

用FPGA实现SAR实时信号处理的进一步研究

孙志坚 刘学梅 黄建喜 肖欣

青岛建工学院(266033)

2008-10-23

摘要: 高时效性是SAR成像系统的一个关键性能,要求处理的数据量大、运算量大和快速进行实时处理。SAR成像运算量主要集中在距离向和方位向的压缩处理上,常用的压缩处理方案是采用高速DSP实现,这种方法曾被认为是SAR实时处理的最佳硬件实现方案。但是,近几年可编程器件的发展,使得FPGA成为比DSP更为优越的压缩处理方式。结合工程实践,介绍了采用ALTERA公司的Stratix系列芯片实现SAR实时处理机系统的具体设计方案。

关键词: DSP 合成孔径雷达(SAR) 实时信号处理 FFT算法 数据压缩

SAR成像处理运算量非常巨大,要达到准实时甚至全实时的成像处理速度,就需要高性能的处理设备。常用的数据处理方案是采用高速DSP芯片实现。这种方法曾被认为是SAR实时处理最佳的硬件实现方案,但是近几年可编程器件的发展使得FPGA已经成为比DSP更优越的压缩处理方式。与DSP相比,FPGA的主要优点有:①速度快,FPGA有内置的高速乘法器和加法器。②DSP内部一般没有大量的存储器,但是SAR实时处理运算需要存储大量的数据,只能外接存储器,这样往往会使得运算速度下降,同时电路也会更复杂和不稳定。目前,高档的FPGA中有巨量的高速存储器,不用外接存储器便可实现SAR实时处理运算,其速度更快,电路也更简单。③FPGA是硬件可编程的,比DSP更加灵活,DSP往往需要外部的接口和控制芯片配合工作,FPGA则不需要,这样使得硬件更简单和小型化。

在SAR成像中,对方位维和距离维分别进行处理,不对成像质量造成太大的损失。采用二维分离处理的算法结构可以大大节省计算量,减少算法的复杂程度,而且比较灵活实用。其算法流程图如图1所示。

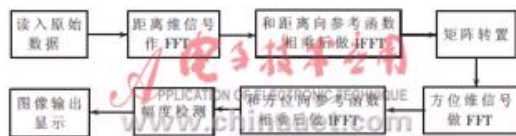


图1 SAR成像数据处理算法流程图

在实际中,IFFT的实现可以通过将输入数据取共轭之后做FFT运算,再取共轭,再乘1/N来完成,所以基于高速FPGA的FFT算法的实现成了SAR实时信号处理机系统的核心部分,而正确的地址产生是整个FFT运算的关键。用DSP和FPGA实现1024点FFT运算的性能比较如表1所示。

表1 DSP和FPGA实现1024点FFT运算的性能比较

芯片类型	DSP (TMS320C64124-SHARC)	FPGA Stratix系列 (ALTERA)
FFT实现类型	流水线,块浮点	并行,块浮点
完成1024点FFT时间	~12.8μs	~12.8μs

1 FFT在FPGA中实现的基本结构

1.1 基2 FFT在FPGA中的实现

基2 FFT运算的基本单元是蝶形运算单元,其每一级的运算都由N/2个蝶形运算构成,第m级的一个蝶形运算的两节点分别为X_m(n)和X_m(n+N/2^m)。基2的蝶形运算可表示为:

$$X_m(n) = X_{m-1}(n) + X_{m-1}(n + N/2^m)$$

$$X_m(n + N/2^m) = [X_{m-1}(n) - X_{m-1}(n + N/2^m)] W_N^r$$

式中,m表示第m级蝶形算法,n为数据所在的行数,N为所要计算的数据的点数,W_N^r为旋转因子。蝶形算法如图2所示,由一次复乘、两次复加组成。

从上式可以得出基2蝶形运算需四个乘法器,但在硬件实现中,旋转因子w_N^r和w_N^r都先被归一化,在本系统中,对实部和虚部分别按32位有符号数归一化,所以用于数据归一化还需两个乘法器,也就是说一共需要六个乘法器。Stratix系列芯片提供内置乘法器内核,EP1S25一共有80个8位乘法器。由8个8位乘法器可以组成一个32位乘法器,所以一共可以提供10个32位乘法器。在本系统中,FFT运算需要6个32位乘法器,参考函数相乘需要4个32位乘法器,一共需要10个32位乘法器,EP1S25刚好满足要求。

程序采用模块化结构,可以很方便地修改点数,满足不同的系统要求,也可以以此为基础改写为基4的FFT程序,应用于更高档的芯片,满足更高系统的要求。用FPGA实现FFT的基本结构如图3所示。

Altera公司

- 公司介绍
- 联系公司
- 公司新闻
- 加入收藏夹

热点专题

- 2008--嵌入式技术创新及应用高峰论坛
- 2008飞思卡尔技术论坛
- Altera公司SOPC World 2008专题报道
- 第十届高交会电子展
- 科技闪耀北京奥运
- ADLINK DAY—2008年量测与自动化技术国际高峰论坛
- 中国电子学会Xilinx杯开放源码硬件创新大赛
- 赛灵思公司Virtex-5系列FPGA
- 3G知识
- IPTV
- 触摸屏技术
- RoHS

杂志精华

- 基于CC2430的无线传感器...
- 无线传感器网络应用系统综述
- 无线传感器网络在野外测量中的...
- 基于竞争的无线传感器网络
- 用于矿井环境监测的无线传感器...
- 具有自适应通信能力的无线传感...
- 基于传感器网络技术的深孔测径...
- 基于无线传感器网络的家庭安防...
- 基于ATmega128L与C...
- 无线传感器网络中移动节点设备...

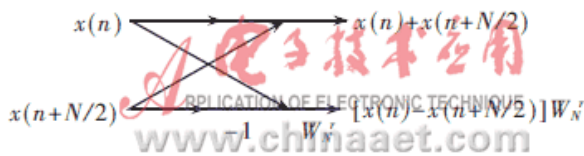


图2 基2蝶形运算结构图

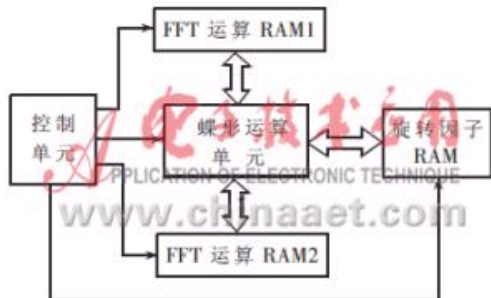


图3 FFT运算基本结构

以整个16点DIF为例的基2 FFT的运算流程图如图4所示。

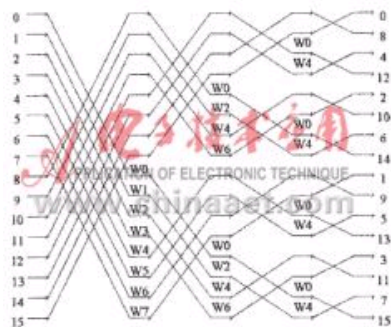


图4 16点基2 FFT运算流程图

1.2 基4 FFT 在FPGA中的实现

如果采用更高档的FPGA芯片，由于片内乘法器的增多，可以实现更高基的FFT运算，比如基4或者混合基FFT运算，这样可以成倍地提高运算速度。以4K点FFT为例，如果时钟频率是100MHz，即每个时钟周期是10ns，基2 FFT运算一级需要2K个时钟周期，而 $4K = 2^{12}$ ，即4K点、基2 FFT一共12级，所以4K点、基2FFT一共需要的时间是 $2K \times 12 \times 10ns = 240Kns$ ；4K点、基4 FFT一级运算只需要1K个时钟周期，且 $4K = 4^6$ ，即4K点、基4 FFT一共只有6级基4 FFT运算，所以如果时钟频率不变，基4 FFT所需要的总时间只有基2算法的1/4，即 $1K \times 6 \times 10ns = 60Kns$ ，其它点数的运算也是如此。基4算法速度是基2算法的四倍，这对于更高要求的实时处理系统非常有价值。

用FPGA实现基4 FFT的基本结构和实现基2FFT是一样的(如图3所示)，基4FFT的蝶形运算单元的结构图如图5所示。

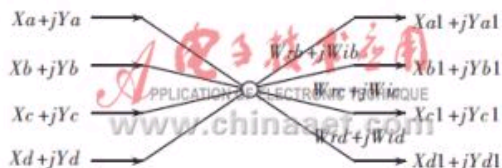


图5 基4 FFT蝶形运算单元结构图

蝶形运算单元的输入、输出关系如下：

$$Y_{a1} = X_a + X_b + X_c + X_d$$

$$Y_{b1} = Y_a + Y_b + Y_c + Y_d$$

$$Y_{c1} = (X_a + Y_b - X_c - Y_d) W_{rb} + (Y_a - X_b + Y_c + X_d) W_{ib}$$

$$Y_{d1} = (Y_a - X_b + Y_c + X_d) W_{rb} + (X_a + Y_b - X_c - Y_d) W_{ib}$$

$$Y_{c1} = (X_a - X_b + X_c - X_d) W_{rc} + (Y_a - Y_b + Y_c - Y_d) W_{ic}$$

$$Y_{d1} = (Y_a - Y_b + Y_c - Y_d) W_{rc} + (X_a - X_b + X_c - X_d) W_{ic}$$

$$Y_{d1} = (X_a - Y_b + X_c + Y_d) W_{rd} + (Y_a + X_b - Y_c - X_d) W_{id}$$

$$Y_{d1} = (Y_a + X_b - Y_c - X_d) W_{rd} + (X_a - Y_b + X_c + Y_d) W_{id}$$

可以看出，基4的蝶形运算单元可以同时输入输出四个数据进行并行蝶形运算，这就比基2蝶形运算的速度提高一倍。从运算公式还可以看出，基4 FFT运算需要12个乘法器，如果数据仍取32位，则需要 $12 \times 8 = 96$ 个8位乘法器，再考虑到参考函数相乘需要的 $4 \times 8 = 32$ 个8位乘法器，一共需要 $96 + 32 = 128$ 个8位乘法器。采用Stratix系列的EP1S60及EP1S60以上的芯片就可以实现。图6 是64点基4 FFT前两级的流程图。

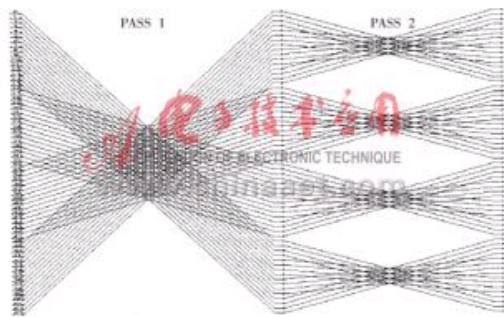


图6 $N=64$ 点的基4 FFT 前两级流程图

在研制过程中,对基4 FFT也进行了软件编程和仿真调试,选用的是用Altera公司的EP1S60F1020C60芯片实现基4方式的16K点数据压缩,达到了预期的目的。

1.3 FPGA中实现FFT的几个关键技术

1.3.1 运算中的精度和溢出问题

在FFT运算过程中,一个数据要经过很多次的乘法和加法。以4K点FFT运算为例,一共有12级运算,每一级运算又是由两级加法和一级乘法运算构成,也就是说,原始数据要经过24级加法和12级乘法才能得到最终结果,所以在运算中如何保证精度和不溢出是一个重要问题。由于硬件资源的限制,如何解决溢出问题,又能使精度损失在系统中可以接受是很关键的。

在FFT的每一级完成之后,对结果数据进行块浮点处理,可以保证在不溢出的情况下将有效数据前移,这样可以使得下一级运算保持更高的精度。

1.3.2 FFT运算中的地址产生问题

在FFT运算过程中,所有的数据都存放在RAM中,所以不管是读数据还是写数据,都涉及到RAM的地址,地址产生是整个FFT程序的关键问题之一。

由FFT运算流程图可以看出,DIF形式的FFT输入数据地址是顺序的,但每一级输出数据地址不再是原来的正序,正确的位反转技术是保证每次操作地址正确的关键。

2 系统软硬件调试及结果

本系统的调试可以概括为软件仿真和硬件调试两个部分。基2算法的实时信号处理机已经通过调试验证,并交付使用;基4算法由于受硬件条件限制,现只在实验室环境中通过仿真验证和调试。下面分别介绍调试中遇到的一些典型问题并给出各阶段及系统最终的调试结果。

2.1 FFT算法实现的验证

数据压缩的核心内容是FFT运算,所以首先对实现FFT运算的数据压缩板进行调试。众所周知,方波经过FFT运算之后应该是一个sinc函数。从测试口输入一个方波作为原始数据,如果输出的结果是一个sinc函数就可以证明FFT运算成功。图7是原始数据实部和虚部都为20点的方波(数据总长度为512点)经过实际硬件FFT运算之后的实部、虚部及求模后的波形。可以看出结果是一个sinc函数,说明实际硬件的FFT运算是正确的。

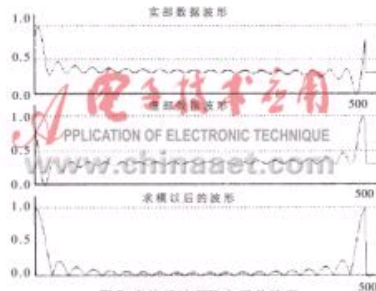


图7 方波经过FFT之后的波形

2.2 数据压缩板实现数据压缩运算

要验证数据压缩运算,实际上就是要验证FFT+参考函数相乘+IFFT运算。考虑到两个宽度相同的方波相卷积的结果是一个三角波,时域的卷积又等效于频域相乘,所以,如果令原始数据为方波,参考函数为同样宽度的方波,则经过FFT之后的数据为sinc函数,最终结果将是一个三角波。图8为实验输出结果,可以看出结果是一个三角波,说明FFT+参考函数相乘+IFFT运算是正确的。图9是实际硬件完成的SAR点目标压缩波形。

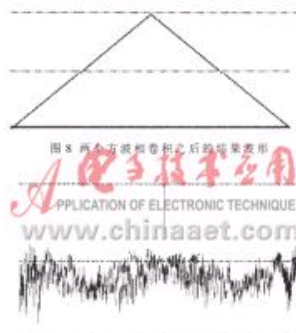


图8 两个方波相卷积之后的结果波形

图9 点目标压缩波形

通过比较可以看出以上测试结果与理论值吻合的很好，证明数据压缩板工作的很好。

2.3 数据压缩系统联调

将同一帧原始数据送入实时信号处理系统中，经过距离压缩、矩阵转置、方位压缩后所得结果与软件成像结果进行对比，实验表明两者得出的图像几乎一致，证明硬件成像结果正确。

本文研究了基于高速FPGA的SAR实时信号处理机系统的组成，详细讲述采用基2和基4算法在FPGA中实现FFT算法以及关键的技术。通过实验表明，这套方案是正确的，并且这套系统已经交付使用。它的运算精度和数据处理速度能够达到要求。

参考文献

- 1 Donald R. Wehner. High Resolution Radar. Artech House, 1987
- 2 C. Prati, F. Rocca. Focusing SAR Data with Time-varying Doppler centroid. IEEE Trans. on Geoscience and Remote Sensing, 1992; Ge-30(3)
- 3 Alberto Moreira, Yonghong Hong. Airborne SAR Processing of Hightly Squinted Data Using a Chirp Scaling Approach with Integrated Motion Compensation. IEEE GRS. September 1994; 32(5)
- 4 张贤达. 现代信号处理. 北京: 清华大学出版社, 1995

在线联系

[添加到收藏夹](#)

关于“[用FPGA实现SAR实时信号处理的进一步研究](#)”，我有如下需求或意向：

用户名: 密码: 验证码:  [欢迎注册](#)

相关应用

- 智能大厦视频图像数字压缩系统设计
- TMS320C54xx DSP的USB接口实现
- GPS定位数据压缩算法的设计与实现
- CF卡与双核DSP的实现
- DSP系统应用中FLASH在线编程方法
- 中性线有源滤波器重复学习控制及其数字算法实现

[版权声明](#) | [投稿须知](#) | [《电子技术应用》投稿](#) | [网站地图](#) | [帮助中心](#) | [广告中心](#) | [关于我们](#) | [管理员信箱](#)

[回到顶端](#)

《电子技术应用》编辑部版权所有

地址: 北京海淀区清华东路25号电子六所大厦

联系电话: 82306084 / 82306085 传真: 62311179 京ICP备05053646号

推荐分辨率1024*768 IE6.0版本

