}$ | $\dot{6}$ |

$\dot{5}$ $\dot{4}$ $\dot{3}$ $\dot{2}$ | $\dot{7}$ $\dot{6}$ $\dot{5}$ $\dot{3}$ | $\dot{4}$ $\dot{3}$ $\dot{2}$ $\dot{1}$ | $\dot{1}$ $\dot{3}$ $\dot{3}$ $\dot{5}$ $\dot{3}$ $\dot{3}$ | $\dot{1}$ $\dot{3}$ $\dot{3}$ $\dot{5}$ $\dot{3}$ $\dot{3}$ |

$\dot{5}$ $\dot{6}$ $\dot{5}$ $\dot{4}$ $\dot{3}$ $\dot{2}$ | $\dot{1}$ $\dot{5}$ | $\dot{1}$ $\dot{3}$ $\dot{1}$ $\dot{7}$ $\dot{6}$ $\dot{3}$ | $\dot{5}$ $\dot{5}$ | $\dot{6}$ $\dot{7}$ $\dot{6}$ $\dot{5}$ $\dot{4}$ $\dot{3}$ | $\dot{2}$ $\dot{6}$ |

1. 我和我的祖国，一刻也不能分割。无论我走到哪里，
2. 我的祖国和我，像海和浪花一朵。浪是海的赤子，

$\dot{7}$ $\dot{6}$ $\dot{5}$ $\dot{5}$ $\dot{1}$ $\dot{2}$ | $\dot{3}$ $\dot{3}$ | $\dot{5}$ $\dot{6}$ $\dot{5}$ $\dot{4}$ $\dot{3}$ $\dot{2}$ | $\dot{1}$ $\dot{5}$ | $\dot{1}$ $\dot{3}$ $\dot{1}$ $\dot{7}$ $\dot{2}$ $\dot{1}$ | $\dot{6}$ $\dot{6}$ |

都流出一首赞歌。我歌唱每一座高山，我歌唱每一条河。
海是那浪的依托。每当大海在微笑，我就是笑的漩涡。

$\dot{1}$ $\dot{7}$ $\dot{6}$ $\dot{5}$ | $\dot{6}$ $\dot{5}$ $\dot{4}$ $\dot{3}$ | $\dot{7}$ $\dot{6}$ $\dot{5}$ $\dot{2}$ | $\dot{1}$ $\dot{1}$ | $\dot{1}$ $\dot{2}$ $\dot{3}$ $\dot{2}$ $\dot{1}$ $\dot{6}$ | $\dot{7}$ $\dot{6}$ $\dot{3}$ $\dot{5}$ $\dot{5}$ |

袅袅炊烟，小小村落，路上一道辙。啦啦啦 啦啦啦 啦啦啦啦
我分担着 海的忧愁，分享海的欢乐。啦啦啦 啦啦啦 啦啦啦啦

$\dot{1}$ $\dot{2}$ $\dot{3}$ $\dot{2}$ $\dot{1}$ $\dot{6}$ | $\dot{7}$ $\dot{5}$ $\dot{3}$ $\dot{6}$ $\dot{6}$ | $\dot{5}$ $\dot{4}$ $\dot{3}$ $\dot{2}$ | $\dot{7}$ $\dot{6}$ $\dot{5}$ $\dot{3}$ | $\dot{4}$ $\dot{2}$ $\dot{1}$ | $\dot{1}$ $\dot{10}$ |

啦啦啦 啦啦啦 啦啦啦啦 你用你那 母亲的温情 和我诉说。
啦啦啦 啦啦啦 啦啦啦啦 永远给我 碧浪清波，心中的

$\dot{1}$ $\dot{10}$ | $\dot{1}$ $\dot{2}$ $\dot{3}$ $\dot{2}$ $\dot{1}$ $\dot{6}$ | $\dot{7}$ $\dot{6}$ $\dot{3}$ $\dot{5}$ | $\dot{1}$ $\dot{2}$ $\dot{3}$ $\dot{2}$ $\dot{1}$ $\dot{6}$ | $\dot{7}$ $\dot{5}$ $\dot{3}$ $\dot{6}$ |

歌。啦啦啦 啦啦啦 啦啦啦啦 啦啦啦啦 啦啦啦啦 啦啦啦啦

$\dot{5}$ $\dot{4}$ $\dot{3}$ $\dot{2}$ | $\dot{7}$ $\dot{6}$ $\dot{5}$ $\dot{3}$ | $\dot{4}$ $\dot{2}$ $\dot{1}$ | $\dot{1}$ $\dot{10}$ | $\dot{5}$ $\dot{2}$ $\dot{1}$ | $\dot{1}$ $\dot{1}$ |

永远给我 碧浪清波，心中的歌。心中的歌。

三、实验准备

该实验需要使用 Audio 功放模块、SW1-SW8 模块、F1-F10 模块。将控制拨码开关 CTRL_SW 中 SEL1 拨置于上,SEL2 拨置于下,逻辑电平为 10 使 DP9 数码管显示 2, 可以使用 Audio 音频模块, 如果把 BZ1 中的跳线器连接在 1-2 的方式, 那么可以使用 LS_BUZ1 有源蜂鸣器, 如果把 BZ1 中的跳线器连接在 2-3 的方式, 那么可以使用 2W 的喇叭; F1-F6 已经固定连接到实验平台中的 FPGA_CON1 处, F7-F10 是复用的 I/O, 当实验平台上控制拨码开关模块 LCD_ALONE_CTRL_SW 中 KSI 拨置于下, 可以使用按键模块中的 F7,F8, F9, F10; SW1-SW8 已经固定连接到实验平台中的 FPGA_CON1 和 FPGA_CON2 处。

四、实验原理

问题 1: 如何发出不同音调的声音?

问题 2: 如何控制音符的节拍?

在此, 首先介绍一下硬件电路的发声原理。我们知道, 声音的频谱范围约在几十到几千赫兹, 如果将一定频率的方波接上扬声器就能发出相应频率的声音。而乐曲中的每一音符对应着一个确定的频率, 因此, 要想 FPGA 发出不同音符的音调, 实际上只要控制它输出相应音符的频率即可(音符和频率的关系见下表 1)。

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	261.6	中音 1	523.3	高音 1	1045.5
低音 2	293.7	中音 2	587.3	高音 2	1174.7
低音 3	329.6	中音 3	659.3	高音 3	1318.5
低音 4	349.2	中音 4	698.5	高音 4	1396.9
低音 5	392	中音 5	784	高音 5	1568
低音 6	440	中音 6	880	高音 6	1760
低音 7	493.9	中音 7	987.8	高音 7	1975.5

乐曲都是由一连串的音符组成, 因此按照乐曲的乐谱依次输出这些音符所对应的频率, 就可以在扬声器上连续地发出各个音符的音调。而要准确地演奏出一首乐曲, 仅仅让扬声器能够发声是不够的, 还必须准确地控制乐曲的节奏, 即每个音符的持续时间。由此可见, 乐曲中每个音符的发音频率及其持续的时间是乐曲能够连续演奏的两个关键因素。

而简易电子琴, 工作原理与乐曲演奏一样, 只是将固定预置乐曲变成了手动按键输入, 节拍时间取决于按键的停留时间, 如果合适, 同样能播放出完整的歌曲来。

频率的高低决定了音调的高低。音乐的十二平均率规定: 每两个八度音(如简谱中的中音 1 和高音 1)之间的频率相差一倍。在两个八度音之间又分为十二个半音。另外, 音名 A(简谱中的低音 6)的频率为 440Hz, 音名 B 到 C 之间、E 到 F 之间为半音, 其余为全音。由此可以计算出简谱中从低音 1 到高音 1 之间每个音名对应的频率, 所有不同频率的信号都是从同一个基准频率分频得到的。由于音阶频率多为非整数, 而分频系数又不能为小数, 因此必须将计算得到的分频数四舍五入取整。因此, 要想 FPGA 发出不同音符的音调, 实际上只要控制它输出相应音符的频率即可。本文中选取 50MHZ 作为 CLK 的分频计数器的输入分频信号。乐曲都是由一连串的音符组成, 因此按照乐曲的乐谱依次输出这些音符所对应的频率, 就可以在扬声器上连续地发出各个音符的音调。采用 50M 时钟, 分频表 2 如下所示:

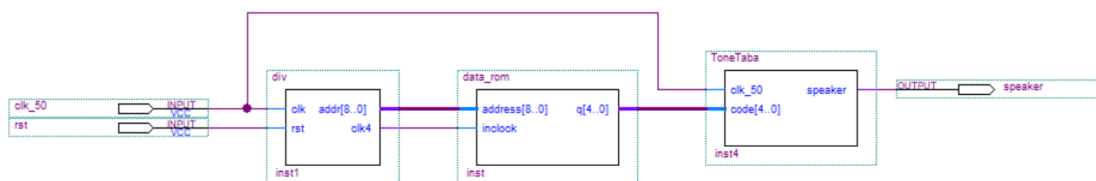
表 2 频率与分频系数

音名	频率/Hz	分频系数	音名	频率/Hz	分频系数	音名	频率/Hz	分频系数
低音 1	261.6	191132	中音 1	523.3		高音 1	1045.5	
低音 2	293.7		中音 2	587.3		高音 2	1174.7	
低音 3	329.6		中音 3	659.3		高音 3	1318.5	
低音 4	349.2		中音 4	698.5		高音 4	1396.9	
低音 5	392		中音 5	784		高音 5	1568	
低音 6	440		中音 6	880		高音 6	1760	
低音 7	493.9		中音 7	987.8		高音 7	1975.5	

音符的持续时间须根据乐曲的速度及每个音符的节拍数来确定。因此，想控制音符的音长，就必须知道乐曲的速度和每个音符所对应的节拍数，在这个设计中所演奏的乐曲的最短的音符为四分音符，如果将全音符的持续时间设为 1s 的话，那么一拍所应该持续的时间为 0.25 秒，则只需要再提供一个 4HZ 的时钟频率即可产生四分音符的时长。要想让系统知道现在应该演奏哪个音符，而这个音符持续的时间应该是多少，就必须编写乐曲文件，在乐曲文件中音符是按地址存放的，当系统工作时就按 4Hz 的频率依次读取简谱，当系统读到某个音符的简谱时就对应发这个音符的音调，持续时间为 0.25 秒，而如果在曲谱文件中这个音符为三拍音长，那又该如何控制呢？其实只要将该音符连续书写三遍，这时系统读乐曲文件的时候就会连续读到三次，也就会发三个 0.25 秒的音长，这时我们听上去就会持续了三拍的时间，通过这样一个简单的操作就可以控制音乐的音长了。

五、设计原理框图

基于 FPGA 的乐曲硬件演奏电路顶层设计如下：



1) div 模块：

用于产生 4Hz 分频和 ROM 地址

2) data_rom 模块：

《我和我的祖国》音乐的存放 ROM 中的 mif 文件中，部分歌谱如下：

zuguo.mif								
Addr	+0	+1	+2	+3	+4	+5	+6	+7
0	12	12	13	13	12	12	11	11
8	10	10	9	9	8	8	8	8
16	8	8	5	5	5	5	5	5
24	8	8	10	10	15	15	14	14
32	13	13	13	10	12	12	12	12
40	12	12	12	12	12	12	12	12
48	13	13	14	14	13	13	12	12

注意：简谱中 0 表示停顿，其频率值设定为 3750000Hz，需要 134 分频，此时不发声。

3) ToneTaba 模块：

查表输出所有音符频率

六、引脚分配

注意：实验平台采用模式2。

设计端口	芯片引脚	开发板模块	备注
clk_50	T1	sys_clk	系统时钟 50MHz
rst	N18	SW1	拨码开关： 上：“1” 下：“0”
speaker	B6	蜂鸣器/喇叭	DP9 显示“2” SEL1.SEL2=10 1、BZ1 跳线 1-2 连接，蜂鸣器 2、BZ1 跳线 2-3 连接，喇叭