

基于定点 DSP 的 MP3 解码器的系统级仿真与实现

王笑钢, 付宇卓

(上海交通大学计算机科学与工程系, 200030)

摘要: 便携式 MP3 播放器目前非常流行。在定点 DSP 上实现 MP3 解码的关键在于处理有限字长效应。该文介绍了一种用 OCAPI 对 MP3 解码算法进行基于 C++ 的系统级定点仿真的方法。通过对各个处理步骤迭代求精, 从而引出各数据处理步骤中间运算结果及滤波系数的最佳字长及 Q 值。最后, 在 DSP56858 上实时地实现了与 C++ 仿真代码比特精准的汇编代码。测量得到的解码输出的信号/量化噪声比约为 87dB, 总的 MIPS 开销约为 60。

关键词: 数字信号处理器; 有限字长效应; 定点仿真; 实时; 信号/量化噪声比

中图分类号: TN911.72 **文献标识码:** A

System Level Simulation and Implementation of the Fixed-point DSP Based MP3 Decoder

WANG Xiao-gang, FU Yu-zhuo

(Department of Computer Science and Engineering, Shanghai Jiaotong Univ., Shanghai 200030, China)

ABSTRACT: The portable MP3 player is now very popular. The key to realize the MP3 decoder on fixed-point DSP is to deal with the finite word-length effects. A C++ based system level fixed-point simulation method of the MP3 decoder algorithm via OCAPI is introduced. After iterative refinement of each processing step, the optimal word-length and Q-Value of each intermediate results and coefficients are evolved. Finally, the bit-exact assemble code is implemented on DSP56858 in real-time. The SQNR of the decoded output is measured, about 87 dB, with the overall MIPS overhead about 60.

KEYWORDS: DSP; Finite word-length effects; Fixed-point simulation; Real-time; SQNR

1 概述

MP3 全称 MPEG-1 音频第 3 层 (MPEG-1 Audio Layer III), 它融合了 MPEG 音频第 1 层第 2 层中的 MUSICAM (掩蔽型通用子带综合编码及复用) 算法和 ASPEC (自适应谱分析听觉熵编码) 算法, 并且通过应用混合滤波提供额外的频率分辨率, 另外还采用非均匀量化、自适应分段、熵编码等, 获得了最高的压缩比, 已成为网络数字音频的压缩标准。

在消费电子领域涌现出大量的便携式 MP3 播放器, 其着重考虑的是音质、功耗、尺寸, 在实现上有基于 DSP+MCU 的和基于 ASIC 的。本文选取的目标机摩托罗拉 DSP56858 在单片内真正融合了 MCU 的控制功能和 16 位定点 DSP 的处理能力, 对外呈现单一指令集, 可大大降低总体器件成本, 减小电路板空间并具有更高的可靠性。

MP3 解码算法定点化的关键在于在有限的 MIPS 开销及内存使用的约束下尽量减小有限字长造成的量化误差, 保证音频解码输出具有较高的 SQNR。OCAPI 充分考虑到定点

DSP 固有的有限字长效应, 其定点类全面地描述了定点数的数值行为特性。作者首先简介 MP3 解码算法, 提取出关键处理步骤; 随后用 OCAPI 对 MP3 解码算法进行基于 C++ 的系统级定点仿真, 对目标 DSP 的数据通道建模, 迭代求精, 快速引出各数据处理步骤中间运算结果及滤波系数的最佳字长及 Q 值, 得到内存使用情况, 给出了 SQNR 评估方法及结果; 最后在目标机上为各数据处理基本块实现了比特精准的实时汇编, 并给出了 MIPS 开销分布。

2 MP3 解码算法描述

2.1 解码过程

MP3 解码过程如图 1, 整个解码算法所涉及的滤波系数

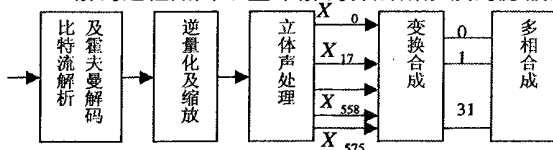


图 1 MP3 解码过程

都已在标准中指定。首先对输入缓冲中的音频数据按照 ISO/IEC 11172-3 规定的帧格式进行比特流解析, 捕获同步头, 恢复出各种头信息(版本、层数、抽样率等)和边信息; 接着对量化样值进行霍夫曼解码, 至此, 所处理的数据均为定点整数; 然后, 根据前面得到的边信息、全局增益、子块增益、窗口类型等信息, 对霍夫曼解码得到的频谱量化值作逆量化及缩放, 得到均匀量化的频谱线; 对频谱线做混叠消除, 经 IMDCT(反向的改进离散余弦变换)得到各个子带原始的频谱线; 最后通过多相子带合成滤波器处理, 完成频域-时域变换, 输出 PCM 音频样值。

2.2 关键处理步骤

可以对以上描述的处理步骤进行进一步分析。从霍夫曼解码后才有浮点型数据的处理, 之前的步骤不需要定点化, 也就不影响数值性能。提取出影响数值性能且计算密集的三个主要处理步骤: 逆量化及缩放、IMDCT、子带合成滤波。

2.2.1 逆量化及缩放

频谱逆量化及缩放过程是一非线性计算, 可由式(1)描述。

$$\begin{aligned} \text{xr}(i) = & \text{is}(i)^{4/3} * 2^{.25 * (\text{global_gain}[g] - 64 - 8 * \text{subblock_gain}[\text{window}[g]] * \\ & * 2^{.25 * (-2 * (1 + \text{scalefac_scale}[g]) * \text{scalefac}[\text{cb}][\text{window}[g]] - 2 * \text{preflag}[g] \\ & * (1 + \text{scalefac_scale}[g]) * \text{pretilt}[d])} \end{aligned} \quad (式 1)$$

$$\text{可以简化为下式: } \text{xr} = \text{Sign}(\text{is}) * |\text{is}|^{4/3} * 2^m \quad (式 2)$$

式(2)可分为左、中、右三部分: 左边 $\text{Sign}(\text{is})$ 为符号函数; 右边 2 的幂运算可使用 DSP 的移位操作完成; 中间的求 $|\text{is}|^{4/3}$ 次方如果显式计算的话, 计算量太大, 应该用静态查找表方式做近似计算。Is 的理论动态范围是 $[-8191, 8191]$, 如果直接查找的话需要一个 8192 大小的查找表, 而且表项的动态范围也比较大, 达 2 的 52/3 次方, 使得剩余的精度不够。因此, 使用一种查找表结合线性内插的方法。式(2)中间部分可展开:

$$\begin{aligned} |\text{is}|^{4/3} &= |\text{is}| \times |\text{is}|^{1/3} = |\text{is}| \times (|\text{is}/8| \times 8)^{1/3} \\ &= |\text{is}| \times |\text{is}/8|^{1/3} \times 8^{1/3} = |\text{is}| \times |\text{is}|^{1/3} \times 2 \end{aligned} \quad (3)$$

将查找表大小压为 257(双精度), 表中保存的是 $|\text{is}|^{1/3}$ 的 1/3 次方而不是 $|\text{is}|^{4/3}$ 次方, 这样表项的动态范围急剧缩小, 仅 0.7, 极大地提高了精度。小于等于 256 的直接在表中查得; 对大于 256 先使其缩至 1/8(右移 3 位)或 1/64(右移 6 位)后落入区间 $[0, 256]$ 执行查找, 为了减小缩小 1/8 或 1/64 时截断造成的近似误差, 再做牛顿法线性内插。当然做线性内插需要计算一次均差和修正值, 增加了一定计算量。

2.2.2 IMDCT 及子带合成滤波

这两部分均含大量的 MAC(乘-累加)操作, 非常适合 DSP 实现。由于计算负载过大, 有必要采用一定的快速算法。IMDCT 的计算公式如式(4), 长窗口的余弦系数有 $36 * 18 = 648$ 个, 短窗口的余弦系数有 $12 * 6 = 72$ 个。子带合成滤波的加窗矩阵的 512 个系数都已在标准中给出。

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

$$\text{for } i = 0 \text{ to } n-1 \quad (4)$$

在此, IMDCT 采用了 9 点的 Lee 快速 IDCT 算法^[1], 比标准推荐的常规算法减少 2/3 的计算量; 子带合成滤波的矩阵乘法采用了 64 点的 Lee 快速 DCT 算法, 比标准推荐的常规算法减少了 1/2 的计算量, 同时内存使用也减少了 1/2。一般来说快速算法会比标准推荐的常规算法带来更大的量化误差。作者在后续的仿真中比较了所采用的快速算法与常规算法的数值性能差别, 发现快速算法比常规算法的 SQNR 只下降了 0.6 左右, 相对 16 位定点 DSP 达 96dB 的 SQNR 而言完全可以接受。

3 算法的定点仿真

传统的定点仿真有基于 Matlab 和基于标准 C 的。其中 ITU-T 和 ETSI 开发了一套定点算术操作的库函数, 用以模拟定点 DSP 的行为。这些库使得开发者在 PC 或工作站对 16/32 位通用 DSP 做 ANSI C 一级的定点仿真。这些库不但提供了比特精准的数值性能, 而且包含了 DSP 算法复杂度的运行时度量支持(通过对基本操作计数以及对基本操作赋加权系数)。这种定点仿真有缺点: 1) 它只是伪汇编代码, 系统级的描述能力不够; 2) 它并不能完全开发 DSP 的效率, 因为象重排序或是扩展精度寄存器等操作会改变算法的数值行为。

因此, 需要用一种系统级的建模仿真平台, 能快速有效地为算法的基本处理块建模, 在仿真过程中监控各基本处理块的数值性能(与浮点算法相比较的 SQNR), 以较少的迭代次数获取各数据处理基本块中间运算结果及滤波系数的最佳字长及 Q 值, 为各数据处理基本块的比特精准实时汇编编码奠定数值基础。OCAPI 正是这样一个平台。

3.1 OCAPI

OCAPI 是由欧洲最大的独立的微电子研究中心 IMEC 开发出来的一个系统芯片(SOC)设计工具。OCAPI 用 C++ 语言以统一的模型形式在不同层上进行功能描述, 且独立于所采用的体系结构与实现方式。而且它是一个设计自动化系统, 能帮助用户正确折中权衡系统中的软件和硬件, 并从一个单一的 C++ 描述生成软件和硬件部分。在此作者使用 OCAPI 的一个功能受限版本, 只用于定点仿真, 不能做自动代码生成。OCAPI 以一个 GNU C++ 二进制库文件和一族头文件的形式发布, 主要提供了以下的几类:

- class dfix, 定点类, 定点仿真最重要的类, 浮点数/定点数统一的数据类型。

- class ddbfix, 队列类, 基本块之间通信的 FIFO, 其队列元素是 dfix, 队列类在软件上体现为 ROM 查找表, 全局堆。它是数值性能统计的对象, 同时也是内存使用的主体。

- class base, 基本块类, 数据流图(Data Flow Diagram)中的结点, 派生出输入/输出块, 数据处理块(用户自定义), 在软件上体现为各个数据处理步骤, 如仿真文件操作、逆量化及缩放、IMDCT、子带合成滤波等, 基本块之间共享一个或队列。它是 MIPS 统计的对象。

- class _sig, 内部信号类。
- class schedule, 调度机类: 离散事件仿真的引擎。它显式地驱动每个基本块的执行。
- class FSM, 有限状态机类。

其中做为核心的定点类 dfix 的部分定义如下:

```
class dfix {
    double DVa; // IEEE 754 双精度数, 仿真引擎中数值表达的真正载体

    int mode; // 仿真模式: 浮点(默认), 或是定点
    int W; // 总共的字长, MSB+LSB, 其中 MSB 用来描述定点数的动态范围
    int L; // LSB 位宽, 即小数点之后的有效位数, 用来描述定点数的精度
    int rep; // 定点数表示方法: 补码, 即有符号整数(默认), 或是无符号整数

    int ovf; // MSB 的行为模式: 溢出(默认), 或是饱和
    int rnd; // LSB 的行为模式: 截断(默认), 或是舍入
    ....
}
```

定点类 dfix 非常灵活, 下面有几个构造实例:

- dfix a(3.875); 定义一个浮点数, 初值为 3.875。默认的类型就是浮点数。
- dfix a(3.875, 16, 4); 定义一个定点数, 初值为 3.875, 字长为 16 小数点后有 4 位。未超出动态范围, 精度也足够, 没有产生量化误差。
- dfix a(3.875, 16, 2); 定义一个定点数, 初值为 0.875, 字长为 16, 小数点后有 2 位。未超出动态范围, 但是精度有损失! 由于 LSB 默认是截断, 所以定点化后的数值为 3.75, 定点化产生的绝对误差为 0.125, 相对误差为 3.23%。
- dfix a(3.875, 16, 14, dfix::tc, dfix::wp, dfix::rd); 定义一个定点数, 初值为 0.875, 字长为 16 小数点后有 14 位。精度是足够了, 但是超出了动态范围! 由于 MSB 指定为溢出, 表示方法指定为补码, 所以定点化后的数值为 1.875, 定点化产生的绝对误差为 2.0, 相对误差为 51.61%。

3.2 仿真步骤

用 OCAPI 做定点仿真的步骤如下: 1) 根据 MP3 解码算法的数据流程图概括地划分基本块和全局共享数据(用定点类定义所有的数据, 而且使用默认的精度, 即 IEEE 754 双精度数), 并用调度机类指定基本块的推进序列, 得到浮点 C++ 程序; 2) 不改变浮点 C++ 程序的控制结构, 只对有待定点化的变量 X 在其构造函数里指定其字长、LSB、是否有符号、溢出或是饱和、截断或是舍入, 过程如图 2 所示。从中可以看到: 1) 对某一变量, 先定范围再定精度, 随后精度不够就加宽字长, 这是因为溢出比截断更可怕; 2) 每次迭代的反馈都是基于统计特性的; 3) 每个基本块都可以控制自己的 SQNR 衰减。这是一种自顶向下、带有约束性的、迭代求精的设计方法。

为了达到有效的算法—体系架构映射, 用 OCAPI 提供的类, 对目标 DSP 的数据通道建模, 在硬件平台就绪之前, 在仿真阶段逐步加入目标平台实现的细节, 得到一个数据通道模拟器。未加入时序特性的话, 实现起来比较简单, 相当于数值统计分析工具, 又可做为 Profiling 工具, 可以统计出每个基本块的执行次数及内存使用状况。

对 16 位 DSP 而言, 变量的字长只可能是 16 位(单精度)或 32 位(双精度), 因而需要迭代尝试的次数比较少。对于被定为双精度的变量, 其乘运算为 32 位 * 16 位, 要分解为高低机器字两个 16 位 * 16 位乘法, MIPS 开销比较大。DSP56858 的累加器有 4 比特的保护位, 理论上可以有 15 次的双精度数溢出。为解决定点乘法的比特增长问题, 乘法运算的中间结果必须以 32 位(双精度)表示。为避免连续 MAC 操作的中间结果溢出, 运算过程必须利用累加器的 4 比特保护位 36 位(扩展双精度), 如果还是溢出, 则只有在输入支路上加支路衰减因子(右移若干位, 从 1 开始尝试), 限制输入信号动态范围, 防止溢出。

3.3 仿真结果

经过迭代仿真, 将所有浮点数据定点化, 其中得到的所有重要队列的字长及 Q 值如表 1 所示。未列出的所有余弦滤波系数均为 16 位(单精度), 1.15 格式有符号数。除了霍夫曼解码, 所有的中间运算结果都是 32(双精度)。在仿真完成时, 可得到算法的数值性能和内存使用, 至于准确的 MIPS 开销, 要等到实时汇编优化完成才能得到。内存使用(以机器字计算)情况为: 1) ROM 表, 5.7K; 2) 数据 RAM, 11.6 K; 3) 程序 RAM, 8.12 K。

表 1 关键队列定点化后的构造信息

队列名称	变量意义	MSB	LSB	符号	队列长度 (字)
is	霍夫曼解码的输出	16	0	有	1152
is_pl3_tab	Is 的 1/3 次方查找表	3	29	无	257 * 2
global_gain	全局增益	14	2	无	4
xr	逆量化及缩放的输出	17	15	有	1152 * 2
Cs - Ca	混叠消除蝶型运算系数	0	16	无	8.8
Xr_DCT	IMDCT 的输出	16	16	有	1152 * 2
synth_window	子带合成的加窗矩阵	2	14	有	512
pcm_out	PCM 解码输出	1	15	有	576

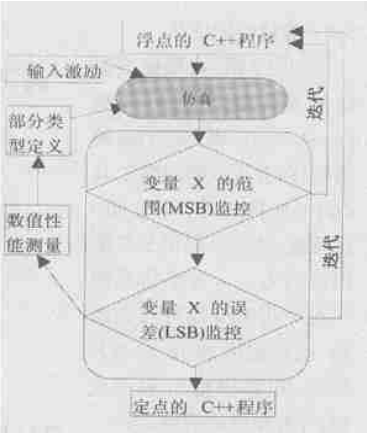


图 2 OCAPI 的 C++ 定点类迭代求精的设计流

3.4 数值性能评估

如图 3 所示, 用标准提供的测试比特流输入到 Fraunhofer IIS 提供的 FhG 编码器(内置于音频编辑工具 Cool Edit Pro 2 中)得到 MP3 比特流, 然后分别用 FhG 解码器和 C++ 定点仿真解码器对此 MP3 比特流做解码, 比较输出的 16 位 PCM 样值, 在解码输出端用式

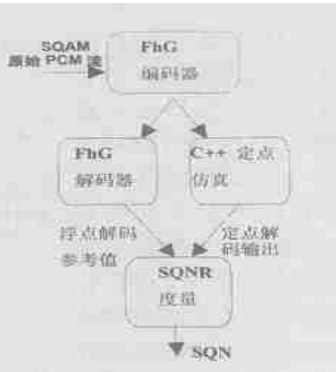


图 3 输出端定点化数值性能评

(5) 计算最终的信噪比:

$$\{ 20 \times \log_{10} (2^{16} - 1) \} - \{ 20 \times (\log_{10} 2) \times \text{RMS}_{\text{PCM_DIFF}} \} \text{ (dB)}$$

其中 $\text{RMS}_{\text{PCM_DIFF}}$ = PCM 差值的均方根。根据仿真得到的统计可知 16 位 PCM 的差值最大是 4 绝大多数小于 2 即不到 2 比特的精度损失。最终由式(5) 计算得出的 SQNR 约为 87dB。

4 算法的实时汇编实现

最后作者用 DSP56858 的汇编语言对 OCAPI 定点仿真 C++ 代码的各基本块实现了 OCAPI 定点仿真 C++ 代码的比特精准、实时的汇编代码, 总的 MIPS 开销约为 60。具体分布如表 2 所示。DSP56858 最高计算能力为 120MIPS(120MHz 时)。从中作者总结了一些手工汇编优化的要点: 1) 用内嵌 (inline intrinsic) 减少过程调用的开销; 2) 用机器指令完成如饱和、舍入、规格化等特殊的算术操作; 3) 对局部块进行寄存器手工染色, 块操作中尽量避免访存的“颠簸”; 4) 必要时局部块指令重排, 填充空操作指令(NOP)以避免流水线停顿; 5) 以硬件循环代替软件循环, 消除循环结束判断的开销; 6) 嵌套循环情形下, 内循环尽量紧凑, 适当将内循环展开调度; 7) 充分利用保护位做扩展双精度的 MAC 操作; 8) 开发功能部件并行性, 在单周期内完成 MAC、读取两个源操作数、更新两个源操作数地址; 9) 将内存数据适当对齐, 以利用循环寻址。

表 2 目标板上测得的基本块 MIPS 开销

基本块(处理步骤)	MIPS 开销
缓冲管理/ MP3 比特流解析	0.12
缩放因子解码/ 霍夫曼解码	6.65
逆量化及缩放	5.44
联合立体声/ 混叠消除/ 重排序	1.89
IMDCT	17.30
子带合成滤波	28.62
总计	60.02

最小原型系统的周边接口电路如图 4。DSP56858 片内未集成 USB Host/ Device 模块, 考虑外挂一块 USB device 控制器。软件上支持 USB 大容量存储协议及标准 FAT16 文件格式。与 PC 连接时 MMC 闪存卡成为 U 盘。

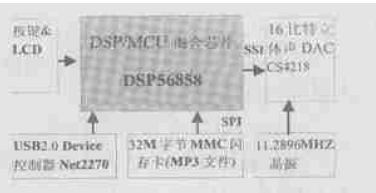


图 4 原型系统功能框图

5 结语

至此, 一个独立的 MP3 解码器原型系统的仿真及实现就完成了, 解码输出端测得的 SQNR 约为 87dB。总的 MIPS 开销约为 60。系统级的建模仿真工具 OCAPI 有效地加快了原型系统的开发。还有一些问题需要改进, 例如手工汇编时要更多地避免流水冲突, 降低 MIPS 开销; 变量复用以减少内存使用。

参考文献:

[1] Byeong Gi Lee, A New Algorithm to Compute the Discrete Cosine Transform[J] . IEEE Transactions on ASSP, December 1984, 32(6): 1243—1245.

[2] R Cmar, L Rijnders, P Schaumont, S Vernale and I Bolsens. A methodology and design environment for DSP ASIC fixed point refinement[C] . Design Automation and Test in Europe Conference and Exhibition 1999. Proceedings, 1999, 271 — 276.

[3] Keun-Sup Lee, Hyen-O Oh, Young-Cheol Park, Dae Hee Youn. High quality MPEG—audio layer III algorithm for a 16—bit DSP[C] . The 2001 IEEE International Symposium on Circuits and Systems (ISCAS 2001), May 2001, 2(6—9): 205 — 208.

[4] Hong-Seok Kwon, Si-Ho Kim, Keun-Sung Bae. Real time implementation of MPEG—1 Layer III audio decoder with TMS320C6201 DSP[C] . 2000 IEEE Workshop on Signal Processing Systems (SiPS 2000), 11—13 Oct. 2000, 761 — 770.

[5] Jin-Won Kim, Nam-Hun Jeong, Jun-Seok Kim, Keun-Sup Lee, Chung-Yong Lee. VLSI Implementation of the MPEG—1 Layer III Audio Decoder[C] . Proceeding of the 1999 Korean Signal Processing Conference, 1999 12(1): 847—850.

[6] 丁玉美, 高西全. 数字信号处理(第二版)[M] . 西安电子科技大学出版社, 2000—12.

[7] Ken C Pohlmann 著, 苏菲译. 数字音频原理与应用(第四版)[M] . 电子工业出版社, 2002—2.



[作者简介]

王笑钢(1978—), 男(汉族), 江西萍乡人, 上海交通大学计算机科学与工程系硕士研究生, 主要研究方向为数字信号处理、软硬件协同设计;

付宇卓(1968—), 男(汉族), 黑龙江哈尔滨人, 上海交通大学计算机科学与工程系副教授, 主要研究方向为 MPEG—2 编码器的结构、运动估计。