

# 基于可重构计算系统的矩阵三角化分解硬件并行结构研究

刘书勇, 吴艳霞, 张博为, 张国印, 戴 葵

(哈尔滨工程大学计算机科学与技术学院, 黑龙江哈尔滨 150001)

**摘 要:** 可重构计算系统成为加速计算密集型应用的重要选择之一. 在众多受到关注的计算密集型问题中, 矩阵三角化分解作为典型的基础类应用始终处于研究的核心地位, 在求解线性方程组、求矩阵特征值等科学与工程问题中有重要的研究价值. 本文面向矩阵三角化分解中共有的三角化计算过程, 通过分析该过程的线性计算规律, 提出一种适于硬件并行实现的子矩阵更新同一化算法及矩阵三角化计算 FPGA (Field Programmable Gate Array) 并行结构. 针对 LU 矩阵三角化分解在并行结构模板上的高性能实现及优化方法开展了研究. 理论分析表明, 该算法针对矩阵三角化计算过程具有更高的数据并行性与流水并行性; 实验结果表明, 与通用处理器的软件实现相比, 根据该算法实现的矩阵三角化分解 FPGA 并行结果在关键计算性能上可以取得 10 倍以上的加速比.

**关键词:** 矩阵三角化分解; 三角化过程; 并行算法; LU 分解; 现场可编程门阵列

**中图分类号:** TP302.1      **文献标识码:** A      **文章编号:** 0372-2112 (2015)08-1642-09

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

**DOI:** 10.3969/j.issn.0372-2112.2015.08.026

## Research of Parallel Hardware Architecture for Matrix Triangularization Decomposition Based on Reconfigurable Computing System

LIU Shu-yong, WU Yan-xia, ZHANG Bo-wei, ZHANG Guo-yin, DAI Kui

(College of Computer Science and Technology, Harbin Engineering University, Harbin, Heilongjiang 150001, China)

**Abstract:** The reconfigurable computing system became an important choice according to accelerating compute-intensive applications. Among most compute-intensive applications, the matrix triangularization decomposition always was in the central position of research subjects and presented a great value to solve linear equation systems and matrix eigenvalue problems in science or engineering area. This paper analyzed the linear computing process of triangularization and proposed a hardware-adaptive parallel sub-matrix identity updating algorithm and a high-performance parallel structure hardware template for matrix triangularization on FPGA (Field Programmable Gate Array) according to the common triangularization computing process of the matrix triangularization decomposition. The research focused on the high-performance FPGA parallel structure implementation and optimization methods for the LU matrix triangularization decomposition. In theoretical analysis, the proposed algorithm presents better pipeline-parallelism and data-parallelism during the matrix triangularization process. The experimental result shows that the proposed structure gets over decuple speedup compared to general-purpose processors and the previous works in vital performance.

**Key words:** matrix triangularization decomposition; triangularization process; parallel algorithm; LU decomposition; field programmable gate array

## 1 引言

随着半导体制造工艺的进步, FPGA (Field Programmable Gate Array) 迅速发展, 基于 FPGA 的可重构计算系统以其兼顾通用处理器 (General Purpose Processor, GPP) 和专用集成电路 (Application Specific Integrated Circuit, ASIC) 的优点, 在众多计算密集型的领域中被广泛设计与应用, 使 FPGA 可重构计算系统成为加速科学计算的一种重要的选择<sup>[1]</sup>. 在众多计算密集型问题中, 矩

阵计算处于核心地位, 大部分科学和工程问题最终都要归结成为矩阵计算问题<sup>[2~4]</sup>, 而大规模矩阵计算问题被认为是其中最具挑战性之一.

国外方面, 美国南加州大学 (University of Southern California) 的一个研究小组 (<http://halcyon.usc.edu/~pk/prasannawebsite/index.php>) 从 2002 年开始研究基于 FPGA 矩阵运算的算法和结构<sup>[5,6]</sup>, 针对降低功耗开展研究<sup>[7]</sup>, 对提出的矩阵乘法结构进行优化与参数化<sup>[8]</sup>, 使用软硬件协同设计的方法实现了矩阵的 LU 分解<sup>[9]</sup>.

英国伦敦帝国理工学院(Imperial College)康斯坦丁德斯教授(Prof. Constantinides)领导的研究小组从 2008 年开始 FPGA 矩阵技术方面的研究,该小组分别提出单精度 MINRES 算法求解线性方程矩阵<sup>[10]</sup>和 CG 算法求解稠密线性方程组的并行结构<sup>[11]</sup>的 FPGA 实现.2009 年,他们围绕迭代法中迭代次数和计算精度问题提出的加速 MPC(Model Predictive Control)中大量小规模线性方程组求解问题.通过探索设计空间,平衡迭代次数和计算精度,在双精度情况下,计算速度达到软件实现的 7 倍<sup>[12]</sup>.另外,还有许多代表性研究机构在 FPGA 矩阵计算与矩阵分解研究上取得成果,在此不逐一描述.

国内方面,国防科大窦勇教授领导的研究小组在 FPGA 科学计算上取得了较多成果.该研究小组在 2005 年提出了一种分块矩阵乘法并行算法和对应的线性阵列结构,该结构可以进行扩展以实现任意规模的矩阵乘法<sup>[13]</sup>.2009 年,该小组又提出用以处理大规模矩阵求逆的结构,并在其具有代表性的 LEAP 细粒度可重构并行结构框架下实现矩阵 QR 分解的三种算法,并试图将该结构框架扩展到所有矩阵分解中<sup>[14]</sup>.2010 年又提出了高精度浮点(128 位和 256 位)运算部件,并应用到 LU 分解中,在数值稳定方面取得不错的效果.

已有的矩阵分解的加速硬件在设计目的上分布在通用化与专用化的两端,在特定的需求范畴内无法兼顾一般性与计算效率.为此,本文基于可重构计算系统,面向矩阵三角化分解中共有的三角化计算过程,通过分析三角化计算迭代过程的子矩阵更新过程的线性规律,提出一种具有一般性的子矩阵更新同一化并行算法,进而提出基于该算法的计算复杂度较低的矩阵三角化分解并行结构,针对 LU 矩阵三角化分解在并行结构上的高性能实现及优化方法开展了研究.

## 2 子矩阵更新同一化并行算法

矩阵分解可分为单个矩阵分解和矩阵束分解两类.单个矩阵分解可分为对角化分解、三角化分解和三角-对角化分解.矩阵三角化分解将矩阵  $A$  分解为正交矩阵与三角矩阵之积,或分解为一个上三角矩阵与一个下三角矩阵之积,主要有以下三种形式:(1)Cholesky 分解:  $A = GG^T$ ,其中,  $G$  为下三角矩阵(针对对称正定矩阵的三角化分解);(2)QR 分解:  $A = QR$  或  $Q^T A = R$ ,其中,  $Q$  是正交矩阵,而  $R$  是上三角矩阵;(3)LU 分解:  $A = LU$ ,其中,  $L$  是单位下三角矩阵,而  $U$  是上三角矩阵.本文针对 LU 矩阵三角化分解在并行结构模板上的高性能实现及优化方法进行了应用研究.

### 2.1 矩阵三角化计算的子矩阵更新变换

在矩阵三角化计算中,若  $n$  阶方阵  $A$  行向量  $a_i$  ( $1 \leq i < n$ ) 满足  $a_{ij} \neq 0$ ,取行向量  $a_k$  ( $1 \leq k < n, k \neq i$ ) 与

$a_i$  的某种线性组合,以 2 阶矩阵  $M_{k,i,j}$  表示,令  $a_{kj}$  输出零值并更新  $a_k$ ,则称此线性运算为一次行消元过程.称  $M_{k,i,j}$  为行消元变换(矩阵).式(1)给出了当  $i = j = 1$  时的行消元过程.对矩阵  $A$  所有满足条件的  $a_k$  进行行消元过程,可以获得式(2)表示的矩阵  $A'$ .

$$\begin{pmatrix} a_{11} & a_{12} \cdots a_{1n} & b_1 \\ 0 & t_{k2} \cdots t_{kn} & b'_k \end{pmatrix} = M_{k,1,1} \begin{pmatrix} a_{11} & a_{12} \cdots a_{1n} & b_1 \\ a_{k1} & a_{k2} \cdots a_{kn} & b_k \end{pmatrix},$$

$$M_{k,1,1} = \begin{pmatrix} 1 & 0 \\ -a_{k1}/a_{11} & 1 \end{pmatrix} \quad (1)$$

$$A' = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ 0 & a'_{22} & a'_{23} & a'_{24} & b'_2 \\ 0 & a'_{32} & a'_{33} & a'_{34} & b'_3 \\ 0 & a'_{42} & a'_{43} & a'_{44} & b'_4 \end{pmatrix} \quad (2)$$

在行消元过程的基础上,可以推导得到整个矩阵三角化的线性计算方程.设  $n$  阶方阵为  $A$ ,对矩阵  $A$  所有满足条件的  $a_k$  进行行消元过程的结果方阵为  $A'$ ,称此线性计算为一次子矩阵更新过程.其中,  $A'$  中的行向量是  $A$  中行向量关于矩阵  $T_{i,j}$  列系数的线性组合,称  $T_{i,j}$  为关于主元  $a_{i,j}$  的子矩阵更新变换(矩阵).取  $n = 4$ ,矩阵  $A$  子矩阵更新过程如式(3)所示.这一过程也称为由子矩阵更新变换矩阵  $T_{i,j}$  所确定的线性变换.

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ 0 & a'_{22} & a'_{23} & a'_{24} & b'_2 \\ 0 & a'_{32} & a'_{33} & a'_{34} & b'_3 \\ 0 & a'_{42} & a'_{43} & a'_{44} & b'_4 \end{pmatrix} = T_{1,1} \cdot \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ a_{21} & a_{22} & a_{23} & a_{24} & b_2 \\ a_{31} & a_{32} & a_{33} & a_{34} & b_3 \\ a_{41} & a_{42} & a_{43} & a_{44} & b_4 \end{pmatrix}, \quad (3)$$

$$T_{1,1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -a_{21}/a_{11} & 1 & 0 & 0 \\ -a_{31}/a_{11} & 0 & 1 & 0 \\ -a_{41}/a_{11} & 0 & 0 & 1 \end{pmatrix}$$

继续在一次子矩阵更新过程的结果矩阵  $A'$  中递归地进行子矩阵更新变换,可以得到三角矩阵  $T$ .式(4)中描述了获得三角矩阵的整个计算过程,同时给出每次子矩阵更新变换  $\{T_{1,1}, T_{2,2}, T_{3,3}\}$ .

$$T = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ 0 & t_{22} & t_{23} & t_{24} & b'_2 \\ 0 & 0 & t_{33} & t_{34} & b'_3 \\ 0 & 0 & 0 & t_{44} & b'_4 \end{pmatrix} = T_{3,3} \cdot T_{2,2} \cdot T_{1,1} \cdot \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ a_{21} & a_{22} & a_{23} & a_{24} & b_2 \\ a_{31} & a_{32} & a_{33} & a_{34} & b_3 \\ a_{41} & a_{42} & a_{43} & a_{44} & b_4 \end{pmatrix},$$

$$\begin{aligned}
 T_{1,1} &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ -a_{21}/a_{11} & 1 & 0 & 0 \\ -a_{31}/a_{11} & 0 & 1 & 0 \\ -a_{41}/a_{11} & 0 & 0 & 1 \end{pmatrix}, \\
 T_{2,2} &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -a'_{32}/a'_{22} & 1 & 0 \\ 0 & -a'_{42}/a'_{22} & 0 & 1 \end{pmatrix}, \\
 T_{3,3} &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -a''_{43}/a''_{33} & 1 \end{pmatrix} \quad (4)
 \end{aligned}$$

$\{T_{1,1}, T_{2,2}, T_{3,3}\}$  代表每次子矩阵更新过程的计算特征, 可以通过分析  $T_{i,j}$  中向量组的线性相关性来获得计算的特征信息. 以消元变换  $T_{1,1}$  为例, 考察一次子矩阵更新过程的计算特征: 根据向量组线性相关的定义, 对其中 4 维行向量组  $t_1 = (1\ 0\ 0\ 0)$ ,  $t_2 = (-a_{21}/a_{11}\ 1\ 0\ 0)$ ,  $t_3 = (-a_{31}/a_{11}\ 0\ 1\ 0)$ ,  $t_4 = (-a_{41}/a_{11}\ 0\ 0\ 1)$ , 发现不存在一组不全为 0 的数  $k_1, k_2, k_3, k_4$  使得  $k_1 t_1 + k_2 t_2 + k_3 t_3 + k_4 t_4 = 0$ , 故  $t_1, t_2, t_3, t_4$  线性无关. 而且观察发现,  $t_1$  为单位行变换  $(1\ 0\ 0\ 0)$ , 即  $a'_1 = a_1$ ;  $t_2, t_3, t_4$  组成的矩阵  $(t_2, t_3, t_4)^T$  可以看成列向量  $(-a_{21}/a_{11} - a_{31}/a_{11} - a_{41}/a_{11})^T$  与 3 阶单位阵组成的矩阵. 矩阵  $A$  与消元变换  $T_{1,1}$  的线性组合只完成向量  $(-a_{21}/a_{11} - a_{31}/a_{11} - a_{41}/a_{11})^T$  与  $a_1$  求积, 再与  $(a_2\ a_3\ a_4)^T$  相加的行间两两计算. 称  $(-a_{21}/a_{11} - a_{31}/a_{11} - a_{41}/a_{11})^T$  为行变换关键向量. 同理, 通过矩阵初等变换可以得到,  $(-a_{32}'/a_{22}' - a_{42}'/a_{22}')^T$  和  $(-a_{43}''/a_{33}'')^T$  也为行变换关键向量. 而以上结论亦可以在  $T_{2,2}$  和  $T_{3,3}$  中推导得到.

### 2.2 子矩阵更新同一化并行算法

以求解 GF(2) 域上线性方程组的算法为例说明子矩阵更新同一化并行算法的推导过程. 在 GF(2) 域中, 模 2 加法 “+” 等同于布尔运算当中的布尔异或运算 (XOR), 文中用 “+” 表示; 而模 2 乘法 “×” 等同于布尔运算中的与运算 (AND), 用 “∧” 表示. 该算法矩阵三角化过程子矩阵更新过程 ( $n = 4$ ) 可以表示为式 (5) 和式 (6), 其中  $a_{ij} \in \{0, 1\}$ .

$$\begin{aligned}
 T &= \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ 0 & t_{22} & t_{23} & t_{24} & b'_2 \\ 0 & 0 & t_{33} & t_{34} & b'_3 \\ 0 & 0 & 0 & t_{44} & b'_4 \end{pmatrix} \\
 &= T_{3,3} \cdot T_{2,2} \cdot T_{1,1} \cdot \begin{pmatrix} a_{11} & a_{12} & a_{11} & a_{14} & b_1 \\ a_{21} & a_{22} & a_{22} & a_{24} & b_2 \\ a_{31} & a_{32} & a_{33} & a_{34} & b_3 \\ a_{41} & a_{32} & a_{32} & a_{32} & b_4 \end{pmatrix},
 \end{aligned}$$

$$\begin{aligned}
 T_{1,1} &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ a_{21} & 1 & 0 & 0 \\ a_{31} & 0 & 1 & 0 \\ a_{41} & 0 & 0 & 1 \end{pmatrix}, T_{2,2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & a'_{32} & 1 & 0 \\ 0 & a'_{42} & 0 & 1 \end{pmatrix}, \\
 T_{3,3} &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & a''_{43} & 1 \end{pmatrix} \quad (5)
 \end{aligned}$$

求解线性方程组的后向回代过程可以看成是三角化过程的继续. 与三角化过程类似, 后向回代过程显然可以表示为式 (6). 其中  $D_{4,4}, D_{3,3}$  和  $D_{2,2}$  为后向回代变换矩阵. 若考虑系数矩阵  $A$  为非奇异的情况, 则必有  $d_{11} = d_{22} = d_{33} = d_{44} = 1$ , 因此未知项  $(x_1, x_2, x_3, x_4)^T$  的解即为  $(b'_1, b'_2, b'_3, b'_4)^T$ .

$$\begin{aligned}
 D &= \begin{pmatrix} d_{11} & 0 & 0 & 0 & b'_1 \\ 0 & d_{22} & 0 & 0 & b'_2 \\ 0 & 0 & d_{33} & 0 & b'_3 \\ 0 & 0 & 0 & d_{44} & b'_4 \end{pmatrix} \\
 &= D_{2,2} \cdot D_{3,3} \cdot D_{4,4} \cdot \begin{pmatrix} t_{11} & t_{12} & t_{11} & t_{14} & b_1 \\ 0 & t_{22} & t_{22} & t_{24} & b_2 \\ 0 & 0 & t_{33} & t_{34} & b_3 \\ 0 & 0 & 0 & t_{44} & b_4 \end{pmatrix}, \quad (6) \\
 D_{4,4} &= \begin{pmatrix} 1 & 0 & 0 & t_{14} \\ 0 & 1 & 0 & t_{24} \\ 0 & 0 & 1 & t_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix}, D_{3,3} = \begin{pmatrix} 1 & 0 & t'_{13} & 0 \\ 0 & 1 & t'_{23} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \\
 D_{2,2} &= \begin{pmatrix} 1 & t''_{12} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}
 \end{aligned}$$

通过合并某些消元变换矩阵和后向回代变换矩阵中的行变换关键向量来同时完成矩阵的三角化过程与后向回代过程. 一种直观的策略是, 合并具有相同主元的消元变换矩阵和后向回代变换矩阵的行变换关键向量. 在本例中合并的变换矩阵对为  $\{T_{2,2}, D_{2,2}\}$  和  $\{T_{3,3}, D_{3,3}\}$ . 改变后的变换矩阵如式 (7) 所示.

$$T'_{2,2} = \begin{pmatrix} 1 & a'_{12} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & a'_{32} & 1 & 0 \\ 0 & a'_{42} & 0 & 1 \end{pmatrix}, T'_{3,3} = \begin{pmatrix} 1 & 0 & a''_{13} & 0 \\ 0 & 1 & a''_{23} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & a''_{43} & 1 \end{pmatrix} \quad (7)$$

将这个扩展的矩阵三角化过程称为矩阵单位对角化过程 ( $d_{11} = d_{22} = d_{33} = d_{44} = 1$ ). 该过程可以描述为式 (8) 的计算.

$$\begin{aligned}
 \mathbf{D} &= \begin{pmatrix} d_{11} & 0 & 0 & 0 & b'_1 \\ 0 & d_{22} & 0 & 0 & b'_2 \\ 0 & 0 & d_{33} & 0 & b'_3 \\ 0 & 0 & 0 & d_{44} & b'_4 \end{pmatrix} \\
 &= \mathbf{T}'_{4,4} \cdot \mathbf{T}'_{3,3} \cdot \mathbf{T}'_{2,2} \cdot \mathbf{T}_{1,1} \cdot \begin{pmatrix} a_{11} & a_{12} & a_{11} & a_{14} & b_1 \\ a_{21} & a_{22} & a_{22} & a_{24} & b_2 \\ a_{31} & a_{32} & a_{33} & a_{34} & b_3 \\ a_{41} & a_{32} & a_{32} & a_{32} & b_4 \end{pmatrix}, \\
 \mathbf{T}_{1,1} &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ a_{21} & 1 & 0 & 0 \\ a_{31} & 0 & 1 & 0 \\ a_{41} & 0 & 0 & 1 \end{pmatrix}, \mathbf{T}'_{2,2} = \begin{pmatrix} 1 & a'_{12} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & a'_{32} & 1 & 0 \\ 0 & a'_{42} & 0 & 1 \end{pmatrix} \\
 \mathbf{T}'_{3,3} &= \begin{pmatrix} 1 & 0 & a''_{13} & 0 \\ 0 & 1 & a''_{23} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & a''_{43} & 1 \end{pmatrix}, \\
 \mathbf{T}'_{4,4} = \mathbf{D}_{4,4} &= \begin{pmatrix} 1 & 0 & 0 & a'''_{14} \\ 0 & 1 & 0 & a'''_{24} \\ 0 & 0 & 1 & a'''_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (8)
 \end{aligned}$$

通过初等变换,所有的子矩阵更新变换矩阵可以同一化,使得在构建算法映射到阵列结构时能实现计算单元计算步骤操作的同一化,进而实现计算的高度并行化.设计如下的子矩阵更新同一化并行算法:

**步骤 1** 引入置换矩阵  $\mathbf{P}$ , 令

$$\mathbf{P} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \quad (9)$$

**步骤 2** 通过  $\mathbf{T}_{1,1}$  左乘置换矩阵  $\mathbf{P}$  的初等变换,获得新的消元变换矩阵,称为  $\mathbf{T}_d$ .

$$\mathbf{T}_d = \begin{pmatrix} a_{21} & 1 & 0 & 0 \\ a_{31} & 0 & 1 & 0 \\ a_{41} & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} = \mathbf{P} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ a_{21} & 1 & 0 & 0 \\ a_{31} & 0 & 1 & 0 \\ a_{41} & 0 & 0 & 1 \end{pmatrix} \quad (10)$$

**步骤 3** 通过  $\mathbf{A}$  右乘置换矩阵  $\mathbf{P}$  的初等变换,获得新的系数矩阵,称为  $\mathbf{Z}$ .

$$\mathbf{Z} = \begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} & z_{15} \\ z_{21} & z_{22} & z_{23} & z_{24} & z_{25} \\ z_{31} & z_{32} & z_{33} & z_{34} & z_{35} \\ z_{41} & z_{42} & z_{43} & z_{44} & z_{45} \end{pmatrix} = \begin{pmatrix} a_{12} & a_{13} & a_{14} & b_1 & a_{11} \\ a_{22} & a_{23} & a_{24} & b_2 & a_{21} \\ a_{32} & a_{33} & a_{34} & b_3 & a_{31} \\ a_{42} & a_{43} & a_{34} & b_4 & a_{41} \end{pmatrix}$$

$$= \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ a_{21} & a_{22} & a_{23} & a_{24} & b_2 \\ a_{31} & a_{32} & a_{33} & a_{34} & b_3 \\ a_{41} & a_{42} & a_{43} & a_{44} & b_4 \end{pmatrix} \cdot \mathbf{P} \quad (11)$$

将  $z_{ij}$  看成更新后的  $a_{ij}$ . 设:  $z_{i-1,j-1}^{(t)} = g^{(t)}(i, j, k) = z_{ij}^{(t-1)} \oplus (z_{i,i-1}^{(t-1)} \wedge z_{1,j}^{(t-1)})$ ,  $\mathbf{Z}^{(0)} = \mathbf{Z}$ ,  $1 \leq k \leq n$ ,  $1 < i, j \leq n$  且  $0 < t \leq n$ . 则一次同一化的矩阵子矩阵更新的过程可以表示为式(12).

**步骤 4** 在每次子矩阵更新后递归地执行步骤 3, 即可以在不需要使用  $\mathbf{T}_d$  外的消元变换矩阵的情况下经过  $n$  次子矩阵更新完成矩阵的单位对角化. 该过程亦可以表示为式(13),  $0 < t \leq n$ .

$$\mathbf{Z}^{(1)} = \begin{pmatrix} g^{(1)}(2,2,1) & g^{(1)}(2,3,1) & g^{(1)}(2,4,1) & g^{(1)}(2,5,1) & z_{14}^{(1)} = 0 \\ g^{(1)}(3,2,1) & g^{(1)}(3,3,1) & g^{(1)}(3,4,1) & g^{(1)}(3,5,1) & z_{24}^{(1)} = 0 \\ g^{(1)}(4,2,1) & g^{(1)}(4,3,1) & g^{(1)}(4,4,1) & g^{(1)}(4,5,1) & z_{34}^{(1)} = 0 \\ z_{41}^{(1)} = z_{12} & z_{42}^{(1)} = z_{13} & z_{43}^{(1)} = z_{14} & z_{44}^{(1)} = z_{15} & z_{44}^{(1)} = 1 \end{pmatrix} = \mathbf{T}_d \cdot \mathbf{Z} \quad (12)$$

$$\begin{cases} \mathbf{D}^{(t)} = \mathbf{P} \cdot (\mathbf{T}_{1,1} \cdot \mathbf{A}^{(t-1)}) \cdot \mathbf{P} \\ \mathbf{A}^{(0)} = \mathbf{A} \end{cases} \quad (13)$$

### 3 算法针对 LU 分解的应用及 FPGA 并行结构设计

LU 分解将  $n \times n$  矩阵  $\mathbf{A}(a_{ij}, 1 \leq i, j \leq n)$  分解成  $\mathbf{A} = \mathbf{L}\mathbf{U}$  的形式,其中  $\mathbf{L}(l_{ij}, 1 \leq j \leq i \leq n)$  为  $n$  阶单位下三角矩阵(对角元素为 1),  $\mathbf{U}(u_{ij}, 1 \leq j \leq i \leq n)$  为  $n$  阶上三角矩阵,如式(14)所示.

$$\mathbf{L} = \begin{bmatrix} 1 & & & & & \\ l_{21} & 1 & & & & \\ \dots & \dots & \dots & & & \\ l_{k1} & l_{k2} & \dots & 1 & & \\ \dots & \dots & \dots & \dots & \dots & \\ l_{n1} & l_{n2} & \dots & \dots & l_{n,n-1} & 1 \end{bmatrix}, \quad (14)$$

$$\mathbf{U} = \begin{bmatrix} u_{11} & u_{21} & \dots & u_{k1} & \dots & u_{n1} \\ & u_{22} & \dots & u_{k2} & \dots & u_{n2} \\ & & \dots & \dots & \dots & \dots \\ & & & u_{kk} & \dots & u_{nk} \\ & & & & \dots & \dots \\ & & & & & u_{nn} \end{bmatrix}$$

$\mathbf{L}(l_{ij})$  矩阵与  $\mathbf{U}(u_{ij})$  矩阵可以通过式(15)与式(16)求得.

$$l_{ij} = \begin{cases} i > j = 1: a_{ij}/u_{ij} \\ i > j > 1: (a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj})/u_{ij} \end{cases} \quad (15)$$

$$u_{ij} = \begin{cases} 1 & i \leq j: a_{ij} \\ 1 < i \leq j: a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj} \end{cases} \quad (16)$$

将  $z_{ij}$  看成更新后的  $a_{ij}$ . 设:  $z_{i-1,j-1}^{(t)} = g^{(t)}(i, j, k) = z_{ij}^{(t-1)} - (z_{i,1}^{(t-1)}/z_{11}^{(t-1)}) \cdot z_{1,j-1}^{(t-1)}$ ,  $1 \leq k \leq n$ ,  $1 < i, j \leq n$ ,  $0 < t \leq n$ ,  $\mathbf{A}^{(0)} = \mathbf{A}$ , 即:  $g^{(t)}(i, j, k) = z_{ij}^{(t-1)} - l_{ik} \cdot u_{kj}$ . 取  $n = 4$ , 则一次 LU 分解子矩阵的更新过程可以表示为式 (17). 全部的 LU 分解过程由式 (17) ~ (19) 表示的子矩阵更新过程组成.

$$\mathbf{A}_{4 \times 4}^{(0)} = \begin{pmatrix} z_{11}^{(0)} & z_{12}^{(0)} & z_{13}^{(0)} & z_{14}^{(0)} \\ z_{21}^{(0)} & z_{22}^{(0)} & z_{23}^{(0)} & z_{24}^{(0)} \\ z_{31}^{(0)} & z_{32}^{(0)} & z_{33}^{(0)} & z_{34}^{(0)} \\ z_{41}^{(0)} & z_{42}^{(0)} & z_{43}^{(0)} & z_{44}^{(0)} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix},$$

$$\mathbf{l}_1 = (z_{21}^{(0)}/z_{11}^{(0)} \quad z_{31}^{(0)}/z_{11}^{(0)} \quad z_{41}^{(0)}/z_{11}^{(0)})^T$$

$$\mathbf{u}_1 = (z_{12}^{(0)} \quad z_{13}^{(0)} \quad z_{14}^{(0)}) \quad (17)$$

$$\mathbf{A}_{3 \times 3}^{(1)} = \begin{pmatrix} z_{11}^{(1)} & z_{12}^{(1)} & z_{13}^{(1)} \\ z_{21}^{(1)} & z_{22}^{(1)} & z_{23}^{(1)} \\ z_{31}^{(1)} & z_{32}^{(1)} & z_{33}^{(1)} \end{pmatrix}$$

$$= \begin{pmatrix} g^{(1)}(2,2,1) & g^{(1)}(2,3,1) & g^{(1)}(2,4,1) \times \\ g^{(1)}(2,3,1) & g^{(1)}(3,3,1) & g^{(1)}(3,4,1) \times \\ g^{(1)}(2,4,1) & g^{(1)}(3,4,1) & g^{(1)}(4,4,1) \times \\ \times & \times & \times \end{pmatrix},$$

$$\mathbf{l}_2 = (z_{21}^{(1)}/z_{11}^{(1)} \quad z_{31}^{(1)}/z_{11}^{(1)})^T, \mathbf{u}_2 = (z_{12}^{(1)} \quad z_{13}^{(1)}) \quad (18)$$

$$\mathbf{A}_{2 \times 2}^{(2)} = \begin{pmatrix} z_{11}^{(2)} & z_{12}^{(2)} \\ z_{21}^{(2)} & z_{22}^{(2)} \end{pmatrix} = \begin{pmatrix} g^{(2)}(2,2,2) & g^{(2)}(2,3,2) \times \\ g^{(2)}(2,3,2) & g^{(2)}(3,3,2) \times \\ \times & \times \end{pmatrix},$$

$$\mathbf{l}_3 = (z_{21}^{(2)}/z_{11}^{(2)}), \mathbf{u}_3 = (z_{12}^{(2)})$$

$$\mathbf{A}_{1 \times 1}^{(3)} = g^{(3)}(2,2,3) \quad (19)$$

LU 分解的计算依赖图如图 1 所示. 因每一趟子矩阵更新均对完整的  $n$  阶伴随矩阵操作, 故该依赖图仅给出了两趟矩阵消元间计算依赖关系. 对应的 LU 分解并行结构 PE 的配置如图 2 所示, LU 分解阵列结构如图 3 所示.

图 3 描述的 LU 分解阵列结构工作流程如下:

(1) 选主元与阵列初始化

在设计阵列结构中, 以 IEEE-754 单精度浮点数为例, 每个 PE 均通过 34-bits 位宽的输入端与其正上方 PE 相连, 故选主元操作其实为 shift-down 操作. 因为是循环(cyclic)操作, 所以二者可以等同. 定义一种行状态标识 used-flag, 该值为 ‘1’ 时表示该行已作为主行参加过矩阵消元. 在判断  $P_{1,1}$  的 BcPE 中 REG1 的值接近零时, 所有 used-flag 为 ‘0’ 的行将在下一个时钟周期

发起一次选主元操作: 将 PE (BcPE 中 REG1, IcPE 中 REG2, rPE 中 REG0) 中寄存的浮点数发送到正下方的 PE 中, 同时循环接收上方 PE 送来的数; 所有 used-flag 为 ‘1’ 的行, PE 中的 MUX 选通 used-flag 旁路 (bypass for used-flag). 选主元操作的具体硬件过程同时可以用来初始化整个矩阵分解的并行结构: 首先, 对阵列结构中所有 PE 置位清零; 然后, 通过 shift-down 操作, 阵列首行从 Smart Buffer 中获取线性方程组增广 (系数) 矩阵的行向量并下移至下方阵列行中. 以此类推, 经过  $n$  个时钟周期增广 (系数) 矩阵可被装载到阵列中, 从而完成整个阵列计算的初始化.

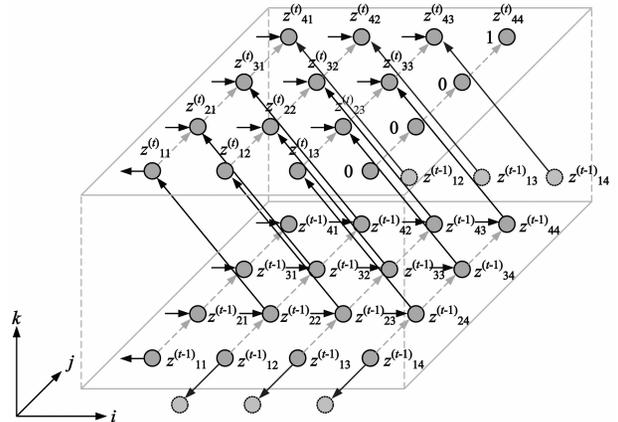


图 1 LU 分解的计算依赖图

(2) 子矩阵更新操作

**步骤 1** 阵列的并行结构在列方向开展流水并行, 在行方向开展数据并行.

**步骤 2** 在列向量  $(a_{2,1}, \dots, a_{n,1})$  移动进入 BcPE 的同时, 与各行消元参考变量  $a_{k,1}$  对应的行向量  $\mathbf{a}_k$  也通过 rPE 向上的传递依次进入 IcPE. 在 IcPE 中通过 FIFO 对浮点乘法-减法运算与除法运算结果  $l_{k,1}$  进行输入同步. 计算结果依次传递给下方的 rPE 步进移动,  $n-1$  个节拍后, 更新完的  $a_{i,j}^{(1)}$  仍旧回到位置为  $P_{i,j}$  的 PE 中.

**步骤 3** 将所有子矩阵更新操作得到的结果再通过一个周期经过斜角邻接数据通道更新到左上方的 rPE 或 cPE 中. 对得到的单位系数阵再做一次浮点除法求得未知项  $x$ , 完成  $n$  阶实系数线性方程组的求解, 结果可通过对第一列的线性输出获得.

表 1 LU 分解并行结构中 PE 单元的综合结果

	BcPE-1	IcPE	rPE
Number of Slice Registers	1167	341	38
Number of Slice LUTs	1229	858	41
Number of LUT Flip Flop Pairs	1239	877	43
Number of DSP48Es	0	2	0
Max Frequency (MHz)	198.01	160.31	549.83

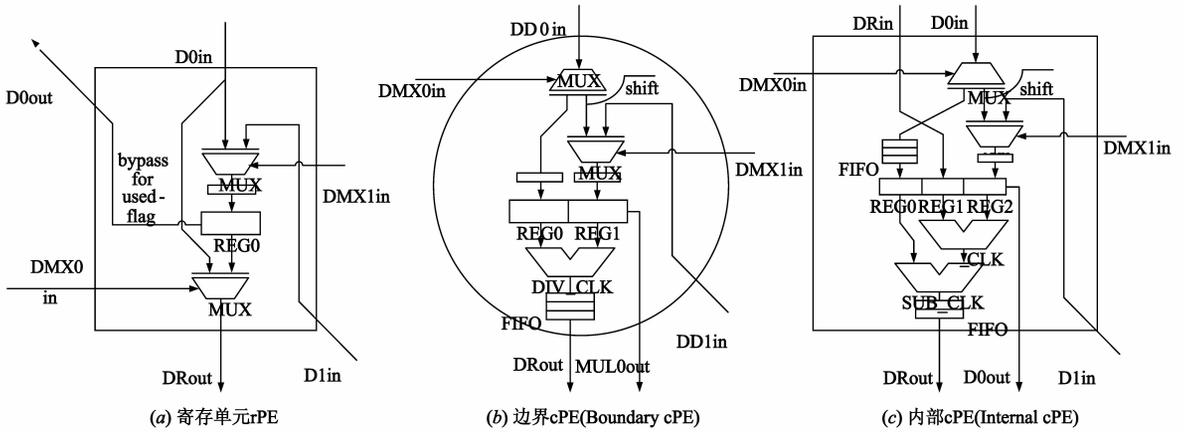


图2 LU分解并行结构PE的配置

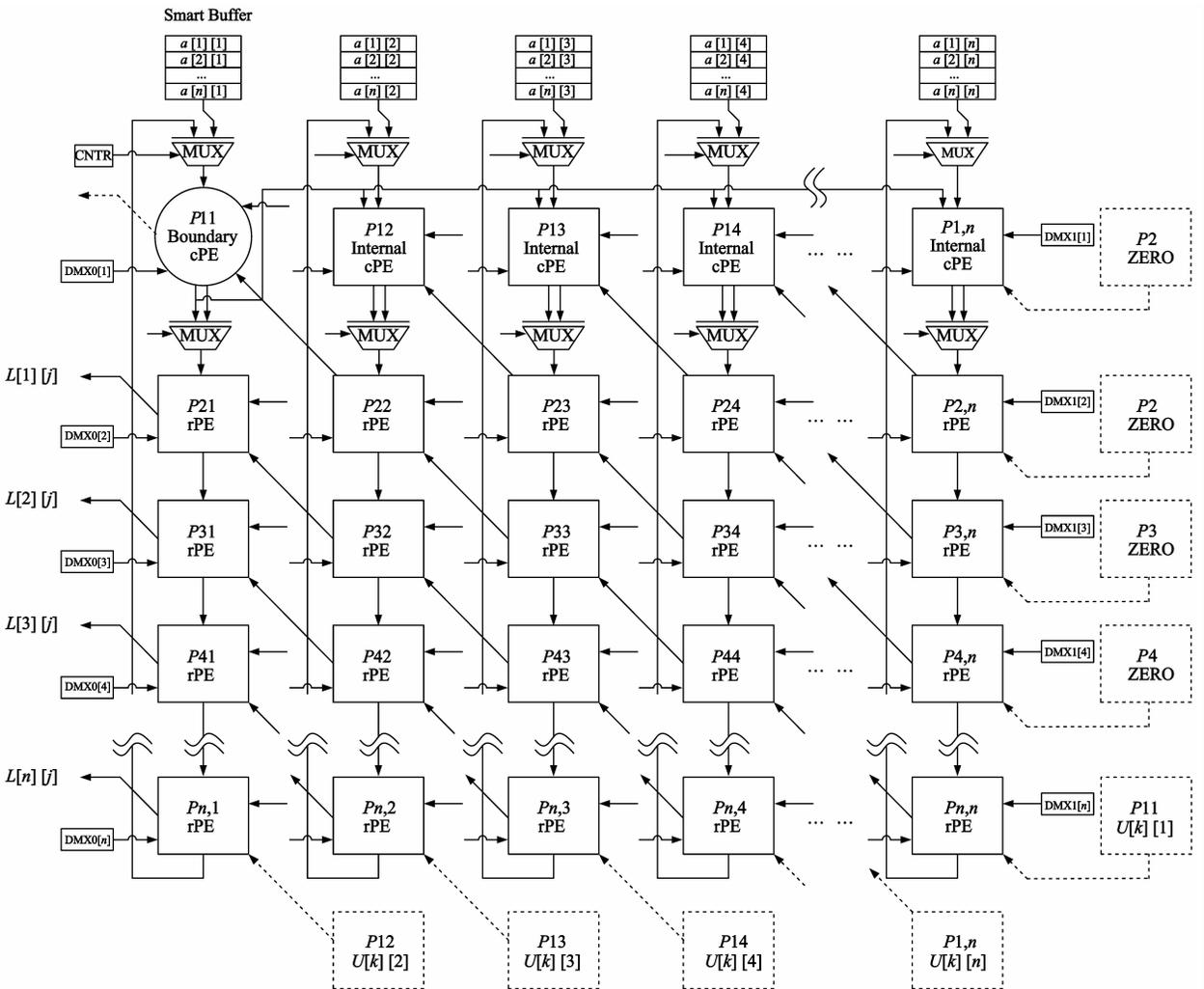


图3 LU分解阵列结构

### 4 实验结果与性能比较

#### 4.1 FPGA 资源占用

本文提出的设计采用 VHDL (Very High speed integrated hardware Description Language) 语言实现. 对设计采

用的实验方案为:使用 Mentor Graphic Modelsim v6.5 仿真验证工具对设计进行仿真后在 Xilinx ISE 11.1 工具中完成综合,并给出资源占用结果;对设计的性能进行估计与分析;最后,将设计加载到 FPGA 上进行实际运行与功能验证.本文使用的 FPGA 为 xc5vlx330t.

文中涉及的浮点计算(加法、减法、乘法、除法以及开平方)均使用开源算术浮点库 libhdfltp 的单精度流水浮点计算部件实现,遵照 IEEE754 标准的浮点数据格式.表 1 给出了 LU 分解并行结构中 PE 单元的综合结果.

表 2 并行阵列结构与相关工作的性能比较

相关工作	本文工作	文献[15]	文献[16]	文献[17]
Time	$n$	$4n + \lceil n/k \rceil - 6$	$6n - 6$	$n^3/P$
PE Number	$P$	$P/2$	$P/2$	$P$
PE Complexity	Simple	Simple	Simple	Complex
Storage Requirement	$n^2$	$n^2/2$	$n^2/2$	$2 \times P$

## 4.2 性能比较

对提出的矩阵三角化计算并行结构和相关工作进行比较,从设计需求角度出发,只与专用矩阵计算或矩阵分解并行结构进行比较.表 2 给出了本文提出的矩阵三角化计算并行结构与文献[15~17]结构在计算性能上的比较.以完成求解线性方程组为目标,对各个相关工作提出的阵列结构的计算复杂度和存储需求进行估计.其中,假设每种结构中 PE 的存储带宽与计算均一致,且都运行在同一个频率上.在这些相关工作中,文献[16]为基于 Systolic Array 提出的二维结构,而文献[15]为其基础上的改进研究,其中  $k$  为广播间隔常数.其存储需求为本文的并行阵列结构的一半,但一次运行只能完成三角化过程或三角回代过程,如求解线性方程组则需要对阵列进行二次装载.另外即使在最好的情况下,运算时间也是本文提出并行阵列结构的 4 倍.而文献[17]的设计为一维线性结构,其计算复杂度为  $O(n^3)$ ,并采用了较为复杂的 PE 结构.与之相比,本文的并行阵列结构具有更高的运行频率和更高的性能.

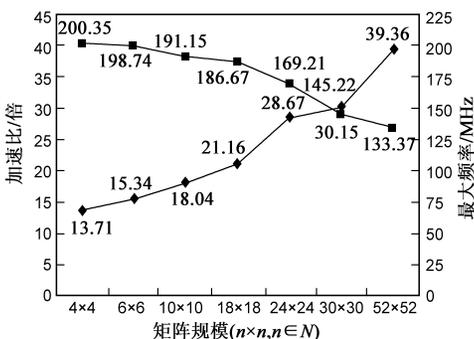


图4 LU分解阵列并行结构vs.通用处理器的加速比和最大运行频率随矩阵规模增加的变化

将设计与通用计算平台的 LU 分解进行对比,所比较的 desktop 计算机配置为 Intel CoreTM2 Duo 2.33GHz 处理器、4GB 主存及 CentOS 5.5 操作系统. C 语言编译器为 gcc 4.3,优化选项设为 O2.图 4 给出 LU 分解阵列并行结构与通用处理器上软件实现相比所获得的加速

比随矩阵规模(阵列规模)增加的变化,其中,虚线代表的是所设计的并行结构随着矩阵规模变化的最大频率变化,实线是随着矩阵规模变化加速比的变化.可以看到,在各个阵列规模上的计算中硬件并行结构均可以取得 10 倍以上的加速比,即使可运行的最大频率在降低,加速比也随着矩阵规模的增加逐步提升.在 XC5VLX330-1FF1760 器件上资源允许的范围,中等规模的 52 阶矩阵可达到 40 倍左右的加速比.

从通用度量标准的角度将 LU 分解并行结构与相关工作进行性能比较.使用每秒 10 亿(=  $10^9$ )次浮点运算能力的 GFLOPS(Giga FLOPS)来衡量并行结构的性能,其中 GFLOPS 的运算如式(20)所示.

$$\text{GFLOPS} = (\text{Total Floating-point Operations}) / (\text{Time(s)} \times 10^9) \quad (20)$$

LU 分解阵列并行结构的计算总时间为:

$$\begin{aligned} \zeta_{\text{LU}} &= \zeta_{\text{load}} + \zeta_{\text{run}} \\ &= \zeta_{\text{cycle}} \times n + \zeta_{\text{cycle}} \times (n-1) \times (\text{divL} + \text{mulL} + \text{subL} + 1 + n) \\ &= \zeta_{\text{cycle}} \times (n-1) \times (\text{divL} + \text{mulL} + \text{subL} + 2 + n) + \zeta_{\text{cycle}} \end{aligned} \quad (21)$$

其中,LU 分解阵列并行结构的总浮点运算数为:

$$\begin{aligned} \text{Total Floating-point Operations} \\ &\approx (n-1) \times (\text{divL} + \text{mulL} + \text{subL} + 2 + n) \\ &\approx n^2 + 20n \end{aligned} \quad (22)$$

表 3 给出了了在一些主流计算平台上的 LU 分解的性能和计算效率,包括 Intel MKL 10.2 中 LAPACK LU 子程序的性能,运行于 Intel Core 2 Quad Q6600 四核 CPU,该 CPU 运行频率为 2.4GHz,拥有 8 MB L2 Cache 和 8 GB DDR2.

因采用的 libhdfltp 浮点库提供的浮点运算部件为全流水结构,相当于对浮点运算的全流水并行,故硬件设计的 GFLOPS 峰值性能为:

$$\begin{aligned} \text{Peak GFLOPS} \\ &= \text{Number of FP operations} \times \text{Hardware Clock Frequency} \\ &\approx (1 \times \text{divL} + (n-1) \times (\text{mulL} + \text{subL})) \\ &\quad \times \text{Hardware Clock Frequency} \end{aligned} \quad (23)$$

表 3 通用处理器相关工作的 GFLOPS

Platform	Peak GFLOPS	Sustained GFLOPS	Efficiency (%)
Power 5, 1.9 GHz	7.60	6.21	81.71
Pentium 4 Xeon, 3.6 GHz	7.20	5.20	72.22
Opteron, 2.8 GHz	5.60	4.17	69.10
Core 2 Quad Q6600, 2.4 GHz	38.40	26.23	68.31
Cell BE, 3.2 GHz	14.63	9.46	64.66

对应的 LU 分解阵列并行结构各个规模实现的 GFLOPS 如表 4 所示. 从表中可以看出, 随着矩阵规模的增加, 设计的 LU 分解并行结构的峰值性能、持续性能以及计算效率均不断提升. 在实际测试中, 65 阶矩阵是 XC5VLX330-1FF1760 器件所能达到的单个最大规模阵列, 此时硬件的峰值可以达到 46.00 GFLOPS, 持续性能也可以达到 37.88, 已经超过表 3 中所有的处理器.

表 4 LU 分解并行结构的 GFLOPS

Matrix Size	Peak GFLOPS	Sustained GFLOPS	Efficiency(%)
4 × 4	6.41	2.40	37.50
6 × 6	8.74	3.70	42.31
10 × 10	13.00	6.50	51.00
18 × 18	21.65	13.11	60.53
24 × 24	25.72	16.95	65.91
30 × 30	27.30	19.11	70.28
52 × 52	42.68	33.79	79.17
65 × 65	46.00	37.88	82.35

## 5 结论

面向计算密集型应用的可重构加速硬件在高性能计算领域中体现出巨大的发展潜力. 矩阵分解尤其是矩阵三角化分解作为典型的基础类应用始终处于可重构加速硬件研究的核心地位. 本文对矩阵三角化分解及实现高性能硬件并行结构的关键问题进行了研究, 提出了面向矩阵三角化特征计算的专用并行结构及其设计方法, 从求解一般系数矩阵的线性方程组问题出发, 对实数矩阵 LU 分解及其高性能硬件并行结构设计的关键问题进行了研究. 本文提出的子矩阵更新同一化算法从分析线性计算特征出发, 通过增加与线性关系对应的数据移动通路方式较好地实现了适于硬件的数据并行与流水并行工作模式, 与已有的相关工作相比, 具有算法简单、计算复杂度低、易于实现的特点.

实验结果表明, 与通用处理器上软件实现相比, 基于所设计算法实现的硬件并行结构均可以取得 10 倍以上的加速比, 中等规模的 52 阶矩阵可达到 40 倍左右的加速比, 达到了预期研究目标. 本文提出的并行结构为二维阵列结构, 受浮点计算部件开销和现有器件资源的制约, 在求解大规模实数矩阵的三角化分解问题时, 需采用 Stresson 算法对大规模矩阵进行分块分治策略, 这部分工作将在未来的工作中展开.

## 参考文献

[1] 余慧, 王健. 一种专用可重配置的 FPGA 嵌入式存储器模块的设计和实现[J]. 电子学报, 2012, 40(2): 215 - 220.

Yu Hui, Wang Jian. The design and implement of a special reconfigurable FPGA embedded BRAM [J]. Acta Electronica Sinica, 2012, 40(2): 215 - 220. (in Chinese)

[2] 王佰玲, 田志宏, 张永铮. 奇异值分解算法优化[J]. 电子学报, 2010, 38(10): 2234 - 2239.

Wang Bai-ling, Tian Zhi-hong, Zhang Yong-zheng. Optimization of singular vector decomposition algorithm[J]. Acta Electronica Sinica, 2010, 38(10): 2234 - 2239. (in Chinese)

[3] 薄华, 马缚龙, 焦李成. 图像纹理的灰度共生矩阵计算问题的分析[J]. 电子学报, 2006, 34(1): 155 - 158.

Bo Hua, Ma Fu-long, Jiao Li-cheng. Research on computation of GLCM of image texture[J]. Acta Electronica Sinica, 2006, 34(1): 155-158. (in Chinese)

[4] 于苏东, 刘雷波, 尹首一, 等. 嵌入式粗颗粒度可重构处理器的软硬件协同设计流程[J]. 电子学报, 2009, 37(5): 1136 - 1140.

Yu Su-dong, Liu Lei-bo, Yin Shou-yi, et al. Hardware-software co-design flow for embedded coarse-grained reconfigurable processor[J]. Acta Electronica Sinica, 2009, 37(5): 1136 - 1140. (in Chinese)

[5] J Jang, S Choi, V K Prasanna. Area and time efficient implementation of matrix multiplication on FPGAs[A]. Proceedings of the First IEEE International Conference on Field Programmable Technology [C]. Piscataway, NJ, United States: IEEE Inc, 2002. 93 - 100.

[6] J Jang, S Choi, V K Prasanna. Energy-efficient matrix multiplication on FPGAs[A]. Proceedings of the 12th International Conference on Field Programmable Logic and Application [C]. Heidelberg, Germany: Springer Verlag, 2002. 534 - 544.

[7] S Choi, V K Prasanna. Time and energy efficient matrix factorization using FPGAs[A]. Proceedings of the 13th International Conference on Field Programmable Logic and Applications [C]. Heidelberg, Germany: Springer Verlag, 2003. 507 - 519.

[8] L Zhuo, V K Prasanna. High-performance and parameterized matrix factorization on FPGAs[A]. Proceedings of the 16th International Conference on Field Programmable Logic and Applications [C]. Heidelberg, Germany: Springer Verlag, 2006. 1 - 6.

[9] L Zhuo, V K Prasanna. Hardware/software co-design on reconfigurable computing systems[A]. Proceedings of the 21st IEEE International Parallel&Distributed Processing Symposium [C]. Piscataway, NJ, United States: IEEE Inc, 2007. 1 - 10.

[10] D Boland, G A Constantinides. An FPGA-based implementation of the MINRES algorithm[A]. Proceedings of the 18th International Conference on Field Programmable Logic and Applications [C]. Heidelberg, Germany: Springer Verlag, 2008. 379 - 384.

[11] A R Lopes, G A Constantinides. A high throughput FPGA-based floating point conjugate gradient implementation[A].

Proceedings of the International Symposium on Applied Reconfigurable Computing [C]. Heidelberg, Germany: Springer Verlag, 2008. 75 – 86.

- [12] A R Lopes, A Shahzad, et al. More flops or more precision accuracy parameterizable linear equation solvers for model predictive control[A]. Proceedings of the 17th IEEE Symposium on Field-Programmable Custom Computing Machines [C]. Piscataway, NJ, United States; IEEE Inc, 2009. 209 – 216.
- [13] Y Dou, S Vassiliadis, et al. 64-bit floating-point FPGA matrix multiplication[A]. Proceedings of the 13th ACM/SIGDA International Symposium on Field Programmable Gate Arrays [C]. NY, USA; ACM, 2005. 86 – 95.

[14] Y Dou, J Zhou, et al. Unified co-processor architecture for matrix decomposition[J]. Journal of Computer Science and Technology, 2010, 25(4): 874 – 885.

- [15] D Kim, S V Rajopadhye. An improved systolic architecture for LU decomposition [A]. Proceedings of Application-Specific Systems, Architectures and Processors [C]. Piscataway, NJ, United States: IEEE Inc, 2006. 231 – 238.
- [16] E Casseau, D Degrugillier. Linear systolic array for LU decomposition[A]. Proceedings of the IEEE International Conference on VLSI Design [C]. Los Alamitos, CA, United States; IEEE Inc, 1994. 353-358.
- [17] 鄔贵明. FPGA 矩阵计算并行算法与结构[D]. 长沙: 国防科学技术大学, 2011.

## 作者简介



刘书勇 男, 1978 年出生于河南省长垣县, 博士研究生, 主要研究方向为嵌入式系统、可重构编译系统.

E-mail: liushuyong@hrbeu.edu.cn



吴艳霞(通信作者) 女, 1979 年 9 月出生  
于哈尔滨市, 副教授, 硕士生导师, 主要研究方  
向为可重构编译系统、可信计算.

E-mail: wuyanxia@hrbeu.edu.cn