实验名称：实验十 音频输出实验

姓名：张涵之

学号：191220154

班级：周一5-6

邮箱：[191220154@smail.nju.edu.cn](mailto:191220154@smail.nju.edu.cn)

实验时间：2020/11/22

8.4 实验内容

请将之前实验实现的键盘与本实验的音频输出结合，实现一个简单的键盘电子琴功能。可以根据按下的键的键值，决定播放的正弦的频率，从而实现电子琴的功能。基本要求：实现至少8个音符。支持每次按下单个按键。按键按下后开始发音，按键期间持续发音，松开后停止发音。按无关键不发音。无杂音和爆破音等干扰。

扩展要求：可调节音量。

实验目的：将键盘与音频输出结合，实现一个简单的键盘电子琴。

实验原理：

1）音频输出原理：音频设备如扬声器或耳机等所接收的音频信号是模拟信号，即时间上连续的信号。由于数字器件只能以固定的时间间隔产生输出，需要通过数字/模拟转换将数字信号转换成模拟信号输出。根据采样定律，数字信号的采样率应不低于信号频率的两倍。

在实际信号输出时，我们一般不采用浮点数而选用整数值来表示每个样本点的大小。

生成频率为f的正弦波的过程如下：

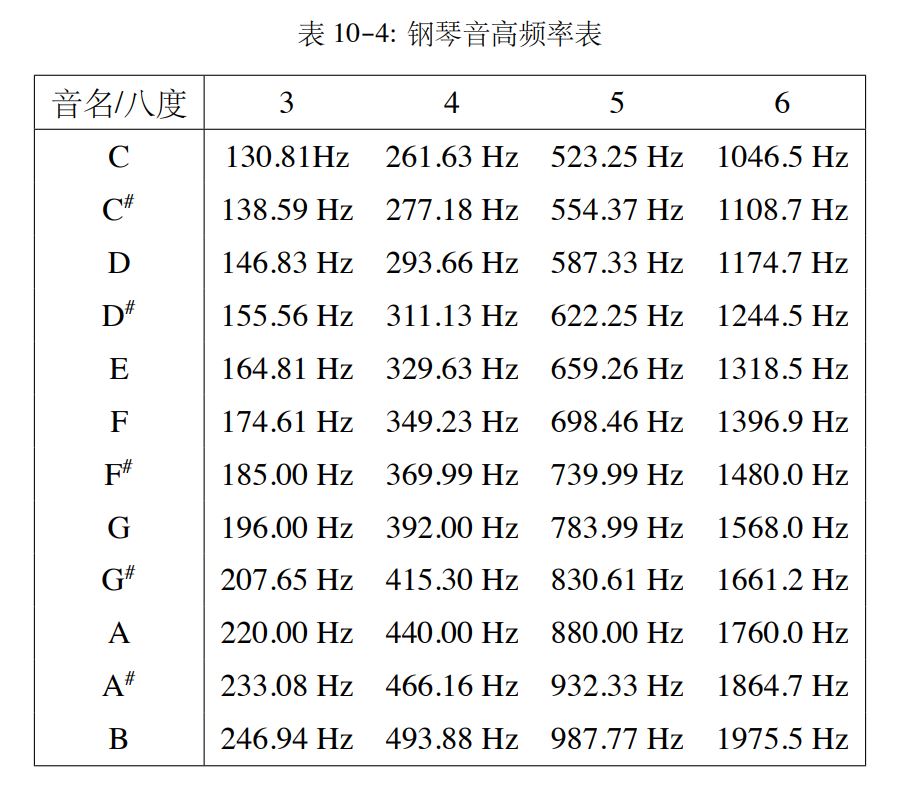
1. 根据频率f计算递增值d = f×65536/48000；

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。

实验提供了钢琴音高频率表如下：



则可列表出C5，D5，… ，B5，C6等音符对应的递增值、字母和键码：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字符 | Q | W | E | R | T | Y | U | I |
| 键码 | 15 | 1d | 24 | 2d | 2c | 35 | 3c | 43 |
| 音名 | C5 | D5 | E5 | F5 | G5 | A5 | B5 | C6 |
| 频率 | 523.25 | 587.33 | 659.26 | 698.46 | 783.99 | 880.00 | 987.77 | 1046.5 |
| 递增 | 714 | 802 | 900 | 954 | 1070 | 1201 | 1349 | 1429 |

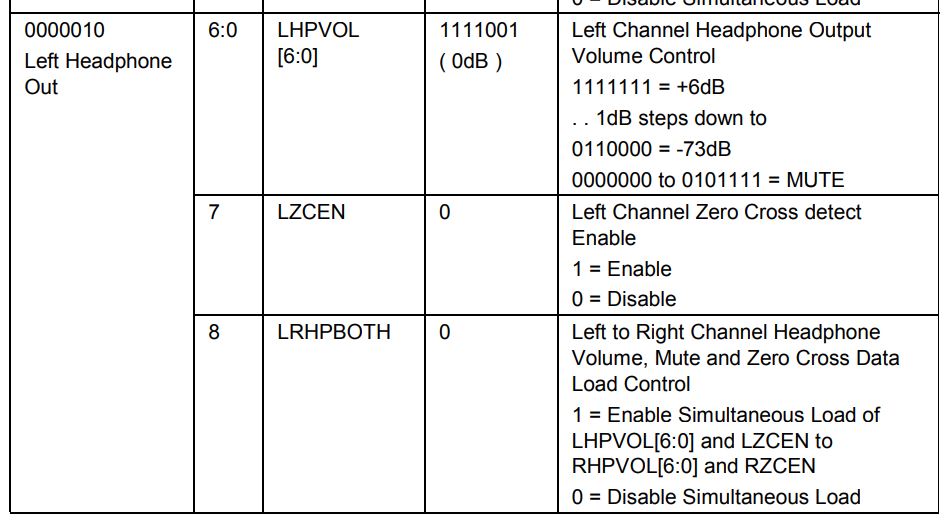
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字符 | A | S | D | F | G | H | J | K |
| 键码 | 1c | 1b | 23 | 2b | 34 | 33 | 3b | 42 |
| 音名 | C#5 | D#5 |  | F#5 | G#5 | A#5 |  | C#6 |
| 频率 | 554.37 | 622.25 |  | 739.99 | 830.61 | 932.33 |  | 1108.7 |
| 递增 | 757 | 850 |  | 1010 | 1134 | 1273 |  | 1514 |

2）音频接口：生成每个时间点上的音频波形后，信号通过音频接口送给耳机或者扬声器。

3）I2S接口：通过I2S接口来发送和接收数字音频信号。音频信号的基准时钟AUD\_XCK设为采样频率的256倍或384倍。本实验中采样频率为48kHz，AUD\_XCK设为其384倍，即48000 × 384=18.432MHz。调用Quartus提供的标准IP库来产生这类特殊的时钟。

在生成完时钟信号后，只需要按要求将每个样本点生成的16bit有符号整数数据按高位前发送即可。可以将左右声道设置为一样的数据，以实现单声道播放。

4）音量调整：查表可得寄存器低七位为音量，调节范围为0110000~1111111。



可以在keyboard上层模块key2note中通过键码的识别控制和修改volume，将volume作为参数传入I2C\_Audio\_Config模块，在always语句中修改audio\_cmd的值。

程序代码或流程图：

ps2\_keyboard获取键码 —> \

key2note根据键码/音名/频率关系得出递增值 —> \

Sin\_Generator根据递增值得到audiodata —> \

I2C\_Audio\_Config与I2S\_Audio控制音频元件 —> \

audio\_clk和clkgen生成时钟 —> 顶层模块综合

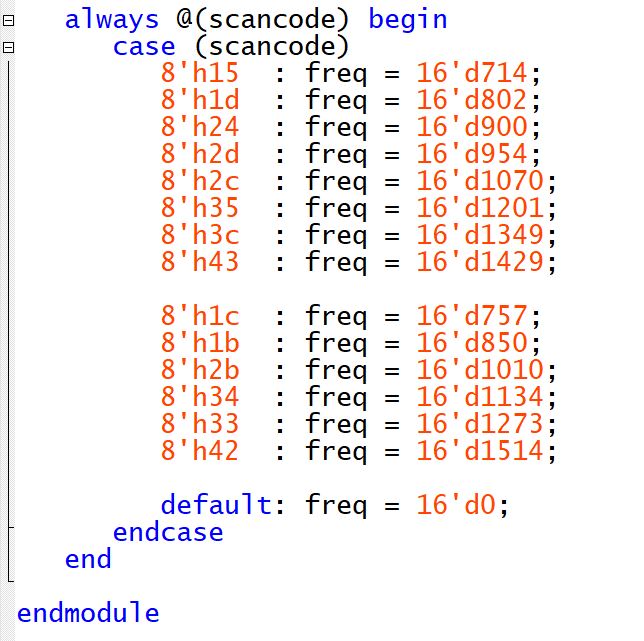
audio\_clk //调用Quartus标准IP库产生的18.432MHz的时钟

clk\_gen //用开发板上的CLOCK\_50生成10k的I2C时钟

ps2\_keyboard //键盘控制器模块，用于从键盘获取输入（键码）

key2note //键盘上层控制模块，通过键码判断音调和音量





//clk为时钟信号，clrn为清零信号

//nextdata，ready和overflow即键盘控制器中的nextdata\_n，ready和overflow

//scancode用于暂存从键盘获得的键码，volumn用于音量控制

//en为是否发出声音的使能控制端，在顶层模块中被调用

//flag为调节en的控制信号，在按键松开（接受0xf0后一个键值）时en置零

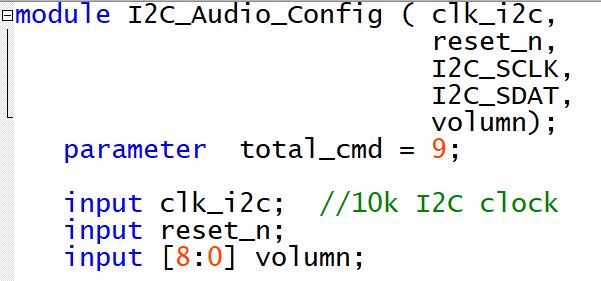
其中QWERTYUI为C5，D5，…，C6，下面为对应的升调（如果有的话）

音量控制为按下松开按键，<(,) 表示音量减小，>(.) 表示音量增大。

Sin\_Generator //根据递增值生成相应的正弦波信号

I2C\_Controller //I2C接口单个命令发送

I2C\_Audio\_Config //音频芯片配置示例

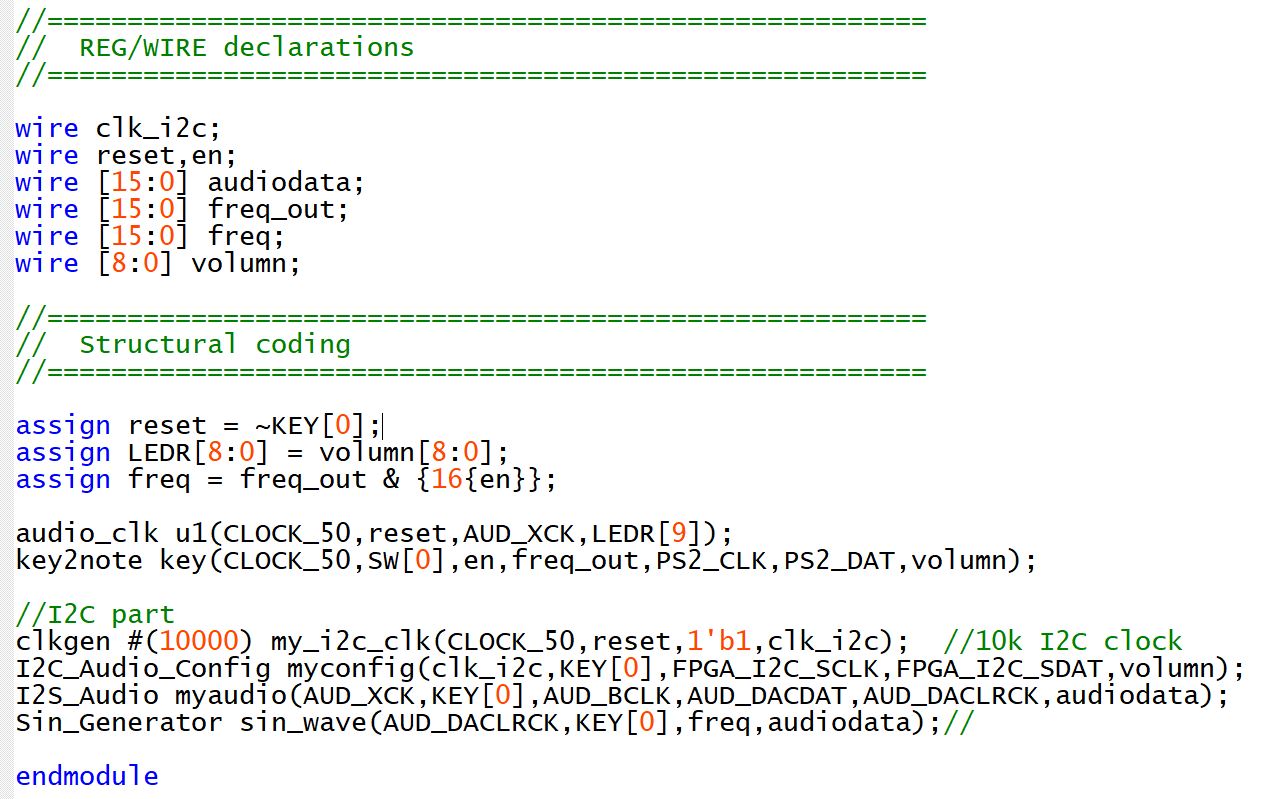


//加入了输入volumn，用于音量大小的控制

//接下来试了很多种方法，都没有成功

I2S\_Audio //与开发板上的音频模块交互

exp10 //顶层综合模块，读取按键，选择音调并调节音量播放



//用按钮KEY[0]控制重置，SW[0]控制键盘使能，LEDR[8:0]显示音量

//当en为1时说明按键按下了，根据声音频率得到相应递增量，否则置零

实验环境/器材：实验箱一个，笔记本电脑一台，键盘一个。

实验步骤/过程：

设计键盘按键和音名的对应关系，查表得到频率，计算得到对应的递增量；

根据实验八的键盘模块进行适当修改，实现键码到递增量的转换和音量选择；

阅读和理解示例工程中的代码，将其添加到自己的工程中；

对示例代码进行适当修改，增加参数和调整语句以实现音量控制；

在顶层模块中调用上述几个模块，实现键盘输入、获取音调、调节音量等功能。

测试方法：按下键盘，从耳机中听到声音。

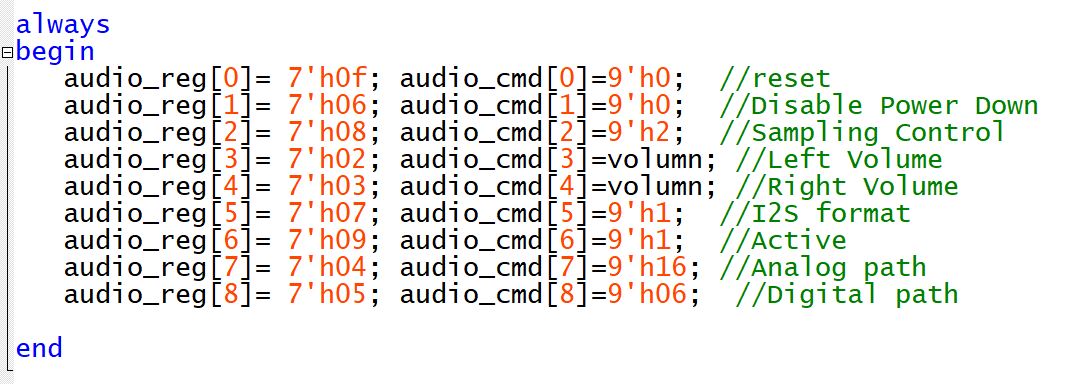
实验结果：耳机中音调的输出符合预期，音量始终无法更改。

实验中遇到的问题及解决办法：感到提供的音频实现代码十分难懂。

解决办法：多读几遍，思考每个变量的含义和作用，在顶层模块中该如何调取。

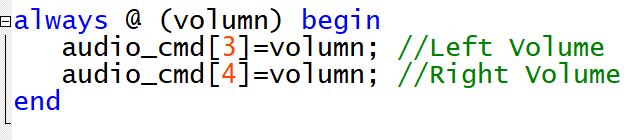
换了好几种方法都没能实现音量控制。

解决方法：首先将initial模块改成always，试图在其中直接赋值：



效果：似乎仍然只有初始化的时候赋了一次值，后续没有任何更改；

增加了一个always模块，当volumn发生变化时进行赋值

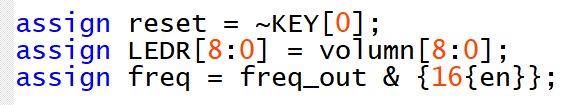


效果：根本就没有声音了！这就是课件里说的“如果试图在同一周期内读取或

写入这个RAM中的两个地址的数据，系统将无法综合这块RAM的硬件，

并且不会报错，直接结果就是编译通过但没有声音”吗！改回去了。

其次怀疑生成的volumn参数有问题，将其接到二极管上显示：



效果：按下音量键时二极管显示发生了变化，看来volumn参数有修改；

接下来怀疑对参数的理解有问题，查手册看寄存器含义：

效果：只有后七位是表示音量的，前面两位也不知道在干啥。

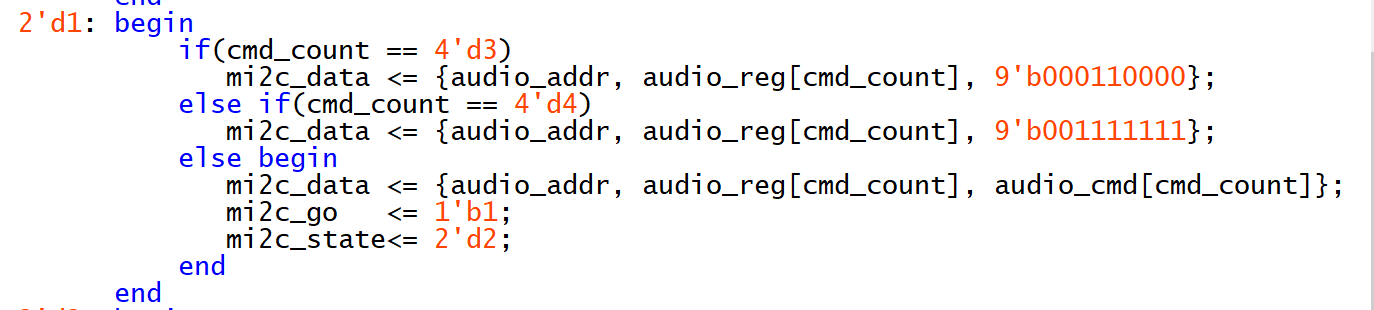
总之音量的调节范围是0110000~1111111，看起来没什么问题；

怀疑是每次增量过小听不出来，在initial模块分别赋差别很大的值，听声音效果：

效果：至少赋值0110000和1111111区别很大，一个听不见，一个快聋了

二极管跳来跳去声音不变，说明还是我的代码有问题

在I2C\_Audio\_Config下面的case语句里面给mi2c\_data直接赋值，如：



效果：仍然没有任何变化！左右声道都还是initial里面的初始值！

它到底有没有运行到这里过！还是我对mi2c\_data的写入有什么误解！

综上所述，因为我的朋友全都没做出来，也没人可以问，所以我也不做了。

实验得到的启示：学会放弃。

意见和建议：尽管手册里有“请在设计时注意读取和写入需要按RAM的操作规范进行”，遗憾的是，我至今仍然不能十分理解RAM的操作规范，也不知道问题出在哪里。

参考代码已经实现了大部分的功能，但对于学生来说阅读和理解这些代码都在干啥是很费劲的，修改和扩展这些代码的时候往往也很忐忑，明明需要写的代码量很少，却让人感觉很累而且学不到东西。也许可以出一些简单的、参考代码较少、学生自主性更强的实验？