2020-11-23

秦嘉余

191220088

1348288404@qq.com

数字电路实验十 音频输出实验

目录

[1.实验目的 1](#_Toc57567333)

[2.实验原理 1](#_Toc57567334)

[1.生成正弦波 1](#_Toc57567335)

[2. 音频接口 1](#_Toc57567336)

[3.实验环境/器材 2](#_Toc57567337)

[4. 键盘电子琴功能 2](#_Toc57567338)

[1. 设计思路 2](#_Toc57567339)

[2. 实验代码 2](#_Toc57567340)

[3.RTL视图 4](#_Toc57567341)

[4.引脚分配 4](#_Toc57567342)

[5.实验结果 4](#_Toc57567343)

[5.拓展功能 4](#_Toc57567344)

[6.实验实验中遇到的问题及解决办法 5](#_Toc57567345)

1.实验目的

本实验的主要目的是学习音频信号的输出方式以及如何将数字信号转换为 模拟信号的基本原理。

2.实验原理

## 1.生成正弦波

人耳可以听到的声音的频率范围是 20-20kHz。音频设备如扬声器或耳机等 所接收的音频信号一般是模拟信号，即时间上连续的信号。但是，由于数字器 件只能以固定的时间间隔产生数字输出，我们需要通过数字/模拟转换将数字信号转换成模拟信号输出。

由以下四步生成频率为f的正弦波:

1. 根据频率 f 计算递增值 d = f\*65536/48000

2. 在系统中维持一个 16bit 无符号整数计数器，每个样本点递增d

3. 根据 16 位无符号整数计数器的高 10 位来获取查表地址 k，并查找 1024 点 的正弦函数表

4. 使用查表结果作为当前的数字输出

## 2. 音频接口

DE10-Standard 开发板上集成了一块WM8731 音频编解码芯片，其参考手册可以在课程网站上找到。该音频编解码芯片提供 24bit 的音频接口，支持8kHz 到 96kHz 的采样率。

在音频发送的实现上分为两个步骤。

第一步，是配置音频编解码芯片。在数字系统中，大量的设置是通过读写设备上或芯片上的寄存器来 实现的。每个寄存器会有一个独立的地址，并会有自己特定的功能。CPU 或主 控单元通过在寄存器上写入特定的值来将设备设置到特定状态，或者让设备执 行特定的动作。例如，要对音频芯片进行 reset，只需要在地址0F的寄存器中写入00 即可。

第二步是音频流发送。如10 . 1所述，我们的音频流是一串音频的电压数字。我们需要通过 I2S 接口将这串数字传输给音频芯片。传输的具体格式是在 第一步中通过寄存器设置的。一般音频是包括左右两个声道的，我们需要以 48kHz 的速率发送两个声道的数据给音频芯片。在设置正确时，音频芯片就会将对应的数字值转换为耳机的输出，放出声音来。

3.实验环境/器材

本次实验的环境为Quartus17.1版本

本次实验的器材为DE10 Standard开发板

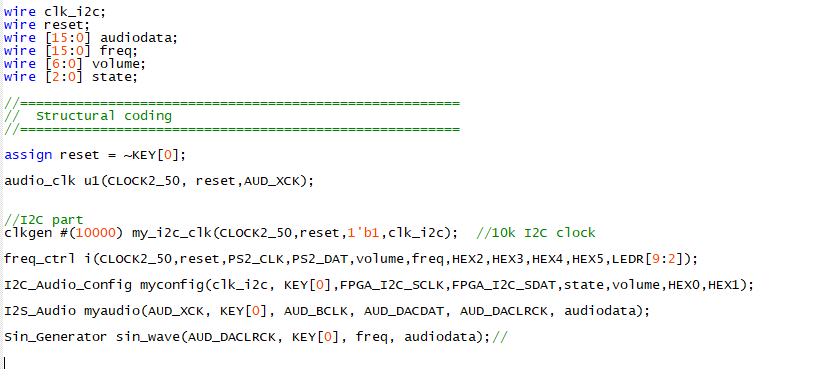
4. 键盘电子琴功能

## 1. 设计思路

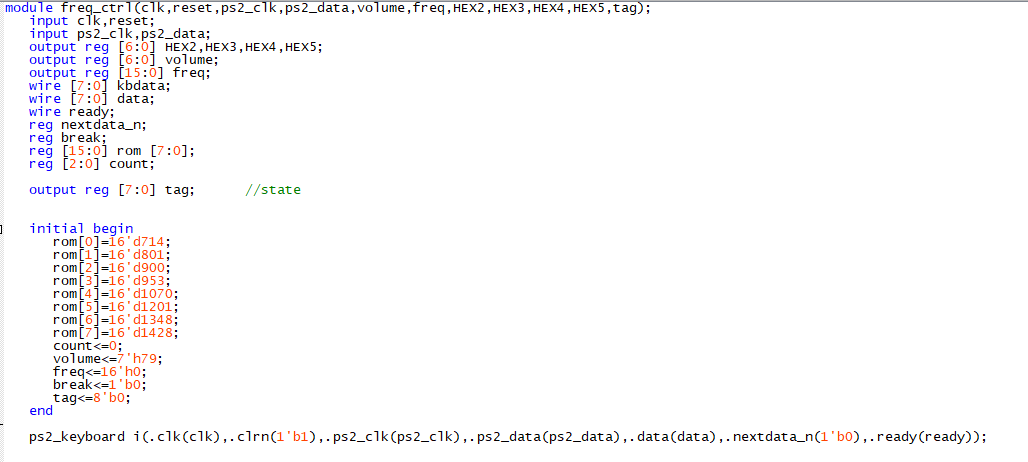
本次实验中，主要与音频有关的配置输入输出文件都已给出，只需要根据键盘的键码输入设置不同的freq参数传入sin\_wave模块即可。对于键盘输入部分，在实验8的基础上重写部分代码即可。拓展部分中，设置音量需要对配置寄存器修改，并重置音频芯片，此时音量参数volume也同时由键盘控制，和音部分需要记录当前按下的键，对按下的每个键对应的freq取平均值即可。

## 2. 实验代码

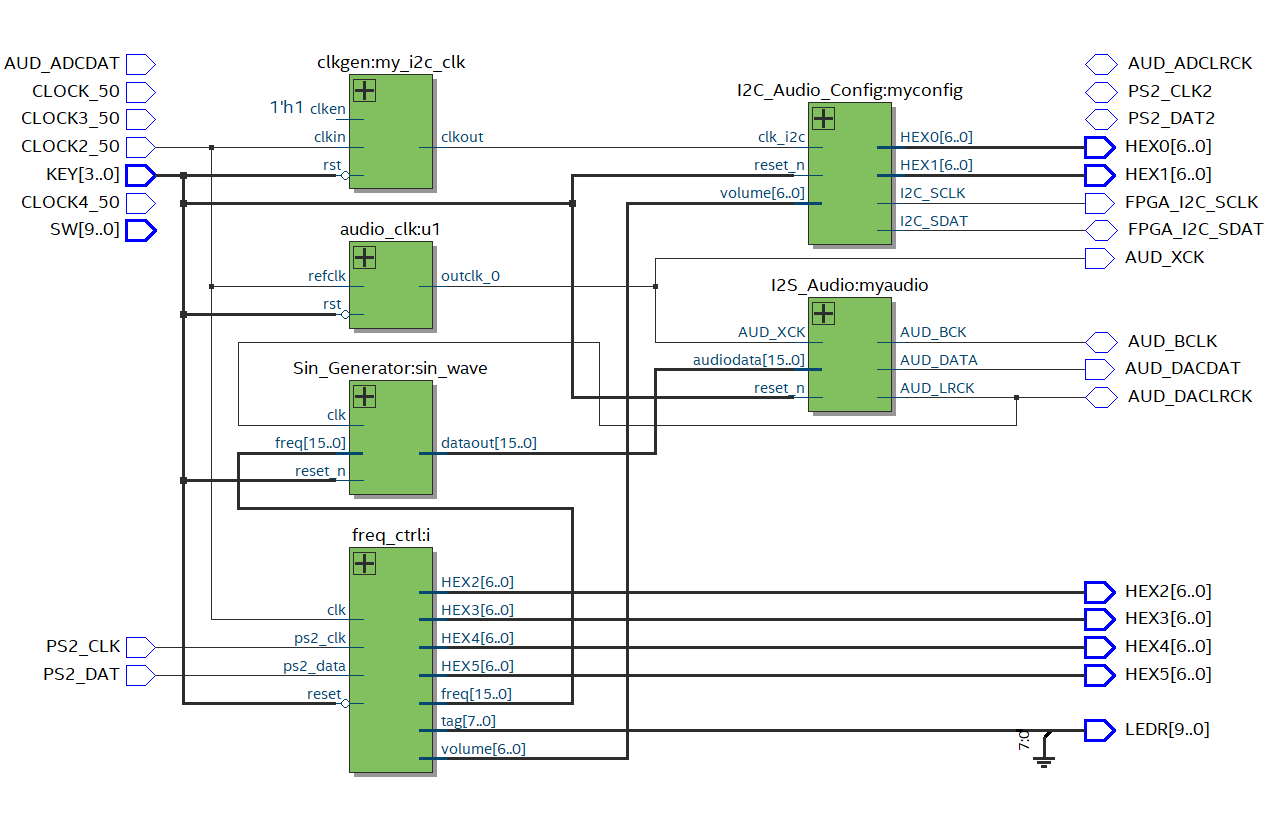
顶层模块如下，增加了freq\_ctrl模块来控制freq参数与volume参数，除此之外，也修改了一些参数。例如将按下哪些键码用LED灯显示，利用七段显示管显示freq与volume便于观察。



以下为控制模块的基本初始化设置与参数设置，其中rom存储8种音对于的freq值，count记录当前按下的按键数量，tag为8位标记位，tag[i]为1代表i号音对应的按键按下，否则未按下，具体逻辑代码见压缩包对应文件



## 3.RTL视图



## 4.引脚分配

该实验为system软件自动分配的引脚，所以不再展示、

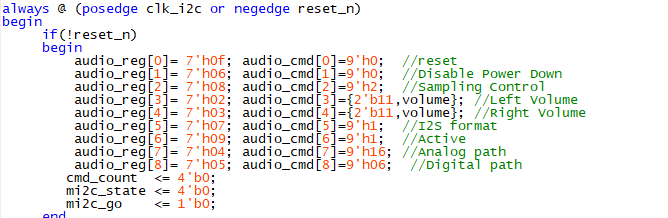
## 5.实验结果

本实验已在开发板上验收

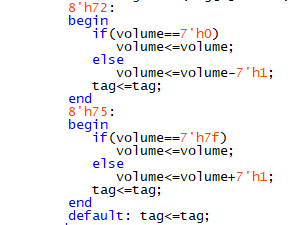
5.拓展功能

**1.修改音量**

在Audio\_Config模块中，修改如下：

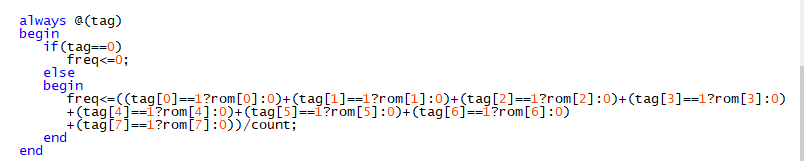


查阅手册得知，audio\_cmd中低7位为音量，最高位需要设置为1才可以修改音量，所以在命令执行的时序逻辑中修改寄存器，按下reser后完成音量修改，volume 的修改同样在控制模块中，如下：



**2.和音**

和音部分的实现首先利用一个标识变量tag，记录8个音的按下与否，tag[i]为1代表i号音按下，否则未按下。count记录当前按下的按键数。这两个变量在键盘按键按下或者松开时修改，在上述图片中已经展示，下面为根据count和tag计算freq的部分



对应按键按下取当前按键对于的freq，否则取0，最后取平均数即可

6.实验实验中遇到的问题及解决办法

本次实验的难点在于对音频芯片的理解以及对于键盘代码的复用与重写。在实验中遇到了以下一些问题：

**1.**

最开始，将freq修改为固定之后，耳机连接音频始终没有声音，后来发现是当前开发板的音频芯片损坏，更换开发板测试后正常。由此见得，实验前的检测准备工作尤为重要，否则在功能复杂后再在开发板验证将难以排查出问题

**2.**

修改音量部分始终无法生效，在查阅手册后，发现，对于9位的audio\_cmd， 其中低7位为数据位，最好位为0时，使用默认设置，为1时，才会采用后面7位作为音量来设置

**3.**

计算和音的freq时，始终无法正确的设置，freq更改过快或累计过大，在7段显示管上无法正常显示，由此判断是时序逻辑发生了问题，所以将count，tag的改变放在键码的判断的always语句块中，使得键码按下与count，tag 的改变同步，此时counr与tag都改变正确。freq的计算也需要时钟同步，直接循环计算由于硬件电路的限制，难以办到先对freq清零再累加取平均这样的操作次序，导致freq异常。后来使用一个较长的表达式直接计算freq即可使得freq正确，这在硬件电路上就不会因为时序的关系导致逻辑不同步，更加符合硬件语言的设计。