·开发研究与工程应用· 文章编号：1000—3428(2013)04—0287—04 文献标识码：A 中图分类号：TP391

基于 FPGA 线性方程组的存储优化设计

彭 宇，仲雪洁，王少军

(哈尔滨工业大学电气工程及自动化学院，哈尔滨 150080)

摘 要：将基于现场可编程门阵列(FPGA)的改进 Cholesky 分解应用于大规模线性方程组求解时，会出现存储资源限制和带宽瓶颈问题。为此，提出一种基于层次化存储策略和多端口分块式访问方式的解决方案。结合片内双极随机存取存储器(BRAM)与片外同步动态随机存取存储器(SDRAM)，构成分层存储结构，通过片内存储复用降低存储资源需求。采用多端口分块式方式访问片外 SDRAM，提高带宽并规避随机数据存取的访问延迟。测试结果表明，相对于 Xeon CPU，该方案能够实现 17 倍~215 倍的效率提升。

关键词：现场可编程门阵列；线性方程组；矩阵；改进 Cholesky 分解；带宽

Design of Storage Optimization Based on FPGA Linear Equations System

PENG Yu, ZHONG Xue-jie, WANG Shao-jun

(School of Electrical Engineering and Automation, Harbin Institute of Technology, Harbin 150080, China)

【Abstract】When using Cholesky decomposition to solve large-scale linear equations system based on Field Programmable Gate Array(FPGA), the storage size limits and access bandwidth becomes its bottleneck. This paper proposes a solution based on hierarchical storage strategy and multi-ports block access method. The hierarchical storage structure is constituted with internal Bipolar Random Access Memory(BRAM) and external Synchronous Dynamic Random Access Memory(SDRAM). The internal BRAMs are mostly reused to further decrease the storage limits. Accessing external storage through multi-port and block data access methods enhances the effective bandwidth utilization and avoids the random access delay. Experimental results show that, the design can realize 17 to 215 times efficiency speedup compared with Xeon CPU.

【Key words】Field Programmable Gate Array(FPGA); linear equations system; matrix; modified Cholesky decomposition; bandwidth

DOI: 10.3969/j.issn.1000-3428.2013.04.066

1 概述

线性方程组求解作为一类典型的计算密集型任务，广泛存在于数据挖掘、信号处理、数值逼近等科学计算领域。利用现场可编程门阵列(Field Programmable Gate Array, FPGA)的高并行和深流水线操作，可在有限的硬件资源下实现较高的计算性能[1]。因而 FPGA 成为进行线性方程组求解计算的一种有效手段，以 FPGA 实现线性方程组求解为目的的矩阵分解方法已获广泛研究[2]。

存储资源限制是 FPGA 实现大规模线性方程组求解问题的瓶颈。目前针对此问题的研究比较有限，但作为 FPGA 实现计算密集型任务的共性问题，研究者已在科学计算领域针对存储资源限制问题开展了相关研究工作[3]。较为通常的解决方案是采用片外存储单元分担存储压力[4]，但片外存储器的访问带宽限制了计算效率。文献[5-6]提出通过细化计算流程减少对片外存储器访问次数的优化方案，文献[7]则改进计算结构，提高带宽利用率。上述研究均是根据具体算法特点，选择合理的存储结构和存储器访问方式对设计进行存储优化，虽对本文的优化方法有一定借鉴意义，但针对线性方程组求解问题，还应结合相应算法特点和处理数据规模进行分析，以进一步提升系统性能。

本文基于矩阵改进 Cholesky 分解原理，针对 FPGA 实现过程中存储资源和带宽限制问题，利用层次化存储策略，在保证计算效率的同时有效利用存储资源，并通过对片外

基金项目：教育部新世纪优秀人才支持计划基金资助项目(NCET-10-0062)；教育部高等学校博士学科点专项科研基金资助项目(20092302110013)

作者简介：彭 宇(1973－)，男，教授、博士生导师，主研方向：可重构计算；仲雪洁，硕士研究生；王少军，博士研究生

收稿日期：2012-05-09 **修回日期：**2012-06-24 **E-mail：**pony911@163.com

存储器的访问带宽优化措施，解决带宽瓶颈问题。

2 线性方程组求解原理

线性方程组求解问题可以概括为 $\boldsymbol{Bx}=\boldsymbol{y}$。其中，$\boldsymbol{B}\in\boldsymbol{R}^{n\times n}$；$\boldsymbol{x}\in\boldsymbol{R}^{n\times 1}$；$\boldsymbol{y}\in\boldsymbol{R}^{n\times 1}$。改进 Cholesky 分解适用的分解矩阵为对称正定阵，故对于一般性的 n 阶实矩阵 $\boldsymbol{B}$，若要利用改进 Cholesky 分解获得线性方程组的解，则需通过正交化得到等效方程组 $\boldsymbol{B}^{\mathrm{T}}\boldsymbol{Bx}=\boldsymbol{B}^{\mathrm{T}}\boldsymbol{y}$，即 $\boldsymbol{Ax}=\boldsymbol{B}^{\mathrm{T}}\boldsymbol{Bx}=\boldsymbol{B}^{\mathrm{T}}\boldsymbol{y}=\boldsymbol{b}$。

对称正定阵 $\boldsymbol{A}$ 可以分解为 $\boldsymbol{A}=\boldsymbol{LDL}^{\mathrm{T}}$。其中，$\boldsymbol{D}$ 为对角阵；$\boldsymbol{L}$ 是对角线元素为 1 的下三角矩阵。$\boldsymbol{L}$ 和 $\boldsymbol{D}$ 的元素按下式求解：

$$\begin{cases} d_r=\left(a_{rr}-\sum_{k=1}^{r-1}l_{rk}^2d_k\right) \\ l_{ir}=\left(a_{ir}-\sum_{k=1}^{r-1}l_{ik}d_kl_{rk}\right)/d_r \end{cases}$$

$\boldsymbol{LDL}^{\mathrm{T}}\boldsymbol{x}=\boldsymbol{b}$ 可以通过 $\boldsymbol{Lz}=\boldsymbol{y}$、$\boldsymbol{Dr}=\boldsymbol{z}$、$\boldsymbol{L}^{\mathrm{T}}\boldsymbol{x}=\boldsymbol{r}$ 这 3 个方程依次求解。其中，$\boldsymbol{z}$ 和 $\boldsymbol{r}$ 为引入的 n 维列向量。

故线性方程组求解可分为 2 个步骤：第 1 步为改进 Cholesky 分解；第 2 步为三角线性方程求解。

3 层次化存储设计

存储层次的合理划分是解决存储资源限制的首要条件，需根据计算过程中涉及的不同数据的规模，兼顾带宽和效率 2 个因素进行设计。另外，由于片内存储资源有限但访问速度较快，因此需要对其使用策略进行重点关注。

3.1 运算数据规模分析

在改进 Cholesky 分解中，运算数据可以分为待分解矩阵 $\boldsymbol{A}$(元素为 a)、下三角阵 $\boldsymbol{L}$(元素为 l)和对角阵 $\boldsymbol{D}$(元素为 d)3 种。此外，根据式(1)可知，中间变量 d_kl_{rk} 为求解 d_r 和 l_{ir} 的共同部分。为了减少 3 次乘法次数，降低计算延迟，添加 d_kl_{rk} 作为数据暂存，故运算数据可归纳为 a、l、d 和 dl。三角线性方程求解中运算数据可以简单归纳为已知向量 $\boldsymbol{y}$ 及待求解向量 $\boldsymbol{x}$，$\boldsymbol{z}$ 和 $\boldsymbol{r}$ 采用资源复用的方法，不单独占用片内存储资源。故整体运算数据有 a、l、d、dl、$\boldsymbol{x}$ 和 $\boldsymbol{y}$，规模如表 1 所示，其中，n 表示维数。

表 1 运算数据规模

名称	规模
a	$n\times(n+1)/2$
l	$n\times(n+1)$
dl	$n\times(n+1)$
x	n
b	n
d	n

如表 1 可知，线性方程组求解空间复杂度为 $O(n^2)$。若计算规模达到 10 000 维，以单精度浮点制计算为例，需要的存储资源将达到 1.6 GB。而目前主流 FPGA 内部双极随机存取存储器(Bipolar Random Access Memory, BRAM)资源仅在 Mb 级别，难以满足数据存储需求，故采取 FPGA 片上 BRAM 与片外 DDR2 同步动态随机存取存储器(Synchronous Dynamic Random Access Memory, SDRAM)分层存储策略。

3.2 整体存储层次划分

根据上述分析，设计如图 1 所示的线性方程组求解系统结构。

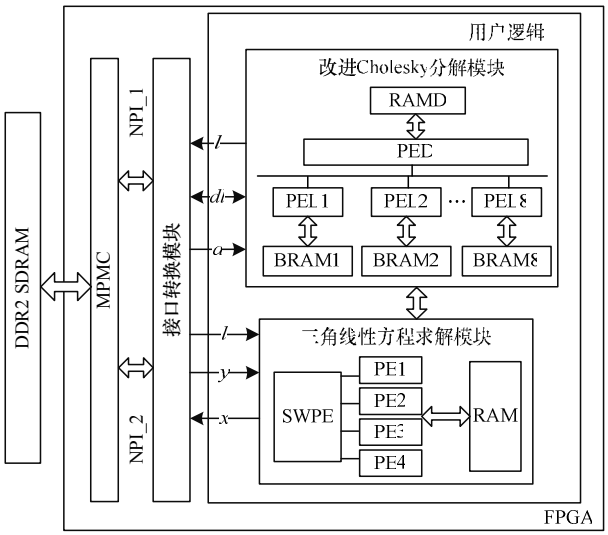

图 1 线性方程组求解系统结构

如图 1 可知，系统整体结构可分为 FPGA 和 DDR2 SDRAM 2 个部分。根据数据规模分析，将 a、l_{ir}、d_kl_{rk}、存入外部 DDR2 SDRAM 中，为兼顾计算效率，其余运算数据在片上 BRAM 存储。对外数据交互通过自定义的接口转换模块与控制访问片外存储器专用 IP 核多端口存储控制器(Multi-port Memory Controller, MPMC)互连，并通过 MPMC 所能提供访问接口中传输速度最高的本地端口总线(Native Port Interface, NPI)作为数据传输通道。

FPGA 内部的用户逻辑是实现线性方程组求解功能的核心部分。用户逻辑分为改进 Cholesky 分解和三角线性方程求解模块，采用流水线并行设计思想，利用多个并行计算单元(Process Element, PE)完成整个运算任务。其中，PEL(PE for L)为下三角阵 $\boldsymbol{L}$ 的计算单元；PED(PE for D)为对角阵 $\boldsymbol{D}$ 计算单元；PE1~PE4 为三角方程求解 PE；SWPE 为数据选择器。在此仅给出关键互连关系，设计细节参见课题组相关工作[7]，受篇幅原因不再赘述。

3.3 内部存储单元复用

在 FPGA+片外存储器的整体框架下，为了进一步增加 FPGA 内部存储资源利用率，设计重用内部 BRAM 资源。本文设计中 90%的片内存储资源采用复用原则，RAM 复用次数均大于 1 次，且不仅局限于单一功能模块资源复用，也实现了不同功能模块间 BRAM 资源的交叉使用。例如图 1 中三角方程求解模块 RAM，其为待求向量 $\boldsymbol{y}$，中间求解向量 $\boldsymbol{z}$、$\boldsymbol{r}$，最终结果 $\boldsymbol{x}$ 的复用存储单元。而改进 Cholesky 分解模块中的 BRAM1~BRAM8，也在三角方程求解模块中通过 SWPE 切换复用。这样既减少对于外存的依赖，降低

存储带宽压力，又提高了系统的计算效率。

4　片外存储器访问带宽优化设计

采用层次化存储策略后，片外存储器访问带宽成为影响片内并行处理单元运算效率的关键因素。针对此问题，通过访问方式优化，结合计算流程改进提高带宽利用率。

4.1　多端口分块式数据访问设计

片外存储器的访问带宽与 NPI 端口数量和访问方式直接相关。本文采用多 NPI 端口及分块式数据访问方式，以提高访问带宽。

4.1.1　多端口数据访问

当多路 PE 并行计算时，单路 NPI 存储带宽远远不能满足数据传输需求，因此，应增加 NPI 端口数目，采取多端口数据访问方式。

由图 1 可知，用户逻辑对外数据交互类型有 a、l、d、dl、$\boldsymbol{x}$ 和 $\boldsymbol{y}$ 6 种，且彼此之间存在时序重叠。在改进 Cholesky 分解模块，对外交互数据有 a、l 和 dl 3 类，其中，a 为输入数据；l 为输出数据；dl 为双向数据。且 a 和 dl 输入数据存在时序重叠，考虑效率问题及带宽利用率最大化原则，设计双通道 NPI 接口，分别连接 3 类数据，达到效率最优化。而在三角线性方程求解模块，其数据交互类型较少，考虑传输总线分时复用原则，可利用改进 Cholesky 分解模块中已有的双通道 NPI 接口传输，有效利用硬件资源，提高总线利用率。

针对 a 和 dl 数据时序重叠导致 NPI 接口冲突，采用 MPMC 仲裁机制进行优先级判断[8]，利用 Custom 模式设置数据交互更为频繁的 dl 为更高优先级，有效解决冲突问题。

4.1.2　分块式数据访问

根据测试，NPI 接口工作于 256 Byte 块传输模式时速率最高可达 742.6 MB/s，而随机传输模式速率仅为 46.1 MB/s。显然不同传输模块速度差异较大，应尽量采取分块式存储访问方式，有效提高带宽利用率。为了能够实现块传输优势、保证较高的通信带宽，在 NPI 接口和用户逻辑之间设计接口转换模块。

接口转换模块与用户逻辑间采用自定义总线接口，可支持对 NPI 接口的 256 Byte、128 Byte、32 Byte、4 Byte 的块传输模式，且优先级依次降低。由 NPI 协议可知，NPI 最小支持的传输块大小为 4 Byte，不同字节数块传输模式下的起始地址要求为 4 Byte 数的整数倍，故面对不同数据规模的传输任务时，接口转换模块通过对起始访问地址的解析，选择最优大小的块传输组合，实现最高传输速度。

以起始地址为 b 的一定量数据传输为例，依照该设计的程序流程如图 2 所示。从图中可以看出，接口转换模块首先将地址 b 依次与不同字节传输模式下所要求的地址进行匹配，选择目前最快传输模式，再通过更新地址并重新进行地址匹配和传输模式选择，直至完成所有数据传输任务。在起始地址不理想的情况下，如传输数据数目足够大且地址连续，最终可实现单次 256 Byte 的高速传输。

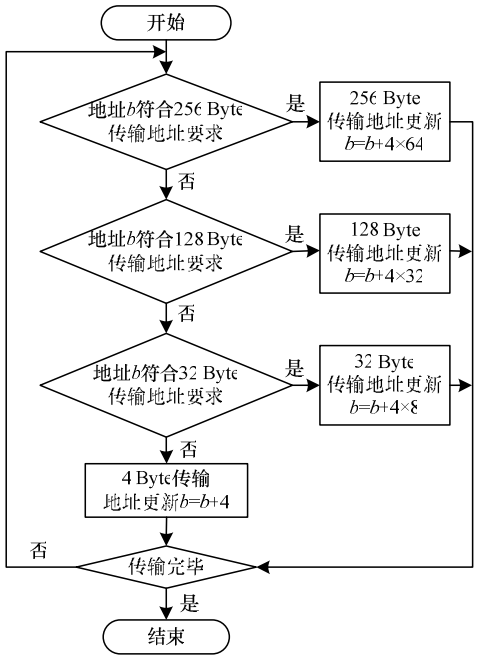

图 2　接口转换模块数据传输流程

另外，接口转换模块还将接口信号访问方式变为请求/应答模式，并通过控制缓存读写使能，解决了用户逻辑跨时钟域问题，扩展了用户逻辑的适用范围，简化和优化了设计。

4.2　计算流程优化设计

结合算法特点，合理设计运算流程以适应带宽需求也是解决带宽瓶颈问题的有效手段。结合前文 NPI 接口的多端口分块式传输策略，对算法流程和片内缓存的设计进行了优化。

4.2.1　基于块传输的算法流程优化

由于块传输模式对于存储地址有一定要求，在算法实现过程中，要尽量避免频繁更换地址导致的传输延迟。以矩阵 $\boldsymbol{A}$ 第 m 行分解时 $d_k l_{rk}$ 数据传输过程为例进行说明。实现流程如图 3 左侧所示。

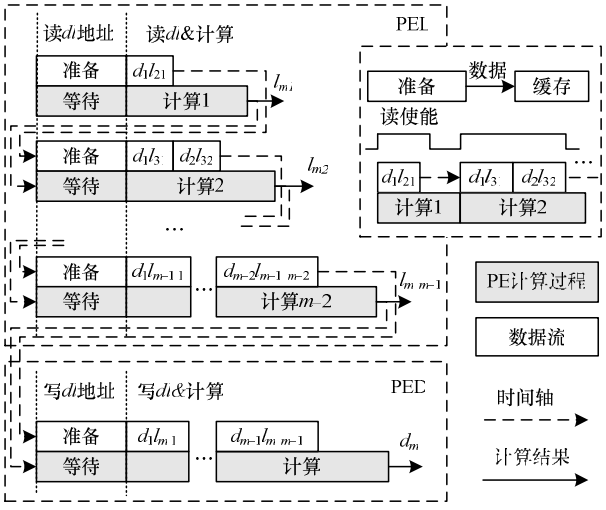

图 3　矩阵分解第 m 行计算流程

主要过程由上而下顺次进行，PEL 工作过程中进行 $d_k l_{rk}$ 读操作，PED 工作过程中实现 $d_k l_{rk}$ 写操作。准备阶段主要为 $d_k l_{rk}$ 读写请求/响应、地址更新、数据等待等过程。通过存储地址优化，将 $d_k l_{rk}$ 在运算过程读写操作均设计为连续的地址控制，在第 m 轮开始计算时即发出数据读入请求，给出首地址更新，通过上文提出的访存策略进行大块数据访问，本轮不再更新地址，直至停止信号。读入的数据暂时缓存，通过控制读使能信号保证数据和 PE 计算同步，如图 3 右侧所示。这样避免了由于地址不断更新造成的计算延迟，并最终通过高效的存储访问模式，提高了计算效率。

4.2.2 乒乓存储

在采用多端口分块式存储访问策略提升带宽速度的同时，设计应尽量避免数据传输过程中时间损耗，实现数据流的连续传输，提高带宽利用率。因此，本文采用交互式存储/计算方式，即乒乓存储。该方法摒弃了传统的“存储-计算-存储”模式，而是将内部存储单元分组，使计算和数据存储同时进行，既避免了因计算延迟导致的数据传输等待，又保证了计算效率。以求解模块中 $\boldsymbol{L}^{\mathrm{T}}\boldsymbol{x}=\boldsymbol{r}$ 为例，需同时进行外部存储器中 $\boldsymbol{L}$ 的读入和计算。故将 8 个 PEL 中的 BRAM 分为 2 组，箭头方向表示数据流向。通过数据选择器 SW 的切换，实现 2 组 BRAM 分时从外存中依序读入 $\boldsymbol{L}$；通过 SWPE，实现 2 组 BRAM 与运算单元的切换，实现存储和计算的同时进行。实现框图如图 4 所示。

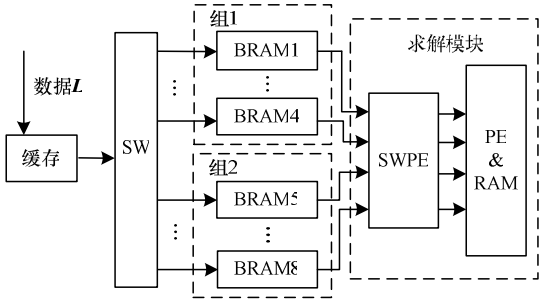

图 4 乒乓存储实现框图

5 实验分析

本文设计选用 Vertex XC5VFX130T FPGA，支持最高时钟为 150 MHz。为评价系统性能，测试系统不同运算规模下端口有效带宽，即交互数据规模与传输时间之比。相对于 NPI 数据传输最大速度 742.6 MB/s，得出相应带宽利用率，如表 2 所示。

表 2 端口有效传输带宽

维数	有效带宽/(MB·s^{-1})	带宽利用率/(%)
256	664.0	89
512	714.9	96
1 024	731.1	98
2 048	737.3	99
4 096	739.9	99
8 192	741.1	99

由表 2 可知，本文系统有效带宽始终保持在较高水平，且随计算规模的增加显著提高，最终可接近所能实现的 NPI 极限带宽速度，带宽利用率较高。

为评价计算效率，采用 8 组不同维数线性方程组求解进行实验，分别与 PC 平台和文献[9]实验进行对比。其中，PC 平台的 CPU 为 Xeon X5570@2.93 GHz，具有 64 GB DDR2 SDRAM 存储器。由于现有线性方程组求解 FPGA 实现相关文献无等价对比数据，文献[9]实验虽未给出全部求解的结果数据，但其已具备占整体计算量绝大部分的矩阵分解实验数据。另外，文献[9]实验中 FPGA 型号为 XC5VLX50TFF113，与本文的目标 FPGA 为同一系列；并且其采用与本文相同的 8PE 单元并行模式、180 MHz 时钟亦与本文基本相同，故文献[9]实验同本文线性方程组求解实现具有可比性。表 3 为不同方法实现时间及对比结果。

表 3 实现时间及对比结果

维数	本文设计时间/s	CPU/s	文献[9]实验时间/s	与 CPU 的加速比	与文献[9]实验的加速比
256	0.005	0.094	-	17.29	-
512	0.031	0.922	0.032	30.13	1.03
1 024	0.195	7.891	0.276	40.38	1.41
2 048	1.371	70.766	2.719	51.62	1.98
4 096	10.205	1 028.300	23.226	100.76	2.28
8 192	78.611	16 916.230	-	215.19	-

可以看出，随着维数的增大，本文设计相对 PC 平台和文献的运算优势愈加明显，若只针对改进 Cholesky 分解模块，本文方法较文献会达到更高的加速比。目前，本文所完成研究工作支持的数据最大计算维数可达 10 000，若进一步扩展 DDR2 SDRAM 和 BRAM 存储资源，则可实现更大规模线性方程组的求解，为大规模线性方程组求解的 FPGA 实现提供更加切实可行的解决方案[10]。

6 结束语

针对 FPGA 实现大规模线性方程组中存在的存储受限和带宽瓶颈问题，本文提出了层次存储策略和分块式存储访问等相应解决方法。通过对运算数据规模进行分析，确定了 FPGA 和外部存储器的实现架构；采用访问方式优化提高带宽，同时优化计算流程减少带宽需求，从而进一步改善系统性能。通过上述优化方法，扩展了数据处理规模，实现了带宽利用率的提升，从而为大规模数据计算进行了方法准备。与 PC 平台及相关文献实验的对比证明了本文设计的有效性。

参考文献

[1] 宋庆增, 顾军华. 稀疏矩阵向量乘的 FPGA 设计与实现[J]. 计算机工程, 2011, 37(23): 214-216.

[2] 邬贵明. FPGA 矩阵计算并行算法与结构[D]. 长沙: 国防科学技术大学, 2011.

[3] Liu Jie, Zakharov Y V, Weaver B. Architecture and FPGA Design of Dichotomous Coordinate Descent Algorithms[J]. IEEE Trans. on Circuits and Systems, 2009, 56(11): 2425-2438.

(下转第 295 页)