



Y1759392

### 独创性（或创新性）声明

本人声明所呈交的论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京邮电大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名： 卫飞宇 日期： 2009.12.30

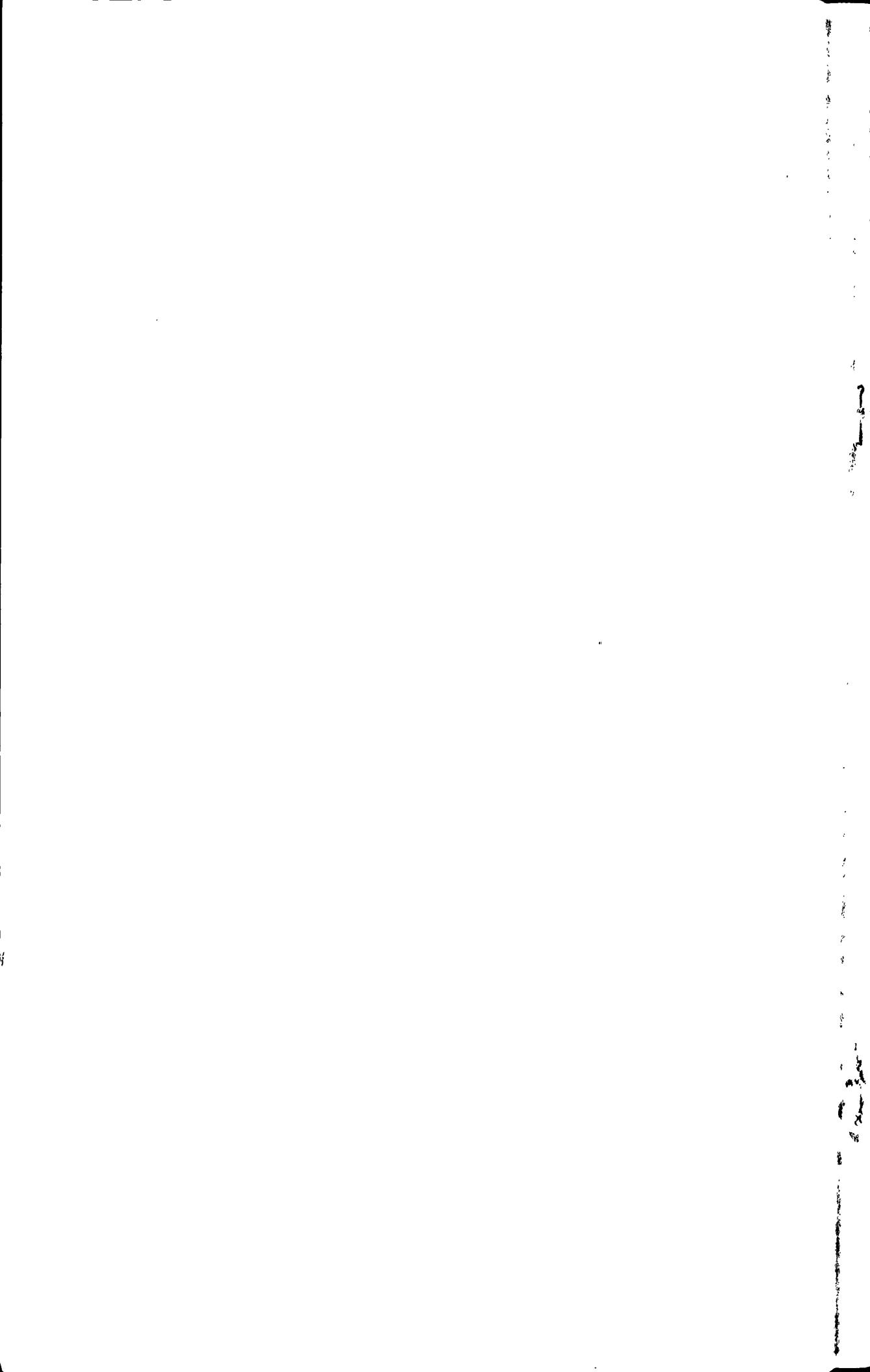
### 关于论文使用授权的说明

学位论文作者完全了解北京邮电大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属北京邮电大学。学校有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存、汇编学位论文。（保密的学位论文在解密后遵守此规定）

注释：本学位论文不属于保密范围，适用本授权书。

本人签名： 卫飞宇 日期： 2009.12.30

导师签名： 金子一 日期： 2009.12.30



## 基于 MIPS 嵌入式系统的 AC3 音频解码处理

### 摘 要

随着科学技术日新月异的发展,数字音视频技术成为信息产业的重要领域,高清数字电影、电视的应用使得人们对音频的处理要求越来越高,对其音质、立体声效果的要求也与日俱增。目前有很多音频编码格式被采用,如 MPEG、WAV、AAC、RMVB 及 AC3。其中杜比 AC3 技术在多声道音频编码领域的发展中占有一定的优势,在 DVD 中被广泛的应用。

论文根据 AC3 的特点及其音频格式规划算法实现 AC3 的软解码过程,并将其移植到 MIPS 上正确工作,课题选用的 MIPS 内核是 MIPS 24KE。在 VC6.0 的平台上实现了 AC3 软解码及其正确性的测试,并对各个解码模块的运算负载进行了评估,为后面的优化工作服务,将其移植到 MIPS 芯片上可以在 MIPS 系统中正确解码播放,并结合 MIPS 内核的特点对解码过程进行语言级的优化和 DMA 数据搬移。最后针对功能需求添加卡拉 OK 功能。

本论文的主要工作和特色:

(1) 通过调研大量的文献,分析了当前数字音视频编码压缩技术的发展现状,分析目前应用广泛的几种音频压缩格式,重点研究了杜比 AC3 的应用和发展,讨论了声音信号压缩的理论依据:人耳听觉的心理声学特性。

(2) 深入分析了 AC3 的帧格式,以及 AC3 的解码算法,对各个模块的算法实现进行深入研究。在 VC6.0 平台上实现了 AC3 解码过程,并对算法正确性进行了验证和复杂性分析。

(3) 设计 MIPS 系统流程,将解码过程其移植到 MIPS24KE 内核中处理,使其可以在 MIPS 系统中正确工作,结合 MIPS 的处理特点针对运算复杂性代码进行汇编优化。数据搬移运用 DMA 操作来优化,提高解码速率。

(4) 在 MIPS 系统中添加卡拉 OK 的功能,使系统可以处理两路音频信号,完成混音和混响的功能。

**关键词:** 音频编码技术 杜比 AC3 解码 优化 MIPS24KE



# IMPLEMENTATION OF AC3 AUDIO DECODING BASED ON MIPS EMBEDDED SYSTEM

## ABSTRACT

With the rapid development of science and technology, digital audio and video processing technology becomes a key area in information industry. The wide application of High-definition digital movies and television leads an increasing demand on audio processing as well as the sound quality and stereo effect. Currently a lot of audio encoding formats have been adopted, such as MPEG, WAV, AAC, RMVB, and AC-3. Among which Dolby AC3 is widely used in DVD due to its advantage in multi-channel audio coding.

This paper implements the soft decoding process for AC3 according to its characteristics and planning algorithm of its audio format. It works correctly when transplanted to MIPS hardware, here the core of MIPS is MIPS 24KE. Steps are as follows: implement AC3 soft decoding and correctness test in VC6.0 platform, conduct an assessment of the computation load on decoding modules for the subsequent optimization work, transplant it to MIPS chip and make sure it can be correctly decoded and play on MIPS system, perform language-level optimization and DMA data transition for decoding process based on the characteristics of MIPS, finally add Karaoke function.

The main work and features of this paper:

(1) Analyze the current status of digital audio and video coding compression technology development and analyze several currently widely used audio compression formats based on an extensive study of the literatures. The focuses are on the application and development of Dolby AC3, discuss the audio signal compression of the theoretical basis: psychoacoustics.

(2) In-depth analysis of AC3 frame format and AC3 decoding algorithm, the algorithm of each module is examined in depth. Implement AC3 decoding on

VC6.0 platform, verify the algorithms and perform complexity analysis.

(3) Design system processes for MIPS, implement decoding process in MIPS24KE and get it work correctly in the MIPS system, perform compilation level optimization for code with high computational complexity based on the characteristics of MIPS, to improve decoding speed by performing data transition optimization with DMA operation.

(4) Add karaoke function in the MIPS system, so that the system can handle two-channel audio signals to realize of mixing and reverberation functions.

**KEY WORDS:** audio coding technology Dolby AC3 decoding optimization MIPS24KE

## 目 录

第一章 绪 论.....	1
1.1 引 言.....	1
1.2 数字音频信号简介.....	2
1.2.1 人耳的听觉特性.....	2
1.2.2 音频压缩技术.....	2
1.2.3 常用音频格式简述.....	3
1.3 MIPS 简介.....	4
1.4 论文的主要内容及组织结构.....	6
1.5 本章小结.....	6
第二章 MIPS 系统设计.....	7
2.1 系统消息处理机制.....	7
2.1.1 消息队列.....	8
2.1.2 外部消息.....	8
2.1.3 内部消息.....	10
2.2 MIPS1 系统工作流程.....	10
2.3 音频解码模块说明.....	13
2.4 本章小结.....	17
第三章 AC3 音频解码处理.....	18
3.1 AC3 的帧格式.....	19
3.1.1 帧同步信息 SI.....	20
3.1.2 比特流信息 BSI.....	20
3.1.3 音频块 AB.....	20
3.1.4 CRC 奇偶校验位.....	22
3.2 AC3 音频解码.....	22
3.2.1 AC3 解码流程总述.....	22
3.2.2 指数解码.....	24
3.2.3 比特分配.....	28
3.2.4 尾数量化与解码.....	29
3.2.5 Downmix 实现.....	33
3.3 解码过程的测试及优化.....	34



3.3.1 解码测试.....	34
3.3.2 语言级优化.....	37
3.3.3 Ringbuf 的设计.....	39
3.3.4 DMA 的应用.....	40
3.4 音频播放的相关总线协议.....	41
3.4.1 I2C 总线协议.....	41
3.4.2 I2S 总线协议.....	44
3.4.3 I2C 与 I2S 在系统中的工作机制.....	46
3.5 本章小结.....	47
第四章 MIPS 系统中卡拉 OK 混响实现.....	48
4.1 系统平台设计.....	49
4.2 理论研究.....	49
4.3 具体实现.....	51
4.4 测试分析.....	53
4.5 本章小结.....	55
第五章 工作总结与展望.....	57
5.1 工作总结.....	57
5.2 工作展望.....	58
参考文献.....	60
致 谢.....	62

表 2-1 MIPS 消息的格式.....	8
表 2-2 Command 的格式.....	9
表 3-1 AC3 同步信息.....	20
表 3-2 比特流信息.....	21
表 3-3 指数策略编码.....	24
表 3-4 D15 模式差分指数映射值.....	25
表 3-5 D25 模式差分指数映射值.....	25
表 3-6 D45 模式差分指数映射值.....	26
表 3-7 LEF 指数策略编码.....	26
表 3-8 量化级别与尾数比特值.....	29
表 3-9 bap=1 (3 级量化).....	30
表 3-10 bap=2 (5 级量化).....	31
表 3-11 bap=3 (7 级量化).....	31
表 3-12 bap=4 (11 级量化).....	31
表 3-13 bap=5 (15 级量化).....	32
表 3-14 DOLBY 测试向量组分析.....	35
表 3-15 MIPS24KE 的寄存器.....	38
表 3-16 MIPS24KE 指令操作时间.....	38
图 2-1 MIPS1 上的多媒体处理框图.....	11
图 2-2 MIPS1 系统流程.....	12
图 2-3 音频解码工作流程.....	14
图 2-4 音频模块实现功能.....	17
图 3-1 AC3 同步帧格式.....	19
图 3-2 AC3 解码流程图.....	23
图 3-3 Ringbuf 设计.....	39
图 3-4 I2C 总线时序图.....	41
图 3-5 I2C 控制字节配置.....	42
图 3-6 I2S 总线工作时序.....	43
图 3-7 I2C&I2S 控制发声流程.....	46
图 4-1 卡拉 OK 混响实现系统框图.....	48
图 4-2 室内立体声的形成模型.....	49
图 4-3 室内立体声脉冲特性曲线.....	50
图 4-4 梳状滤波器及其脉冲响应和频率响应.....	50
图 4-5 全通滤波器.....	51

图 4-6 IIR 滤波器的设计框图 .....	52
图 4-7 梳状滤波器延迟时间参数的影响 .....	53
图 4-8 梳状滤波器回响采样率的影响 .....	54
图 4-9 全通滤波器回响深度的影响 .....	55

# 第一章 绪论

## 1.1 引言

随着现代计算机技术及宽带网络技术的飞速发展,对数字多媒体的需求日益增多,图像和音频的存储、处理、传输的速率和质量要求也越来越高。在保证音视频质量的情况下,要求对其操作存储的空间和传输所需的带宽越来越少,针对这一情况的解决方案就是数字压缩技术。研究发现,在音频领域直接采用 PCM 码流进行存储和传输存在非常大的冗余度。在无损的条件下对声音至少可进行 4:1 压缩<sup>[1]</sup>,也就是说只用 25%的数字量保留所有的信息,而在视频领域压缩比甚至可以达到几百倍。因而,为利用有限的资源,压缩技术从一出现便受到广泛的重视。音频压缩技术是指对原始数字音频信号流 PCM 采用适当的数字信号处理技术,在不损失有用信息量的前提下,或者所引起的损失是可以忽略的条件下,降低其码率,称为压缩编码,而且它必须具有相应的逆变换,称为解压缩或解码过程。目前广泛应用的音频格式有 MPEG、WAV、AAC、RMVB 及 AC3 等数字音视频压缩技术<sup>[2][3]</sup>。

随着科学技术的进步,人们生活水准的提高,原来的立体声形式已经不能满足大众对声音节目的欣赏要求,此时具有更强定位能力和空间效果的三维声音技术得到蓬勃发展,在三维声音技术中最具有代表性的就是多声道环绕声技术。环绕声是一种声音恢复形式,其新技术表现在随之发展起来的一些数字压缩标准上。环绕声技术发展至今已相当成熟,已日渐成为未来声音形式的主流。在对环绕声压缩的研究上产生了很多音频压缩格式,比如杜比 AC3、THX、DTS 及 MPEG-2 等<sup>[4]</sup>。这些技术在不同的场合,尤其是在电影院、家庭影院系统及将来的高清晰度电视(HDTV)等系统中得到了广泛的应用。

杜比 AC3 技术是由美国杜比实验室主要针对环绕声开发的一种音频压缩技术。杜比 AC3 提供了 5 个全频域声道和 1 个超低音声道,支持 32kbps 到 640kbps<sup>[5][6]</sup>的传输率,其标准配置为 384kbps/48kHz/5.1 声道,压缩比约为 10:1<sup>[1]</sup>,能够达到低码率传输高音质的目的。在杜比 AC3 中,使用了许多先进的行之有效的压缩技术。如前/后向混合自适应比特分配、公共比特池、TDAC 滤波、频谱包络编码、及低码率条件下使用的多声道高频耦合等。其中许多技术对其它的多声道环绕声压缩技术的发展都产生了一定的影响。

杜比数字音频压缩格式 AC3 以其高质量的音质和编码效率在高清数字电

视 (HDTV)、数字影院和消费类电子产品 (如 LD、DVD) 及直播卫星等方面获得了广泛的应用。

## 1.2 数字音频信号简介

### 1.2.1 人耳的听觉特性

人耳对声音的方位、响度、音调及音色的敏感程度是不同的, 存在较大的差异。首先介绍心理声学的几个基本概念<sup>[7][8]</sup>:

#### 1、等响度曲线

人的听觉的灵敏度随着频率而改变。即通常两个功率一样但频率不同的音调听起来并不一样响。人耳对 4KHZ 的频率最灵敏, 即在 4KHZ 下能被察觉出来的声音压力水平 (响度), 在其他频率下并不能被察觉。这就给在一些不太灵敏的频率下失真提供了条件。

#### 2、声音掩蔽

所谓声音掩蔽就是强的声音信号把弱的声音信号覆盖, 导致我们无法察觉。而且, 当两个声音在时间和频率上很接近时, 屏蔽效应就会很强。因此, 我们可以在编码时对被屏蔽的部分不编码、不传送。这样, 音质依然没有大的损失, 人耳也不易察觉。

#### 3、临界频带

对于人类的听觉来说, 对声音的感知特性并不是以线形频率为尺度来变化的, 可以用被称为临界频带的一系列有限的频段来表达。把整个频带划分成几段, 在这每个频段里, 人耳的听觉感知是相同的, 即心理声学特性都是一样的。我们把可以听到的声音按倍频关系分为 3 份, 确定低、中、高频段。低音频段 20Hz~160Hz, 中音频段 160Hz~2500Hz, 高音频段 2500Hz~20000Hz。

### 1.2.2 音频压缩技术

音频信号数字化以后需要很大的存储容量来存放, 所以很早就有人开始研究音频信号的压缩问题。音频信号的压缩不同于计算机中二进制信号的压缩, 在计算机中, 二进制信号的压缩必须是无损的, 也就是说, 信号经过压缩和解压缩以后, 必须和原来的信号完全一样, 不能有一个比特的错误, 这种压缩称为无损压缩。但是音频信号的压缩就不一样, 它的压缩可以是有损的, 只要压

缩以后的声音和原来的声音听上去一样就可以了。因为人的耳朵对某些失真并不灵敏，所以，压缩时的潜力就比较大，也就是说压缩的比例可以很大。音频信号在采用各种标准的无损压缩时，其压缩比顶多可以达到 1.4 倍，但在采用有损压缩时其压缩比就可以很高。

音频压缩的原理<sup>[7]</sup>：利用人耳听觉的心理声学特性（频谱掩蔽特性和时间掩蔽特性等）以及人耳对信号幅度、频率、时间的有限分辨能力，编码时凡是人耳感觉不到的频率不编码、不传送，也就是说凡是对人耳辨别声音信号的强度、声调、方位没有贡献的部分（称为不相关部分或无关部分）都不编码和传送。对感觉不到的部分进行编码时，允许有较大的量化失真、并使其处于听阈（即人耳所能听到的最低音量）以下，人耳仍然感觉不到。

### 1.2.3 常用音频格式简述

PCM 脉冲编码调制是 Pulse Code Modulation 的缩写。PCM 编码的最大的优点就是音质好，最大的缺点就是体积大。我们常见的 Audio CD 就采用了 PCM 编码，一张光盘的容量只能容纳 72 分钟的音乐信息。属于无损编码的范畴。

MP3 是目前应用最广泛的有损压缩数字音频格式之一，全称是 MPEG (Moving Picture Experts Group) AudioLayer-3。MP3 文件是由帧 (frame) 构成的，帧是 MP3 文件最小的组成单位。MP3 的帧速度大概是 30 帧/秒。每个帧又由帧头和帧数据组成，帧头记录着该帧的基本信息，包括位率索引和采样率索引。MP3 编码可以使音质达到现在的水平，有两次飞跃：人体听觉心理学模型的导入和 VBR 技术的应用。VBR 是 Variable Bit Rate 的缩写，意思是可变比率，就是 MP3 文件压制的时候声音元素较多，比率较高时，将自动减低压缩比特率，在比特率需求比较低时自动升高比特率，这样做的目的是在保证音质基本不被损害的情况下增加文件在线播放时的速度，和减少在本机播放时所占的系统资源。

AAC 是高级音频编码 (Advanced Audio Coding) 的缩写，它是由 Fraunhofer 研究院、杜比和 AT&T 共同研发的。AAC 是 MPEG-2 规范的一部分，它适用于从速率 8Kbps 的单声道电话音质到 160Kbps 多声道的超高质量音频范围内的编码。AAC 与 MP3 相比，增加了诸如对立体声的完美再现、码流效果音扫描、多媒体控制、降噪优化等 MP3 音频格式所没有的特性，使得在音频压缩后仍能完美地再现 CD 音质。它还同时支持多达 48 个音轨、15 个低频音轨、更多种采样率和比特率、多种语言的兼容性、更高的解码效率<sup>[9]</sup>。总之，AAC 可以在比 MP3 文件缩小 30% 的前提下提供更好的音质。

DTS 是在杜比数字环绕声出现后的又一种应用很广泛的数字环绕声系统。首先, 其对高采样率、高量化精度的数字信号, 采用了灵活、先进的相干声学编码技术, 使其编码的音乐达到试听室的水平, 高于 CD 的音质, 而多声道格式是要使得家庭影院的声音重放质量在保真度及声像准确度方面得到全面的提高, 其次是其压缩算法应是广泛适用而且灵活的, 多媒体应用限制了数据带宽, 因此需要工作在 384kbps 或更低的 5.1 声道模式。而专业音乐应用要有更高的采样频率、更长的量化数及多路分立音频通道, 并且更需要无损压缩, DTS 相干声学包括了所有这些特性, 最后就是确保了所有的解码器算法相对简单且向前兼容。

AC3 美国杜比实验室从 1980 年开始对数字音频技术进行研究, 以进一步降低比特率。分别研制 AC1 (Audio Coding-1)、AC2 和 AC3 技术。目前 AC3 在多种音频处理系统中得到广泛的应用, 美国现在已经把 Dolby AC3 标准作为高清晰度电视(HDTV)和数字有线电视的音频编码格式。

AC3 是一种高效率编码标准, 它采用自适应变换编码, 具有很多优点, 它更好地模拟了人耳的听觉特性。AC3 滤波器组的频率选择性非常接近人耳的掩蔽效应。时频变换采用了基于时域混叠抵消 (TDAC) 加窗处理, 而且窗口长度是可变的。由于音频信号的时变特性, 为了同时满足时间分辨率和频率分辨率的要求, 在编码中采用了自适应分块技术以便有效地控制预回声的发生。AC3 编码器还采用了耦合技术和组合技术。耦合技术利用人耳对高频信号不能分辨出两个频率非常接近的信号的方向的特性, 在编码器中将多个声道信号的高频部分耦合到一个公共声道中。当信道传输码率很低, 单独对各个声道进行处理仍然达不到信道要求的码率时采用耦合技术。

编码依据不同的声道用不同的比特数表示数据。根据传输码率和频率分辨率的要求采用三种模式进行编码, 三种模式的分辨率不同。AC3 技术包含着灵活性, 它可以重现一到五个声道的各种播放配置, 再加上一个任选的低频增强声道, AC3 方案可使用 32kbps 到 640kbps 之间的可调整数据率, 可配置成通道安排的各种组合。AC3 数据格式用于在已经采用了高度误码校正的环境中应用。目前 AC3 已升级至增强 E-AC3 技术规范标准<sup>[10][11][12]</sup>, 进行了多方面的扩展和补充, 具有更强的灵活性、多样性及更高的编码效率。

### 1.3 MIPS 简介

MIPS 是世界上很流行的一种 RISC 处理器。MIPS 的意思是“无内部互锁

流水级的微处理器” (Microprocessor without Interlocked Piped Stages), 其处理机制是尽量利用软件办法避免流水线中的数据相关问题。它最早是在 80 年代初期由斯坦福大学 Hennessy 教授领导的研究小组研制出来的。MIPS 公司的 R 系列就是在此基础上开发的 RISC 工业产品的微处理器。这些系列产品为很多计算机公司采用构成各种工作站和计算机系统。

MIPS 技术公司是美国著名的芯片设计公司, 它采用精简指令系统计算结构 (RISC) 来设计芯片。和英特尔采用的复杂指令系统计算结构 (CISC) 相比, RISC 具有设计更简单、设计周期更短等优点, 并可以应用更多先进的技术, 开发更快的下一代处理器。MIPS 是出现最早的商业 RISC 架构芯片之一, 新的架构集成了所有原来 MIPS 指令集, 并增加了许多更强大的功能。

MIPS 的系统结构及设计理念比较先进, 其指令系统经过通用处理器指令体系 MIPS I、MIPS II、MIPS III、MIPS IV 到 MIPS V, 嵌入式指令体系 MIPS16、MIPS32 到 MIPS64 的发展已经十分成熟。在设计理念上 MIPS 强调软硬件协同提高性能, 同时简化硬件设计。

MIPS 科技日前推出首个集成了 MIPS DSP ASE (特定应用扩展) 的 MIPS32 24KE 内核系列。24KE 内核系列采用高性能 24K 微架构, 有效地增加了 DSP 功能, 同时极大地降低了整体片上系统的裸片面积及功耗。一直以来, 很多片上系统都采用 RISC CPU 内核用于运行系统, 再采用一个单独的 DSP 内核用于信号处理。而现在, 通过具有 400-625MHz 性能范围及 DSP 功能的 24KE 内核系列, 并也不需要单独的 DSP 内核, 从而降低了整个片上系统的裸片面积, 简化了系统和软件。和没有 DSP ASE 的 RISC 执行相比较, 24KE 内核使从语音到视频等众多嵌入式应用的 DSP 性能提高 200%。

MIP32 DSP ASE 可增强性能的设备包括 VoIP 处理器、窄带和宽带通信、数字视频、图像、视频和成像。DSP ASE 加速的实例包括提高 68%效率的 IDCT 处理器 (用于视频压缩) 和加速 106%的 IMCT (用于 MP3 解码)。复杂 DSP 内核功能如 FIR 滤波器可提高 234%的性能, 标准 32×32 DCT 的饱和可改善 315%。

MIPS32 24KE 内核目标的市场包括机数字电视 (DTV)、顶盒 (STB)、语音交换机、数码相机、DVD 刻录机、IP 电话、调制解调器、打印机、汽车通信设备、住宅网关等。可通过 24KE 内核提升的典型应用包括声音处理、窄带、宽带、数字音频、图形、视频、图像等。低功耗使这些内核成为电池供电和有热限制的设备的理想选择。



## 1.4 论文的主要内容及组织结构

全文共分五章，主要分布如下：

第一章是绪论，分析了当前数字音视频编码压缩技术的发展现状，介绍了目前应用广泛的几种音频压缩格式，重点研究了杜比 AC3 的应用和发展，讨论了声音信号压缩的理论依据：人耳听觉的心理声学特性。

第二章阐述了 MIPS1 系统的工作流程，在 MIPS0 和 MIPS1 两个处理器相互协调工作下，MIPS0 为核心处理器，MIPS1 相当于 MIPS0 的外置驱动，对 MIPS0 送来的数据进行解析处理。讲述了两个 MIPS 之间的通信机制，主要讲述了 MIPS1 各个模块的功能及 MIPS1 对于音频解码的具体流程，详细的讲述了音频处理的各个模块功能。

第三章详细本章详细讲述了 AC3 的帧格式和软解码流程，重点讲述了指数解码和尾数的量化解码部分，以及多声道到两声道的 DOWNMIX 操作是具体如何实现的。在讲述完解码流程后，进一步对解码流程的测试进行了说明，列表形象的表现出对杜比向量组的测试，错误及原因分析，最后使得解码程序符合杜比向量组的要求，保证解码过程正确性。在保证解码程序正确的前提下对 AC3 解码过程进行优化，实现 DMA 数据搬移操作和汇编语言的优化。

第四章基于室内声学模型，实现 MIPS 处理器中卡拉 OK 混响功能，对麦克输入声音进行混响处理。混响处理器通过四级梳状滤波器和两级全通滤波器实现。调节各级滤波器的延迟时间、增益及混响深度可调节混响效果从而得到所需要的效果，详细分析了各个参数对混响效果的具体影响。

第五章对当前工作进行总结，并对未来工作进行了展望。

## 1.5 本章小结

本章是全文的绪论部分，通过调研大量的文献，分析了当前数字音频技术的发展现状及今后的发展趋势，多声道环绕声技术应用的必然性。介绍了目前发展比较成熟的几种音频压缩编码标准，重点讲述了杜比 AC3 技术的应用和发展，研究了音频压缩技术（无损压缩和有损压缩），并研究了音频压缩的原理：人耳听觉的心理声学特性。最后介绍了论文的主要工作与组织结构。

## 第二章 MIPS 系统设计

系统中选用的是 MIPS32 24KE 内核, 24KE 内核系列采用高性能 24K 微架构, 有效地增加了 DSP 功能, 极大地降低了整体片上系统的裸片面积及功耗。之前很多片上系统都采用 RISC CPU 内核用于运行系统, 再采用一个单独的 DSP 内核用于信号处理<sup>[1]</sup>。而现在通过具有 400 到 625MHz 性能范围及 DSP 功能的 24KE 内核系列, 不需要单独的 DSP 内核, 从而降低了整个片上系统的裸片面积, 简化了系统和软件。和没有 DSP ASE 的 RISC 执行相比较, 24KE 内核使从语音到视频等众多嵌入式应用的 DSP 性能提高 200%。

MIPS32 24KE 内核的特点是:

- 1, 更低的开发成本, 可采用现有的基于主机的工具、代码和基础架构; 消除了仅针对 DSP 的工具;
- 2, 增强的片上系统的灵活性: 采用单一的 DSP 和 RISC 设计平台, 缩短熟悉过程及上市时间;
- 3, 更低的芯片面积成本: 取消了 DSP、其他处理内核及相关附件, 降低了整个片上系统的裸片面积。

在整个系统中采用两个 MIPS 处理器进行工作, MIPS0 用作中心处理器, 主要用来控制信号, 和外界进行信号交互和处理, MIPS1 作为 CODEC 的处理芯片, 在系统中相当于 MIPS0 的外置驱动, 去分析处理数据, 如设备的初始化、音视频编解码处理等。本论文中主要讲述 MIPS1 的工作原理, 也就是数据的具体处理流程。

### 2.1 系统消息处理机制

MIPS1 系统采用循环的消息触发、消息处理机制, 它能及时响应中断并做出相应的处理, 实质上就是一个主循环的过程。消息的来源有两种: 一是 MIPS0 发给 Codec 的命令, 再经过“翻译”后的消息; 一个内部模块、中断服务程序等发送的内部消息。尽管来源不同, 但是这两种消息处理的方式是一样的。

### 2.1.1 消息队列

MIPS1 工作的动力来自源源不断的消息，为此设置了两个消息队列，队列的深度都为 100，因为正常情况下一条消息的处理速度是非常快的，队列的满度至多为 10 左右，所以设置为 100 是足够的，但是如果队列出现了溢出，那么系统必然产生了异常。这两个队列一个是中断服务程序里发送的消息，优先级高，一个是普通的消息，优先级低。每一个消息实际上就是一个无符号整数，它的各个 bit 包含不同的意义，如表 2-1 所示。

表 2-1 MIPS 消息的格式

[31:24]	[23:20]	[19:16]	[15:12]	[11:4]	[3:0]
N/A	Module type	Media type	N/A	Option type	N/A

module type: 模块类型    media type: 媒体格式    option type: 操作选择

每一次主循环开始的时候，首先去队列里获取一个消息，高优先级的消息优先处理，如果没有消息会一直等到有消息为止。根据消息包含的模块号，再将消息送到各个 Framework 的模块分别处理，处理完以后再回到主循环，继续下一个消息的获取和处理。整个过程中，不允许模块内包含无限循环或是可能产生死循环的查询，以解码器为例，每解完一帧或者一个单元数据后，必须退出，但是在退出的时候解码器模块会主动向消息队列发送一个继续解码的消息，这样当这个消息再被处理的时候，解码器才继续解下一帧或者下一个单元，如此往复直到停止解码的消息被处理。

消息队列是先进先出队列，每次队列头的消息出队后，剩余的消息会依次往前挪一位，新来的消息放在队列末尾，不支持其他的插入、删除或查找等模式。

### 2.1.2 外部消息

与外部消息息息相关的，是两个 MIPS 之间共享的命令队列、Mailbox 设备（包括 Lock 寄存器），下面做详细的介绍。

命令队列（邮箱通信机制）：

MIPS0 和 MIPS1 各自包含一个队列，二者功能完全一样。以 MIPS0 给 MIPS1 发命令为例：应用（APP）调用 PDK 发某个命令的接口，PDK 会向当前命令队列的写索引所在地方写入需要发送的命令，但是发命令之前必须有两个前提条件：

1) MIPS1 的上一个 Mailbox 中断已经处理完毕。

2) 当前写索引对应的 Lock 寄存器没有锁住。(Lock 寄存器是和 mailbox 配合使用的寄存器，一共 32 个，“写”动作清 0，“读”动作置 1)。

命令的格式如下：

表 2-2 Command 的格式

Flag(1 Byte)	Size(1 Byte)	Cmd ID(2 Byte)	
		Module (4bit)	Option (12bit)
Cmd Param0 (4 Byte)			
Cmd Param1 (4 Byte)			
Cmd Param2 (4 Byte)			
Cmd Param3 (4 Byte)			
Cmd Status (4 Byte)			
Cmd Sync (4 Byte)			
Reserved (4 Byte)			

由于会出现参数过多，一条命令不够的情况，所以采用 Flag 来区别：

Flag: 0: 继续发送/接收的数据；

1: 开始发送/接收的数据；

2: 最后发送/接收的数据；

3: 单条发送/接收的数据；

Size: Flag 为 0 时没有意义；

Flag 为 1 或 3 时表示命令所带的所有参数的大小，以字节为单位；

Flag 为 2 时表示当前这一个包的有效数据的大小；

cmd ID: 模块的类型和操作类型；

cmd param(0-3): 命令所带的数据；

cmd status: 命令完成的状态；

cmd sync: 由 mips0 产生的一个随机数，mips1 发回的时候将其发回，用于同步。

Reserved: 保留的用于扩展；

写完后将写索引加 1。然后 PDK 会设置 Mailbox 的中断寄存器，产生一个 Mailbox 的中断。MIPS1 收到 Mailbox 中断后，先去清中断，然后去检查相应的 Lock 寄存器，再从队列里将整个命令取出存放到本地，最后让读索引加 1。反之，从 MIPS0 发往 MIPS1 的命令也是类似的。MIPS0 向 MIPS1 发送的命令

采用同步返回的方式, MIPS1 处理完该条命令后, 需要把这条命令再发回 MIPS0, 发回的时候需要修改命令的状态, 返回一些必要的参数, 但是 cmd ID 和 cmd Sync 必须保持不变。对于同一设备 (PDK 把 codec 分为了不同的设备) 要发送下一条命令必须是在上一条命令返回以后才能执行, 这样也就要求 MIPS1 处理这条命令的时间不能过长, 如果是时间较长的命令, 必须先返回, 然后在完成以后再主动向 MIPS1 发送一个异步命令, 这个命令是不需要 MIPS0 做回应的。它和同步命令的区别是 cmd Sync 为 0, 而不是随机数。比如在播放音频的时候, 解码器在正确的解完第一帧后, 会向 MIPS0 发送一个设置 DAC 参数的命令, cmd id 为 F003, cmd param 会包含采样率等信息, 但是它的 cmd sync 是 0, 就表示它是一条异步命令。

由于命令和消息的格式在定义上有不同, 所以会有一个转换的过程。而本地也会保存一个命令的副本, 每一个模块只保留一个命令, 也就是当前正要处理的命令 (目前的机制是保证不会被覆盖的), 在 Framework 的对应模块工作的时候, 如果需要命令的其他的信息, 这个副本能够提供所有的信息。

### 2.1.3 内部消息

内部消息是 MIPS1 自身模块内部或者模块之间的一种激励方式, 它和外部发送的命令无关, 目前主要是两种用法: 一种是有模块需要多次调用, 在一次调用结束后发送一个消息来引起下一次调用; 一种则是在中断处理的时候, 为了让中断能尽快返回, 发送一个后续处理消息, 在中断服务结束后, MIPS1 得到这条消息以后再做后续的工作, 这样的好处是让中断尽量少的打断系统的正常运行。

## 2.2 MIPS1 系统工作流程

整个系统中 MIPS1 从 MIPS0 处通知消息队列来获取信息, 然后根据解析消息来处理相应的模块。MIPS1 的处理机制相当于一个大的循环过程, 在这个循环的过程中等待消息的触发, 如果得到消息就会启动其工作, 否则一直处于等待状态。MIPS1 的总体框架如图 2-1 所示, 从此框架图中可以看到各模块的分布位置。最上面的位置是 MIPS0 和 MIPS1 之间的接口, 然后通过 MAILBOX 机制获取消息, 通过对消息的解析进入底层相应的模块进行处理。

在 MIPS1 中主要完成的是数据的处理: MP3、AAC、RA8、AC3 音频格

式的编解码、视频解码、卡拉OK的功能实现等。下面主要介绍MIPS1的工作流程如图2-2所示，以及音频解码流程在整个系统中的处理方式。

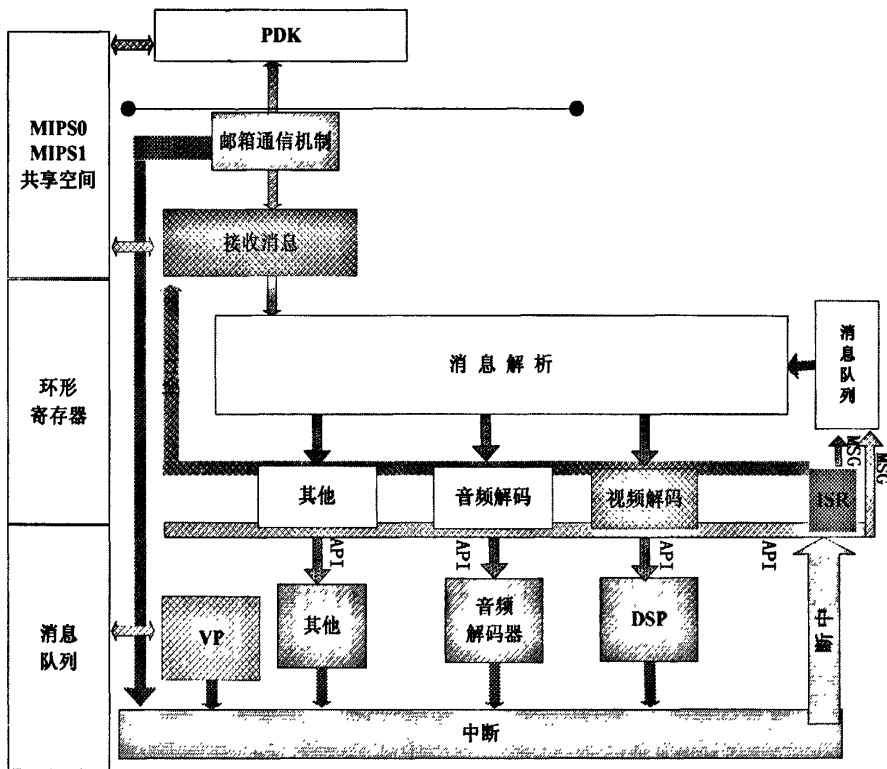


图 2-1 MIPS1 上的多媒体处理框图

MIPS1 在打开文件时会分析文件的后缀来确定数据格式类型，并记录。得知文件类型后进入系统，以音频格式 AC3 处理为模块进行说明。

### 1) 初始化

进入 MIPS1 系统后首先进行初始化工作：初始化 MIPS1 上的各个设备，包括 CPU 中断的初始化、设置寄存器、设备的使能、串口初始化、设置中断程序的入口地址以及两个处理器通信的消息队列的创建；

### 2) 进入主循环

进入主循环，先从消息队列中获取信息，解析后进入相应的大分支，分为：视频解码处理模块、音频解码处理模块、音频编码处理模块、其他处理模块。如果解析后的消息对应音频解码处理，则会进入音频解码模块分支；

### 3) 进入音频处理分支

在这个分支，先解析消息，来判断对音频做怎样具体的操作，如：打开音频文件、解码、播放、关闭、暂停、恢复、定点播放以及卡拉OK的一系列对应的操作。根据解析，做相应的操作。

4) 进入具体功能模块进行处理

每个功能模块结束时都会返回相应的状态值，来表明下一步的处理分支，所以进入功能模块的时候要查询当前的音频处理状态，然后做相应的处理。

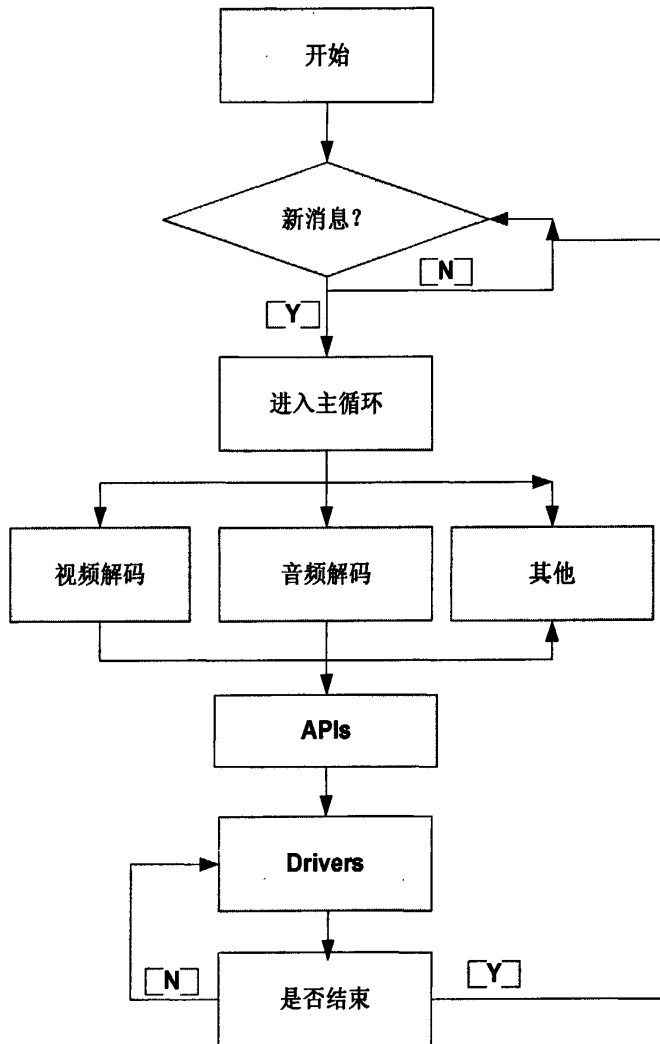


图 2-2 MIPS1 系统流程

在 AC3 解码第一次操作时进入打开音频文件模块，在打开的文件的时候会获取音频格式类型，让解码器知道所要解码的音频格式是 AC3。打开文件成功后会返回解码请求，然后在根据解码请求进入解码功能模块进行解码，每完成一帧解码返回一个值，判断该值是否解码结束，如果没有则继续下一帧的解码，否则返回播放请求，在播放处理前，会作一个判断，如果目前的状态不是停止、暂停、定点播放则正常播放，否则不处理。

以上是音频格式的正常处理流程，在这个过程中外界会有其他的功能请求，得到这些请求信息后，解析进入相应的功能处理。

MIPS0 向 MIPS1 发送的命令采用同步返回的方式，MIPS1 处理完该条命令后，需要把这条命令再发回 MIPS0，发回的时候需要修改命令的状态，返回一些必要的参数。对于同一设备要发送下一条命令必须是在上一条命令返回以后才能执行，这样也就要求 MIPS1 处理这条命令的时间不能过长，如果是时间较长的命令，必须先返回，然后在完成以后再主动向 MIPS1 发送一个异步命令，这个命令是不需要 MIPS0 做回应的。它和同步命令的区别是同步信息为 0，而不是随机数。比如在播放音频的时候，解码器在正确的解完第一帧后，会向 MIPS0 发送一个设置 DAC 参数的命令，设置参数会包含采样比特数、采样率等信息。

## 2.3 音频解码模块说明

在处理音频模块的时候，主要工作流程是先将音频文件打开，从文件后缀名获得音频文件格式，将数据放入输入 RingBuf 中，进入相应格式处理模块，从输入 RingBuf 中获取数据，首先要解析头文件获得音频数据流的采样比特数，采样率，帧长，每次处理一帧数据，完毕后判断是否要继续解码，如果是继续则从输入 RingBuf 中取数据，继续解码，将解码数据放入输出 RingBuf 中，音频输出从输出 RingBuf 读出数据即可。在解完一帧后会向 PDK 发送一个设置音频 DAC 的异步通知消息。在开始解码后，如果输出 Ring buffer 大于 8K 就把 I2S 的输出使能，直到要暂停或停止解码的时候才关闭。

在解码过程中为了保证数据的流畅和不丢失的情况，对输入 RingBuf 进行了边界判定，当其内的数据大于 4K 的时候才开始解码，防止解码速率过快，从中不能及时的得到数据，对输出 RingBuf（容量为 128K）大于 100K 时，暂停解码，当其数据被取走后小于 100K 才继续解码，以免 RingBuf 溢出，造成数据丢失。具体流程如图 2-3 所示。

所有的音频解码器都是在 MIPS1 上工作的，对于解码器的操作都必须通过消息来控制，各个模块是通过消息的传递来控制工作流程的，音频解码器模块包含以下一些消息：

EVENT_ID_AUD_DEC_OPEN_REQ	//打开文件命令
EVENT_ID_AUD_DEC_PLAY_REQ	//播放功能命令
EVENT_ID_AUD_DEC_PAUSE_REQ	//暂停功能命令



EVENT\_ID\_AUD\_DEC\_RESUME\_REQ //恢复功能命令  
 EVENT\_ID\_AUD\_DEC\_STOP\_REQ //停止功能命令  
 EVENT\_ID\_AUD\_DEC\_DOFRAME\_IND //解码功能命令  
 EVENT\_ID\_AUD\_DEC\_GET\_INFO\_IRQ //获取信息命令  
 EVENT\_ID\_AUD\_DEC\_SEEK\_REQ //定点播放命令  
 EVENT\_ID\_AUD\_DEC\_VOL\_SET //声音大小设置命令

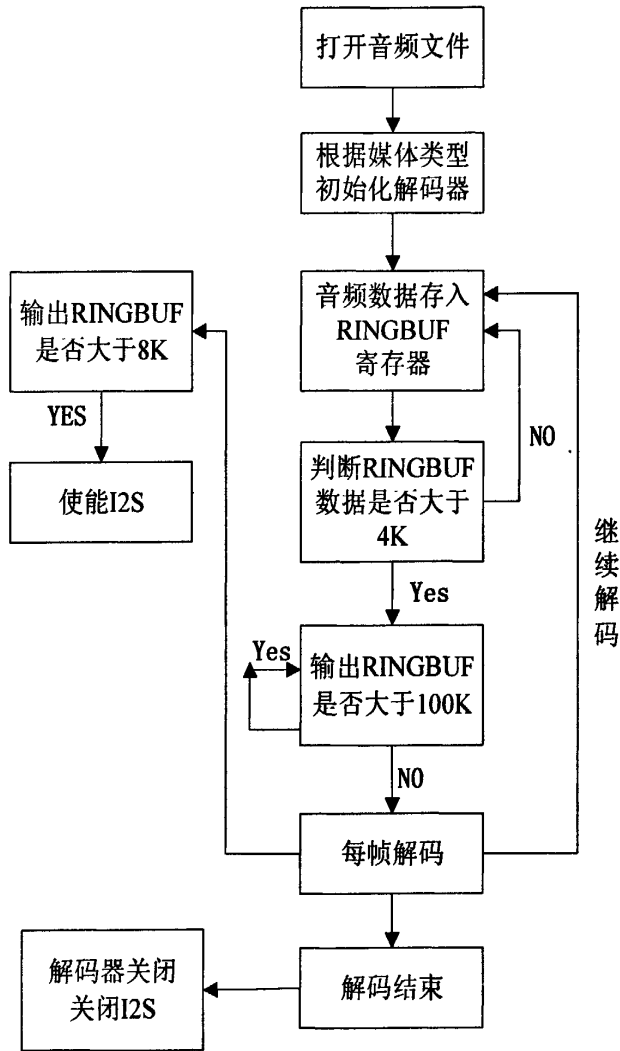


图 2-3 音频解码工作流程

MIPS 首先针对 Audio 解码器的操作封装了一套 API，包括以下模块。

1, 获取当前音频的类型;

AUDIO\_TYPE\_ID AudDec\_GetAudType(UINT32 uMsgID);

2, 打开解码器设备, 初始化解码 Buffer, 将 Framework 的 API 和解码器

的 API 关联:

AUDEC\_ERR\_T AudDec\_Open\_Handle(INT32 AudDecType);

3, 停止解码, 解除 API 的关联:

AUDEC\_ERR\_T AudDec\_Stop\_Handle(INT32 AudDecType);

4, 解码器解一帧, 将解码数据输出到 Ring buffer:

AUDEC\_ERR\_T AudDec\_DoFrame\_Handle(INT32 AudDecType);

5, 获取解码器的信息:

AUDEC\_ERR\_T AudDec\_GetHdrInfo\_Handle\

(INT32 AudDecType,AUDEC\_HDR\_INFO \*audHdrInfo);

下面会详细讲述 MIPS1 对音频处理的模块设计和各个模块的接口。包括音频的打开功能模块、播放功能模块、解码功能模块、暂停功能模块、恢复暂停模块、定点播放功能模块和关闭功能模块。每个模块都有输入输出接口设计, 接口设计方便各个模块之间通信, 具体介绍如下:

1, 打开功能模块:

MMF\_AudDecOpen\_ReqHdl()

输入接口: 消息 EVENT\_ID\_AUD\_DEC\_OPEN\_REQ

输出接口: AUDEC\_ERR\_NO: 正常, 其它: 错误。

具体描述: 根据消息里所带媒体类型, 初始化相应的解码器, 将命令携带的参数传给解码器, 如 ring buffer id 等, 详细的在 command 相关文档有说明。正常情况下, 会在退出前发送一个消息 EVENT\_ID\_AUD\_DEC\_PLAY\_REQ。

2, 播放功能模块

MMF\_AudDecPlay\_ReqHdl()

输入接口: 消息 EVENT\_ID\_AUD\_DEC\_PLAY\_REQ

输出接口: AUDEC\_ERR\_NO: 正常, 其它: 错误。

具体描述: 当输入 ring buffer 中可用数据大于阈值时, 才开始调用 DoFrame 的接口, 否则发消息 EVENT\_ID\_AUD\_DEC\_PLAY\_REQ 后退出。如果解码正确, 在解完这一帧后会向 PDK 发送一个设置音频 DAC 的异步通知消息 (实际上是 CMD), 并且发送消息 EVENT\_ID\_AUD\_DEC\_DOFRAME\_IND; 如果解码出错, 则发送消息 EVENT\_ID\_AUD\_DEC\_PLAY\_REQ, 然后退出。

3, 解码功能模块

MMF\_AudDecDoFrame\_IndHdl()

输入接口: 消息 EVENT\_ID\_AUD\_DEC\_DOFRAME\_IND

输出接口: AUDEC\_ERR\_NO: 正常, 其它: 错误。

具体描述: 首先判断输入的 Ring buffer 是否大于阈值, 同时输出 Ring buffer

必须小于 100K(总大小为 128K), 都满足的时候才能进行解码, 不满足的时候发送消息 EVENT\_ID\_AUD\_DEC\_DOFRAME\_IND 后退出。解码只解一帧。在开始解码后, 如果输出 Ring buffer 大于 8K 就把 I2S 的输出使能, 直到要暂停或停止解码的时候才关闭。

#### 4, 暂停功能模块

MMF\_AudDecPause\_ReqHdl()

输入接口: 消息 EVENT\_ID\_AUD\_DEC\_PAUSE\_REQ

输出接口: AUDEC\_ERR\_NO: 正常, 其它: 错误。

具体描述: Disable I2S 的输出。达到暂停的效果。

#### 5, 恢复暂停功能模块

MMF\_AudDecPauseResume\_ReqHdl()

输入接口: 消息 EVENT\_ID\_AUD\_DEC\_RESUME\_REQ

输出接口: AUDEC\_ERR\_NO: 正常, 其它: 错误。

具体描述: 将 I2S 输出再次使能。从暂停恢复正常。发送消息 EVENT\_ID\_AUD\_DEC\_PLAY\_REQ 后退出。

#### 6, 定点播放功能

MMF\_AudDecSeek\_ReqHdl()

输入接口: 消息 EVENT\_ID\_AUD\_DEC\_SEEK\_REQ

输出接口: AUDEC\_ERR\_NO: 正常, 其它: 错误。

具体描述: 调用 AudDec\_Stop\_Handle, 解码器关闭。

#### 7, 查找恢复功能模块

MMF\_AudDecSeekResume\_ReqHdl()

输入接口: 消息 EVENT\_ID\_AUD\_DEC\_RESUME\_REQ

输出接口: AUDEC\_ERR\_NO: 正常, 其它: 错误。

具体描述: 重新初始化解码器。调用 AudDec\_Open\_Handle, 发送消息 EVENT\_ID\_AUD\_DEC\_DOFRAME\_IND 后退出。

#### 8, 关闭功能模块

MMF\_AudDecClose\_ReqHdl()

输入接口: 消息 EVENT\_ID\_AUD\_DEC\_STOP\_REQ

输出接口: AUDEC\_ERR\_NO: 正常, 其它: 错误。

具体描述: 调用 AudDec\_Stop\_Handle, 解码器关闭。

在音频处理过程中, 除了正常的播放和关闭功能外, 音频暂停/恢复、定点随机播放、重新播放的功能的工作状态关系如图 2-4 所示。

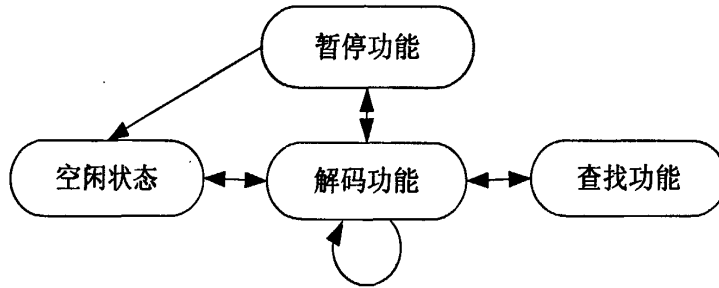


图 2-4 音频模块实现功能

以上详细的描述了音频处理的各个功能模块以及接口定义，方便各种音频格式的处理。

## 2.4 本章小结

本章首先介绍了系统选用的 MIPS32 24KE 内核特点。系统中选用两枚 MIPS 工作的：MIPS0 和 MIPS1 两个处理器相互协调工作，MIPS0 为核心处理器，MIPS1 相当于 MIPS0 的外置驱动，负责对 MIPS0 的数据进行处理。

本章首先讲述了两个 MIPS 芯片的通信机制：消息队列和邮件机制，MIPS1 的工作动力是源源不断的消息来源，当 MIPS1 接收到 MIPS0 所给的消息后，进行解析，然后进入相应的模块处理，完成处理后会反馈给 MIPS0 确认信息，两枚芯片是同步工作的。

然后详细阐述了 MIPS1 系统的整体设计和处理音频数据的工作流程。MIPS1 的处理机制相当于一个大的循环过程，各个模块的功能是相互独立的，通过模块间传递消息通信，从而协调工作，模块的设计理念是高内聚低耦合。重点讲述了 MIPS1 对于音频解码模块的设计和具体处理流程，分析了音频处理的各个模块的功能和接口设计。

## 第三章 AC3 音频解码处理

Dolby AC3 是杜比实验室开发的一种感觉数字音频编码技术,它在编码效率、声音质量、通用性方面是空前的。它从 1992 年以来在电影院里提供多声道数字声音,在 1994 年开始用在双通道 DBS (数字广播系统),它显示出的提供多通道数字环绕的能力已在消费电子领域产生了轰动,它被选中提供美国 1997 年开始的 HDTV 广播的环绕声。

Dolby AC3 是一种高效灵活的数字压缩标准,它能压缩从 1 到 5.1 个音频声道的 PCM 信号成 32kps-640kps 的比特流,音频数据一般压缩到原始数据的 1/10。AC3 比特流容许 48kHz, 44.1kHz, 或者 32kHz 的采样频率,支持多种音频模式: 1+1、1/0、2/0、3/0、2/1、3/1、2/2、3/2。DVD 和美国数字电视的音频都采用 Dolby AC3 标准,另外在卫星数字广播,数字演播室,家庭影院,娱乐媒体设备等领域也得到广泛应用<sup>[8]</sup>。AC3 采用利用人耳的掩蔽效应的先进感觉模型编码,保持了极高的音质,又极大的压缩了数据。

Dolby AC3 的特点是:

- 向下混合
- 声音响度控制
- 向后兼容性
- 完整的动态范围控制系统
- 高分辨率音频包络编码
- 复合前向/后项自适应比特分配
- 复杂度适中,非常高的编码增益
- 利用耦合技术避免了极端信号要求时的比特不足的情况
- 低的终端到终端延迟模式

Dolby AC3 数字音频压缩标准在 1994 年 11 月被采纳为 ATSC (advanced television systems committee) 采纳为标准。数字音频压缩的目的是产生音频信号的最少数字表示,使它解码以后听起来和初始声音一样。AC3 encoder 将 5.1 声道的信号多于 5Mbps (6 声道\*48kHz\*18bits = 5.184Mbps) 的 PCM 信号变成 384bps 的比特流,解码后又变为原来的 5.1 声道音频节目。

AC3 接收器接收 PCM 信号并产生跟标准一致的比特流,解码后产生满足特定应用的音频信号。AC3 算法通过粗略的量化音频信号的频域信号的频域表示实现了高的编码效率(输入和输出的比特率之比)。编码的第一步是,把时

域内的 PCM 取样数据变换成频域内成块的一系列变换系数,每个变换系数以二进制指数形式表示,即由一个指数和一个尾数构成。指数集反映信号的频谱包络,用频谱包络决定分配给每个尾数多少比特。最后由六个块的频谱包络、粗量化的尾数及相应的参数组成 AC3 数据帧格式,连续的帧汇成数码流输出,AC3 比特流是一系列 AC3 帧。

AC3 解码器的解码原理基本上是编码的反过程,解码器首先必须与编码数据流同步,经过误码纠正后从数码流中分离出控制数据、系统配置参数、编码后的频谱包络及量化后的尾数等数据。然后根据声音的频谱包络产生比特分配信息,对尾数部分进行反量化,恢复变换系数的指数和尾数。再经过合成滤波器组,把数据由频域变换到时域,最后输出重建的 PCM 样值信号。

### 3.1 AC3 的帧格式

AC3 音频的比特流是由一系列的同步帧组成的,帧与帧之间是相互独立的,AC3 的帧结构由同步字、CRC、同步信息 (SI)、码流信息 (BSI)、音频块和附加数据等组成,帧长度与 TDAC 变换的长度有关,在长度为 512 点时,帧长为 32 毫秒,即每秒 31.25 帧。每个同步帧包含 6 个音频块 (Audio Block) AB,音频块包含所有的声道信息 (一般为 5.1 声道),每个声道有 256 个采样点<sup>[5][18]</sup>。每一帧最前面的帧同步信息 SI (Synchronization Information) 包含了为获得同步和维持同步所需要的信息。后面的比特流信息 (Bit Stream Information) BSI 包含编码的参数,从中解析来服务于解码。6 个音频块后可能会存在附加信息 an auxiliary data (Aux),每一帧的最后会有 CRC 校验,用来检错。在 SI 中还有一个附加的 CRC 校验字可被解码器利用,是可选的。具体的结构如图 3-1 所示。

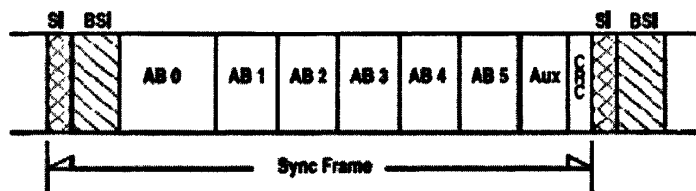


图 3-1 AC3 同步帧格式<sup>[5]</sup>

### 3.1.1 帧同步信息 SI

SI 中包含了音频的一些头信息，依次为同步字 `syncword`，附加的 CRC 校验字 `CRC1`，`fscod`，`frmsizecod`，如表 3-1 所示<sup>[5]</sup>：

表 3-1 AC3 同步信息

Syntax	Word Size
<code>syncinfo()</code>	
{	
<code>syncword</code>	16
<code>crcl</code>	16
<code>fscod</code>	2
<code>frmsizecod</code>	6
} /*endof syncinfo */	

- `syncword`: 16bits，其值为 `0x0B77`，直到数据检测到该值后才认为之后的数据为有效的帧数据
- `CRC1`: 16bits，用来检测每一帧前 5/8 的数据错误
- `fscod` (Sample Rate Code): 2bits，用来表示码流的采样率：‘00’表示 48Khz，‘01’表示 44.1Khz，‘10’表示 32Khz，‘11’保留状态。
- `frmsizecod`: Frame Size Code, 6 bits，该值对应着不同的比特率 (32-640Kbps)，与采样率相结合可以获知该帧的字长，准确获取该帧的数据进行解码处理。

### 3.1.2 比特流信息 BSI

比特流 BSI 中包含了 AC3 编码时的一些参数，BSI 中的参数可以是设置好的也可以是保留的，但是对于一些参数如果保留的话可能会造成音频无法解码，而一些数据的保留则无所谓。BSI 中包含的信息如表 3-2 所示，从中可以得到编码的格式、声道数、是否有环绕音/低重音、以及混音比率等，为解码提供了很好的信息。

### 3.1.3 音频块 AB

AB0-AB5 为 6 个音频编码块，其中第一个音频块 AB0 包含所必须的全部音频信息，后面的 AB1-AB5 可以根据相关策略复用 AB0 的信息，从而可以提

高编码效率，但帧间信息是不能复用的。一个音频块的结构如下：

表 3-2 比特流信息<sup>[5]</sup>

Syntax	Word Size
bsi()	
{	
bisd	5
bsmod	3
acmod	3
if(acmod & 0x1) && (acmod != 0x1) /* if 3 front channels */ {cmixlev}	2
if(acmod & 0x4) /* if a surround channel exists */ {surmixlev}	2
if(acmod == 0x2) /* if in 2/0 mode */ {dsurmod}	2
lfeon	1
dialnorm	5
compre	1
if(compre) {compr}	8
langcode	1
if(langcode) {langcod}	8
audprodie	1
if(audprodie)	
{	
mixlevel	5
roomtyp	2
}	
if(acmod == 0) /* if 1+1 mode (dual mono, so some items need a second value) */	
{	
dialnorm2	5
compr2e	1
if(compr2e) {compr2}	8
lngcod2e	1
if(lngcod2e) {langcod2}	8
audprodi2e	1
if(audprodi2e)	
{	
mixlevel2	5
roomtyp2	2
}	
}	
copyrightb	1
origbs	1
timecod1e	1
if(timecod1e) {timecod1}	14
timecod2e	1
if(timecod2e) {timecod2}	14
addbsie	1
if(addbsie)	
{	
addbsil	6
addbsi	(addbsil+1)× 8
}	
} /* end of bsi */	



- 块开关标志blksw[ch]: 表明当前块在由时域变换到频域时被分成了两个256个样本的子块;
- 抖动标志dithflag[ch]: 表明当前块应启用抖动策略;
- 动态范围控制dynrng: 表明对各声道增益范围的个性化调节;
- 耦合策略: 提供各声道的耦合策略信息;
- 耦合坐标: 提供耦合声道与被耦合声道的关联信息;
- 指数策略: 表明音频块内的指数复用策略信息;
- 指数: 绝对指数的编码;
- 数据分配参数: 为后面的比特指派提供参数依据;
- 尾数: 尾数的量化编码;

### 3.1.4 CRC 奇偶校验位

一帧的最后是 CRC 循环冗余校验码, 如果 SI 中的 CRC 已选, 则前者承担 BSI, AB0, AB1 的校验, 后者承担 AB2-AB5 及 AUX 的校验, 也即前者校验一帧的前 5/8, 后者校验一帧的后 3/8, 这样将一帧分为两部分校验有利于提高解码的速度。在接收完一帧前就已经对音频块 0 进行解码而不是在等到接收完一帧后, 这样不但提高了解码的速度, 同时也减少了一帧所需的缓冲存储空间, 因此 SI 中的 CRC 也一般是已选的。

## 3.2 AC3 音频解码

### 3.2.1 AC3 解码流程总述

杜比 AC3 中音频输入是以音频块为单位的, 块长度为 512 个样值, 在 48KHz 采样率时即为 10.66 毫秒, 各声道单独处理; 音频输入在经过 3Hz 高通滤波器去除直流成分后, 通过另一高频带通滤波器以检测信号的瞬变情况, 并用它来控制 TDAC 变换的长度, 以期在频域分辨率和时域分辨率之间得到最好的折中效果; TDAC 变换的长度一般为 512 点, 而数据块之间的重叠长度为 256 点, 即 TDAC 每 5.33 毫秒进行一次; 在瞬变条件下, TDAC 长度被等分为 256 点, 这样杜比 AC3 的频域分辨率为 93.75Hz, 时域最小分辨率为 2.67 毫秒。杜比 AC3 的解码系统整体上属于变换域解码范畴, 它采用的是 MDCT 变换编码, 是一种优秀的高质量音频压缩编码算法<sup>[13][14][15]</sup>。

AC3 解码器的具体流程如图 3-2 所示:

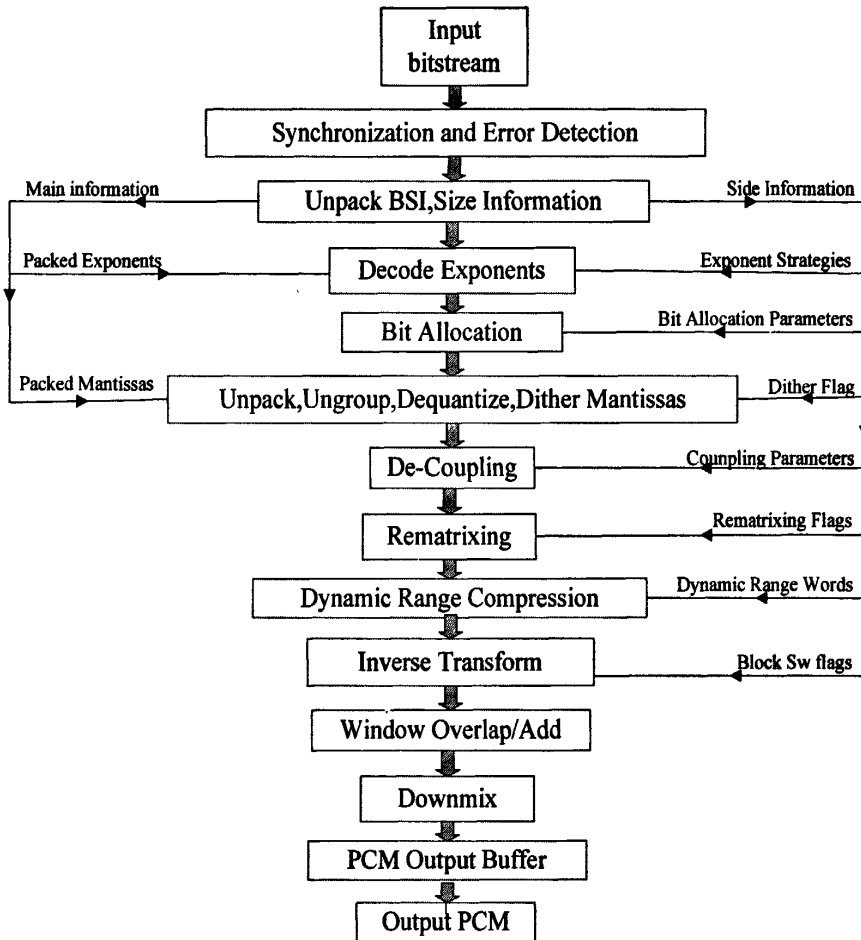


图 3-2 AC3 解码流程图

- 1) 首先对输入的 AC3 数据流进行帧同步和 CRC 校验检测。
- 2) 在确认该帧数据无错后，首先解包比特流信息，将主信息和边信息分开存放，主信息包含打包的指数和量化打包的尾数，其余则属于边信息。
- 3) 根据指数策略将编码指数值分解为实际指数，再根据比特分配的参数和指数序列计算比特分配，而不需要编码时的迭代运算，它只是一个顺序计算过程，以用于恢复编码时的比特分配指针。
- 4) 在解包尾数、分离尾数并将其转换为浮点数时，对为零的尾数则要根据编码流中相关的边信息来指定，是否采用抖动策略随机数来设置。
- 5) 如果比特流中采用了耦合策略，应将耦合声道系数依据耦合参数信息还原到各个声道。
- 6) 对于在双声道模式下，如果使用了矩阵重置，此时要进行解矩阵化的处理。依据动态范围值进行动态范围压缩处理。

7) 通过综合滤波器完成频域到时域系数的变换, 并将时域系数归一化为 16 比特的 PCM 值。

解码器获得 AC3 数据流后, 首先会检测同步字 syncword 其值为 0x0B77<sup>[5]</sup>, 直到数据检测到该值后才认为之后的数据为有效的帧数据。之后会得到相关的数据信息, 对 BSI 解包获取解码所需要的相关信息。在解码的过程中主要是对指数的解码和尾数量化, 然后根据需要进行解耦合、矩阵重置、动态范围压缩、IMDCT 反变换、声道向下混合操作<sup>[8]</sup>。

### 3.2.2 指数解码<sup>[5]</sup>

在 AC3 解码的过程中其中很重要的一个环节就是指数解码部分, 下面来详细的阐述下指数解码的过程。

对于每个独立或者耦合的声道的指数策略都存在两个比特位 chexpstr[ch] 来表示。当存在耦合声道的时候, 就会处理 chexpstr 策略代码。

在解码过程中, 首先要根据指数策略将编码指数包分解为实际的指数值。因为这些指数值实际上体现了音频样本的频谱包络。一方面要以此恢复编码样本的实际频率值, 另一方面也要利用这些编码后的样本包络值加上比特分配信息以计算出比特分配指针, 从而恢复出实际的尾数。

表 3-3 指数策略编码

chexpstr[ch], chexpstr	指数策略	每组的指数
00b	重复使用上一个指数值	0
01b	D15	3
10b	D25	6
11b	D45	12

AC3 标准中, 每个频率系数由浮点数表示, 并将它归一化为 0 到 1 之间的小数, 然后分解为指数和尾数两部分。设频率系数为  $a$  ( $0 \leq a < 1$ ):

$$a = m \times 2^{-\text{exp}} \quad \text{式(3-1)}$$

其中  $0.5 \leq m < 1$ ,  $m$  为尾数,  $\text{exp}$  为指数。也就是将  $a$  用二进制小数表示, 小数点与小数点后的第一个 1 之间的 0 的个数即为指数, 指数被限制在 0 与 24 之间, 即  $0 \leq \text{exp} < 24$ 。

若  $\text{exp} > 24$ , 则尾数  $m$  可以小于 0.5,  $\text{exp}$  仍然限制为 24。AC3 对第一个

绝对指数值限制在 0 到 15 之间, 采用 4 比特编码, 对于其它指数, 则采用差分编码对前后绝对指数之差进行编码, 这样就可用 4 比特表示, 若用直接编码则需 5 比特才能表示 0 到 24 之间的数值, 所以差分编码节省了一个比特。

AC3 音频块中指数部分信息有两种类型, 除了实际的指数编码值以外, 还含有指数编码策略的信息。在指数编码中, 数据量与频率分辨率是相互矛盾的。杜比 AC3 提供了 3 种指数编码策略来平衡这对矛盾: D15 模式、D25 模式、D45 模式。

D15 模式提供最高的频率分辨率, 但要求的数据量最大, 而 D45 模式则相反, D25 模式处于两者之间。D15 模式是将前后差分指数限制在 -2, -1, 0, +1, +2 的范围之内, 然后再将相应的差分指数转换成相应的编码值。

表 3-4 D15 模式差分指数映射值

差分指数	编码映射值
+2	4
+1	3
0	2
-1	1
-2	0

D25 模式则是对每一对指数做与 D15 相同的处理, 在处理之前, 首先要调整这对指数使它们相等。这样做节省了比特, 但也降低了频率分辨率。D45 模式则是对每 4 对指数作相同的处理, 取相等的值, 因而所占用的比特最小, 频率分辨率也最差。

表 3-5 D25 模式差分指数映射值

差分指数 n	差分指数 n+1	编码映射值
+2	0	4
+1	0	3
0	0	2
-1	0	1
-2	0	0

如果直接对 M1, M2 和 M3 编码, 则每个需要 3 个比特, 共 9 个比特位。因而对得到的编码值还要再进一步的组合后才是 AC3 比特流中指数真正的存在形式。将相邻的三个编码值 M1, M2 和 M3 合为一组, 如下式所示编码为 7 比特值:

$$\text{比特组编码值} = 25 \times M1 + 5 \times M2 + M3 \quad \text{式 (3-2)}$$

由此可见，采用这样的组合后可节省 2 个比特位。

表 3-6 D45 模式差分指数映射值

差分指数 n	差分指数 n+1	差分指数 n+2	差分指数 n+3	编码映射值
+2	0	0	0	4
+1	0	0	0	3
0	0	0	0	2
-1	0	0	0	1
-2	0	0	0	0

指数解码步骤如下：

- 1) 从比特流中获得指数编码策略信息和所有指数编码组值  $gexp$ ,
- 2) 从一组 7bit 的映射值中解出相邻的三个编码值：

$$M1 = \text{truncate}(gexp / 25) \quad \text{式 (3-3)}$$

$$M2 = \text{truncate}((gexp \% 25) / 5) \quad \text{式 (3-4)}$$

$$M3 = \text{truncate}((gexp \% 25) \% 5) \quad \text{式 (3-5)}$$

根据指数编码策略和相应的编码值计算出指数差分值序列。解差分编码以恢复出实际的指数。

当低重音声道启用时  $lfeexpstr$  参数起作用，解码对应表如下：

表 3-7 LEF 指数策略编码

$lfeexpstr$	指数策略	每组指数值
'0'	重复使用上一个指数值	0
'1'	D15	3

在比特流中紧随指数策略之后的是一系列的声道带宽比特位  $chbwcod[ch]$ ，这仅用于当前块中存在新指数的独立声道（非耦合声道），该声道带宽代码定义了该声道的最后尾数二进制值，公式<sup>[13]</sup>如下：

$$\text{endmant}[ch] = ((chbwcod[ch] + 12) \times 3) + 37 \quad \text{式 (3-6)}$$

耦合声道而言，该声道的最后尾数二进制值是由耦合声道起初的二进制值定义<sup>[13]</sup>的：

$$\text{endmant[ ch ]} = \text{cplstrtmant} \quad \text{式 (3-7)}$$

对于耦合通道, 频率带宽信息由参数耦合策略信息  $\text{cplbegf}$  和  $\text{cplendf}$  中获得。耦合通道的开始和结束尾数二进制数目<sup>[5]</sup>被定义为:

$$\text{cplstrtmant} = (\text{cplbegf} \times 12) + 37 \quad \text{式 (3-8)}$$

$$\text{cplendmant} = ((\text{cplendf} + 3) \times 12) + 37 \quad \text{式 (3-9)}$$

如果存在低频效果声道, 其总是开始于二进制值 0 拥有相同数量的尾数:

$$\text{lfestrtmant} = 0$$

$$\text{lfestrtmant} = 7$$

所有声道指数编码的第二组的字段都会标示出当前块中新的指数值。这些字段指定  $\text{exps[ch][grp]}$  服务于独立声道和耦合声道,  $\text{cplexps[grp]}$  针对于耦合声道,  $\text{lfeexps[grp]}$  针对于低频作用的声道, 其中  $\text{exps[ch][0]}$  和  $\text{lfeexps[0]}$  总是 4 比特的绝对值。对于这些声道, 指数的绝对值总是包含第一个传输系数的指数值 (二进制#0), 这 4 比特值对应于 5 比特的指数值, 范围限制在 0-15 来代替之前的 0-24。对于耦合声道的绝对指数值  $\text{cplabsexp}$  仅作为耦合声道开始解码差分指数的参考; 音频块中  $\text{cplabsexp}$  包含了的 4 比特值, 然而它映射的是 5 比特的值。

对于每个编码指数会设置分组指数 (不包括第一个绝对指数), 从比特流解码推导如下:

对于独立声道和耦合声道而言<sup>[5]</sup>:

$$\text{nchgrps[ch]} = \text{truncate} \{ (\text{endmant[ch]} - 1) / 3 \} \quad //D15 \quad \text{式 (3-10)}$$

$$\text{nchgrps[ch]} = \text{truncate} \{ (\text{endmant[ch]} - 1 + 3) / 6 \} \quad //D25 \quad \text{式 (3-11)}$$

$$\text{nchgrps[ch]} = \text{truncate} \{ (\text{endmant[ch]} - 1 + 9) / 12 \} \quad //D45 \quad \text{式 (3-12)}$$

对于耦合声道而言<sup>[13]</sup>:

$$\text{ncplgrps} = (\text{cplendmant} - \text{cplstrtmant}) / 3 \quad // D15 \quad \text{式 (3-13)}$$

$$\text{ncplgrps} = (\text{cplendmant} - \text{cplstrtmant}) / 6 \quad // D25 \quad \text{式 (3-14)}$$

$$\text{ncplgrps} = (\text{cplendmant} - \text{cplstrtmant}) / 12 \quad // D45 \quad \text{式 (3-15)}$$

对于低频作用的声道而言:

$$\text{nlfegrps} = 2$$

解码的编码设置组合指数将产生5位绝对指数，指数解码过程如下：

1) 映射值 ( $gexp$ ) 每7比特的分组解码使用该编码的逆过程进行解码<sup>[5]</sup>：

$$M1 = \text{truncate}(gexp / 25) \quad \text{式 (3-16)}$$

$$M2 = \text{truncate}((gexp \% 25) / 5) \quad \text{式 (3-17)}$$

$$M3 = \text{truncate}((gexp \% 25) \% 5) \quad \text{式 (3-18)}$$

2) 每个映射值通过减去映射偏移量转换成差分指数 ( $dexp$ )：

$$dexp = M - 2$$

3) 一系列的差异指数通过把每个差分指数加到前一个绝对指数，转换成当前绝对指数：

$$\text{exp}[n] = \text{exp}[n-1] + dexp[n] \quad \text{式 (3-19)}$$

4) 对于D25和D45模式，每个绝对指数将会被拷贝到两个或者四个其余成员中。

### 3.2.3 比特分配

Dolby AC3 采用了基于人耳的感知特性，综合了前向和后向自适应比特分配算法的优点，在一个核心后向自适应比特分配算法的基础上提出了新的混合算法：参数比特分配<sup>[5][16]</sup>。因此在AC3比特流中只需传送一些影响掩蔽曲线的形状和幅度，从而影响比特分配的心理声学模型的关键参数，而这些参数所占用的比特远小于总体比特分配信息。在不改变解码设备的情况下，编码器可以通过改变这些参数从而改善系统的性能。因此，AC3解码器必须根据编码流来产生相应的比特分配信息，并采用编码流传送过来的参数修正比特分配，它不需确定比特分配策略，只须与编码器相同的顺序计算即可。比特分配主要是通过计算解码后的频谱包络（视为功率谱密度）和掩蔽曲线的相关性来进行的；由于比特分配中采用了前/后向混合自适应比特分配以及公共比特池等技术，因而可使有限的码率在各声道之间、不同的频率分量之间获得合理的分配。

解码器中的比特分配计算模块分为以下七个步骤<sup>[5]</sup>：

- 1) 利用数码流中的相关参数初始化计算解码声道的起始和终止频率；
- 2) 把指数谱包络映射计算为功率谱密度；
- 3) 进行功率谱密度积分；
- 4) 计算激励函数；

- 5) 从激励函数计算掩盖（噪声电平阈值）曲线；
- 6) 应用从编码器发送来的对掩盖曲线的修改信息进行增量比特分配修改；
- 7) 最后计算比特分配指针，以指出把多少比特分配给每个尾数。

从解码过程分析可知，由于编码器已将从心理声学模型得到的掩盖曲线参数发送给解码器，因而解码器端不必关心编码器所采用的心理声学模型，只需从比特流中的相关参数进行解码即可，从而大大简了解码器端的投入，也更加方便实用。

### 3.2.4 尾数量化与解码

AC3 解码的过程中相对于指数来说的尾数量化也是很重要的。所有的尾数通过相应的参数 *bap* 都会量化到一个精确的固定水平。尾数使用对称量化会被量化到 15 级或更少的水平。使用常规的 2 的补码表示的非对称量化，尾数的量化级别多于 15 级。

表 3-8 量化级别与尾数比特值<sup>[5]</sup>

bap	量化级别	量化类别	尾数比特值 group bits/num in group
0	0	无	0
1	3	对称量化	1.67 (5/3)
2	5	对称量化	2.33 (7/3)
3	7	对称量化	3
4	11	对称量化	3.5 (7/2)
5	15	对称量化	4
6	32	非对称量化	5
7	64	非对称量化	6
8	128	非对称量化	7
9	256	非对称量化	8
10	512	非对称量化	9
11	1024	非对称量化	10
12	2048	非对称量化	11
13	4096	非对称量化	12
14	16384	非对称量化	14
15	65536	非对称量化	16



在解码过程中，尾数数据流解析成不同长度的单个尾数值，穿插在由三个或者成对的尾数结合的编码组中。在比特流，每个指数的尾数设置按频率升序排列。然而，这只会发生在组中的第一个尾数。组中随后的尾数比特流不会解压。

对于比特分配指针所指的一系列数值， $6 \leq \text{bap} \leq 15$ ，使用非对称 2 的小数补量化。每个尾数连同其指数是一个转换系数的浮点表示。小数点放在最高位的左边；因此尾数的字代表范围<sup>[5]</sup>：

$$(1.0 - 2^{-(\text{qntztab}[\text{bap}] - 1)}) \text{ 到 } -1.0$$

尾数 k 长度可从比特流中可以获得，为  $\text{qntztab}[\text{bap}[k]]$ 。通过右移其指数的尾数值，重新转换到定点表示，这个过程表示如下式所示<sup>[5]</sup>：

$$\text{transform\_coefficient}[k] = \text{mantissa}[k] \gg \text{exponent}[k] \quad \text{式 (3-20)}$$

没有分组是为尾数不对称量化的。bap 的取值范围是  $1 \leq \text{bap} \leq 5$ ，尾数是由编码值表示。编码后的值被转换为标准的 2 的分数补码，数值可查表获得其二进制字。由尾数的 bap 表示的比特位数从比特流中获得，并且右对齐。这种编码值被视为一个表的索引，用于查找尾数价值。由此产生的尾数右移其对应的指数值后就生成转换系数的值<sup>[5]</sup>。

$$\text{transform\_coefficient}[k] = \text{quantization\_table}[\text{mantissa\_code}[k]] \gg \text{exponent}[k]$$

$$\text{式 (3-21)}$$

尾数的编码值到实际的尾数值映射关系如表 3-9 到表 3-13 所示。

表 3-9 bap=1 (3 级量化)

尾数编码值	尾数值
0	-2/3
1	0
2	2/3

当 bap 的值为 1、2、4 的时候，通过向代表三个尾数的不同组中合并 3 级字和 5 级字，或者向代表两个尾数的不同组中合并 11 级字，进一步压缩尾数编码值。组中包含了尾数处理的顺序。如果指数集中的尾数数目在组中并不完整，则组中可以共享指数集，块中接下来的指数集会继续填写其余的组。如果 3 级或 5 级的量化转换系数的总目得出的字都不能被 3 整除，或者 11 级的字都不能被 2 整除，块最后的组会填充虚拟尾数来完成这个合成组，而这些虚拟值会被解码器忽略。用 bap 衍生出的长度，每组从比特流中抽取出来。尾数的 3 级量化 (bap=1) 可以分为三组，每组 5 比特；尾数的 5 级量化 (bap=2) 被

分为三组，每组 7 比特；尾数的 11 级量化 (bap=4) 可分为两组，每组 7 比特。

表 3-10 bap=2 (5 级量化)

尾数编码值	尾数值
0	-4/5
1	-2/5
2	0
3	2/5
4	4/5

表 3-11 bap=3 (7 级量化)

尾数编码值	尾数值
0	-6/7
1	-4/7
2	-2/7
3	0
4	2/7
5	4/7
6	6/7

表 3-12 bap=4 (11 级量化)

尾数编码值	尾数值
0	-10./11
1	-8./11
2	-6./11
3	-4./11
4	-2./11
5	0
6	2./11
7	4./11
8	6./11
9	8./11
10	10./11

表 3-13 bap=5 (15 级量化)

尾数编码值	尾数值
0	-14./15
1	-12./15
2	-10./15
3	-8./15
4	-6./15
5	-4./15
6	-2./15
7	0
8	2./15
9	4./15
10	6./15
11	8./15
12	10./15
13	12./15
14	14./15

解码器方程<sup>[5]</sup>如下:

bap=1:

$$\text{mantissa\_code}[a] = \text{truncate}(\text{group\_code} / 9) \quad \text{式(3-22)}$$

$$\text{mantissa\_code}[b] = \text{truncate}((\text{group\_code} \% 9) / 3) \quad \text{式(3-23)}$$

$$\text{mantissa\_code}[c] = (\text{group\_code} \% 9) \% 3 \quad \text{式(3-24)}$$

bap=2:

$$\text{mantissa\_code}[a] = \text{truncate}(\text{group\_code} / 25) \quad \text{式(3-25)}$$

$$\text{mantissa\_code}[b] = \text{truncate}((\text{group\_code} \% 25) / 5) \quad \text{式(3-26)}$$

$$\text{mantissa\_code}[c] = (\text{group\_code} \% 25) \% 5 \quad \text{式(3-27)}$$

bap=4:

$$\text{mantissa\_code}[a] = \text{truncate}(\text{group\_code} / 11) \quad \text{式(3-28)}$$

$$\text{mantissa\_code}[b] = \text{group\_code} \% 11 \quad \text{式 (3-29)}$$

其中尾数 a 在尾数 b 之前到来, 尾数 b 在尾数 c 之前到来。

### 3.2.5 Downmix 实现<sup>[5]</sup>

当原始音频码流的声道数大于解码后所要求的声道数时, 就要进行 Downmix 操作进行声道的整合处理。Downmix 实现是两个声道数据合并后进行 3DB 衰减获得一个声道信息。在做 3DB 衰减操作的时候采用定点操作, 3DB 衰减是原始数值的 0.707, 32 位的数据最高位为整数位, 其余为小数位, 对应的定点表示为 0x5A82799A。将 32 位的原始数据与 0x5A82799A 相乘后扩展到 64 位, 取高 32 位。

Downmix 的具体操作是输入音频数据流的音频模式和需要输出音频数据的模式共同决定的, 从输入音频模式中可以得到原始码流的的声道数和声道类型, 根据输出音频数据模式的要求对音频进行 Downmix 处理。

在许多重放系统中, 扬声器的数目不能同编码的音频声道的数目匹配。为了重现完整的音频节目需要向下混合。在帧同步中, AB0~AB5 中记录着六个独立声道的音频数据, 按照 AC3 重放时的安排, 声道类型 L、R、C、Ls、Rs、LFE。一般用于向下混合的过程中, 低音增强 LFE 通道记录的音频信号主要用于渲染烘托气氛, 所以向下混合时, 只用其中的 L、R、C、Ls、Rs。AC3 数据流可以直接传输后经解码器解码为 5.1 通道音频信息进行重放, 也可以向下混合为两个声道信号, 然后经不同的解码器得到不同的重放模式。就单一环绕声道 (n/1 模式) 而言, 把 S 称为单个环绕声道。从图中可看出, 向下混合提供两种类型: 向下混合为 Lt、Rt 矩阵环绕编码的立体声对; 向下混合为通常的立体声信号 Lo、Ro。向下混合的立体声信号 (Lo、Ro 或 Lt、Rt) 可进一步向下混合为单声道 M, 通过两个声道简单的相加即可。如果将 Lt、Rt 向下混合为单声道, 环绕信息将会丢失。当希望需要一个单声道信号时则 Lo、Ro 向下混合更可取。

用于 Lo、Ro 立体声信号的一般 3/2 向下混合方程式为:

$$\text{Lo} = 1.0'L + \text{clev}'C + \text{slev}'Ls \quad \text{式 (3-30)}$$

$$\text{Ro} = 1.0'R + \text{clev}'C + \text{slev}'Rs \quad \text{式 (3-31)}$$

如果接着 Lo、Ro 被组合成单声道信号重放, 有效的向下混合方程式为:

$$\text{M} = 1.0'L + 2.0'\text{clev}'C + 1.0'R + \text{slev}'Ls + \text{slev}'Rs \quad \text{式 (3-32)}$$

如果只出现单个环绕声道 S (3/1 模式), 则向下混合方程式为:

$$Lo = 1.0'L + clev'C + 0.7'slev'S \quad \text{式 (3-33)}$$

$$Ro = 1.0'R + clev'C + 0.7'slev'S \quad \text{式 (3-34)}$$

$$M = 1.0'L + 2.0'clev'C + 1.0'R + 1.4'slev'S \quad \text{式 (3-35)}$$

其中  $clev$ 、 $slev$  分别代表中间声道混合声级系数和环绕声道混合声级系数, 在 BSI 数据中由  $Cmixlev$ 、 $Surmixlev$  比特字段来指出相对应的值。

用于  $Lt$ 、 $Rt$  立体声信号的一般 3/2 向下混合方程式为:

$$Lt = 1.0'L + 0.707'C - 0.707'Lv - 0.707'Rs \quad \text{式 (3-36)}$$

$$Rt = 1.0'R + 0.707'C + 0.707'Lv + 0.707'Rs \quad \text{式 (3-37)}$$

如果只出现单个环绕声道 S (3/1 模式), 则向下混合方程式为:

$$Lt = 1.0'L + 0.707'C - 0.707'S \quad \text{式 (3-38)}$$

$$Rt = 1.0'R + 0.707'C + 0.707'S \quad \text{式 (3-39)}$$

经过对独立声道的音频信号进行不同的分配及矩阵重组, 实现了 AC3 数据流的向下兼容性, 通过不同的解码需求, 可以得到杜比数字 5.1 声道环绕声、立体声、单声道以及杜比的虚拟环绕声方式。其中  $Lo$ 、 $Ro$  与  $Lt$ 、 $Rt$  的最大区别就是  $Lt$ 、 $Rt$  是记录的全部的 L、R、环绕声的信息, 经过矩阵重解可得到环绕声信息, 而  $Lo$ 、 $Ro$  则是将环绕声信息增加支立体声信号中, 无法再重现环绕声信号信息。

### 3.3 解码过程的测试及优化

#### 3.3.1 解码测试

验证 AC 解码过程正确性的依据是杜比提供的 20 个向量组, 每个向量组针对 AC3 解码器的各个模块, 具有全面性和针对性。所以在测试的过程中主要是针对向量组的测试结果进行 DEBUG, 测试结果分析如表 3-14 所示。

在此对各个向量进行了详细的测试, 对于解码出错的向量, 根据其具体的测试功能, 分析其出错原因进行修正, 最后符合杜比向量组要求的解码过程。

表 3-14 DOLBY 测试向量组分析

编号	向量名	向量功能	测试结果	错误及原因分析
1	ACMOD (0-8)	测试各个模式的音频解码过程 acmod = 0-7	全部通过	无
2	BA (0-19)	检测比特分配的进程是否正确	BA2 错误 BA2: 第 0 帧和第一帧错误。	BA2 解码对于标准解码出现 FF7F 和 0080 解码出的结果是错的, 且错误随机, 其他的都是正确的。(第一帧的第四个 BLC)。 原因: 饱和截取的问题; 由于 OVERFLOW 没有打开。已解决。
3	BLKSZ (0-5)	测试帧内块长度的限制	全部通过	无
4	BSI (0-18)	测试 BSI 比特流对解码过程的控制	全部通过	无
5	COMP (0-41)	测试动态范围, dialog norm, compr	COMP4, 6, 7, 8, 17, 18, 19, 40 错误, COMP13, 14, 26, 27, 35, 39, 41 错误	原因: 1) -x,-y 设置定点代替浮点, 已解决。 2) 16 位交替错误, 从第二帧开始错误, xy 的设置问题与饱和截止的问题。 3) 一路输出全为 0, 数据写出时声道处理有问题, 已解决。
6	CPL (0-12)	测试耦合功能	全部通过	无
7	DITH (0-7)	测试抖动功能	全部通过	无

8	DWNMX (0-21)	测试 down mixing 的功能	全部通过	无
9	EXP (0-8)	测试解码过程中指数策略模块	全部通过	无
10	FRMSZ (0-196)	测试各种帧的大小和采样率	FRMSZ(38-63) FRMSZ (166-191) FRMSZ196	原因: (1) 帧同步字分析均出错, 不予解码。 (2) frmsizecod(6bit 0-63)有效取值是 0-37 和 fscod(2bit 0-3)有效取值是 0-2; (3) 当超出上面两个参数的任意有效值是被视为无效采样速率不解码。
11	GNRNG (0-9)	测试增益范围模块	GNRNG2/4(ster eo -m2: 2/0) GNRNG3/5(mono -m1: 1/0)	GNRNG3, 5: ac3dec_devb 输出全为 0,-m1 状态的时候 2 声道处理上的问题已解决。以上错误均存在 7FFF 和 8000 的问题, 饱和截止的问题, 已解决。
12	LFE (0-7)	测试 LFE 声道的处理	全部通过	无
13	MANTS (0-12)	测试尾数的量化与解码模块	全部通过	无
14	MTRX (0-11)	测试在 2/0 模式 rematrixing 操作	全部通过	无
15	MUSIC (0-4)	Music files for listening tests	全部通过	无
16	SYNC (0-8)	测试同步检测功能和 CRC	全部通过	由 SYNC 的数据特点, Devb 版本与标准在对

		校验功能		待这两个错误的处理不尽相同。
17	XFRM (0-14)	测试反变换模块功能	XFRM0, 1, 2, 3	发生少量分散的错误, 均是 7FFF 或 8000 的错误。已解决。
18	KARAOKE (0-6)	测试卡拉 OK 模式操作模块	KARAOKE2, 3 KARAOKE4, 5, 6	出错的原因是在版本中并没有加入 KARAOKE 的功能, 需要加入相应的功能设置。KCAPABLE 和 KAWARE 两种情况, 在 DOWNMIX 声道处理时, 有很大的区别。
19	XFMTEST (0-7)	测试反变换的信噪比表现的质量	全部通过	无
20	PHSCOR (0-7)	测试略过数据的解压缩过程和声道向下匹配正确水平。	PHSCOR0, 1, 2, 3, 4, 5, 6, 7	出错的原因是在版本中并没有加入 PHSCOR 的判断处理, 需要改进程序的编写, 使得处理更加完善。

### 3.3.2 语言级优化

对于 AC3 解码过程的优化主要是根据 MIPS24KE 的汇编指令级对解码过程中运算复杂或流水线不充分的代码段进行汇编优化, 可以使其流水线充分提高代码效率。

表 3-15 列出了 MIPS24KE 的寄存器名称以及用法, 在程序中经常用到的寄存器是 t0-t9、s0-s7, 这些寄存器主要用来在程序处理中存储临时变量, a0-a3 用来做函数的参数传递。在做优化的时候, 这些寄存器的数量是有限的, 要合理的重复利用才能满足需要。

MIPS24KE 内核采用的是高性能 24KE 微架构, 有效的增加了 DSP 的功能, 同时极大的降低了整体 SoC 的裸片面积和功能, 很多 SoC 都采用 RISC CPU



内核用于运行系统的，单独采用一枚 DSP 内核来处理信号，而 24KE 内核系列自带 DSP 功能，不需要单独的 DSP 内核了，降低了 SoC 的裸片面积，简化了系统和软件，24KE 内核使从语音到视频等众多嵌入式应用的 DSP 性能提高了 200%。24KE 具有强大的信号处理功能，自带一套汇编指令集。

表 3-15 MIPS24KE 的寄存器

寄存器名称	软件名	寄存器用法
\$0	zero	常 0 值，总返回 0 值
\$1	at	汇编语言临时寄存器
\$2..\$3	v0,v1	函数返回整型值
\$4..\$7	a0-a3	即将传入的参数
\$8..\$15	t0-t7	临时寄存器
\$16-\$23	s0-s7	存储寄存器
\$24-\$25	t8,t9	临时寄存器
\$26-\$27	\$k0,\$k1	故障处理程序(操作系统内核中保留位)
\$28 or \$gp	gp	全局数据指针
\$29 or \$sp	sp	栈指针
\$30 or \$fp	s8/fp	帧指针 (saved)
\$31	ra	返回地址

表 3-16 MIPS24KE 指令操作时间

指令	操作比特位	延迟指令数
MULT/MULTU/MADD/MADDU/MSUB/MSUBU	32 bit	0
MUL	32 bit	4
DIVU	32 bit	31
	24 bit	23
	16 bit	15
	8 bit	7
DIV	32 bit	33
	24 bit	25
	16 bit	17
	8 bit	9

从 MIPS24KE 的指令操作表中可以看到执行每条指令所需要的指令数，对

于32bit的操作乘法指令完成需要4个指令完成,除法指令则需要31个,对于各个指令都有确定的指令执行时间的。在做汇编优化的时候,要根据各个指令的特点进行合理编写,提高并行度使流水线充分,提高指令执行效率。在指数解码、尾数量化和解码过程都用汇编代码处理,软件实现滤波器的代码也是用汇编指令来处理的,免除了很多空指令的等待<sup>[26]</sup>。

另外的语言级优化就是浮点转为定点的处理,一般说来,定点数的运算在计算机中实现起来比较简单,效率较高;而浮点数的运算在计算机中实现起来比较复杂,效率相对较低<sup>[17]</sup>。

经过语言级的优化后,AC3解码速率得到了一定的提高,符合系统的要求。设计的音频解码芯片已经应用在高清数字电视中,实现功能得到了应用。

### 3.3.3 Ringbuf 的设计

在MIPS系统中设计了128个Ringbuf,音频处理和视频处理各自有独立的Ringbuf,Ringbuf中有两个指针,一个是写指针一个是读指针,这两个指针读写的方向是一致的,读指针到写指针的长度为Ringbuf中数据的长度,在硬件设计的基础上软件可以配置相应的寄存器来控制Ringbuf读写。

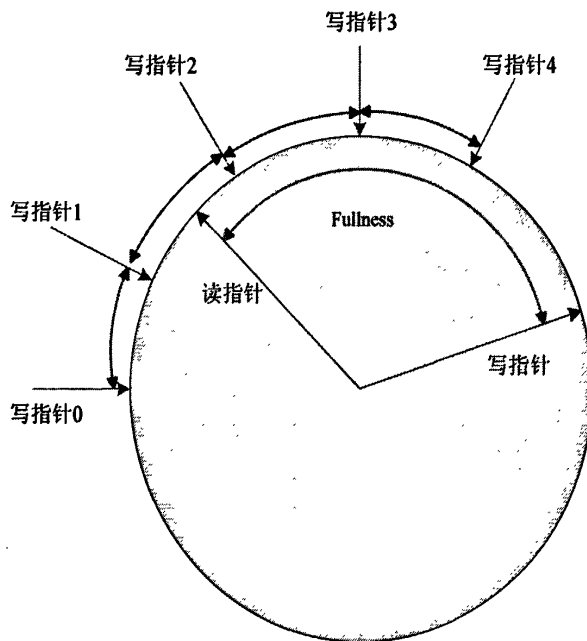


图 3-3 Ringbuf 设计

Ringbuf 软件配置中会对其进行边界保护, 在 Ringbuf 基本要空 (almost empty) 和基本要满 (almost fullness) 的时候会停止读写操作, 防止 underflow 和 overflow 的情况出现, 这里的 Ringbuf 的基本要空和基本要满的状态是读指针和写指针的相对位置来决定的。读指针应该在写指针之后要保证数据的正确提供, 当读指针超过写指针时就会发生 underflow 的情况, 当写指针超过读指针时会发生数据的覆盖, 发生 overflow 的情况, 所以边界保护是非常必要的。Ringbuf 是一段地址空间连续首尾相接的寄存器, 如图 3-3 所示。

### 3.3.4 DMA 的应用

DMA (Direct Memory Access) 直接存储器访问, 在实现 DMA 传输时, 是由 DMA 控制器直接掌管总线, 会存在总线控制权转移的问题。也就是 DMA 传输前, CPU 会把总线的控制权交给 DMA 控制器, 在 DMA 传输结束后, DMA 控制器会立即把总线控制权再交还给 CPU。这样就完成了一次 DMA 的搬移操作。

一个完整的 DMA 传输过程必须经过下面的 4 个步骤:

#### 1, DMA 提出请求

DMA 请求 CPU 对 DMA 控制器进行初始化, 并向 I/O 接口发出操作命令, I/O 接口提出 DMA 请求。

#### 2, 总线应答, DMA 开始传输

DMA 响应 DMA 控制器对 DMA 请求判别优先级的屏蔽, 向总线裁决逻辑提出总线请求。当 CPU 执行完当前总线周期就可以释放总线的控制权。此时, 总线裁决逻辑输出总线应答, 表示 DMA 已经响应, 通过 DMA 控制器通知 I/O 接口开始 DMA 传输。

#### 3, DMA 传输

DMA 传输 DMA 控制器获得总线控制权后, CPU 即刻挂起或只执行内部操作, 由 DMA 控制器输出读写命令, 直接控制 RAM 与 I/O 接口进行 DMA 传输。

#### 4, DMA 结束

DMA 结束所规定的的数据传送后, DMA 控制器立即释放总线的控制权, 并向 I/O 接口发出结束信号。当 I/O 接口收到结束信号后, 一方面停止 I/O 设备的工作, 另一方面向 CPU 提出中断请求, 使 CPU 从不介入的状态解脱, 并执行一段检查本次 DMA 传输操作正确性的代码。最后, 带着这次操作结果及状态继续执行原来的程序。

DMA 传输方式无需 CPU 直接控制传输,也没有中断处理方式那样保留现场和恢复现场的过程,通过硬件为 RAM 与 I/O 设备开辟一条直接传送数据的通路,使 CPU 的效率大为提高。

在音频解码的过程中有两次大的数据搬移,一次是将原始的音频数据搬到 Ringbuf 中为解码提供数据,另外一次是解码完成后从 Ringbuf 中读取解码后的音频数据流,使其发声。在两次的数据操作起先是采用 CPU 点对点的拷贝操作来完成的,非常费时,对于 WINDOW 系统和 LINUX 系统在做拷贝的时候还要进行大小端的调整,是非常麻烦的,采用 DMA 搬移操作来代替 CPU 点对点的操作是则不存在这个问题了,DMA 搬移无需 CPU 直接控制传输,CPU 可以去做其它的工作,大大节省了数据搬移的时间。

所谓大小端的问题,在 WINDOW 系统下采用的是小端(LSB),而 LINUX 系统下是大端(MSB),所谓 MSB 数据高字节放在高地址,低字节放在低地址,LSB 则刚好相反,数据高字节放在低地址,低字节放在高地址。ARM 则 MSB 和 LSB 都可以,是通过片外逻辑确定的。

### 3.4 音频播放的相关总线协议

#### 3.4.1 I2C 总线协议

I2C (Inter-Integrated Circuit) 总线是由飞利浦公司开发的一种总线协议,其为内部 IC 控制的两线式串行总线,用来连接微控制器及其外围设备。I2C 总线的设计起初是为音频和视频相关设备的开发,如今已经广泛应用在服务器管理中,包括单个组件状态的通信。

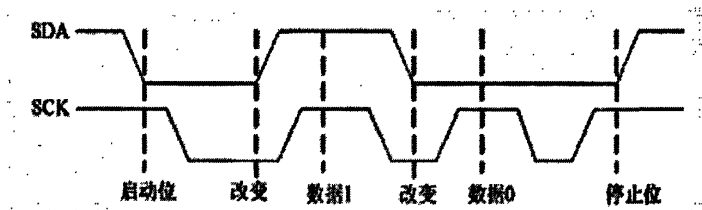


图 3-4 I2C 总线时序图

由数据线 SDA 和时钟 SCL 构成的 I2C 串行总线,可以发送和接收数据。在微处理器与被控的芯片之间、芯片与芯片之间进行双向传送,最高传送速率

可达 100kbps。被控制的各个电路都是并联在这条总线上的，每个电路或者模块只有获得自己的地址才能做出相应的工作，所以每个电路和模块都有其唯一的地址，在信息传输的过程中，I2C 总线上并行连接的每一模块电路既是主控制器又是被控器，既可以作为发送器也可以作为接收器，这取决于它要完成的功能。微处理器发出的控制信号分为地址码和控制量两部分，地址码是用来选址的，即接通需要控制的电路，确定控制的种类；控制量决定该调整参数的类别（如对比度、亮度等）及需要调整的量。所以各控制电路并联在同一条总线上，彼此之间是相互独立的，互不相关。

I2C 总线协议在传送数据的过程中总共有三种类型的信号，分别是：开始信号、结束信号和应答信号。

开始信号：时钟信号 SCL 为高电平时，数据信号 SDA 由高电平向低电平跳变，开始传送数据。

结束信号：时钟信号 SCL 为高电平时，数据信号 SDA 由低电平向高电平跳变，结束传送数据。

应答信号：接收数据的芯片在接收到 8 比特的数据后，向发送数据的芯片发出特定的低电平脉冲，表示其已经收到数据。微处理器向受控单元发出一个信号后，等待受控单元发出应答信号，微处理器接收到应答信号后，根据实际情况作出是否继续传递信号的判断，如果没有收到应答信号，会判断为受控单元出现故障。

这三个信号其中起始信号是必需的，结束信号和应答信号都不是必需的。

I2C 总线规则是运用主/从双向通讯，器件发送数据到总线上，则定义为发送器，器件从总线上接收数据则定义为接收器。主器件和从器件都可以工作于接收和发送状态。总线必须由主器件（通常为微控制器）控制，主器件会产生串行时钟（SCL）来控制总线的传输方向，并产生起始条件和停止条件。数据 SDA 线上的数据状态仅在时钟信号 SCL 为低电平时才能改变，时钟信号 SCL 为高电平的期间，数据 SDA 状态的改变被用来表示起始条件和停止条件，SDA 由高电平跳至低电平为起始条件，SDA 由低电平跳至高电平为停止条件。

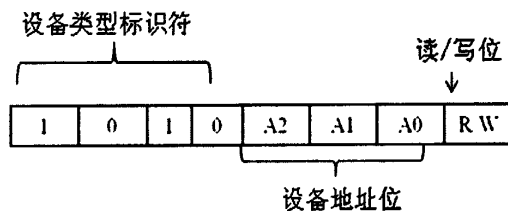


图 3-5 I2C 控制字节配置

1, 控制字节

在起始条件之后, 必须存在器件的控制字节, 控制字节的高四位为其器件类型的识别符(不同的芯片类型有不同的定义, EEPROM一般应为1010b), 接着三位为片选信号, 最后一位是控制读写位, 当其为1的时候为读操作, 为0时则是写操作。

2, 写操作

写操作分为字节写和页面写两种操作, 对于页面写是根据芯片的一次装载的字节数不同而不同。

3, 读操作

读操作有三种基本操作: 当前地址读、随机读和顺序读。

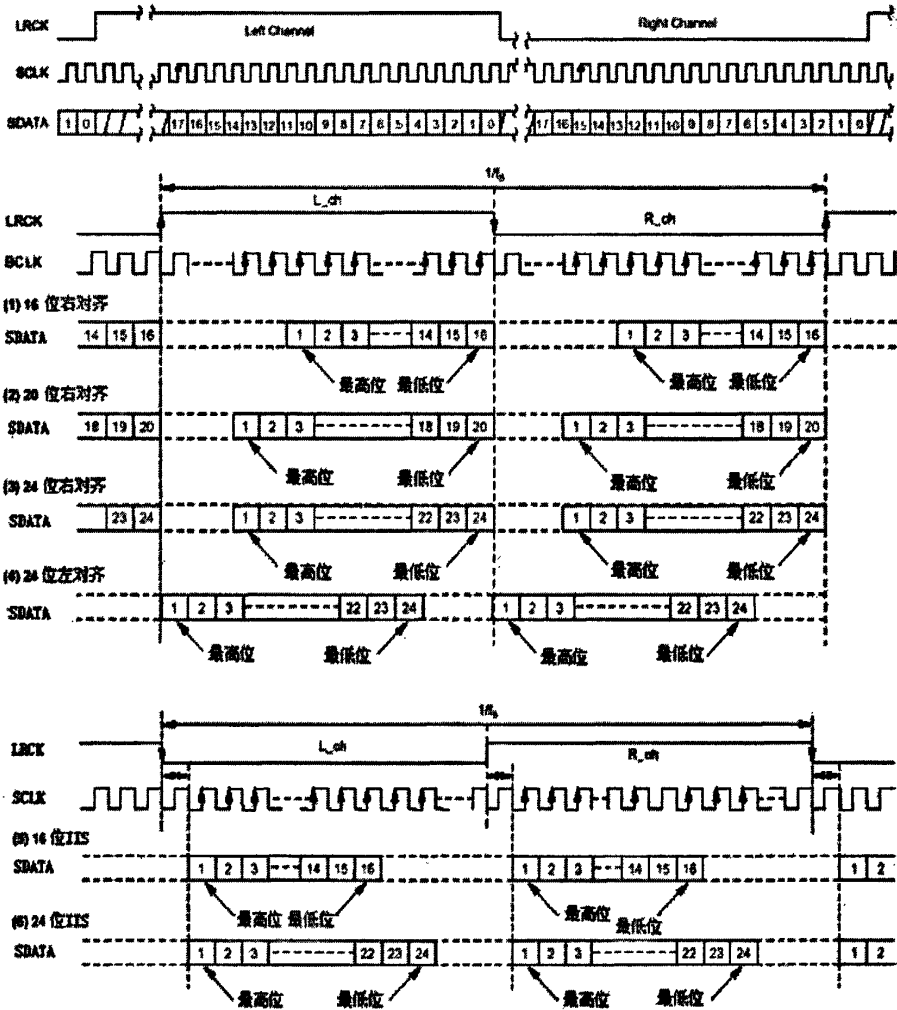


图 3-6 I2S 总线工作时序

### 3.4.2 I2S 总线协议

I2S (Inter—IC Sound) 总线是飞利浦公司为数字音频设备之间的音频数据传输而制定的一种总线标准, 该总线专责于音频设备之间的数据传输, 广泛应用于各种多媒体系统。

它采用了沿独立的导线传输时钟与数据信号的设计, 通过将数据和时钟信号分离, 避免了因时差诱发的失真, 为用户节省了购买抵抗音频抖动的专业设备的费用。在飞利浦公司的 I2S 标准中, 既规定了硬件接口规范, 也规定了数字音频数据的格式。

I2S 有 3 个主要信号:

1, 串行时钟 SCLK, 也叫位时钟 (BCLK), 即对应数字音频的每一位数据, SCLK 都有 1 个脉冲。SCLK 的频率 =  $2 \times$  采样频率  $\times$  采样位数。

2, 帧时钟 LRCK, 用于切换左右声道的数据。LRCK 为“1”表示正在传输的是左声道的数据, 为“0”则表示正在传输的是右声道的数据。LRCK 的频率等于采样频率。

3, 串行数据 SDATA, 就是用二进制补码表示的音频数据。

有的时候为了使系统间能够更好地同步, 还需要另外传输一路信号 MCLK, 称为主时钟, 也叫系统时钟, 是采样频率的 256 倍或 384 倍。

● 串行数据 (SD):

I2S 格式的信号无论有多少位有效数据 (16 位/20 位/24 位/32 位), 数据的最高位总是出现在 LRCK 变化后的第 2 个时钟 SCLK 脉冲处。这就使得接收端与发送端的有效位数可以不同。如果接收端所能处理的有效位数少于发送端所能处理的有效位数, 可以放弃数据帧中多余的低位数据; 如果接收端所能处理的有效位数多于发送端所能处理的有效位数, 可以自行补足剩余的位。这种同步机制使得数字音频设备之间的互连更加方便, 且不会造成数据错位。

对于多种不同的数据格式, 可以根据 SDATA 数据相对于 LRCK 和 SCLK 的位置不同, 分为左对齐 (较少使用)、I2S 格式 (飞利浦规定的格式) 和右对齐 (普遍使用), 如下图所示。为了保证数字音频信号的正确传输, 发送端和接收端应该采用相同的数据格式和长度, 但是对 I2S 格式来说数据长度可以是不同。

● 声道选择 (帧时钟 LRCK)

命令选择线表明了正在被传输的声道:

LRCK=0, 表示正在传输的是左声道的数据。

LRCK=1, 表示正在传输的是右声道的数据。

LRCK 可以在串行时钟的上升沿或者下降沿发生改变, 并且 LRCK 信号不需要一定是对称的。在从属设备中, LRCK 在时钟信号的上升沿发生改变。LRCK 总是在最高位传输前的一个时钟周期发生改变, 这样可以使从属设备得到与被传输的串行数据同步的时间, 并且使接收端存储当前的命令以及为下次的命令清除空间。

- 电气规范:

输出电压:  $V_L < 0.4V$        $V_H > 2.4V$

输入电压:  $V_{IL} = 0.8V$        $V_{IH} = 2.0V$

- 时序要求:

在 I2S 总线中任何设备都可以通过提供必需的时钟信号 SCLK 成为系统的主导设备, 而从属设备则可以通过外部时钟信号来得到它的内部时钟信号, 这就意味着必须重视主导设备和数据以及命令选择信号之间的传播延迟, 总的延迟主要由两部分组成: 一是外部时钟和从属设备的内部时钟之间的延迟, 二是内部时钟和数据信号以及命令选择信号之间的延迟。

对于数据和命令信号的输入, 外部时钟和内部时钟的延迟不占据主导地位, 它只是延长了有效的建立时间。延迟的主要部分是发送端的传输延迟和设置接收端所需要的时间。

假设  $T$  是时钟周期,  $T_{min}$  是最小允许时钟周期, 只有当  $T > T_{min}$  时, 发送端和接收端才能满足数据传输速率的要求。

对于所有的数据速率, 发送端和接收端都会发出一个具有固定传空比 (mark-space ratio) 的时钟信号, 所以低传空比时钟信号和高传空比时钟信号是由时钟周期  $T$  所定义的。最小传空比时钟信号和高传空比时钟信号必须大于  $0.35T$ , 这样信号在从属设备端就可以被检测到。延迟时间和最快的传输速度是相关的

为了允许数据在下降沿被记录, 时钟信号上升沿及  $T$  相关的时间延迟应该给予接收端充分的建立时间 (set-up time)。数据建立时间 (set-up time) 和保持时间 (hold time) 不能小于指定接收端的建立时间和保持时间。

I2S 总线接口可以作为一个编码解码接口与外部 8/16 位的立体声音频解码电路相连, 从而实现微唱片和便携式应用。它支持 I2S 数据格式和大段调整 (MSB) 数据格式。I2S 总线接口为先进先出队列 (FIFO) 的访问提供 DMA 传输模式来取代中断模式, 可以同时发送和接收数据, 也可以只发送数据或接收数据。



### 3.4.3 I2C 与 I2S 在系统中的工作机制

I2C 与 I2S 配置工作的流程如图 3-7 所示：

在音频解码系统中 I2C 和 I2S 共同协作来完成音频数据的输出发声，在系统中音频处理的比特位普遍为 16 比特，DVD 中 LPCM 的输出为 20bit、24bit、32bit，为了兼容多样化需求以及采样率的多样性，I2C 与 I2S 的封装要考虑周全。采样率是经过三级分频得到的，主频为 12.288MHz，经过一级分频 BCK，再经过二级分频 LCRK 得到最终的采样率，所以要得到 48KHz 的采样率，BCK 要分频 32，LCRK 分频 8，这样  $12.288\text{MHz} / (32 * 8) = 48\text{KHz}$ 。在软件设计中设置 BCK 和 LCRK 的分频数可获得所需要的采样频率。

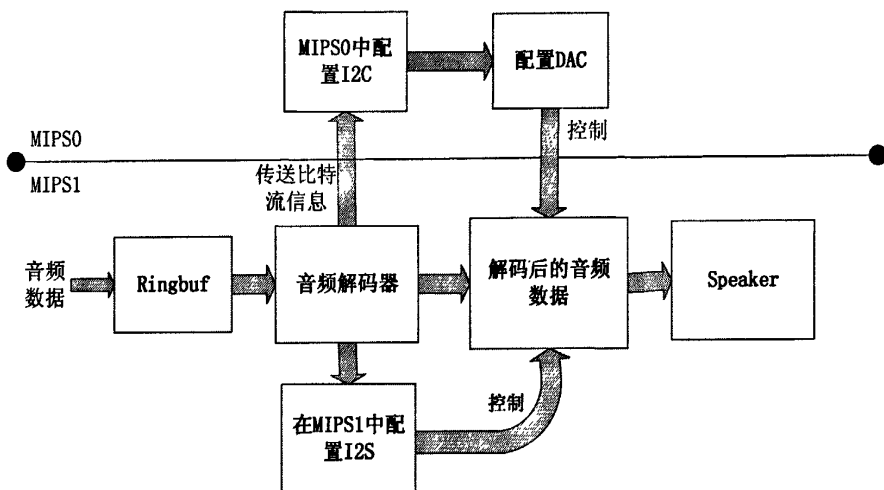


图 3-7 I2C&I2S 控制发声流程

在音频数据解完一帧后就会将从音频数据中获取的采样率和采样比特数反馈给 MIPS0，在 MIPS0 中进行 I2C 的配置，然后去配置 DAC，使 DAC 去控制数据流的输出。MIPS1 系统中当音频输出的 Ringbuf 数据流大于 8K 的时候就会配置 I2S，输出音频数据流从而发声。

I2S 硬件设计中有三个 DMA，分别为 r、t1、t2，其中 t1 和 t2 为写出操作服务，即解码后的音频数据流要读出时用这两个寄存器。r 为写入操作服务，音频的原始码流的写入时用此寄存器。I2S 会设置音频数据流读取的速率和每个采样数据的比特位。在对 I2S 使用之前要将 DMA 使能，再将 I2S 使能，紧接着对其相应的配置寄存器进行配置来控制音频数据流的输出发声。在音频数据传送结束后关闭 I2S、DMA。

### 3.5 本章小结

本章详细分析了 AC3 的帧格式和软解码流程，重点讲述了指数解码和尾数的量化解码部分，以及多声道到两声道的 DOWNMIX 操作是如何实现的。在讲述完解码流程后，进一步对解码流程的测试进行了说明，列表形象的表现出对杜比向量组的测试，错误表现及原因分析，最后使得解码程序符合杜比向量组的要求，保证解码过程的正确性。

在保证解码程序正确的前提下对 AC3 解码过程进行优化：DMA 数据搬移操作的应用和汇编语言的优化，DMA 的搬移来代替 CPU 点对点的操作，在音频数据输入输出的时候有大量的数据要进行搬移，这样大大节省了时间数据搬移时间；C 语言解码过程中有些复杂的运算存在比如 IFFT 实现，会有大量的运算，流水线很不好，合理利用 MIPS 指令集，采用汇编编码来代替 C 代码提高程序并行度，使流水线充分起来，大大提高了运算速率，另外在程序中避免浮点运算，将浮点转化成定点运算并保证其精度。

最后讲述了与音频播放息息相关的两条总线协议 I2C 和 I2S，并阐述了其在系统中的工作机制。

## 第四章 MIPS 系统中卡拉 OK 混响实现

众所周知,在同一个房间不同的声源产生的听觉效果不同;不同设计材质的房间内同一个声源所产生的听觉效果也是不一样的。主要是因为室内声源停止发声后,由于房间边界面或其中障碍物使声波多次反射或散射而产生声音延续的现象。声波在室内传播时,要被墙壁、天花板、地板等障碍物反射,每反射一次都要被障碍物吸收一些<sup>[20]</sup>。这样,当声源停止发声后,声波在室内要经过多次反射和吸收,最后才消失,我们就感觉到声源停止发声后声音还继续一段时间,这种现象叫做混响,这段时间叫做混响时间。

混响时间的长短是音乐厅、剧院、礼堂等建筑物的重要声学特性,在通过墙壁反射时声音被墙壁吸收强度跟墙壁的材质和声音的频率都是有关系的。混响是室内声音的一种自然现象,反射声使原声具有延续性,听起来圆润动听,具有明显的环境特性。对于卡拉 OK 系统,可以基于室内声学模型对声音进行混响处理,增加人工反射声来使声音丰满动听,让歌者在歌唱时不会感到费力。卡拉 OK 系统中的混响实现是该系统非常重要的环节<sup>[21]</sup>。

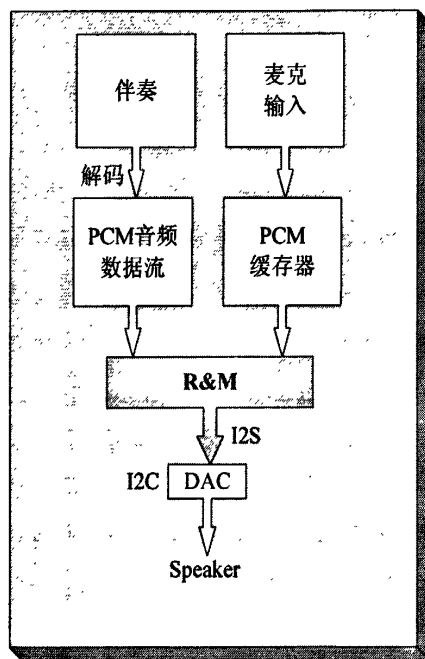


图 4-1 卡拉 OK 混响实现系统框图

## 4.1 系统平台设计

MIPS 卡拉 OK 系统中，主要有两路音频输入：一路是伴奏音，一路是麦克输入。如图 4-1 所示，系统的核心部分是 R&M（混响和混音实现）模块，该模块主要完成两个任务，其一是对麦克输入数据流进行混响处理，也就是反射声的模拟，产生立体声效果，其二是将混响后的声音与伴奏进行混音，作为卡拉 OK 的最终输出。下面主要讲述 R&M 模块的功能实现。

## 4.2 理论研究

室内的立体声是声源本身产生的直达音和通过墙壁的反射声共同产生的<sup>[2]</sup>，如图 4-2 所示<sup>[22]</sup>，当室内声音产生后，听者首先听到从声源直接传过来的声音（图 4-2 中的 1 号），稍后听者将听到墙壁反射回来的声音（2、3、4 号），各种反射声与直达声共同作用形成最终的混响声音。

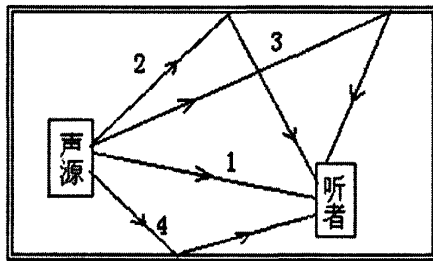


图 4-2 室内立体声的形成模型<sup>[28]</sup>

进一步理解混响的概念，如图 4-3 所示室内声脉冲的特性曲线，发现声音的声压呈指数衰减，可以用数学公式表示混响时间：

$$RT = 0.161 \times (V/A) \quad \text{式 (4-1)}$$

其中  $V$  表示室内的体积（立方米）， $A$  表示假想的室内“吸收窗”（平方米）， $A$  是通过室内总的表面相应吸收系数计算出的。不同材质的墙面对于不同频率的声源其吸收系数是不同的<sup>[22]</sup>，在此就不做列举。

在了解混响的产生原理后，下面将详细介绍混响滤波器如何设计才能产生混响效果。由室内立体声的产生原理可以知道混响主要是由各路反射声叠加而成的，为了模拟这些反射声，设计梳状滤波器来实现，具体实现过程下面会做具体的介绍。在实现反射声的模拟后要与原声进行混合，此实现是通过全通

滤波器来实现的。所以综上所述混响滤波器是用梳状滤波器和全通滤波器相结合来实现。

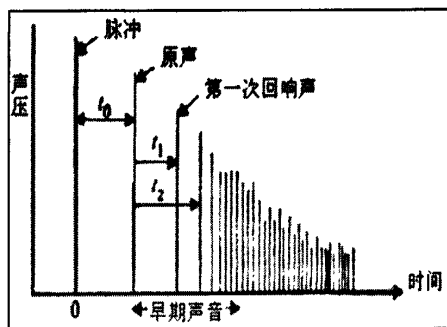


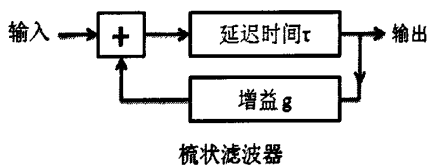
图 4-3 室内立体声脉冲特性曲线

梳状滤波器用来模拟各路回音的增益和延迟时间，在图 4-4 的频谱响应中体现了梳状滤波器的最小混响值与最大混响值<sup>[22]</sup>：

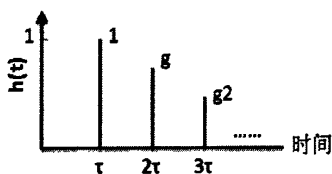
$$|H(f)|_{\min} = \frac{1}{1+g} \tag{式 (4-2)}$$

$$|H(f)|_{\max} = \frac{1}{1-g} \tag{式 (4-3)}$$

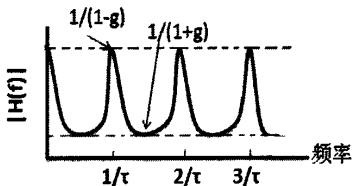
其中循环增益  $g$  一般不超过 0.85<sup>[22]</sup>。



梳状滤波器



脉冲响应



频率响应

图 4-4 梳状滤波器及其脉冲响应和频率响应

梳状滤波器所需要的延迟时间:

$$T_m = \frac{60}{-20 \log |g_m|} \quad \tau_m = \frac{3}{-\log |g_m|} \tau_m \quad \text{式 (4-4)}$$

可以从公式(1)中计算出来, 从而设计出模拟室内声学系统的梳状滤波器。

为了获得合适的相位延迟, 梳状滤波器的输出需通过全通滤波器, 进一步增强混响效果, 全通滤波器的实现如图 4-5。

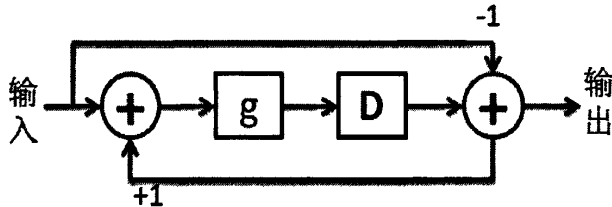


图 4-5 全通滤波器

为实现全通滤波器, 增益  $g$  必须等于  $1/\sqrt{2}$  确保全通滤波器的稳定性<sup>[22]</sup>, 为梳状滤波器传来的数据提供正确的相位延迟。

这样就完成了用于室内建模的混响滤波器的简单设计。

### 4.3 具体实现

根据上述理论, 为了更好的产生立体声的效果, 卡拉 OK 混响实现选用四级梳状滤波器和两级全通滤波器来组成混响滤波器。如图 4-6 所示。

混响滤波器中设计四级梳状滤波器来模拟不同路的反射声, 四级梳状滤波器的增益和延迟时间是不同的。回声越小, 对应的延迟时间越长, 延迟时间可以由相应的增益获得, 其值为  $(-\log_{10}(\text{增益值})/3.0)$ 。根据所需的混响效果梳状滤波器的增益和延迟时间不同。全通滤波器的输入为四级梳状滤波器的平均值。全通滤波器的增益如上述应为  $1/\sqrt{2}$  为混响提供正确的相位延迟。

通过四级梳状滤波器和两级全通滤波器后, 就完成了反射声的模拟, 将其按一定的混响深度与原声叠加就得到了最终的混响效果, 在此可以调节混响深度来控制混响幅度。

卡拉 OK 系统完成了麦克输入的混响处理后, 还需将其与伴奏进行混音处理, 混音时也可调节混响声所占比例。这样就最终完成了卡拉 OK 系统的混响

实现。在此系统中不但可以调节相关参数获得所需混响效果，还可以调节伴奏和麦克声音的比例。

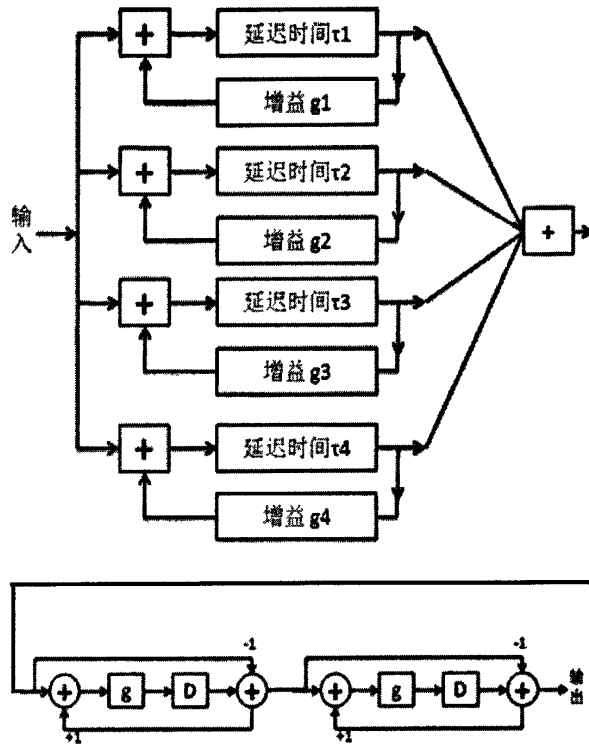


图 4-6 IIR 滤波器的设计框图

本设计中，为用户开放了三个参数来调节混响的延迟时间和混响强度：梳状滤波器的延迟时间参数，声音的采样率以及混响深度参数。用户通过调节这三个参数来得到所需的混响效果。这三个参数有边界控制可保证功能的稳定性。

在卡拉 OK 系统中梳状滤波器的延迟时间参数设置可产生 0 到 4 秒的延迟效果；声音的采样率为 8KHz 到 48KHz；混响深度参数可产生 0 到 1 的混响深度。

在卡拉 OK 系统中混响时间合理才能使声音圆润动听，混响时间过短，声音会发干，枯燥无味，混响时间过长，声音会含混不清，一般调节在 1 至 2.5 秒之间比较合适。合理调节该参数可以让歌唱者在歌唱时感觉声音圆润丰满。卡拉 OK 系统中不仅可以调节混响深度来控制混响大小，也可以调节声音与伴奏的比例来控制声音的相对大小。

在该程序的设计中，其硬件系统不支持浮点操作的，为了完善其兼容性，

需同时支持浮点和定点运算, 移植的过程中将浮点转为定点处理, 在保证混响质量的前提下运算效率得到了很大的提高。

在下面的测试分析中会详细介绍这混响延迟时间和混响深度的具体影响。

#### 4.4 测试分析

如上述所说混响时间过短, 声音会发干, 混响时间过长, 则会使声音含混不清。混响深度越大, 混响中回声的幅值越大, 回声效果越明显<sup>[23]</sup>。系统中如何设置调节接口, 以及这些接口对混响效果的具体影响下面将做具体的测试分析。

在卡拉 OK 系统中, 混响效果与混响滤波器中的延迟时间和增益密切相关, 在此针对上述的三个参数进行测试, 说明其具体影响。在程序的设计中梳状滤波器的延迟时间和声音采样率对混响延迟时间都有影响。

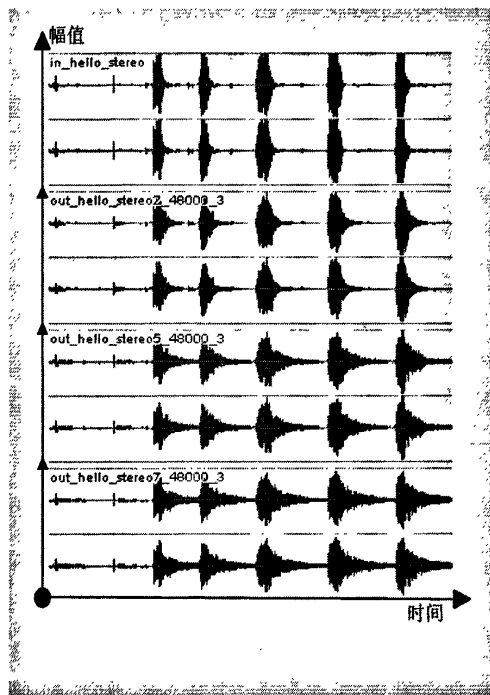


图 4-7 梳状滤波器延迟时间参数的影响

梳状滤波器的延迟时间参数不同, 引起的变化如图 4-7 所示。这四组的混响采样率为 48KHz, 混响深度为 0.3。第一组是未经处理的原声; 第二组到第四组延迟时间分别为 0.8s、2s、2.8s。从图中可以看到延迟时间不同引起的混响效果不同, 当延迟时间越大, 回声维持的时间越长。所以希望回音效果越持



久，该参数的取值越大。

声音采样率对混响效果的影响，如图 4-8 所示：这四组的混响延迟时间为 2 秒，混响深度为 0.3。第一组是未经处理的原声，第二组到第四组是声音采样率分别为 16KHz、32 KHz、48 KHz 所引起的不同混响效果，从图中可以看到，当混响采样率越大，回声时间越久。

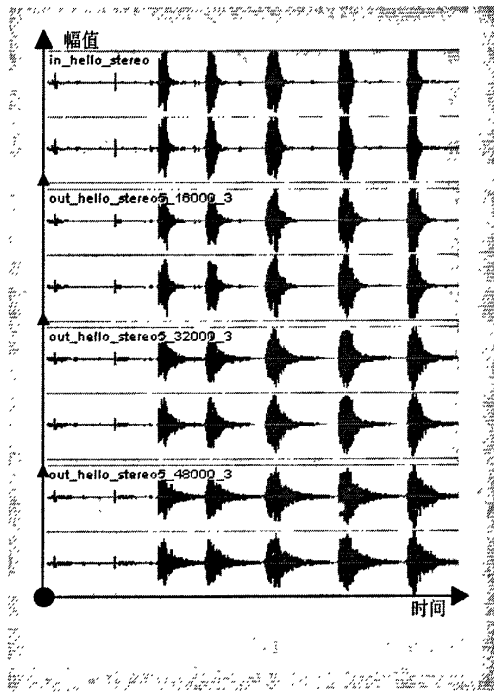


图 4-8 梳状滤波器回响采样率的影响

在上述说明中可以看到梳状滤波器的延迟时间和声音采样率变大都可以使回声维持时间持久。混响时间短，声音的清晰度较高；混响时间长，声音的丰满度好。

最后介绍混响深度对混响效果的影响，如图 4-9 所示。这四组的混响延迟时间为 2 秒，混响采样率为 48KHz。第一组是未经处理的原声，第二组到第四组的混响深度分别为 0.2、0.5、0.7，从图中可以看到混响深度引起的混响效果不同，当混响深度越大，混响中回声的幅值越大。

根据以上对三个参数的具体实验分析，用户可以通过这三个参数来调节混响的延迟时间和幅值大小，得到所需的音效。

卡拉 OK 系统在完成混响后，混响声要与伴奏音进行混音处理，在此会设置一参数来调节混响声和伴奏音的比例。

分析各个参数对混响效果的具体影响是有实际的意义的，因为不同用途的厅堂，最佳混响时间是不相同的。一般来说，音乐厅和剧场的最佳混响时间比

讲演厅要长些，而且因情况不同而不同，轻音乐要求节奏鲜明，混响时间要短些，交响乐的混响时间可以长些。难于听懂的剧种混响时间一长，就更难于听懂了。节奏较慢而偏于抒情的剧种，混响时间则可以长些。总之，要有一定的、恰当的混响时间，才能把演奏和演唱的感情色彩表现出来，收到应有的艺术效果。

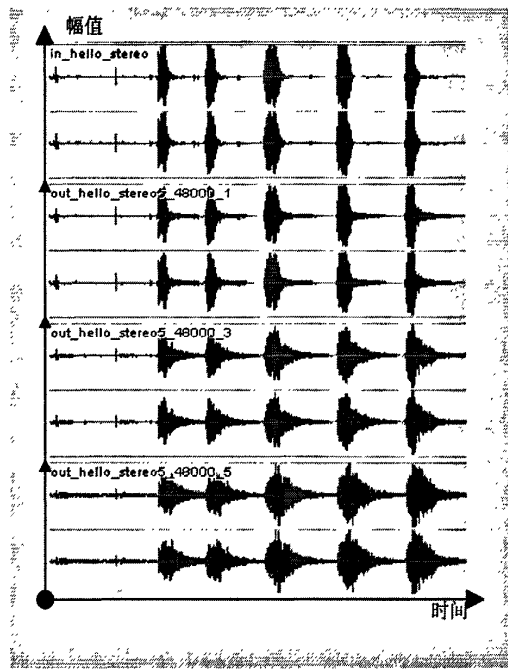


图 4-9 全通滤波器回响深度的影响

如果要求混响时间短一些，语音的清晰度要高，反射声的强度设置要小，混响比例设置小从而保证需求。

## 4.5 本章小结

本章是基于室内声学模型，对卡拉 OK 系统中的声音进行混响处理。详细讲述了卡拉 OK 系统中混响的具体实现，混响处理器是四级梳状滤波器和两级全通滤波器组成的。梳状滤波器用来模拟各路反射声的，来完成混响效果的实现，而全通滤波器则是完成原声与反射声的混音处理。在程序设计中定义三个接口来调节混响效果，调节各级梳状滤波器的延迟时间、增益及混响深度可得到所需的混响效果。详细分析了梳状滤波器的延迟时间参数，声音的采样率以及混响深度参数对混响效果的影响。卡拉 OK 系统中对人声的混响处理使声音

圆润动听，对声音和伴奏的混音处理使歌唱和伴奏很好的结合起来，作为卡拉 OK 系统的最终输出，实现卡拉 OK 混响处理。混响处理不仅是卡拉 OK 系统中的重要环节，还广泛应用在体育馆、音乐厅、歌剧院中，根据不同的需求混响参数设置不同，在制作不同的音效上混响也起了很重要的作用。

## 第五章 工作总结与展望

### 5.1 工作总结

随着宽带数字音频编码技术的发展,对音视频信号处理的要求也越来越高,在保证音视频质量的情况下,要求对其操作存储的空间和传输所需的带宽越来越少,针对这一情况的解决方案数字压缩技术的要求也越来越高。而具有更强定位能力和空间效果的三维声音技术也得到了蓬勃发展,在三维声音技术中最具有代表性的就是多声道环绕声技术。其中一种音频技术为 AC3,杜比 AC3 提供了 5 个全频域声道和 1 个超低音声道,支持 32kbps 到 640kbps 的传输率,压缩比约为 10:1,能够达到低码率传输高音质的目的。

AC3 技术在多声道音频编码领域的发展稍占优势,有着广泛的市场应用前景,对其进行研究有着重要的现实意义。但在不久的将来,建立一种综合多种技术优势的更有效的编码标准将成为一种趋势。

本论文主要做了以下四个方面的工作:

- 1, 通过调研大量的文献,分析了当前数字音频技术的发展现状及今后的发展趋势,多声道环绕声技术应用的必然性。介绍了目前发展比较成熟的几种音频压缩编码标准,重点讲述了杜比 AC3 技术的应用和发展,研究了音频压缩技术和音频压缩的原理:人耳听觉的心理声学特性。

- 2, 介绍了系统选用的 MIPS32 24KE 内核特点。详细讲述了两个 MIPS 芯片的通信机制:消息队列和邮件机制,MIPS1 的工作动力是源源不断的消息来源,当 MIPS1 接收到 MIPS0 所给的消息后,进行解析,然后进入相应的模块处理,完成处理后会反馈给 MIPS0 确认信息,两个芯片是同步工作的。然后详细阐述了 MIPS1 系统的整体设计和处理音频数据的工作流程。MIPS1 的处理机制相当于一个大的循环过程,各个模块的功能是相互独立的,通过模块间传递消息通信,从而协调工作,模块的设计理念是高内聚低耦合。重点讲述了 MIPS1 对于音频解码模块的设计和具体处理流程,分析了音频处理的各个模块的功能和接口设计。

- 3, 深入分析了 AC3 的帧格式,完成了 AC3 软解码过程的实现,测试和优化。在论文中重点阐述了指纹解码、尾数的量化解码和声道 DOWNMIX 操作的具体实现。在实现解码过程后,运用杜比向量组对解码模块进行针对性测试和解 BUG,保证解码过程正确性。在保证解码程序正确的前提下对 AC3 解码

过程进行语言级优化,包括 DMA 操作、汇编语言的优化、浮点定点化, DMA 的搬移来代替 CPU 点对点的操作,在音频数据输入输出的时候有大量的数据要进行搬移,这样大大节省了时间数据搬移时间, C 语言解码过程中有些复杂的运算存在比如 IFFT 实现,会有大量的运算,流水线很不好,采用汇编编码来代替提高程序并行度,使流水线充分起来,大大提高了运算速率,另外在程序中避免浮点运算,将浮点转化成定点运算并保证其精度。在音频播放时中实现了总线协议 I2C 和 I2S 的封装。

4, 基于室内声学模型,实现了 MIPS 系统中的卡拉 OK 声音的混响操作,并对影响混响效果的参数做了分析说明。

卡拉 OK 系统中混响是通过混响处理器来实现的,混响处理器由四级梳状滤波器和两级全通滤波器组成的。梳状滤波器用来模拟各路反射声的,来完成混响效果的实现,而全通滤波器则是完成原声与反射声的混音处理。在程序设计中定义三个接口来调节混响效果,调节各级梳状滤波器的延迟时间、增益及混响深度可得到所需的混响效果。详细分析了梳状滤波器的延迟时间参数,声音的采样率以及混响深度参数对混响效果的具体影响。卡拉 OK 系统中对人声的混响处理使声音圆润动听,对声音和伴奏的混音处理使歌唱和伴奏很好的结合起来,作为卡拉 OK 系统的最终输出,实现卡拉 OK 混响处理。混响处理不仅是卡拉 OK 系统中的重要环节,还广泛应用在体育馆、音乐厅、歌剧院中,根据不同的需求混响参数设置不同,在制作不同的音效上混响也起了很重要的作用。

## 5.2 工作展望

本课题实现了 AC3 的软解码,对解码过程的正确性进行了验证,并对其进行了汇编级优化工作,提高程序的并行性,加快解码速率。最后将解码过程移植到 MIPS 上,完成 MIPS 芯片上的音频正确解码。该芯片已成功应用到高清数字电视中作为音视频处理器。

但是整个 AC3 解码过程的处理主要是软件实现的,为了使 AC3 解码过程更加有效率,可以与硬件设计相结合对解码过程进行处理,可以进一步改进比特分配、指数解码、尾数量化与解码以及其他模块的算法实现,提高这些模块的整体性能和可靠性。将运算复杂的模块合理的分配到硬件上进行处理,减轻软件处理的负担<sup>[24][25]</sup>,可以将数据量大并且运算复杂的处理模块放在 DSP 上实现。

随着数字音视频高新技术的发展, DVD、高清数字电视迅速发展<sup>[26][27][28]</sup>。作为DVD的音频部分, 杜比AC3解码器及5.1声道功放机有着巨大的市场需求, 而目前其核心技术和解码芯片均依赖于美国等少数发达国家。因此, 如果我们能够及时研究开发具有自主知识产权的产品, 不仅能大大提高我国多媒体信息处理的能力和相关专业芯片的质量, 促进DVD及高清数字电视产业的形成, 而且能带动我国集成电路等一大批相关电子信息产业的发展, 其应用前景十分广大。因此, 在这一领域中进行进一步的深入研究是十分有必要的。

## 参考文献

- [1]佚名,“数字音频压缩技术的发展”,中华视频网,2006.6.24
- [2]徐盛,陈健,“数字音频编码技术的回顾与发展”,电声技术,1999(8):3-11
- [3]钱征海,吕宗伟,“数字音频技术与杜比 AC3”,电子技术应用,2005(9):49-51
- [4]江虹,韦岗,“心理声学在宽带音频压缩编码中的应用”,广东通信技术,2002,21(2):30-33
- [5]“Digital Audio Compression standard(AC3) Revision B Document A/52B”,  
[www.atsc.org/standards/a\\_52b.pdf](http://www.atsc.org/standards/a_52b.pdf),June.14.2005
- [6]韩宪柱,“数字音频技术及应用”,北京:中国广播电视出版社,2003
- [7]朱丽,郭从良,“心理声学模型在数字音频中的应用”电声技术,2002(8):11-14
- [8]E.Zwicker, H.Fastl, “Psychoacoustics Facts and Models”, Springer-Verlag,1990
- [9]房建,左涛,“数字音频压缩编码技术及其应用”,信息技术,2004(2):9-11
- [10]蔡书华,“ac3 系统功能及原理浅析”,广西民族学院学报(自然科学版),2002,8(2):42-45
- [11]潘兴德,朱晓明,“EAC 音频技术编码”,电声技术,2003(2):11-15
- [12]徐孟侠译,乐陶校,“E-AC3 比特流技术规范(草案)中包含的特征简介”,电视技术,2004(8),82-84
- [13]Vladimir Nikolajevic, Gerhard Fettweis, “New Recursive Algorithms for the Forward and Inverse MDCT”, IEEE Signal Processing Systems, 2001, 26(28):51-57
- [14] Vladimir Nikolajevic, Gerhard Fettweis, “Computation of Forward and Inverse MDCT Using Clenshaw’s Recurrence Formula”, IEEE Transaction on Signal Processing, 2003,51(5):1439-1444
- [15]Hwang Cheng Chiang, Jie Cheng Liu, “Regressive Implementations for the Forward and Inverse MDCT in MPEG Audio Coding”, IEEE Signal Processing Letters, 1996,3(4):116-118
- [16]余永聪,尹俊勋,“音频压缩编码中的参数比特分配技术”,电声技术,2002,(7):6-9
- [17]II-Taek Lim, Bahn J, “Fixed-point analysis and simulations of AC3 algorithm”, Applications of Signal Processing to Audio and Acoustics, 1997.1997 IE EE ASSP Workshop on 19-22 Oct.1997: 4

- [18]Seokjun Lee and Wonyong Sung, "A Parser Processor for MPEG-2 Audio and AC3 Decoding", IEEE International Symposium on Circuits and System. HongKong. June 9-12,1997
- [19]Bergher, J. Boehm, X. Figariet, "Dolby AC3 and MPEG2 Audio Decoder IC With 6 Channels Output", IEEE Transactions on Consumer Electronics, Vol 43, No3, 1997.
- [20] 混响室及其进展[J],张林昌,电子质量,2003, (1):39-423.
- [21] 立体声数字卡拉OK回响系统, 钱恩聪, 音响技术 1998--1.
- [22] Room Acoustics Modeling .1996.Phipil Edelbrock
- [23] 混响室测量技术[J], 逯贵祯, 安全与电磁兼容, 2004, (3):22-24
- [24]韦晓东, 周琼芳等, "嵌入式RISC核MPEG2/AC3解码器", 半导体技术, (28)7: 47-55
- [25]刘鹏. 李东晓. 姚庆栋等, "面向HDTV解码应用的RISC核的软硬件协同设计[A]", 中国电子学会电路与系统分会第十六届年会[C], 中国宁波, 2001
- [26]国狄非, 国澄明, 张涛等, "数字HDTV信源解码器的研究与实现", 天津大学学报, 35(6): 787-791
- [27]刘鹏, 李东晓, 姚庆栋等, "面向 HDTV 解码应用的 RISC 核的软硬件协同设计", 中国电子学会电路与系统分会第十六届年会, 中国宁波, 2001
- [28]Ando Y. Concert Hall Acoustics. New York: Springer-Velag, 1985.
- [29]王立, 李桂等, "MPEG和Dolby AC3音频码流分析软件的设计与实现", 电声技术, 2004(4): 37-40
- [30]蔡书华, "ac3 系统功能及原理浅析", 广西民族学院学报(自然科学版), 2002, 8(2): 42-45
- [31]C.R.Cave, "Perceptual Modeling for Low-rate Audio Coding", Master, McGill University2002
- [32]Stephen Li, Jon Rowlands, Pius Ng, Maria Gillet, "An AC3/MPEG Multi-standard Audio Decoder IC", IEEE Custom Intergrated Circuits Conference, 1997
- [33]Chi-Min Liu, Wen-Chieh Lee, Shyh-Yan Juang, "Design of the coupling schemes for the AC3 coder in stereo coding", IEEE Transactions on Consumer Electronics, 1998, Vol 44, No3: 878-882
- [34]韦健, 张明, 刷琼芳等, "超标量、超流水线定点RISC核的设计[J]", 电路与系统学报, 2001, 6(4): 56-60



## 致 谢

首先，感谢我的导师全子一老师、门爱东老师和杨波老师。全老师虽然年事已高，但还是对实验项目投入了巨大的热情和关注。门老师以他渊博的知识和丰富的经验，给予我方向性的指导。而杨老师则事无巨细，耐心的指引我解决项目中的问题，使我少走了很多弯路。

感谢况超，谢林师兄，他们比我早加入视音频编解码组，在工作上给了我很大的指导和帮助，在软件开发遇到问题的时候，他们会给我醍醐灌顶的建议，在此深表谢意。感谢我的同事，虽然我们处于不同的组，责任不同，但是在联调程序的时候，大家一起讨论问题共同面对困难，一起解决问题，使得工程可以顺利进行，在此也非常感谢他们。

感谢实验室的其他师兄、师姐和同学，特别是同级同学，在生活和学习上给我营造了完美的氛围，使我快乐地度过了研究生阶段。

最后，衷心感谢我的父母和亲友对我无条件的支持。

## 作者攻读学位期间发表的学术论文

- [1] 卫飞宇, 杨波; 基于室内声学模型的卡拉 OK 混响实现; 核心期刊: 电声技术, 2010.1

