实验十 音频输出实验

实验报告

181860085 汤昊

数字电路与数字系统二班

邮箱：1174639585@qq.com

2019.11.22

一．实验目的

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。

二．实验原理

(1) 数字信号转化为模拟信号

人耳可听声音的频率范围是 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。数字器件只能以固定的时间间隔产生数字输出，故需要通过数字/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（每秒钟产生的 数字样本数量）应不低于信号频率的两倍。所以，数字音频一般采用44.1kHz (CD音频)或 48kHz 的采样率。

(2) 采样

在 48kHz 的采样率下，即每间隔 1/48000 秒（1/48 毫秒）的时间产生一个数字输出样本点。

这样正弦波的信号就可以表示为：

$$s(n) = \sin(2\pi fn/48000)$$

其中 f 代表频率，n 是数字样本的序号。

(3) 量化

在实际信号输出时，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，。假设用带符号的 16bit 整数 （补码）来表示单个样本点，此时 32767 即对应输出的最大值（例如 +1V 电 压），-32768 即对应输出的最小值（例如-1V 电压）。这时，就可以通过循环输出 50 个点的整数值 s(n)'=(round(s(n)*32767 来产生一个 sin 波形。

(4) 不同频率的处理

存储一张 1024 点的 sin 函数表。即存储器中以地址 k =0...1023 存储了 1024 个三角函数值（以 16bit 补码整数表示），地址为 k 的 数值设置为

$$round(\sin(2\pi k/1024)\times 32767) \quad (1)$$

第 n 个样本点需要输出的值为：

$$round(\sin(2\pi nf/48000)\times 32767 \quad (2)$$

两式联立得到

$$k/1024=nf/48000$$

故每次查表对应的位置为：

$$round(nf*1024/48000)mod1024$$

(5) I2C 接口

通过设置音频芯片的寄存器值对各种参数如音频通道、改变音量、调整采样率和音频数据格式等设置。两根数据线 SDIN（或 SDAT）和时钟线 SCLK，每次发送的 16 位数据前七个位寄存器地址，后九个为寄存器值

(6) I2S 接口

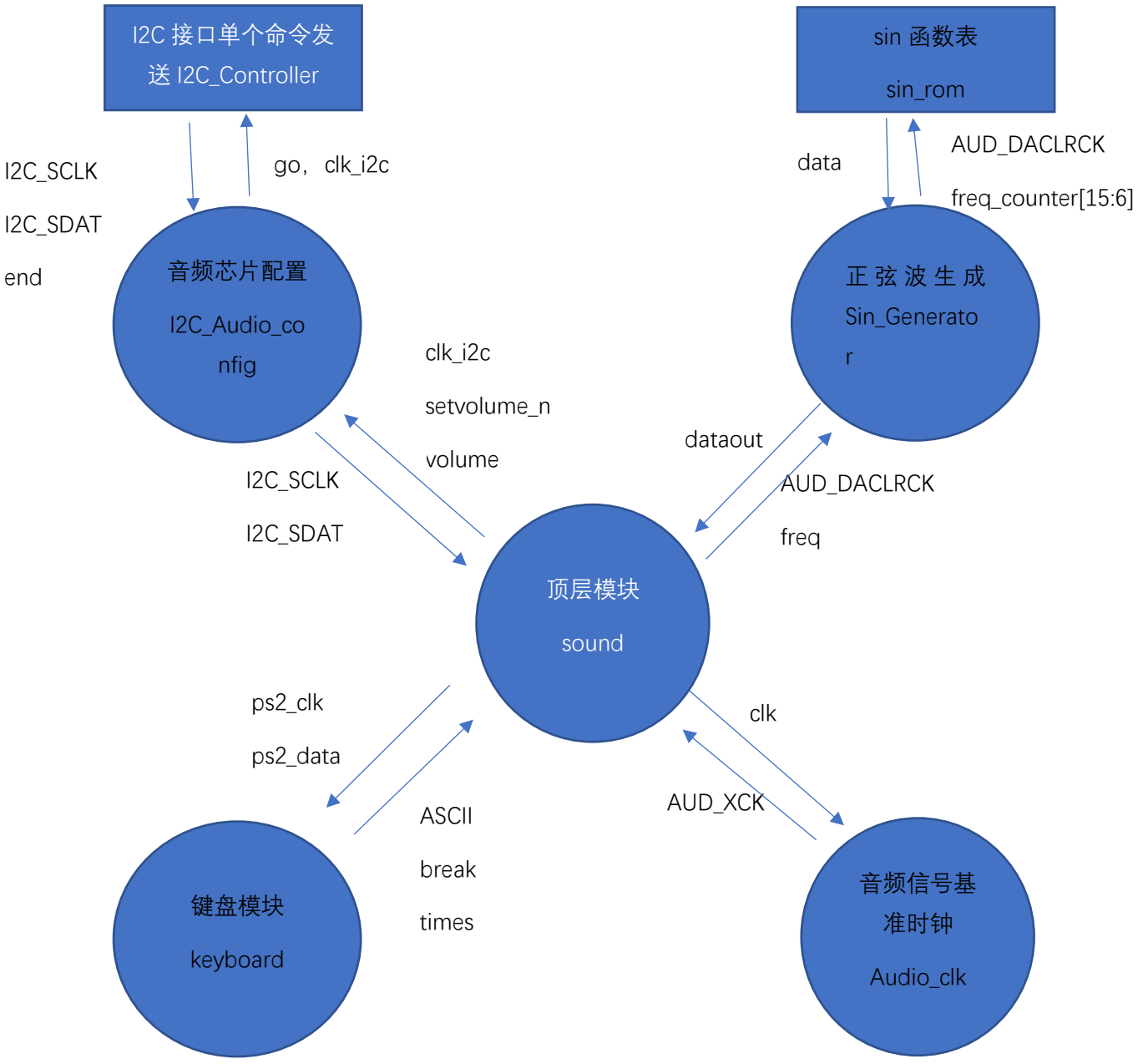
音频流是一串电频的电压数字，通过 I2S 接口传输，以 48khz 的速率发送两个声道的数据给音频芯片。

AUD_XCK：音频信号的基准时钟 AUD_BCLK 为音频数据每个比特同步时钟
AUD_DACDAT 为输出数字信号数据 AUD_DACLRCK用于输出的左右声道同步
AUD_ADCDAT, AUD_ADCLRCK 用于输入音频信号的

三．实验环境及器材

开发软件：quartus prime 17.1
开发器材：DE-standard 开发板
ps2 键盘

四．模块划分



(1) I2C_Audio_Config

用 SD_COUNTER 每次 clk_i2c 上升计数，从 0 到 32 使用 case 语句分别对应不同的操作。

流程为:

- 发送起始信号：I2C_SDAT 低电平，I2C_SCLK 高电平
- 寻找从节点：发送 7 个 bit
- 读/写信号：一个 bit，实验中总为 0，即写入
- I2C_SDAT 高阻，接受 ACK 信号
- 发送两次 8bit 数据，每次发送完 I2C_SDAT 高阻，接受 ACK 信号

(2) I2C_Audio_Config

mi2c_state 每次 clk 上升沿计数，从 0 到 3，cmd_counter 指示当前发送的命令是第几个，共 9 个命令

- 0：停止状态，准备进入下一个阶段
- 1：读入数据，准备进入下一个阶段
- 2：如果命令发送结束，进入下一个阶段
- 3：如果是最后一个命令则结束，否则进入 0 状态，准备发送下一个

(3) sin_rom

以 sintable.mif 为初始化文件的 rom

(4) Sin_Generator

维护一个 16 位计数器，以 AUD_DACLRCK 为时钟信号，每次加上固定增量，以前十位作为地址去读取 sin_rom 中读取数据

(5) Audio_clk

生成 18.434MHz 的基准时钟

(6) keyboard

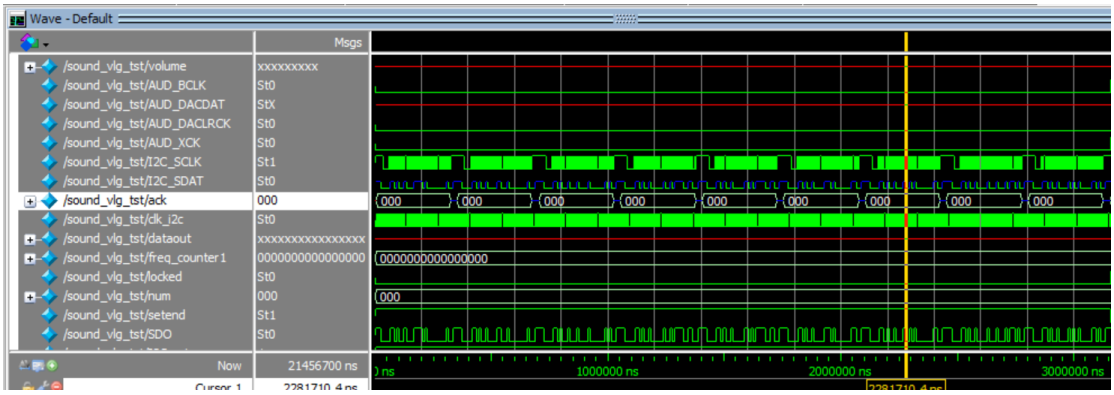
实验 8 中的键盘模块，使用到其中的 ASCII，times（按键计数），break（断码信号）

（7） sound

- 生成 10kHz 的时钟信号 clk_i2c，通过 I2C_Audio_Config 完成音频芯片配置，
- 根据 AUD_XCK 信号生成 AUD_BCK，AUD_DACLRCK，AUD_XCK 频率为 18.432MHz，AUD_BCK 为 1.536MHz，AUD_DACLRCK 为 48kHz
- 根据键盘的按键数据信号决定输出声音频率对应的增量
- 根据 Sin_Generator 的 dataout 生成 AUD_DACDAT

五．功能实现及仿真模拟

(1) 芯片配置

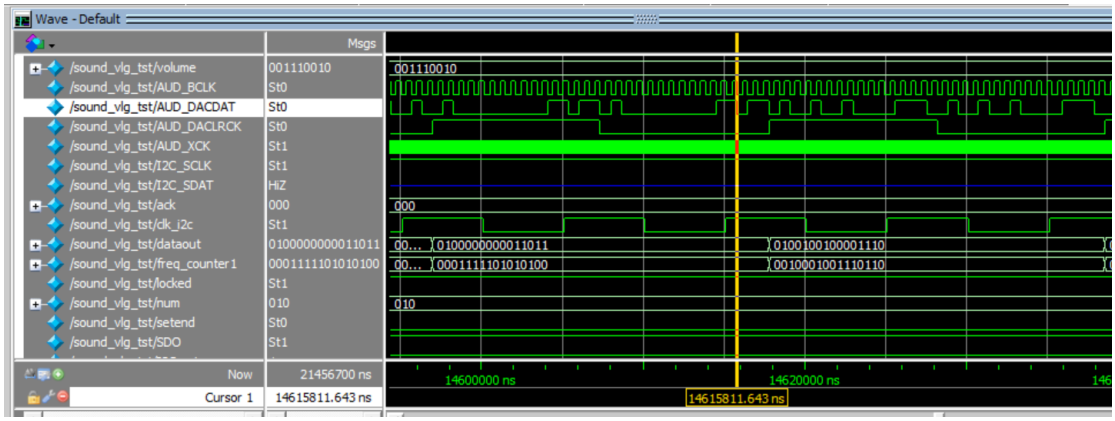


显示了发送九条命令的过程

(2) 按键发音

键盘	音名	频率/Hz	增量
Q	C4	261.63	357
W	D4	293.66	401
E	E4	329.63	450
R	F4	349.23	477
T	G4	392.00	535
Y	A4	440.00	601
U	B4	493.88	674
I	C5	523.25	714
O	D5	587.33	802
P	E5	659.26	900

根据 times 的变化判断是否有按键，有则根据 ASCII 码通过上表得到对应的增量，从而输出正弦波。num 寄存器记录按下的按键个数，times 变化加一，break 信号有效减一，Sin_generator 根据 num 信号决定开启几个 16 位计数器读取 sin 函数表



显示了两次左右声道发送的情况，可以看到 AUD_DACDAT 和 AUD_DACLRCK 和 AUD_BCLK 的下降沿对齐，左右声道传递数据相同，此时 num=010，代表按下了两个键（但只显示了一个 16 位计数器 freq_counter1 的情况）

（3）音量控制

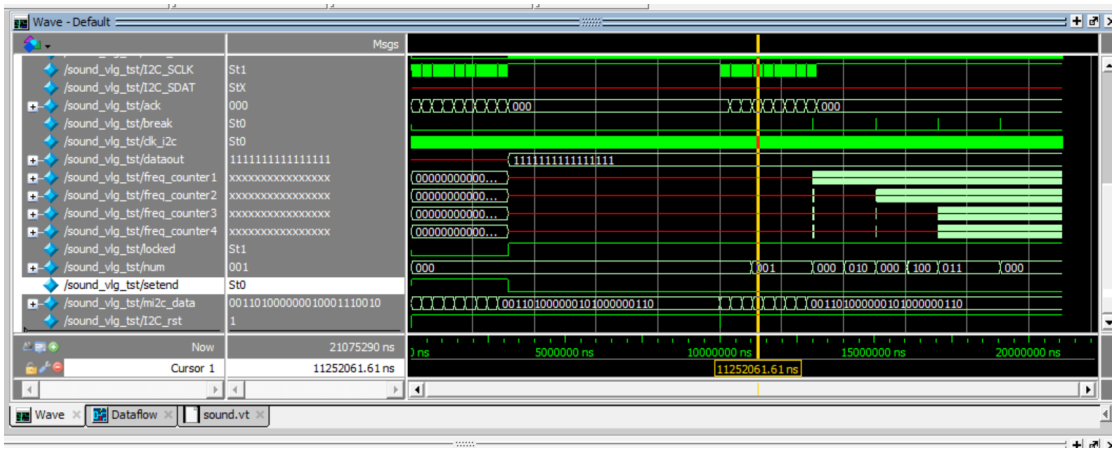
给芯片设置模块 I2C_Audio_Config 发送 RESET 信号，同时给出需要设定音量，绕过 audio_cmd[3]直接在发送音量信号时将数据 mi2c_data 改成给出的 volume，完成音量设置。

（4）和音

正弦波生成模块 Sin_Generator 一共设置四个计数器，会根据 num 开启对应数目的计数器，

如 num=3‘b010 时，只有 freq_counter1 和 freq_counter2 会计数，其余一直置 0，四个计数器的前十位作为地址去 sin_rom 中读取数据，这样读到四个数据，再根据 num 决定用哪些数据合成最终数据，如 num=3’b010 时，data1 和 data2 的数据分别除 2 相加，num=3’b100 时，四个数据除四后相加，除法使用算术右移完成

下图显示的是音量设置和和音的模拟，可以看到 mi2c_data 的后 7 位为 7‘h72，即为给定的音量数据，设置完毕后分别按下一个，两个，四个键，四个计数器依次开启



六．实验反思和总结

- I2C_SDAT 是 inout 接口，既能输出也能输入，开始设置为 output，导致接受 ack 信号失败，显示 ack 不全为 0
- inout 接口在仿真模拟时 testbench 会生成一个 treg_I2C_SDAT 变量作为其输入，在本实验中，assign I2C_SDAT= treg_I2C_SDAT，由于没有对 treg_I2C_SDAT 设置，导致得不到写音频芯片的输出，把 assign 语句注释即可得到输出
- audio_cmd 按 ram 寄存器实现，需要时钟信号和地址，实验中为了得到音量值，assign v=audio_cmd[3]，这样是一直读取寄存器，导致 quartus 无法生成 audio_cmd，放不出声音，后来绕过 audio_cmd，写 mi2c_data 解决
- 如果模块的引脚过多且未分配，在编译时可能会出现无法 compile,design 的情况，实验中将四个计数器设置为 output 但未分配引脚导致编译失败，取消 output 解决
- 实验中曾今尝试利用 ASCII 码的变化决定是否按下键，但这样在两次按下相同键时第二次就不会有声音，后来采用利用按键次数 times 检测是否有键按下
- 了解了使用 IP 核生成浮点数频率时钟的方法
- 了解了用多于实际位数的整数实现小数递增的方法