

VLSI 布局算法分析

张家齐¹ 蔡文涛² 沈剑良¹ 朱珂¹

(1. 国家数字交换系统工程技术研究中心, 河南 郑州 450002; 2. 河南商务学校, 河南 郑州 450001)

摘要: 布局是 VLSI 物理设计阶段的关键步骤, 用于确定模块在芯片上的位置。随着电路设计复杂度不断提高, 高效的自动布局算法变得愈发重要。对布局问题进行了描述, 总结了布局算法中常用的线长模型, 分析了传统的 3 类布局算法及可布线性驱动的布局算法, 并给出了布局算法的未来发展趋势。

关键词: 超大规模集成电路; 布局算法; 线长驱动布局; 可布线性驱动布局

中图分类号: TN402

文献标识码: A

文章编号: 1671-0673(2016)05-0597-08

Analysis of VLSI Placement Algorithms

ZHANG Jiaqi¹, CAI Wentao², SHEN Jianliang¹, ZHU Ke¹

(1. National Digital Switching System Engineering and Technological Research Center, Zhengzhou 450002, China;

2. Henan Business Vocational School, Zhengzhou 450001, China)

Abstract: Placement is a key step in VLSI physical design flow and it determines the positions of circuit blocks. As the number of circuit elements and interconnections is still growing, automatic placement algorithms are in great need in the design flow. This paper first describes the placement problem. It then summarizes common wirelength models in placement algorithms, followed by analysis of three kinds of traditional placement algorithms and routability-driven placement algorithms. Last it points out the developing trends in the future.

Key words: VLSI; placement algorithms; wirelength-driven placement; routability-driven placement

集成电路芯片的发展基本遵循了摩尔定律, 随着电子芯片集成度的提高, 芯片内部晶体管数目越来越多, 电路密度以指数级增长, 电路设计的复杂度越来越高, 电子系统设计自动化 (electronic design automation, EDA) 技术就变得愈发重要。其中, 布局是电路设计的重要环节。合理的布局布线, 可以减小总的布线长度, 降低芯片成本, 还可以缩减芯片面积, 增加一个晶圆上所能集成的芯片个数, 提高芯片性能及成品率。

近 50 年来, 学术界和工业界针对超大规模集成电路 (very large scale integration, VLSI) 布局的各个方面进行了广泛的研究。布局问题的目标和

约束也经历了多次修改, 并仍在继续调整。另外, 随着布局实例规模的不断增大, 对高效布局工具的选择也产生了较大影响。针对布局问题, 借助计算机辅助技术, 布局算法的提出使得布局问题能够得到高效的解决。传统的布局算法可大体分为划分式、启发式和解析式, 为了确保电路的可布通性, 又有了可布线性驱动的布局算法。

本文对布局问题进行描述, 介绍了布局算法中常用的线长模型; 重点分析了 3 类传统的布局算法和可布线性驱动的布局; 讨论了布局算法未来的研究方向。

收稿日期: 2015-09-23; 修回日期: 2015-11-27

基金项目: 国家自然科学基金资助项目 (61309020); 国家 863 计划资助项目 (2014AA01A704)

作者简介: 张家齐 (1993 -), 男, 硕士生, 主要研究方向为数字集成电路设计与分析, E-mail: zhangjqfriend@126.com。

1 布局问题描述

布局就是在一定的约束条件下把电路模块放置在芯片合适的位置,电路的布局对芯片面积、可布性、功耗、时序等有直接影响,它的性能很大程度上决定了后续过程的设计质量。

集成电路布局问题描述^[1]: 设 B_1, B_2, \dots, B_n 是需要放置在芯片上的模块, 每个模块的高和宽分别为 H_i 和 W_i 。 $N = \{N_1, N_2, \dots, N_m\}$ 表示连接不同线网的集合, $Q = \{Q_1, Q_2, \dots, Q_k\}$ 表示模块间用于布线的空区域, L_i 表示线网 N_i 的估计线长。为每个模块寻求一个矩形区域 R_i 使其满足下列条件: 每个模块 B_i 都能放置于相应的区域 R_i 中; 任意两个矩形不重叠, 即 $R_i \cap R_j = \emptyset (i \neq j)$; 布局是可布的, 即 Q_j 足够用于布线; 所有 R_i 和 Q_j 的面积之和最小; 总线长 $\sum L_i$ 最小。

由于现代电子芯片中越来越多宏模块 (macro block) 的存在, 除了标准单元的布局以外, 有时还必须考虑这些面积较大的宏模块的布局, 这类布局问题被称为混合结构 (mixed-size) 的布局。

2 线长模型

总线长作为衡量布局结果的关键指标, 往往是布局算法的首要优化目标。由于布局时还没有真正进行布线, 首先需要一些模型来估计线长。线长模型为布局提供总线长的准确估计, 很大程度上决定了代价函数的建立, 同时也是详细布线前对于时延、可布线性等评估的基础。常见的线长估计模型如图 1 所示, 其中对总线长的估计通常采用边界框半周长模型或最小斯坦纳树模型, 而团模型、星型模型以及 B2B 模型则广泛应用于解析式算法中。

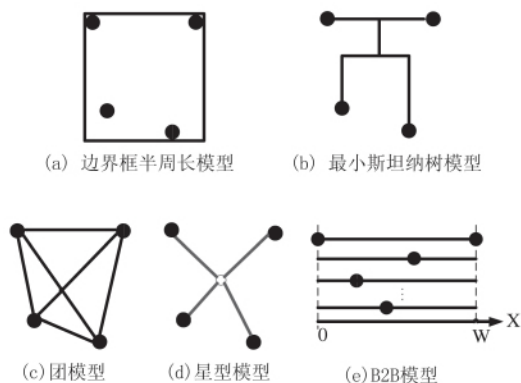


图 1 常见的线长估计模型

2.1 边界框半周长模型 (HPWL)

HPWL^[2] 用覆盖一个线网所有模块的最小矩形周长的一半作为线长的估计值, 如图 1 (a) 所示。线网 e 的 HPWL 计算如 (1) 式, 继而能计算出总线长 TotalHPWL 如 (2) 式。

$$HPWL(e) = HPWL_x(e) + HPWL_y(e) = \max_{i,j \in e} |x_i - x_j| + \max_{i,j \in e} |y_i - y_j| \quad (1)$$

$$TotalHPWL = \sum_{e \in E} HPWL(e) = \sum_{e \in E} HPWL_x(e) + \sum_{e \in E} HPWL_y(e) \quad (2)$$

对于一个两端或 3 端的线网, HPWL 的估计结果是较为准确的。但对于多端线网, 由于没能考虑到线网中所有单元的连接导致其估计值偏小, 误差会相对较大。目前 HPWL 因其计算的简便性仍是线长估计的首选模型。

2.2 最小斯坦纳树模型 (MST)

MST 通过构造连接线网内所有模块的最小斯坦纳树进行线长估计, 如图 1 (b) 所示。斯坦纳树模型可实现较高的估计精度, 适用于较大规模的线网。然而, 由于需要进行斯坦纳树的构造, MST 关于引脚数 n 的时间复杂度为 $O(n \log n)$, 相对较高, 相比之下 HPWL 的时间复杂度则是线性的。

2.3 团模型 (clique) 与星型模型 (star)

对于解析式算法来说, 目标函数必须是可微分的, 所以解析式布局器大多采用加权欧几里得距离的平方和进行线长估计:

$$\sum_{e \in E} w_e \times l^2 \quad (3)$$

其中 w 表示权重, l 表示模型中线网连线长度。

团模型^[3] 用一个 n 个顶点 $n(n-1)/2$ 条边的完全图来表示 n 引脚线网的连接, 如图 1 (c) 所示。实际应用中, 团模型仅适用于小规模线网, 对于较大规模的线网而言其计算复杂度过高。相比而言, 星型模型^[4] 更适用于大规模线网。星型模型首先计算出线网的几何中心, 用引脚到中心连线的距离表示线网连接, 如图 1 (d) 所示。通过适当设置线网权重, 团模型和星型模型在二次布局中是等效的, 并基于二者等效性, 提出了混合模型: 2 端或 3 端的线网用团模型; 4 端以上的线网用星型模型。混合模型能提高二次布局的效率。

2.4 B2B 模型 (bound2bound)

在二次解析式布局算法中, 对于较大规模的线网可应用 B2B 模型^[5] 进行拆分。B2B 模型不考虑所有线网内部引脚的连接, 仅考虑与最边界上引脚的连接, 由于二次目标函数中 x, y 的独立性, 先以 x 方向为例说明, 如图 1 (e) 所示。这样, 两个引脚

间的所有连接可能只有3种情况: 两端边界的连接; 左侧边界与内部引脚的连接; 右侧边界与内部引脚的连接。为每条连接定义权重:

$$w_{x_{pq}} = \begin{cases} 0, & \text{当 } p \text{ 和 } q \text{ 同是内部引脚时} \\ \frac{2}{N-1} \frac{1}{|x_p - x_q|}, & \text{其它} \end{cases} \quad (4)$$

将目标函数定义为

$$L_x = \frac{1}{2} \sum_{p=1}^N \sum_{q=p+1}^N w_{x_{pq}} (x_p - x_q)^2 = w \quad (5)$$

B2B 模型能够完全准确地替代 HPWL, 目标函数 L_x 和 L_y 与边界框半周长模型的长和宽是相等的, 能适用于规模较大的线网。其不足是只能根据确定的布局来估计, 当单元位置改变时连线长度和权值都需要跟着改变。

2.5 线长模型的应用

Gordian^[6] 采用团模型得到二次目标函数; Sim-PLR^[7] 中集成的 BFG-R^[8] 采用 MST 模型将多端线网分解为两端子线网, 基于这些子线网得到代价函数; Ripple^[9] 将最小化 HPWL 作为优化目标, 采用 B2B 模型来建立解析目标函数。这些线长模型各有优劣, 没有最佳的模型, 在布局中应根据线网及布局算法的特点等进行合理选择。目前, 在线长的估计上对简便易行的 HPWL 应用仍是最为广泛的, 而在解析式布局中则往往根据实际电路的特点选用团模型、星模型或 B2B 模型来建立目标函数。

3 传统的布局算法

布局问题出现了很多经典的算法, 大体分为划分式、解析式和启发式3种。20世纪60年代, 工业界首先出现了线网划分的思想, 并对后来的图划分启发式算法起了推动作用。20世纪80年代初开始了解析式布局器, 随着模拟退火算法的兴起, 基于模拟退火的布局器在工业界及学术界一度居于领先地位。20世纪90年代中期, 随着电路规模的扩大和复杂度的提高, 模拟退火算法已经越来越难以适应电路设计需求。这时以 Fiduccia-Mattheyses (FM) 算法^[10] 为代表的划分式算法重新引领了布局算法的主流。2005年, 一些解析式技术变得逐渐成熟, 在当时的大规模电路实例上取得了优于最小割算法的布局结果。加入对可布通性、物理综合、电源密度、大尺寸 I/O 模块、IP 核等的考虑, 核心布局区域经常包含大量的空白^[11], 这为解析式算法提供了发挥效用的空间。

布局算法的分类及当前领先的布局器如表1

所示。

表1 3种主要的布局算法及当前领先的布局器

启发式	Timberwolf ^[12]	
	VPR ^[13]	
划分式	Capo ^[14]	
	Dragon ^[15]	
解析式	FengShui ^[16]	
	非线性的	mPL ^[17]
		Aplace ^[18]
		NTUPace ^[19]
		Vaastu ^[20]
	基于划分	PROUD ^[21]
		Gordian ^[6]
		BonnPlace ^[22]
		hATP ^[23]
	二次的	RQL ^[24]
		FDP ^[25]
		FAR ^[26]
		mFAR ^[27]
		FastPlace ^[28]
		Kraftwerk ^[29]
		SimPL ^[30]
		WARP ^[31]
	基于变形	

3.1 划分式算法

基于划分的布局算法采用递归的思想, 同时对线网和布局区域进行迭代划分。对线网的划分通常基于一个代价函数, 使得穿过相邻子区域间的连线数最少, 如图2所示, 所以划分式算法有时也被称为最小割算法。常用的划分算法有 Kernighan-Lin (KL) 算法、FM 算法和 hMetis 算法等。

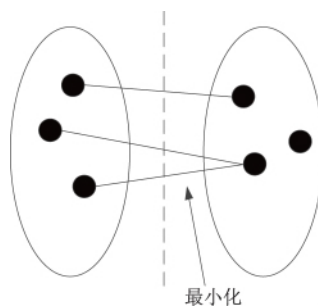


图2 线网的最小割划分

使用划分式算法的布局器有 Capo^[14]、Dragon^[15] 和 FengShui^[16] 等。基于划分的布局算法通常是高效并可扩展的, 能够适用于大规模电路, 然而, 由于基于最小割的算法并没有将线长作为目标, 最小化割线数并不等价于最小化线长, 所以其布局质量往往难以满足需求。

3.2 启发式算法

布局问题是一个最优化问题, 启发式的优化方法在解决布局问题中也能得到很好的应用。启发

式布局算法有模拟退火算法、遗传算法和蚁群算法等。其中模拟退火算法的应用最为广泛。

模拟退火算法(simulated annealing algorithm, SA)用固体退火过程模拟组合优化问题。在处理布局问题时,退火过程变为寻找代价函数最优解的过程。算法首先生成一个初始布局,然后随机地交换两个模块的位置,接受较优解,同时以一定概率接受较差解,迭代该过程直到温度降至最低点,这时代价函数便能趋于全局最优解。其中随着温度的降低接受较差解的概率也会越来越低。

布局布线工具 TimberWolf^[12]和 VPR^[13]都采用了模拟退火算法。模拟退火算法具有求解质量高和鲁棒性强等优点,但其退温过程会消耗大量时间。随着电路规模的不断扩大,已经越来越难以满足布局的要求。

3.3 解析式算法

解析式算法针对布局问题定义了一个代价函数,通过一系列的优化方法求该代价函数的最小值,所得结果便是单元的位置坐标。解析式算法的目标通常是使总的 HPWL 线长最小化,如(1)式所示。但是由于(1)式是非凸的,难以对其直接求解,所以便需要使用一些平滑的近似公式替代。当代价函数是凸的时,就可以采用共轭梯度法(conjugate gradient, CG)进行高效地求解。根据代价函数的不同,可将解析式算法进一步分为两种:二次的和非线性的。

3.3.1 二次解析算法

二次布局器将线长描述为一个二次的代价函数,可以通过求解线性方程组来快速得到最优解。二次型函数的一般形式如(6)式所示,根据所采取线长模型的不同(团模型或 B2B 模型等),其具体形式也会进行相应变化:

$$\sum_{e \in E} \left(\sum_{v_i, v_j \in e, i < j} w_{ij} (x_i - x_j)^2 + \sum_{v_i, v_j \in e, i < j} w_{ij} (y_i - y_j)^2 \right) \quad (6)$$

其中 x, y 分别表示单元的位置坐标, w 则表示权重,通常都取值为 0.5。

由于仅仅使线长最小化可能会导致相当大数目的模块重叠,二次布局器需要对重叠进行处理。根据处理方式的不同,可将二次布局器划分为基于划分的、力导向的和基于变形的 3 种:

①基于划分的有 PROUD^[21]、Gordian^[6]、BonPlace^[22]和 hATP^[23]等,它们对电路和芯片区域迭代地进行划分,在每个划分的层次求解二次代价函数的最小值来放置模块;

②力导向的二次布局器有 FAR, FastPlace, simPL, RQL 以及 Eisenmann 的 Kraftwerk 等,通过仿真物理力学的概念,根据单元间的“弹力”和“引力”来扩散芯片上的单元。力导向的方法能提供质量较好的布局结果和较低的运行时间;

③基于变形的二次布局器通过改变芯片区域的形状来间接移动模块,消除重叠。

3.3.2 非线性解析算法

非线性模型对总线长和重叠进行了约束,通常使用惩罚函数的方法进行求解。非线性的布局器通常基于一个非线性的代价函数,例如将线长表述为 L_p 标准函数、log-sum-exp(LSE)函数等。非线性布局器有 APlace, mPL, NTUPlace 和 Vaastu 等。

L_p 标准函数:

$$\sum_{e \in E} \left(\left(\sum_{v_k \in e} x_k^p \right)^{\frac{1}{p}} - \left(\sum_{v_k \in e} x_k^{-p} \right)^{-\frac{1}{p}} + \left(\sum_{v_k \in e} y_k^p \right)^{\frac{1}{p}} - \left(\sum_{v_k \in e} y_k^{-p} \right)^{-\frac{1}{p}} \right) \quad (7)$$

LSE 函数:

$$\gamma \sum_{e \in E} \left(\log \sum_{v_k \in e} \exp\left(\frac{x_k}{\gamma}\right) + \log \sum_{v_k \in e} \exp\left(\frac{-x_k}{\gamma}\right) + \log \sum_{v_k \in e} \exp\left(\frac{y_k}{\gamma}\right) + \log \sum_{v_k \in e} \exp\left(\frac{-y_k}{\gamma}\right) \right) \quad (8)$$

当 $p \rightarrow 0$ 和 $\gamma \rightarrow 0$ 时, (7) 式和 (8) 式均近似于:

$$\max_{v_k \in e} x_k - \min_{v_k \in e} x_k + \max_{v_k \in e} y_k - \min_{v_k \in e} y_k = HPWL(e) \quad (9)$$

在布局算法中,将 p 或 γ 取一个接近于 0 的很小的值,便可使用非线性函数作为 HPWL 的近似函数。

非线性的布局器可以获得布局信息的整体视图,具有良好的性能。但由于求解非线性问题通常耗时过长,非线性的方法不具有很好的扩展性。

3.4 传统布局算法的应用现状

早期的布局算法中,划分式算法得到了广泛应用。划分式算法具有高效的求解速度,适用于大规模电路。但由于其求解质量不够高,目前已经很少直接用于布局的求解,而是常被用于预处理过程和算法的中间改善步骤。布局中的多层级架构便是借助了划分式算法的思想。

启发式算法能够取得较高的求解质量,曾经成为布局算法的主流,但随着电路规模的日益增大,其求解速度慢的缺点也逐渐放大,已经越来越难以满足需求。解析式算法具有较高的求解速度和求解质量等优点。随着电路规模的增大,描述约束方程式越来越多,往往采用多层级架构的思想减小问题规模,再行求解。目前领先的布局器大多采用了解析式算法。

4 可布线性驱动布局

布局算法中大多使用 HPWL 估计线长,追求总线长的最小化。这样得到的布局结果可能使布线分布不均,造成难以布线甚至不可布线。针对这个问题产生了着重考虑电路可布性的布局算法。

拥挤度是衡量可布性的重要指标,需要对其进行合理的量化分析。文献[32]分析了拥挤度与线长之间的联系:总线长最小的布局同时也是平均拥挤度最小的,减少线长能降低芯片的整体布线需求;却容易出现局部拥挤,因为局部的布线密度可能增大,布线的具体分布情况是无法预知的,如图3所示。可布线驱动的布局首先需要建立拥挤度分析模型,然后对拥挤进行处理。

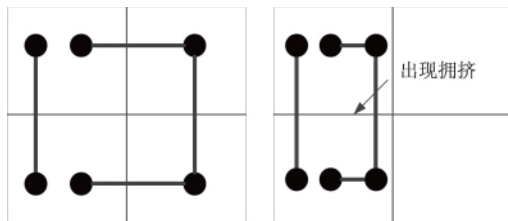


图3 优化线长与减少拥挤并不完全等价

4.1 拥挤度评价函数

目前对拥挤度的研究大都基于全局布线图的结构。在 DAC 2012 可布性驱动布局竞赛中,使用了全局布线图结构把整个版图分成一个个网格,把这些网格叫做全局布线单元(global routing grid cell, GCell),这些网格之间的交界称为全局布线图边界(global routing graph edge, GEdge),如图4所示。

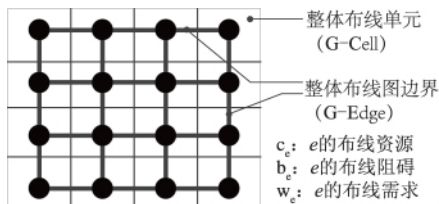


图4 整体布线图

对每个 GEdge 做如下定义: $OF(e) = \max(w_e + b_e - c_e, 0)$, 即当布线需求超过布线资源时发生溢出。其拥挤度 $Cong(e) = 100 \times (w_e + b_e) / c_e$ 。两个常用的拥挤度衡量指标是总溢出(total overflow, TOF) 以及最大溢出(maximal over, MOF):

$$TOF = \sum OF(e) \quad (10)$$

$$MOF = \max(OF(e)) \quad (11)$$

TOF 和 MOF 都是从整体角度来描述拥挤度,

无法体现局部拥挤度。故引入新的衡量指标:平均拥挤度 $ACE(x)$ (average congestion of gcell Edges), 代表拥挤度最高的前 $x\%$ GEdge 的平均拥挤度。令 x 取不同的值, 便可计算出一组 ACE 值, 反映出更精确的拥挤度。

最高权值拥挤度(peak weighted congestion, PWC):

$$PWC = \frac{\sum (K_x \times ACE(x))}{\sum K_x} \quad (12)$$

其中 $x=0.5, 1, 2, 5$, K_x 是对应 $ACE(x)$ 的权重。

布线拥挤度(routing congestion, RC):

$$RC = \max(100, PWC) \quad (13)$$

在 HPWL 和 RC 的基础上, 给出衡量布局结果可布线性的指标:

$$Routability_Metric = HPWL \times (1 + PF \times (RC - 100)) \quad (14)$$

其中 PF 是拥挤度的线长惩罚因子。例如, 取 $K_x = 1.0$, $PF = 3$, x 取任意值, 那么可布性指标便表示: 对每 1% 的额外拥挤度(超过 100% 的部分), 都会有 3% 的线长惩罚。

可布线性指标建立了拥挤度的准确评价标准, 并将拥挤度转换为线长惩罚, 这样可使用统一的指标同时衡量布局结果的线长及拥挤度。可布线驱动的布局要解决的关键问题便是对拥挤度的分析及消减。

4.2 拥挤度模型

为了对布局期间的拥挤度进行评估, 必须建立高效准确的拥挤度估计模型。当前的拥挤度估计模型主要可分为: 静态模型、概率模型和构造模型, 如表 2 所示。

表2 当前的拥挤度估计模型

分类	具体模型
静态模型	基于 Rent 规则 ^[33]
	边界框模型 ^[34]
	构造斯坦纳树 ^[35]
	引脚密度 ^[36]
概率模型	Bin 内线网计数 ^[37]
	概率图样布线 ^[38]
	伪构造线长 ^[39]
	RUDY(统一布线密度) ^[40]
构造模型	二维布线网络上应用 A* search 算法生成布线 ^[41]
	IPR ^[42] 中集成的 FastRoute
	Ripple2.0 ^[43] 中的简化快速布线及 NCTUgr 算法
	SimPLR 中集成的 BFG-R

静态模型针对一个确定的布局实例对拥挤度进行估计,往往结果不够准确,无法反映真实的布局情况。概率模型的线网拓扑还未确定,有多种可能的布线选择,可通过走线概率进行估计,概率模型与实际走线有较好的一致性。构造式模型使用简化的整体布线算法生成近似布线结果,构造式模型能得到的拥挤评估结果准确度最高。从 ISPD 2011 可布性驱动布局竞赛结果来看,概率模型和构造式模型都具有较好的可行性和可扩展性。

4.3 拥挤度消减技术

目前针对拥挤的处理主要通过各种方法将拥挤区域的单元或线网移出,或者给拥挤区域加入空白来达到降低拥挤的目的。根据解决问题所处的阶段不同,分为整体布局阶段、详细布局阶段和后处理的消减策略,如表 3 所示。

表 3 当前的拥挤度消减技术

分类	具体方法
整体布局阶段	可移动目标重定位 ^[44]
	单元膨胀 ^[45]
	宏单元稀疏性 ^[46]
	引脚密度控制
	膨胀/收缩布局区域 ^[47]
详细布局阶段	局部布局完善
	基于小窗口中斯坦纳长度的线性布局 ^[48]
	目标函数中考虑拥挤度 ^[49]
	单元交换 单元移位
后处理	空白注入或重新分配 ^[50]
	模拟退火 ^[51]
	线性规划 ^[52]
	网络流
	基于 GCell 膨胀的模块移位 单元膨胀

在整体布局阶段,最常用的技术是可移动单元重定位和单元膨胀。根据布局算法的类型不同,比如二次解析、基于最小割或模拟退火的,实施这些方法时需要布局算法进行修改,包括改变优化函数等。还可以在布局中间阶段进行其它的优化措施,然后交付设计流的下一步骤。在详细布局阶段,最常用的技术是单元交换和单元移位,往往通过修改目标函数的方法来进行拥挤度的控制。后处理策略则是在完成布局之后再对单元或空白的位置进行优化,以降低拥挤度。

5 结束语

传统的自动布局算法侧重点在于减小总线长,而对可能引发的局部布线困难等问题考虑不足。

可布线驱动的布局算法通过对拥挤度的分析和消减策略,提高了电路的可布性。随着现代 IC 的发展,电子芯片中电路元件和连线的数量可达千万级,并仍在继续增长,平面布局未来可能的发展方向如下:

关于布线拥挤度的研究,电路的可布线性是当前研究的热点问题。对拥挤度评价模型进一步改进,提高估计效率,获得更客观合理的拥挤度状况。拥挤度消减策略不可避免地以增加线长为代价,如何更好的平衡这一矛盾,开发出更高效的拥挤度消减算法仍有待研究。

多目标优化,除了总线长及拥挤度外,需要对电路的功耗、时延等约束投入更多的关注,在多目标优化之中寻求最佳平衡点。

布局结果评估指标的研究,当前的布局结果评价体系仍然是以总线长为中心,可布线驱动的布局评估也是在线长的基础上加上总拥挤度、平均拥挤度等的考虑。如何更科学地反映出实际的布局质量,考察布局算法的优化度和可扩展性,建立统一可靠的评价体系等问题仍有待研究。

3D 布局的研究,3D 布局技术为提升整体电路性能提供了新的思路。第 3 个维度为物理设计提供的更高的灵活度,可以进一步降低平均连线长度和电路时延,但大大增加了算法复杂度,如何高效地进行 3D 布局也是未来应关注的重点之一。

参考文献:

- [1] Sherwani N A. Algorithms for VLSI physical design automation [M]. Kluwer academic publishers, 1995.
- [2] Shahookar K, Mazumder P. VLSI cell placement techniques [J]. ACM Computing Surveys (CSUR), 1991, 23(2): 14720.
- [3] Hanan M, Wolff P K, Agule B J. Study of Placement Techniques [J]. Journal of Design Automation & Fault-Tolerant Computing, 1976, 1(1): 28-61.
- [4] Sigl G, Doll K, Johannes F M. Analytical placement: A linear or a quadratic objective function [C]//Proceedings of the 28th ACM/IEEE Design Automation Conference. 1991: 427-432.
- [5] Spindler P, Schlichtmann U, Johannes F M. Kraftwerk2—a fast force-directed quadratic placement approach using an accurate net model [J]. Computer-Aided Design of Integrated Circuits and Systems, 2008, 27(8): 1398-1411.
- [6] Kleinhans J M, Sigl G, Johannes F M, et al. Gordian: VLSI placement by quadratic programming and slicing optimization [J]. Computer-Aided Design of Integrated Cir-

- cuits and Systems ,1991 ,10(3) : 356-365.
- [7] Kim M C , Hu J , Lee D J , et al. A SimPLR method for routability-driven placement [C]//Computer-Aided Design (ICCAD) , 2011 IEEE/ACM International Conference on. 2011: 67-73.
 - [8] Hu J , Roy J A , Markov I L. Completing high-quality global routes [C]//Proceedings of the 19th international symposium on Physical design. 2010: 35-41.
 - [9] He X , Huang T , Xiao L , et al. Ripple: an effective routability-driven placer by iterative cell movement [C]//Proceedings of the International Conference on Computer-Aided Design. 2011: 74-79.
 - [10] Fiduccia C M , Mattheyses R M. A linear-time heuristic for improving network partitions [C]//Design Automation ,19th Conference on. 1982: 175-181.
 - [11] Nam G J , Cong J J. Modern circuit placement: best practices and results [M]. Springer Science & Business Media ,2007.
 - [12] Sechen C , Sangiovanni-Vincentelli A. The TimberWolf placement and routing package [J]. IEEE Journal of Solid-State Circuits ,1985 ,20(2) : 510-522.
 - [13] Betz V , Rose J. VPR: A new packing , placement and routing tool for FPGA research [C]//Field-Programmable Logic and Applications. 1997: 217-222.
 - [14] Roy J A , Papa D A , Adya S N , et al. Capo: robust and scalable open-source min-cut floorplacer [C]//Proceedings of the 2005 international symposium on Physical design. 2005: 221-226.
 - [15] Taghavi T , Yang X. Dragon2005: Large-scale mixed-size placement tool [C]//Proceedings of the 2005 international symposium on Physical design. 2005: 245-247.
 - [16] Agnihotri A R , Ono S , Li C , et al. Mixed block placement via fractional cut recursive bisection [J]. Computer-Aided Design of Integrated Circuits and Systems , 2005 ,24(5) : 748-761.
 - [17] Chan T , Cong J , Sze K. Multilevel generalized force-directed method for circuit placement [C]//Proceedings of the 2005 international symposium on Physical design. 2005: 185-192.
 - [18] Kahng A B , Wang Q. Implementation and extensibility of an analytic placer [J]. Computer-Aided Design of Integrated Circuits and Systems ,2005 ,24(5) : 734-747.
 - [19] Chen T C , Jiang Z W , Hsu T C , et al. A high-quality mixed-size analytical placer considering preplaced blocks and density constraints [C]//Computer-Aided Design , ICCAD ' 06 IEEE/ACM International Conference on. 2006: 187-192.
 - [20] Agnihotri A R , Madden P H. Fast analytic placement using minimum cost flow [C]//Design Automation Conference , ASP-DAC07. 2007: 128-134.
 - [21] Tsay R S , Kuh E S , Hsu C P. PROUD: A sea-of-gates placement algorithm [J]. Design & Test of Computers , 1988 ,5(6) : 44-56.
 - [22] Brenner U , Struzyna M. Faster and better global placement by a new transportation algorithm [C]//Design Automation Conference. 2005: 591-596.
 - [23] Nam G J , Reda S , Alpert C J , et al. A fast hierarchical quadratic placement algorithm [J]. Computer-Aided Design of Integrated Circuits and Systems ,2006 ,25(4) : 678-691.
 - [24] Viswanathan N , Nam G J , Alpert C J , et al. RQL: global placement via relaxed quadratic spreading and linearization [C]//Design Automation Conference ,44th ACM/IEEE. 2007: 453-458.
 - [25] Kennings A , Vorwerk K P. Force-directed methods for generic placement [J]. Computer-Aided Design of Integrated Circuits and Systems , 2006 ,25(10) : 2076-2087.
 - [26] Hu B , Marek-Sadowska M. FAR: Fixed-points addition & relaxation based placement [C]//Proceedings of the 2002 international symposium on Physical design. 2002: 161-166.
 - [27] Hu B , Marek-Sadowska M. Multilevel fixed-point-addition-based VLSI placement [J]. Computer-Aided Design of Integrated Circuits and Systems ,2005 ,24(8) : 1188-1203.
 - [28] Viswanathan N , Chu C C N. FastPlace: efficient analytical placement using cell shifting , iterative local refinement , and a hybrid net model [J]. Computer-Aided Design of Integrated Circuits and Systems ,2005 ,24(5) : 722-733.
 - [29] Eisenmann H , Johannes F M. Generic global placement and floorplanning [C]//Proceedings of the 35th annual Design Automation Conference. 1998: 269-274.
 - [30] Kim M C , Lee D J , Markov I L. SimPL: An effective placement algorithm [J]. Computer-Aided Design of Integrated Circuits and Systems ,2012 ,31(1) : 50-60.
 - [31] Xiu Z , Ma J D , Fowler S M , et al. Large-scale placement by grid-warping [C]//Proceedings of the 41st annual Design Automation Conference. 2004: 351-356.
 - [32] Wang M , Yang X , Sarrafzadeh M. Congestion minimization during placement [J]. Computer-Aided Design of Integrated Circuits and Systems ,2000 ,19(10) : 1140-1148.
 - [33] Yang X , Kastner R , Sarrafzadeh M. Congestion estimation during top-down placement [J]. Computer-Aided Design of Integrated Circuits and Systems ,2002 ,21

- (1): 72-80.
- [34] Mayrhofer S, Lauther U. Congestion-driven placement using a new multi-partitioning heuristic [C]//Computer-Aided Design, IEEE International Conference on. 1990: 332-335.
- [35] Roy J A, Markov I L. Seeing the forest and the trees: Steiner wirelength optimization in placement [J]. Computer-Aided Design of Integrated Circuits and Systems, 2007, 26(4): 632-644.
- [36] Brenner U, Rohe A. An effective congestion-driven placement framework [J]. Computer-Aided Design of Integrated Circuits and Systems, 2003, 22(4): 387-394.
- [37] Wang M, Yang X, Eguro K, et al. Multi-center congestion estimation and minimization during placement [C]//Proceedings of the 2000 international symposium on Physical design. 2000: 147-152.
- [38] Lou J, Thakur S, Krishnamoorthy S, et al. Estimating routing congestion using probabilistic analysis [J]. Computer-Aided Design of Integrated Circuits and Systems, 2002, 21(1): 341.
- [39] Kahng A B, Xu X. Accurate pseudo-constructive wirelength and congestion estimation [C]//Proceedings of the 2003 international workshop on System-level interconnect prediction. 2003: 61-68.
- [40] Spindler P, Johannes F M. Fast and accurate routing demand estimation for efficient routability-driven placement [C]//Proceedings of the conference on Design, automation and test in Europe. 2007: 1226-1231.
- [41] Westra J, Groeneveld P. Is probabilistic congestion estimation worthwhile [C]//Proceedings of the 2005 international workshop on System level interconnect prediction. 2005: 99-106.
- [42] Pan M, Chu C. IPR: an integrated placement and routing algorithm [C]//Proceedings of the 44th annual Design Automation Conference. 2007: 59-62.
- [43] He X, Huang T, Chow W K, et al. Ripple 2.0: High quality routability-driven placement via global router integration [C]//Design Automation Conference (DAC), 50th ACM/EDAC/IEEE. 2013: 1-6.
- [44] Jiang Z W, Su B Y, Chang Y W. Routability-driven analytical placement by net overlapping removal for large-scale mixed-size designs [C]//Design Automation Conference, 45th ACM/IEEE. 2008: 167-172.
- [45] Hou W, Yu H, Hong X, et al. A new congestion-driven placement algorithm based on cell inflation [C]//Proceedings of the 2001 Asia and South Pacific Design Automation Conference. 2001: 605-608.
- [46] Hsu M K, Chou S, Lin T H, et al. Routability-driven analytical placement for mixed-size circuit designs [C]//Proceedings of the International Conference on Computer-Aided Design. 2011: 80-84.
- [47] Parakh P N, Brown R B, Sakallah K A. Congestion driven quadratic placement [C]//Proceedings of the 35th annual Design Automation Conference. 1998: 275-278.
- [48] Saxena P. On controlling perturbation due to repeaters during quadratic placement [J]. Computer-Aided Design of Integrated Circuits and Systems, 2006, 25(9): 1736-1743.
- [49] Zhang Y, Chu C. CROP: fast and effective congestion refinement of placement [C]//Computer-Aided Design-Digest of Technical Papers, IEEE/ACM International Conference on. 2009: 341-350.
- [50] Li C, Xie M, Koh C K, et al. Routability-driven placement and white space allocation [J]. Computer-Aided Design of Integrated Circuits and Systems, 2007, 26(5): 858-871.
- [51] Cheng C L E. RISA: accurate and efficient placement routability modeling [C]//Proceedings of the 1994 IEEE/ACM international conference on Computer-aided design. 1994: 690-695.
- [52] Li Z, Wu W, Hong X. Congestion driven incremental placement algorithm for standard cell layout [C]//Proceedings of the 2003 Asia and South Pacific Design Automation Conference. 2003: 723-728.