

面向 HDTV 应用的音频解码软硬件协同设计¹

汪 斌 姚庆栋 刘 鹏 张 明 韦晓东

(浙江大学信息与电子工程学系 杭州 310027)

摘 要: 该文以 Dolby 实验室的音频 AC3 算法为基础,研究了在 RISC 核 Virgo 上 HDTV 音频解码的软硬件协同设计方法,提出了通过对程序关键子函数建模来实现软硬件划分的软硬件协同设计方法。即在软件实现 AC3 解码的基础上,通过建立模型分析音频程序的关键操作的方法来扩展 RISC 的指令集,从而加快了音频解码速度,减少了存储空间,并在总体上减少了硬件开销。其次,该文给出了部分扩展指令的具体硬件结构。最后,通过软硬件协同评估的方法进行硬件改进后的软硬件代价分析。

关键词: 软 / 硬件协同设计, 音频, AC3, HDTV, RISC, MPEG-2

中图分类号: TP391.7, TN495 **文献标识码:** A **文章编号:** 1009-5896(2004)07-1082-06

The Hardware/Software Co-design of Audio Decoding Used in HDTV

Wang Bin Yao Qing-dong Liu Peng Zhang Ming Wei Xiao-dong

(Dept. of Info. Science and Electronic Eng., Zhejiang Univ., Hangzhou 310027, China)

Abstract In this paper, Dolby AC3 standard is used as the audio decoding standard to research the hardware/software co-design method of HDTV audio decoding based on Virgo-IP-Core and a new method is present to realize the hardware/software partition by setting models for the key sub-functions of AC3 decoding program. That is, after decoding AC3 bit stream by software, the models are set up for the key sub-functions to get the extended instructions of RISC core. Through this extension, the speed of AC3 decoding is fastened, the memory space is reduced and the gates of the whole system are lessened. Next, the detail hardware of extended instruction is given. Finally, the performance of the whole system is analyzed by using the hardware/software co-evaluation method.

Key words Hardware/software co-design, Audio, AC3, HDTV, RISC, MPEG-2

1 引言

在系统集成芯片中, RISC(Reduced Instruction-Set Computing)核成为最常用的部件,通常承担着基本的控制任务和一些算法任务。在 MPEG-2 MP@HL 集成解码芯片中,与 MIPS R3000 指令集兼容的 32 位嵌入式 RISC 核 - Virgo^[1] 承担着音频 AC3^[2](Advanced Compression 3) 解码和系统解码的双重任务。为了减少系统芯片的面积和功耗,并加快音频 AC3 解码,可以考虑在 Virgo 核中增加通用而又非常适合于 AC3 应用的指令。文献 [3] 中提出了一般系统的软硬件协同设计方法,即在确定系统性能要求的前提下,经软硬件划分、定位、调度、映射从而完成软硬件协同设计的方法。文献 [4, 5] 则对 CPU 核进行软硬件协同设计,通过增加或减少指令确定核的优化指令集。在以上工作的基础上,本文提出在确定初步的软硬件划分后,对软件进行建模分析并对系统进一步进行软硬件划分并提取出专用指令从而实现系统优化的软硬件协同设计方法。最后,采用了软硬件协同估计方法来考虑分析专用指令对系统的总的性能影响。

¹ 2003-02-21 收到, 2003-08-04 改回

国家自然科学基金(批准号 69972043)和国家高技术研究发展计划 863(No.2002AA1Z1140)资助课题

本文内容安排如下: 第 2 节介绍 AC3 解码系统的软硬件实现, 第 3 节对 AC3 软件中的关键运算符函数建立模型进行分析并提取出专用指令。第 4 节介绍专用指令的具体硬件实现。第 5 节用软硬件协同评估方法进行硬件改进后的软硬件代价分析。

2 AC3 解码的实现

MPEG-2 MP@HL 集成解码芯片主要完成系统解码, 视频解码和音频解码 3 个任务。系统芯片中含有 RISC 核负责系统层码流的解包和语义解释处理, 同时也对系统芯片中各功能部件以及数据进出进行管理。由于处理能力有富余, AC3 解码也安排在 RISC 核-Virgo 上以运行软件解码程序的方式来完成。系统 TS(Transport Stream) 流输入后, Virgo 核用 TS 解码程序解包给出音频 AC3 码流和视频码流, 音频码流通过总线送到 RISC 中解码, 解码后输出的 PCM(Pulse Code Modulation) 值通过总线送到音频回放单元回放。

如图 1 所示, AC3 码流输入后, 先经过误码检测和帧的解包等预处理, 然后经指数解码解出频率系数的指数信息, 再经比特指派计算出频率系数的尾数占用的比特数, 根据该比特数信息从码流中取出频率系数的尾数并进行反量化得出频率系数的尾数。结合频率系数的尾数和指数我们就可得到频率系数信息。对于耦合声道此时要进行声道的去耦合得到耦合声道的实际频率系数。最后通过综合滤波器组进行逆变换和加窗重叠运算将频域系数变换为时域系数得到最后的 PCM 值送到缓冲区播放。其中, AC3 数据帧的解格式化过程对应于 AC3 解码程序中的取帧头信息、取音频块信息函数和取尾数信息函数, 频谱包络的解码对应于指数解码函数, 比特指派对应于比特指派函数, 尾数的解量化对应于尾数解码函数, 综合滤波器组对应于逆变换 / 加窗重叠函数。

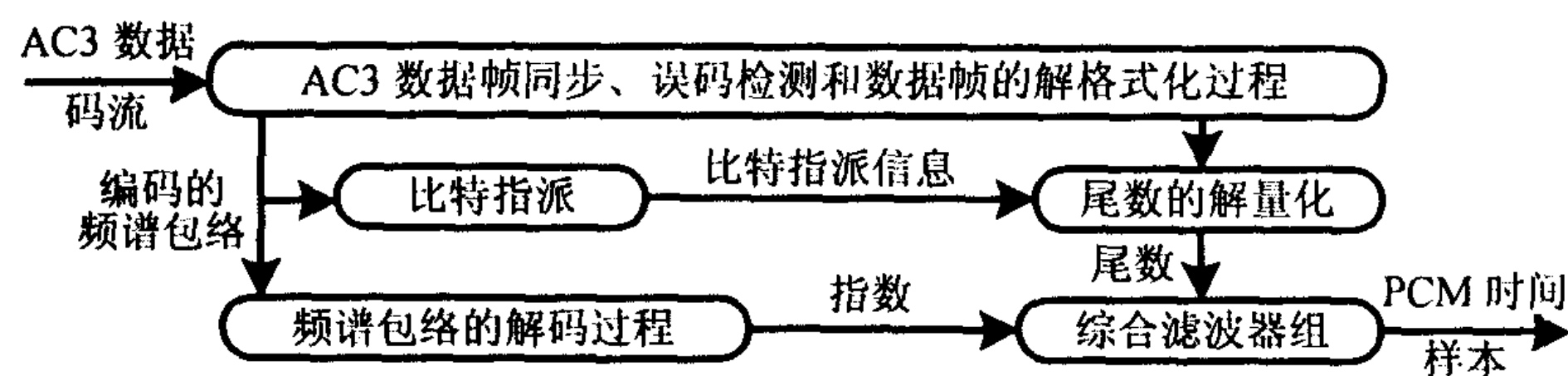


图 1 AC3 解码流程图

基于以上流程, 我们根据 AC3 标准编写了 AC3 解码的浮点 C 程序, 由于 Virgo 核是定点运算核, 所以必须在算法上把浮点运算 C 程序改变成定点运算 C 程序。然后将定点 C 程序转化为汇编程序, 在 SPIM 仿真器上仿真, 确定无误后用汇编器将其转化为机器代码在虚拟机上能够最终调试通过。

3 AC3 解码建模分析及专用指令提取

在文献 [3] 中提出的软硬件协同设计过程大致分为软硬件划分、定位、调度和映射 4 个步骤, 即在确定和开发了算法后, 将算法实现划分成小的模块, 然后根据各个子模块的特点将其定位到以在处理器和 DSP(Digital Signal Processor) 上运行软件的方式或直接以硬件方式实现。在确定了各个子模块的时间片的调度后将硬件单元映射成库单元, 将软件映射成可执行的二进制代码。文献 [4, 5] 研究了通过增加或减少核的指令集的方式进行进一步的软硬件划分的软硬件协同设计方法。在以上工作的基础上, 为了实现系统总体性能的优化, 针对 AC3 解码的特点, 可以考虑通过增加核的指令来进行进一步的软硬件划分, 将一部分软件的工作交给硬件实现。为了确定增加哪些指令可以使系统性能最优化, 必须对 AC3 程序进行分析来确定哪些操作在 AC3 解码程序中出现频率较高及能否用硬件实现。

根据以上分析, 首先统计 AC3 汇编程序中经常要用到的操作, 即提取其中运算密集型操作(占用较多指令运行周期), 然后提取出新的指令。AC3 解码程序运算周期分析如表 1。表中数据由解 5.1 声道, 比特率为 384kbps, 采样率为 48kHz 的一帧所得。

表 1 AC3 解码子函数运算周期百分比

解码函数	占用运算周期比例 (%)
取帧头信息 (Get_bsi_info)	0.1
取音频块信息 (Get_audblk_info)	0.9
指数解码 (Exponent_decode)	8
比特指派 (Bits_allocation)	38
取尾数信息 (Get_aud_mant_info)	19
尾数解码 (Mant_decode)	5
加窗重叠 (Inverse_overlap)	21
其他 (Others)	8

从表 1 可以看出：运算周期比例较高的是 bit_allocation(比特指派) 和 inverse_overlap(加窗, 重叠) 及 get_aud_mant_info(取尾数信息)3 部分。为了减少解出一个 AC3 帧的解码时间和保证音频实时解码, 从这 3 个函数中可提取出常用操作。进一步分析运算周期占用比例较高的 3 个函数, 如表 2 所示。比特指派函数可以划分为顺序运行的 7 个子步骤, 逆变换 / 加窗重叠函数可划分成顺序执行的 5 个子步骤, 而取尾数信息函数结构比较单一, 很难进行分割。为了方便给以上各函数的运算周期占用最多的部分建立运算模型, 对于结构较复杂的比特指派函数和加窗重叠函数可统计其中各子部分分别占用的运行周期所占的百分比如表 2 所示。表中数据由解 5.1 声道, 比特率为 384 kbps, 采样率为 48 kHz 的一帧所得。

表 2 比特指派函数和加窗重叠函数子步骤运算周期百分比

比特指派函数子步骤	运算周期百分比 (%)	逆变换 / 加窗, 重叠函数	运算周期百分比 (%)
初始化	1	前 FFT 复数相乘	15
指数映射为 PSD	6.7	FFT 变换	43
PSD 积分	27.9	后 FFT 复数相乘	12
计算激励函数	17.8	加窗、去交织	25
计算掩盖曲线	9.8	重叠、相加	5
增量比特指派	0.1		
计算比特指派	36.7		

从表 2 看出：比特指派函数中运算比例最高的是计算比特指派子步骤。加窗重叠函数中运算比例最高的是 FFT(Fast Fourier Transform) 变换, 但通过对 FFT 变换部分的模型分析发现很难在增加指令的基础上减少运算时间, 因此可对占用运算周期较多的加窗、去交织子步骤建模。根据以上分析, 可对比特指派函数中运算比例最高的计算比特指派子步骤和加窗重叠的加窗、去交织子步骤及单一的取尾数信息函数建模, 如图 2 所示。

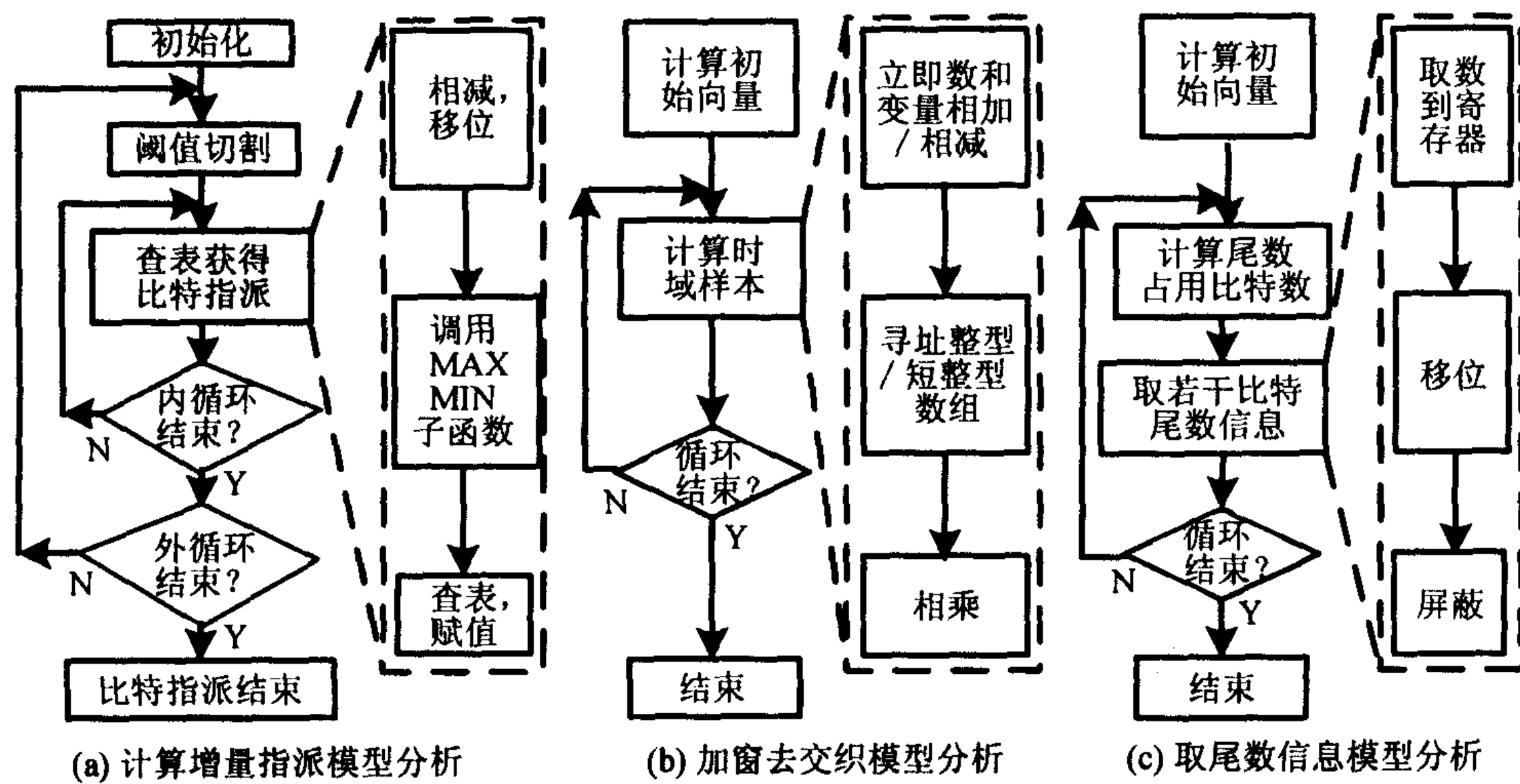


图 2 关键子步骤模型分析图

从图 2 可以看到, 计算增量指派子步骤的关键运算中有相减移位, 调用 MAX, MIN 子函数和查表赋值运算, 对于第一类和最后一类运算很难归纳出指令加以优化, 可以考虑将调用 MAX, MIN 子函数归纳成两条指令如下: $\max\ rd,rs,rt, \min\ rd,rs,rt$. 其中 \max 指令功能是将 rs 和 rt 寄存器值的较小数存入寄存器 rd 中, \min 指令功能是将 rs 和 rt 寄存器值的较大数存入寄存器 rd 中.

同样, 加窗去交织子步骤, 关键运算有立即数和变量的倍数相加 / 相减操作、寻址整型 / 短整型数组操作和相乘操作, 对于寻址整型和短整型数组, 由于 RISC 核是以字节寻址操作数, 所以对于整型数组寻址, 在计算数据地址时是将数组的索引右移两位后与数组基址相加; 同样, 对于短整型寻址, 计算数据地址是将数据的索引右移一位后与数据基址相加. 因此, 可以考虑增加指令如下: $\text{adddb}\ rd,rs,rt, \text{addff}\ rd,rs,rt$. adddb 指令功能是将 rs 寄存器值和 rt 寄存器值的两倍相加的结果存入寄存器 rd 中. addff 指令的功能是将 rs 寄存器值和 rt 寄存器值的四倍相加的结果存入寄存器 rd 中.

最后, 分析取尾数函数, 取若干比特尾数信息操作可分为取数到寄存器、移位、屏蔽 3 个子操作, 它们共同完成一个取若干比特码流的操作, 因此可将取比特信息操作归纳为指令: $\text{getbits}\ rd, \#imm, \text{getbitsv}\ rd,rs$. 其中 getbits 指令的功能是从码流中取出 imm 比特码流入入寄存器 rd 中. getbitsv 指令功能从码流中取出 $rs[3:0]$ 比特码流入入寄存器 rd 中.

以上分析可归纳成具体流程, 如图 3 所示: 首先分析 AC3 解码汇编程序, 通过统计运算量的方式分析出占用运算周期数比例最多的子函数, 然后对这些子函数进行建模, 通过建模的方式提取出新的指令, 同时在硬件上增加相应的硬件单元, 最后通过软硬件协同评估方法分析增加该指令是否能在总体上提高系统的性能. 在图 3 流程中, 虚拟机即为用 verilog 语言描述的与 MIPS 指令集兼容的 32 位嵌入式 RISC 核——Virgo. 用汇编器可将汇编程序编译成二进制代码, 将二进制代码加载在虚拟机的存储器中即可启动虚拟机仿真.

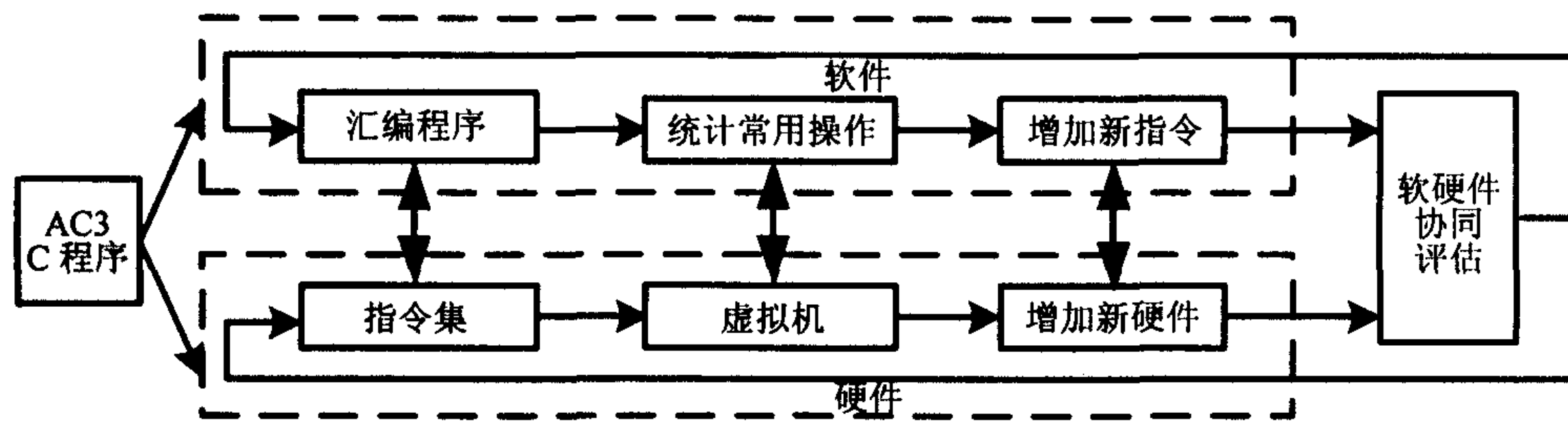


图 3 AC3 软硬件协同设计流程图

4 特殊指令硬件结构

取码流指令硬件框图如图 4 所示. 图中灰色填充的方块是寄存器, 其他模块是组合逻辑单元, 线段上的数字为数据位宽. Index 寄存器存放码流的索引地址, 即决定从一个 32 位字的哪一位开始取码流; Word pointer 寄存器存放码流的数据地址, 即 32 位码流字的存储地址; ID_imm 寄存器存放取码流的比特数, 由 imm 和 $rs[4:0]$ 在 RISC 的 ID(Instruction Decode) 级 (即 EX(EXecution) 级的前一个流水级) 经选择得到; getbits_op 即 getbits 指令的译码信号, 当 EX 级是 getbits 或 getbitsv 指令时此信号为高电平; Stream buffer 是码流缓冲寄存器, 当 MEM(MEMory) 流水级的 getbits_op 信号为高电平时, Cache 中取出的 32 比特数据 LD_dout 被送到 Stream buffer 中; Mask generate 是屏蔽向量产生单元, 根据输入的 5 位数据产生屏蔽向量.

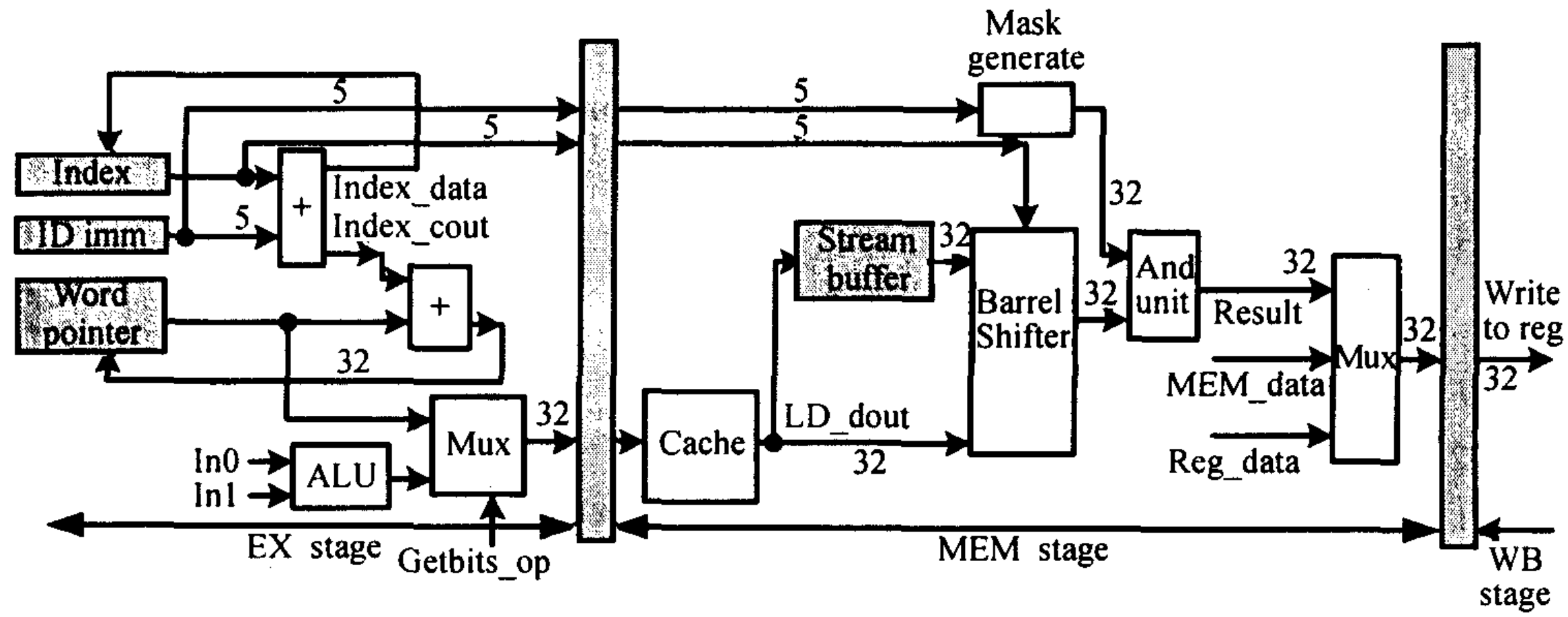


图 4 取码流指令硬件结构图

当系统复位时，Index 寄存器和 ID_imm 及 Word pointer 寄存器被清零。在第一次取码流之前，必须将码流的起始地址送到 Word pointer 寄存器中。取码流时，Word pointer 寄存器内容经 Mux 选择后作为存储地址寻址 Cache，取出的内容 LD_dout 作为低 32 位与 Stream buffer 的内容拼接成 64 比特数据再经桶型移位器移位，移位后的信号经过屏蔽信号屏蔽后即取出的码流信息 Result，最后 Result 经 Mux 选中后送到目的寄存器中。

其它指令的具体硬件结构限于篇幅不再细述。

5 面向 AC3 解码的 RISC 处理器性能评估

嵌入式 RISC 核是 MPEG 解码芯片的关键部件，在本设计中，RISC 处理器以 MIPS R3000 为基础。同时，为了提高面向 AC3 解码的 RISC 处理器的性能，我们扩展了该 RISC 的指令集，增加了若干 AC3 解码的专用指令。为了确保扩展的指令集能够从总体上提高系统的性能，我们采用了软硬件协同评估的方法。所谓软硬件协同评估即结合软件评估结果（如程序执行周期数或程序数据占用空间）和硬件评估结果（如硬件综合后占用的门数）进行协同评估得出最优方案的估计方法。

在实际中，软件评估上，我们用 Verilog 语言描述的虚拟机来运行扩展 MIPS 指令版本的汇编 AC3 程序得到程序运行周期数。在硬件方面，我们用 SYNOPSIS 的 Design compiler 综合扩展 RISC 处理器的 RTL 描述得到总门数。

表 3，4 所示是上述指令增加前和增加后的程序运行周期数和硬件占用门数对比。表中数据由解 5.1 声道，比特率为 384 kbps，采样率为 48 kHz 的一帧所得，综合所用是 UMC 0.18 μm 的标准单元库，门数以一个与非门为一个门计数。

表 3 添加指令前后减少程序运行周期数及其占子函数运算量百分比

扩展指令	未添加该指令运行周期数 (cycle)	添加该指令运行周期数 (cycle)	减少运行周期数 (cycle)	减少运行周期数占调用该指令的子函数运算量的百分比 (%)
Getbits/Getbitsv	3129960	3012823	117137	18.7
Max/Min	3129960	3105735	24225	5
Addb/Addff	3129960	3109942	20018	1

表 4 添加指令程序减少空间及指令硬件占用门数

扩展指令	未添加该指令程序占用空间 (byte)	添加该指令程序占用空间 (byte)	程序减少占用空间 (byte)	该指令硬件占用总门数 (gate)
Getbits/Getbitsv	20948	20288	660	1882
Max/Min	20948	20812	136	375
Addb/Addff	20948	20784	164	228

由表 3, 表 4 可以看出: 通过增加 2485 gate, 减少了 960 byte 程序空间, 一帧减少了 161380 cycle 运行周期数。若以一个比特存储器占用 1 gate 计算, 在增加了 2485 gate 并节省了 7680 gate 的存储空间, 即在节省 5195 gate 的情况下对于每帧减少了 161380 cycle 的运行周期数。

从以上协同评估可以看出: 我们增加了较少的硬件开销, 减少了程序占用存储空间, 较大幅度地减少了单位帧的指令运行周期。

6 结论

本文在 RISC 核 Virgo 上实现 AC3 音频解码的基础上, 扩展指令集增加特殊指令减少了 AC3 解码的时间和解码程序占用的空间。通过对 AC3 解码汇编程序及其在 Virgo 核上运行的结果进行分析提取出占用 CPU 运行时间较多的子函数, 再对这些子函数建立模型提取出关键操作并将其综合成特殊指令, 文中给出了部分指令的具体硬件实施框架和原理。最后, 通过软件上统计存储空间和运行时间, 硬件上对比添加指令硬件单元前后的综合门数的软硬件协同评估方法分析了添加指令后的系统性能。结果表明, 添加指令硬件单元减少了指令汇编程序占用空间和程序运行时间, 并从总体上减少了系统的硬件开销。

参 考 文 献

- [1] 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年会第 16 届年会 (ICCAS2001), 宁波, 2001.5: 352-356.
- [2] James C. McKinney, et al. 著, 严静兰, 乐陶, 译. ATSC 数字音频压缩 (AC3) 标准 (ATSC A/52). 电子部电视电声研究所, 北京: 广播电视出版社, 1995: 1-84.
- [3] Wolf W H. Hardware-software co-design of embedded systems. *Proc. IEEE*, 1994, 82(7): 967-988.
- [4] 杨伟建. 面向 HDTV 信源集成解码芯片的软硬件协同设计 [博士论文]. 杭州, 浙江大学, 2001, 第三章.
- [5] 杨伟建. 基于 RISC 核的 HDTV 系统层解码设计. *信号处理*, 2001, 17(3): 258-263.
- [6] Salapura V. A fuzzy RISC processor. *IEEE Trans. on Fuzzy Systems*, 2000, 8(6): 781-790.
- [7] Wantanabe H, Chen D. Evaluation of min/max instructions for fuzzy information processing. *IEEE Trans. on Fuzzy Systems*, 1996, 4(3): 369-374.

汪 斌: 男, 1976 年生, 博士生, 主要研究方向为音频编解码、集成电路设计、软硬件协同设计、计算机体系结构、信号处理等。

姚庆栋: 男, 1932 年生, 教授, 博士生导师, 主要研究方向为集成电路设计、系统软件、计算机体系结构、软硬件协同设计、并行处理和多媒体通信等。

刘 鹏: 男, 1970 年生, 副教授, 主要研究方向为集成电路设计、系统软件、软硬件协同设计等。

张 明: 男, 1962 年生, 教授, 博士生导师, 主要研究方向为集成电路设计, 计算机体系结构, 实时信号处理。