

应用于大规模 ASIC 的线长驱动的合法化算法

高文超^{1,2} 周强² 钱旭¹ 蔡懿慈²

(中国矿业大学(北京)机电与信息工程学院计算机系 北京 100083)¹

(清华大学计算机科学与技术系 北京 100084)²

摘要 总体布局完成之后的核心任务就是单元位置的合法化,即在将所有的单元安放到布局区并且与合理位置(site)对齐的同时,消除单元间的重叠。为了高效地实现大规模 ASIC(Application Specific Integrated Circuit, 专用集成电路)的布局过程,提出一种基于线长驱动的合法化算法。它以电路总线长为优化目标,同时考虑单元布局合理位置的约束和预布障碍。在 ISPD'11 和 DAC'12 竞赛例子上进行的测试结果表明了算法在线长保护及优化方面的效果。这些测试用例都是来源于现代工业 ASIC 设计的实例,由此说明了算法可以稳定有效地解决工业界中大规模 ASIC 多种特征电路的布局合法化问题。

关键词 布局合法化,线长驱动,大规模,集成电路

中图分类号 TN47 **文献标识码** A

Wire-length Driven Legalization Algorithm for Large-scale ASIC

GAO Wen-chao^{1,2} ZHOU Qiang² QIAN Xu¹ CAI Yi-ci²

(School of Mechanical Electronic & Information Engineering, China University of Mining and Technology, Beijing 100083, China)¹

(Department of Computer Science & Technology, Tsinghua University, Beijing 100084, China)²

Abstract Legalization is core task of the detail placement after global distribution of generous cells. It removes cells overlap, aligns them to the sites and places them to their final position. A wire-length driven legalization algorithm was presented in this paper. The algorithm takes the total wire-length as objective, considers the site constraint and predefined blockage. Experiments on ISPD'11 and DAC'12 benchmarks show that this legalization algorithm can get good wire-length results. These test cases are derived from modern industrial ASIC design, which also testify the algorithm can effectively solve the varying characteristics circuit layout legalization issues of large-scale ASIC.

Keywords Placement legalization, Wire-length driven, Large-scale, ASIC

1 引言

近年来,ASIC 以其强大的功能和优良的特性迅速发展,受到世界范围内电子工程设计人员的极大青睐和普遍欢迎。

ASIC 物理设计中,布局起到重要作用,它的性能在很大程度上决定后续步骤的质量。现代布局工具通常把布局分为总体布局和详细布局两个子步骤。在总体布局中,电路元素通过优化定位得到一个总体合理的相对位置。详细布局由合法化和后优化过程组成,在总体布局结果的基础上,把所有单元的位置合法化到标准单元行内的单元位置(site)上的过程称为合法化。

D. Hill 等人^[1]提出的一种标准单元的合法化算法,为每个单元选择代价最小的行来安放。该算法比较简单,复杂度很低,很多其它布局算法在合法化过程中都采用该算法或者改进了该算法,比如 APlace^[43,66,72]等。

Fastplace^[3]通过模拟退火方法求得宏模块的合法解。之后为每一个标准单元选择最小代价区域安放。此算法比较耗时,只能针对宏单元数量较少的设计。

文献[4]通过求解线性方程来得到宏模块的最终位置。然后从左到右扫描标准单元,为它们寻找一个使线长最短的位置进行安放。而文献[5]提出一种最小代价流的动态规划求解算法。

根据当前工业 ASIC 的设计特点,本文提出了一种线长驱动的合法化布局算法,流程如图 1 所示。它在将单元放置到 site 上消除重叠的同时,最大限度地优化保持线长,主要策略体现在:

(1)布局格式化采用就近原则,能够达到最终格式化后相对单元初始位置总的移动距离最小,保证了线长增加的最小化。

(2)行间重分配时,用非法单元的移动量作为代价值,最后选择代价最小的行来安放单元,这样就使线长的变化量最小。

(3)行内消除重叠时,采用贪婪算法找到重叠区和空白区,使重叠单元在周围的空白区域内移动,消除重叠,这样能实现线长最小程度地增加。

实验使用的是 ISPD 2011^[6]和 DAC 2012^[7]布局竞赛的

到稿日期:2012-12-10 返修日期:2013-03-11 本文受国家自然科学基金(61176035,60833004)资助。

高文超(1986—),女,博士生,主要研究方向为集成电路电路布局算法,E-mail: conanayi@sina.com;周强(1961—),男,博士,教授,博士生导师,主要研究方向为物理设计中的布局、布图规划、面向 FPGA 平台的设计方法学。

例子,都是来源于实际工业 ASIC 设计。测试用例反映了现代 ASIC 设计的复杂性,包括含有许多布局障碍、更多的金属层以及金属层不同的宽度和间距等。本文第 2 节详细介绍合法化的算法步骤;第 3 节给出实验结果;最后给出结论和未来的工作展望。

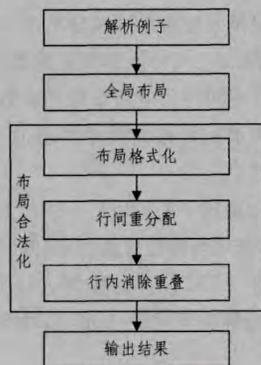


图 1 布局流程

2 布局合法化

2.1 布局格式化算法

在总体布局算法阶段,单元可以被安排在布局区内的任何地方,通常不会考虑行以及 site 的位置。但是在详细布局阶段,必须将单元的最终位置确定,所以单元都要跟行中的 site 对齐。将总体布局之后的单元安放到 site 上,这一过程称为布局格式化,单元总是跟 site 对齐,并且不会超出布局区。

本文中的算法采用就近原则来格式化单元,使线长不会过度增加。根据实际 ASIC 设计的例子,我们先将矩形和非矩形固定单元进行格式化,因为固定单元位置不变,只需完成 site 的标记。之后,遍历所有标准单元,找离单元最近的 site 安放。最先放到 site 上的单元都认为是合法的,并被存放到每行行内合法单元的数组中,它所占的所有 site 都被标记为已占用。之后若再有单元被格式化到已被占用的 site 上,就认为是不合法的,并被存放到每行不合法的数组中,以便接下来的消除重叠操作。

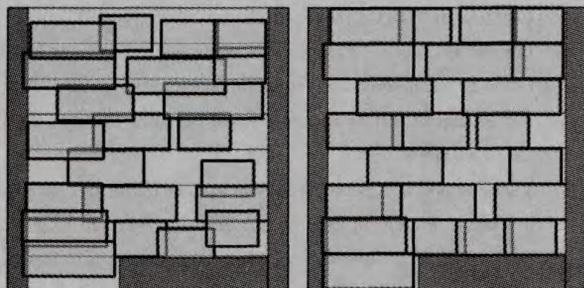


图 2 布局格式化

2.2 行间重分配算法

在将总体布局的结果进行就近安放,使得所有单元都位于布局区并与 site 对齐之后,在某些区域可能会有重叠发生。为了消除这些重叠,必须移动其中的一些单元,也就是上面我们存放到每行不合法的数组中的单元。

行间分配算法需要令布局区中的每一行均能够容纳下在该行内的单元。某一个行有 n 个 site,行内所有单元总共占用 m 个 site,如果 $n \geq m$,那么显然在这一行内就一定能够消

除重叠;否则必须移动一些单元到别的行中使得 m 最终小于等于 n 。

布局格式化后被列为不合法的单元要进行行间分配,放置到可以容纳它的行,同时使其移动量尽可能小,这样才能保证最小的线长增加量。每个非法单元所带来的移动量可以分成两部分,一部分是非法单元在进行再分配时自身的移动量,这一部分称为移动量;另一部分是非法单元再分配后同行内其它单元发生的一定的重叠。为了消除这些重叠,必须在行内执行一些单元的移动,避免由此带来的移动量重叠。这两部分其中之一有可能为零,但不可能两者都为零。

图 3 是一个非法单元再分配的例子,单元 cell 从最初的位置往上移动了两行,设移动量的权重是 w_1 ,重叠量的权重是 w_2 。由于单元 cell 仅在竖直方向上产生了移动,因此单元 cell 的移动量就是竖直方向的移动量,总的移动量是 $offset$;而对于重叠量,从图中可以看出,移动之后的位置和已有的单元发生了重叠,重叠量为 $overlap$ 。最后我们就得到了单元 cell 从当前位置移动到目标位置的总代价:

$$cost = w_1 \times offset + w_2 \times overlap \quad (1)$$

同理计算其它行的代价,最终我们为单元选择一个代价最小的行安放。

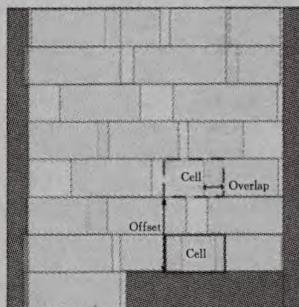


图 3 非法单元再分配

2.3 行内消除重叠算法

在行间分配算法完成之后,所有单元之间的重叠都认为只发生在行内,并且行一定可以容纳下行内的单元。行内消除重叠所需要完成的任务就是在行内消除单元之间的重叠,并且使得总的移动量最小。问题形式化描述如下:

行 R 中有 n 个标准单元 $c_i (1 \leq i \leq n)$, R 有左边界 $lower$ 和右边界 $upper$,每个单元 c_i 有初始位置 x_i 和宽度 w_i ,合法化之后的位置假定为 x_i' 。要求 R 中的所有单元位置顺序都确定,消除单元间的重叠,并且使得总的移动量最小。问题可以用如下公式来描述:

$$lower \leq x_1, x_i + w_i \leq x_{i+1}, x_n + w_n \leq upper \quad (2)$$

$$\text{且最小化 } \sum_{i=1}^n |x_i - x_i'|$$



图 4 行内单元消除重叠

行间重新分配后重叠只集中在行内,在被固定宏单元划分的每行内分段消除重叠,将重叠单元移到附近的空白区

域,以此来保证线长不会过大增加。我们使用贪婪算法找到重叠区域和空白区域,并按照单元的大小将其放到合适的空白区域内,以此消除所有重叠。消除完行内重叠后的单元如图4所示。

3 实验结果

算法用C++语言实现,运行在使用 Intel Xeon 3.0GHz CPU、6G内存的 Linux 服务器上。在实验中,随机产生总体布局结果,以此作为合法化程序的输入,以总体布局后的半周长线长和合法化后的半周长线长作为比较对象,如表1所列。第1列是测试用例,第2列是测试例子的可移动单元的数量,第3列是随机产生总体布局的线长结果,第4、5列是合法化算法得到的线长的结果及其所用时间。

表1 总体布局线长和合法化后线长的结果对比

测试例子	单元数目	总体布局结果(e9)	合法化后结果(e9)	运行时间(sec)
superblue1	847k	13.18	13.03	314.74
superblue2	1.01M	24.25	22.43	374.42
superblue4	500k	7.03	6.94	274.16
superblue5	772k	14.35	12.95	300.56
superblue10	1.13M	20.90	20.80	385.18
superblue12	1.29M	19.85	18.93	436.14
superblue15	1.12M	11.21	11.16	352.40
superblue18	483k	5.23	4.94	180.26
superblue2	0.92M	24.25	22.43	3397.9
superblue3	0.83M	14.41	13.81	1602.03
superblue6	0.91M	16.06	15.62	985.81
superblue7	1.27M	23.59	21.56	1858.35
superblue9	0.78M	11.89	11.52	798.02
superblue11	0.85M	20.63	19.77	1321.98
superblue12	1.27M	19.85	18.93	1518.03
superblue14	0.56M	7.74	7.49	320.72
superblue16	0.86M	9.17	8.89	550.11
superblue19	0.50M	6.50	6.31	420.69
平均		1.00	0.95	

从表中可以看出,经算法合法化后的总的线长比总体布局结果减少了5%。这说明了本文算法的可行性,实现了优化线长的目标。同时,经过 dac2012_check_legality^[8] 检查,证明算法得到的布局结果是合法的,单元之间不存在任何重叠,它们都放置到相应的 site 位置。

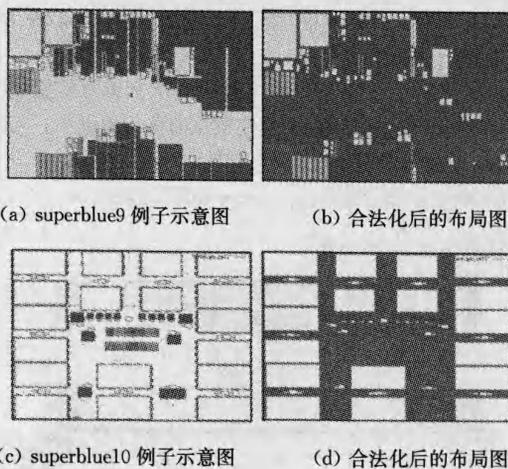


图5 布局结果示意图

图5是用 gnuplot^[9]画出的 superblue9 和 superblue10 两个例子及其合法后的布局图。例子中空心的模块框都是固定的矩形单元,而实心的模块都是固定的非矩形单元。superblue3 中所有的固定单元都集中在布局区域的边缘,这给布局工具在区域中心附近留下很大的自由空间。而 superblue10 中,固定单元把布局区域分割成多个子区域。这揭示了现代集成电路的不同特征和相应的挑战,同时也证明了本文算法对 ASIC 不同特征电路布局的有效性。superblue12 和 superblue7 所含有的可移动单元都超过了 100 万,这也测试出算法的稳定性。

结束语 本文提出了一种以最小线长为目标的合法化算法,其考虑了单元布局合理位置的约束,同时避开了预先的障碍。用来源于工业 ASIC 设计的实例进行了测试,结果说明本算法可以有效地解决大规模 ASIC 多种特征电路的布局合法化问题。

本文的进一步工作将在合法化之后的后优化方面做改进,以实现完整的详细布局流程,以及与当今主流的详细布局算法的全面质量比较。未来将扩展该模型,将约束直接反映到电路的实际约束,比如线长、可布性、拥挤度等。

参考文献

- [1] Hill D. Method and system for high speed detailed placement of cells within an integrated circuit design; US, 6370673[P]. <http://www.google.com.hk/patents/US6370673.pdf>, 2002-04
- [2] Kahng A, Reda S, Wang Q. Architecture and details of a high quality, large-scale analytical placer[A]//Proceedings of IEEE/ACM International Conference on Computer-Aided Design, 2005 [C]. California, USA, 2005; 890-897
- [3] Viswanathan N, Pan M, Chu C. FastPlace 2.0: An efficient analytical placer for mixed-mode designs[A]//Proceedings of Asia South Pacific Design Automation Conference, 2006[C]. Yokohama, Japan, 2006; 195-200
- [4] Cong J, Xie M. A robust detailed placement for mixed size IC designs[A]//Proceedings of Asia South Pacific Design Automatic Conference, 2006[C]. Yokohama, Japan, 2006; 188-194
- [5] Brenner U, Paulia, Vygen J. Almost optimum placement legalization by minimum cost flow and dynamic programming[A]//Proceedings of Intl. Symp. on Physical Design (ISPD), 2004[C]. USA, 2004; 2-8
- [6] Viswanathan N, Alpert C, Sze C, et al. The ISPD-2011 Routability-Driven Placement Contest and Benchmark Suite[A]//Proceedings of ACM International Symposium on Physical Design, 2011[C]. California, USA, 2011; 141-146
- [7] Viswanathan N, Alpert C, Sze C, et al. The DAC 2012 Routability-driven Placement Contest and Benchmark Suite[A]//Proceedings of Design Automation Conference, 2012[C]. California, USA, 2012; 774-782
- [8] http://archive.sigda.org/dac2012/contest/other_files/dac2012_check_legality.gz
- [9] <http://www.gnuplot.info/download.html>