

基于多核 DSP 处理器的插值和抽取滤波器的设计

熊鹏鹏

北京邮电大学电信工程学院, 北京 (100876)

E-mail: xiongpengpeng@tom.com

摘要: 插值和抽取滤波器被广泛应用于现代通信系统中, 然而基于传统 DSP 或者 FPGA 的滤波器, 具有数据率低和占用资源多的缺点。为了克服这些缺点, 本文针对一种多核 DSP 处理器, 提出了一种新的变速率滤波器设计方法。并且以 WiMAX802.16e 系统为例, 详细给出了方法的实现流程。实践证明本文中的算法能够很好的实现处理速度与占用资源的折衷。

关键词: 插值, 抽取, 多相滤波法, 多核 DSP 处理器, picoArray, AE

1. 引言

在实际的工作中, 经常会遇到抽样率转换的问题, 要求一个数字系统能工作在多抽样率 (multirate) 状态。例如, 对于同时具有语音、视频、数据等多种媒体的传输, 由于存在不同的频率成分, 所以该系统应该具有多种抽样率, 并自动完成抽样率的转换; 又如, 为了减少抽样率太高造成的数据的冗余, 而需要降低抽样率; 再如信号在具有不同时钟频率的系统之间传输时, 为了便于信号的处理、编码、传输和存储, 则要求根据时钟频率对信号抽样率加以转换^[1]。如今, 建立在抽样率转换理论基础上的多抽样率数字信号处理已经成为数字信号处理这门学科中的一项重要内容^[2]。

2. 多相分解

抽样率的转换有多种方法可以实现, 其中采用数字滤波器的方法是最直接也是最合理的方法。由于 FIR (有限冲激响应) 滤波器的参数非常对称, 其基本的滤波操作是移位和乘累加操作, 其中卷积部分的运算量占整个运算量的绝大部分。因此为了减少运算量, 提高运算速度, 首先介绍一下多相分解的滤波器设计方法^[3]。

将一个序列表示成 M 组子序列的叠加, 其中每一组都由该序列中每隔 M 个一次延迟的序列值所组成, 这就得到了一个序列的多相分解。多相分解后, 插值和抽取子滤波器的个数等于插值或抽取的倍数。具体的说, 考虑某一冲激响应 $h(n)$, 将其分解成 M 组子序列 $h_k(n)$ 如下:

$$h_k(n) = \begin{cases} h(n+k) & n = M \text{ 的整倍数} \\ 0 & \text{其他} \end{cases}$$

将这些子序列依次延迟相加就能恢复原冲激响应 $h(n)$, 即

$$h(n) = \sum_{k=0}^{M-1} h_k(n-k)$$

抽取滤波器的多相实现框图和插值滤波器的多相实现框图分别如图 1 和图 2 所示。从图中可以看出, 目标滤波器被分解为若干子滤波器的并联。

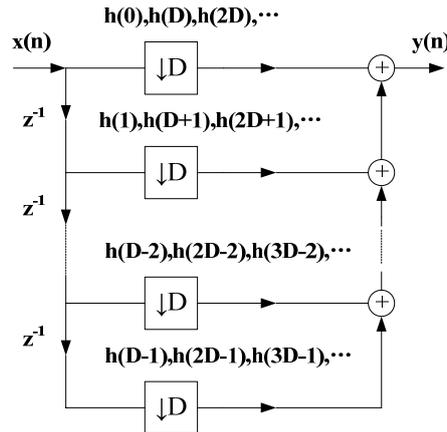


图 1: 抽取多相滤波器框图

对于抽取多相滤波器，最后的输出序列 $y(n)$ 是 D 路子滤波器输出的叠加。图 1 中的 D 代表一个抽取子滤波器。

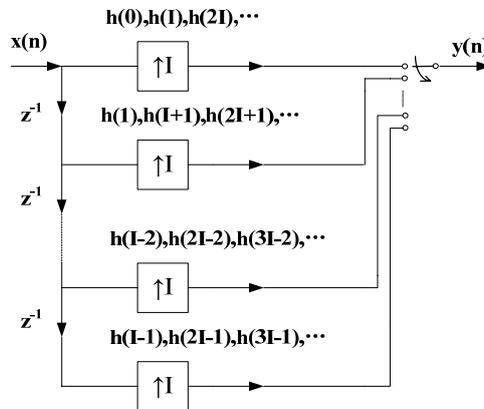


图 2: 插值多相滤波器框图

对于插值多相滤波器，最后的输出序列 $y(n)$ 依次是第 I 路子滤波器输出。即每次仅有一路子滤波器的输出作为最后的输出。图 2 中的 I 代表一个插值子滤波器。

3. 硬件设计

3.1 多核 DSP 处理器简介

所谓多核处理器是指基于单个半导体的一个处理器上拥有两个或多个一样功能的处理器核心。换句话说，将两个或多个物理处理器核心整合入一个核中。多核处理器技术的引入是提高处理器性能的有效方法。因为处理器实际性能是处理器在每个时钟周期内所能处理指令数的总量，因此增加一个内核，处理器每个时钟周期内可执行的指令数将增加一倍。

本文提到的多核处理器芯片不同于 Intel、AMD、IBM 等芯片厂商推出的两核或四核处理器芯片，而是指具有数十、数百乃至数千个核的同构或异构多核处理器。由于芯片数量众多，使得各个小处理器之间可以并行运算，各个小处理器则实现整个系统的不同功能模块。目前市场上常见的多核处理器芯片主要有英国 PicoChip 公司的 picoArray 系列芯片，德国 Connex Technology 公司的 CA1024 芯片，美国 Rapport 公司的 1025 核心 CPU 等等。本文所提的插值和抽取滤波器的设计与实现正是基于 PicoChip 公司的 picoArray 芯片。

3.2 picoArray 简介

picoArray 是一种大规模并行的、多重指令多重数据 (MIMD) 架构，该架构是由获得专利的 picoBus 内连技术连接在一起的多种类型的处理单元构成的。picoArray 是一种粗粒度的超大规模并行异构 16 位处理器阵列，其运算和通信资源是静态分配的。与 TI 和 Freescale 的多核 DSP 采用高达 1GHz 内核不同的是，picoArray 含有 300 多个处理单元，每个处理单元都是一个功能强大的 16 位器件。由于每个相互连接的处理单元都能并行运算，picoArray 提供了超强的处理能力，在 160MHz 的主频下能提供 100GIPS (1GIPS=每秒 10 亿指令) 的性能。

图 3 简单地显示了 picoArray 的整体架构^[4]。图中每个被标记为 P 的方块表示一个处理器，这个处理器被称作“阵列单元” (Array Element) 或简称“阵元” (AE)。AE 分为以下三种不同的类型：标准类型 (STAN)，控制类型 (CTRL) 和存储类型 (MEM)。对于不同类型的 AE，除了支持的端口数和数据存储空间不同之外，还有不同的指令集可以使用。

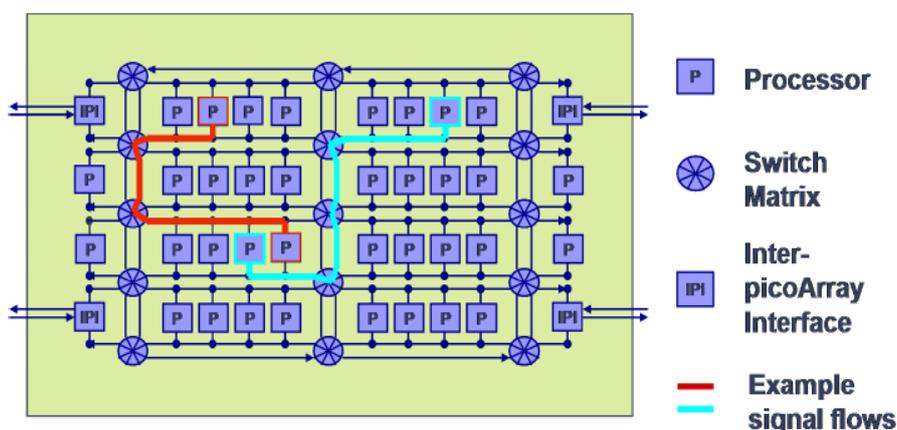


图 3: 简化的 picoArray 的整体架构

3.3 插值滤波器的设计方法

具体实现的时候，除了要考虑滤波器的功能实现外，还需要考虑速度以及资源的要求。为了达到 802.16e 协议中给定的滤波器设计指标，分别设计了 2 倍、4 倍和 5 倍插值与抽取滤波器。不同倍速的插值滤波器使用了不同的设计方法，而抽取滤波器的设计方法则完全相同。因为所有倍速的插值滤波器都使用了多相滤波法，因此首先给出多相滤波法设计插值滤波器的原理图。然后再分别讨论各种倍速的插值滤波器所综合使用的其他设计方法。

以一个 15 阶的 4 倍插值滤波器为例，图 4 中仅给出了第一组子滤波器的抽头系数的计算流程。从图中可以看出，该组抽头系数就如同同一把梳子，计算得到输出 y_n ，而 $y_{n+1}, y_{n+2}, y_{n+3}$ 则由其他三组抽头系数计算得到。当四组抽头系数都计算完一遍后，各组抽头系数向右移一个单元格，计算得到输出 $y_{n+4}, y_{n+5}, y_{n+6}, y_{n+7}$ ，依此类推。图中 x 序列表示输入信号， c 序列表示抽头系数， y 表示输出信号。

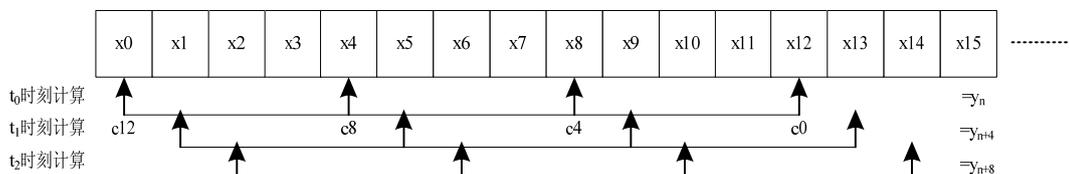


图 4 插值滤波器的原理框图 (以 4 倍插值滤波器为例)

图 4: 插值滤波器的原理框图 (以 4 倍插值滤波器为例)

3.3.1.2 倍插值滤波器

为了满足 FIR 滤波器的线性相位的特性, 滤波器的抽头系数都设计为对称的, 同时为了提高 2 倍插值滤波器的运算速率, 综合采用了对称法和多相滤波法来设计。为此 2 倍插值滤波器设计为奇数个抽头系数, 这样分到两条支路上去的两组子滤波器抽头系数的值也刚好是对称的。这样将对应于同一个滤波器抽头系数值的两个输入先做加法, 再与滤波器的抽头相乘, 减少了乘累加的次数, 相应的提高了运算速度。对于每个 AE 中的两个乘累加运算单元 (acc 单元, STAN 类型的 AE 有两个 40 位的乘累加器, 分别为 acc0 和 acc1), 可以采用交替运算的方式, acc0 计算第一组抽头系数, acc1 计算第二组抽头系数。图 5 给出 2 倍插值滤波器的设计框图。

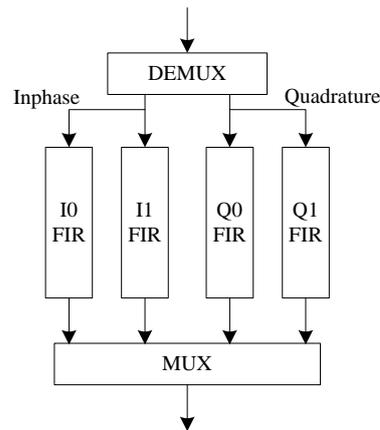


图 5: 2 倍插值滤波器的设计框图

从图 5 可以看出, 具体实现的时候需要将同相分量 (Inphase) 和正交分量 (Quadrature) 进行分离 (即图 5 中的 DEMUX 部分), 即后面的滤波操作都是分别对同相分量和正交分量进行操作。

由于 PC102 芯片的主频是 160MHz, 对于 WiMAX802.16e 系统输出采样率 22.4Msample/s, 因此平均到每个 sample 的指令周期 (cycle) 数为: $\frac{160}{22.4} = 7.143$ 。对于所

设计的 48 阶 2 倍插值滤波器来说, 计算每个 sample 需要 25 次乘累加和 1 次跳转, 根据 picoArray 的指令周期, 总共需要 26 个 cycles。因此为了满足运算速度的要求, 必须设计的

并行计算的支路数为: $\left\lceil \frac{26}{7.143} \right\rceil = 4$ (式中 $\lceil \cdot \rceil$ 表示向上取整), 两条支路用于计算同相分

量, 两条支路用于正交分量。以同相分量来举例, 对于第一个子滤波器 (图 5 中的 I0 FIR) 从第一个输入开始计算, 第二个子滤波器 (图 5 中的 I1 FIR) 从第二个输入开始计算, 每个子滤波器都要计算两组滤波器的抽头, 对于正交分量也是如此, 其子滤波器的计算流程如图 6 所示。图中 x 序列表示输入信号, c 序列表示抽头系数, y 表示输出信号。

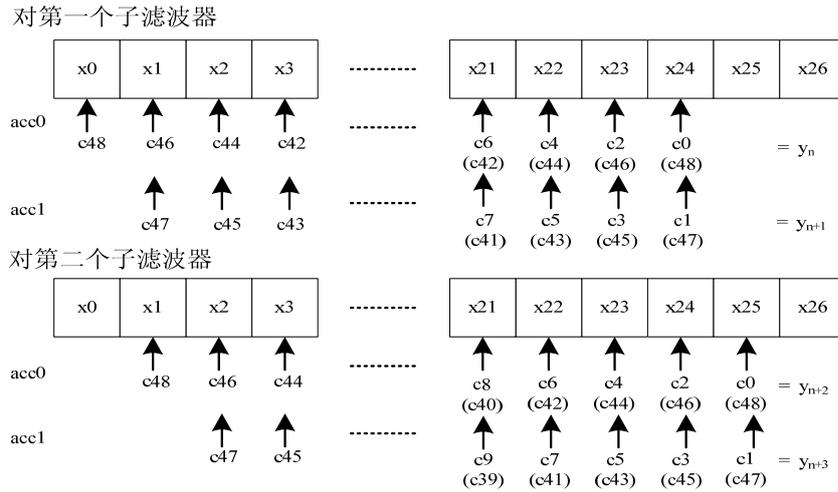


图 6: 2 倍插值滤波器子滤波器计算流程图

最后 4 个子滤波器 (I0、I1、Q0、Q1) 的输出都输入到一个完成合并功能的 AE 中 (图 5 中的 MUX 部分)。在该 AE 中, 将同相分量求和的正交分量进行合并, 得到经过 2 倍插值滤波器后的最终输出。

3.3.2.4 倍插值滤波器

对于 4 倍插值滤波器, 采用多相滤波的设计方法。由于滤波器的抽头系数必须分到 4 组子滤波器中, 虽然总的滤波器抽头具有对称性, 但分到每个子滤波器之后, 对称性消失, 所以不能采用设计 2 倍插值滤波器所用的方法, 即不能采用对称法来实现滤波器。对于每个 AE 中的两个乘累加运算单元, acc0 计算第一组和第三组抽头系数, acc1 计算第二组和第四组抽头系数。这样在一个 AE 中完成了所有滤波器抽头系数的乘累加运算。

此外为了满足运算速度的要求, 需要设计三路并行计算的子滤波器来达到要求。具体的运算流程和设计框图与 2 倍插值滤波器的方法相同, 不再赘述。

3.3.2.5 倍插值滤波器

对于 5 倍插值滤波器, 由于滤波器的阶数较高, 相应的抽头系数较多, 而每个 AE 的代码空间是有限的。为了不浪费每个 AE 的运算资源同时满足代码空间的要求, 在采用多相滤波法的同时将滤波器的抽头系数进行分段, 每个 AE 只计算抽头系数的一部分, 共需 3 个 AE 来完成。3 个 AE 通过级联的方式来实现最终的滤波器, 这样每个 AE 仅仅得到一部分的卷积结果, 最后需要将各个 AE 对应的卷积结果进行相加才是最终的输出。需要注意的是: 对于一个 40 位的乘累加器来说, 从 acc 中读取出来的结果只有其中的 32 位, 因此, 采用滤波器级联的方式计算将会带来额外的运算精度损失。

具体实现的时候, 滤波器的阶数定为 104 阶。多相分解后每个子滤波器的抽头系数为 21 个。为了满足运算速度的要求, 也需要设计成三路并行计算的子滤波器, 这三路子滤波器分别计算经多相分解后的一部分抽头系数。对于第一个子滤波器, 依次计算五组子滤波器抽头中每组的前面 7 个抽头系数, 对于第二个子滤波器, 依次计算五组子滤波器抽头中每组的中间 7 个抽头系数, 对于第三个子滤波器, 依次计算五组子滤波器抽头中每组的最后 7 个抽头系数, 这样对于每个子滤波器来说, 分别只计算了每组子滤波器抽头的其中 7 个抽头, 这样在后面的合并 AE 中, 除了要完成 IQ 合并的功能外, 还需要将每个子滤波器的输出结果进行相加, 才能得到最后的经过每组子滤波器抽头的结果。另外对于 5 倍插值滤波器, 计

算前面 4 组滤波器抽头的时候可以交替使用 acc0 和 acc1，但对于最后一组滤波器抽头，仅仅只需要用到 acc0 即可。具体的运算流程如图 7 所示。图中 x 序列表示输入信号， c 序列表示抽头系数， y 表示输出信号。

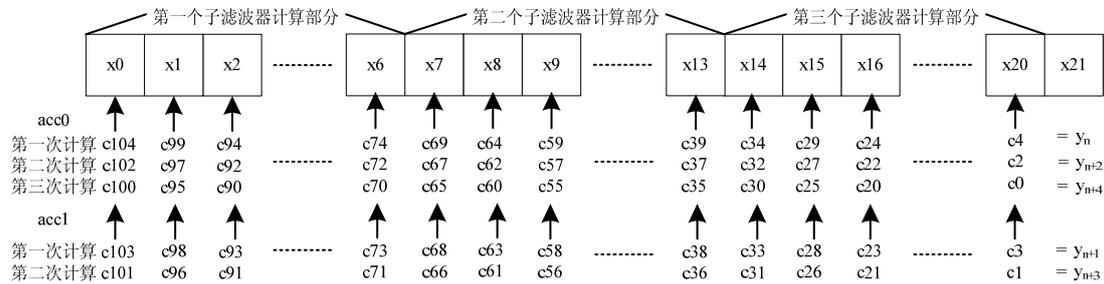


图 7: 5 倍插值滤波器子滤波器计算流程图

3.4 抽取滤波器的设计方法

对于抽取滤波器的实现，各倍速抽取滤波器的设计方法完全相同，仅仅是滤波器抽头系数的个数不同。采用了对称法的设计方法来达到设计要求。为了减少乘累加的次数，利用抽头系数的对称性将对应相同抽头系数的输入先作加法，然后再与抽头系数相乘，这样可以使乘累加的次数减少为原来的一半。给出抽取滤波器的设计框图如图 8 所示。

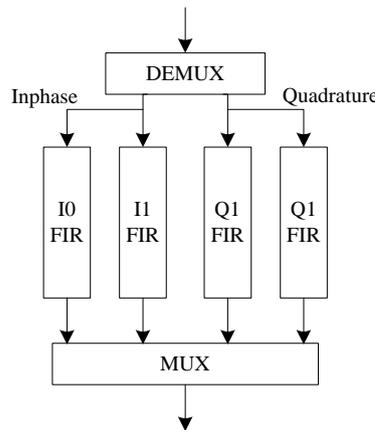


图 8: 抽取滤波器的设计框图

具体实现时，与插值滤波器相同，对于抽取滤波器，也需要将同相分量和正交分量进行分离，即后面的滤波操作都是单独分别对同相分量和正交分量进行操作。

对于抽取滤波器的设计，为了满足运算速度的需求，需要设计成两路并行计算的子滤波器的结构。另外不像插值滤波器那样将滤波器的抽头分为若干组，而是对于所有的抽头系数都要完成一遍乘累加操作。以 2 倍抽取滤波器为例：对于第一个子滤波器，从第一个输入开始进行乘累加运算，而对于第二个子滤波器，需要偏移抽取因子个数（对于 2 倍抽取滤波器来说，该值为 2）的输入开始计算进行乘累加运算。其他倍速的抽取滤波器设计方法相同，仅仅是偏移的个数不同。具体的计算流程如图 9 所示。图中 x 序列表示输入信号， c 序列表示抽头系数， y 表示输出信号。

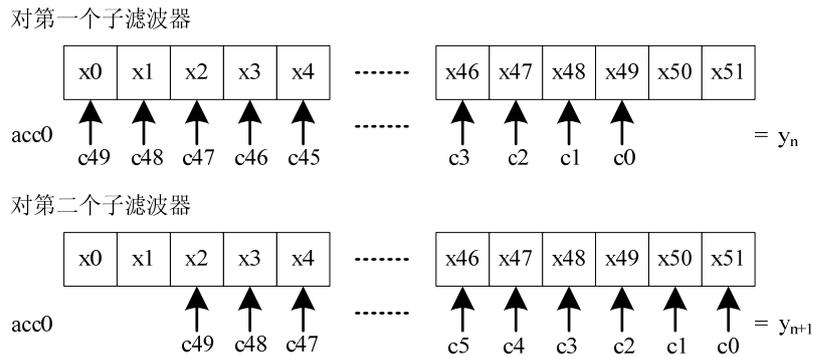


图 9: 2 倍抽取滤波器子滤波器计算流程图

4. 设计方法总结

基于多核 DSP 芯片上设计插值和抽取滤波器，设计方法的选择主要体现在插值滤波器的设计上，而对于抽取滤波器的设计均采用对称法进行。所以在这里仅给出插值滤波器的设计方法。

比较前面给出的三种不同倍速的插值滤波器的设计方法，总结如表 1 所示。从表 1 中可以看出，三种插值滤波器都采用了多相滤波法。另外根据多相分解后子滤波器的抽头系数的对称性，对于 2 倍插值滤波器还综合使用了对称法，提高运算速率。同时由于滤波器抽头个数较多，不能在一个 AE 中完成对所有抽头系数的乘累加运算，因此对于 5 倍插值滤波器综合使用了级联法。

表 1: 各种倍数插值滤波器设计方法总结

不同倍速的插值滤波器	设计方法
2 倍插值滤波器	对称法、多相滤波法
4 倍插值滤波器	多相滤波法
5 倍插值滤波器	多相滤波法、级联法

综合以上特点，给出在多核处理器上插值滤波器的设计流程图如图 10 所示。

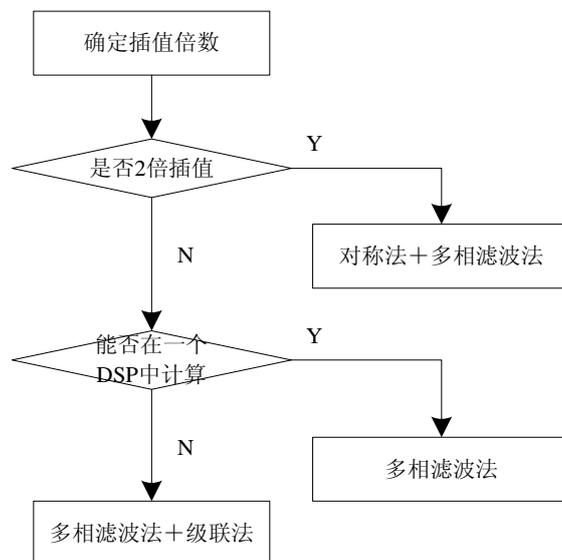


图 10: 插值滤波器设计流程图

5. 结束语

本文给出了基于多核 DSP 处理器芯片上常用的几种插值和抽取滤波器的设计方法，并按照 WiMAX802.16e 协议要求给出了 2 倍、4 倍、5 倍插值和抽取滤波器的设计方法和实现框图。在综合采用了多种滤波器的设计方法下，硬件程序很好的实现了插值和抽取滤波器的功能，并满足了 WiMAX802.16e 系统中对于插值和抽取滤波器在性能和运算速度方面的要求。另外通过对给出的不同倍速的插值滤波器的设计的对比归纳，总结出一种通用的在多核 DSP 处理器上实现各种倍速插值滤波器的设计流程图。

参考文献

- [1] 程佩青. 《数字信号处理教程（第二版）》[M]. 北京：清华大学出版社. 2001
- [2] 胡广书. 《数字信号处理 理论、算法与实现（第二版）》[M]. 北京：清华大学出版社. 2003
- [3] Alan. V. Oppenheim, Ronald W. Schaffer, John R. Buck . 《离散时间信号处理（第二版）》[M]. 刘树棠, 黄建国译: 西安交通大学出版社. 2001
- [4] picoChip Designs Ltd. picoTools Documentation. 2005

Design of Interpolation and Decimation Filters on Multi-core DSP Processor

Xiong Pengpeng

School of Telecommunication Engineering, BUPT, Beijing, PRC (100876)

Abstract

Interpolation and decimation filters are widely used in modern communication system. But the filters based on traditional DSP and FPGA have the disadvantages of low data rate and consuming more resources. In order to get over these disadvantages, this paper proposes a new implementation method for rate variable filter. Then it takes the filters in WiMAX802.16e system as an example, give the flowchart of the implantation in detail. Finally it turns out that the new method makes a good tradeoff between processing rate and resource consumption.

Keywords: interpolation, decimation, polyphase filter, multi-core DSP processor, picoArray, AE

作者简介: 熊鹏鹏, 男, 1983 年生, 硕士研究生, 信号与信息处理专业, 研究方向为移动 WiMAX 系统的物理层关键技术及实现。