

新型电路版图布局布线算法设计^{*}

袁 也,王 刚,刘晓光,李雨森

(1. 南开大学计算机学院,天津 300350;
2. 天津市网络与数据安全重点实验室(南开大学),天津 300350)

摘 要:调研了电路自动布局布线技术的国内外研究现状,在此基础上设计了一种面向中等规模电路布局布线算法,主要用于大型版图设计软件的模块测试环节,为用户提供各模块初步的布线布局结果,方便用户高效查找并修正错误点,填补了我国在相关领域的空白。建立了超图模型并转换为图模型,改进了 Stoer-Wagner 算法并利用该算法和 Fiduccia-Mattheyses 算法对图进行了基于最小割理论的划分,从而构建出一棵划分树。在这棵树的基础上设计了一种二元相对移动算法来确定各个电路元件的位置,大大降低了布局拥挤度,提高了美观度,对于数百元件的电路均能在 0.5 s 内得出布局结果。基于 A^{*} 算法在多个方面做了改进,提高了布线速度,对于线路数 1 000 以下的元件能在 0.1 s~60 s 内得出结果,实现了 100% 布通率以及均匀的布局布线效果。

关键词:电路设计自动化(EDA);Stoer-Wagner 算法;超图;模拟退火算法;A^{*} 算法

中图分类号:TP301

文献标志码:A

doi:10.3969/j.issn.1007-130X.2021.07.006

A novel circuit layout and routing algorithm

YUAN Ye, WANG Gang, LIU Xiao-guang, LI Yu-sen

(College of Computer Science, Nankai University, Tianjin 300350;

Tianjin Key Laboratory of Network and Data Security Technology(Nankai University), Tianjin 300350, China)

Abstract: This paper investigates the research status of automatic circuit layout and routing technology at home and abroad, and designs a layout and routing algorithm for medium-sized circuits. It is mainly used in the module test of large-scale layout design software, and provides users with preliminary wiring and layout results of each module, which is convenient for users to find and correct error points efficiently, and fills the gap in related fields in China. In this paper, a hypergraph model is established and transformed into a graph model. The Stoer-Wagner algorithm is improved, and the algorithm and the Fiduccia-Mattheyses algorithm are used to divide the graph based on the minimum cut theory, and construct a partition tree. Based on this tree, this paper designs a binary relative moving algorithm to determine the location of each circuit component, which greatly reduces the layout congestion and improves the aesthetics. For a circuit with hundreds of components, its layout can be obtained within 0.5 seconds. Based on the A^{*} algorithm, the routing is improved in many aspects, and the routing speed is improved. For a circuit with <1000 wires, the routing results can be obtained within 0.1 s to 60 s, achieving 100% routing rate and uniform layout and routing effect.

Key words: electroic design automation (EDA); Stoer-Wagner algorithm; hypergraph; simulated-annealing algorithm; A^{*} algorithm

^{*} 收稿日期:2020-06-30;修回日期:2020-11-09

基金项目:国家自然科学基金(U1833114,61872201,61702521);天津市人工智能重大专项(18ZXZNGX00140,18ZXZNGX00200)

通信作者:王刚(wgzwp@163.com)

通信地址:300350 天津市津南区南开大学计算机学院

Address: College of Computer Science, Nankai University, Jinnan District, Tianjin 300350, P. R. China

1 引言

发达国家在集成电路领域的高新技术成果方面始终对我国形成封锁。近年来,集成电路自动化设计技术高速发展,为了推动我国迈入该领域发展的快车道,越来越多的学者开始研究电路自动化设计的相关技术。

当前我国虽然在大规模电路的布局布线方面已有许多成果,但是对于中小规模电路的布局布线研究却很少,更多的是追求快速求得整体结果和解决大规模问题的卓越能力,而暂时忽略了软件商用时用户自主设计的便捷性和舒适性以及用户使用软件时一些更具体的需求。对此本文改进了划分算法,提升了划分结果质量,提出了全新的布局算法,满足对美观度的要求,提升了 A* 算法的布线速度,提高了布线质量,形成一套适用于中小规模模块的布局布线方案,方便用户在实际布线前检查电路各模块的逻辑错误,填补了相关研究领域的空白。

2 相关工作

目前电路布局算法主要有 2 类,一类是直接布局算法,另一类是面向划分的布局算法。直接布局算法大多采用传统启发式算法如模拟退火算法^[3,4]、遗传算法^[5,6]和粒子群优化算法^[7]等,是在一个初始布局的基础上通过尝试性改变的效果决定下一步迭代改进的方向,循环往复直至结果满足一定条件为止。这类算法的特点是布局效果较好,不易陷入局部最优,但算法运行速度慢、时间长。

随着电路元件规模达到数百甚至上千,这些元件组合成极其复杂的超图,直接布局难度较大。面向划分的布局算法将大规模的电路划分为很多小规模的部分再布局,有效降低了布局难度,还可使各个功能模块更加聚集,使得布局布线效果更好,缺点是部分算法的划分结果常常不均衡。

电路布线技术多是在 A* 寻路算法^[8]的基础上进行设计,该算法能够在网格化空间内有效寻找最短路径,缺点是运行速度慢。

在布局阶段本文改进了 Stoer-Wagner 算法^[9],弥补了其划分结果不均衡的缺点,同时使用 Fiduccia-Mattheyses 算法^[10]优化该算法的结果并形成一棵划分树,在划分树的基础上设计了二元相对移动算法,能够在兼顾布局拥挤度和布局空间大

小的同时以较低的时间复杂度完成布局。在布线阶段本文改进了 A* 寻路算法,提高了其运行速度,并重新设计了其搜索函数,满足了中小规模电路对较少的线路交叉、线路转折以及较低的布线拥挤度的需求。

3 布局算法设计

本文设计的布局布线算法面向中小规模电路,试图实现对元件的布局布线,区别于传统的物理设计,本文设计的算法强调运行速度和布局布线结果的美观性和逻辑性,而相对忽略布线空间的耗费。

在本文中布局算法的目标有 3 点:(1)各元件按照功能模块划分和聚集;(2)各部分之间的割值尽量小;(3)划分的各部分节点个数相差较小;(4)布局区域尽量小。

由于面向划分的布局算法布局均匀,且相关技术较为成熟,因而本文基于面向划分的布局算法进行设计,选择 Stoer-Wagner 算法生成初始划分,并对算法做了改进,避免出现不平衡的划分结果,然后选择 Fiduccia-Mattheyses 算法对初始划分做进一步改进。

3.1 模型构建

电路图是非常典型的超图模型,其中每条线路可能联结 2 个以上的电路元件。本文首先建立了超图模型,将数据读入一张表中,例如表 1,其中 a 、 b 、 c 代表超边,数字表示节点,表项表示超边与该节点联结的权重。

Table 1 Edge node matrix

表 1 边节点矩阵

	a	b	c
1	0	0	0
2	1	2	3
3	1	3	4
4	0	2	0
5	1	5	3
6	1	3	5
7	1	2	0

随后将度为 2 以上的超边,即联结了 2 个以上节点的超边转化为节点,每个超边转化成的节点与原超边关联的每一个节点都有一条度为 2 的超边,这样超图中则不存在度为 2 以上的超边,则超图转化为图,如图 1 所示。数据存储在邻接矩阵中,如表 2 所示,数字表示节点,表项表示节点间边的权重。之后的布局算法在此邻接矩阵形式的基础上进行设计。

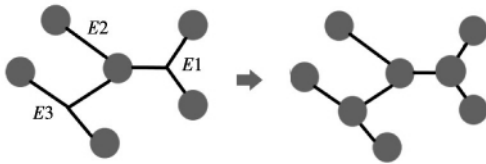


Figure 1 Transformation of hypergraph

图 1 超图的转化

Table 2 Adjacency matrix

表 2 邻接矩阵

	1	2	3	4	5	6	7
1	1	0	1	0	1	1	1
2					
3	...						
4							
5							
6							
7							

3.2 使用 Stoer-Wagner 算法获得初始划分

Stoer-Wagner 算法是用于搜索无向图的全局最小割的高效算法,普通的查找最小割的算法是 Ford-Fulkerson 算法,能在 $O(n^2m)$ 的时间复杂度内查找到将确定的 s,t 2 节点划分到 2 部分的最小割,其中 n 为节点数, m 为边数,在复杂网络下 m 甚至能达到 $O(n^2)$ 的复杂度,因此若使用 Ford-Fulkerson 算法查找全局最小割,复杂度可能会达到 $O(n^5)$,对于成百上千节点的情况,这样的时间消耗显然是无法接受的。Stoer-Wagner 算法能在 $O(n^3)$ 的时间复杂度内得出全局最小割,时间复杂度远远优于 Ford-Fulkerson 算法。

Stoer-Wagner 算法基于这样一种思想:对于图中任意 2 个节点,它们或者属于全局最小割的 2 个不同划分,或者属于同一个划分。如果是后者,那么合并这 2 个节点后并不影响全局最小割。

基于这种思路,若每次能求出将图中某 2 个节点 s,t 划分到 2 部分的最小割,记录下割值和划分后将这 2 个节点合并为 1 个节点,再求某 2 个节点的最小割,重复上述操作直至全部节点收缩为 1 个,此时记录中割值最小的划分就是所求的划分。

其中在每次求将 s,t 划分到 2 部分的最小割时基于一种贪心的策略:首先任选某一节点作为原点,从剩余节点中找出和原点相连的边权重最大的一个,暂时将其与原点合并,重复此操作直至除原点外仅剩余一个节点为止,这个节点和最后一个与原点合并的节点即为上述的 s,t ,该节点和原点当前的相连边权重即为求得的割值。

如图 2 所示节点 3 号和节点 4 号即为 s,t ,最小割值为 2。详细证明见参考文献[11]。

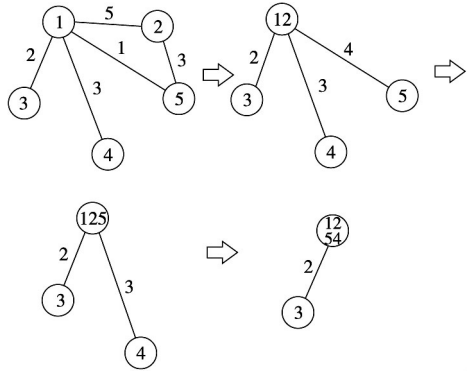


Figure 2 Stoer-Wagner algorithm

图 2 Stoer-Wagner 算法

Stoer-Wagner 算法能在 $O(n^3)$ 的时间复杂度内得出有效解,效果远远好于传统的最小割算法,但却很少应用于版图算法,原因在于:(1)时间复杂度较高,难以用于大规模电路设计;(2)Stoer-Wagner 算法求出的最小割是理论最小割,这样的结果往往忽视了划分的平衡要求,也就是说 Stoer-Wagner 算法常常会得出 1:99 的元件比例的划分结果,这样的结果显然不适合用于电路划分。

但是,本文主要研究中小规模电路版图,因而虽然 Stoer-Wagner 算法时间复杂度较高,但仍能在较短时间内获得结果。因此,本文对 Stoer-Wagner 算法进行修改使其能用于获得高质量的初始划分。为了兼顾平衡约束和割值最小,对算法进行如下修改:在每次选取与原点关联最多的节点并入原点时,若并入后满足平衡条件,则记录下此时原点中包含的节点以及原点不包含的节点作为划分结果。在算法运行结束时并不直接选择割值最小的划分,而是在记录的割值与划分结果中选择割值最小的划分作为初始划分。

3.3 使用 Fiduccia-Mattheyses 算法优化初始划分

Fiduccia-Mattheyses 算法的主要思想是:首先将图随机二等分,每次从 2 部分中选择 1 个节点划分到另一部分来最大程度地减小二划分的割值,重复此过程直至划分结果优于某一限度并且满足平衡条件。这里的平衡条件按式(1)计算:

$$n/2 \cdot (1 - p) < balance < n/2 \cdot (1 + p) \quad (1)$$

其中, p 为根据需求自定义的平衡参数,根据对运行效果和运行速度的需求进行调整。评价当前划分的优劣时既考虑使割值较小,又要保证划分出的 2 部分元件数量之差小于某一范围。

Fiduccia-Mattheyses 算法能在 $O(n^2)$ 的时间

复杂度和 $O(n)$ 的空间复杂度内对初始划分进行有效优化,优化结果见第 5 节。

3.4 二元相对移动算法

本文使用 Fiduccia-Mattheyses 算法和 Stoer-Wagner 算法将全部元件划分为一棵二叉划分树,提出了一种二元相对移动算法对划分树进行布局。

二元相对移动算法是在网格上布局元件并搜寻最优布局的算法,因此首先需对元件进行网格化建模,暂时忽略元件的电路表示形式,用数个网格构成的长方形代表元件。

在介绍算法的详细步骤之前先给出优度值的计算公式如式(2)所示:

$$gdegree = \alpha \cdot cd + (1 - \alpha) \cdot pl / maxl \quad (2)$$

其中, α 是根据具体算法对布线长度和布线拥挤度的相对侧重程度确定的参数; cd 为空间拥挤程度,是区域 $section(A, B)$ 内被占用方格占全部方格的比重,区域 $section(A, B)$ 是由 A, B 的 8 个顶点中位于最左上的顶点与最右下的顶点分别沿横向和纵向做延伸线所围成的区域,如图 3 所示; $maxl$ 表示元件间布线长度的最大值; pl 为元件之间布线长度的估计值,此处用元件中心坐标间的曼哈顿距离表示,如式(3)所示:

$$pl = |(Asx + Aex)/2 - (Bsx + Bex)/2| + |(Asy + Aey)/2 - (Bsy + Bey)/2| \quad (3)$$

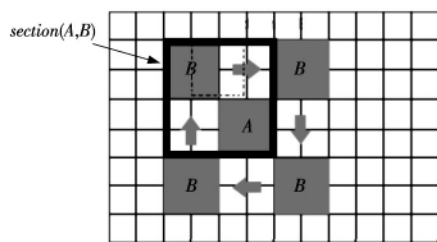


Figure 3 Moving process of binary moving algorithm

图 3 二元移动算法的移动过程

布局时先使用 Stoer-Wagner 算法和 Fiduccia-Mattheyses 算法组合成的划分算法对全部元件进行递归划分:先将元件划分为 2 个割值较小且相对平衡的部分,再对割出的 2 个部分运行此划分算法,重复此操作直至所有的元件均处于相互孤立的叶节点中,同时构建出一棵二叉划分树,如图 4 所示。

从二叉划分树的叶节点开始对任意 2 个兄弟叶节点的元件测试在不同相对位置下排布的优度,找出优度值最大的排布方式,按照此排布确定 2 个元件的相对位置,组合成一个新的元件返回给父节点作为父节点的元件,再对父节点及其兄弟节点重

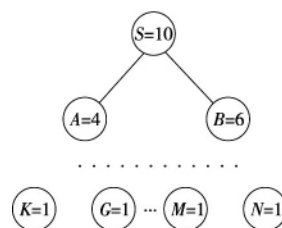


Figure 4 Partition tree

图 4 划分树

复上述操作直至返回根节点,则获得全部元件的整体布局情况。

二元相对移动算法存在一些不足,在运行二元相对移动算法时会导致布局空间的浪费,因为环绕排布的方式会导致元件间有空隙,因此本文在此基础上对布局空间进行自顶向下的规划,从根节点开始对左右子树根据其中的元件数和连线数设置矩形布局空间的长和宽的上限。在运行二元相对移动算法时不得越过上述界限。

此外在运行二元相对移动算法时,为了给布线留出足够的空间,在元件外层套上一层空白层后再移动位置,如图 5 所示,图中 $sectionAC$ 表示元件 A 及其外围包裹的空白网格构成的空间, $sectionBC$ 表示元件 B 及其外围包裹的空白网格构成的空间。

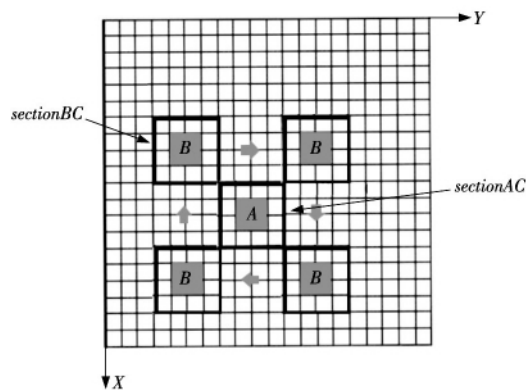


Figure 5 Extended A* algorithm

图 5 扩张后的 A* 算法

4 布线算法设计

本节设计并改进了布线算法,在第 1 节布局结果的基础上根据各个元件之间的联结关系使用 A* 寻路算法布置线路,随后针对 A* 算法的一些不足进行了改进。

4.1 A* 寻路算法

A* 算法的基本思想是在当前搜索空间的基础上选取较优方向的网格加入搜索空间,而不是将所

有方向的网格都加入搜索空间从而限制了搜索空间的膨胀,加快了搜索速度。

A* 算法的基本流程是:建立一个开表和闭表,先将起点网格放入开表,每次从开表中选取与终点曼哈顿距离值 f 最小的网格放入闭表,并将其周围的空白网格放入开表,重复此操作直至选取出的网格为终点网格为止,此时则搜索到了一条起点到终点的路径。其中若即将加入开表的空白网格已在开表中,则比较其原本前驱网格与当前选出网格与起点的距离,选取距离较小者作为其前驱网格。

4.2 A* 寻路算法改进

在面对成百上千的元件时,A* 算法的时间开销在数分钟到数十分钟之间,时间开销是影响 A* 算法实用性的重要因素。分析 4.1 节 A* 算法的流程可知,A* 算法的时间主要消耗在从开表中选出 f 值最小的网格,若使用排序算法查找最优网格则每次查找的时间复杂度为 $O(mNum \cdot \log mNum)$,其中 $mNum$ 为开表中网格数量。这里用 L 表示网格区域的边长,平均情况下的复杂度为 $O(L)$,开表搜索的运行次数也为 $O(L)$ 量级,则布线一次的时间复杂度平均情况下为 $O(L^2 \cdot \log L)$ 。因此,本文改进 A* 算法在选择最优 f 时不使用排序算法,而是维护一个最小堆,每次从堆顶直接取出 f 值最小的网格后再整堆,则布线一次的时间复杂度由 $O(L^2 \cdot \log L)$ 降为 $O(\log L)$ 。

传统 A* 算法的目标在于找到最短距离,但本文的布线需求除了距离较短外还追求美观度,而使用本网格到终点网格的曼哈顿距离来估计值,忽略了布线对布局拥挤程度的影响,因此在进行较大规模的布线运算时,较晚布置的线路会因为较早布置的线路对可布线网格的占用而无法布通。因此,为了兼顾距离最短、拥挤度和美观程度本文选择构造式(4)来满足本算法的需求:

$$f = (\mu \cdot c + (1 - \mu)d)dis \tag{4}$$

其中, μ 为一个小于 1 的参数; c 为本网格的拥挤程度,拥挤程度越大值越大,这样就使得走线时尽量选择拥挤度较低的方向,从而减小拥挤度,提高布通率。此处用此网格周围 5 层网格中障碍网格的占比表示:在图 6 中,浅色灰度网格的 c 值为黑色障碍网格数除以黑色方框区域内的网格数。 d 为网格类型惩罚值,为了保证布线结果的美观,应尽量避免出现线路交叉或转折,因此本文根据走线类型将网格分为 3 类,对 3 类不同的网格赋予不同的惩罚值,使得走线时尽量选择惩罚值较小的网格,从而保证美观程度。如图 7 所示,类型 1 的惩罚值

为 0.5,类型 2 的惩罚值为 1,类型 3 的惩罚值为 0.3。另外走线类型是根据网格与父网格、祖先网格的相对关系以及布线次数确定的:对于类型 1 的网格,本网格与父网格的父网格的横纵坐标都不相同;对于类型 2 的网格,是根据布线数为 2 这一特征确定网格类型的;对于类型 3 的网格,本网格与父网格的横纵坐标中有且仅有一对不相同。 dis 为本网格与终点网格的曼哈顿距离。

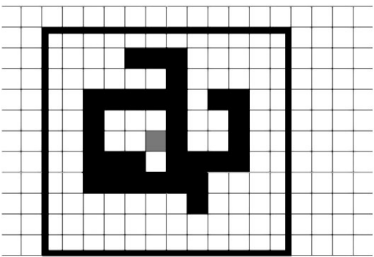


Figure 6 Calculation of c value

图 6 c 值的计算

经过上述改进,A* 算法时间复杂度降为 $O(L^2)$,布线结果更加美观,布通率大大提升。

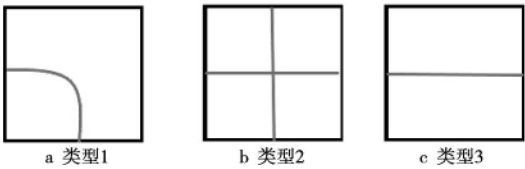


Figure 7 Grid types

图 7 网格类型

5 实验和结果分析

实验环境为如下配置的虚拟机:单核 CPU,主频 2.8 GHz,内存 1 GB,操作系统为 CentOS 6.4-64 bit。

本文使用 C++ 语言实现并测试整个布局布线算法,包括布局阶段的 Fiduccia-Mattheyses 算法、Stoer-Wagner 算法、二元相对移动算法、模拟退火算法和布线阶段 A* 算法的实现。电路信息如表 3 所示。

Table 3 Circuit information

表 3 电路信息

ID	节点数	元件数	超边数	原边数	线路数
0	35	32	3	51	160
4	32	24	8	23	122
6	98	64	34	66	320
8	52	20	32	117	198
11	36	28	8	20	200
16	332	320	12	332	1 280

5.1 布局算法测试

在表 4 中,括号中的数据是 3.4 节的改进方法实现前的划分算法和布局算法的效果,括号外的数据为改进后的算法效果。可以看出,改进后的算法运行速度提升了数倍到数十倍,拥挤度降低了数倍到数十倍,节点数越多提升越大。此外布局面积大幅减小,为布线算法降低了搜索难度。由表 4 可以看出,布局区域形状均匀,在保证拥挤度小于 0.5 的前提下布局区域较小。划分时间与节点数呈正相关,布局时间和线路数呈正相关,对于数百节点,上千条线路的电路图,本文使用的算法均能在 0.5 s 内得出划分和布局结果。

Table 4 Layout results

表 4 布局结果

线路数	布线时间/s	交叉数	转折数	拥挤度	布线总长度(网格数)
160	0.24 (19.33)	1 463 (1 856)	990 (1 044)	0.15 (0.20)	23 803 (26 843)
122	0.09 (3.62)	189 (284)	187 (146)	0.15 (0.15)	12 767 (16 647)
320	0.8 (74.44)	3 162 (3 622)	343 (358)	0.21 (0.183)	60 197 (106 795)
198	0.31 (34.3)	1 501 (1 622)	677 (785)	0.16 (0.23)	35 046 (39 468)
200	0.31 (24.3)	1 407 (1 752)	593 (654)	0.17 (0.22)	30 206 (41 254)
1 280	40.66 (>600)	46 712 (69 542)	19 057 (20 452)	0.35 (0.44)	462 524 (689 544)

5.2 布线算法测试

由表 5 可以看出,4.2 节的改进方法实现后的布线算法相对于改进前速度有了近百倍的提升,交叉数、转折数、拥挤度和布线总长度大幅缩小。本文提出的改进策略实现了较好的效果,在尽量保证较低拥挤度的前提下减少了布线总长度,使得布线总长度随着线路数的增加近似保持线性增长,同时尽量减少了交叉数和转折数,使得交叉数和转折数大约保持在布线总长度的 10%,保证了布局布线结果的美观性。此外表 5 中电路的布通率均为 100%。

在运行速度方面,本文算法在布线数小于 500 时,布线时间不超过 0.5 s;布线数超过 1 000 时布线时间大约需要 20 s~1 min,本文算法的运行速度能够满足数十到数百电路元件的布局布线需求。图 8 展示了电路 17 的最终布局布线效果。

6 结束语

本文在当前电路布局布线相关研究的基础上

Table 5 Wiring results

表 5 布线结果

节点数	划分时间/s	布局时间/s	拥挤度	布局区域(行*列)
35	<0.01 (<0.01)	<0.01 (1.22)	0.01 (0.268)	409*409 (433*414)
32	<0.01 (<0.01)	<0.01 (0.89)	0.006 7 (0.198)	295*295 (436*430)
98	0.01 (0.1)	0.01 (11.58)	0.009 4 (0.012)	541*541 (767*1020)
52	<0.01 (<0.01)	0.01 (0.15)	0.011 (0.12)	474*474 (689*874)
36	0.01 (0.05)	<0.01 (<0.01)	0.01 (0.04)	427*427 (644*684)
332	0.29 (3.12)	0.08 (2.4)	0.01 (0.05)	1160*1160 (2522*2436)

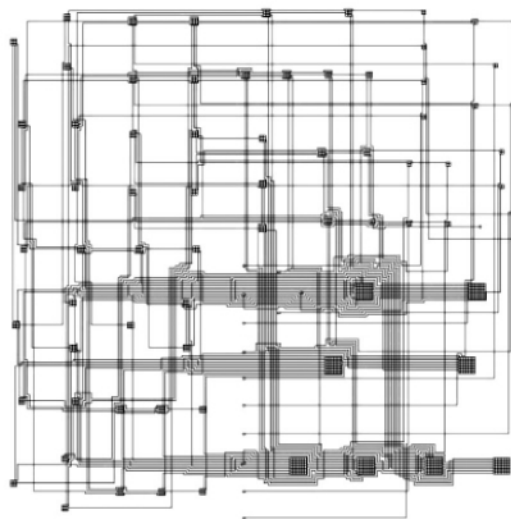


Figure 8 Circuit 17 layout and wiring effect exhibition

图 8 电路 17 布局布线效果展示

设计并实现了一个布局布线方案,用于满足大规模电路设计软件中用户对于中小规模模块的快速排查错误的需求。在设计布局算法时本文使用 Stoer-Wagner 算法生成初始划分,用 Fiduccia-Mattheyses 算法优化划分结果,提出并改进了二元相对移动算法生成布局结果。在设计布线算法时本文在 A* 寻路算法的基础上实现并优化了布线算法。整个设计的布局布线结果较为美观,在控制布局面积和布线总长度的前提下拥挤情况较为良好,元件和线路分布较为均匀,对于规模小于数百元件、数千线路数的电路能在 0.1 s~1 min 内得出结果,满足了用户对中等规模电路功能模块的快速定位和排查错误的需求。

参考文献:

- [1] Song Jun-de, Xin De-lu. Introduction to VLSI and system design [M]. Chengdu: University of Electronic Science and Technology of China Press, 1989. (in Chinese)

- [2] Chu Jing. Research on graph theory in VLSI routing[D]. Huainan: Anhui University of Science and Technology, 2017. (in Chinese)
- [3] Metropolis N, Rosenbluth A W, Rosenbluth M N, et al. Equation of state calculations by fast computing machines[J]. Journal of Chemical Physics, 1953, 21: 1087-1092.
- [4] Kirkpatrick S, Gelatt Jr C D, Vecchi M P. Optimization by simulated annealing[J]. Science, 1983, 220(4598): 671-680.
- [5] 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, 1968, 4(2): 100-107.
- [6] Yang M, Almaini A E A, Wang L, et al. FPGA placement using genetic algorithm with simulated annealing[C]//Proc of the 6th International Conference on ASIC, 2005: 808-811.
- [7] Liu An, Feng Jin-fu, Liang Xiao-long, et al. Algorithm of hardware/software partitioning based on genetic particle swarm optimization[J]. Journal of Computer-Aided Design and Computer Graphics, 2010, 22(6): 927-933. (in Chinese)
- [8] Gudise V G, Venayagamoorthy G K. FPGA placement and routing using particle swarm optimization[C]//Proc of the IEEE Computer Society Annual Symposium on ISVLSI, 2004: 307-308.
- [9] Stoer M, Wagner F. A simple min-cut algorithm[J]. Journal of the ACM, 1997, 44(4): 585-591.
- [10] Fiduccia C M, Mattheyses R M. A linear-time heuristic for improving network partitions[C]//Proc of the 19th Design Automation Conference, 1988: 175-181.
- [11] Xie Zhi-hong. Optimization and analysis of FPGA placement and routing algorithm[D]. Xi'an: Xidian University, 2012. (in Chinese)

附中文参考文献:

- [1] 宋俊德, 辛德禄. 超大规模集成电路与系统设计导论[M]. 成都: 电子科技大学出版社, 1989.
- [2] 褚静. 超大规模集成电路布线中的图论问题研究[D]. 淮南: 安徽理工大学, 2017.
- [7] 刘安, 冯金富, 梁晓龙, 等. 基于遗传粒子群优化的嵌入式系

统软硬件划分算法[J]. 计算机辅助设计与图形学学报, 2010, 22(6): 927-933.

- [11] 谢志宏. FPGA 布局布线算法的研究与优化[D]. 西安: 西安电子科技大学, 2012.

作者简介:



袁也(1997-), 男, 河北张家口人, 硕士生, 研究方向为电路布局布线。E-mail: 1637138484@qq.com

YUAN Ye, born in 1997, MS candidate, his research interest includes integrated circuit layout & routing.



王刚(1974-), 男, 北京人, 博士, 教授, 研究方向为海量信息存储、并行与分布式计算。E-mail: wgzwp@163.com

WANG Gang, born in 1974, PhD, professor, his research interests include massive information storage, parallel & distributed computing.



刘晓光(1974-), 男, 河北安国人, 博士, 教授, 研究方向为存储系统、搜索引擎和并行计算。E-mail: liuxg74@hotmail.com

LIU Xiao-guang, born in 1974, PhD, professor, his research interests include storage system, search engine, and parallel computing.



李雨森(1982-), 男, 辽宁朝阳人, 博士, 副教授, 研究方向为云计算和分布式系统。E-mail: liyusen@nbj1.nankai.edu.cn

LI Yu-sen, born in 1982, PhD, associate professor, his research interests include cloud computing, and distributed system.