

引用格式:行华彧,景乃锋. 一种基于多阶段模拟退火的异构可重构阵列布局算法[J]. 微电子学与计算机,2020,37(6):1-5.
[XING H Y, JING N F. A placement algorithm for HGRA based on multi-stage simulated anneal[J]. Microelectronics & Computer, 2020, 37(6): 1-5.]

一种基于多阶段模拟退火的异构可重构阵列布局算法

行华彧,景乃锋

(上海交通大学 电子信息与电气工程学院, 上海 200240)

摘 要: 异构可重构阵列是传统粗粒度可重构阵列的扩展,通过多样的异构单元可以实现更加丰富的应用类型.但是由于阵列异构单元在阵列中的位置、资源多种多样,使得传统的模拟退火布局方法在异构结构下会出现布通率下降甚至难以布通的问题.本文通过构建基于图距离的新型代价函数,来解决传统代价函数对异构阵列建模不精确的问题;进而根据不同异构单元的特性,提出分阶段模拟退火的布局方案.与传统方案应用在异构阵列上的结果相比,应用新型代价函数的多阶段模拟退火算法在同等互连资源的条件下可以平均提高 29.6% 的布通率.

关键词: 异构可重构阵列;模拟退火;图距离;布局布线

中图分类号: TP303

文献标识码: A

文章编号: 1000—7180(2020)06—0001—05

A placement algorithm for HGRA based on multi-stage simulated anneal

XING Hua-yu, JING Nai-feng

(School of Electronic Information and Electrical Engineering,
Shanghai Jiao Tong University, Shanghai 200240, China)

Abstract: The heterogeneous reconfigurable array (HGRA) is a kind of expansion of traditional architecture. Traditional mapping algorithms such as simulated anneal are not suitable for new structure. This paper proposes a new multi-stage simulated anneal placement algorithm, who has a new type of cost function based on graph distance. By using new algorithm, the success rate on HGRA increases by 29.6% on average.

Key words: heterogeneous reconfigurable array; simulated anneal; graph distance; placement and routing

1 引言

可重构计算是一种灵活性与专用性的折中方案,既可以保持相对于专用集成电路(ASIC)的较高性能,又可以实现类似 FPGA 一样的高自由度.在可重构阵列的工作流程中,将应用图(Data Flow Graph, DFG)映射在阵列上是至关重要的步骤,并且这个步骤被证明是一个 NP 难^[1]的问题.

对于传统的同构可重构阵列,DFG 图的映射已经有了较为成熟的解决方案,主要方法为以模拟退火为核心的布局过程与布通率驱动的布线过程.但

是随着发展,有较多新型异构阵列^[2-4]被提出.在异构阵列中存在不同类型计算资源的位置约束,传统的映射方案中没有对这类异构问题进行建模,因此会出现布通率下降甚至无法布通的问题.

为此,本文针对上述问题提出基于图距离的分阶段模拟退火方法,利用图距离来构建异构资源的位置约束,利用分阶段模拟退火的方法来分离异构资源在同一次模拟退火中的相互影响,从而大大提高异构阵列在极限互连资源情况下的布通率.也可以在同资源的阵列下布通更大规模的 DFG 图.

2 异构阵列的执行模式

传统的同构可重构阵列具备两种可重构的特性,分别是处理单元(Processing Element, PE)可重构和互连资源可重构,这两者共同支持阵列完成不同的应用.异构阵列在可重构性上依然是上述两者,但是在细节上与同构阵列有很大不同.以本文采用的异构阵列为例,如图 1 所示,与同构的区别主要体现在两个方面:

一是处理单元不再单一,会有多种不同类型的处理单元分散在阵列的不同位置,他们具备不同的计算能力,以扩展阵列可以执行的应用程序的范围.在图 1 中,主体部分为粗粒度处理单元 PE,执行主要的计算;外围有存储处理单元(Load Store Element, LS),负责管理访存需求;其次是循环控制单元(Loop Control, LC),主要负责循环头尾控制与寄存循环变量.从映射的角度来说,DFG 图中特定功能的点必须被映射到可以执行这种功能的位置上去,例如执行 LS 功能的点必须被映射在阵列的最外围.这就意味着布局的好坏会极大的影响布线.在 DFG 图中如果有大量异构功能之间数据的交互,那么传统的模拟退火方法产生的布局会让阵列中出现大量跨度极大的互连需求,布线资源紧缺的情况下很容易出现无法布通的情况.

二是互连资源不再同构,有些类型的计算单元周围会有较为丰富的互连资源,而有些周围较为稀少.如何设计不同类型资源周围的互连能力会极大的影响布线阶段的压力.互连资源提供的过多会产生大规模的浪费,提供的过少会大幅度降低布通率.

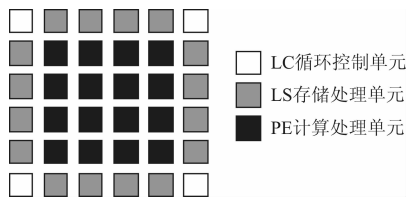


图 1 典型异构可重构阵列

本文将以图 1 所示的异构阵列为基础,通过改进模拟退火算法来改进上述第一个问题,同时在一些典型的互连模式下进行探索,寻找针对上述第二个问题较为合适的解决方案.

此外,可重构阵列在执行模式上通常有两种设计方式,区别取决于每一个执行单元内部是否可以执行指令,也即在同一个应用程序期间的同一个执

行单元,是否可以每个周期切换指令来执行不同的功能.包含指令的设计方式通常称为基于时间的执行模式(Time-based Mapping),反之称为基于空间的执行模式(Spatial-based Mapping).由于基于时间的执行模式允许单元时分复用,在布局布线的问题上压力较小,因此本文阵列采用的为基于空间的执行模式,要求 DFG 图在整个阵列上完全被映射,不允许计算单元的时分复用.

3 分阶段模拟退火的布局方案设计

在模拟退火布局算法中,代价函数是及其重要的一环,它直接影响了布局质量与布通率.代价函数最关键的部分在于正确的对布线难易程度进行建模.在经典 FPGA 布局布线工具 VPR 中,采用线性拥挤函数^[5]来评估布线质量,优势是针对 FPGA 建模准确,并且函数计算简单,在同构阵列下,线性拥挤函数有很好的效果,但是在异构阵列下不能精准的反应位置约束.在文献[6]中,每次布局的交换之后,都进行全局布线,根据全局布线质量计算准确的代价函数,这样的优势是没有了估计布线质量的问题,因为每次交换都进行了全局布线,缺点是大大增加了算法的 CPU 时间.

本文采用另一种方式,对代价函数与模拟退火进行改进,采用分阶段的模拟退火方案和两种代价函数,以针对不同的异构资源.在 PE 的部分,采用 VPR 中的线性拥挤函数来建模.这是因为 PE 部分少有约束,与传统的粗粒度可重构阵列没有太大差异.而在 LS 与 LC 的两种异构资源上,采取棋盘距离与图距离结合的代价函数.

3.1 基于图距离的代价函数

代价函数需要对当前布局结构的可布线性进行建模.在线性拥挤的计算方式中,算法遍历所有 DFG 图中的边,计算源点与终点在阵列中的棋盘距离.但是在分阶段模拟退火下,上述方案不再适用.因为在分阶段的模拟退火中,前一次的退火过程中缺少后一次退火的点的位置信息,不能在 DFG 图中逐边遍历.因此引入图距离,来对缺失的信息进行补足.

为了计算这种代价函数,需要进行图分析,来获得图距离.给定 DFG 图,图距离被定义为将 DFG 图无向化之后,两点间的最短路径的边数,如图 2 所示.

设 DFG 中需要计算的子图包含 N 个节点,用 CD_{ij} 表示两点之间的棋盘距离, GD_{ij} 表示两点之间

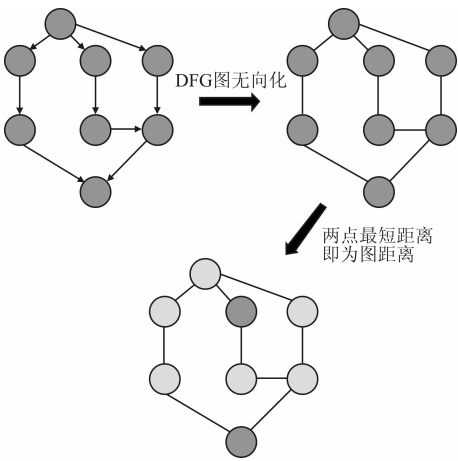


图 2 图距离的计算方法

的图距离,则子图的代价函数为:

$$Cost = \sum_{i=0}^N \sum_{j=i}^N CD_{ij} * GD_{ij}^k \# \tag{1}$$

式中, k 为权重因子,用来调整图距离在整个代价函数中的作用比例. 当 k 为 0 的时候,图距离不起作用,代价函数退化为普通的线性拥挤函数. 当 k 大于 0 的时候,图距离参与代价函数的运算.

图距离本身代表了两个点在图中潜在的连通距离. 在布局的过程中,希望具有更短连通距离的点在阵列上离得更近.

权重因子 k 代表希望补足的图距离在模拟退火的过程中起多大的主导作用,这与每一个应用程序本身息息相关,当应用本身的 DFG 图中有大量的异构资源之间的交互时,提高 k 值会有更好的布局效果. 根据实验,需要图距离参与时,通常 k 取值为 2.

3.2 分阶段模拟退火

本文采用的阵列结构包含三种异构资源,因此模拟退火分为三个阶段,LC 退火阶段,LS 退火阶段,PE 退火阶段. 在每一个退火阶段,都仅布局 DFG 图中对应类型的节点. 下一个阶段的布局会基于上一个阶段布局的结果.

有三个参数可以控制每个阶段模拟退火的进程,分别是每个温度点移动数目参数 InnerNumber,退火条件参数 ϵ ,以及影响接受概率 γ 的退火表.

InnerNumber 影响每个温度点进行交换与移动的次数,本文采取文献[7]的公式:

$$SwapNum = InnerNum (N_{node})^{\frac{4}{3}} \# \tag{2}$$

式中, N_{node} 为本次退火的总节点数. 通常 InnerNumber 取 10,增加 InnerNumber 可以增加每个温度点下的交换次数,有利于跳出局部解,但是会增加 CPU 时间. 此外这个值不宜过大,过大会导致

每个温度迭代的布局变化都过于随机,很难收敛.
退火条件参数 ϵ 影响退火的中止条件,值越小,退火的进程越长. 中止条件为:

$$Temperature < \epsilon \frac{Cost}{N_{nets}} \# \tag{3}$$

通常 ϵ 取 0.1 ~ 0.001,退火质量不会对 ϵ 非常敏感.
退火表为模拟退火的温度更新策略,在传统的模拟退火中,温度更新策略仿照真实的热力学条件呈负指数下降. 但是根据文献[7] 的研究,对于布局的模拟退火,采用一个分阶段的线性函数 $\gamma(\alpha)$ 已经足够. 本文也采用类似的温度更新策略,退火表如表 1 所示.

表 1 退火温度表

α	γ
$\alpha > 0.96$	0.5
$0.8 < \alpha \leq 0.96$	0.9
$0.15 < \alpha \leq 0.8$	0.95
$\alpha \leq 0.15$	0.8

在执行模拟退火时,三个阶段依次为 LS 退火, LC 退火,最后是 PE 退火. 选择优先退火的原则是根据异构资源对阵列的约束. 在本文采用的结构下,根据针对典型应用的 DFG 图分析,LS 访存单元对阵列约束最大,原因有二,一是根据 DFG 不同,会有不同数量的 PE 频繁连接到 LS,有大量访存需求的 DFG 中会有大量的 PE 连接到 LS;二是在阵列中 LS 的位置受限,只能分布在最外面的一圈,当布局效果不理想时,布线会非常困难. 因此首先对 LS 进行模拟退火,选定代价函数中图距离权重因子 k 为 2 或者 3,这样可以使模拟退火更倾向于让 DFG 图中相近的 LS 在布局中更靠近,潜在的减少布线过程中路由的产生频率. 根据同样的思想,接下来进行 LC 退火,根据应用分析,LC 在一个 DFG 图中出现不会很多,通常为 1 到 2 个,不会对布局产生过大的约束. 最后执行 PE 布局,在 LS,LC 固定之后,单纯的 PE 的模拟退火与传统的粗粒度可重构阵列的模拟退火并无二异.

在代价函数的选择上,前两个阶段为针对异构资源的退火过程,因此采用基于图距离的代价函数 (1),在最后一个 PE 退火阶段,则标准的线性拥挤函数,理由有二,一方面 PE 阶段作为最后一个退火阶段,已经有了所有点的布局信息,不需要计算图距离来辅助退火;另一方面由于采用图距离的代价函数会带来更大的计算量,PE 为数量最多的资源类型,采用图距离的函数会大幅度降低退火速度.

4 实验结果与分析

4.1 互连方案与数据路由方案

文献[8]给出了几种典型的互连结构,如图 3 所示,其中 way4 为每一个 PE 都与上下左右四个 PE 可以存在数据交互. 其余同理,互连的丰富度逐渐提高.

本文在 PE 部分会分别利用上述四种互连方式进行探索. 除此之外,每一个 PE 还可以与自己同行同列的 LS 进行数据交互,增加这部分互连是为了减轻 PE 与 LS 之间大规模数据交互的互连压力.

这样的互连结构依然不能解决长距离的互连需求. 对于长距离的,不存在既有互连的数据交互,传统的方法是,PE 可以仅转发数据,而不执行功能. 这样一来,远距离的数据可以通过中间的一些 PE 转发来进行交互. 但是这种操作模式会导致 PE 的大量浪费. 因为 PE 如果用来转发数据,就不再执行计算功能,ALU 没有工作;并且在空间的映射中,数

据转发是大量且普遍存在的,需要大量 PE 来进行转发工作.

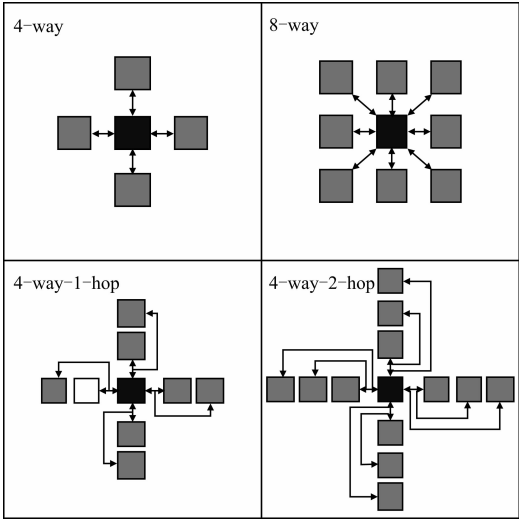


图 3 典型互连方案

表 2 在不同互连模式下两种退火方案布通率对比

应用	DFG 节点数	Way4		Way8		Way4Hop1		Way4Hop2	
		一次退火	阶段退火	一次退火	阶段退火	一次退火	阶段退火	一次退火	阶段退火
脉动矩阵乘	23	1	1	1	1	1	1	1	1
ADPCM 编码	42	0	0.31	1	1	0.17	1	1	1
NW 序列检测	42	0.17	0.42	1	1	0.43	1	1	1
BFS	70	0	0	0	0	0.07	0.08	0.83	0.71
边缘检测	73	0	0	0	0	0.62	0.83	1	1
稀疏矩阵乘	84	0	0	0	0	0	0	1	1

表 3 在不同互连模式下两种退火方案所需时间对比(秒)

应用	DFG 节点数	Way4		Way8		Way4Hop1		Way4Hop2		平均时间增长(%)
		一次退火	阶段退火	一次退火	阶段退火	一次退火	阶段退火	一次退火	阶段退火	
脉动矩阵乘	23	4.731	7.997	4.618	8.210	4.753	7.861	4.712	8.002	70.5
ADPCM 编码	42	N/A	24.095	14.486	23.088	14.258	21.871	15.021	21.909	52.9
NW 序列检测	42	14.377	23.911	14.144	22.558	14.651	22.889	14.322	22.391	59.5
BFS	70	N/A	N/A	N/A	N/A	42.224	54.397	35.827	54.831	40.9
边缘检测	73	N/A	N/A	N/A	N/A	36.710	62.658	32.635	58.793	75.4
稀疏矩阵乘	84	N/A	N/A	N/A	N/A	N/A	N/A	51.010	80.228	57.3

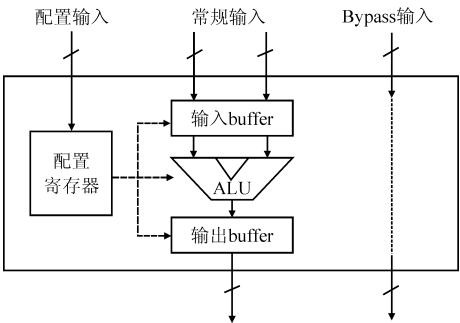


图 4 不影响 PE 功能的 Bypass 结构

因此本文采取另一种方案,通过增加端口来实现

数据的路由功能,例如 PE 的结构如图 4 所示,右侧的路由通道不影响 PE 本身内部功能的执行,PE 在通过路由端口进行数据转发的同时,依然可以通过正常端口执行 ALU 操作. 采用这种方式相当于每一个单元都增加了一组端口,即增加了互连资源量与阵列的配置量. 由于数据转发是大量存在的,因此通过资源量来换取阵列单元的高利用率是值得的.

4.2 结果与分析

MachSuite^[9]是一个针对通用计算的应用程序样本集,并且按照经典的 Dwarf 分类^[10]对应用程序进行了划分,具有一定代表性. 因此本文从 Mach-

Suite 中挑选六个适合阵列加速的应用进行测试,从布通率与算法时间两个角度对算法与互连质量进行分析。

本文采用的阵列结构规模为 10×10 ,对每一种算法与互连结构的组合,都重复执行算法 50 次,记录其中成功完成布局布线的次数,占比作为布通率。从表 2、表 3 可以看出,采用分阶段模拟退火的方法可以在极限情况下以消耗时间为代价,提高布通的可能性。根据不同应用的不同特点,分阶段模拟退火对布通率的影响也不同。主要的影响因素是异构资源直接产生交互的多少,有越多的异构资源交互,分阶段模拟退火就有更好的结果。除去两种算法都完全布通与完全布不通的极端情况,分阶段模拟退火相较传统模拟退火平均提高 29.6% 的布通率。

就互连资源角度来说,way4hop2 模式的互连属于非常丰富的互连,即使是稀疏矩阵乘这样的大规模高压力的应用,都可以在任何算法下完成轻松的布局。但是相对的,way4hop2 引入的互连资源冗余较大,在本文的阵列中,way4hop2 引入的配置量比 way4hop1 高出 25%。综合来看,way4hop1 是资源量与布通率折中下较好的互连方案,实际上对于稀疏矩阵乘以及更大规模的应用,实际配置中更倾向于拆分到多个阵列上协同完成而不是在单阵列上完成,因此并不需要追求将如此高压力的 DFG 图在单个阵列上布通。

由于分阶段模拟退火本身是多次较小规模的模拟退火,因此时间上相较传统的模拟退火会有增长,根据应用的特点,增加的时间不定,根据表 3 可以看出,平均情况下分阶段模拟退火会比传统模拟退火方案增加 40% 到 70% 的 CPU 时间。

5 结束语

针对异构资源可重构阵列,本文提出基于图距离的新型代价函数与分阶段模拟退火的布局方法,来解决异构资源带来的互连挑战。并且针对不同模式的互连方案进行了测试。对于同互连结构,新算法相比传统布局算法可以以 40% 到 70% 的 CPU 时间为代价,平均提高 29.6% 的布通率;对于同算法,根据本文的探索,way4hop1 形式是考量资源量与布通率后综合表现较好的互连方案。

参考文献:

[1] YOON J W, SHRIVASTAVA A, PARK S, et al.

A Graph Drawing Based Spatial Mapping Algorithm for Coarse-Grained Reconfigurable Architectures[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2009, 17(11):1565-1578.

[2] 俞磊,景乃锋,王琴.一种基于数据流驱动的混合粒度可重构阵列架构[J].微电子学与计算机,2019,36(5):19-22.

[3] WAINGOLD E, TAYLOR M, SRIKRISHNA D, et al. Baring it all to software: Raw machines[J]. Computer, 1997, 30(9):86-93.

[4] NIKTASH A, PARIZI H T, BAGHERZADEH N. Application of a Heterogeneous Reconfigurable Architecture to OFDM Wireless Systems[C]//IEEE International Symposium on Circuits & Systems. New Orleans, USA: IEEE, 2007: 2586-2589.

[5] BETZ V, ROSE J. VPR: a new packing, placement and routing tool for FPGA research[C]//International Workshop on Field Programmable Logic and Applications. Berlin, Germany: Springer, 1997: 213-222.

[6] SISTLA A K, PARDE N, PATEL K, et al. Cross-Architectural Study of Custom Reconfigurable Devices Using Crowdsourcing[C]//Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2013 IEEE 27th International. Cambridge, MA: IEEE, 2013: 222-230.

[7] BETZ V, ROSE J, MARQUARDT A. Architecture and CAD for deep-submicron FPGAs[J]. Springer International, 1999, 497: 51-55.

[8] MEHTA G, PATEL K K, PARDE N, et al. Data-Driven Mapping Using Local Patterns[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2013, 32(11):1668-1681.

[9] REAGEN B, ADOLF R, SHAO Y S, et al. Mach-Suite: Benchmarks for accelerator design and customized architectures[C]//2014 IEEE International Symposium on Workload Characterization (IISWC). Raleigh, NC: IEEE, 2014:110-119.

[10] ASANOVIC K, BODIK R, CATANZARO B C, et al. The Landscape of Parallel Computing Research: A View from Berkeley[J]. Technical Report Uc Berkeley, 2006, eecs-2006-183: 8-10.

作者简介:

行华彧 男,(1995-),硕士研究生,研究方向为计算机体系结构与可重构计算。Email: hsinghy@sjtu.edu.cn.

景乃锋 男,(1982-),博士,副研究员,研究方向为计算机体系结构。