

面向FPGA的布局与布线技术研究综述

田春生^{1,2}, 陈 雷², 王 源¹, 王 硕², 周 婧², 张瑶伟², 庞永江², 周 冲², 马筱婧²,
杜 忠², 薛 钰²

(1. 北京大学信息科学技术学院微纳电子学系, 北京 100871; 2. 北京微电子技术研究所, 北京 100076)

摘 要: 随着大规模集成电路器件复杂度与容量的不断提升, 现场可编程门阵列(Field Programmable Gate Array, FPGA)以高度的并行、可定制和可重构的特性得到了广泛的关注与应用. 在制约FPGA发展的众多因素中, 最为关键的便是电子设计自动化(Electronic Design Automation, EDA)技术, 作为FPGA EDA流程中的关键环节, 布局和布线技术的研究对于FPGA的重要性不言而喻. 本文综述了面向FPGA的布局和布线技术, 包括基于划分的布局、基于启发式的布局、基于解析式的布局、FPGA串行布线和FPGA并行布线等技术, 分析对比了不同技术方法的优缺点, 在此基础上, 本文还展望了未来FPGA布局和布线技术的发展趋势, 将为FPGA未来健康可持续发展提供有力支撑.

关键词: 现场可编程门阵列; 电子设计自动化; 布局; 布线; 并行计算

中图分类号: TN47; TP301

文献标识码: A

文章编号: 0372-2112(2022)05-1243-13

电子学报URL: <http://www.ejournal.org.cn>

DOI: 10.12263/DZXB.20210637

Review on Technology of Placement and Routing for the FPGA

TIAN Chun-sheng^{1,2}, CHEN Lei², WANG Yuan¹, WANG Shuo^{2,2}, ZHOU Jing²,
ZHANG Yao-wei², PANG Yong-jing², ZHOU Chong², MA Xiao-jing², DU Zhong², XUE Yu²

(1. Department of Micro/nanoelectronics, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China;

2. Beijing Microelectronics Technology Institute, Beijing 100076, China)

Abstract: With the continuous increase in the complexity and capacity of large-scale integrated circuit devices, field programmable gate array(FPGA) has received extensive attention and applications for its high degree of concurrency, customizable and reconfigurable features. Among the many factors that restrict the development of FPGA, the most critical is electronic design automation(EDA) technology. As a key link in the FPGA EDA process, the importance of placement and routing technology for FPGA is self-evident. This article reviews the technology of placement and routing for the FPGA, including partition-based placement, heuristic-based placement, analytical-based placement, FPGA serial routing and FPGA parallel routing. The advantages and disadvantages of different technologies are analyzed and compared. On this basis, the development trend of FPGA placement and routing technology in the future is also prospected, which will provide strong support for the healthy and sustainable development of FPGAs.

Key words: field programmable gate arrays; electronic design automation; placement; routing; parallel computing

1 引言

1984年, Xilinx公司首次提出现场可编程门阵列(Field Programmable Gate Array, FPGA)的概念, 作为一种半定制电路, FPGA凭借其设计成本低、速度快、异构逻辑资源丰富等优势, 一经面世便广泛应用于现代数字系统的设计中^[1,2]. 随着各种新兴技术的不断涌现, FPGA的应用范围也在逐步从普通的消费电子向物联网(Internet of Things, IoTs)^[3]、高性能运算^[4]、云计

算^[5]、人工智能^[6,7]等领域不断拓展. FPGA的集成度也在不断增大, 从最初只含有64个逻辑单元发展到迄今为止的超过900万个逻辑单元以及350亿个晶体管. 集成度的不断提升使得FPGA能够被用于设计更加复杂的电路系统, 但同时也对FPGA电子设计自动化(Electronic Design Automation, EDA)软件工具造成了负担^[8]. FPGA的研发离不开EDA工具的支持, 以硬件描述语言(Hardware Description Language, HDL)设计的电

路,通过FPGA EDA工具会被编译为二进制的码流文件,通过该码流文件能够完成对FPGA芯片的配置最终实现所需的电路功能^[9]。但随着FPGA芯片集成度以及电路规模的不断提高,FPGA EDA工具将电路描述文件编译为二进制格式的文件所需花费的时间也越来越长,这将直接降低FPGA在设计阶段的开发效率、增加时间成本,制约FPGA芯片的健康发展^[10,11]。

FPGA EDA工具的设计流程主要包括设计输入、行为综合、逻辑映射、打包、布局、布线以及二进制码流生成。其中,布局与布线作为FPGA EDA工具设计流程中至关重要的环节,其最终结果直接决定了所设计的电路在FPGA芯片上实现后的性能。因此,布局和布线技术的研究对于FPGA的健康可持续发展具有重大的意义。本文全面综述了FPGA布局、布线问题的研究现状与进展,完成了相关理论成果和技术方法的梳理,对FPGA布局和布线的关键技术进行了详细阐述,最后对FPGA布局、布线技术未来的发展趋势进行了展望。

2 FPGA布局关键技术

使用HDL语言设计的电路在经过行为综合、逻辑映射和打包等操作后,会被转化为一个电路网表,FPGA布局就是指根据线长、时延、功耗以及面积等约束条件,将电路网表中的异构逻辑单元与实际FPGA芯片中的物理位置建立一一映射的过程^[12]。现有的FPGA布局技术主要包括基于划分的布局技术、基于启发式的布局技术以及基于解析式的布局技术。

2.1 基于划分的布局技术

基于划分的布局技术是指根据已知的布局约束关系,在最原始的布局实例的基础上,通过递归迭代对布局实例进行划分,直至所有布局实例均小于事先设定的阈值为止^[13]。SimPL布局方法便是基于划分的布局技术中的一种典型的代表^[14],该方法以线长为约束条件,首先分别计算线长的下界和上界,接下来通过采用一种基于自顶向下的几何分割和非线性缩放的前向合法化方法不断迭代,最终能够使逻辑单元收敛到合法的位置。随着图论以及超图划分理论方法的不断演进^[15-19],最小割布局方法也得到了研究学者们的广泛关注。基于划分的布局技术的优点在于通过对较大规模的电路执行递归划分操作,将原始布局问题转化为相互正交的不同的子问题,收敛到最终布局结果的速度较快,但通过使用基于划分的布局技术得到的布局结果,不能保证布局解的最优性,容易陷入到局部最优解当中。

2.2 基于启发式的布局技术

基于启发式的布局技术以现有科学领域中的一些自然现象为设计原理,对FPGA逻辑单元的布局过程进行建模。典型的包括遗传算法^[20,21]、蚁群算法^[22-24]和模

拟退火算法^[25]等,其中又以模拟退火算法的研究最为广泛。1953年,Metropolis等最早提出了模拟退火算法理论^[26],模拟退火算法是一种基于Monte-Carlo求解方式的随机搜索算法,其原理是基于物理中固体物质的退火过程与一般组合优化问题之间的相似性。1988年,Carl等^[27]首次将模拟退火算法的思想引入到超大规模集成电路(Very Large Scale Integration, VLSI)布局领域。随后,越来越多的研究学者开始关注模拟退火算法在VLSI、FPGA等布局领域的应用。学术界较为流行的FPGA通用布局布线(Versatile Placement and Routing, VPR)工具,其布局器的基础框架正是基于模拟退火算法。

使用传统的模拟退火算法对FPGA执行布局操作时,首先会得到一个初始的布局状态,初始布局状态是布局器随机生成的,随后布局器会通过执行逻辑单元的交换或是移动等操作对当前的布局解进行扰动,进一步得到新的布局解并计算得到初始温度。布局器的主体包含内外两层循环,内层循环控制交换的逻辑单元,外层循环执行温度的更新以及逻辑单元交换半径的计算等操作。内层循环每次选择一个逻辑单元,随后依据交换半径在FPGA芯上选择一个位置,若该位置为空则直接将所选择的逻辑单元移动到该位置上,若该位置上已放置有逻辑单元则执行两个逻辑单元的位置交换操作^[28]。以 ΔC 表示逻辑单元交换前后成本函数的差值,如果 $\Delta C < 0$,表示新的布局解要优于原始的布局解,则接受本次的移动或是交换过程;如果 $\Delta C > 0$,则以 $e^{-\Delta C/T}$ 的概率接受新的布局解,其中 T 表示当前温度。这样做的好处在于能够使布局器具备一定的爬坡能力。当内层循环的迭代次数达到上限后,退出内层循环进入外层循环,外层循环会依据当前的布局状态对逻辑单元的交换半径以及温度进行更新。当温度小于特定阈值时跳出外层循环,结束布局过程。模拟退火算法的本质还是一种基于概率的算法,能够在一个很大的空间内寻找近似最优解,在时间允许的情形下,能够得到全局最优解,并且能够有效避免陷入到局部最优陷阱。其缺陷在于当电路规模较大时,算法收敛的时间较长,因此对模拟退火算法的收敛时间进行优化研究显得至关重要^[29]。

利用多个中央处理器进行加速是减少FPGA布局运行时间的有效手段,2010年,Brik等^[30]利用事务性内存(Transaction Memory, TM)对模拟退火算法进行并行加速,在布局质量几乎没有任何下降的前提下,能够获得可观的加速比。2011年,Altera公司(现Intel公司)研究人员^[31]提出了一种确定性的FPGA并行布局算法,能够对关键路径进行优化,在整个退火的过程中最高能够实现高达3.7倍的加速比。2014年,An等^[32]以VPR工具为基础,探索了基于模拟退火机制的FPGA并行布局方

法如何能够在保证布局质量的同时,确保布局结果的确定性.实验结果表明,在保证布局质量以及确定性的情况下,能够获得 5.9 倍的加速比.2018 年,Hu 等^[33]使用多线程同时处理逻辑单元的移动过程,避免了数据流的冲突,实现了 FPGA 布局过程的加速.

通过对模拟退火方法中逻辑单元的移动过程进行优化,从而加速 FPGA 布局过程的收敛同样是一个非常意义的研究方向^[34-37].2009 年,Vorwerk 等^[34]提出了多种策略来创建逻辑单元的移动过程,通过计算每一种移动过程的有效性,以此有效性为指标决定布局器选取每一种移动过程的概率,以便更有效的在 FPGA 布局空间中搜索最优的解决方案,能够获得更好的线长以及时序上的实现效果.Intel 公司的 Quartus II 布局器中便采用了多达 10 余种的直接移动(Directed Moves, DM)过程,但在布局时 DM 过程的选择机制却从未对外披露^[31].2017 年,Yuan 等^[38]以线长为指标,提出了基于区域范围的移动过程,将逻辑单元的移动操作限制在某一特定的区域内,同时通过设置优先级的方法,优先对那些性能较差的线网执行布局操作.2019 年,Yuan 等^[39]对此项工作做了进一步拓展,在布局的过程中加入了时序优化.此外,随着机器学习、人工智能等技术的发展,越来越多的研究学者利用机器学习类方法来优化基于模拟退火算法的 FPGA 布局过程^[40].2019 年,Murray 等^[41]使用单状态强化学习策略提出了一种自适应的 FPGA 布局方法,在每一次布局的过程中能够自动地选取需要交换的逻辑单元,节省了布局所需要的时间,与 VPR 工具相比较,在获得同等性能参数的前提下,能够获得 2 倍以上的加速比.在此项工作基础上,2020 年,El-gammal 等^[42]做了进一步探索,使用多状态强化学习策略以优化 DM 过程的选取过程,基于模拟退火方法的 FPGA 布局器的性能得到了进一步优化.

2.3 基于解析式的布局技术

基于解析式的布局技术将 FPGA 布局问题建模为一个连续优化问题,通过梯度下降的方法得到布局问

题的最优解.根据目标函数的不同,可以分为二次方布局技术^[43-46]以及非线性布局技术^[47,48].二次方布局技术是指通过二次函数来逼近最终的布局解,而非线性布局技术是指利用更加具有表现力的高阶方法来完成对布局结果的近似.

二次方布局技术通常以线长为优化目标完成对目标函数的建模,随后通过 Krylov 子空间方法将上述优化问题转化为对称与正定(Symmetric and Positive-Definite, SPD)线性系统的求解问题,最终得到优化问题的解^[49].但纯粹以线长为优化目标,最终的布局结果将导致越来越多的逻辑单元重叠在一起,因此需要采用合适的方法对二次方布局技术进行优化,以减少各逻辑单元间的重叠.其中,力导向法得到了广泛的关注.在每一次布局迭代的过程中,力导向法能够使逻辑单元均匀分散开,这些逻辑单元的位置将被作为锚节点,随后,将指向这些逻辑单元的额外的拓展力应用于下一次迭代中,以辅助逻辑单元的布局,不断重复上述过程,最终能够得到一个所有逻辑单元的位置均互不重叠的布局解决方案^[50,51].与二次方布局技术不同,非线性布局技术使用不可微分的非线性函数来表示非重叠的约束条件,并将其与线长一起在统一的目标函数中进行求解.由于高阶非线性函数比二次函数更具有表现力,因此,通过非线性布局技术得到的布局解的质量要优于二次方布局技术,但这种布局质量的改进需要牺牲更多的系统运行时间^[52].

基于解析式的布局技术其布局结果的可接受度以及系统运行时间要优于基于划分的布局技术与基于启发式的布局技术.然而,为了更加高效地利用 FPGA 芯片资源,基于解析式的布局技术在一些具体的布局细节方面仍需进一步优化,这些具体的布局细节通常以低温模拟退火方法或是一些基于贪婪思想的启发式方法来实现^[53,54].

本文总结了基于划分的布局技术、基于启发式的布局技术与基于解析式的布局技术的优势与不足,具体如表 1 所示.

表 1 FPGA 布局技术总结

布局技术	优势	不足
基于划分的布局技术	将复杂的电路设计分割为较简单的电路设计,能够快速对局部电路进行处理,收敛速度快	布局质量上存在劣势,容易陷入局部最优解
基于启发式的布局技术	相对于最优算法而提出,在时间允许的情况下,能够获得布局的最优解	运行时间长,布局收敛速度慢
基于解析式的布局技术	完全依赖于数学方法,最终的布局质量能够通过数学分析来证明,具有较高的布局质量,收敛较快	较高的数学功底以抽象 FPGA 布局问题,不同的 FPGA 抽象形式不同,不能通用

3 FPGA 布线关键技术

FPGA 布线问题是指通过对芯片中的可编程互连

开关进行配置,从而完成对电路逻辑单元间的连接,同时需要保证所有的互连资源都不能被重复使用.在布

线时, FPGA 芯片内的硬件资源会被抽象为有向的布线资源图(Routing Resource Graph, RRG) $G=(V, E)$, 其中, V 表示 RRG 中顶点的集合, 每一个顶点 $v, \forall v \in V$ 表示 FPGA 芯片内的互连线资源或是逻辑单元的引脚. E 表示 RRG 中边的集合, 每一条边 $e_{i,j}, \forall e_{i,j} \in E$ 表示逻辑资源引脚与互连线资源或是互连线资源间的可编程互连开关^[55]. RRG 作为布线算法与 FPGA 硬件间的纽带, 能够将 FPGA 布线问题转化为图论中最短路径的求解问题, 布线算法直接在 RRG 上对电路网执行布线操作即可. 每个电路网 N_i 由一个源端 $s_i, \forall s_i \in V$ 与一个或多个漏端 $t_{i,j}, \forall t_{i,j} \in V$ 组成. 有了 RRG 的概念, 线网 N_i 的布线问题被转化为能否在 RRG 中找到合法的路径连接 s_i 与 $t_{i,j}$, 这些路径共同组成线网 N_i 的布线树(Routing Tree, RT) $RT(N_i) \subset G$.

1995 年, McMurchie 等^[56]首次提出了基于拥塞协商思想的 FPGA 布线方法 PathFinder, 它允许 FPGA 中互连资源被重复使用的情况下执行线网的布线操作, 迭代地对电路中的线网进行“拆线-重布”, 每次迭代时 PathFinder 会以协商的方式确定各线网间互连资源的分配. 随着迭代次数的增加, PathFinder 会逐渐增加对重复使用的互连资源的惩罚力度, 以此来逐步消除互连资源的重复使用.

PathFinder 方法的实现是一个内外三层循环的嵌套结构, 最外层的循环被称为全局布线器, 在每一次迭代时会调用第二层循环中的线网布线器, 对电路中的线网逐一执行布线操作. 全局布线器在所有线网完成布线迭代后, 会将布线资源的历史拥挤度进行更新, 并根据布线迭代的结果对线网执行时序分析的操作. 检查是否有布线资源被重复使用, 若所有布线资源均没有被重复占用, 则结束布线流程; 否则, 此次布线迭代过程非法, 继续执行下一次布线迭代操作. 当线网布线器对电路中的第 i 个线网 N_i 执行布线操作时, 线网布线器会释放线网 N_i 所对应的 $RT(N_i)$ 上的资源, 并对节点的当前拥挤度进行更新, 随后根据连接的关键度信息对线网 N_i 中所有的漏端 $t_{i,j}$ 执行降序排列操作, 最后线网布线器会调用最内层的迷宫布线器依次对线网 N_i 内的每一个漏端 $t_{i,j}$ 进行布线. PathFinder 拥塞协商方法起到了引路者以及奠基人的角色, 它是目前大部分 FPGA 布线算法的基础, 学术上通用的 VPR 工具以及 nextpnr 工具在布线阶段使用的都是在 PathFinder 基础上改进的布线方法. 同时, 现阶段几乎所有的 FPGA 厂商都在使用基于拥塞协商的 PathFinder 布线方法, 或是在这个方法基础上引申出来的其他布线方法, 但随着电路规模的增大, PathFinder 布线方法耗时越来越长, 这一问题随着 FPGA 的发展将愈发突出. 因此, 亟需探索新的解决方案或是对现有解决方案进行优化以适应

FPGA 不断发展的需求^[57]. 现有的解决方案可以分为 FPGA 串行布线与 FPGA 并行布线两种类型.

3.1 FPGA 串行布线技术

现有的 FPGA 串行布线技术都是基于 PathFinder 方法的改进. 将 FPGA 电路中常用的逻辑单元打包成宏并提前进行编译, 是减少布线时间的有效手段^[58,59]. 2011 年, Lavin 等^[59]通过设置硬宏的方法将 FPGA 电路中常用的一些逻辑单元提前进行编译并在数据库中加以保存, 在对用户设计电路进行编译时便可以从数据库中选择与之相匹配的硬宏. 这种方法可以明显减少电路编译的时间, 同 Xilinx 公司的 ISE 工具相比较, 此方法能够获得 10 到 50 倍的加速比, 改进效果是非常可观的. 2012 年, Coole 等^[60]构建了一种新的 FPGA 布局布线工具, 通过设置宏单元能够使布线方法在更高的层次上运行, 能够获得指数级的加速比. 但由于宏单元是提前设置并封装好的, 在实际布线时其中的连接关系是无法更改的, 增加了 FPGA 布线时的局限性. 简化 FPGA 的布线结构是另一种降低 FPGA 布线时间的方法^[61]. PathFinder 方法在每次对线网执行布线迭代时需要确定具体使用的是布线通道中的哪条布线资源, 2015 年, Ferreira 等^[62]提出在使用 PathFinder 方法确定布线通道后, 将布线通道中互连资源的分配问题建模为布尔满足问题^[63,64], 随后使用求解器 MiniSat 对上述问题进行了求解^[65]. 通过简化 FPGA 结构的方法, 能够明显降低 FPGA 布线方法复杂度, 但此类方法过分依赖 FPGA 硬件架构, 并不适用于商用 FPGA 芯片的设计流程, 实用性有待进一步加强.

PathFinder 方法以线网为单位每次执行“拆线-重布”的操作, 在检测到存在布线资源重复使用的情况时, 对所有线网进行拆分, 依次执行重新布线. 上述过程是非常耗时的. 为此, 2016 年, 比利时根特大学研究人员^[66,67]提出了基于连接的 FPGA 布线方案, 连接由线网 N_i 中的单个源端 s_i 与单个漏端 $t_{i,j}$ 组成, 在每次迭代的过程中以连接为单位执行“拆线-重布”的操作, 只有当不同的连接属于同一个线网时才可以共享相同的布线资源节点. 与 VPR 7.0 相比基于连接的 FPGA 布线方案在布线阶段能够提供 3.4 倍的加速比^[67]. 与基于连接的 FPGA 布线方案不同, 2020 年, Murray 等^[68]设计了一种自适应的增量布线器, 能够保持一种自然的基于线网的结构, 在每次“拆线-重布”时只拆除非法的布线资源节点, 从而保证在下次迭代时, 合法的路径能够被重复使用, 减少了在 RRG 中重复搜索所需要的时间. 同时, 针对一些具有高扇出属性的线网制定了特殊的前向搜索机制, 即每次只对与漏端 $t_{i,j}$ 空间上相近的布线节点进行搜索, 以降低布线所需时间. 相比于基于连接的 FPGA 布线方案, 能够进一步减少 17% 的布线时间.

3.2 FPGA 并行布线技术

随着多核处理器与图形处理器的普及,越来越多的学者将并行计算应用到 FPGA 布线领域以减少 FPGA 在布线时花费的时间^[69]. 根据并行化方式的不同,又可以将并行布线技术分为粗粒度 FPGA 并行布线技术^[70]、细粒度 FPGA 并行布线技术^[71]以及两种不同并行布线技术的混合^[72]. 粗粒度 FPGA 并行布线技术通过将线网划分为不同的分区,在每个分区内执行独立的布线操作. 最终布线后的性能参数取决于线网的划分方式、并行运算过程中数据的同步模式以及冲突的回避方式等多种因素. 细粒度 FPGA 并行布线技术能够对单个线网的布线过程实现并行加速,最终的性能参数取决于并行布线过程中的并行度以及共享数据的同步方式等诸多因素. 此外,评判某一 FPGA 并行布线方法的性能还需将布线方法的可扩展性与确定性等问题联合起来进行考虑. 本文对不同的 FPGA 并行布线技术进行了分析总结(如表 2 所示).

3.2.1 粗粒度 FPGA 并行布线技术

(1) 基于字典序的 FPGA 并行布线

1997 年,Chan 等^[73]首次尝试将 PathFinder 方法执行并行化操作,研究人员首先将所有线网按字典序进行排序,从而使那些可能共享资源的线网相邻并尽可能分配给同一个中央处理器(Central Processing Unit, CPU). 随后,将线网列表划分成互不相交的集合,每个集合静态地分配给 CPU 执行并行布线. 并行布线的过程中,每个 CPU 会单独保存线网布线的拥塞状态等信息,一旦某个 CPU 的拥塞状态信息发生变化,该 CPU 将首先更新本地保存的数据信息,并通过 UNIX 套接字通知其他处理器对拥塞信息进行更新. 与原始 PathFinder 算法相比较,上述方法最高能够获得 3.2 倍的加速比. 但此方法的布线结果具有不确定性,布线结果依赖于线网的布线顺序,并且由于是静态的线网划分,每个 CPU 间存在负载不均衡的问题. 2000 年,上述团队的研究人员^[74]对 CPU 负载不均衡的问题进行了优化,但布线结果仍然具有不确定性.

(2) 基于负载均衡的 FPGA 并行布线

2012 年,Gort 等^[75]提出了多进程并发执行的方式以加速布线过程,这些并发执行的进程运行在不同的 CPU 内核中(如图 1 所示). 每个进程运行一个单独的 VPR 实例,负责对分配给它的线网执行布线操作并维护自身的数据结构. 不同 VPR 实例间的信息同步通过消息传递接口(Message Passing Interface, MPI)来实现.

VPR 实例在完成某一信号的布线后,会向其他 VPR 实例发送无阻塞更新信息,随后继续执行布线操作而无需等待从其他 VPR 实例接收更新消息. VPR 实

例发送的更新信息中包含了最新的布线信息,这些信息被保存在 MPI 的信息队列中. 为了接收更新的消息, VPR 实例会在布线过程中的特定位置发出阻塞接收的调用请求,并且为了保证布线结果的确定性,这些位置在每次运行过程中都是相同的. 如果某一时刻 VPR 实例想要接收信息但更新消息却不可用,由于先前已经调用了阻塞接收的请求, VPR 实例会暂停其布线流程直至消息更新可用为止. 为了最大程度地减少 VPR 实例驻留时间,作者会在任务划分的过程中保证每个 VPR 实例的负载都尽可能相等,并且仅在 MPI 消息队列中存在 VPR 实例想要更新的消息时,才会允许 VPR 实例发出阻塞接收的调用请求. 与此同时,为了能够精准地确定阻塞接收调用请求的发出时间,每个 VPR 实例都会维护一个工作计数器以评估每个 VPR 实例所执行的工作量.

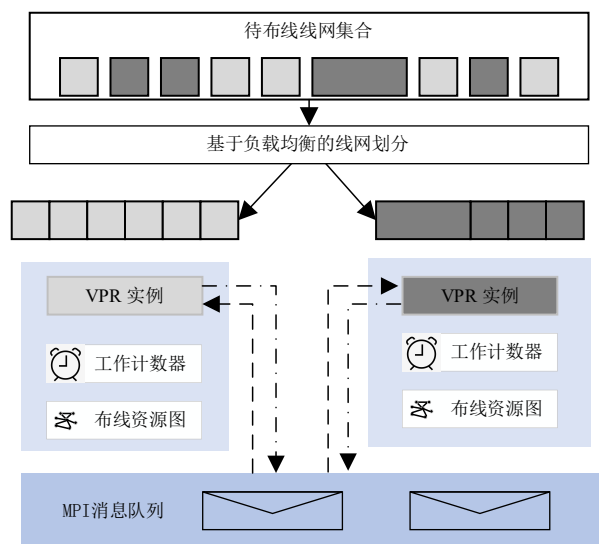


图1 基于负载均衡的FPGA并行布线

(3) 基于迭代划分的FPGA并行布线

北京大学团队研究人员^[76]与中山大学团队研究人员^[77]在前人研究工作的基础上探索了基于迭代划分的FPGA并行布线方案,通过递归划分的方法能够将线网划分为多个集合,每个集合内的线网互不相交,进而可以执行并行布线的操作.

首先,使用二分法能够将FPGA划分为两个子区域,根据划分的结果能够得到三个不同的线网集合:横跨两个子区域的线网集合以及完全在两个子区域内的线网集合. 由于在两个子区域内的线网互不相交,因此可以执行并行布线的操作. 可以通过迭代的方式重复上述步骤以增加布线方法的并行度.

图2展示了基于迭代划分的FPGA并行布线方法的原理. 首先,图中的蓝色实线将FPGA分成了上下两个区域: S_1 与 S_2 . 随后,图中绿色的虚线进行了进一步

的划分:将区域 S_1 划分为 S_3 与 S_4 两个子区域,将区域 S_2 划分为 S_5 与 S_6 两个子区域.最后,图中红色的虚线又进一步执行了划分的操作:将区域 S_3 划分为 S_7 与 S_8 两个子区域,将区域 S_4 划分为 S_9 与 S_{10} 两个子区域,将区域 S_5 划分为 S_{11} 与 S_{12} 两个子区域,将区域 S_6 划分为 S_{13} 与 S_{14} 两个子区域.上述划分过程可以使用一个 L 级二叉树表示(如图2(a)所示).划分完成后,布线问题被建模为分布式内存系统的任务调度问题:任务 T_0 负责对所有的线网执行布线操作,任务 T_1 负责对子区域 S_1 内的线网执行布线操作,任务 T_2 负责对子区域 S_2 内的线网执行布线操作,依此类推,任务 T_i 负责对子区域 S_i 内的线网执行布线操作.上述任务被静态地映射到不同进程,不同进程使用MPI进行数据间的同步(如图2(b)所示).布线过程从主进程 $p=0$ 开始执行,即在一个 L 级的完美二叉树上执行任务 T_0 . T_0 首先将线网划分为 S_1 与

S_2 两个子区域,接下来 T_0 会对横跨两个子区域的线网集合执行串行布线的操作,布线结束后,它会将右侧的 $L-1$ 级子树分配给进程 $p+2^{L-1}=p+4$.最后, T_0 通过递归调用函数ParMaze()会将左侧的 $L-1$ 级子树分配给任务 T_1 .随后,进程 $p+4$ 与任务 T_2 会返回子区域 S_2 内线网的布线树,任务 T_1 会返回子区域 S_1 内线网的布线树.不断重复上述过程,直至所有的线网都完成了布线操作.

但随着递归迭代次数的增加,跨区域线网的数量也在同时增多,导致FPGA并行布线方法的并行度越来越低.针对上述问题,2020年,Wang等^[78]提出了一种混合线网划分机制,在线网递归划分的过程中将线网划分为互斥线网集合与重叠线网集合,随后针对两组不同的集合采取不同的并行布线策略,并且在布线的过程中可以很好地解决资源冲突以及负载不均衡的问题.

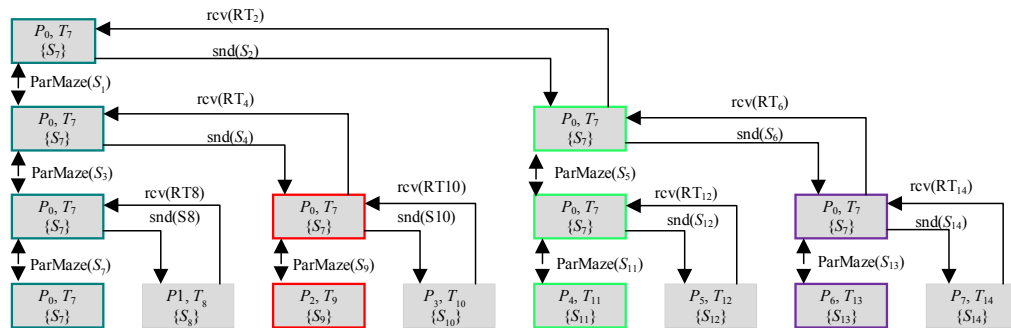
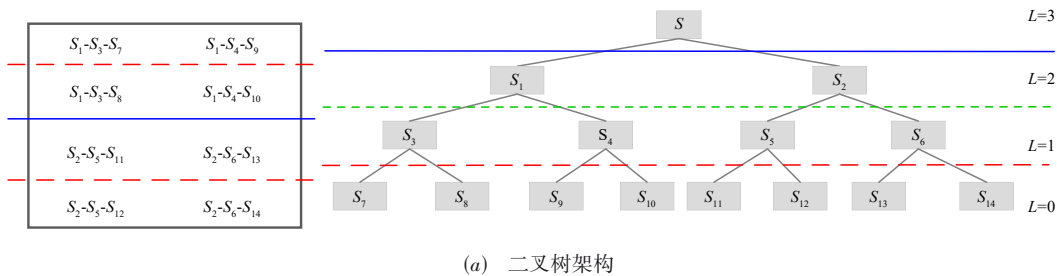


图2 递归划分原理

3.2.2 细粒度FPGA并行布线技术

不同于粗粒度FPGA并行布线技术,细粒度FPGA并行布线技术在布线时不会改变线网的布线顺序.据统计,约70%的布线时间花费在了搜索邻居节点以及优先级队列的处理上^[79].因此,如何对PathFinder算法的迷宫布线器进行细粒度加速是当前亟需解决的问题^[80-82].

(1) 基于Galois模型的FPGA并行布线

2014年,Moctar等^[83]率先探索了使用Galois模型对PathFinder算法中的迷宫布线器进行并行加速的操作,并在VPR 5.0的基础上对该项工作进行了实现.

多线程的并行策略基于Galois迭代合并机制,即允许单个线程同时处理多个迭代.每个线程都会维护一个本地优先级队列,优先级队列中存储了需要进行拓展的节点,若本地优先级为空,线程将访问存储在共享内存空间中的全局优先级队列.上述工作使用互斥锁实现不同线程间的同步,但当某些情况下无法获取互斥锁时,Galois会根据先前保存的数据副本将冲突进行回滚,以尽可能降低不同线程间的数据冲突.同VPR 5.0相比,基于Galois模型的FPGA并行布线方法最高能够获得5.46倍的加速比.但根据此方法得到的布线结果具有不确定性,布线结果依赖于线程完成工

作的先后顺序. 此外, Galois 在回滚的过程中将会产生大量的开销, 严重影响了该方法的可扩展性. 2018 年, 上述团队研究人员^[84]对此项工作进行了扩展, 消除了因数据回滚操作而带来的影响, 通过 Galois 模型多线程能够同时对单个线网执行并行布线的操作, 在布线结果具有确定性的情况下, 能够实现 3.67 倍的加速比.

(2) 基于 Posix 线程的 FPGA 并行布线

2010 年, Gort 等^[79]提出使用 Posix 线程来加速迷宫布线器中成本函数的计算过程. 当存在 N 个可用的线程时, 将其拆分为一个主线程与 $N-1$ 个辅助线程(如图 3 所示). 主线程执行迷宫布线器的串行部分以及 $1/N$ 的并行部分. 每个线程都会维护自己的优先级队列, 主线程首先会查看所有优先级队列的顶部节点, 随后借助同步路障选取出成本函数最低的节点, 在此基础上, 主线程会拓展这个成本函数最低的节点并通知所有的辅助线程做好准备工作. 接下来, 所有的线程会计算待拓展节点的成本函数并将其添加到各自的优先级队列中. 一旦主线程完成了添加节点的操作, 它会在第二道同步路障处等待其他线程, 不断重复上述步骤, 直至迷宫布线器完成布线.

Zhu 等^[85]同时考虑粗粒度与细粒度的 FPGA 并行布线, 提出利用线程来代替进程以减少通信开销, 根据线网的扇出将线网划分为高扇出线网与低扇出线网两种类型. 高扇出线网会被进一步划分为互不重叠的低

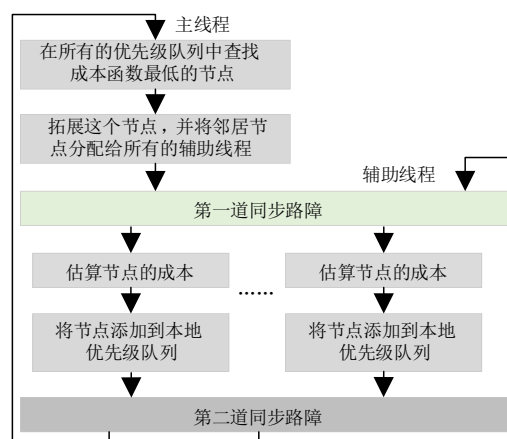


图3 基于 Posix 线程的 FPGA 并行布线

扇出线网的集合, 从而能够执行并行布线的操作. 对于低扇出线网, 该团队研究人员使用互不重叠的边界框对线网进行标记, 并同时执行布线操作. 上述布线方法最终在 VPR 5.0 架构的基础上通过 Posix 多线程技术进行了实现. 与文献^[81]的方法相类似, 在实现的过程中由主线程负责所有的辅助线程, 并通过设置同步路障的方式完成线程间的同步. 此并行布线方法同样包含两道同步路障, 第一道同步路障负责唤醒所有的辅助线程对线网执行并行布线操作. 当所有辅助线程完成布线任务后, 第二道同步路障负责通知主线程将布线结果合并、更新阻塞信息等, 进入下一次布线迭代. 通过该方法得到的布线结果具有较好的确定性.

表2 FPGA 并行布线技术

分类	Benchmarks	串行对比基准	并行模型	同步方法	确定性	可扩展性	文献
粗粒度	字典序法	alu4, s1A	PathFinder	分布式内存	消息传递	不具备	扩展性低 [73]
	负载均衡法	MCNC, VPR 5.0	VPR 5.0	分布式内存	消息传递	具备	扩展性低 [75]
	迭代划分法	VPR 7.0	VPR 7.0	共享式内存	同步锁	具备	扩展性高 [70]
		VPR 7.0	VPR 7.0	分布式内存	消息传递	具备	扩展性低 [76]
		VPR 7.0	VPR 7.0	分布式内存	消息传递	具备	扩展性高 [77]
		VPR 7.0, Titan	VPR 7.0	共享式内存	同步锁	具备	扩展性高 [78]
细粒度	Posix 线程	VPR 8.0	VPR 8.0	共享式内存	消息传递	具备	扩展性高 [71]
		MCNC, VPR5.0	VPR5.0	分布式内存	同步路障	具备	扩展性低 [79]
		VPR 7.0	VPR 7.0	共享式内存	同步锁	具备	扩展性高 [80]
		VPR 7.0	VPR 7.0	共享式内存	同步锁	具备	扩展性高 [82]
	Galois 模型	IWLS	VPR 5.0	共享式内存	同步锁	不具备	扩展性高 [83]
		IWLS	VPR 5.0	共享式内存	同步锁	具备	扩展性高 [84]
混合粒度	/	Titan	VPR 7.0	共享式内存	同步锁	不具备	扩展性高 [72]
		MCNC	VPR 5.0	共享式内存	同步路障	具备	扩展性低 [85]

4 未来发展趋势

迄今为止, 国内外科研人员对 FPGA 布局与布线技术有了较为深入的研究, 并提出了一些具有代表性的研究成果. 从当前的研究成果可以得出: 关于 FPGA 布

局技术, 主要可以分为基于划分的布局技术、基于启发式的布局技术以及基于解析式的布局技术三种类型, 三种类型的 FPGA 布局技术具备各自的优缺点, 在应用的过程中可根据需求进行选择. 关于 FPGA 布线技术,

均是以基于拥塞协商的 PathFinder 布线技术为根基,在此基础上又能够细分为 FPGA 串行布线技术与 FPGA 并行布线技术。相比较于串行布线技术,并行布线技术能够获得更高的加速比,但在布线的过程中也面临着资源竞争、布线方法的确定性以及可扩展性等诸多问题。通过上述分析可知,对 FPGA 布局和布线技术的研究仍然任重而道远,基于现有的研究基础,布局与布线技术未来的发展趋势可以归纳为以下几个方面:

(1) 利用人工智能技术优化 FPGA 布局和布线的流程。FPGA 电路规模越大,对布局与布线技术的要求就越高,系统运行时间便越长。FPGA 布局与布线问题属于典型的启发式探索问题,变量空间大,难以寻找全局最优解,而人工智能技术是非常适合于此类高维数据空间问题求解的。但现阶段却鲜有将人工智能技术应用于 FPGA 布局与布线领域的研究。因此,随着人工智能技术与相关工具的发展,可以尝试突破现有方法的架构,将人工智能等方法应用到 FPGA 布局与布线的流程中,在不影响收敛速度前提下,提升 FPGA 布局与布线解的质量。

(2) 探索多驱动的布局和布线策略。随着 FPGA 应用领域的不断拓展,在布局与布线的过程中仅考虑线长、时序等信息是远远不够的。例如,现阶段在航空、航天等各重要核心领域的电子系统中,FPGA 在正扮演一个其它芯片无法替代的角色。但随着 FPGA 的大规模使用,其在空间应用中的辐照问题也越来越突出。宇宙空间中存在着大量的高能粒子,这些高能粒子会对 FPGA 电子器件的功能造成影响,产生软错误。但如果在布局与布线的过程中就考虑到了软错误的影响,便有可能将软错误产生的概率降至最低。因此,在实际应用的过程中,需要将线长、时延、软错误、拥挤度、功耗等诸多因素联合起来进行考虑,以满足未来 FPGA 全方位发展的需求,同时这也是在今后研究中需要进一步探索的方向。

(3) 研究更加简单可行的 FPGA 并行布局与布线方法。通过并行计算的方法能够显著提升 FPGA 布局和布线的速度,减少使用 EDA 工具将 FPGA 电路编译为二进制码流所需要的时间。但随着 FPGA 电路的规模越来越大,现有的并行布局与布线方法的执行难度也在不断加大,并行方法的确定性以及可扩展性问题越来越突出。在未来的工作中,可以尝试研究更加简单可行的 FPGA 并行布局、布线方法,从而加大布局、布线方法在执行过程中的并行度,进一步减少 FPGA EDA 工具在布局 and 布线阶段所需花费的时间。

5 结论

随着科技的不断发展,现代工业中所使用的 FPGA 电路的规模也在不断增大,对 FPGA 布局和布线技术提出了更高的要求。本文围绕着 FPGA 布局与布线问题

展开,从 FPGA 布局关键技术与布线关键技术 2 个方面对研究进展做了分析与总结。在此基础上,对未来的发展趋势进行了展望,以其对未来本领域的研究具有借鉴意义。我们坚信,随着研究人员对布局与布线技术的不断探索,终将满足未来 FPGA 发展的需求,为 FPGA 更普遍的应用提供有力支撑。

参考文献

- [1] TRIMBERGER S M. Three ages of FPGAs: A retrospective on the first thirty years of FPGA technology[J]. Proceedings of the IEEE, 2015, 103(3): 318-331.
- [2] TRIMBERGER S M. Three ages of FPGAs: a retrospective on the first thirty years of FPGA technology: this paper reflects on how Moore's law has driven the design of FPGAs through three epochs: the age of invention, the age of expansion, and the age of accumulation[J]. IEEE Solid-State Circuits Magazine, 2018, 10(2): 16-29.
- [3] IMANI M, ZOU Z W, BOSCH S, et al. Revisiting hyperdimensional learning for FPGA and low-power architectures [C]//2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA). South Korea: IEEE, 2021: 221-234.
- [4] ZHANG J X, YANG T, LI Q Z, et al. An FPGA-based neural network overlay for ADAS supporting multi-model and multi-mode [C]//2021 IEEE International Symposium on Circuits and Systems (ISCAS). South Korea: IEEE, 2021: 1-5.
- [5] KAN H W, LI R G, SU D D, et al. Trusted edge cloud computing mechanism based on FPGA cluster [C]//2020 IEEE 8th International Conference on Computer Science and Network Technology (ICCSNT). China: IEEE, 2020. 146-149.
- [6] 刘公绪, 史凌峰, 辛东金. 基于 FPGA 的零误差大数阶乘算法的设计与实现[J]. 电子学报, 2019, 47(5): 1180-1184.
LIU G X, SHI L F, XIN D J. FPGA-based zero-error factorial algorithm for large integer[J]. Acta Electronica Sinica, 2019: 47(5): 1180-1184. (in Chinese)
- [7] 蹇强, 张培勇, 王雪洁. 一种可配置的 CNN 协加速器的 FPGA 实现方法[J]. 电子学报, 2019, 47(7): 1525-1531.
JIAN Q, ZHANG P Y, WANG X J. An FPGA implementation method for configurable CNN co-accelerator[J]. Acta Electronica Sinica, 2019, 47(7): 1525-1531. (in Chinese)
- [8] LIN Y B, JIANG Z X, GU J Q, et al. DREAMPlace: Deep learning toolkit-enabled GPU acceleration for modern VLSI placement[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2021, 40(4): 748-761.
- [9] LUU J, GOEDERS J, WAINBERG M, et al. VTR 7.0:

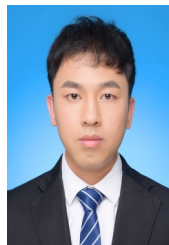
- next generation architecture and CAD system for FPGAs[J]. *ACM Transactions on Reconfigurable Technology and Systems*, 2014, 7(2): 1-30.
- [10] DHAR S, SINGHAL L, IYER M, et al. FPGA accelerated FPGA placement[C]//2019 29th International Conference on Field Programmable Logic and Applications (FPL). Barcelona: IEEE, 2019: 404-410.
- [11] ZAPLETINA M A, ZHELEZNIKOV D A, GAVRILOV S V. Improving pathfinder algorithm performance for FPGA routing[C]//2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering(El-ConRus). Russia: IEEE, 2021: 2054-2057.
- [12] 王德奎. 一种利用资源协商的 FPGA 布局方法[J]. *西安电子科技大学学报*, 2019, 46(6): 17-22
WANG D K. Approach to FPGA placement using resource negotiation[J]. *Journal of Xidian University*, 2019, 46(6): 17-22. (in Chinese)
- [13] KHATKHATE A, LI C, AGNIHOTRI A R, et al. Recursive bisection based mixed block placement[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2005, 24(5): 748-761.
- [14] KIM, M C, LEE D J, MARKOV I L. SimPL: an effective placement algorithm[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2012, 31(1): 50-60.
- [15] MAIDEE P, ABABEI C, BAZARGAN. Timing-driven partitioning-based placement for island style FPGAs[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2005, 24(3): 395-406.
- [16] FIDUCCIA C M, MATTHEYSES R M. A linear-time heuristic for improving network partitions[C]//19th Design Automation Conference. USA: IEEE, 1982: 175-181.
- [17] KARYPIS G, KUMAR V. Parallel multilevel k-way partitioning scheme for irregular graphs[C]//Supercomputing Proceedings of the 1996 ACM/IEEE Conference on Supercomputing. USA: IEEE, 1996: 1-1.
- [18] KARYPIS G, KUMAR V. Multilevel k -way hypergraph partitioning[C]//Proceedings 1999 Design Automation Conference (Cat. No.99CH36361). USA: IEEE, 1999: 343-348.
- [19] KERNIGHAN B W, LIN S. An efficient heuristic procedure for partitioning graphs[J]. *The Bell System Technical Journal*, 1970, 49(2): 291-307.
- [20] VEREDAS F J, CARMONA E J. FPGA placement improvement using a genetic algorithm and the routing algorithm as a cost function[C]//2018 21st Euromicro Conference on Digital System Design(DSD). Czech Republic: IEEE, 2018: 70-76.
- [21] SADEGHI A, LIGHVAN M Z, PRINETTO P. Automatic and simultaneous floorplanning and placement in field-programmable gate arrays with dynamic partial reconfiguration based on genetic algorithm[J]. *Canadian Journal of Electrical and Computer Engineering*, 2020, 43(4): 224-234.
- [22] HERATH K, PRAKASH A, JIANG G Y, et al. Ant Colony Optimization based module footprint selection and placement for lowering power in large FPGA designs[C]//2018 International Conference on ReConfigurable Computing and FPGAs(ReConFig). Mexico: IEEE, 2018: 1-8.
- [23] DANASSIS P, SIOZIOS K, SOUDRIS D. ANT3D: simultaneous partitioning and placement for 3-D FPGAs based on Ant Colony Optimization[J]. *IEEE Embedded Systems Letters*, 2016, 8(2): 41-44.
- [24] JING C. Ant-Colony Optimization based algorithm for energy-efficient scheduling on dynamically reconfigurable systems[C]//2015 Ninth International Conference on Frontier of Computer Science and Technology. China: IEEE, 2015: 127-134.
- [25] SARI A, PSARAKIS M. Scrubbing-aware placement for reliable FPGA systems[J]. *IEEE Transactions on Emerging Topics in Computing*, 2020, 8(3): 564-576.
- [26] STEINBRUNN M, MOERKOTTE G, KEMPER A. Heuristic and randomized optimization for the join ordering problem[J]. *The VLDB Journal*, 1997, 6: 191-208.
- [27] CARL S. VLSI Placement and Global Routing Using Simulated Annealing[M]. Boston, MA, USA: Springer Science & Business Media, 2012.
- [28] MURRAY K E, PETELIN O, ZHONG S, et al. VTR 8: high performance CAD and customizable FPGA architecture modelling[J]. *ACM Transactions on Reconfigurable Technology and Systems*, 2020, 13(2): 1-55.
- [29] KAGAWA H, ITO Y, NAKANO K, et al. Fully-pipelined architecture for simulated annealing-based QUBO solver on the FPGA[C]//2020 Eighth International Symposium on Computing and Networking(CANDAR). Japan: IEEE, 2020: 39-48.
- [30] BRIK S, STEFFAN J G, ANDERSON J H. Parallelizing FPGA placement using transactional memory[C]//2010 International Conference on Field-Programmable Technology. China: IEEE, 2010: 61-69.
- [31] LUDWIN A, BETZ V. Efficient and deterministic parallel placement for FPGAs[J]. *ACM Transactions on Design Automation of Electronic Systems*, 2011, 16(3): 1-23.
- [32] AN W, STEFFAN J G, BETZ V. Speeding up FPGA

- placement: parallel algorithms and methods[C]//2014 IEEE 22nd Annual International Symposium on Field-Programmable Custom Computing Machines. USA: IEEE, 2014: 178-185.
- [33] HU C Y, LU P, YANG M, et al. A SA-based parallel method for FPGA placement[J]. IEICE Electronics Express, 2018, 15(24): 1-8.
- [34] VORWERK K, KENNINGS A, GREENE J W. Improving simulated annealing-based FPGA placement with directed moves[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2009, 28(2): 179-192.
- [35] BETZ V, ROSE J, MARQUARDT A. Architecture and CAD for Deep-submicron FPGAs[M]. Boston, MA, USA: Springer Science & Business Media, 2012.
- [36] HENTSCHEKE R F, REIS R. Improving simulated annealing placement by applying random and greedy mixed perturbations[C]//Proceedings of the 16th Symposium on Integrated Circuits and Systems Design. Brazil: IEEE, 2003: 267-272.
- [37] CHEN G, CONG J. Simultaneous timing-driven placement and duplication[C]//Proceedings of the 2005 ACM/SIGDA 13th International Symposium on Field-Programmable Gate Arrays. Monterey, USA: ACM, 2005: 51-59.
- [38] YUAN J Q, ZHOU X G, XIA Y S, et al. RBSA: range-based simulated annealing for FPGA placement[C]//2017 International Conference on Field Programmable Technology(ICFPT). Australia: IEEE, 2017: 1-8.
- [39] YUAN J Q, CHEN J L, WANG L L, et al. ARBSA: adaptive range-based simulated annealing for FPGA placement[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2018, 38(12): 2330-2342.
- [40] AI-HYARI A, SHAMLI A, MARTIN T, et al. An adaptive analytic FPGA placement framework based on deep learning[C]//2020 ACM/IEEE 2nd Workshop on Machine Learning for CAD(MLCAD). Iceland: IEEE, 2020: 3-8.
- [41] MURRAY K E, BETZ V. Adaptive FPGA placement optimization via reinforcement learning[C]//2019 ACM/IEEE 1st Workshop on Machine Learning for CAD(MLCAD). Canada: IEEE, 2019: 1-6.
- [42] ELGAMMAL M A, MURRAY K E, BETZ V. Learn to place: FPGA placement using reinforcement learning and directed moves[C]//2020 IEEE International Conference on Field-Programmable Technology. USA: IEEE, 2020: 1-6.
- [43] XU Y H, KHALID M. QPF: efficient quadratic placement for FPGAs[C]//International Conference on Field Programmable Logic and Applications. Finland: IEEE, 2005: 555-558.
- [44] GOPALAKRISHNAN P, LI X, PILEGGI L. Architecture-aware FPGA placement using metric embedding[C]//2006 43rd ACM/IEEE Design Automation Conference. USA: IEEE, 2006: 460-465.
- [45] GORT M, ANDERSON J H. Analytical placement for heterogeneous FPGAs[C]//22nd International Conference on Field Programmable Logic and Applications(FPL). Norway: IEEE, 2012: 143-150.
- [46] GESSLER F, BRISK P, STOJILOVIC M. A shared-memory parallel implementation of the RePlace global cell placer[C]//2020 33rd International Conference on VLSI Design and 2020 19th International Conference on Embedded Systems(VLSID). India: IEEE, 2020: 78-83.
- [47] LIN T H, BANERJEE P, CHANG Y W. An efficient and effective analytical placer for FPGAs[C]//2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC). USA: IEEE, 2014: 1-6.
- [48] CHEN Y C, CHEN S Y, CHANG Y W. Efficient and effective packing and analytical placement for large-scale heterogeneous FPGAs[C]//2014 IEEE/ACM International Conference on Computer-Aided Design(ICCAD). USA: IEEE, 2014: 647-654.
- [49] ABUOWAIMER Z, MAAROUF D, MARTIN T, et al. GPlace3.0: routability-driven analytic placer for Ultra-Scale FPGA architectures[J]. ACM Transactions on Design Automation of Electronic Systems, 2018, 23(5): 1-33.
- [50] LI W X, DHAR S, PAN D Z. UTPlaceF: A routability-driven FPGA placer with physical and congestion aware packing[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2018, 37(4): 869-882.
- [51] LI W X, PAN D Z. A new paradigm for FPGA placement without explicit packing[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2019, 38(11): 2113-2126.
- [52] LI W X, LIN Y B, PAN D Z. elfPlace: Electrostatics-based placement for large-scale heterogeneous FPGAs[C]//2019 IEEE/ACM International Conference on Computer-Aided Design(ICCAD). USA: IEEE, 2019: 1-8.
- [53] VERCRUYCE D, VANSTEENKISTE E, STROOBANDT D. Liquid: High quality scalable placement for large heterogeneous FPGAs[C]//2017 International Conference on Field Programmable Technology(ICFPT).

- USA: IEEE, 2017: 17-24.
- [54] CHEN G J, PUI C W, CHOW W K, et al. RippleFPGA: Routability-driven simultaneous packing and placement for modern FPGAs[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2018, 37(10): 2022-2035.
- [55] 王德奎. FPGA 高效布线方法研究[D]. 西安: 西安电子科技大学, 2019.
- WANG D K. Research on Efficient FPGA Routing Algorithm[D]. Xi'an, China: Xidian University, 2019. (in Chinese)
- [56] MCMURCHIE L, EBELING C. PathFinder: A negotiation-based performance-driven router for FPGAs[C]//Third International ACM Symposium on Field-Programmable Gate Arrays. USA: IEEE, 1995: 111-117.
- [57] WANG D K, DUAN Z H, TIAN C, et al. A runtime optimization approach for FPGA routing[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2018, 37(8): 1706-1710.
- [58] LAVIN C M. Using Hard Macros to Accelerate FPGA Compilation for Xilinx FPGAs[D]. Provo: Brigham Young University, 2012.
- [59] LAVIN C, PADILLA M, LAMPRECHT J, et al. HMFlow: Accelerating FPGA compilation with hard macros for rapid prototyping[C]//2011 IEEE 19th Annual International Symposium on Field-Programmable Custom Computing Machines. USA: IEEE, 2011: 117-124.
- [60] COOLE J, STITT G. BPR: fast FPGA placement and routing using macroblocks[C]//Proceedings of the Eighth IEEE/ACM/IFIP International Conference on Hardware/Software Codesign And System Synthesis. USA: IEEE, 2012: 275-284.
- [61] GORT M, ANDERSON J H. Combined architecture/algorithm approach to fast FPGA routing[J]. IEEE Transactions on Very Large Scale Integration(VLSI) Systems, 2013, 21(6): 1067-1079.
- [62] FERREIRA R, ROCHA L, SANTOS A G, et al. A runtime FPGA placement and routing using low-complexity graph traversal[J]. ACM Transactions on Reconfigurable Technology and Systems, 2015, 8(2): 1-16.
- [63] NGUYEN A H N, AONO M, HARA-AZUMI Y. FPGA-based hardware/software co-design of a bio-inspired SAT solver[J]. IEEE Access, 2020, 8: 49053-49065.
- [64] MA S H, AMPADU P. Optimal SAT-based minimum adder synthesis of linear transformations[C]//2019 IEEE 62nd International Midwest Symposium on Circuits and Systems(MWSCAS). USA: IEEE, 2019: 335-338.
- [65] HE A P, YU L Y, ZHANG H T, et al. A FPGA based SAT solver with high random and concurrent strategies[C]//2018 IEEE International Conference on Software Quality, Reliability and Security Companion(QRS-C). Portugal: IEEE, 2018: 221-228.
- [66] VANSTEENKISTE E. New FPGA Design Tools and Architectures[D]. Ghent: Ghent University, 2016.
- [67] VERCRUYCE D, VANSTEENKISTE E, STROOBANDT D. CRoute: A fast high-quality timing-driven connection-based FPGA router[C]//2019 IEEE 27th Annual International Symposium on Field-Programmable Custom Computing Machines(FCCM). USA: IEEE, 2019: 53-60.
- [68] MURRAY K E, ZHONG S, BETZ V. AIR: A fast but lazy timing-driven FPGA router[C]//2020 25th Asia and South Pacific Design Automation Conference(ASP-DAC). China: IEEE, 2020: 338-344.
- [69] STOJILOVIC M. Parallel FPGA routing: Survey and challenges[C]//2017 27th International Conference on Field Programmable Logic and Applications(FPL). Belgium: IEEE, 2017: 1-8.
- [70] HOO C H, HA Y H, KUMAR A. ParaFRo: A hybrid parallel FPGA router using fine grained synchronization and partitioning[C]//2016 26th International Conference on Field Programmable Logic and Applications(FPL). Switzerland: IEEE, 2016: 1-11.
- [71] SHEN M H, LUO G J, XIAO N. Exploiting box expansion and grid partitioning for parallel FPGA routing[C]//2018 IEEE 26th Annual International Symposium on Field-Programmable Custom Computing Machines(FC-CM). USA: IEEE, 2018: 209-209.
- [72] KORILIJA D, STOJILOVIC M. FPGA-assisted deterministic routing for FPGAs[C]//2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). Brazil: IEEE, 2019: 155-162.
- [73] CHAN P K, SCHLAG M D F. Acceleration of an FPGA router[C]//The 5th Annual IEEE Symposium on Field-Programmable Custom Computing Machines Cat. USA: IEEE, 1997: 175-181.
- [74] CHAN P K, SCHLAG M D F, EBELING C, et al. Distributed-memory parallel routing for field-programmable gate arrays[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2000, 19(8): 850-862.

- [75] GORT M, ANDERSON J H. Accelerating FPGA routing through parallelization and engineering enhancements special section on PAR-CAD 2010[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2012, 31(1): 61-74.
- [76] SHEN M H, LUO G J. Accelerate FPGA routing with parallel recursive partitioning[C]//2015 IEEE/ACM International Conference on Computer-Aided Design(ICCAD). USA: IEEE, 2015: 118-125.
- [77] SHEN M H, LUO G J, XIAO N. Coarse-grained parallel routing with recursive partitioning for FPGAs[J]. IEEE Transactions on Parallel and Distributed Systems, 2020, 32(4): 884-899.
- [78] WANG D K, DUAN Z H, TIAN C, et al. ParRA: A shared memory parallel FPGA router using hybrid partitioning approach[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2020, 39(4): 830-842.
- [79] GORT M, ANDERSON J H. Deterministic multi-core parallel routing for FPGAs[C]//2010 International Conference on Field-Programmable Technology. China: IEEE, 2010: 78-86.
- [80] SHEN M H, XIAO N. Fine-grained parallel routing for FPGAs with selective expansion[C]//2018 IEEE 36th International Conference on Computer Design(ICCD). USA: IEEE, 2018: 577-586.
- [81] LIU L, KAPRE N. Timing-aware routing in the Rapid-Wright framework[C]//2019 29th International Conference on Field Programmable Logic and Applications (FPL). Spain: IEEE, 2019: 24-30.
- [82] SHEN M H, LUO G J, XIAO N. Exploring GPU-accelerated routing for FPGAs[J]. IEEE Transactions on Parallel and Distributed Systems, 2018, 30(6): 1331-1345.
- [83] MOCTAR Y O M, BRISK P. Parallel FPGA routing based on the operator formulation[C]//2014 51st ACM/EDAC/IEEE Design Automation Conference(DAC) [C]. USA: IEEE, 2014. 1-6.
- [84] MOCTAR Y, STOJILOVIC M, BRISK P. Deterministic parallel routing for FPGAs based on Galois parallel execution model[C]//2018 28th International Conference on Field Programmable Logic and Applications(FPL). Ireland: IEEE, 2018: 21-25.
- [85] ZHU C, WANG J, LAI J M. A novel net-partition-based multithread FPGA routing method[C]//2013 23rd International Conference on Field programmable Logic and Applications. Portugal: IEEE, 2013: 1-4.

作者简介



田春生 男,1993年生于吉林梅河口.现为北京大学信息科学技术学院与北京微电子技术研究所联合博士后.主要研究方向为集成电路自动化设计.
E-mail: tiancs@pku.edu.cn



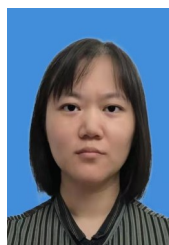
陈雷(通讯作者) 男,1978年生于安徽淮南.现为北京微电子技术研究所研究员,硕士生导师.主要研究方向为超大规模集成电路设计.
E-mail: chenleinpu@vip.126.com



王源 男,1979年生于陕西扶风.现为北京大学教授,博士生导师.主要研究方向为大规模集成电路设计.
E-mail: wangyuan@pku.edu.cn



王硕 男,1985年生于河南南阳.现为北京微电子技术研究所高级工程师.主要研究方向为FPGA布局布线算法和FPGA软错误缓解技术.
E-mail: 66188893@qq.com



周婧 女,1986年生于山东烟台,现为北京微电子技术研究所高级工程师,主要研究方向为FPGA CAD工具研发.
E-mail: zhoujenny0915@126.com



张瑶伟 男,1997年生于山西运城.现为中国运载火箭技术研究院电子科学与技术硕士研究生.主要研究方向为FPGA的三模冗余、高层次综合技术.
E-mail: zyw18810532787@163.com