

# Stolt 插值在多核多 DSP 上并行实现

梁之勇<sup>1,2,3</sup>

- (1. 中国电子科技集团公司第三十八研究所, 安徽合肥 230088;  
2. 数字阵列技术重点实验室, 安徽合肥 230088;  
3. 孔径阵列与空间探测安徽省重点实验室, 安徽合肥 230088)

**摘 要:** 随着 SAR 成像向高分辨率、大测绘带发展,  $\omega K$  算法应用越来越广。Stolt 插值作为  $\omega K$  算法核心, 运算量大同时对计算精度要求高, 传统的 DSP 已很难满足实时处理要求, 多核多 DSP 并行处理是一个重要发展趋势。介绍了一种基于 TMS320C6678 通用信号处理平台, 该处理器内置 8 核, 定点、浮点处理能力强, 并以此平台论述了 Stolt 插值多核多 DSP 并行实现。程序同时采用了 DMA CHAIN 技术来实现数据存取, 降低了数据存取时间, 有效地提高了系统实时性。

**关键词:** SAR 成像; TMS320C6678 多核处理器; Stolt 插值; 并行实现

**中图分类号:** TN957      **文献标志码:** A      **文章编号:** 1672-2337(2016)01-0065-04

## Multi-Core and Multiple DSP Design for Stolt Interpolation

LIANG Zhiyong<sup>1,2,3</sup>

- (1. No. 38 Research Institute of CETC, Hefei 230088, China;  
2. Key Laboratory of Digital Array Technology, Hefei 230088, China;  
3. Key Laboratory of Aperture Array and Space Application, Hefei 230088, China)

**Abstract:** With the development of SAR imaging towards high resolution and wide swath,  $\omega K$  algorithm is applied more and more widely. Stolt interpolation is the core of  $\omega K$  algorithm, which requires the large amount of computations and high calculation precision. The traditional DSP is difficult to meet the demand. Multi-core and multi-DSP parallel processing is an important development trend. This paper introduces a kind of 4 pieces of TMS320C6678 general signal processing platform, which has built-in 8 cores, strong fixed-point and floating-point processing ability. As a case study of stolt interpolation, the multi-core and multi-DSP parallel implementation is discussed. Using the DMA CHAIN technology to implement the data access reduces the data access time effectively and improves the real-time performance of the system.

**Key words:** SAR imaging; TMS320C6678 multi-core processor; Stolt interpolation; parallel implementation

## 0 引言

经过多年的发展, SAR 数字成像处理技术日趋成熟。目前常用的成像算法有改进 RD、CSA、 $\omega K$  等算法。改进 RD 算法难以实现高分辨宽测绘带处理, CS 算法在高分辨下距离-方位耦合特性较为明显, 而  $\omega K$  算法可以有效校正波束内所有目标的距离徙动。徙动校正正在方位频率域通过精确插值完成, 可有效补偿高次相位, 实现精确聚焦, 同时该算法可以有效应对高分辨下距离方位耦合的问题。因此  $\omega K$  算法是目前应用较广的高精度

成像算法之一<sup>[1]</sup>。

$\omega K$  算法的主要流程包括二维傅里叶变换、参考函数相乘, 距离频域 Stolt 插值、二维傅里叶逆变换, Stolt 插值完成了残余距离徙动校正、残余二次距离压缩、残余方位压缩, 这样便实现了所有目标的聚焦。

Stolt 插值是双精度运算, 同时随着测绘带变大, 128K 点甚至更大点数插值也越来越多。传统单核 DSP 在片内内存、运算能力、数据吞吐等方面的能力不足问题也随之出现。TI 公司新推出的多核 DSP TMS320C6678 可以有效地解决 Stolt 插值的处理瓶颈。

## 1 理论分析

设照射区某点目标到飞行航迹垂距为  $R_x$ , 则对回波信号进行两维傅里叶变换后的回波相位<sup>[2]</sup>可表示为

$$S_{\text{Angle}}(f_r, f_a) = -\frac{4\pi R_x}{c} \sqrt{(f_c + f_r)^2 - \frac{c^2 f_a^2}{4v^2}} - \frac{\pi f_r^2}{k_r} + 2\pi f_a \frac{y}{v} + \frac{4\pi R_0 f_r}{c} \quad (1)$$

式中,  $v$  为平台飞行速度,  $f_c$  为雷达中心频率,  $f_r$  为距离向频率,  $f_a$  为方位向频率,  $y$  为目标方位位置,  $c$  为光速,  $R_0$  为雷达采集起始距离。

对式(1)进行参考距离函数补偿, 补偿函数相位可表示为

$$H_{\text{Angle}}(f_r, f_a) = \frac{4\pi R_{\text{ref}}}{c} \sqrt{(f_c + f_r)^2 - \frac{c^2 f_a^2}{4v^2}} + \frac{\pi f_r^2}{k_r} - \frac{4\pi R_0 f_r}{c} \quad (2)$$

补偿后的信号为

$$S_{\text{Angle}}(f_r, f_a) = -\frac{4\pi(R_x - R_{\text{ref}})}{c} \sqrt{(f_c + f_r)^2 - \frac{c^2 f_a^2}{4v^2}} + 2\pi f_a \frac{y}{v} \quad (3)$$

对不同的  $f_a$ , 通过对数据进行 Stolt 插值, 可将信号变换成关于  $f'_r$  的线性函数, 即

$$S_{\text{Angle}}(f'_r, f_a) = -\frac{4\pi(R_x - R_{\text{ref}})}{c} (f_c + f'_r) + 2\pi f_a \frac{y}{v} \quad (4)$$

$f_r$  为距离向频率向量, 为非线性向量。  $f'_r$  为插值后距离向频率向量, 为线性向量。两者关系为

$$f_c + f'_r = \sqrt{(f_c + f_r)^2 - \frac{c^2 f_a^2}{4v^2}} \quad (5)$$

常规成像只需将式(4)作两维逆傅里叶变换就可得到该目标的两维脉冲压缩响应。因此可以看出 Stolt 插值是  $\omega K$  算法的核心。

## 2 多核多 DSP 上设计实现

随着分辨率、作用距离、测绘带宽等指标越来越高, 系统计算量也越来越大, 而 Stolt 插值为双精度复杂运算, 所需要的计算时间也大大增加。传统的 SAR 成像处理器 ADSPTS101、ADSPTS201 等已经不能满足系统实时性要求<sup>[3]</sup>。TI 公司新推

出的 C6678, 该芯片内部有 8 核, 运算能力强, 主要性能<sup>[4-5]</sup>如下:

1) C6678 单片处理能力: 内核主频最高 1.25 GHz, 浮点运算能力达到 160 GFlops;

2) 内部存储资源: 单片 C6678 每核内置 32KB L1P、32KB L1D、512KB L2, 同时 8 个内核共享 4MB L3SRAM;

3) 外设接口: SRIO 接口可以支持 4 个 1X、2 个 2X、1 个 4X, 每一个通道最高速率 5 Gbit/s, PCIe 接口有 2 个通道, 每个通道最高速率支持 5 Gbit/s; HyperLink 接口有 4 通道, 支持的带宽高达 50 Gbit/s。

### 2.1 硬件平台介绍

本文采用的基于 C6678 的通用信号处理硬件平台如图 1 所示。

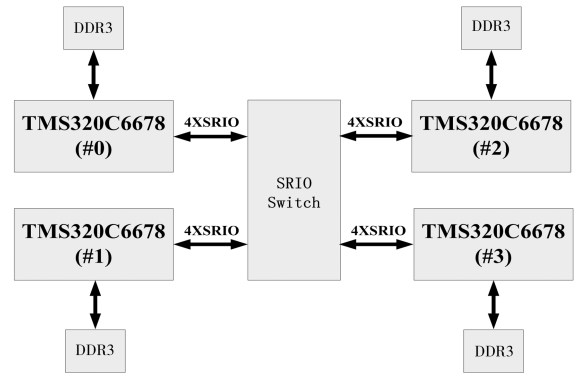


图 1 TMS320C6678 通用信号处理平台

该信号处理平台单板主要性能如下:

1) 定点处理能力达到 1280 GMACs、浮点处理能力最高 640 GFlops;

2) 单板外挂 32GB DDR3;

3) SRIO: Switch 实现 4 个 6678 的 SRIO 之间的交换。

### 2.2 任务分配

在一个成像孔径内距离向处理点数为  $RanLenth$ , 方位向处理点数为  $AzmLenth$ 。Stolt 插值在距离向操作, 同时方位向之间没有耦合, 因此 4 片 C6678 按照方位向平均分配任务, 每片 DSP 的 8 核按照距离向分配任务, 如图 2 所示。具体如下: DSP0 处理  $0 \sim 1/4 AzmLenth$ , DSP1 处理  $1/4 \sim 1/2 AzmLenth$ , DSP2 处理  $1/2 \sim 3/4 AzmLenth$ ,

DSP3 处理  $3/4 \sim 1 AzmLenth$ 。DSP 片内 8 核按距离向任务分配如下:核 0 处理  $0 \sim 1/8 RanLenth$ ,核 1 处理  $1/8 \sim 2/8 RanLenth$ ,核 2 处理  $2/8 \sim 3/8 RanLenth$ ,核 3 处理  $3/8 \sim 4/8 RanLenth$ ,核 4 处理  $4/8 \sim 5/8 RanLenth$ ,核 5 处理  $5/8 \sim 6/8 RanLenth$ ,核 6 处理  $6/8 \sim 7/8 RanLenth$ ,核 7 处理  $7/8 \sim 1 RanLenth$ <sup>[6]</sup>。通过上述任务分配实现了多核多 DSP 并行处理,极大地提高了系统实时性。

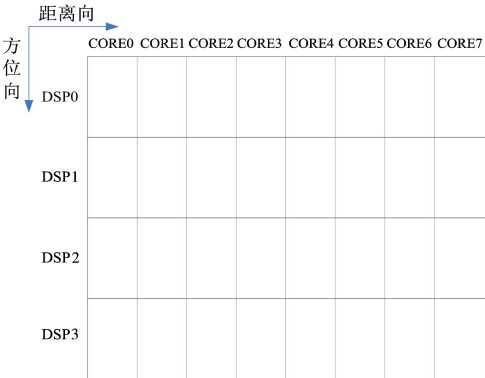


图2 4 DSP 任务分配

2.3 内存分配

为了提高实时性,所有计算变量、缓冲均存放在片内 RAM 上,L3 有 4MB,空间分配如表 1 所示,从内存分配可以看出最大能支持 128K 复数点 Stolt 插值。

表 1 L3 内存分配

地址起始	存储变量	内存大小
0xc000000	系统变量	1MB
0xc100000	原始数据	1MB
0xc200000	插值前频率向量	1MB
0xc300000	插值结果	1MB

L2 有 512KB,主要存放插值缓冲区,及插值后频率向量,由于 L2 是各核独有的,因此每核只需要插值点数的  $1/8$  存储空间,具体分配如表 2 所示。

表 2 L2 内存分配

地址起始	存储变量	内存大小
0x800000	系统变量	256KB
0x840000	插值计算缓冲	128KB
0x860000	插值后频率向量	128KB

2.4 程序设计

按照任务分配,一条距离线的 Stolt 插值由 8 个核同时完成。插值流程是首先将输出分成 8 段,计算每段输出频率范围,为保证分段插值结果正

确,输入频率范围至少要等于输出频率范围;然后根据输入频率范围和输入每点频率计算各核插值起始点和插值结束点位置;最后将各核对应起始点和结束点适当扩大,输入频率范围便大于输出频率范围,以保证交界处插值结果正确性。8 核 Stolt 插值计算流程如图 3 所示。

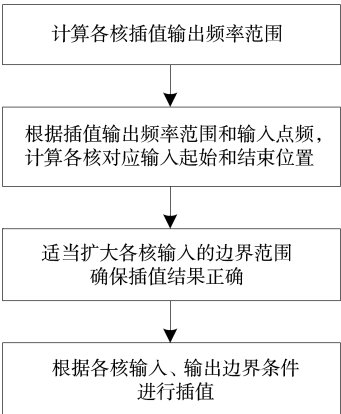


图3 8 核 Stolt 插值流程

根据前面的任务分配,4 片 DSP 根据距离线号的不同可以进行多 DSP 并行计算,有效提高了 Stolt 插值实时性。一般情况下插值原始数据按照一定方式存储在 DDR 中,因此 DSP 需要花费较多的时间将外存中的数据导入到内存中,计算结束之后再将内存中的结果存储到外存中,而 DSP 访问外存的速度要远远慢于内存,导致数据的存取可能要消耗较多的时间。这样使得 DSP 处理效率降低,满足不了系统实时性。考虑到这种情况设计了一种 DMA CHAIN 技术,即通过 DMA 存储计算结果后启动 DMA 读下一条距离线,这样数据存取可以通过 DMA 在后台完成,DSP 继续完成计算工作。这样存取数就不需要占用 CPU 时间。数据存取时间就可以掩盖在数据计算里面,进一步提高了 Stolt 插值实时性。图 4 是整个插值流程,其中  $N = \frac{1}{4} AzmLenth \times DspId$ ,  $AzmLenth$  为方位向点数,  $DspId$  为 DSP 号,分别为 0,1,2,3,这样 DSP 处理的距离线号便和自身的 DSP 号关联起来,实现了多 DSP 并行计算。

3 实验结果

3.1 并行 DMA CHAIN 设计效率分析

Stolt 插值测试环境:DSP 主频 1 GHz,DDR3 时钟 700 MHz,距离向 8 192 点插值。表 3 为插值时间。从图中可以看出 8 核并行插值运算时间需

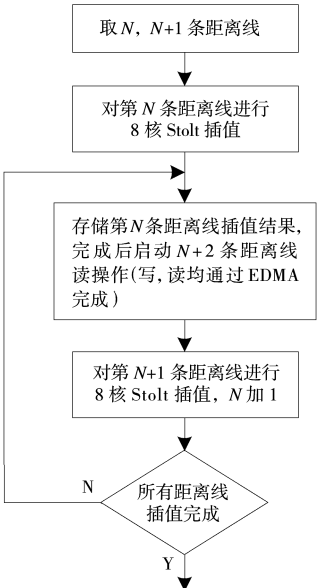


图 4 DSP 计算流程

要 31 795 时钟周期,取数据时间为 11 612 时钟周期,存数据时间为 11 620 时钟周期。没有 DMA CHAIN 完成插值总时间为 55 027 时钟周期。采用 DMA CHAIN 技术后时钟周期为 32 395 时钟周期,比插值计算时间略多,主要是因为 DMA 配置和启动需要一些时钟周期数。前者时间较长主要是因为取数、计算、存数整个流程是串行的,后者在采用 DMA CHAIN 技术后存取数和计算是并行的,时间性能提高了 41. 13%。因此 DMA CHAIN 技术有效地提高了插值的实时性。

表 3 插值时间分析

插值流程	取数时间	计算时间	存数时间	插值完成时间	
				无	有
				DMA CHAIN	DMA CHAIN
时钟周期	11 612	31 795	11 620	55 027	32 395

3. 2 多核 Stolt 插值结果分析

在 Stolt 插值多核分段的具体实现中,相邻分段之间如何处理,是一个需要特别关注的问题。正常情况下多核任务分配是将输入、输出平均分成 8 段进行插值,程序中距离向处理点数为 8 192 点,输入频率为 0 到 8 191 线性递增,输入幅值为对应频率的一半,输出频率向量根据式(5)计算得到,这样便可通过 Stolt 插值计算输出的幅值。该情况下插值结果如图 5 所示,可以看出部分插值结果有异常值。图 6 为局部放大图,从图中可知第 1024 点插值结果为 0,该点是核 0 和核 1 分段交界处。图 5 中插值异常值均出现在核间交界处。原因是将输入、输

出平均分 8 段来计算,输入输出处理的点数一样,但是输入和输出每点频率不一致导致输入输出频率范围不对应,导致各核交界处插值结果有误。

本文采用的方法是将输出分成 8 段,程序自动计算输入范围。插值结果如图 7 所示,各核交界处的插值结果均已正确。

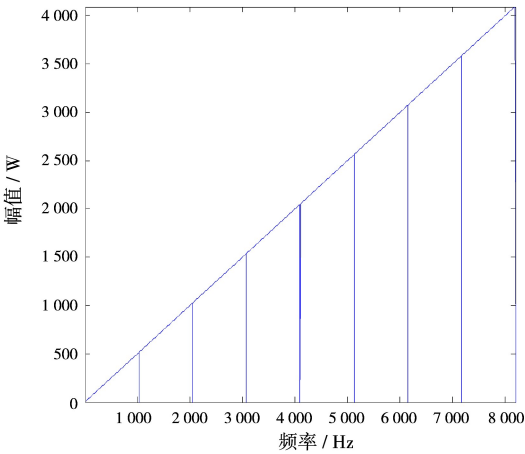


图 5 8 核插值结果

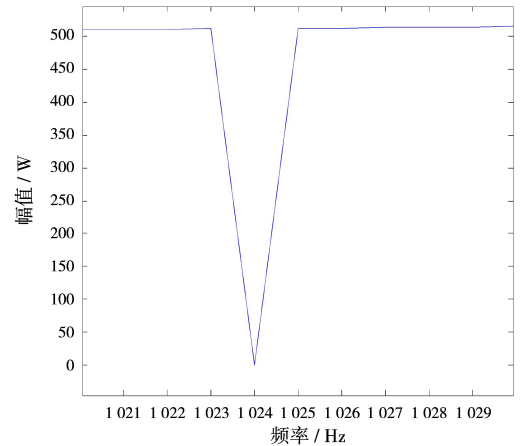


图 6 插值结果局部图

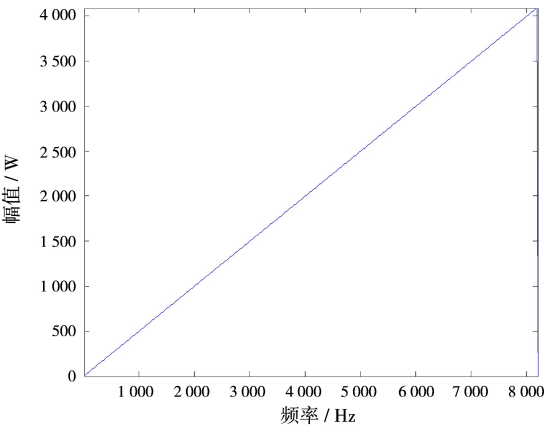


图 7 改进后 8 核插值结果 (下转第 75 页)