

文章编号: 2095-4980(2017)06-1066-05

## 一种多线程 FPGA 时序驱动布线算法

于梦薇<sup>1,2</sup>, 刘洋<sup>1</sup>, 尹韬<sup>1</sup>, 杨海钢<sup>\*1,2</sup>

(1.中国科学院 电子学研究所, 北京 100190; 2.中国科学院大学, 北京 100049)

**摘要:** 随着现场可编程门阵列(FPGA)器件尺寸不断增大, 计算机辅助设计(CAD)工具运行时间成为突出的问题。布线是 FPGA 的 CAD 流程中最为耗时的一个阶段, 一种能有效缩短布线时间的方法就是并行布线。本文提出一种减少 FPGA 时序驱动布线算法运行时间的多线程方法。该算法首先将信号按照线网的扇出数量进行排序, 再将排序后的线网均匀分配到各个线程中, 最后并发执行所有的线程。在布线质量没有受到显著影响的前提下, 即线长增加 2.58%, 关键路径延时增加 1.78% 的情况下, 相对于传统通用布局布线工具(VPR)时序驱动布线算法 8 线程下的加速比为 2.46。

**关键词:** 现场可编程门阵列(FPGA); 计算机辅助设计(CAD); 并行布线; 时序驱动布线; 多线程

中图分类号: TN47

文献标志码: A

doi: 10.11805/TKYDA201706.1066

## A multithread FPGA timing-driven routing algorithm

YU Mengwei<sup>1,2</sup>, LIU Yang<sup>1</sup>, YIN Tao<sup>1</sup>, YANG Haigang<sup>\*1,2</sup>

(1.Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China;

2. University of the Chinese Academy of Sciences, Beijing 100049, China)

**Abstract:** As the size of Field-Programmable Gate Array(FPGA) has surged rapidly, the deteriorative runtime of Computer-Aided Design(CAD) tools has become a major concern. Routing is one of the most time-consuming stage in CAD process, while introducing parallel routing algorithm could be an effective way to cut the runtime. A multithread timing-driven routing technique based on this idea is proposed in this paper. In the parallelization approach, nets are sorted according to their fan-out amount sequence firstly, then the sorted nets are averagely assigned into threads and all the threads are executed concurrently. Using this method, the total wire length and critical path delay are increased by 2.58% and 1.78% respectively, which means the routing quality is well preserved. However, in terms of runtime, it wins approximately 2.46 × speedup versus the widely used Versatile Place and Route(VPR) in the case of 8 threads.

**Keywords:** Field-Programmable Gate Array(FPGA); Computer-Aided Design(CAD); parallel routing; timing-driven routing; multithread

FPGA 配套的电子设计自动化(Electronic Design Automation, EDA)工具的运行时间会随着包含的逻辑器件规模的增大而日益增长, 从而成为制约 FPGA 发展的瓶颈。FPGA 设计通常包括以下步骤: 逻辑综合, 工艺映射, 布局布线, 仿真验证和下载等<sup>[1]</sup>。其中, 布线耗时在整体运行时间占到非常大的比例, 这是由于布线算法流程复杂, 包括大量线网的路径搜索和循环迭代。由于 EDA 工具运行速度问题越来越引发关注, 因此并行化问题成为研究焦点。并行计算能够有效提高 EDA 软件中布线算法的性能。2000 年, 文献[2]首次对 FPGA 布线算法进行了并行化加速。在该研究中, 所有信号线被分配到不同的处理器上进行布线, 在 3 个处理核下实现了 2.5 倍加速比。2010 年, 文献[3]给出了一种区分粒度的方法对布线算法进行并行化, 分别从粗粒度角度和从细粒度角度处理信号。这种算法能在 4 核环境中加速 2.3 倍, 在处理器机群下能加速 2.85 倍。文献[4]给出一种基于区域划分的多线程布线方法, 将信号分为高扇出线网和低扇出线网, 将高扇出线网划分成几个子线网并行布线, 将整个布线区域按照线程数划分, 对低扇出线网则根据边界盒按照区域划分。该算法在 4 个线程时可将布线速

收稿日期: 2016-12-17; 修回日期: 2017-01-04

\*通信作者: 杨海钢 emai:lyanghg@mail.ie.ac.cn

度提高到 1.9 倍。文献[5]使用开源工具 Galois，在一个共享内存的多处理器平台上实现了布线算法的动态并行。在 8 线程下，该算法实现了 5.46 倍加速比。文献[6]将整体布线区域分成 3 组，第 1 组线网的线网间可能会有冲突，而另外 2 组线网保证没有冲突，可以执行并行布线。在整个过程中递归地将其中某一组线网再进行划分，直到所有线网布线完毕。该算法在 32 个处理器上最高可实现 7.06 倍加速。

然而，以上所有文献都只是针对布通率驱动布线算法进行并行化，而没有考虑时序驱动布线算法。时序驱动布线算法的处理流程比布通率驱动算法更为复杂，所以上述布通率驱动的并行布线算法不能直接应用到时序驱动布线算法的并行优化中。由于时序驱动布线器能实现更快的电路速度，降低电路延时的特性，并且在布线中耗时远远小于布通率驱动布线器，随着时钟频率的增加，时序驱动布线算法在工业界运用更为广泛。本文着力于解决时序驱动布线算法的并行化问题。

## 1 原理

### 1.1 时序驱动布线算法

来自多伦多大学的研究团队开发了 FPGA 配套计算机辅助设计工具软件——通用布局布线工具(Versatile Place and Route, VPR)，其中的布线算法能够代表学术界研究的最高水平<sup>[7]</sup>。在学术界，大多数布线算法的研究都基于 VPR 软件中的布线算法。VPR 中的布线器分为两大类，布通率驱动布线器和时序驱动布线器。其中时序驱动布线算法考虑到电路的延时情况，是当前非常先进的布线算法。本文算法也是对 VPR 中的时序驱动布线算法进行并行化。布通率驱动布线算法基于 Pathfinder 路径搜索算法，分 2 步完成布线。第 1 步，允许多个线网使用同一布线资源，在对每条线网搜索时都试图找到最短路径，在这个过程中对资源的占有进行标记，称为拥挤度。第 2 步，使用拆线重布的办法进行迭代重布线，每轮迭代的拥塞情况直接影响着下一轮迭代的成本，以实现减少拥塞的目标<sup>[8]</sup>。结点  $n$  连接到结点  $m$  的成本函数是

$$Cost(n) = b(n)h(n)p(n) + BendCost(n, m) \quad (1)$$

式中： $b(n)$ 表示基本成本； $h(n)$ 表示历史拥挤度； $p(n)$ 表示当前拥挤度； $BendCost(n, m)$ 表示对转弯连线的惩罚因子。经过一轮迭代后，结果逐渐收敛，直到出现没有拥塞的情况则视为电路布通，在布通率驱动布线器中停止迭代。

时序驱动布线算法在实现电路布通的基础上，对电路的关键路径延时进行优化。VPR 中的时序驱动算法在消除拥塞现象与减小关键路径延时这两方面进行折衷考虑<sup>[9]</sup>。时序驱动布线算法引入了关键度来平衡拥塞与电路速度，关键度  $Crit(i, j)$ 的定义：

$$Crit(i, j) = \max\left(\left[MaxCrit - \frac{slack(i, j)}{D_{max}}\right]^{\eta}, 0\right) \quad (2)$$

式中： $D_{max}$ 是关键路径延时； $slack(i, j)$ 是源点到漏端之间的延时裕量； $MaxCrit$ 和 $\eta$ 是控制拥塞和延时折中关系的参数。那么，布线的成本函数也相应地更新为：

$$Cost(n) = Crit(i, j)delay_{Elmore}(n, topology) + [1 - Crit(i, j)]b(n)h(n)p(n) \quad (3)$$

在 FPGA 时序驱动布线算法中，时序的约束是由从输入到输出的延迟进行限制的，所以每条从输入到输出的路径都有指定的延迟范围。对于一条路径，延时裕量是要求的时间和沿着路径传播的实际时间之差。每轮迭代后通过时序分析模块提取线网延时，更新延时裕量，从而影响到关键度和成本函数，指导下一轮布线。

### 1.2 布线树结构

与布通率驱动布线器相比，时序驱动布线器每轮迭代中除了记录布线路径的轨迹结构，还要增加一项布线树结构，这是由于传统轨迹存储结构不包含时序驱动布线所需的全部信息。布线树结构用来储存时序布线中实时的布线结果，用于时序分析阶段的增量分析，提高效率。

布线树结构是一个临时的结构，通过链表实现。在每轮迭代中，每个线网开始布线时都会创建一棵新树，并首先用线网的源点初始化这棵树，即以源点为根节点。在布线过程中，每到达一个漏端就更新布线树，将最新产生的布线资源线段加入到布线树中。这里将利用已经产生的布线结果继续拓展新的漏端，直到所有的漏端都被连接。时序分析过程不是实时的，不在每次布线树有增量时就进行分析，而是当这个线网在这轮迭代的布线任务全部结束后，将会利用布线树更新当前线网的延时。此时只有布线轨迹还需要保留，在更新成本和保存布线结果时将会调用，而布线树结构将会被删除，在下一轮布线时重复一遍布线树建立、初始化的过程，与上一轮中的布线树无关。

所以，布线树结构相对于线网具有独立性，各个线网之间在这方面不存在数据交流，由于各个线网是分配在各个线程上的，线网具有完整性，所以布线树结构非常适合同算法一起进行并行化。各个布线树在线程间完全没有交流，并在线程停止前全部被删除。

### 1.3 POSIX 线程库

POSIX(Portable Operating System Interface of UNIX) 线程，表示可移植操作系统接口。POSIX 线程(POSIX threads)，简称 Pthreads，是线程的 POSIX 标准，一个经典的线程模型<sup>[10]</sup>。程序中主要使用了 POSIX 线程库中的 2 个功能来实现控制：路障和信号量。

如图 1 所示，路障可以保证所有线程在程序中处于同一个位置，从而同步线程。只有所有线程都到达路障后，各线程才可以继续运行下去，否则都会阻塞在路障处。

信号量是一种无符号整型变量，主线程将信号量初始化为 0，其他线程都可以修改信号量。如果在一个线程的某个地方设置当信号量为 0 则阻塞，那么别的线程可以通过修改信号量控制该线程。

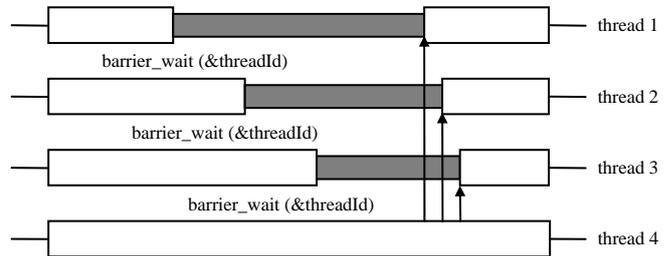


Fig.1 Schematic diagram of barrier working  
图 1 路障工作示意图

## 2 算法设计

### 2.1 算法思想

并行任务划分主要有 2 种方式。第 1 种是基于所在位置将线网进行切割，然后划分到各个并行处理单元中；第 2 种是将线网分成组分配给处理单元，保持线网的完整性<sup>[11-12]</sup>。本文采取第 2 种方式，主要基于两方面考虑：一是由于线网间在布线过程中相对独立，只有拥塞信息是共用的，在数据结构上相对来说并行实现更为容易；二是因为时序驱动布线算法的特殊性，为了方便布线树结构的建立，需要在一轮迭代中，每个线网保持完整，这样会大大降低并行处理的复杂程度。

如果在布线的过程中实时更新成本函数，那么各线程之间的数据交流太过频繁，会造成很大的通信开销。如果一轮迭代之后再更新，那么又会因为更新不及时而造成每轮迭代收敛变慢。所以本文提出了一种折中的方案。在布线开始后，先取出其中一部分线网分配到各线程进行布线，在这部分线网布线完毕后用主线程更新成本函数，这个过程是串行的。资源的占用情况由主线程控制，在布线过程中各子线程只能读而不能修改，在一部分线网布线完毕后各子线程开始向主线程更新这些信息。这也是运用了线程本身共享变量的特性。然后重复这个过程，直到所有线网布线完毕。

### 2.2 算法流程

如图 2 所示，算法的具体实现方法是，在数据结构创建并初始化后，建立  $N$  个线程，用路障对各线程进行同步。分配一部分线网给各线程，这部分线网的数量是  $N$  的倍数。分配时按照第 1 个线程分配第  $1, N+1, 2N+1, \dots$  个线网，第 2 个线程分配第  $2, N+2, 2N+2, \dots$  个线网的顺序，以保证各线程负载基本平衡。当每个线程都完成了当前分配的任务，通过信号量控制让每个线程阻塞，同时主线程更新成本信息。在更新完毕后再通过信号量激活各个线程开始工作，即再取出一部分线网重复这个过程，直到所有线网布线完毕，将各个子线程合并到主线程中。

此时，这一轮迭代已经完成，进入到串行程序中，对其进行全局成本函数更新和时序分析等串行步骤，然后再进入下一轮迭代，直到布线成功。

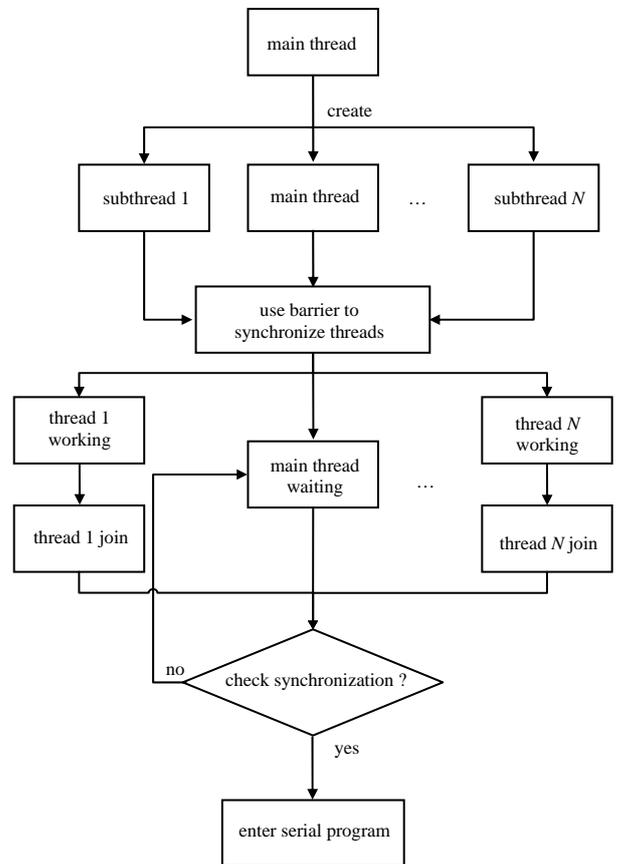


Fig.2 Flow chart of threads working  
图 2 线程工作流程图

### 3 实验结果

#### 3.1 实验方法

本文中的实验在 2.27 GHz,4 GB 内存的实验平台下运行。选取 MCNC 电路测试集中的 10 个电路，使用 Altera 公司 Stratix IV 系列的 siv51004 结构。先用 VPR 软件中的装箱、布局功能对测试电路进行装箱、布局，再使用 VPR 软件中的时序驱动布线器确定布通该电路的最小通道宽度。然后将通道数固定为最小通道宽度，对同一个布局结果使用本文中的布线器和 VPR 软件中的时序驱动布线器分别进行布线，将布线结果进行比较。每次运行开启  $N$  个线程，对  $N=2,4,8$  分别进行实验。

布线资源的有限性将会最终影响应用电路的实现面积和延时性能，所以从运行时间、总线长、关键路径延时 3 个方面进行比较。加速比是串行算法的运行时间与并行算法的运行时间之比，体现了并行算法的性能。布线质量主要体现在关键路径延时和总线长 2 个指标。总线长即所有线网使用的互连线的长度之和。关键路径是电路中组合逻辑最长的一段路径，决定着电路的时序性能。一般来说，总线长和关键路径延时小于 5% 的增加则证明对性能没有显著影响。

本文提出的算法是对 VPR 中的时序驱动布线算法进行并行化而得到的，由于文献所采用的布线算法均为布通率驱动算法的并行化实现，无法与本文算法直接比较，因此将所得实验结果与 VPR 中的时序驱动布线算法性能相比较。

#### 3.2 测试结果

如图 3~5 及表 1 所示，实验结果表明，本文提出的算法对布线质量的影响在可接受的范围内。关键路径延时方面， $N=2$  时平均减小 2.19%， $N=4$  时平均减小 0.41%， $N=8$  时平均增加 1.78%。总线长比 VPR 中时序驱动布线算法略微增加， $N=2,N=4,N=8$  时平均增加 1.52%,2.16%,2.58%。加速比在  $N=2,N=4,N=8$  下平均为 1.28,1.79,2.46。也就是说，本文算法在布线质量没有受到显著影响的前提下，即线长最大增加 2.58%，关键路径延时最大增加 1.78% 的情况下，8 线程加速比相对于传统 VPR 软件中时序驱动布线算法为 2.46。

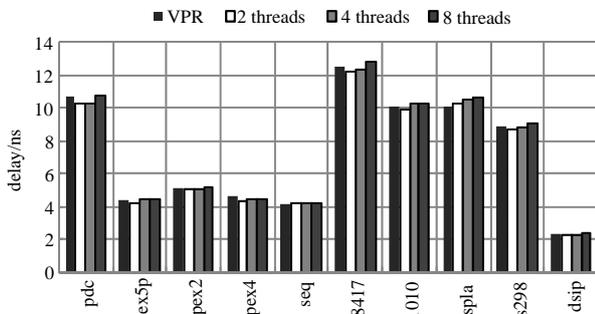


Fig.3 Critical path delay(ns)  
图 3 关键路径延时(ns)

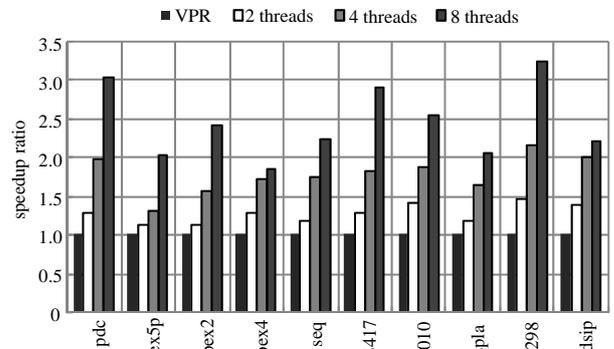


Fig.4 Speedup ratio  
图 4 加速比

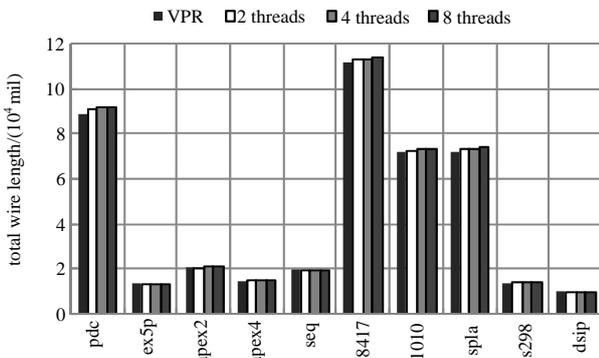


Fig.5 Total wire length  
图 5 总线长

表 1 测试电路运行时间  
Table1 Test circuit runtime

circuit	runtime/s			
	N=1	N=2	N=4	N=8
pdcc	45.188	34.924	22.900	14.928
ex5p	3.522	3.109	2.667	1.743
apex2	5.431	4.805	3.475	2.238
apex4	6.290	5.206	3.548	2.750
seq	6.173	5.206	3.548	2.759
s38417	99.646	77.841	54.665	34.313
ex1010	20.261	14.361	10.760	7.937
spla	18.666	15.745	11.269	9.069
s298	13.113	8.987	6.040	4.039
dsip	5.955	4.293	2.957	2.677

## 4 结论

本文提出一种缩短 FPGA 时序驱动布线算法运行时间的多线程方法。与传统针对布通率驱动布线器的并行方法相比,本文针对时序驱动布线算法特有的性质及数据结构进行分析,并用 POSIX 线程库实现多线程改造,实现高效共享变量,在控制数据通信开销的同时,加速布线迭代收敛。实验结果表明,对于测试选用的 MCNC 电路测试集,该算法在布线质量不受显著影响的前提下,即平均总线长增加 2.58%,平均关键路径延时增加 1.78%的情况下,8 线程平均加速比相对于传统 VPR 软件中的时序驱动布线算法为 2.46。该算法能在保证布线质量的前提下有效提高 FPGA 时序驱动布线算法的效率,具有很强的实用性。

### 参考文献:

- [1] 谭斌,刘平净,李锋. 基于 FPGA 的指纹预处理技术[J]. 太赫兹科学与电子信息学报, 2010,8(2):190-195. (TAN Bin,LIU Pingjing,LI Feng. Fingerprint image preprocessing based on FPGA[J]. Journal of Terahertz Science and Electronic Information Technology, 2010,8(2):190-195.)
- [2] CHAN P K,SCHLAG M D F. Acceleration of an FPGA router[C]// Proceedings of the 5th IEEE symposium on FPGA-based custom computing machines. Washington,DC,USA:IEEE Computer Society, 1997:175-181.
- [3] GORT M,ANDERSON J H. Deterministic multi-core parallel routing for FPGAs[C]// Proceedings of Field Programmable Technology (ICFPT'10). Beijing,China:IEEE, 2010:61-69.
- [4] ZHU C,WANG J,LAI J. A novel net-partition-based multithreaded FPGA routing method[C]// Proceedings of Field Programmable Logic and Applications(FPL'13). Porto,Portugal:IEEE, 2013:2-4.
- [5] MOCTAR Y,BRISK P. Parallel FPGA routing based on the operator formulation[C]// Proceedings of ACM/IEEE Design Automation Conference. San Francisco,CA,USA:IEEE, 2014:1-6.
- [6] SHEN Minghua,LUO Guojie. Accelerate FPGA routing with parallel recursive partitioning[C]// Proceedings of 2015 IEEE/ACM International Conference on Computer-Aided Design(ICCAD). Austin,TX,USA:IEEE, 2015:118-225.
- [7] BETZ V,ROSE J. VPR: a new packing,placement and routing tool for FPGA research[C]// Proceedings of the 7th International Workshop on Field Programmable Logic and Applications. London,UK:Springer-Verlag, 1997:213-222.
- [8] MCMURCHIE L,EBELING C. PathFinder:a negotiation-based performance-driven router for FPGAs[C]// Proceedings of the 3rd International Symposium on Field Programmable Gate Arrays. Napa Valley,CA,USA:IEEE, 1995:111-117.
- [9] 周锋,童家榕,唐璞山. 带时延约束的 FPGA 布线算法[J]. 半导体学报, 1999,20(9):831-836. (ZHOU Feng,TONG Jiarong,TANG Pushan. Routing algorithm for FPGA with time constraints[J]. Chinese Journal of Semiconductors, 1999,20(9):831-836.)
- [10] PACHECO Peter. An introduction to parallel programming[M]. [S.l.]:Elsevier Inc, 2011.
- [11] 陈迅. 面向 FPGA 设计及应用的 EDA 关键技术研究[D]. 长沙:国防科学技术大学, 2011. (CHEN Xun. EDA key technology for the design and application of FPGA[D]. Changsha,Hunan,China:National University of Defense Technology, 2011.)
- [12] 杜昶旭. 面向模拟和数模混合集成电路的自动布线方法研究[D]. 北京:清华大学, 2008. (DU Changxu. Research on automatic routing methodologies for analog and mixed-signal integrated circuits[D]. Beijing,China:Tsinghua University, 2008.)

### 作者简介:



于梦薇(1993-),女,天津市人,在读硕士研究生,主要研究方向为大规模集成电路计算机辅助设计的布线算法及其设计优化.email:yumengwei14@mails.ucas.ac.cn.

刘洋(1983-),男,河南省正阳县人,博士,副研究员,主要研究方向为大规模集成电路设计自动化。

尹韬(1980-),男,山东省青岛市人,博士,研究员,主要研究方向为高性能微传感器接口电路设计、数模混合集成电路设计。

杨海钢(1960-),男,上海市人,博士,研究员,博士生导师,主要研究方向为高速可编程逻辑芯片设计技术、数模混合信号 SOC 设计技术.email:yanghg@mail.ie.ac.cn.