

基于 ITU-T G.723.1 标准语音处理芯片的 LSP 算法研究*

张常年, 夏日升, 姜岩峰

(北方工业大学 信息工程学院, 北京 100041)

摘要: 描述了 ITU-T G.723.1 算法的编码结构原理及实现, 介绍了基于该算法的芯片设计。重点对线性预测编码技术 LPC (Linear Prediction Coding) 中 LPC 系数转换为线谱对 LSP (Line Spectrum Pair) 系数算法进行了详细的推导分析, 并给出了用标准 C 的实现的代码及在芯片设计过程中对算法进行优化的方法。

关键词: 语音处理芯片 预测阶数 线性预测系数 线谱对 残差信号

ITU-T G.723.1 标准是 ITU-T 于 1996 年为多媒体通信制定的语音压缩标准, 可工作在 5.3kbps 或 6.3kbps 的两种信道速率下, 也可根据信道情况在两种速率之间切换, 其中 6.3kbps 速率的语音编码采用多脉冲最大似然量化(MP-MLQ)算法, 而 5.3kbps 速率的语音编码采用代数编码激励线性预测(ACELP)算法^[5]。

由于该标准在以上两种传输码率的情况下, 对数据压缩具有较好的语音质量和较高的压缩比(压缩率在 20 以上), 并且在算法实现上运算量较少, 易于实现, 所以被广泛用于视频会议、可视电话、语音存储、实时语音通信等需要低速率语音压缩的领域。

ITU-T G.723.1 语音编解码器是一种基于线性分析合成(Linear Prediction Analysis-by-Synthesis)模型的高效编解码器, 它采用了线性预测技术、分裂矢量量化技术、两级码本搜索等先进技术, 其压缩率达到 20 倍以上且保持了较高的语音质量。研究该算法以及芯片对该算法的实现不仅具有实际意义, 同时对理解其他语音处理算法和芯片的实现有重要的参考价值。本文介绍了基于该算法的芯片设计, 重点分析了线性预测技术中 LPC

系数转换为 LSP 系数的算法实现过程, 以及设计芯片对此算法的系统实现。

1 基于 G.723.1 算法的硬件系统芯片设计介绍

传统的实现算法都是使用 DSP 平台, 而使用 DSP 会使成本上升, 系统面积增大。由于 G.723.1 算法在实现时有较少的运算量, 且易于实现, 因此设计一个专用语音芯片能实时地处理该算法, 既可节省成本, 又减少了系统复杂度。

本文所用芯片是基于 C_CORE IP 设计的。它的特点如下:

- (1) 指令采用了 32 位的 C_CORE RISC 系统;
- (2) 采用高效的 4 级流水线执行指令, 使得多数指令能在单个时钟里得以执行;
- (3) 运行能力 31.4MIPS @ 33MHz, 1.8V, 13.6mW;
- (4) 全静态设计、动态电源管理和低电压操作等措施以减小功耗。

但是基于该核的运行能力还远不能达到本算法的实时处理要求, 这是因为在芯片运算单元中的乘法运算(32bit)分别需要 18、37 个时钟周期, 这样大大地减弱芯片对该算法的实时性处理。所以在该算法中需要专门

设计高效的乘除法运算单元。芯片系统时钟的设计采用外接晶振方案。芯片内指令存储器采用片内 ROM, 数据存储器的设计采用静态读写存储器(SRAM), 它的特点是不用经常充电刷新, 设计简单。而中断主要可以实现中断优先级排队、中断嵌套、自动响应中断和实现中断返回的功能。音频控制器用来实现数字采样的存储和传送。DMA 控制器可完成内部及外部装置的数据传送到内存中, 以及从内存中传送数据至外部装置。语音处理芯片的硬件框图如图 1, 其中虚线部分为片内部分^[7-9]。它对算法实现过程如下:

编码: 在采样率为 8kHz 的情况下, CODEC 采样语音数据, 音频控制器通过 C_CORE 处理器将数据暂存在音频控制器的输入 FIFO 中。然后, 由 DMA 控制器通过中断方式, 将数据从 FIFO 中传送到指定存储区域(SRAM)中。在 C_CORE 的控制下, 运行 G.723.1 编码程序, 对语音进行压缩。

解码: 运行 G.723.1 解码程序, 对压缩的语音进行解码。每解满 1 帧数据后, 由 DMA 控制器通过中断方式把数据传送到音频控制器的输出 FIFO, 再通过 CODEC 驱动放音设备(耳机、扬声器等)输出^[10]。

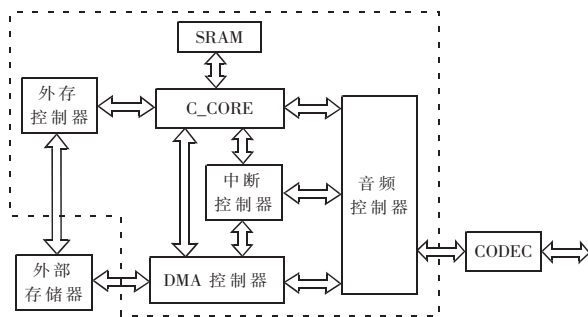


图 1 语音处理硬件框图

2 编码的结构原理

G.723.1 编码器是按帧对语音进行处理的, 在采样率 8kHz 情况下, 每帧 30ms, 即 240 个样点。编码器的结构原理如图 2。下面简单介绍各结构模块的功能及编码

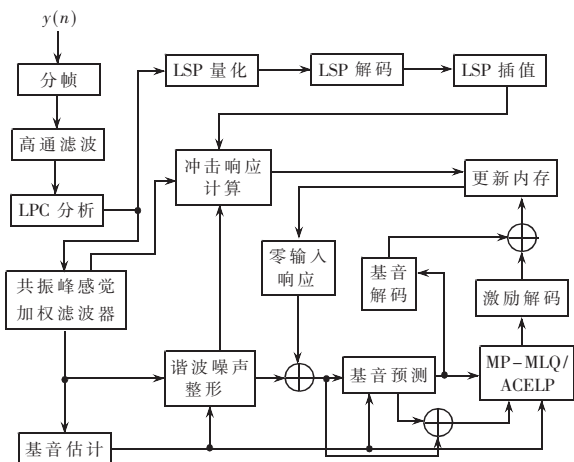


图 2 G.723.1 语音编解码器框图

的实现过程。

首先进行高通滤波, 去掉直流分量; 然后将一帧语音信号分成 4 个子帧, 每子帧为 60 个样点, 分别对每子帧用 10 阶线性预测编码(LPC)分析, 即先对子帧进行加窗, 用加窗后的样本求出它们的自相关系数, 根据均方误差最小准则, 利用 Durbin 算法即可求出各子帧的 LPC 系数。把最后一子帧的 LPC 参数转换成 LSP 系数, 进行矢量量化编码, 将矢量量化编码的标号传到解码器, 以便后面对语音的解码。利用未量化的 LPC 参数构造短时感觉加权滤波器, 对信号滤波后得到感觉加权的语音信号。对感觉加权滤波后的信号按每两子帧(前两个子帧和后两子帧)120 个样点分组, 然后用每组 120 样点来计算开环基音值, 并以此为依据构造一个谐波噪声形成滤波器, 对感觉加权滤波信号进行滤波。每一子帧的 LPC 综合滤波器、感觉加权滤波器和谐波噪声形成滤波器相级联构成一个联合滤波器。利用它的冲击响应和开环基音周期, 对每一子帧进行闭环基音搜索, 再对开环搜索的结果进行修正。同时通过一个五阶基音预测器对信号进行预测, 得到相应子帧的残差信号。最后进行固定码本搜索, 即对每一子帧的残差信号进行矢量量化; 求出脉冲位置和幅度参数, 打包后形成以帧为单位的编码数据流。这是 G723.1 算法的整个编码实现过程, 下面将详细介绍分析 LPC 系数转换为 LSP 系数的算法实现过程^{[11][6]}。

3 LPC 系数转换为 LSP 系数的算法

在量化语音短时谱包络信息时, 由于线性预测(LP)系数动态范围比较大而且对量化精度要求比较高, 不易保证合成滤波的稳定性, 因此 LP 系数不太适合直接作为量化的对象。而线谱对 LSP(Line-Spectrum Pair)也称线谱频率 LSF(Line-Spectrum Frequency), 是与线性预测系数和部分相关系数完全等价的另一种表示方式。但与线性预测系数相比, LSP 系数具有更好的内插特性和量化特性, 使得产生相同质量的合成语音所需的数码率得以降低。下面详细介绍 LPC 系数转为 LSP 系数的计算过程和算法。

(1) 根据 Levinson-Durbin 递推算法求 LPC 系数的递推公式为:

$$a_j^{(p)} = a_j^{(p-1)} - k_p a_{p-j}^{(p-1)}, \quad 1 \leq j \leq p-1 \quad (1)$$

其中 p 为线性滤波器的阶数, a_j 为所求的 LPC 系数。表示成矩阵形式为:

$$\begin{bmatrix} 1 \\ a_1^{(p)} \\ a_2^{(p)} \\ \vdots \\ a_{p-1}^{(p)} \\ a_p^{(p)} \end{bmatrix} = \begin{bmatrix} 1 \\ a_1^{(p-1)} \\ a_2^{(p-1)} \\ \vdots \\ a_{p-1}^{(p-1)} \\ 0 \end{bmatrix} - k_p \begin{bmatrix} 0 \\ a_{p-1}^{(p-1)} \\ a_{p-2}^{(p-1)} \\ \vdots \\ a_1^{(p-1)} \\ 1 \end{bmatrix} \quad (2)$$

将(2)式两边分别乘以 $[1 z^{-1} z^{-2} L z^{-p}]$,得到下式:

$$A_p(z) = A_{p-1}(z) - k_p z^{-p} A_{p-1}(z^{-1}) \quad (3)$$

式中:

$$A_p(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + L + a_p z^{-p} \quad (4)$$

$$A_p(z^{-1}) = 1 + a_1 z^1 + a_2 z^2 + L + a_p z^p$$

当预测阶数增加 1 时,下式成立:

$$A_{p+1}(z) = A_p(z) - k_{p+1} z^{-(p+1)} A_p(z^{-1}) \quad (5)$$

令 k_{p+1} 等于 +1 或 -1, 分别对应于声门完全张开或闭合的情形,用 $P(z)$ 和 $Q(z)$ 表示。由此得到:

$$P(z) = 1 + (a_1 - a_p)z^{-1} + (a_2 - a_{p-1})z^{-2} + L + (a_p - a_1)z^{-p} - z^{-(p+1)} \quad (6)$$

$$Q(z) = 1 + (a_1 + a_p)z^{-1} + (a_2 + a_{p-1})z^{-2} + L + (a_p + a_1)z^{-p} + z^{-(p+1)} \quad (7)$$

如果已知 $P(z)$ 和 $Q(z)$ 的根,就可以得到 $A_p(z)$,也即转化为求 $P(z)$ 和 $Q(z)$ 的解。

(2) 从式(6)和(7)中容易看出, $P(z)$ 和 $Q(z)$ 分别有一根 $z=1(w=0)$ 与 $z=-1(w=\pi)$ 。由于这两个根不带任何语音特征,去除两根后,得到下式:

$$\begin{aligned} F_1(z) &= P(z)/(1+z^{-1}) \\ F_2(z) &= Q(z)/(1-z^{-1}) \end{aligned} \quad (8)$$

并且每个多项式共有 5 对共轭根($e^{\pm jw_i}$)在单位圆上,因此两式又可表示为:

$$\begin{aligned} F_1(z) &= \prod_{i=1,3,5,7,9} (1 - 2q_i z^{-1} + z^{-2}) \\ F_2(z) &= \prod_{i=2,4,6,8,10} (1 - 2q_i z^{-1} + z^{-2}) \end{aligned} \quad (9)$$

其中 $q_i = \cos(w_i)$,且满足 $0 < w_1 < w_2 < L < w_{10} < \pi$,称 q_i 为余弦域的线谱对。由于 $F_1(z)$ 和 $F_2(z)$ 的对称性,所以仅须求出每个多项式的 5 个系数即可。多项式 $F_1(z)$ 和 $F_2(z)$ 的系数 $f_1(i)$ 和 $f_2(i)$ 有下列递推关系(for $i = 0$ to 4):

$$\begin{aligned} f_1(i+1) &= a_{i+1} + a_{m-i} - f_1(i) \\ f_2(i+1) &= a_{i+1} + a_{m-i} + f_2(i) \end{aligned} \quad (10)$$

其中, a_i 为已求出的 LPC 系数, m 为预测阶数,并且 $f_1(0) = f_2(0) = 1.0$ 。在求出 $f_1(i)$ 和 $f_2(i)$ 值后, $F_1(z)$ 和 $F_2(z)$ 的值用切比雪夫多项式计算,任一多项式在 $Z=e^{jw}$ 处的值可表示为:

$$F(w) = 2e^{-j5w} C(x) \quad (11)$$

其中 $C(x) = T_5(x) + f(1)T_4(x)Z + f(2)T_3(x) + f(3)T_2(x) + f(4)T_1(x) + f(5)/2$ 。

$T_m(x) = \cos(mx)$ 是 m 阶切比雪夫多项式^[2-4]。

(3) 将单位圆的上半圆进行 256 等分, x 取 256 等分后弧度 $i \times \pi/256 (i=0,1,\dots,255)$ 对应的余弦值,以此值带入 $F_1(z)$ 中,若相邻的两次值为同号,则继续取值,直到两点的值为异号(说明这两点之间存在根)。依次一直往下搜索,即可求得所有的根。同理按此方法搜索出 $F_2(z)$ 的根,然后分别计算反余弦值即可得到 LSP 参数^[1]。

4 C 代码及芯片设计的实现

从以上 LPC 系数转换算法分析过程中可看出,用 C 实现该算法主要分为两部分:

(1) 在利用 LPC 系数求 $F_1(z)$ 和 $F_2(z)$ 的系数 $f_1(i)$ 与 $f_2(i)$ 时,由于 $F_1(z)$ 和 $F_2(z)$ 的对称性,只需求出 5 个系数即可;

(2) 求出系数 $f_1(i)$ 和 $f_2(i)$ 后,在圆上搜索 $F_1(z)$ 和 $F_2(z)$ 的根时,要计算当 $x=i \times \pi/256 (i=0,1,\dots,255)$ 时, m 阶切比雪夫多项式 $T_m(x)$ 的值。

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_2(x) &= 2x^2 \\ T_3(x) &= 4x^2 - 3x \\ T_4(x) &= 8x^4 - 8x^2 + 1 \\ T_5(x) &= 16x^5 - 20x^3 + 5x \end{aligned} \quad (12)$$

可以先求出多项式的值并作为数组,直接调用。这样可以提高程序执行效率。由于该系数的转换是一个比较耗时的计算,除了采用上述较优 C 代码对程序优化外,同时还可从芯片的设计角度来提高算法的执行效率。

从上面系数转换算法可以看出,它的主要执行算法是乘加运算,而且反复调用。在芯片的 ALU 中笔者专门设计了针对该运算的乘加电路以提高它的执行速度。

本文详细介绍了 ITU-T G.723.1 标准线性预测技术中将 LPC 转化为 LSP 系数的算法及实现,并介绍了基于 C_CORE IP 核芯片对该语音算法的处理。同时从软件和硬件两方面对算法进行优化。由于线性预测技术在语音编码、语音合成、语音识别和说话人识别等语音处理领域中的广泛应用,了解该算法不仅有助于理解 ITU-T G.723.1 算法,同时对理解其它语音处理算法也有很大帮助。

参考文献

- [1] ITU-T Recommendation G.723.1 Dual Rate Speech Coder For Multimedia Communications Transmitting at 5.3 and 6.3kbit/s.
- [2] 张雄伟,陈亮,杨吉斌.现代语音处理技术及应用.北京:机械工业出版社,2003:76-96.
- [3] 王炳锡.语音编码.西安:西安电子科技大学出版社,2002:80-90.
- [4] 鲍长春.低比特率数字语音编码基础.北京:北京工业大学出版社,2001:74-106.
- [5] 张浩,官骏鸣,孙恩昌.ITU-T G.723.1 语音编码器中自适应码本搜索算法分析.无线通信技术,2005,(3):28-30.
- [6] 王家庆,吴并臻.G.723.1 语音编解码的 DSP 实时实现.中国数据通信,2005,(1):51-53.
- [7] 钟旭恒.8 位 RISC MCU 软核的设计.合肥工业大学硕士学位论文,2002.
- [8] 栾玉霞.基于 RISC 结构 MCU 的设计.西安电子科技大学硕士学位论文,2004.
- [9] 荆涛,彭艺频,杨军.基于 ARMSoC 的 G.729A 编码实现.电子工程师,2004,30(12):8-10.
- [10] 张兵,金晶,凌明.基于 ARM7TDMI 的 SoC 语音处理系统的设计. Microcontrollers & Embedded Systems, 2005, (10):44-47.

(收稿日期:2006-09-24)