

DOI:10.19816/j.cnki.10-1594/TN.2021.02.086

超大规模集成电路布线算法综述

李天羽, 苏楚凡, 贾小涛

(北京航空航天大学集成电路科学与工程学院 北京 100191)

摘要: 作为超大规模集成电路物理设计的重要组成部分, 布线算法根据布局结果以及网表完成逻辑线网的连接, 同时满足时延、设计规则、可制造性等要求。介绍了布线领域的核心技术路线, 阐述了主要的路径搜索算法及全局布线、详细布线的主要算法及其特点, 还介绍了在新兴工艺节点下面向可制造性设计的布线算法。

关键词: 超大规模集成电路; 电子设计自动化; 布线

中图分类号: TN470.2 文献标识码: A 国家标准学科分类代码: 510.30

A survey of routing algorithms for VLSI Design

LI Tianyu, SU Chufan, JIA Xiaotao

(School of Integrated Circuit Science and Engineering, Beihang University, Beijing 100191, China)

Abstract: As an important part of the physical design of VLSI, the target of routing algorithm is to achieve 100% routability based on the placement results and netlist while satisfying the requirement of timing, design rule, manufacturability and so on. Introduce the key approaches of routing, expound the main algorithms of path search, global routing, and detailed routing. Algorithms of design for manufacturability on routing under emerging technology nodes are also introduced.

Keywords: VLSI; EDA; routing

0 引言

近年来, 集成电路领域发展越来越迅猛, 晶体管数量随集成电路制造工艺发展逐年增加, 芯片内包含的逻辑门数量急剧提升, 这给集成电路设计带来了巨大的难题。由于超大规模集成电路 (very-large-scale integration circuit, VLSI) 逻辑的高度复杂性, 其物理设计往往需要使用计算机辅助设计工具来完成。这就向电子设计自动化 (electronic design automation, EDA) 工具提出了严峻挑战。

在物理设计过程中, 布线是极其重要的一环。布线工作占据了EDA过程的大部分时间, 甚至在大部分情况下, 自动布线的结果还需要设计人员在后

期手工调整。具备优秀的布线速度及高布线质量的布线器对缩短芯片设计周期有着至关重要的作用。基于上述背景, 本文将介绍超大规模集成电路物理设计过程中布线工作的各个阶段及其主要方法。

1 布线问题及算法概述

布线过程需要基于芯片逻辑单元的布局结果, 将逻辑单元的各个引脚按网表连接, 优化布线的线长、通孔等代价的同时, 满足芯片在时延、设计规则、可制造性等方面的要求。

在实际的布线问题中, 逻辑单元数量极其庞大以及要连接的线网数量规模之巨大使得连线问题高度复杂化。布线问题已经被证明是一个 NP 完全

李天羽, 硕士, 主要研究方向为大规模集成电路布线算法。E-mail: liitianyu@163.com

苏楚凡, 硕士, 主要研究方向为大规模集成电路布线算法。E-mail: susuchufan1213@163.com

贾小涛, 助理教授, 主要研究方向为大规模集成电路物理设计、贝叶斯深度学习加速器等。E-mail: jiaxt@buaa.edu.cn

(NP-complete)问题^[1-2],因此,为布线问题寻找全局最优解不具备可行性。另外,布线过程受制造工艺制约,存在着很多约束和限制,这也令布线的复杂度进一步提升。

芯片布线阶段的版图中,具备多层用于走线的金属层,其层内器件通过金属线(wire)进行连接,层与层间通过通孔(via)连接。每层金属层具备一个给定的优先走线方向,一般为水平或竖直。相邻层的优先走线方向互相垂直,这样可以大大减少相邻层金属线间的同向并行走线长度,从而降低电容电感等效应,并且可以更好地利用布线资源。

1.1 布线问题概述

本节给出布线问题的主要定义,包含输入、输出、优化目标、约束。

布线问题的输入主要包含:1)芯片的布局信息,其中描述了各个逻辑单元的摆放位置,逻辑单元内引脚(端口)的位置,版图中输入/输出引脚的位置等。2)网表,其中记录了各个逻辑线网的输入和输出端口信息。3)设计工艺,其中包含了各层金属线、各种通孔的几何形状信息,以及各种版图设计规则及约束。4)部分关键线网的时延信息。5)各金属线及通孔的电气特性。

布线问题的输出为网表内各线网的连接方式,包含金属线及通孔的几何描述信息。简而言之就是一个尽可能满足约束条件并更少得需要人工调整线网的布线结果。

布线问题的主要优化目标有:1)布通率,提升布通率是布线问题的首要目标。自动布线完成后,如果线网未完全连接,将导致开路(open),发生开路的线网必须由设计人员手工完成连接。2)布线总线长,线长影响了芯片的制造成本,且线长越长,线网的时延与电容越大,这会使芯片的稳定性下降、功耗增高。3)通孔数量,通孔不仅具备更大的时延,还会增加芯片制造失败的几率。4)串扰和时延,线网间的电容电感等现象会在线网间产生串扰,增大时延。因此布线算法要对时延敏感线网及关键线网做出特殊设计,以降低芯片的总时延,提高芯片稳定性。

布线问题的主要约束有:1)线网连接正确性,指不同的线网之间不能相互交叉,否则会导致短路

(short),从而影响芯片功能的正确性及稳定性。2)布线障碍,指布线过程中被占用的区域,包括逻辑单元及芯片版图中的布线障碍,以及部分已布线网,如电源线网和时钟线网等。3)版图设计规则,主要由制造工艺决定。制造厂商在制造过程中,为满足芯片最基本的可制造性,尽可能让芯片的设计可用,根据现有的制造技术设计了种种版图设计规则。主要包含:金属线的最小宽度(min-width);金属线间及金属线与布线障碍间的最小线间距(min-spacing);一条金属线的最小面积(min-area),当金属线的宽度固定为最小宽度时,最小面积约束也被称作最小长度约束。设计规则违反(design rule checking violations, DRC)的数目很大程度上代表了布线质量的高低。

1.2 布线算法概述

由于布线问题的约束较为复杂,问题的求解规模较大,对整个芯片的布线问题求解全局最优解会带来巨大的时间开销。因此,布线算法采用了分而治之的策略,即将布线分为两个主要部分:全局布线和详细布线。

在全局布线(又称总体布线)阶段,整个芯片被分为多个区域,并将线网的连接关系对应到区域上,在区域间连线。由于全局布线不需要考虑版图设计规则的约束,甚至不需要考虑大部分几何级问题,其布线模型的建模较为简单,即使在具有数以万计的区域的芯片版图上布线,也只产生较低的时间开销。

详细布线在全局布线的基础上,分别解决每个区域内部的连线问题。在有限的区域内,详细布线要实现各个连接关系100%布通率,并尽量最小化线长、通孔数,以及DRC等,因此详细布线问题的建模更为复杂。此外,由于布线区域数量较多,还需要尽可能地节省运行时间和内存消耗。

2 路径搜索算法

路径搜索算法就是寻找源点和目标点之间连通路径的算法,在此基础上还需满足路径最短或路径开销最小。绝大多数布线问题都可以抽象成寻路问题来解决,将线网的连线问题看作是寻找两点之间最短路径的问题,在不同的布线阶段选择合理的路

径搜索算法是布线问题的关键步骤。布线问题中,使用最广泛的是迷宫布线类算法,包括李氏算法、A*算法等,此类算法一般在网格状图结构上,通过递归或迭代搜索线网的最优布线路径,对多个线网进行顺序布线(或称串行布线)。除此之外,还有同属于顺序布线算法的线搜索算法,以及可以对所有线网同时进行布线的同时布线(或称**并行布线**)算法,如多商品流算法。本节将介绍布线问题中最基础同时也是最常用的一些路径搜索算法。

2.1 李氏算法

李氏算法^[3]是一种应用广泛的路径搜索算法。这是一种基于网格状图结构的广度优先搜索算法。

该算法从源点开始,依次访问网格各方向上的相邻节点,并为该节点标记从源点到该点的已知布线代价。对每个访问到的节点重复相同的过程。这会在整个布线图中产生一个连续扩散的波,当这个波扩散到了目标点,即目标节点被他的相邻节点访问到时,就找到了这样一条路径连通源点与目标点。随后算法进入路径回溯阶段,从目标点开始,直到回溯到源点,此时就生成了一条最优路径。

2.2 A*算法

由于李氏算法在节点的各个方向上同时进行搜索扩展,大部分被访问的节点都不在最终的路径方向上,这种行为导致了李氏算法的时间和内存开销极大。

在李氏算法的基础上,对节点加入启发式代价估计的A*算法^[4],将搜索方向逐步向目标点偏移,显著缩小了搜索空间。该算法将当前节点到源点的路径代价,和启发式估算出的当前节点到目标点的潜在代价合并作为当前节点的总代价来找到一条最优的路径。节点代价可由如下公式表示:

$$f(n) = g(n) + h(n) \quad (1)$$

其中, $f(n)$ 表示节点 n 总的代价, $g(n)$ 是源点到节点 n 的路径代价, $h(n)$ 是由某种启发式算法预测出的节点 n 到目标点的预估路径代价。在普通的寻路任务中,使用节点 n 到目标点的曼哈顿距离作为 $h(n)$ 。当 $h(n) = 0$ 时,A*算法退化为普通的李氏算法。

在搜索空间有限且存在最优路径的前提下,当预测函数 $h(n)$ 满足一致性和可接受性时,A*算法可以保证找到最优解^[4]。其中,一致性指预测函数 $h(n)$ 满足 $h(a,b) + h(b,c) \geq h(a,c)$,这确保 $f(n)$ 在搜索过程中递增;可接受性指预测函数 $h(n)$ 满足 $h(a,b) \leq r(a,b)$,其中 $r(a,b)$ 指节点 a 到节点 b 的真实代价。

2.3 线搜索算法

上述迷宫布线类算法可以确保找到最优路径,但由于对路径上的每个节点都进行了搜索和代价计算,其复杂度由源点到目标点的距离决定,在长距离寻路中,会产生极大的时间和内存开销,即使是搜索空间被定向收缩的A*算法依旧不能完全解决这个问题。

Mikami等^[5]在1968年提出了第1个使用线段代替节点搜索的路径搜索算法——线搜索算法。该算法以源点和目标点为基点,首先从两个基点各引出两条相互垂直的直线,当直线遇到障碍物或布线区域边界时,会在当前的直线上选取新的基点并沿垂直方向生成一条新的直线,直到从源点和目标点延伸出的多级直线发生相交,则找到了一条路径连通源点和目标点。由于其并没有遍历所有的搜索空间,所以线搜索算法并不能保证找到最优解。

Hetzel^[6]在2002年提出的Hetzel算法将以线段代替节点进行搜索的思想与A*算法结合,在保证找到最优解的情况下具备比A*算法更快的求解速度。

2.4 同时布线算法

上述顺序布线算法依据预先给定的线网顺序对线网进行布线,布线过程中先布线网会抢占后布线网的布线资源,不同的布线顺序可能会生成完全不同的布线结果。早期的一项研究指出,在布线开始之前找到一个最优布线顺序是NP难问题,且不存在一个在所有情况下都达到最优的布线顺序选取策略^[7]。对所有线网同时进行布线的同时布线算法可以避免线序对布线结果的影响。多商品流(multi-commodity flow, MCF)算法将布线问题看作是一个多商品流问题^[8],并将其建模成整数线性规划(integer linear programming, ILP)问题进行求解。

该算法将每个两端线网看做是一个商品流,由

商品的源点和目标点以及商品流经的路径组成。随后将整个布线区域建模成一个三维的布线网格,网格点可以作为商品流的流经点,点与邻近点之间的线段称作边。

在 ILP 问题的模型中,网格的点和边作为变量,用边和点的流量表示线网对边和点的使用情况,容量代表边和点上同时流经的最大的线网数量(在详细布线中一般取1)。**目标函数为边上所有线网的流量与其相应布线代价的积的累加和。**在约束条件下最小化目标函数可以获得布线的最短线长。其约束主要包含**各个线网在点或边上的流量之和不超过点或边的容量约束、基于流量守恒定律的连通性约束**等。

3 全局布线算法

全局布线的主要目标是为后续的详细布线提供指导,从而降低布线问题的复杂程度,减少整体的时间开销。作为整个布线任务中多种优化目标的主要影响因素,**全局布线要在给定布线资源的情况下,优化线长、通孔数及关键线网时延等目标函数。**

相较于详细布线,全局布线的布线网格具备更粗的粒度,每一个网格被称作**全局布线单元(global routing cell, GCell)**,其内部包含了多条详细布线的网格线。全局布线单元可以是二维排布,也可以是三维立体排布。每一个全局布线单元被视作全局布线问题中的一个点,相邻近的单元所代表的点之间具备一条有**容量**的边,对应相邻单元间的边界,这条边的容量代表能穿过此边界的金属线的最大数目。因此,可以通过各边的容量与穿过边的金属线数量估计全局布线的资源拥挤程度。若穿过某边界的金属线数量大于此边界对应边的容量,称在此边界上发生**溢出(overflow)**。发生溢出的边的溢出数量 $overflow_e = d_e - c_e$,其余边的 $overflow_e = 0$,其中 d_e 为边上的金属线数量, c_e 为边的容量。

全局布线的输出是各个线网的 GCell 级路径,此路径仅仅在较粗粒度的全局布线单元间进行走线,为后续的详细布线提供走线指导。随后详细布线阶段,依据此指导将 GCell 级路径细化成几何路径。评价全局布线算法的主要指标有线长、通孔数、溢出数量、运行时间。

全局布线针对引脚数目不同的线网使用不同的算法处理,主要分为两端线网和多端线网两种。

3.1 两端线网

对于某待连接的两端线网,一般采用迷宫布线类算法或模式布线(pattern routing)类算法两类方法。

对于全局布线,迷宫布线算法及其变体 A* 算法一般通过设计开销函数来躲避拥塞。开销函数的值往往在边界上穿过的金属线数量接近及超过边容量时迅速增大^[9],如式(2)所示。

$$\text{cost} = 1 + \frac{h}{1 + e^{-k(d_e - c_e)}} \quad (2)$$

迷宫布线类算法具备着极其强大的拥塞避免能力,可以**对单个线网给出质量极高的布线结果**,但即使通过开销函数定向减小搜索空间,迷宫布线类算法仍然会**产生较高的时间开销**。

模式布线类算法指使用预先设计的模式对两端线网进行布线,常用的模式有**L型(1拐点)、Z型(2拐点)^[10]与U型(2拐点,绕路连线)**,如图1所示。由于路径的搜索空间远小于迷宫布线,模式布线类算法**具备极快的布线速度**,但其**布线结果相对较差**。对于跨越 $m \times n$ 网格的两端线网,L型模式仅在两种布线方式中选取代价较低的一种,而Z型模式会考虑 $m + n$ 种不同的布线方式。因此,Z/U型模式避免拥塞的**灵活性要远大于L型算法**,但**相应地,其单个线网的布线时间复杂度更高**。

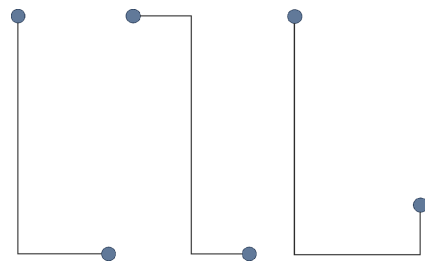


图1 L型/Z型/U型模式

Fig.1 L/Z/U pattern

随布线问题复杂度的上升,L/Z/U型模式布线已经不能满足问题的需求。**文献[11]**提出了单调布线(monotonic routing),其布线路径单调地向右上(左下)或左上(右下)搜索,可以搜索到

$(m+n-2)!/(m-1)!(n-1)!$ 种布线方式,且时间复杂度与Z型相同,为 $O(mn)$,但由于其走线拐点较多,其布线通孔数会增加。文献[12]提出了三拐点布线(3-bend routing),其路径支持最多3个拐点,且可以绕路布线。三拐点布线同样使用 $O(mn)$ 的时间复杂度,具备与迷宫布线相近的拥塞避免能力,且由于其最多使用3个拐点,极大地减少了通孔数。

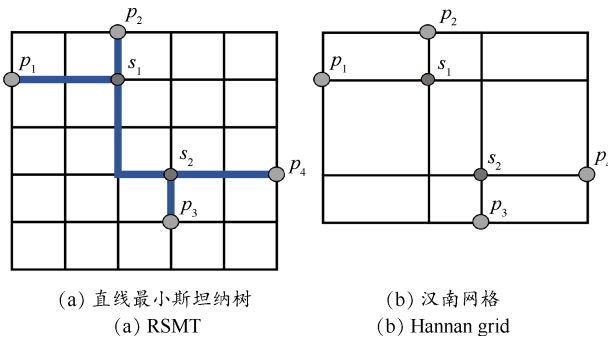


图2 直线最小斯坦纳树与汉南网格的例子

Fig.2 Example of RMST and Hannan grid

3.2 多端线网

对于多端线网,很难直接使用迷宫布线或模式布线进行求解。目前主流的解决方法是寻找线网的矩形最小斯坦纳树(rectilinear steiner minimal tree, RSMT)。

一个线网的直线最小生成树(rectilinear minimal spanning tree, RMST)是连接这个线网所有端点的树中线长最小的树。而RSMT是一颗带有斯坦纳点(steiner point)的RMST,其线网各个端点的连线尽可能多地共享相同的路径,如图2(a)所示。线网中的斯坦纳点从汉南网格(Hannan grid)中选取^[13],汉南网格的格点必定在水平与竖直两个方向上分别与某个线网端点对齐,如图2(b)所示。寻找RSMT的过程是NP完全问题^[14],因此应用中往往会使用启发式算法生成一个RSMT的近似解。基于快速查找表(fast lookup table, FLUTE)的算法^[15]巧妙地预先存储了所有九端及以下线网的斯坦纳树,使用极小的空间代价极大地提升了斯坦纳树的生成速度,并且九端及以下线网都可以找到最优解,高于九端的线网经过拆分,也可以在较短的时间内获得一个十分近似的次优解。一些研究可以花费稍长的时间开销

获得更精确的近似解^[16-18]。

多端线网可以直接选取RSMT或其近似解作为布线结果^[9],但直接使用针对线长生成的RSMT会导致拥塞。FastRoute系列算法^[9, 11-12, 20]使用基于拥塞程度^[9]或通孔数量^[12]生成的RSMT将多端线网拆分成两端线网,并使用3.1节中的方法进行对各两端线网进行布线。

3.3 拥塞估计

随着问题复杂度的上升,全局布线的布线资源越来越紧张,直接进行布线会发生大量无法避免的溢出。

FastRoute^[9]使用FLUTE算法生成的RSMT计算出拥塞图,并依据此拥塞图重新构建RSMT。对于任一线网的RSMT,依据斯坦纳树上的点(端点、斯坦纳点)之间的连接关系生成两端线网。在拥塞图的初始化阶段,对所有两端线网分别进行拥塞程度计算。若两端点在水平或竖直方向具有相同的坐标,则直接将拥塞图上两端点间所有的边的拥塞程度加1.0。否则,对于两方向上坐标都不相同的线网,计算其两种可能的L型连接方式,并将两种可能的路径上的边的拥塞程度都加0.5。

拥塞图的初始化阶段过后,由于所有L型连接方式都在拥塞程度上平均分配,在拥塞程度较大的区域,拥塞程度可能会超过其容量。为了更好地规划拥塞图的拥塞分布,使用一个快速的拆线重布,依次将各L型连接的两端线网提供的拥塞程度去除后,选择拥塞程度更小的一条路径,并将此路径的拥塞程度加1.0。

3.4 线序与拆线重布

上文提到的全局布线算法都是分别对单一线网进行布线,即顺序布线,因此布线顺序对最终布线质量的影响很大。

一般来讲,具备如下几种线序选择策略:1)按端点数量的升序布线。端点数量较多的线网往往会造成内部空间的拥塞。2)按线长升序布线以降低容量溢出,提升可布性。线长较短的线网布线范围比较小,灵活性有限,而线长较长的线网则可以在更大范围内绕路,寻找更优解。3)按线长降序布线以降低总体时延。优先布线长较长的线网,降低线长较长

的关键线网对总体时延的影响,以提升电路的总体性能。4)依据预估的拥塞程度布线,优先对较为拥塞的区域内的线网进行布线。

以上几种启发式的线序选择策略往往不能完全避免溢出。拆线重布(rip-up and reroute)策略在初次布线完成后,对发生溢出的拥塞区域的线网进行拆除,并迭代地调整线序多次重新布线,直到溢出不再减少,或达到运行时间限制为止。一般来讲,在初始布线时使用模式布线,这样在大量简单线网上拥有极快的布线速度,在拆线重布阶段,使用迷宫布线类算法,或时间复杂度较高的模式布线算法,以尽可能地提升重布线网的灵活性,并减少溢出次数,从而在整体的运行速度和布线质量间取得平衡。

文献[21]提出了基于拥塞协商的拆线重布策略,文献[20,22]在此基础上设计了不同的估计拥塞的方法。文献[23]提出了拓扑感知拆线重布策略,在对多端线网进行拆线时选择线网中拥塞程度较大的部分连接进行拆除,保留大部分的拓扑关系。

3.5 同时全局布线

使用多商品流模型对全局布线进行建模,从而在全局求解0-1整数线性规划问题,可以很大程度上改善由线序导致的布线质量下降。但过大的求解规模会导致其运行速度远慢于带拆线重布的顺序布线算法。

文献[24]通过将多商品流模型的整数线性规划近似为线性规划从而提高其可求解性与性能。BoxRouter算法^[25]使用L型模式的两种布线可能作为任一两端线网的候选布线路径,在建模时直接对整条路径而非路径上的每一条边进行约束,同时改善了多商品流模型的定义,在不损失布线质量的同时大大缩小了问题的规模,提升了运算速度。

4 详细布线

在全局布线完成后,整个芯片的布线区域中较粗粒度的连接方式已经初步确定,但全局布线中GCell之间的连接还不能直接表示成金属线,需要对GCell之间的连接进行轨道分配,也是详细布线的前置步骤之一。此后,详细布线要在局部范围内满足线网连通率、设计规则违反等关键约束,并尽量降低

线长和通孔数。

现代VLSI主流的布线策略为区域布线,布线过程中依据全局布线网格将整个芯片的布线区域分割成多个详细布线窗口(detail routing window),每个详细布线窗口可由 $m \times n$ 的GCell组成(一般取 $m = n$),更大的详细布线窗口会导致更大的时间和内存开销,但可以改善布线的质量。

4.1 搜索空间表示

由于详细布线要考虑更多设计规则违反等几何级问题,详细布线在构建布线图时要远比全局布线图更精细。详细布线的布线图,即寻路算法的搜索空间,主要有两种构建方式:基于网格的(grid-based)和无网格的(gridless)。

基于网格的构建方式中,布线算法在网格线和网格点上进行走线。每层金属层有属于自己的布线网格(routing grid),由于金属线一般按照优先走线方向走线,因此网格线一般指该金属层优先走线方向的网格线。网格线间的距离被称为金属层的pitch,一般大于等于设计规则中最小线宽与最小线间距的和。BonnRoute^[26]算法提出了一种构建布线网格的方式,在上述基于pitch的布线网格的基础上构建了基于形状的网络,支持同层内对不同线宽的金属线进行布线,并解决了网格线与线网引脚不对齐等问题。

针对普通的基于网格的构建方式中存在的空间浪费等局限性,无网格布线允许金属线放置在布线区域的任意位置上,具备较高的布线灵活性。其在每个金属层上,使用树结构分割金属层上的几何形状^[27],随后使用瓦片结构^[28]或连接图^[29]构建路径搜索算法的搜索空间。

由于无网格布线算法会产生较大的时间开销,大部分布线算法都使用布线网格构建搜索空间。

4.2 轨道分配

轨道分配(track assignment, TA)^[30]是全局布线与详细布线的中间步骤。全局布线生成的布线结果仅仅完成了GCell级的连接,连接的路径被称为线段(segment)。在轨道分配阶段中,segment特指没有几何形状及位置信息的线段,只表示GCell间的连接关系。而轨道分配算法中考虑其实际几何形状,更贴

近详细布线中金属线的线段被称作 **iroute**, 如图 3 所示。

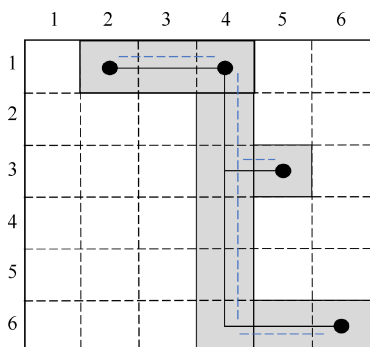


图3 全局布线线段(虚线)与 **iroute**(实线)

Fig.3 Global segments(dashed lines) and iroute(lines)

在全局布线网格中,水平(竖直)方向的一行(列)GCell 共同构成一个面板(panel),其方向与对应布线层的优先走线方向一致,如图 4 所示。每个面板由多条 **布线轨道(track)** 构成,轨道的数量与全局布线边的容量相同。将所有线网的 **iroute** 都分配至轨道上,并保证一个面板内不同线网的各个 **iroute** 在同一轨道上没有重叠,是轨道分配工作的主要任务。

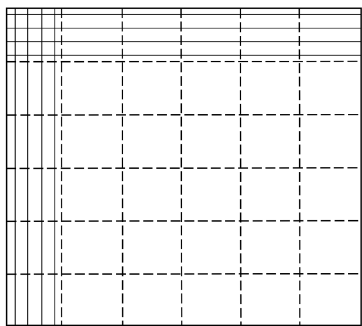


图4 全局布线网格与面板

Fig.4 Global routing grid and panel

找到最优的轨道分配策略被证明是 **NP 难问题**^[30], 因此详细布线一般使用启发式算法求解轨道分配问题,如将其建模成带权的二部匹配问题,使用最短增广路径算法对其进行求解^[31]。

轨道分配阶段结束后,各个 **segment**(或称 **iroute**) 已经在布线区域内具备位置信息。详细布线器会首先将 **segment** 转化为金属线,随后将线网中孤立的金属线、引脚等相连。**RegularRoute** 算法^[32]将轨道分配

过程与详细布线相结合,在轨道分配时直接考虑了 **segment** 与引脚的连接性,分配后的 **segment** 与引脚直接相连。

4.3 多层次布线

多层次布线起源于一种在全局布线和详细布线之间的导线规划算法^[33]。当详细布线失败时,其结果会返回给导线规划模型,重新规划走线后,再进入详细布线更新布线结果。较为流行的多层次布线的布线算法被称为 V 型模型^[34]。该算法采用自底向上,将较精细的网格粗化后,执行本级布线并预测下一级的拥塞情况,当粗化到一定程度后再进行自顶向下的细化布线。V 型模型往往在自底向上过程中使用模式布线进行快速的拥塞估计,并在自顶向下过程中使用迷宫布线类算法进行连线。

4.4 拆线重布

在顺序布线算法中,先布线网占用的布线资源后布线网完全无法访问,不良的线序会严重地降低布通率,增加 DRC 数量。详细布线阶段同样使用了拆线重布策略来提高布线质量。在基于拥塞协商的拆线重布方法中,首先根据布线生成的路径及当前代价,通过标记设计规则违反的类型来增加标记边上网格点的拥塞代价,再拆除已布线网,按照更新后的代价重新进行布线,并多次重复该操作直至 DRC 不再减少,或达到运行时间上限。这种方法可以利用历史布线信息启发式地解决详细布线阶段的拥塞问题,将原有布线结果中拥塞的线网疏散开来,对布线中出现的 short 及 DRC 都有很好地解决效果。

4.5 同时详细布线

由于顺序布线算法的种种限制,多商品流布线算法这类同时布线算法在详细布线问题中也有应用的空间。顺序布线算法中,先布线网对后布线网的影响往往通过代价函数的形式加入到布线算法中,从而使后布线网可以规避最小线间距等 DRC。但在多商品流布线算法中,同时对所有线网进行布线,线网间的影响不能通过开销函数加入到优化目标中,只能以约束的形式存在。且由于其求解的是整数线性规划问题,在考虑设计规则的同时保证约束的线性性成为了同时详细布线算法的主要难点。**MCFRoute**^[35-36]在多商品流模型的基础上设计了最小

线间距约束,分别依据金属线间、通孔外壳间、金属线与通孔外壳间的距离设计约束,使多商品流算法在详细布线问题中的应用存在可能。

5 面向可制造性设计的布线算法

在先进的集成电路工艺节点下,仅仅在制造时使用各种分辨率增强的技术,已经不能保证良好的可制造性。为了克服这一难题,在芯片的设计阶段就需要为芯片的可制造性进行针对性调整,即面向可制造性设计(design for manufacturability, DFM)。

5.1 支持多重版图光刻技术的布线算法

在 14 nm 之后的工艺节点中,对 pitch 较小的金属层(Metal-2、Metal-3)往往会使用多重版图光刻(multiple patterning lithography, MPL)技术进行加工,因此布线器需要针对 MPL 技术带来的约束设计新的布线策略。

文献[37]首次提出了针对 LELE 式 MPL 的详细布线算法,基于网格染色技术对相应设计规则作出支持。文献[35]首次提出了针对间隔式 MPL 的布线算法。

5.2 引脚访问策略

随着工艺发展,版图内的引脚越来越密集,且由于 MPL 技术带来的设计规则使引脚和金属线间的相互影响大大增加,金属线与引脚的连接变得极其困难。引脚可能不在布线网格上,或引脚附近区域过于拥挤,导致金属线无法与引脚相连。引脚访问(pin access, PA)策略通过为引脚增加引脚访问路径(pin access path, PAP)来使金属线访问到引脚。PAP 的添加常常会与周边物体产生 DRC, PAP 之间也经常有冲突产生。

文献[38]提出了一个在标准单元内部添加 PAP 的策略,对标准单元在版图中旋转方向、网格对齐方式、布局障碍等各种不同的布局情况,需要在布线开始之前计算其无冲突的引脚访问方式。

在新的工艺节点中,引脚细小而规整,且都放置在布线网格上。由于标准单元所在的 Metal-1 层几乎没有布线资源,只能通过在引脚上放置通孔并在 Metal-2 层添加 PAP,但标准单元间的 PAP 可能会产生冲突,如图 5 所示。文献[39]提出了一个支持 MPL

技术的 PA 策略,在布线进行过程中计算添加 PAP 的最优方案,相较于同期最先进的支持 MPL 技术的布线算法^[40]大大提高了线网布通率。

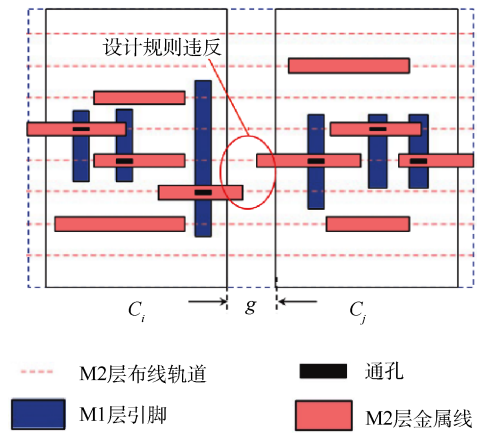


图5 标准单元间的引脚访问路径冲突图示

Fig.5 Illustration of inter-cell conflicts

5.3 冗余通孔插入

在集成电路制造过程中,通孔的制造成功率相对较低,通孔制造失败是芯片良品率下降的主要原因之一。因此,芯片制造厂商推荐在原有通孔附近多打一个冗余通孔,或称双通孔。在布线资源十分紧张的情况下,优秀的冗余通孔插入算法对布线器尤为重要。

文献[41]首次提出了一个在布线过程中考虑冗余通孔插入的详细布线算法,通过增加通孔周围的布线代价使走线尽量避开通孔,有效提升了冗余通孔的插入率,但由于占用了过多的布线资源,其布通率有所降低。文献[42]在轨道分配阶段考虑了冗余通孔插入。文献[43]在后处理阶段进行冗余通孔插入。由于冗余通孔插入后会占用空间,可能使附近的通孔无法进行同样的插入处理。为最大化通孔插入率,算法建立了冗余通孔插入冲突图,通过整数线性规划求解冲突图的最大独立集问题,达到后处理阶段冗余通孔插入的最优。文献[44]提出了考虑冗余通孔插入的多商品流布线算法。该算法在多商品流详细布线算法 MCFRoute 的基础上考虑了冗余通孔插入问题,达到在布线的同时实现冗余通孔插入的目标,提高了冗余通孔插入率。

6 结 论

本文对现代超大规模集成电路布线的主要方法进行了综述,介绍了解决通用布线问题的寻路算法,以及VLSI背景下全局布线与详细布线两阶段算法,另外,介绍了针对先进工艺的面向可制造性设计的布线算法。

对于网格结构的布线图来说,Hetzel的寻路算法是目前较好的选择,在保证最优的同时,具备极快的寻路速度,而对于更广泛的图结构,A*算法则具备更好的鲁棒性和性能。全局布线中,FastRoute系列算法的应用更为广泛。详细布线中,顺序布线的BonnRoute算法布线效果较为突出,MCFRoute算法使得同时详细布线算法的实现变为可能。

参考文献

- [1] RICHARDS D. Complexity of single-layer routing[J]. IEEE Transactions on Computers, 1984(3): 286-288.
- [2] SZYMANSKI T G. Dogleg channel routing is NP-complete[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1985, 4(1): 31-41.
- [3] LEE C Y. An algorithm for path connections and its applications[J]. Ire Trans on Electronic Computers, 1961, 10(3): 346-365.
- [4] HART P E, NILSSON N J, RAPHAEL B. A formal basis for the heuristic determination of minimum cost paths[J]. IEEE Transactions on Systems Science & Cybernetics, 1972, 4(2): 28-29.
- [5] MIKAMI K, TABUCHI K. A computer program for optimal routing of printed circuit conductors[C]. Proceedings of the International Federation for Informatics, 1968: 1475-1478.
- [6] HETZEL A. A sequential detailed router for huge grid graphs[C]. Proceedings Design, Automation and Test in Europe, IEEE, 1998: 332-338.
- [7] ABEL L C. On the ordering of connections for automatic wire routing[J]. IEEE Transactions on Computers, 1972, C-21(11): 1227-1233.
- [8] SHRAGOWITZ E, KEEL S. A global router based on a multicommodity flow model[J]. INTEGRATION, the VLSI journal, 1987, 5(1): 3-16.
- [9] PAN M, CHU C. FastRoute: A step to integrate global routing into placement[C]. 2006 International Conference on Computer-Aided Design (ICCAD'06), 2006: 464-471.
- [10] KASTNER R, BOZORGZADEH E, SARRAFZADEH M. Predictable routing[C]. IEEE/ACM International Conference on Computer Aided Design, ACM, 2000: 110-114.
- [11] MIN P, CHU C. FastRoute 2.0: A high-quality and efficient global router[C]. Design Automation Conference, IEEE, 2013: 250-255.
- [12] XU Y, ZHANG Y, CHU C. FastRoute 4.0: Global router with efficient via minimization[C]. Proceedings of the 14th Asia South Pacific Design Automation Conference, ASP-DAC 2009, Yokohama, Japan, January 19-22, 2009: 576-581.
- [13] HANNA M. On steiner's problem with rectilinear distance[J]. SIAM Journal on Applied Mathematics, 1966: 255-265.
- [14] GAREY M R, JOHNSON D S. Computers and intractability: A guide to the theory of NP-completeness[C]. W. H. Freeman & Co. W. H. Freeman & Co, 1979.
- [15] CHU C, WONG Y C. FLUTE: Fast lookup table based rectilinear steiner minimal tree algorithm for VLSI design[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2008, 27(1): 70-83.
- [16] HAI Z. Efficient steiner tree construction based on spanning graphs[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2004, 23(5): 704-710.
- [17] KAHNG A B, MENDOZA I I, ZELIKOVSKY A Z. Highly scalable algorithms for rectilinear and octilinear steiner trees[C]. Proceedings of the 2003 Asia and South Pacific Design Automation Conference, IEEE, 2003: 827-833.
- [18] GRIFFITH J, ROBINS G, SALOWE J S, et al. Closing the gap: Near-optimal steiner trees in polynomial time[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1994, 13(11): 1351-1365.
- [19] SHERWANI N A. Algorithms for VLSI physical design automation[M]. New York: John Wiley & Sons, Inc., 1998: 111-115.
- [20] ZHANG Y, YUE X, CHU C. FastRoute3.0: A fast and high quality global router based on virtual capacity[C]. IEEE/ACM International Conference on Computer-aided Design. IEEE Press, 2008: 344-349.
- [21] MCMURCHIE L E, EBELING C. Pathfinder: A negotiation-based performance-driven router for FPGAs[J]. Field-Programmable Gate Arrays, International ACM Symposium on, IEEE Computer Society, 1995: 111-117.

- [22] ROY J A, MARKOV I L. High-performance routing at the nanometer scale[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2008, 27(6): 1066-1077.
- [23] CHO M, LU K, YUAN K, et al. BoxRouter 2.0: Architecture and implementation of a hybrid and robust global router[J]. 2007 IEEE/ACM International Conference on Computer-Aided Design, 2007: 503-508.
- [24] ALBRECHT C. Global routing by new approximation algorithms for multicommodity flow[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2001, 20(5): 622-632.
- [25] CHO M, PAN D Z. BoxRouter: A new global router based on box expansion and progressive ILP[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2007, 26(12): 2130-2143.
- [26] GESTER M, D MÜLLER, NIEBERG T, et al. BonnRoute: Algorithms and data structures for fast and good VLSI routing[J]. ACM Transactions on Design Automation of Electronic Systems (TODAES), 2013: 1-24.
- [27] EDELSBRUNNER H. A new approach to rectangle intersections[J]. International Journal of Computer Mathematics, 1983, 13(3-4): 209-219.
- [28] LIU L, TSENG H P, SECHEN C. Chip-level area routing[C]. Proceedings of the 1998 international symposium on Physical design, DBLP, 1998.
- [29] OHTSUKI T. Gridless routers—New wire routing algorithms based on computational geometry[C]. Proceedings of the International Conference on Circuits and Systems, 1985.
- [30] BATTERYWALA S. Track assignment: A desirable intermediate step between global routing and detailed routing[C]. IEEE/ACM International Conference on Computer Aided Design, IEEE, 2002: 59-66.
- [31] JONKER R, VOLGENANT A. A shortest augmenting path algorithm for dense and sparse linear assignment problems[J]. Computing, 1987, 38(4): 325-340.
- [32] ZHANG Y, CHU C. RegularRoute: An efficient detailed router applying regular routing patterns[J]. IEEE Transactions on Very Large Scale Integration Systems, 2013, 21(9): 1655-1668.
- [33] CONG J, JIE F, YAN Z. Multilevel approach to full-chip gridless routing[C]. Computer Aided Design, 2001. ICCAD 2001, IEEE/ACM International Conference on, ACM, 2001: 396-403.
- [34] CHANG Y W, LIN S P. MR: A new framework for multilevel full-chip routing[M]. IEEE Press, 2004.
- [35] JIA X, CAI Y, ZHOU Q, et al. MCFRoute: A detailed router based on multicommodity flow method[C]. IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2014: 397-404.
- [36] CHEN G, PUI C W, LI H, et al. Dr. CU: Detailed routing by sparse grid graph and minimum-area-captured path search[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2020, 39(9): 1902-1915.
- [37] GONCALVES S, ROSA L, MARQUES F. DRAPS: A design rule aware path search algorithm for detailed routing[J]. Circuits and Systems II: Express Briefs, IEEE Transactions on, 2019, 67(7): 1239-1243.
- [38] JIA X, CAI Y, ZHOU Q, et al. A multicommodity flow-based detailed router with efficient acceleration techniques[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2018, 37(1): 217-230.
- [39] CHO M, BAN Y, PAN D Z. Double patterning technology friendly detailed routing[C]. IEEE/ACM International Conference on Computer-Aided Design (ICCAD), San Jose, 2008: 506-511.
- [40] MIRSAEEDI M, RIEGER M L, TORRES J A, et al. Self-aligned double-patterning (SADP) friendly detailed routing[C]. Proceedings of SPIE-The International Society for Optical Engineering, 2011.
- [41] NIEBERG T. Gridless pin access in detailed routing[C]. DAC '11 Proceedings of the 48th Design Automation Conference, San Diego: ACM New York. 2011: 170-175.
- [42] XU X. PARR: Pin access planning and regular routing for self-aligned double patterning[J]. ACM Transactions on Design Automation of Electronic Systems, 2016, 21(3): 1-21.
- [43] LIU I J, FANG S Y, CHANG Y W. Overlay-aware detailed routing for self-aligned double patterning lithography using the cut process[C]. 2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC), ACM, 2016, 35(9): 1519-1531.
- [44] JIA X, CAI Y, ZHOU Q, et al. MCFRoute 2.0: A redundant via insertion enhanced concurrent detailed router[C]. 2016 International Great Lakes Symposium on VLSI, USA, 2016: 87-92.