多制式音频解码关键模块的 FPGA 设计与验证

周 游,贺 珊,李 琳,郭东辉
(厦门大学信息科学与技术学院,福建厦门 360015)

摘 要:在分析音频解码标准 MP3 和 AAC 及其实现方案基础上,提出了基于 FPGA 的支持多制式的音频解码算法设计方案。将分步查表法引入霍夫曼解码,并提出了无乘法器的反量化变换及可兼容的 IMDCT 变换算法的 FPGA 实现方案。板上测试验证表明,该方案可准确完成多制式音频的解码,并提高解码速度与精度。

关键词:音频解码;FPGA 设计;霍夫曼解码;反量化变换;IMDCT 变换
中图分类号:TN-911 **文献标志码**:A **文章编号**:2095-2783(2014)07-0798-05

Design and verification of FPGA-based key modules in multi-standard audio decoding

ZhouYou, He Shan, Li Lin, Guo Donghui

(*Department of Information Science and Technology, Xiamen University, Xiamen, Fujian* 361005, *China*)

Abstract: The audio encoding standards (MP3 and AAC) and their implementation schemes are analyzed. The FPGA design scheme is proposed to support the audio decoding algorithm with multi-standards. The substep lookup table algorithms introduced to implement Huffman decoding on hardware. A FPGA-based scheme, supporting multiplier-less inverse quantization and configurable IMDCT, is proposed. The experiments on demo board show that proposed scheme finish the decoding tasks with higher speed and accuracy.

Key words: audio decoding; FPGA design; Huffman decoding; inverse quantization; IMDCT

MP3(moving picture experts group audio layer III)由于其出色的压缩性能,在音乐存储与传播等领域获得了广泛的应用[1]。在其被推出后,其他音视频标准也纷纷问世,其中尤其值得注意的是由杜比实验室、AT&T、Sony 等公司共同开发的旨在取代 MP3 格式的 AAC(advanced audio coding)[2]。目前,AAC 中的 LC 框架已经获得了苹果公司 iTunes Store 等在线数字媒体商店的大力支持,很有希望成为下一代的主流音频标准。

以 MP3 与 AAC 为代表的压缩与编解码标准与消费类电子产业的发展互相推动,形成了规模巨大的新兴产业。针对该趋势,各类基于音频编解码技术的解决方案也随之被提出[3]。目前较为主流的解决方案可大致分为两类。

1)传统的基于通用处理器或 DSP(digital signal processor)进行音频编解码的软件实现方案。该类方案虽然曾被广泛应用,但其通用指令集并非为音视频编解码而专门设置,且处理器字长一般无法很好地适应音视频处理的实际需求[4],很难做到在高执行效率和低功耗的要求下进一步完成音视频的编解码任务[5]。随着各类移动终端产品功能的多样化和用户要求的进一步提高,可以预计,该类方案的固有缺点也必将日益凸显。

2)针对指定的音视频编解码算法设计的专用解码器或 ASIC(application specific integrated circuit)方案[6]。该类方案执行效率一般相对较高,功耗较小[7],并且可以充分发挥硬件编解码并行执行的优势。但是,该类方案的缺点也同样明显,固化的硬件结构很难根据实际产品的需要作出灵活的调整和改变,随着音视频编解码标准的不断发展,该类方案亦很可能会遭遇技术瓶颈。

针对以上方案可能存在的问题,本文在充分研究音频解码技术的基础上,设计并实现了支持多制式的解码关键模块,既可在多标准软硬件协同编解码中作为处理关键运算的硬件加速方案,从而减少指令消耗与解码时间;亦可根据需要,灵活地扩展为专用解码器与 ASIC。

1 多制式音频解码关键模块选取

典型的 MP3 与 AAC LC 解码流程如图 1 所示。

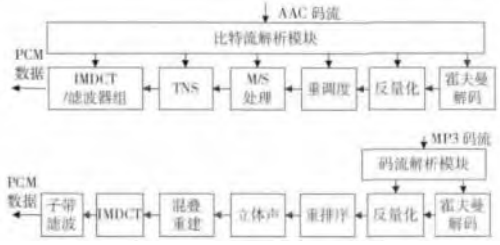

图 1 典型的 MP3 与 AAC LC 解码流程

收稿日期: 2014-03-25
基金项目: 国家自然科学基金资助项目(61274133);高等学校博士学科点专项科研基金资助项目(20090121110019)
作者简介: 周游(1988—),男,硕士研究生,主要研究方向为音视频硬件解码设计
通信联系人:郭东辉,教授,主要研究方向为集成电路设计,dhguo@xmu.edu.cn

以计算量（计算所占用的周期数）为单位，MP3 与 AAC 的各个解码模块以不同方案解码一帧数据过程中所占的比重如图 2 所示。

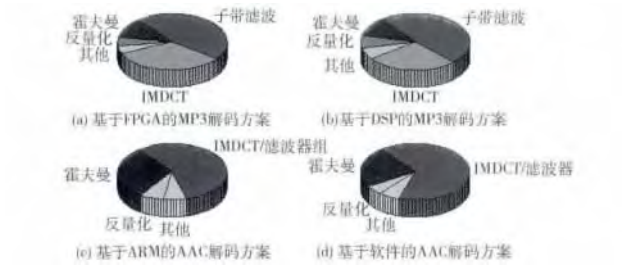

图 2　以不同方案解码一帧数据时各模块的计算量分布

由图 2 可知，一方面，在 MP3 与 AAC LC 的解码过程中，霍夫曼解码和 IMDCT 是解码中运算量较大、较为重要的关键步骤；另一反面，反量化变换所占运算量并不是很大。实际上，反量化变换的重要性并不体现在运算量上，尤其是在基于硬件实现的方案中，反量化变换必须借助近似方法完成，而具体方案的选取将在很大程度上影响解码的信噪比与最终结果。通过以上分析，本文选取霍夫曼解码、反量化变换和 IMDCT 变换作为解码中关键模块的依据。

2　关键模块算法设计

2.1　基于分布查表的霍夫曼解码算法设计

MP3 与 AAC 标准在霍夫曼解码中的对比见表 1。

表 1　MP3 与 AAC 霍夫曼解码对比

标准区别	比例因子	码字组成	码表个数
MP3	定长编码	低频 2 个系数一组 中频 4 个系数一组	16
AAC	霍夫曼编码	低频 4 个系数一组 中频 2 个系数一组	12

目前，霍夫曼解码中所采用的方法可大致分为直接查表法、二叉树查表法和分步查表法。其中，直接查表法将码表全部放入存储空间中，会导致码表空间的浪费。MP3 与 AAC 音频解码标准中的霍夫曼查找表都可以转化成二叉树的形式，以 MP3 编号为 3 的码表为例，将其改编为二叉树码表后，数据依次为 4、1、2、1、0、0、0、1、2、1、0、11、2、1、0、10、4、1、2、1、0、20、0、21、2、1、0、12、2、1、0、2、0、22。开始时，每两个数是一个节点，每个节点的两个数代表左右分支节点的数据向后的偏移量，当一个节点的两个数中左边的数为 0 时，表示其是一个终点。较之直接查表法，二叉树查表法在一定程度上可压缩码表所占用的空间，但每次仅读入一个比特进行跳转，对较长码字的查表速度改进有限。

分步查表法既不同于有接查表法每次读入最长码的码长个比特，也不同于二叉树查表法每次读入一个比特，而是可以根据具体应用的限制，灵活地将查表分为几步完成。3 步查表法如图 3 所示。

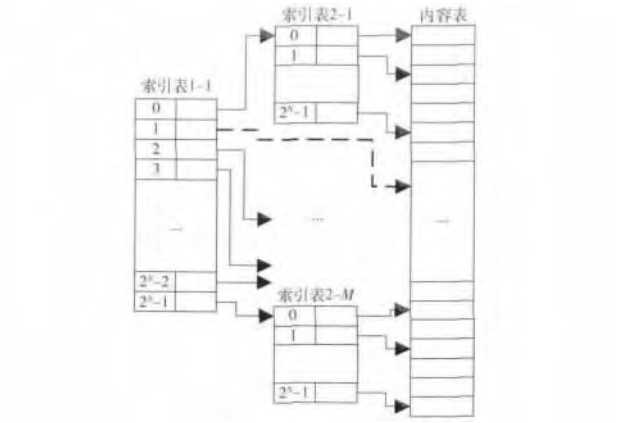

图 3　3 步查表法

分步查表法通过建立只存储地址值的索引表避免对码字的存储，只在最后的一张内容表中记录频率线值，在查表速度与码表空间占用两方面可取得较好的折衷。

针对查表的分步问题，本文提出：无论每步读入的码长的具体个数为多少，都应该可使霍夫曼解码在首次读入码元并完成查表后的概率为最大。

AAC 与 MP3 霍夫曼的码字长度统计如图 4 所示。由图可知，出现概率最高的码字为 4 个比特（MP3）与 5 个比特（AAC）。考虑到 AAC 中码长为 4 的码字出现的概率同样很高（约 20%），如果每次读入 4 个码元，那么相对于选择其他步长，将会有更高的概率一次性完成解码。

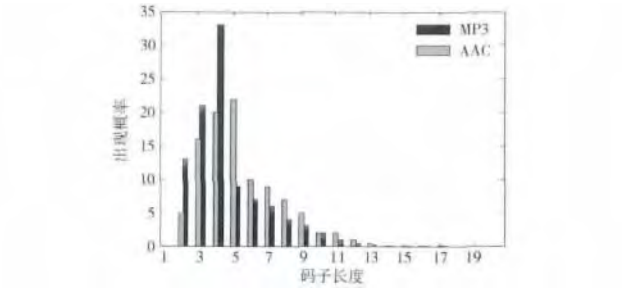

图 4　AAC 与 MP3 霍夫曼码字长度统计

本文借鉴二叉树查表法的节点概念，提出了一种基于“索引树-枝节点-叶节点”的码表组织形式，如图 5 所示。图中：圆形框代表枝节点，对应于软件中的二级索引表；方形框代表叶节点。枝节点中存储的是下一步查表时的起始地址值增量，表示该步解码需进行地址累加，然后跳转到下一步进行（对应软件中索引表的跳转）；叶节点中则存储的是频率线值，到达叶节点即代表着该次解码的结束。

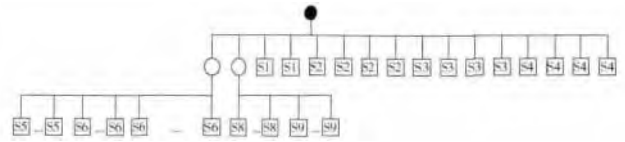

图 5　霍夫曼分步查表法的索引树结构

2.2 无乘法器反量化变换的算法设计

MP3与AAC的反量化变换核心公式为

$$p_i = \mathrm{sign}(q_i) \times |q_i|^{\frac{4}{3}}。 \quad (1)$$

式中，q 为霍夫曼解码结果，取值范围在0～8 206（AAC为0～8 192），q_i 为 q 的反量化结果。由式(1)可见，MP3与AAC的反量化变换最为关键的是非整数次幂的计算。

目前，反量化变换中所采用的方法可以分为除8查表法、除64查表法、线性内插查表法和多项式拟合法。

基本查表法有所占用码表空间大的问题。除8查表法的公式为

$$q^{\frac{4}{3}} = \left(8 \times \frac{q}{8}\right)^{\frac{4}{3}} = 16 \times \left(\frac{q}{8}\right)^{\frac{4}{3}} \cong 16 \times \left(\left[\frac{q}{8}\right]\right)^{\frac{4}{3}} = 16 \times k_i^{\frac{4}{3}}。 \quad (2)$$

式中，k_i 的取值范围为0～1 025。借助于 k_i，在引入系统误差的基础上，除8查表法可将码表所需存储的数据缩减至基本查表法的1/8。

类似地，除64查表法也可实现与除8查表法同样的结果。

线性内插查表法的主要公式为

$$q^{\frac{4}{3}} = q \times \left(8 \times \frac{q}{8}\right)^{\frac{1}{3}} = q \times 2 \times \left(\frac{q}{8}\right)^{\frac{1}{3}} \cong q \times 2 \times \left(\left[\frac{q}{8}\right]\right)^{\frac{1}{3}} = q \times 2 \times k_i^{\frac{1}{3}}。 \quad (3)$$

式中，k_i 的取值范围与式(2)相同。在线性内插查表法中，码表空间中所存储的是 q 的1/3次幂值，需要在设计中引入乘法器，将查表获得的结果进行乘以 q 的运算。

多项式拟合法与之前介绍的几种方法的思路不同，其使用整数的多阶多项式来拟合近似 $q^{\frac{4}{3}}$ 的曲线，主要公式为

$$q^{\frac{4}{3}} = a_0 + a_1 \times q + a_2 \times q^2 + a_3 \times q^3 + \cdots \quad (4)$$

在该方法中，所有幂运算都是整数次幂的运算，只需要存储系数值即可。但仅用一组系数来拟合整条曲线是不切实际的，因此可将目标曲线划分为多个区间进行拟合（Law's拟合）。各种方法的对比见表2。

表2 常见反量化变换方法对比

算法	运算量	信噪比/dB	码表空间
除8查表法	1F	57.6	1 026
除64查表法	1F	42.1	129
线性内插法	1F 1M	68.7	1 026
多项式拟合法	6M 3A	65.9	61

注：F代表查表操作；M代表乘法操作；A代表加法操作

由表2可知，各种方法都很难在计算量、精度和码表空间中取得较好折衷。本文首先分析了除8查表法的误差趋势，其误差趋势如图6所示。

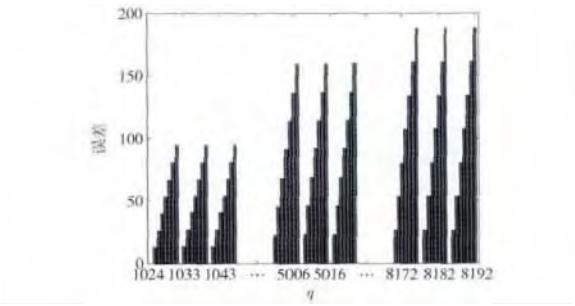

图6 除8查表法的误差趋势

由图6可知，除8查表法的绝对误差总体呈现增长趋势，而在总体范围内，除8后所得余数呈现阶梯状分布。某一区间内，除8后余数相同的 q 值与标准值的相对差值较小，因此，可根据该误差特性设置补偿因子，将补偿因子与区间内除8后余数相同的 q 的查表结果相加，在不引入乘法器的前提下，通过码表空间的略微增加换取误差更小的结果，以达到更好的信噪比。

为进一步减少比例因子占用的码表空间，本文提出的补偿因子法通过控制相对误差值保持恒定（现有方案最小相对误差为0.02%，本文在对精度和码表空间进行折衷后，设定为0.008%），尽可能地使更多除8后余数相同的 q 复用同一补偿因子。相对误差的计算公式为

$$u = \frac{\left| q^{\frac{4}{3}} - \left(\left[\frac{q}{8}\right]^{\frac{4}{3}} + v\right)\right|}{q^{\frac{4}{3}}} \times 100\%。 \quad (5)$$

式中，u 为相对误差，v 为对应于 q 的补偿因子。以除8后余数为7的 q 值为例，首先令 q 与相对误差相乘，得出绝对误差值，并以该值的2倍为区间长度，确定区间范围；随后将该区间内的所有 q 值代入方差公式进行计算，得出满足总方差为最小的 v，并存储进码表空间。

2.3 兼容性IMDCT变换的算法设计

改进的离散余弦逆变换（IMDCT）是一种与傅里叶变换相关的变换，以第四型离散余弦变换（DCT-IV）为基础。该变化适用于处理较大的数据集合，并具有离散余弦变换的能量压缩特性，被众多音频标准广泛采用。其主要公式为

$$x_i = C\sum_{k=0}^{\frac{N}{2}-1} X_k \cos\left(\frac{\pi}{N}\left(i + \frac{1}{2} + \frac{N}{4}\right)(2k+1)\right)。 \quad (6)$$

式中，$i = 0, 1, \cdots, N-1$，X_k 为混叠重建后（或TNS处理后）的数据，x_i 为IMDCT变换后的输出数据。N 在不同的标准中的取值不同，在MP3标准中，分别为12或36；在AAC标准中，分别为256或2 048；在WMA标准中，N 的取值范围为[64，8 192]。

由式(6)与 N 的取值范围可知，在FPGA中，若直接按照式(6)计算IMDCT，需进行 $N \times N/2$ 次的乘法运算和相同次数的加法运算。因此必须采取优

化算法。MP3 解码中所采用的 IMDCT 变换分为 36 点长块和 12 点短块，与 DCT 快速算法所能够计算的 2^N 点数不同。考虑到标准的 IMDCT 变换多为 2^N，本文提出对 MP3 中的 12 点与 36 点变换进行拆分，并采用长度为 32 位的近似系数进行计算，如图 7 所示。

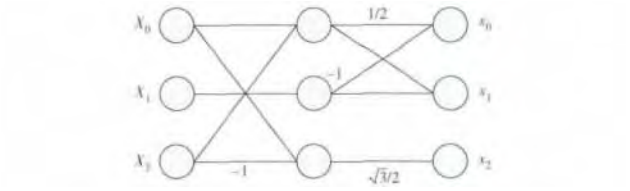

图 7　3 点 DCT 拆分

首先利用余弦函数对称性将式(6)转换为 DCT-IV 的形式，有

$$x_i = \sum_{k=0}^{\frac{N}{2}-1} X_k \cos\left(\frac{\pi}{N}\left(i+\frac{1}{2}\right)(2k+1)\right)。\quad (7)$$

式中，$i = 0,1,\cdots,\frac{N}{2}-1$。对式(7)进行递归，即可得到 DCT-II 的形式，有：

$$\begin{cases} x'_i = \sum_{k=0}^{\frac{N}{2}-1} X'_k \cos\left(\frac{\pi}{N}(2k+1)i\right); \\ x'_i(n) = x_i(n) + x_i(n+1); \\ X'_k = 2X_k \cos\left(\frac{\pi}{2N}(2k+1)\right)。 \end{cases} \quad (8)$$

式中，$i = 0,1,\cdots,N/2-1$。得到式(8)后，即可运用 DCT-II 的递归拆分算法进行计算，并将 12 点与 36 点的 IMDCT 运算全部转化为 3 点的 DCT 运算。

3　模块硬件设计及 FPGA 验证

本文采用 Verilog HDL 对模块进行设计，使用 Modelsim 6.5 进行仿真，并将模块下载至 Xilinx Virtex 5 开发板及专门设计的扩展板中进行测试与实验，如图 8 所示。

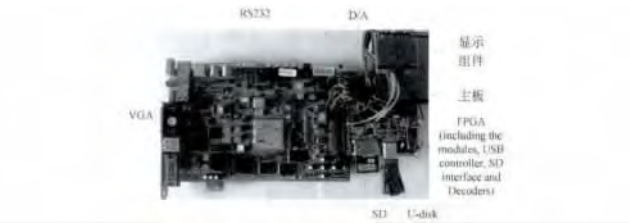

图 8　Virtex 5 开发板与扩展板

对于使用分步查表法的霍夫曼解码模块，索引表与内容表的硬件结构是设计的难点，本文提出了一种码表空间结构，如图 9 所示。

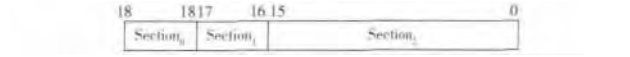

图 9　霍夫曼码表结构

码表位宽为 19 bit，$Section_0$ 用于区分叶节点与枝节点，包括 1 个比特；由于不是每个码字都是 4 的整数倍数，$Section_1$ 的值表示在该次该级查表中所使用的比特个数；$Section_2$ 给出相应的四元或二元频率线值(叶节点)或跳转地址(枝节点)。

分布查表的霍夫曼模块状态转换如图 10 所示。

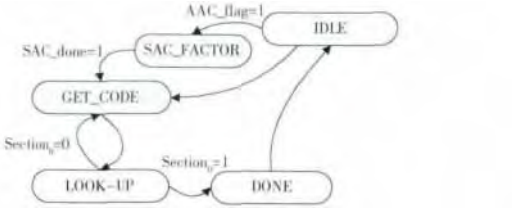

图 10　霍夫曼模块状态转换图

在反量化变换模块中，本文将每个补偿因子所包含的 q 数目设定为 2 的整数次幂，在处理时可以在获得除 8 的余数后，直接通过移位操作，将当前的值作为地址增量与起始地址值相加，通过查表获取补偿因子，并与并行进行的基础表查表的结果相加，进而获取终值。对补偿因子进行存储，并进行了与基础表一致的放大(通过乘以 2^N 实现)后，可在充分利用码表空间的同时进一步减小误差。

反量化模块状态转换如图 11 所示。状态机根据是否需要进行补偿进行状态跳转。

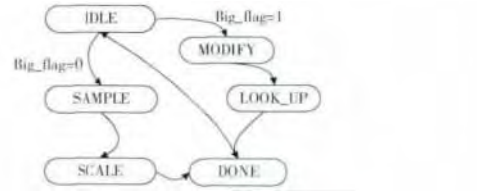

图 11　反量化模块状态转换

在可配置的 IMDCT 模块中，输入的 2^{N+1} 个数据通过预处理，成为式(8)的 DCT-II 的形式，并根据其数据个数进行 N 次递归拆分。IMDCT 8 点拆分流程如图 12 所示。

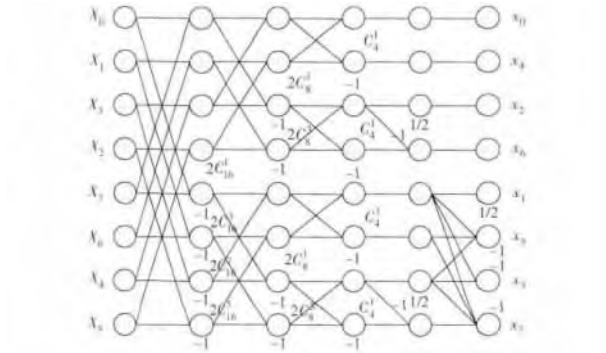

图 12　IMDCT 8 点拆分流程图

根据测试结果可知，本文中霍夫曼解码模块码表的大小为 2 998×19 bits，与文献[8]中二叉树码表的大小相当，而平均查表速度约为其 2.7 倍，实现了

查表速度的显著提升。

相比于表 2 中的方法，本文提出的反量化模块的运算量为 2F 1A，信噪比为 77.4 dB，码表空间占用为 1 246。在计算量与码表占用未明显增加的情况下，实现了精度（信噪比）的数量级提升。对相对误差取对数 Rd，实测曲线如图 13 所示。

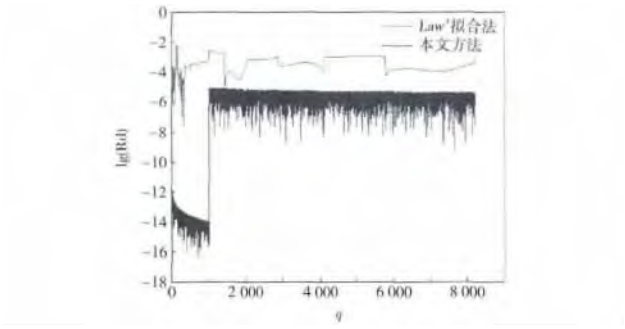

图 13 反量化相对误差对数曲线

本文中所采用的 IMDCT 算法（$N+1$ 点时）的乘法次数为 $\frac{N}{2}\times\log_2(N)+2N\times\log_2(N)+2N$，加法次数为 $N\times\log_2(N)+4N$。使用随机抽取的 5 个样本片段分别进行软件与硬件 IMDCT 解码变换，对比结果见表 3。

表 3 软硬件 IMDCT 对比

测试文件	总数目	不同点个数	最大相对误差/%
1. MP3	414 720	1 132	0.001
2. MP3	345 600	678	0.001
3. AAC	143 360	1 224	0.002
4. AAC	316 800	3 346	0.002
5. AAC	230 400	2 099	0.001

4 结 论

本文设计并实现了可支持多制式的霍夫曼模块、反量化模块以及 IMDCT 模块。借助分步查表、补偿因子以及 DCT 递归拆分算法实现了对应模块的功能。实际测试表明，上述模块在解码精度与速度方面均实现了一定的优化与提升。

[参考文献] (References)

[1] ISOIEC, 11172-3-1993. Information technology-coding of moving and associated audio for digital storage media up to about 1. 5 M bit/s-Part3: audio [S]. 1993.

[2] ISOIEC, 13818-7-2006. Information technology-generic coding of moving pictures and associated audio information-Part7: advanced audio coding [S]. 2006.

[3] 曾文龙，周游，贺珊，等. 基于多通道 DMA 控制器的媒体播放器 SoC 设计与验证[J]. 中国科技论文，2014，9(1)：45-47.
Zeng Wenlong, Zhou You, He Shan, et al. Design and validation of SoC media player based on multichannel DMA controller [J]. China Sciencepaper, 2014, 9(1): 45-47. (in Chinese)

[4] Lee K S, Park Y C, Youn D H. Software optimization of the MPEG audio decoder using a 32 bit MCU RISC processor [J]. IEEE Transaction on Consumer Electronics, 2012, 48 (3): 671-676.

[5] Wang Hui, Xu Wenbin, Dong Xin. Implementation of MPEG-2 AAC on 16-bit fixed-point DSP [C]//IEEE Asia Pacific Conference on Circuits and Systems. Singapore, 2006:1903-1906.

[6] Hedberg H, Lenart T, Svensson H. A complete MP3 decoder on a chip [C]// IEEE International Conference on Microelectronic Systems Education. Anaheim, USA, 2005:103-104.

[7] Tsai Tsung-Han, Liu Chunnan, Wang Yiwen. A pure-ASIC design approach for MPEG-2 AAC audio decoder [C]//IEEE 4th Pacific Rim Conference on Multimedia, Information, Communications and Signal. Singapore, 2003:1633-1636.

[8] Ho S, Law P. Efficient hardware decoding method for modified Huffman code [J]. IEEE Electronics Letters, 1991, 27(10):855-856.