

基于 Systolic 阵的 IQRD—SMI 算法的研究与 FPGA 优化实现

刘禹韬, 包志强, 李龙龙, 苏子昊

(西安邮电大学, 西安 710061)

摘要: 针对自适应波束形成系统中权值求解计算量大的问题, 研究了基于 LCMV 的逆 QR 分解算法 (IQRD—SMI 算法) 及其传统脉动流水的 GR—TSA 实现结构, 在 Matlab 中对算法以脉动阵搭建并进行算法仿真; 分析了 CORDIC 技术的原理以及如何利用它在 FPGA 上实现 Systolic 阵中的处理单元; 在此基础上, 给出一种基于 Systolic 阵列的复用 IQRD—SMI 算法实现结构 GR—ITSA, 比较了它与已有文献中结构的异同; 基于 ISE 软件, 在 Xilinx 的 xc7k325t 型 FPGA 上实现了基于脉动阵列的 IQRD—SMI 算法的底层设计和硬件仿真, 并将 FPGA 定点仿真数据回传 Matlab 与计算机浮点仿真数据对比, 进而进行系统的性能与误差分析。

关键词: 自适应波束形成算法; FPGA; Matlab; 脉动阵; CORDIC 技术; GR—ITSA

Research and FPGA Optimization Implementation of IQRD—SMI Algorithm Based on Systolic Array

Liu Yutao, Bao Zhiqiang, Li Longlong, Su Zihao

(Xi'an University of Posts and Telecommunications, Xi'an 710061, China)

Abstract: For the problem of the weight values need a large amount of calculations in the adaptive beam-forming system, studied the inverse QR decomposition algorithm based on LCMV (IQRD—SMI algorithm) and the structure of traditional Systolic array GR—TSA, built with systolic array structures and simulated in Matlab. The principle of CORDIC technology is analyzed and how to use it on FPGA implement systolic array processing unit. On the basis of this, Based on the reuse of systolic arrays IQRD—SMI algorithm structure is given, Compared it with existing in the literature. Underlying design based on IQRD—SMI algorithm and hardware simulation based on systolic array be realized with xc7k325t in Xilinx FPGA, And the FPGA fixed-point simulation data and the computer floating-point simulation data contrast. The system and error is analysed.

Keywords: adaptive beamforming algorithms; FPGA; Matlab; systolic array; CORDIC techniques; GR—ITSA

0 引言

自适应数字波束形成 (ADBF) 系统中, 为了改善波束性能、增强期望信号的信噪比, 应使用高效的自适应算法对即时采样数据进行处理^[1-3]。但自适应算法运算量庞大、并行与实时性能差、硬件实现资源消耗大的问题在实际应用中比较突出^[4-6]。

为应对以上问题, Teitelbaum^[7]提出了对采样协方差矩阵直接进行 QR 分解的 (QRD—SMI) 算法, 该算法具有高度并行性结构, 易于使用 Systolic 脉动阵列搭建, 在实际应用中能够与高性能的 FPGA 结合实现^[8-9]。但是传统的 QRD—SMI 算法存在需要通过求解两个三角矩阵才能得到加权矢量 w 的缺点, 并且后向带入需用到资源消耗很高的硬件除法器, 这对于系统的实现和权矢量 w 的实时并行获取增加了复杂度。文献 [10] 中对此算法进行了详细的介绍。

文献 [11] 提出了一种逆 QR 分解算法 (IQRD—SMI), 该算法避免了通过前向带入和后向带入的方式求解方程, 在脉动阵结构上的优化达到了可以实时、并行地进行权值获取, 更易于硬件实现。但是并没有实际的工程实现, 并且其结构在具体实现中存在软件编程复杂和硬件资源消耗高的问题。

本文研究了数字逆 QR 分解 SMI 波束形成算法, 结合 Systolic 阵在 Matlab 中搭建仿真实验, 介绍了 CORDIC 算法与使用方法, 并用 Verilog 语言在 ISE 中进行了硬件实现, 进一步提出了一种复用的硬件实现结构 GR—ITSA, 简化了软件编程, 节约了硬件资源成本。并将硬件定点处理数据与软件浮点仿真数据同时回馈到 Matlab 中对比, 进行误差分析。

1 基于逆 QR 分解的 SMI 波束形成算法

IQRD—SMI (逆 QR 分解 SMI) 算法, 是在 QR 分解算法上的进一步改进, 它能够回避 QRD—SMI 算法中的前后向回代问题, 而且该算法仍能使用简单的 Systolic 阵列结构搭建, 使权抽取操作得以全速并行实现, 在软件编程与硬件实现中的便捷性大大提升。

设采样数据矩阵 X 为 n 次快拍得到的 $M \times n$ 维阵, 即

收稿日期: 2015-08-21; 修回日期: 2015-09-17。

基金项目: 国家自然科学基金资助项目 (61271276); 陕西省自然科学基金资助项目 (2012JQ8011)。

作者简介: 刘禹韬 (1988-), 男, 辽宁锦州人, 硕士研究生, 主要从事信息处理技术及应用方向的研究。

$$\mathbf{X} = \begin{pmatrix} x_1(1) & x_1(2) & \vdots & x_1(n) \\ x_2(1) & x_2(2) & \vdots & x_2(n) \\ \vdots & \vdots & \ddots & \vdots \\ x_M(1) & x_M(2) & \vdots & x_M(n) \end{pmatrix} \quad (1)$$

传统采用 QR 分解 SMI 算法首先通过对协方差矩阵 \mathbf{R}_{xx} 的估计, 避免了直接使用 \mathbf{R}_{xx} 矩阵去求解线性方程。然后使用酉矩阵 \mathbf{Q} 通过 Givens 旋转对采样矩阵 \mathbf{QR} 分解, 最终简化为通过求解如 (2) 式所示的线性方程组来取得自适应权值 w :

$$\mathbf{A}_n^H v = s, \mathbf{A}_n w = v \quad (2)$$

式 (2) 中, s 为导向矢量; \mathbf{A}_n 是上三角阵, 通过使用 $n \times n$ 维酉阵将 x_n 三角化而得到; 对数据矩阵 $x_n \mathbf{QR}$ 分解的常用方法有 Givens 旋转和 CORDIC 算法。通过前向回代和后向回代的方法, 可分别对式 (2) 中的上三角阵与下三角阵求解, 从而得到自适应权 $w^{[12]}$ 。

采用 IQR 分解的 SMI 算法在中间变量 v 和自适应权 w 上的求解与 QRD-SMI 方法基本相同。接下来根据式 (2) 设 $v_n = \mathbf{A}_n^H a$, 则可以得到 n 次快拍采样数据的权向量 $w(n) = \mathbf{A}_n^{-1} v_n$ 。

IQRD-SMI 算法在初始化和旋转因子的求解上与前面有所不同。初始化时, 并不依照采样数据矩阵, 而是自行设定一个参考值; 在进行 QR 分解时, 不是直接对采样数据矩阵进行分解, 而是对一个中间变量 $z(n) = [\mathbf{A}_n^{-H}]^* x(n)$ 进行 QR 分解, 进而对自适应权向量进行更新求解。

其算法实现步骤如下:

1) 初始化。 $\mathbf{A}_n^{-1} = \delta^{-1} \mathbf{I}$, δ 是一个小的大于 0 的常数, \mathbf{I} 是单位阵, 得出 \mathbf{A}_0^{-H} 和 v_0 。

2) $k = 1, 2, \dots$, 递推更新。

首先, 计算中间向量 $z(n) = [\mathbf{A}_{n-1}^{-H}]^* x(n)$

接着, 由下列方程确定旋转所用酉矩阵 $\mathbf{P}(n) \begin{bmatrix} z(n) \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ l(n) \end{bmatrix}$;

最后, 用得到的酉阵 $\mathbf{P}(n)$ 更新下三角矩阵 \mathbf{A}_n^{-H} 与中间状态量 v_n , 即

$$\mathbf{P}(n) \begin{bmatrix} \mathbf{A}_{n-1}^{-H} & M & v_{n-1} \\ 0 & M & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_n^{-H} & M & v_n \\ g^H(n) & M & \alpha(n) \end{bmatrix}$$

3) 更新自适应权矢量 $w^T(n) = v^T(n) [\mathbf{A}_n^{-H}]^*$

2 脉动阵 (Systolic 阵)

Systolic 英文解释为“心脏收缩”, H. T. Kung 根据心脏按节拍收缩而使血液流至全身的原理提出了 Systolic 阵列结构, Systolic 阵由一系列仅具有单一功能的处理单元 (PE) 构成, 每个 PE 结构简单、易于实现、耗时耗能少, 所有的 PE 单元都在有节拍的同步运行, 系统工作时将从存储器中按节奏抽取数据并压入脉动阵, 沿途经过并行的 PE 流水线, 由于脉动阵列的结构特点, 使数据能够高效、连续地完成整个处理过程。脉动阵结构原理如图 1 所示。

以 4 阵元为例, 应用 Givens 旋转 IQRD-SMI 算法的三角脉动阵列 (GR-TSA, Givens Rotation-Triangular systolic array) 示意图详见图 2。

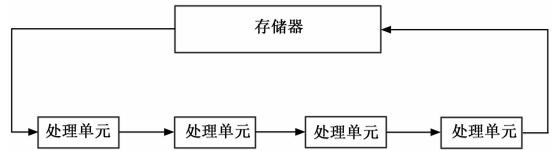


图 1 Systolic 脉动阵列结构原理框图

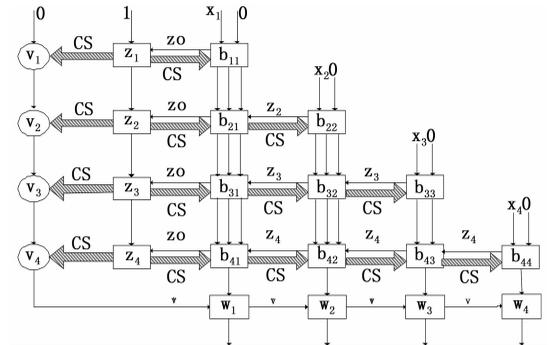


图 2 IQRD-SMI 算法 GR-TSA 阵示意图

内部各单元具体功能阐述如下:

1) bPE 单元

$$(1) \begin{cases} b_{ij}(k) = cb_{ij}(k-1) + s^* y_{in} \\ y_{out}(k) = c y_{in} - sb_{ij}(k-1) \end{cases}$$

$$(2) z_0 = b_{ij}^* x_{in} + z_i$$

$$(3) z_{out} = b_{ij}^*$$

$$(4) x_{out} = x_{in}$$

2) zPE 单元

$$c = \frac{x_{in}}{\sqrt{|x_{in}|^2 + |z_0|^2}}, s = \frac{z_0}{\sqrt{|x_{in}|^2 + |z_0|^2}}$$

$$x_{out} = \sqrt{|x_{in}|^2 + |z_0|^2}$$

3) vPE 单元

$$\begin{cases} v_i(k) = cv_i(k-1) + s^* x_{in}(k) \\ x_{out}(k) = cx_{in}(k) - sv_i(k-1) \end{cases}$$

$$y_{out} = v_i$$

4) wPE 单元

(1) 当非对角元 b_{ij} 到来时完成乘积累加; $w_i(k) = z_{out} y_{in} + w_i(k-1)$;

(2) 当对角元 b_{ii} 到来时完成乘积累加后输出 w_i 并清零;

3 IQRD-SMI 算法的 FPGA 优化及实现

3.1 CORDIC 技术

GIVENS 旋转运算需要进行开方与除法操作, 这些运算会消耗大量的硬件资源, 造成系统功耗过高的问题, 从而达不到设计标准。此时引入 CORDIC (coordinate rotation digital computer) 技术来实现 Givens 旋转。CORDIC 技术的基本思想是将一组固定的角度 θ_i 利用循环迭代算法, 通过不断地摆动进而逼近所需旋转的角度 θ 。由于仅需移位和加/减法即可完成 CORDIC 运算, 这样大大节约了 FPGA 硬件资源, 非常适合在 FPGA 系统上实现。

基于 CORDIC 的无开方复数 GIVENS 旋转思想^[13], 采用多个 CORDIC 单元依次完成复数转化实数的相位翻译、实数

值与前一时刻值的相位翻译, 从而得到 GIVENS 变换的相位, 再经过直接数字式频率合成器 DDS (Direct Digital Synthesizer) 就能得到角度值, 进而经过 systolic 阵的依次处理实现矩阵的 QR 分解。

3.2 IQRD-SMI 算法脉动阵结构的 FPGA 实现

脉动阵列 FPGA 实现的核心模块是旋转单元 (bPE), 旋转单元实现步骤:

- 1) 进行两次 CORDIC 旋转: 首先使用 CORDIC 核来实现对输入数据求模, 取得输入数据的模 $|x_{in}|$ 和旋转角 θ 。再次使用 CORDIC 核来求得 x' 和旋转角 φ ;
 - 2) 计算旋转角 θ 和 φ 的正余弦值, 并计算 c, s 。
- 旋转单元 FPGA 模块化如图 3 所示。

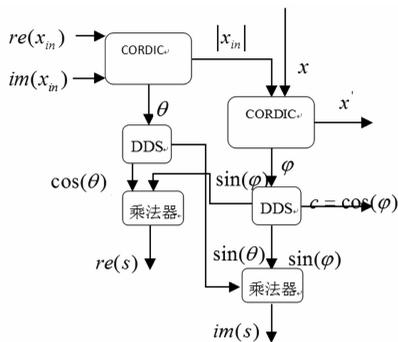


图 3 旋转单元的 FPGA 模块化框图

IQRD-SMI 脉动阵列结构的完整 FPGA 实现同图 3 所示, 在顶层模块中通过时序、寄存器和控制口来操控数据的采集处理节拍与流向。

3.3 改良的复用 FPGA 实现结构 (GR-ITSA, Givens rotation - Improved triangular systolic array)

FPGA 芯片内部集成了常用的硬件资源, 如 DSP Slices、乘法器等, 但是这些资源的数量有限。传统 systolic 阵列结构会随着阵元数的增加而大大消耗硬件资源。为了节约成本、降低硬件资源消耗, 在传统 IQRD-SMI 算法的 systolic 阵列上进行改进, 得到了低消耗的复用脉动阵结构 GR-ITSA, 如图 4 所示。

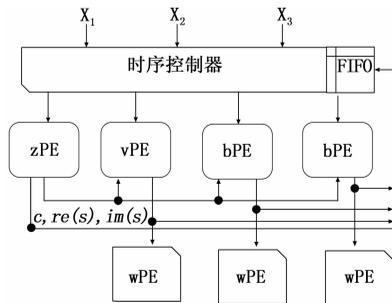


图 4 GR-ITSA 复用脉动阵列结构

关键的修改是在原阵列结构的基础上增加了时序控制器和用于存储中间量的 FIFO。各 PE 单元实现的基本功能与传统结构一致, 时序控制单元主要调度各 PE 的数据采集处理节拍, 并含有寄存器 (Block RAM), 用来接收和传递相应数据到各 PE 端口。

时序控制单元包含 5 种先入先出数据寄存器 (FIFO): FIFO_Data、FIFO_bPE、FIFO_CS、FIFO_vPE、FIFO_wPE。FIFO_Data 由 M 对宽度为 L (M 为阵元个数, L 为量化后的接收数据位宽)、深度为 N (N 为采样快拍总数) 的 FIFO 构成, 用来缓存接收到的采样数据, 并将数据提供给 bPE。FIFO_bPE、FIFO_vPE、FIFO_wPE 由深度为 N 的 FIFO 构成, 用来接收各个 PE 单元的输出数据并为下一次的计算提供数据。FIFO_CS 由 3 个 FIFO 构成, 用来缓存 bPE 单元计算输出的 C, S , 并将 C, S 送至 bPE 和 vPE 单元。

5 种 FIFO 与 4 类 PE 单元完成一次数据阵的 QR 分解需多个步骤, 期间 4 类 PE 单元和 5 种堆栈在不同的时间段被调用, 实现硬件资源共享。

4 IQRD-SMI 算法的软硬件仿真及误差分析

4.1 仿真结果

首先使用 Matlab 模拟生成信号源, 通过 Verilog I/O 接口文件将信号数据输入到 ISE 程序中, 并在 Modesim 软件下进行 FPGA 定点仿真验证, 最后将输出结果返回 Matlab, 同 Matlab 中的浮点仿真计算结果进行比对。仿真参数如表 1 所示, 仿真结果如图 5 所示。

表 1 仿真参数设计

阵元数	采样数	期望方向	信噪比	信干比
$M=4$	$N=8\ 000$	$\theta=50$	$SNR=20\text{ dB}$	$SIR=-30\text{ dB}$

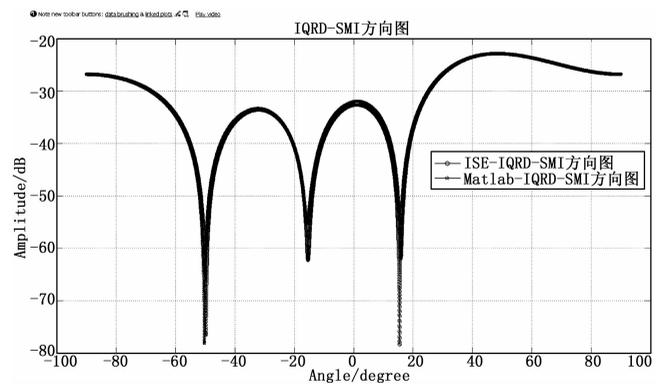


图 5 ISE 与 Matlab 仿真结果对比图

IQRD-SMI 算法在 Matlab 软件浮点仿真与 FPGA 硬件定点仿真的对比波束如图 6 所示, 仿真采用的采样数为 64 次快拍, 满足基本采样数大于 $2M$ 的条件。通过对比图可知, 改进的 GR-ITSA 结构实现了逆 QR 分解自适应波束形成算法, 权向量得以高效实时的获取, 在波束的期望方向 $\theta = 50^\circ$ 处, 算法对信号的增益值达到最大。

4.2 误差分析与资源消耗

为了得到该算法的平均误差百分比, 将复用结构 FPGA 定点计算与 MATLAB 浮点计算的结果引入误差计算公式 (3), 取算术平均值作为最后的结果。

(下转第 246 页)