

数字音频压限器设计与实现^①

简献忠, 严 军, 范建鹏, 侯一欣

(上海理工大学 光电信息与计算机工程学院, 上海 200093)

摘 要: 分析了音频压限器的算法, 对起控时间和释放时间这两个参数的处理进行了说明, 并使用 MATLAB 软件和 DSP 硬件对压限算法进行了验证. 以 DSP 芯片 TMS320VC5501 和微控制器 STC89LE58RD+ 为核心构建系统平台, 给出了硬件和软件设计. 使用同一音源对音频压限器进行了测试, 将测试结果和 MATLAB 结果进行了对比, 两者平均误差 0.38%, 验证了设计方案的正确性, 满足音频压限器高精度要求, 为高精度音频压限器的设计提供了一种新的思路.

关键词: 数字信号处理; 压限器; 动态范围

Design and Implementation of Digital Audio Compressor

JIAN Xian-Zhong, YAN Jun, FAN Jian-Peng, HOU Yi-Xin

(School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: The algorithm of audio compressor is analysed. Especially describe how to process two parameters: attack time and release time. The algorithm is checked by MATLAB. DSP chip TMS320VC5501 and Microcontroller Unit STC89LE58RD are used as cores to set up the platform. Hardware design and software design are given. The compressor is tested by the same audio source which is used in the MATLAB. Compare the result of MATLAB with of compressor and mean error between them is only 0.38%. So the correctness of this design is proved and meets the need of high accuracy of audio compressor. This design can be seen as a new way to high-accuracy audio compressor.

Key words: DSP; compressor; dynamic range

1 引言

音频压限器是一种用于压缩或限制节目信号的动态范围, 避免过激失真的音频信号处理设备^[1]. 压限算法基本原理见《基于 ADSP-21262 的数字压限器设计》^[2], 该论文采用的压缩系数使用泰勒级数展开式和分段的方法来求得, 其误差为 0.6dB, 经过后级设备的放大, 误差也将同时被放大, 将不能满足专业音频压限器的音质要求. 起控时间和释放时间是压缩算法中两个最为重要的参数, 如何实时实现起控时间和释放时间的调控是音频压限器设计必须重点考虑的问题.

为减少误差, 本文尝试将压缩系数使用 C 标准库的 pow 函数直接求得. 由于使用 pow 函数的优点是其精度高, 但是 pow 函数的执行时间比较长, 因此在整

个算法实现中必须考虑时间的因素. 算法的 DSP 实现中, 每 10ms 处理一次数据, 处理函数必须在 10ms 内完成, 否则将影响下一次的数据处理. 在 DSP 算法编写完成以后, 将对算法的执行时间进行测算, 以保证在 10ms 内完成.

为了实时实现起控时间和释放时间的调控, 本文引入增益系数 K, 通过调节 K 来满足起控时间和释放时间调节, 根据不同的参数设置给出 MATLAB 仿真结果. 然后选用 TI 公司的 DSP 芯片 TMS320VC5501 作为实现压限器算法的核心, 又利用音频解码器 TLV320AIC23B-Q1 可以和 TMS320VC5501 进行无缝连接, 将采样数据无间断传送给后者, 并让后者处理. 以此来构建硬件平台. 最后, 对压限器进行了测试, 测

^① 基金项目: 科技部国家科技支撑计划(2010BAK69B22)

收稿时间: 2012-02-11; 收到修改稿时间: 2012-04-16

试结果和 MATLAB 仿真结果进行了对比, 验证了本设计方案的正确性.

2 压限算法

2.1 压限器原理及其参数

如图 1 所示, 当输入信号电平(dB)超过阈值电平(dB)时, 对信号电平按压缩比进行压缩. 当压缩比为 2:1 时, 输出电平为输入电平的一半. 同理, 压缩比为 4:1 时, 输出电平为输入电平的四分之一. 当压缩比大于 8:1 时, 此时压缩器可以称为限幅器.

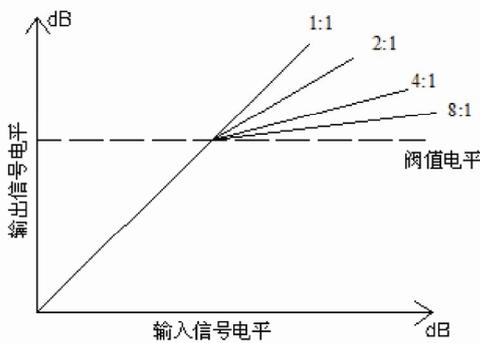


图 1 压限器算法示意图

压限器有四个参数: 阈值电平(threshold)、压缩比(compression rate)、起控时间(attack time)和释放(release time). 阈值电平用于决定是否对信号进行压缩. 压缩比是输入信号与输出信号的分贝数之比. 起控时间是当信号电平超过阈值电平时, 压限器开始进入压缩状态所需的时间. 释放时间是当信号电平低于阈值电平后, 压限器从压缩状态退出所需要的时间.

2.2 有效值

当音频信号电平大于阈值电压时, 压限器才开始工作. 因此, 先求出信号当前的有效值. 求有效值时, 需对信号进行加窗, 窗口大小为 10ms^[2], 即每 10ms 求一次有效值并判断该信号电平是否超过阈值电平, 如超过则进行压限处理. 有效值使用均方根来近似表达, 其公式如下:

$$RMSM = \sqrt{\sum_{i=0}^{440} x(i)^2 / 441}$$

其中, x(i)为音频信号采样值, 由于音源信号的采样频率为 44.1kHz, 因此, 10ms 时间共 441 个采样值.

2.3 增益系数

从 1.1 节可知, 输入电平与输出电平之间的关系

是在采用 db 表示的前提下建立的, 接着, 将这种关系转换成信号在幅值上的关系.

如图 1 所示, 建立如下方程:

$$RMSY^d = \begin{cases} RMSX^d, & RMSX^d \leq T^d \\ T^d + (RMSX^d - T^d)/R, & RMSX^d \geq T^d \end{cases}$$

最后可得压缩系数:

$$\beta = x^{-\alpha [2]}$$

其中, $x=RMSY/RMSX$, $\alpha = 1-1/R$, R 是压缩比, $R \geq 1$.

这里将 $K = 1/\beta$ 称为增益系数, 可得到输出信号与输入信号在幅值上的关系式:

$$RMSY = RMSX \times K, \text{ 其中 } 0 < K < 1.$$

2.4 起控和释放处理

当压限器未工作时, 可将压限器的增益系数视为 1, 即单位增益. 当压限器工作时, 其增益为 K. 根据 2.1 节对起控时间和释放时间的阐述, 可以这样做: 在起控时间段内将信号增益由单位增益逐步下降至 K, 在释放时间段内将信号增益由 K 逐步上升至单位增益.

使用 MATLAB 7.0 软件进行算法仿真. 原始信号为:

$$x(t) = \begin{cases} \sin(600\pi t), & 0 \leq t \leq 0.03, 0.06 \leq t \leq 0.1 \\ 2\sin(600\pi t), & 0.03 \leq t \leq 0.06 \end{cases}$$

阈值电平 0dB, 压缩比 3:1, 起控时间和释放时间都为 10ms. 图 2 为分别显示了原始信号和处理过后的信号, 从中可以明显看到 0.03s 至 0.04s 为起控过程, 增益逐渐减小, 0.06s 至 0.07s 为释放时间, 增益逐渐增大, 恢复至单位增益.

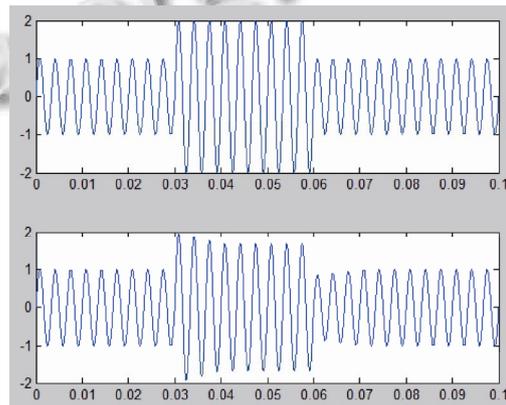


图 2 MATLAB 仿真

图 3 是在不同参数情况下压限器的效果. 两者的阈值电平均为 0dB. 上半图, 起控时间 20ms, 释放时间 30ms, 压缩比 3:1. 下半图, 起控时间 10ms, 释放时间 20ms, 压缩比 6:1.

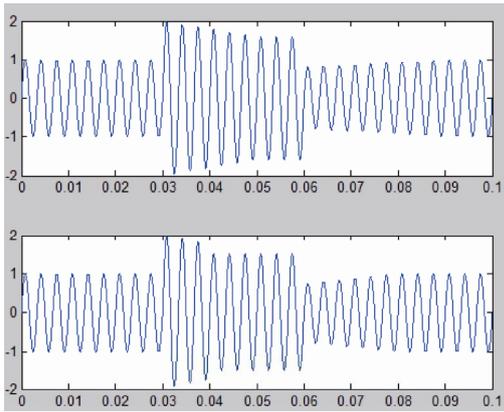


图 3 不同参数的压限器效果

2.5 音频测试

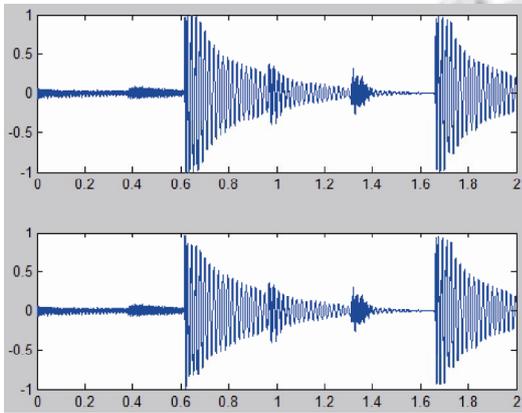


图 4 音乐信号仿真

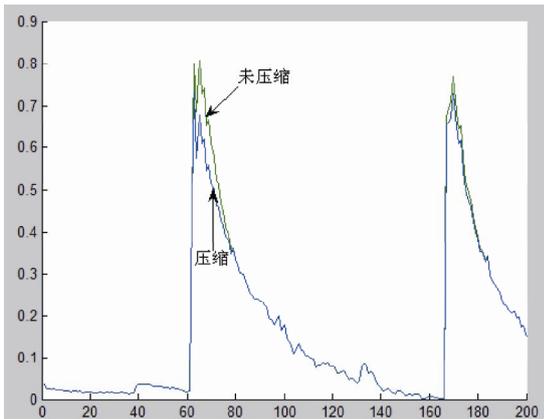


图 5 未压缩和压缩状态下的均方根

使用音乐对压限器算法进行了测试. 音源采样频率 44.1kHz, 单声道. 压限器参数: 阈值电平-4dB, 压缩比 4: 1, 起控时间 10ms, 释放时间 100ms. 结合图 4 和图 5, 可以观察到在 0.65 秒和 1.7 秒附近, 音频信号的幅值变化. 通过试听, 可以感觉到在这两个时间点

附近的鼓声强度被削弱, 造成低频下潜深度减少. 因此, 可以看出来本文提出的算法是正确和有效的.

3 系统硬件设计

系统组成框图如图 6 所示. 微控制器选用 STC89LE58RD+. 首先, 使用它模拟 I²C 总线来对音频解码芯片进行设置, 并且将 4 个参数值保存至 E²PROM; 同时将这些参数通过 HPI 接口传送至 DSP 芯片, DSP 芯片将调用这 4 个参数值对音频数据进行压限处理. DSP 芯片使用 TMS320VC5501. 它是主频为 300MHz 的定点数字信号处理器, 有 6 路 DMA、2 路 McBSP 和 1 个 8 位的 HPI 接口等. DSP 芯片通过 McBSP 和音频编解码器进行双向数据通信. 音频编解码器采用 TLV320AIC23B-Q1. ; 支持 I²C 和 SPI 接口, 通过该接口可以配置该芯片的内部寄存器; 支持 McBSP 接口, 以此实现与 DSP 芯片(带 McBSP)的无缝连接. 该芯片将音源信号(模拟量)通过 A/D 转换成数字信号并通过 McBSP 传送至 DSP 芯片, DSP 芯片将数据处理完以后, 通过 McBSP 传回音频编解码器, 再通过 D/A 输出.

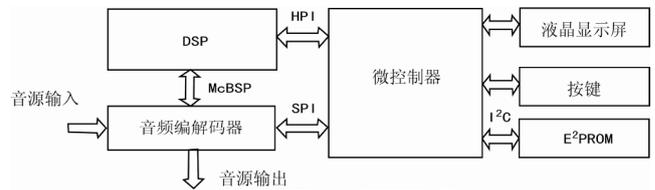


图 6 系统组成框图

4 系统软件设计

DSP 以及单片机软件流程图如图 7、8 所示.

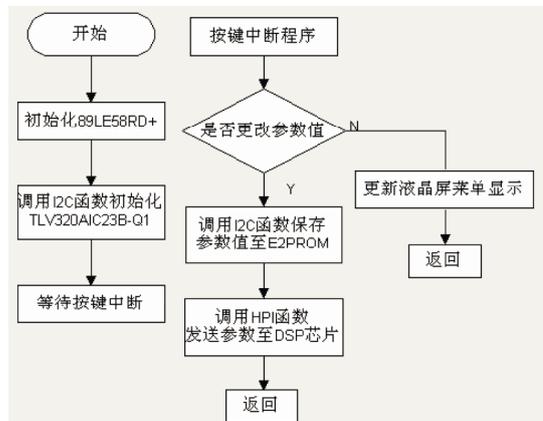


图 7 DSP 软件流程图

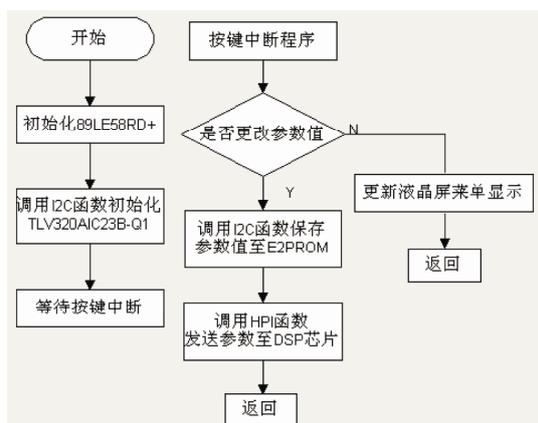


图 8 单片机软件流程图

4.1 主要功能初始化

在初始化 TLV320AIC23B-Q1 中, 使用 DSP 数据格式, 器件工作在主模式; 输入数据长度 16 位; 设置 ADC 和 DAC 采样频率为 44.1kHz(外部晶振频率 13.9344MHz); 打开 ADC 高通滤波。

在初始化 TMS320VC5501 中, 主要对 McBSP 和 DMA 进行配置. McBSP 0 的接受或者发送数据格式配置为: 单相帧, 每帧 2 个字, 字长 16 位, 无数据压缩。

DMA 通道 0 负责从 McBSP0 的数据接受缓冲器(DDR1)中搬移数据至 DARAM 的指定区域. 配置如下: 使能 frame 和 half 中断; 数据源选择为 Peripherals, 数据目的地址为 DARAM; 数据格式: 每个元素 16 位, 每帧 1764 个元素, 每个块 1 个帧; 源地址采用固定模式, 目的地址采用自动后增量模式; 通道 0 的发送由 McBSP 0 接收事件触发, 即 DDR1 每收到一次数据就启动一次 DMA 传输. 当半个块的数据传输完成时触发半帧中断, 整个块传输完成时触发全帧中断。

DMA 通道 1 负责从 DARAM 的指定区域搬移数据至 McBSP0 的数据发送缓冲器(DXR1). 配置如下: 数据源选择为 DARAM, 数据目的地为 Peripherals; 数据格式和 DMA 通道 0 相同; 源地址采用自动后增量模式, 目的地址采用固定模式; 通道 1 的发送由 McBSP 0 发送事件触发。

4.2 数据流流向

在 DARAM 中开辟一个长 7056 字节的缓冲区, 起始地址为 0x5000. A 区为接收缓冲区, 地址从 0x5000 到 0x5DC8; B 区为发送缓冲区, 地址从 0x5DC8 到 0x6B90. 每个区共 1764 个字, 用来存放两次 10ms 的音频数据, 每一次音频数据大小为 882 个字, 左右声道各 441 个字。

数据流开始时, DMA 通道 0 从 A 区起始地址开始存放数据, 当数据存放至地址 0x56E3 后, 即 10ms 音频数据传送完成, 立刻触发半帧中断, 在中断程序中调用压限算法对前 882 个字的数据进行处理, 处理完毕将数据放入 B 区的前 882 个字节, 此时 DMA 通道 1 自动从 B 区起始地址发送数据至 McBSP0. 在执行半帧中断的时候, A 区后 882 个字节的数据仍在填充, 当数据填充至 0x5DC7(此时半帧中断程序必须已经返回, 也就是说中断服务程序必须在 10ms 内完成)将触发全帧中断, 在中断程序中调用压限算法对后 882 个字的数据进行处理, 之后再处理过的数据放入 B 区的后 882 个字节, 由 DMA 通道 1 自动发送. 如此循环下去。

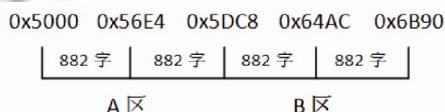


图 9 数据缓冲区示意图

5 结语

当起控时间为 10ms 时, 由于 pow 函数仅在该时间内调用, 同时该时间段需对每个点进行计算, 因此其执行时间是最长的. 使用 CCS 的 clock 功能测量出压限算法需要执行 2078187 条指令, 经计算共需 6.92ms, 小于 10ms. 接着, 用 1.5 节中的音源对压限器进行了测试, 将数据测试结果, 即均方根和 MATLAB 的进行比较. 结果平均误差仅为 0.38%, 误差非常小. 经过试听, 其效果和 MATLAB 的一致。

本文给出了音频压限器的算法, 通过 MATLAB 对算法进行仿真, 同时利用 DSP 平台实现了算法. 经过数据对比和试听, 证明了压限器设计的正确性, 具有一定的使用价值。

参考文献

- 1 Audio Compression and limiter Circuit. Security Distributing and Marketing, 2005,31(6):49-50.
- 2 祁才君,王培良.基于 ADSP-21262 的数字压限器设计.电声器件与电路,2008,32(3):23-25.
- 3 夏茂忠,严加强.视听系统安装与维护.北京:电子工业出版社,2006.58-63.
- 4 李双勋,欧建平.TLV320AIC23 在音频处理中的应用.国外电子元器件.北京,2003.16-19.
- 5 邹彦,唐东,宁志刚,王毓银.DSP 原理及应用.北京:电子工业出版社,2010.2-16.