

# 低成本 SoC 平台 MPEG - 4 AAC 解码器优化策略

高谷刚, 时龙兴, 浦汉来, 周 凡

(东南大学国家专用集成电路系统工程技术研究中心, 南京 210096)

[摘要] 提出了在低成本 SoC 平台的软件优化策略, 包括基于统计分析的浮点到定点转换策略和面向性能优化的片上存储器容量定制策略, 在此基础上针对计算密集性应用提出了完整的优化方法。通过实现 MPEG - 4 AAC 实时解码, 表明该优化策略在性能和成本方面的有效性。这一策略与方法也适合其他 DSP 应用。

[关键词] 软件优化; SoC; 浮点到定点转换; 片上存储器; AAC

[中图分类号] TN47 [文献标识码] A [文章编号] 1009 - 1742(2007)10 - 0060 - 05

## 1 引言

消费电子市场对嵌入式数字音频处理产品的需求十分旺盛, 主流的音频格式是 MPEG - 1 Layer - 3<sup>[1]</sup>, 但 MPEG - 4 先进音频编码(AAC)标准<sup>[2]</sup>正以其卓越的性能获得了越来越多的关注。MPEG - 4 AAC 提供比以往编码方法更高的编码效率, 比 MP3 算法大约节省 30 % 的带宽, 在 128 kb/s 上比特率超过 MP3 而达到 CD 音质<sup>[3]</sup>, 但相应也增加了计算复杂性。

AAC 解码是典型的数字信号处理(DSP)应用, 算法以浮点操作为主, 而出于成本和功耗的考虑嵌入式系统通常是定点架构。定点平台实现 DSP 带来的最大挑战: 如何满足高计算负荷的需求; 以及在算法浮点一定点转换(FFC)中保证计算结果的精度。

嵌入式 AAC 解码产品硬件平台分为专用 ASIC<sup>[4]</sup>, DSP 核<sup>[5, 6]</sup>, 以及 DSP 核或 RISC 核结合专用硬件加速模块<sup>[7~9]</sup>。这些平台存在着成本、功耗、上市时间等问题, 不太适应低成本便携产品的需要。在标准不断发展和新标准不断涌现的背景下, 为了满足上市时间的需求嵌入式系统的趋势是可编程的

解决方案。随着深亚微米制造技术和嵌入式微处理器设计技术的发展, 先进的 RISC 微处理器性能能够支撑多媒体应用。文献[10]在单 RISC 平台实现 MPEG - 2 AAC 实时解码, 但 FFC 过程采用仿真的方法, 软件优化又集中于汇编级, 造成设计时间长, 可移植性差。笔者提出一种先进的优化策略, 并在此基础上提出完整的优化流程, 实现了在低成本 SOC 平台上 MPEG - 4 AAC LC Profile 实时解码, 此策略可应用于其他 DSP 应用。

## 2 目标平台 系统虚拟原型与优化流程

### 2.1 目标平台

目标架构为东南大学 ASIC 系统工程技术研究中心基于 AMBA 总线规范研发的嵌入式片上系统(SoC), 内核 ARM7TDMI<sup>TM</sup>, 面向低成本手持设备和其他通用嵌入式设备。此芯片的主要特性:

- 主频 75 ~ 100 MHz;
- 内嵌 20 kB 片上零等待静态存储器;
- 外部存储器接口控制器(EMI), 内含指令预取 FIFO<sup>[11]</sup>, 支持 SRAM, SDRAM 等 4 类存储器。

随着嵌入式处理器计算能力的提高, 计算负载不再是系统的主要瓶颈, 相反数据通信对性能影响

[收稿日期] 2006 - 09 - 13; 修回日期 2006 - 12 - 21

[基金项目] 国家自然科学基金资助项目(60676011; 90407009); 高等学校博士学科点专项科研基金资助项目(20050286040); 教育部跨世纪优秀人才培养计划基金资助项目

[作者简介] 高谷刚(1975 - ), 男, 江苏高淳县人, 江苏警官学院工程师, 东南大学博士生, 主要从事 SOC 设计和多媒体算法软硬件协同设计研究

越来越重要。ARM7TDMI 内核无 Cache, 访问存储器不具有流水特性<sup>[12]</sup>, 为了提高系统性能, 芯片内嵌 20 kB 片上高速存储部件 (ESRAM), 这为优化策略提供了结构基础。

## 2.2 系统虚拟原型

评估嵌入式软件性能一般有 3 种方法<sup>[13]</sup>: 评估板、软硬件协同验证和指令集模拟器 (ISS)。指令集模拟器采用 C 语言高层建模, 由于抽象层次较高, 精度不如前 2 种方法, 但可以快速建立系统原型, 修改方便, 利于调试, 模拟速度较软硬件协同验证环境有 100 ~ 1000 倍以上的提高。节拍精度的指令集模拟器是模拟所需最高细节需要<sup>[14]</sup>, 采用节拍精度指令集模拟器建立系统高层模型。

ARMulator 是 ARM 公司提供的指令集模拟器<sup>[15]</sup>, 不仅可以仿真 ARM 处理器的体系结构和指令集, 同时提供扩展机制描述硬件, 可建立存储器和处理器外围设备的高层模型。ARMulator 仿真器主要通过指定地址空间的读写实现对扩展设备模型的调用, 因此可以方便地建立总线从设备模型; 总线主设备的模型建立过程比较复杂, 模型不能模拟内核与其他主设备抢占总线情况, 同时与硬件设计思路相去甚远。这样带来的问题: 无法建立用于性能评估的 SoC 系统模型; 建立的模型无法用于指导和验证硬件设计。建立节拍精度总线模型, 提供接近实际硬件的总线接口, 其余模块模型由总线接口和功能实现两部分组成, 总线集成了 EMI 模型以及 SDRAM 和 SRAM 存储器模型。软件代码加载到存储器模型中, 由 ARM ISS 解释执行, 每次内存存取将调用模型, 总线模型仲裁主设备模型和内核的总线请求, 模拟两者抢占总线及对从设备访问等操作, 并得到执行指令数、周期数等各种信息。高层次系统建模中最大的挑战是模型的精确性。利用 Gunzip, CRC 等 Benchmark 程序, 通过 RTL 仿真 (精确的性能评估) 和高层模型仿真进行对比实验, 证明高层模型的准确性可以达到 95 % 左右, 完全达到作为建模和性能分析工具的要求, 且速度大大提高。

最终的系统模型如图 1, 除 ARM ISS, Cache/MMU (memory management unit) 模型, 其余的模型自建。利用这一平台, 结合 ARMulator 的轮廓 (profile)、追踪 (trace) 等工具, 可以进行软件开发和系统性能的评估, 也可以作为软硬件划分的平台。下面提出的优化策略就建立在这一平台上。

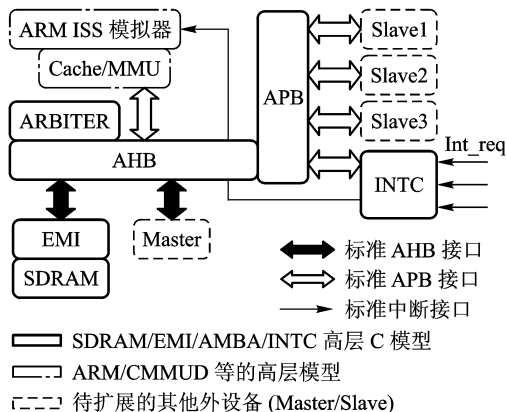


图 1 系统模型架构

Fig. 1 System model architecture

## 3 优化策略

优化流程建立在系统高层模型上, 如图 2 所示。

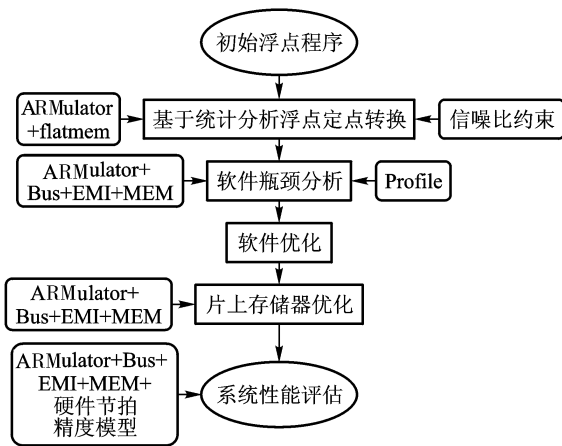


图 2 设计方法学

Fig. 2 Design methodology

低成本嵌入式系统采用定点结构的处理器, 浮点算法的实现采用浮点库仿真的方法, 大大影响了系统的性能, 需要使用近似定点数算法来代替浮点算法。浮点到定点的转换策略一般基于数据仿真方法, 该方法的重大缺陷就是需要大量仿真时间, 而且测试结果具有一定的片面性。采用基于统计分析的方法实现浮点到定点的转换, 在满足信噪比约束的条件下大大缩短转换时间。定点化后软件由于算法选择和编译产生的代码冗余, 需要进一步的优化。在建立的系统模型平台上, 利用 ARMulator 提供的 Profile 功能确定软件瓶颈之后, 进行算法和汇编级优化。优化集中在计算密集部分, 目标是尽量减少软件优化的时间和人力投入。由于处理器与存储器

速度的差异,以及多媒体应用数据结构特点,大量的运行时间消耗在存储器访问上。优化程序执行时间除程序执行路径优化外,可以通过加快存储器介质访问速度实现,即将代码、数据放入片上 ESRAM。出于成本考虑,片上 ESRAM 容量有限,如何最有效利用片上 RAM 成为急需解决的问题,运用面向性能优化的片上存储器容量定制策略,有效地解决了这一问题。

## 4 优化过程

### 4.1 基于统计分析的浮点到定点转换方法

从硬件开销和解码性能角度,希望定点化的字长尽量少;但为了保证解码效果(即计算精度),得出了相反的结论。因此 FFC 过程在性能和效果 2 个约束间达到最佳的平衡是个需要解决的问题。

基于数据仿真的定点化方法,需要手工测得各变量的动态范围,然后在保证精度的情况下确定各自的定标格式,费时费力,容易引入错误。为了节省仿真时间,利用基于统计分析的定点近似程序信噪比(SNR)计算方法,得到 SNR 与硬件字长、溢出概率间的关系<sup>[16]</sup>,在满足信噪比限制和硬件字长的条件下,计算出最佳转换参数。这一策略可以用于软硬件划分过程中确定硬件加速模块的最佳字长。

根据设定信噪比计算出最佳转换参数,实现工具半自动进行浮点到定点的转换,在保证转换效果的前提下,大大减少转换时间。实验结果表明浮点算法和定点算法间信噪比达到了 80 dB 左右,达到了高保真音质。

### 4.2 软件优化

MPEG-4 AAC LC 层解码流程如图 3 所示。各部分的功能参考文献[2]。嵌入式软件优化主要分成 2 个层次:高级语言层次算法优化和汇编语言级的代码优化。软件的优化要结合指令集、微处理器体系结构、流水线以及编译器进行考虑,人力成本较大。多媒体应用的计算负载集中在少数几个规则的算法,针对这类算法在 2 个层次进行了优化,在投入和效果间得到较好的折衷。

定点化后 MPEG-4 AAC 的解码程序,在仿真平台上进行运行,并利用 Profile 工具,得到了程序的函数级的时间分配信息。其中滤波器组(filter bank)占 CPU 运行时间的 66.96%,滤波器组中最重要的算法 IMDCT 占 CPU 运行时间的 57.26%,IMDCT 是计算密集型操作,算法比较规则,因此软

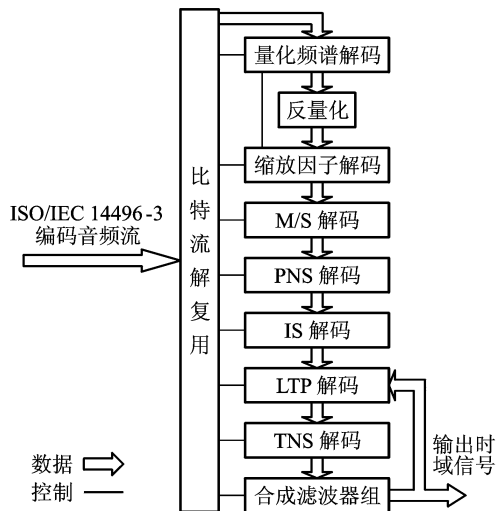


图 3 AAC LC 层解码流程

Fig. 3 AAC LC level decode flow

件优化重点就是 IMDCT。

IMDCT 的表达式如下:

$$x_{i,n} = \frac{2}{N} \sum_{k=0}^{N/2-1} X_{i,k} = \cos \left[ \frac{2\pi}{N} (n + n_0) \left( k + \frac{1}{2} \right) \right]$$

$$n = 0, 1, \dots, N-1 \quad (1)$$

式(1)的计算量很大,Duhamel 提出了利用 N/4 点的 FFT<sup>[17]</sup> 的 IMDCT 快速算法。IMDCT 算法流程如图 4 所示。

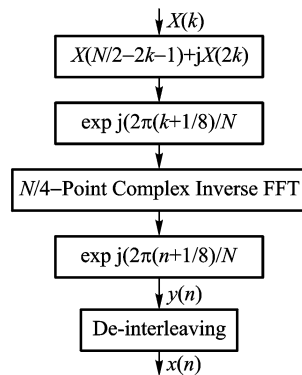


图 4 IMDCT 快速算法

Fig. 4 The fast IMDCT algorithm

在 C 代码优化以后,结合 ARM 处理器指令集进行汇编级优化,因为 FFT 算法包含大量的复数乘法,可以利用处理器提供的有符号数乘法指令 IMUL 和乘加指令 MAL 组成的汇编代码替换 C 代码对应进行优化。

### 4.3 片上存储器优化

系统芯片 (SoC) 设计的难题之一是嵌入式微处理器的高主频速度与片外存储器的低读取速度不匹配,在低成本、无 Cache 平台上,这很大程度上限制了系统的性能,解决方法之一便是引入片上存储器。因为应用程序的执行时间  $T$  由两部分组成:指令流固定执行时间  $T_i$  和访存时间  $T_m$ ,即

$$T = T_i + T_m \quad (2)$$

通过更改应用程序的内存布局,可以减小  $T_m$ ,明显改善程序性能。但是 SoC 片上存储器往往占据了很大的芯片面积,如何兼顾性能优化和芯片面积,便成了一个不可忽视的问题。该问题的核心就是片上存储器容量定制策略。它通过一定的分配策略,选择合适程序内容放入片上存储器,建立片上存储器容量和程序性能之间的联系,从而计算出针对该程序性能的最佳容量。笔者提出一种片上存储器容量定制策略,兼顾数据、堆栈和指令,采用了可变粒度的划分方法,得到最佳性能优化值。

针对片上存储器优化策略的流程主要由两个循环构成,如图 5 所示。片上存储器容量已经确定为 20 kB,所以只需内循环(划分粒度循环)6 步即可。

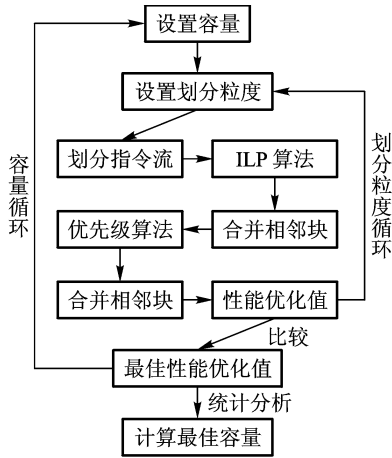


图 5 容量定制策略流程图

Fig. 5 Customizing scheme's flow chart

指令流划分以函数为单位进行。循环指令是函数划分的关键,以它为边界,可以分离出函数中使用最频繁的那些指令块。具体函数划分过程分为两个步骤,如图 6 所示。

Step 1 通过递归分析函数中所有循环指令,把函数(见图 6a)转化为一个节点树。其中,循环指令所包含区域称为  $L$  节点,其他指令所包含区域称为  $B$  节点。函数本身作为树的根节点,被视为一个  $L$

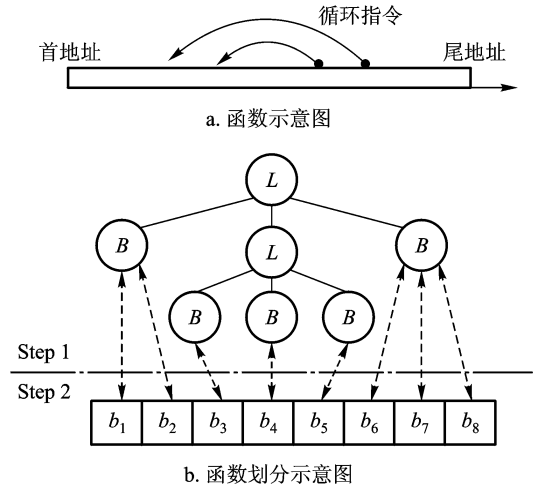


图 6 指令划分过程示意图

Fig. 6 Splitting process of instructions

节点。划分规则如下:

$$1) L \rightarrow mB + nL \quad (3)$$

条件  $S(L) > S_{\min}$  且  $n \geq 1, m > 0$ 。

$$2) L \rightarrow B \quad (4)$$

条件  $S(L) \leq S_{\min}$ ,

或  $L$  节点内部无循环指令,

其中  $S$  表示节点大小,  $m$  和  $n$  表示节点个数。

Step 2 划分所有叶节点。规则如下:

$$B \rightarrow mb_x + b_y \quad (5)$$

条件  $S(B) \geq S_{\max}$  且  $m \geq 0$ ,

其中  $S(b_x) = S_{\min}$ ,  $S(b_y) = S(B) - mS_{\min}$ 。

参数  $S_{\min}$  和  $S_{\max}$  用于控制函数划分粒度,以避免出现过或过小的指令块,可通过分析应用程序而自动获取。最后,函数被划分为一系列大小不超过  $S_{\max}$  的指令块。另外,全局数据变量和堆栈不参与划分,直接变为数据块。所以,整个应用程序被划分为若干指令块和数据块。划分得到的块,由 ILP 算法和优先级算法组合而成的选择算法,选择其子集,放置到片上存储器中,这样可以降低单纯 ILP 算法造成的片上存储器空间浪费,并获得最佳的性能提升。

这种策略在最佳容量即能够节省超过 80% 的片外存储器访问次数<sup>[18]</sup>。根据这种划分策略,用 C 语言设计了一个代码生成器。首先在上述平台上运行解码程序,利用 ARMulator 分析应用程序,生成代码详细的 Profile 和 Trace 信息,使用代码生成器自动完成应用程序的代码搬移,在应用程序运行前完

成内存布局修改。通过这一工具,能大幅提高软件的运行效率,平均达到 1 倍。同时功耗也大幅下降,达到了 70 % 左右,这也符合 Lee<sup>[19]</sup> 在结构不变的情况下,功耗和运行时间近似成正比的结论。

## 5 结果与结论

所提出的优化策略能极大地提高设计的效率,大概花费了 0.18 人年就完成了解码软件的移植和优化。对各个优化步骤后的解码软件性能进行评估,素材为 2 通道,128 kb/s,采样率 44.1 kHz,结果见表 1。

表 1 软件性能表

Table 1 Software performance improvement

| 软件版本  | 解码周期数          | 实时解码频率/MHz |
|-------|----------------|------------|
| 浮点    | 36 442 648 220 | 944        |
| 定点化   | 10 105 973 197 | 224        |
| 软件优化  | 4 736 432 990  | 105        |
| 存储器优化 | 2 214 278 264  | 50         |

提出了基于统计分析的浮点到定点转换策略和面向性能优化的片上存储器容量定制策略,并在此策略基础上提出完整设计流程,实现了在低成本 SoC 平台 MPEG-4 AAC 实时解码。结果表明了所提出的优化策略在性能、成本以及功耗方面的有效性。这一方法也适用于其他相似的 DSP 应用。

下一步工作需完善总线模型,为优化功耗提供更多信息。另外在今后芯片定义时即可根据应用目标确定最佳片上存储器容量,进一步降低成本。

### 参考文献

[1] Information Technology-coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 Mbit/s-Part 3: Audio [S]. ISO/IEC 11172-3, 1993

[2] Coding of Audio-visual Objects-parts 3: Audio [S]. ISO/IEC FDIS 14496-3, 2002

[3] Bosi - M, Brandenburg - K, Quackenbush - S, et al. ISO/IEC MPEG-2 advanced audio coding [J]. Journal of Audio Enginnering Society, 1997, 45(10): 789 ~ 814

[4] Tsai T H, Liu Chunnan, Wang Yiwen. A pure-ASIC design approach for MPEG-2 AAC audio decoder [A]. ICICS-PCM, Vol 3 [C], December 2003. 1633 ~ 1636

[5] Chen Jing, Tai Hengming. MPEG-2 AAC decoder on a fixed-point DSP [J]. IEEE Transactions on Consumer Electronics, 1999, 45(4): 1200 ~ 1205

[6] Mesarovic V, Hemkumr N D, Dokic M. MPEG-4 AAC audio decoding on a 24-bit fixed-point dual-DSP architecture [A]. The 2000 IEEE International Symposium on Circuits and Systems, Vol 3 [C]. 2000. 706 ~ 709

[7] Liu Chunnan, Tsai T H. SoC platform based design of MPEG-2/4 AAC audio decoder [A]. IEEE International Symposium on Circuits and Systems, Vol 3 [C]. 2005. 2851 ~ 2854

[8] Lee K S, Jeong N H, Bang K H, et al. A VLSI implement of MPEG-2 AAC decoder system [A]. IEEE Asia Pacific Conference on ASICs [C]. 1999. 139 ~ 142

[9] Bang K H, Kim J S, Jeong N H, et al. Design optimization of MPEG-2 AAC decoder [J]. IEEE Transactions on Consumer Electronics, 2001, 47(4): 895 ~ 903

[10] Lee - K S, Young C P, Dae H Y. Software optimization of the MPEG-audio decoder using a 32-bit MCU RISC processor [J]. IEEE Transactions on Consumer Electronics, 2002, 48(3): 671 ~ 676

[11] Wang Zhen. Design and optimization of instruction FIFO in the SDRAM controller of embedded microprocessor [J]. Journal of Circuits And Systems, 2005, 10(2): 138 ~ 141

[12] ARM Limited. ARM7TDMI Technical Reference Manual (Rev4) [EB/OL]. <http://www.arm.com>, 2001

[13] Klein R, Travilla K, Lyons M. Performance estimation of MPEG4 algorithms on ARM based designs using co-verification [A]. Proc Embedded Systems Conference [C]. 2002

[14] Rowson, J. Hardware/software co-simulation [A]. Proceeding of the Design Automation Conference [C], 1994. 439 ~ 440

[15] ARM Limited. Arm Developer Suite - Debug Target Guide Version 1.2 [EB/OL]. <http://www.amba.com>, 1999—2001

[16] Zhou Fan, Yang Jun, Ling Ming, et al. Design of fixed-point multimedia hardware accelerator in specific soc based on statistical analysis [A]. International Conference on Communications, Circuits and Systems, Vol 2 [C]. 2005. 1338 ~ 1342

[17] Duhamel P, Mahieux Y, Petit J P. A fast algorithm for the implementation of filter banks based on , (May): time domain aliasing cancellation [J]. ICASSP, 1991, (May): 2209 ~ 2212

[18] Pu Hanlai, Ling Ming. Performance oriented customization of on-chip memory capacity [J]. Journal of Applied Sciences, 2005, 23(4): 364 ~ 369

[19] Lee M, Tiwari V, Malik S, et al. Power analysis and minimization techniques for embedded DSP software [J]. IEEE Transactions on VLSI Systems, 1997, 5(1): 123 ~ 135

(下转第 79 页)