

# 支持任意系数长度和数据类型的 FIR 滤波器向量化方法

刘 仲,陈跃跃,陈海燕

(国防科学技术大学计算机学院,湖南长沙 410073)

**摘 要:** 向量处理器的向量化算法映射是难点问题.提出一种支持任意系数长度和数据类型的 FIR 滤波器向量化方法,将(Finite Impulse Response)滤波器的卷积计算划分为系数长度步向量乘法和加法计算,每一步的向量乘法和加法计算在各个向量处理单元上并行执行,计算一个输出结果的所有乘法和加法计算都在同一向量处理单元上完成,每次循环能够同时完成向量处理单元数量个输出结果的计算.在向量处理器 YHFT-Matrix 的实验结果表明,该向量化 FIR 滤波器能够取得高效的计算性能和加速比.

**关键词:** FIR 滤波器; 向量化; 向量处理器

**中图分类号:** TP391.4

**文献标识码:** A

**文章编号:** 0372-2112 (2013)02-0346-06

**电子学报 URL:** <http://www.ejournal.org.cn>

**DOI:** 10.3969/j.issn.0372-2112.2013.02.022

## A Vectorization of FIR Filter Supporting Arbitrary Coefficients Length and Data Types

LIU Zhong, CHEN Yue-yue, CHEN Hai-yan

(School of Computer, National University of Defense Technology, Changsha, Hunan 410073, China)

**Abstract:** The Vectorization of algorithm mapping for vector processors is a critical issue. This paper presents a vectorization of FIR filter supporting any different coefficients length and data types, in which the FIR filter computation is divided into N-step (length of coefficients) vector multiply and accumulate operations, vector multiply and accumulate operations are executed in parallel by all vector processing elements (VPEs), and all vector multiply and accumulate operations to calculate single output are completed at the same VPE, the number of VPEs outputs will be calculated at once loop. Experimental results on YHFT-Matrix show that the vectorization of FIR filter achieves very efficient performance and speedup.

**Key words:** finite impulse response (FIR) filter; vectorization; vector processor

### 1 引言

随着 LTE 和 4G 高速无线通信、高清视频编解码、实时图像搜索、雷达和水声信号处理等各种高计算密集型应用的不断涌现,单核处理器芯片的性能越来越难以满足这类应用实时、高密度的计算需求,驱动着高性能的多核处理器和向量处理器快速发展.多核处理器和向量处理器给软件设计提出了许多新的研究课题,现有的大量 DSP 算法是基于单核处理器设计的,如何针对多核和向量处理器体系结构特点,充分开发各个层次的并行性,高效地并行和向量化这些应用算法是当前面临的主要困难<sup>[1]</sup>.

FIR (Finite Impulse Response) 滤波器是移动通信、视

频音频编解码、图像处理、水声和雷达信号处理等各类应用中的基本核心算法之一<sup>[2,3]</sup>.面向向量处理器的体系结构特点,提出高效的 FIR 滤波器向量化映射方法是当前向量处理器应用要解决的一个重要问题. Mirzaei<sup>[4]</sup>提出的基于 FPGA 的并行化 FIR 实现方法不适合向量处理器, Kutil<sup>[5-7]</sup>等人研究了基于 SIMD 的 FIR 并行化方法,没有考虑向量处理器和各种数据类型. Dang<sup>[8]</sup>研究了面向 Philips 公司开发的向量处理器的 FIR 向量化方法,但滤波器系数受限制,且只适用于 16 位定点数据,对 32 位浮点和复数不支持.本文提出一种支持任意系数长度和数据类型的 FIR 滤波器向量化方法,能够适用于任意系数长度的 FIR 滤波器,不受限于向量处理器是否支持加法归约,支持 8 位、16 位、32 位定点实数和

复数、32 位浮点实数和 32 位浮点复数各种数据类型.在 YHFT-Matrix 上进行的具体实现与评估表明,它是一种面向向量处理器的高效、通用 FIR 滤波器向量化方法.

## 2 向量处理器 YHFT-Matrix 的体系结构

如图 1 所示, YHFT-Matrix 是 VLIW (Very Long Instruction Word) 体系结构, 面向高密度计算应用的高性能向量处理器. 它包括标量处理部件 (Scalar Processing Unit, SPU) 和向量处理部件 (Vector Processing Unit, VPU), SPU 负责标量任务计算和流控, SPU 和 VPU 可通过共享寄存器交换数据. YHFT-Matrix 每时钟周期发射 10 条指令, 包括 5 条标量指令和 5 条向量指令. 指令派发单元对执行包进行识别, 并将其中的指令派发到相应的功能单元中执行.

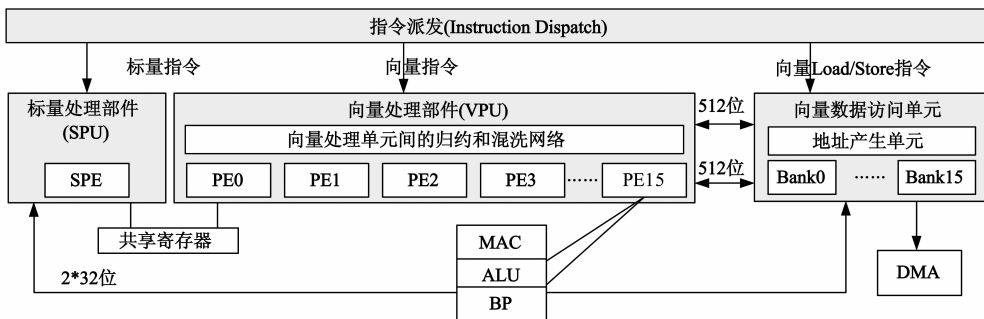


图1 YHFT-Matrix的体系结构

## 3 FIR 滤波器的向量化

### 3.1 向量化方法

FIR 滤波器是输入信号流  $x_k$  与冲击响应  $h_k$  的离散卷积, 一个系数长度为  $N$  的 FIR 滤波器计算公式如下:

$$y_n = \sum_{k=0}^{N-1} h_k x_{n-k} \quad (1)$$

其中  $h_k (0 \leq k \leq N-1)$  是滤波器系数,  $x_k$  是输入信号. 一般来说, 滤波器系数  $h_k$  是预先确定的, 若按倒序存放, 则公式(1)变换为更直观的形式如下:

$$y_n = \sum_{k=0}^{N-1} h_k x_{n+k} \quad (2)$$

用  $Y$  和  $H$  分别表示输出和系数向量,  $X$  是输入矩阵, 则公式(2)可用矩阵形式表达如下:

$$Y = X * H \quad (3)$$

若连续  $S$  个输入信号  $x$ , 计算系数长度为  $N$  的 FIR 滤波器的  $S$  个连续输出, 则公式(3)可表达为如下的矩阵形式:

$$\begin{bmatrix} y_n \\ y_{n+1} \\ \vdots \\ y_{n+S-1} \end{bmatrix} = \begin{bmatrix} x_n & x_{n+1} & \cdots & x_{n+N-1} \\ x_{n+1} & x_{n+2} & \cdots & x_{n+N} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n+S-1} & x_{n+S} & \cdots & x_{n+S+N-2} \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{N-1} \end{bmatrix} \quad (4)$$

VPU 负责向量计算, 包括 16 个向量处理单元 (Vector Processing Element, VPE), 每个 VPE 含一个局部寄存器文件, 4 个 40 位的累加器, 以及 ALU、MAC、BP 3 个并行功能部件. 局部寄存器文件包含 28 个 32 位寄存器, 所有 VPE 的同一编号的局部寄存器在逻辑上又组成一个 512 位的向量寄存器. 3 个功能部件支持定点和浮点操作, 向量指令在各个 VPE 上同时独立运行. 所有 VPE 共同作用的指令有两种: 归约 (Reduction) 指令对 16 个 VPE 的同一编号的局部寄存器进行求和、求最大值和求最小值操作; 混洗 (Shuffle) 指令对 16 个 VPE 的同一编号的局部寄存器的数据进行字节、半字和字三种模式的数据交换操作. 向量数据访问单元支持向量数据的 Load/Store, 提供 1MB 的大容量专用向量存储器, 每周同时支持 2 个 Load/Store 指令.

从公式(4)可以看出, 连续输入信号的 FIR 滤波器计算包括两层循环: 外层循环是以输入信号长度  $S$  为计数器, 内层循环是以系数长度  $N$  为计数器. 通常的向量化方法是向量化内循环, 即计算每个输出值所需要的内循环计算向量化, 分配到各个 VPE 并行执行, 用公式(5)表达如下:

$$y_n = x_n h_0 + x_{n+1} h_1 + \cdots + x_{n+N-1} h_{N-1} \quad (5)$$

公式(5)中的每一项  $x_n h_0, x_{n+1} h_1, \cdots, x_{n+N-1} h_{N-1}$  分别分配到每个 VPE 并行执行, 执行完以后, 对所有 VPE 上的计算结果执行归约求和操作, 得到一个输出值  $y_n$ .

这种方法的优点是程序控制简单, 尤其是当系数长度  $N$  是 VPE 数量的整数倍时, 数据为 8 位或 16 位的定点数据时, 可基于 SIMD 充分挖掘 VPU 的指令和数据并行性. 利用 VPU 的归约求和指令, 易于软件流水, 能够取得高效的计算性能. 但是, 缺点是非常明显的:

(1) 该算法对系数长度有要求, 当系数长度  $N$  不是 VPE 数量的整数倍时, 不能充分发挥所有 VPE 的并行计算能力, 编程控制和软件流水困难.

(2) 该算法依赖于向量处理器的加法归约支持. 向量处理器一般只实现定点加法归约 (如 YHFT-Matrix), 而实现浮点加法归约的硬件开销大, 性能不高, 一般不

支持,因此限制了浮点数据类型的 FIR 应用.

(3)受限于向量数据访问带宽,该算法不支持 32 位复数类型的 FIR 计算.向量数据访问单元一般支持 VPE 数量 \* 32 位带宽,导致复数的 32 位实部与虚部加载到不同 VPE 的寄存器中,不能有效实现复数的乘法和累加.

这些缺点限制了许多 FIR 应用.因此,我们提出一种新的 FIR 滤波器向量化方法.假定向量处理器的 VPE 数量为 16,滤波器系数长度为  $N$ .将公式(5)展开,则连续 16 个输出值的计算如公式(6)所示:

$$\begin{cases} y_n = x_n h_0 + x_{n+1} h_1 + \cdots + x_{n+N-1} h_{N-1} \\ y_{n+1} = x_{n+1} h_0 + x_{n+2} h_1 + \cdots + x_{n+N} h_{N-1} \\ \vdots \\ y_{n+15} = x_{n+15} h_0 + x_{n+16} h_1 + \cdots + x_{n+14+N} h_{N-1} \end{cases} \quad (6)$$

公式(6)可表达为如下的矩阵形式:

$$\begin{bmatrix} y_n \\ y_{n+1} \\ \vdots \\ y_{n+15} \end{bmatrix} = \begin{bmatrix} x_n \\ x_{n+1} \\ \vdots \\ x_{n+15} \end{bmatrix} h_0 + \begin{bmatrix} x_{n+1} \\ x_{n+2} \\ \vdots \\ x_{n+16} \end{bmatrix} h_1 + \cdots + \begin{bmatrix} x_{n+N-1} \\ x_{n+N} \\ \vdots \\ x_{n+14+N} \end{bmatrix} h_{N-1}$$

即:

$$(y_n, y_{n+1}, \cdots, y_{n+15}) = \sum_{k=0}^{N-1} (x_{n+k}, x_{n+k+1}, \cdots, x_{n+k+15}) \otimes (\overbrace{h_k, h_k, \cdots, h_k}^{16}) \quad (7)$$

其中  $(y_n, y_{n+1}, \cdots, y_{n+15})$ ,  $(x_{n+k}, x_{n+k+1}, \cdots, x_{n+k+15})$ ,  $(\overbrace{h_k, h_k, \cdots, h_k}^{16})$  均为长度 16 的向量,与 VPE 的数量相匹配,  $\otimes$  表示向量乘法.因此输出值向量  $(y_n, y_{n+1}, \cdots, y_{n+15})$  可以通过  $N$  次向量乘法以及  $N-1$  次向量加法完成.

更进一步的考虑,若输入信号  $x_k$  和系数  $h_k$  为短数据类型,则可开发 SIMD 的数据并行性,以 16 位数据为例,每一次向量乘法表达如下:

$$\left( (x_{n+k}, x_{n+k+1}), \cdots, (x_{n+k+30}, x_{n+k+31}) \right) \otimes \left( (h_k, h_k), \cdots, (h_k, h_k) \right)$$

则在每个 VPE 上基于 SIMD 同时并行执行两个 16 位乘法,进一步提高数据并行性.

由此,根据公式(7),我们提出一种新的 FIR 滤波器向量化方法,基本思想是将 FIR 滤波器的卷积计算转化为系数长度  $(N)$  步向量乘法和加法计算,每一步的向量乘法和加法计算在各个 VPE 上并行执行.在每个 VPE

上的乘法和加法计算中,计算的源数据和结果都保存在 VPE 的本地寄存器上,避免了 VPE 之间的数据交换,并且计算一个输出结果的所有乘法和加法计算都在本 VPE 上完成,避免了 VPE 之间的加法归约计算.在每次循环结束时,在每一个 VPE 上完成一个输出结果的计算,即同时完成 VPE 数量个输出结果的计算.

与以往的 FIR 滤波器向量化方法相比,该向量化方法具有显著优势:

(1)每一步均为向量乘法和加法计算,向量长度与 VPE 数量相匹配,对任何数据类型都能够充分发挥所有 VPE 的并行计算能力,且不受限于系数长度  $N$  是否是 VPE 数量的整数倍,并且对于 8 位或 16 位的定点数据,可进一步基于 SIMD 充分挖掘 VPU 的数据并行性;

(2)计算一个输出结果的所有乘法和加法计算都在本 VPE 上完成,避免了 VPE 之间的加法归约计算,完全支持浮点类型的 FIR 应用,避免了受限于加法归约树的限制,并且能够取得与定点数据相当的高效性能;

(3)支持 32 位浮点复数的 FIR 计算,只需要通过合理的规划数据存放位置或者通过向量处理器支持的混洗操作,将复数的 32 位实部与虚部加载到同一 VPE 的寄存器,能高效的支持 32 位浮点复数的 FIR 滤波器计算.

### 3.2 向量数据的组织

向量处理器的数据组织直接影响算法执行效率.

对于 32 位定点和浮点实数和 16 位定点复数(实部和虚部都是 16 位定点数),输入信号  $x_k$  是连续存放的,并且复数的实部与虚部交替连续存放,由于 YHFT-Matrix 支持  $16 * 32$  位的向量数据加载,每个 VPE 上的向量乘法和加法计算所需要的源数据和结果在本地寄存器上存取,可充分开发 YHFT-Matrix 的 16 个 VPE 的并行计算性,能够有效实现所提出的向量化方法.

对于短实数数据(8 位、16 位定点),输入信号  $x_k$  是连续存放的,由于 YHFT-Matrix 支持  $16 * 32$  位的向量数据加载,可基于 SIMD 进一步开发数据并行性.以 8 位定点数据为例:

两次向量数据加载及混洗操作以后得到如下向量:

$$\begin{aligned} & x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7 \cdots x_{60} x_{61} x_{62} x_{63} \\ & x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 \cdots x_{61} x_{62} x_{63} x_{64} \\ & x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 \cdots x_{62} x_{63} x_{64} x_{65} \\ & x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10} \cdots x_{63} x_{64} x_{65} x_{66} \end{aligned} \quad (8)$$

对于 32 位浮点复数(实部和虚部都是 32 位浮点)的 FIR 滤波器,有两种数据组织方式,一种方式是数据按照通常的实部与虚部交替连续存放,如式(9)所示:

$$\begin{aligned} & x_0 x_0 x_1 x_1 x_{1i} \cdots x_7 x_7 i \\ & x_8 x_8 x_9 x_9 x_{9i} \cdots x_{15} x_{15} i \end{aligned} \quad (9)$$

通过一次混洗操作得到式(10)所示的数据形式:

$$\begin{aligned} & x_{0r}x_{1r}x_{2r}x_{3r}\cdots x_{14r}x_{15r} \\ & x_{0i}x_{1i}x_{2i}x_{3i}\cdots x_{14i}x_{15i} \end{aligned} \quad (10)$$

从而使得  $x_k$  的实部与虚部在同一 VPE 的本地寄存器上面,便于后续的乘法和加法计算。

另一种方式需要改变单核处理器通常采用的复数实部虚部交替连续存放的数据组织方法,而是所有输入信号  $x_k$  的实部与虚部分离,实部连续存放,虚部连续存放。如式(11)和(12)所示:

$$\begin{aligned} & x_{0r}x_{1r}x_{2r}\cdots x_{15r} \\ & x_{16r}x_{17r}x_{18r}\cdots x_{31r} \\ & \dots \end{aligned} \quad (11)$$

$$\begin{aligned} & x_{0i}x_{1i}x_{2i}\cdots x_{15i} \\ & x_{16i}x_{17i}x_{18i}\cdots x_{31i} \\ & \dots \end{aligned} \quad (12)$$

其中  $x_{kr}$ ,  $x_{ki}$  分别是  $x_k$  的实部与虚部。系数  $h_k$  的实部与虚部也是分离,实部连续存放,虚部连续存放。这样才能保证输入信号  $x_k$  和系数  $h_k$  的实部与虚部加载到同一 VPE,后续的复数乘法和结果累加才能够在 VPE 的本地寄存器进行,以实现公式(7)的向量计算。

系数  $h_k$  通常是预先确定的数据,有两种组织方法,一种是连续存放,数据加载到寄存器以后通过混洗操作得到值相同的系数向量,适合于系数长度  $N$  较大的情况。若系数长度  $N$  不大,可通过冗余数据的方式,即每个系数按照 VPE 的长度进行冗余存放,以 8 位定点数据为例,按照式(13)所示存放系数数据,这种方式的特点是通过适当的数据冗余存储,浪费少量存储空间,优点是不需要混洗操作,直接加载数据即可,能提高计算性能。

$$\begin{aligned} & \overbrace{h_0h_0h_0h_0h_0h_0h_0h_0\cdots h_0h_0h_0h_0}^{64} \\ & h_1h_1h_1h_1h_1h_1h_1h_1\cdots h_1h_1h_1h_1 \\ & h_2h_2h_2h_2h_2h_2h_2h_2\cdots h_2h_2h_2h_2 \\ & h_3h_3h_3h_3h_3h_3h_3h_3\cdots h_3h_3h_3h_3 \end{aligned} \quad (13)$$

对于短数据类型可采用 YHFT-Matrix 的 SIMD 指令实现的向量乘法和加法。

每次完成公式(7)的 4 步计算,计算结果保存在 VPE 的本地寄存器,依次循环下去,在每个 VPE 本地寄存器完成输出值向量累加计算,式(7)的  $N$  步完成以后,在 16 个 VPE 上同时完成 16 个输出值的计算。

## 4 性能评估与分析

在向量处理器 YHFT-Matrix 上实现和测试了 1024 点 50 抽头的向量化 16 位定点 FIR 滤波器的计算性能,图 2 和图 3 从执行时间和执行时钟周期数两方面对比

了几种典型高性能 DSP 的性能结果。

如图 2 所示,在主频 500MHz 的 YHFT-Matrix 上的执行时间仅为  $7.4\mu\text{s}$ ,而在 300MHz 的向量处理器 CVP 上的执行时间为  $12.4\mu\text{s}$ ,在主频 600MHz 的 TI TMS320C6416 上的执行时间高达  $27\mu\text{s}$ 。

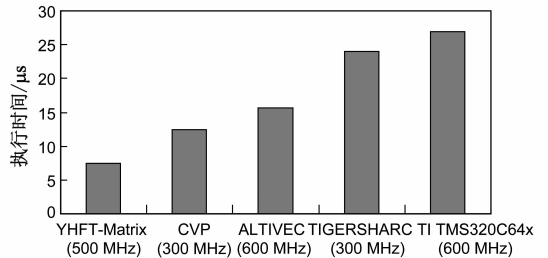


图2 1024点50抽头的16位定点FIR滤波器的执行时间比较

如图 3 所示,即使不考虑主频因素,从执行的时钟周期数来看,在 YHFT-Matrix 上的执行时钟周期数为 3692 拍,远低于 TI TMS320C6416 上的执行时钟周期数 16243,也低于向量处理器 CVP 上的执行时钟周期数 3728。

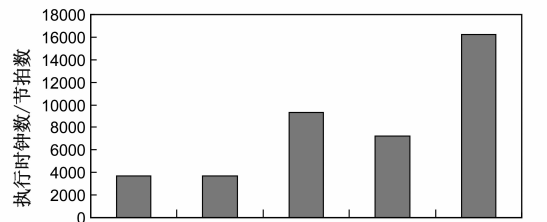


图3 1024点50抽头的16位定点FIR滤波器的执行时钟节拍数比较

因此,通过图 2 和图 3 可以看到,从执行时间和时钟周期数来看,我们提出的向量化 16 位定点 FIR 滤波器在 YHFT-Matrix 上的计算性能均显著优于其他 DSP 上的 FIR 滤波器性能,其原因是 YHFT-Matrix 是一款包含 16 个 VPE 的高性能向量处理器,并且我们提出的 FIR 向量化方法充分挖掘了 YHFT-Matrix 的指令和数据并行性,因此取得高效的执行性能。

图 4(a)和(b)分别对比了 32 位浮点复数和 32 位定点复数 FIR 滤波器在 YHFT-Matrix 上的性能,可以看出本文提出的向量化方法比传统的内循环方法的性能有显著提高,主要原因是本文提出的向量化方法不受限于向量处理器的长度以及向量处理器是否支持加法归约,支持任意系数长度和数据类型。

针对 32 位浮点复数 FIR 滤波器,在 YHFT-Matrix 分别测试 2048 点的不同系数长度的向量化浮点复数 FIR 滤波器的计算性能,根据 TI 公司发布的 TMS320C6713 的 Benchmark 结果值<sup>[9]</sup>,计算 YHFT-Matrix 相对 TMS320C6713 DSP 的加速比。如图 5 所示, YHFT-Matrix

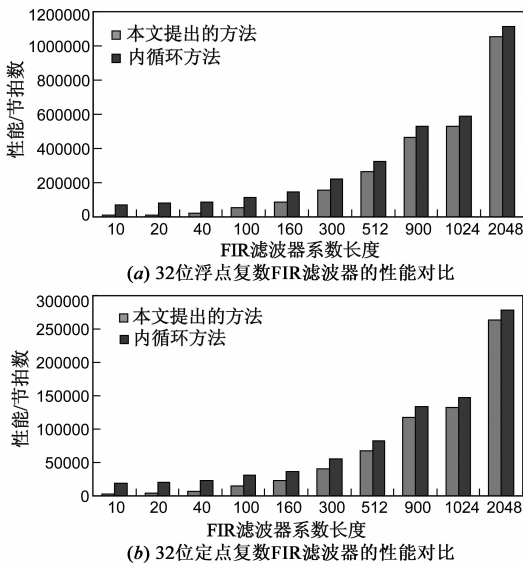


图4 两种向量化方法下复数FIR滤波器的性能对比

上的向量化浮点复数 FIR 滤波器的性能大约是 TMS320C6713 DSP 的 8 倍,当 FIR 系数长度从 10 依次增大到 2048 时,加速比稳定地从 4.92 增长到 7.98,并且在接近 8 时变化平稳。

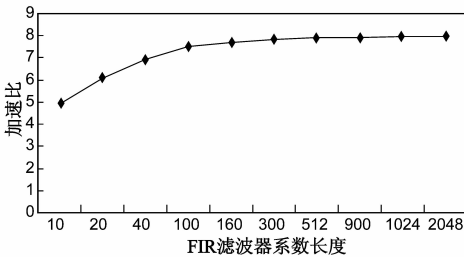


图5 2048点不同系数长度的向量化32位浮点复数FIR滤波器的加速比

针对 32 位定点复数 FIR 滤波器,在 YHFT-Matrix 分别测试 2048 点的不同系数长度的向量化定点复数 FIR 滤波器的计算性能,根据 TI 发布的 TMS320C6416 DSP 的 Benchmark 结果值<sup>[10]</sup>,计算 YHFT-Matrix 相对 TMS320C6416 DSP 的加速比。如图 6 所示, YHFT-Matrix 上的向量化定点复数 FIR 滤波器的性能大约是 TMS320C6416 DSP 的 16 倍,当 FIR 系数长度从 10 依次增大到 2048 时,加速比稳定地从 6.38 增长到 15.88,在

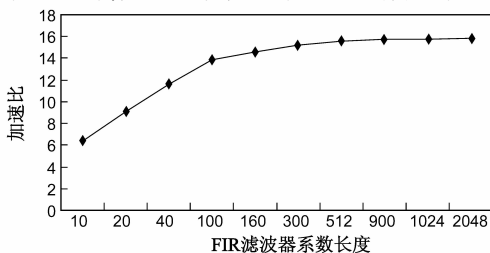


图6 2048点不同系数长度的向量化32位定点复数FIR滤波器的加速比

接近 16 时变化平稳。

值得注意的是,在图 5 和图 6 中可以看到,系数长度 10,20,40,100,300,900 都不是 VPE 数量 16 的倍数,表明系数长度值是否是 VPE 数量的整数倍对向量化 FIR 滤波器的性能没有任何影响,也不影响加速比,表明我们提出的 FIR 滤波器向量化方法能够适用于各种不同滤波器系数,都能够充分挖掘向量处理器全部 VPE 的计算资源,高效地支持 8 位、16 位、32 位定点实数和复数、32 位浮点实数和 32 位浮点复数各种数据类型,是一种面向向量处理器的高效、通用 FIR 滤波器向量化方法。

## 5 结论

本文研究了面向向量处理器的 FIR 滤波器向量化方法,提出了一种面向向量处理器的支持任意系数长度和数据类型的高效、通用 FIR 滤波器向量化方法。与现有方法相比,该方法能够充分挖掘向量处理器的指令级和数据级并行性,支持任意的滤波器系数长度,不受限于向量处理器是否支持加法归约,并且支持 8 位、16 位、32 位定点实数和复数、32 位浮点实数和 32 位浮点复数各种数据类型,实验结果表明,该方法能够取得非常高效的计算性能和加速比。

## 参考文献

- [1] Mehrara M, Jablin T, Upton D, et al. Multicore compilation strnd challenges[J]. IEEE Signal Processing Magazine, 2009, 26(6): 55 - 63.
- [2] 马宝山, 朱义胜. 一种用于基因预测的 FIR 数字滤波器[J]. 电子学报, 2007, 35(9): 1710 - 1713.  
Ma Bao-shan, Zhu Yisheng. FIR digital filter for gene prediction application[J]. Acta Electronica Sinica, 2007, 35(9): 1710 - 1713. (in Chinese)
- [3] 李辉, 张安, 赵敏, 等. 粒子群优化算法在 FIR 数字滤波器设计中的应用[J]. 电子学报, 2005, 33(7): 1338 - 1341.  
Li Hui, Zhang An, Zhao Min, et al. Particle swarm optimization algorithm for FIR digital filters design[J]. Acta Electronica Sinica, 2005, 33(7): 1338 - 1341. (in Chinese)
- [4] Mirzai S, Hosangadi A, Kastner R. FPGA implementation of high speed FIR filters using add and shift method[A]. Proceedings of International Conference on Computer Design[C]. San Jose, California: IEEE, 2006. 308 - 313.
- [5] Shahbahrami A, Juurlink B H H, Vassiliadis S. Efficient vectorization of the FIR filter[A]. Proceedings of the 16th Annual Workshop on Circuits, Systems and Signal Processing [C]. Veldhoven, the Netherlands: IEEE, 2005. 432 - 437.
- [6] Kutil R, Eder P. Parallelization of wavelet filters using SIMD extensions[J]. Parallel Processing Letters, 2006, 16(3): 335 -

349.

- [7] Kutil R. Parallelization of IIR filters using SIMD extensions [A]. Proceedings of the 15th International Conference on Systems, Signals and Image Processing [C]. Bratislava, Slovak Republic: IEEE, 2008. 65 – 68.
- [8] Dang B L, Engin N, Gaydadjiev G N. Efficient filtering with the co-vector processor [A]. Proceedings of the 14th Annual Workshop on Circuits, Systems and Signal Processing [C].

Veldhoven, The Netherlands: IEEE, 2003. 351 – 356.

- [9] Texas Instruments. C67x floating point benchmarks [R/OL]. <http://www.ti.com/sc/docs/products/dsp/c6000/67bench.htm>, 2011 – 11 – 03.
- [10] Texas Instruments. C64x floating point benchmarks [R/OL]. <http://focus.ti.com/dsp/docs/dspplatformscontentaut.tsp?sectionId=2&familyId=477&tabId=496>, 2011 – 11 – 03.

## 作者简介



刘 仲 男, 1971 年 11 月出生, 湖南邵东人. 2005 年获得国防科学技术大学计算机科学与技术专业博士学位, 现为国防科学技术大学计算机学院副研究员, 主要从事芯片验证与性能评测、嵌入式应用与算法优化等方面的研究工作.

E-mail: zhongliu@nudt.edu.cn



陈跃跃 男, 1960 年 3 月出生, 湖南邵东人. 1987 年获得国防科学技术大学计算机科学与技术专业硕士学位, 现为国防科学技术大学计算机学院研究员, 主要从事微处理器设计、体系结构和嵌入式系统等方面的研究工作.