

标准单元布局中的高效结群算法

吴为民¹,洪先龙¹,蔡懿慈¹,顾 钧²

(1. 清华大学计算机科学与技术系,北京 100084;2. 香港科技大学计算机科学系,香港)

摘 要: 本文提出了一个高效的结群算法,用以解决在超大规模条件下标准单元模式集成电路的布局问题.与传统结群算法相比,本算法的特色在于在结群之前,建立和组织了表示所有单元之间连接关系的信息库,使得结群算法具有全局优化性和无冗余计算的特点.本结群算法已应用于二次规划布局过程中.实验结果表明,本算法无论在结群质量和结群速度方面都非常理想,从而成功地解决了超大规模电路的布局问题.

关键词: 标准单元;结群;二次规划;布局

中图分类号: TP302

文献标识码: A

文章编号: 0372-2112 (2001) 02-0148-04

High Efficiency Clustering Algorithm for Standard Cell Placement

WU Wei-min¹, HONG Xian-long¹, CAI Yi-ci¹, GU Jun²

(1. Dept. of Computer Science & Technology, Tsinghua University, Beijing 100084, China;

2. Dept. of Computer Science, The Hong Kong University of Science & Technology, Hong Kong, China)

Abstract: A high-efficiency clustering algorithm is presented for solving standard cell style placement problem with very large number of cells. Compared to traditional clustering algorithms, the main feature of the presented algorithm is that an information library including all interconnect relationships between cells is built and well organized before placement, thereby endowing the clustering algorithm with characteristics of global optimality and non-redundancy computation. The clustering algorithm has been applied to a quadratic placement procedure. Experimental results show that our clustering algorithm is very perfect in both clustering quality and clustering speed, thus solves the placement problem with very large number of cells successfully.

Key words: standard cell; clustering; quadratic programming; placement

1 引言

结群是通过将连接紧密的单元结合在一起,作为一个整体参与布局,从而减小问题求解空间的过程.随着 VLSI 电路规模的不断增大,现有的布局算法越来越难以满足在布局时间上的要求,结群就成为一种有效的手段.目前,结群在基于模拟退火算法布局中应用较多^[1~3].

传统的结群算法,一般是先选择一些单元作为种子(seed),然后以各个 seed 作为原始群,不断结合进与其最紧密的单元.迄今的结群算法所采用的准则有多种,如 seed 群和候选单元之间的公共线网数,结合后局部化线网数,群尺寸等.对于以总线长优化为目标的标准单元模式布局中的结群问题,局部化线网数应是最优先考虑的因素.

Mallela 和 Gover 首先将结群技术引入到基于模拟退火的布局算法中,得到了很好的结果^[1].此方法首先将单元根据互连关系结群,并用传统模拟退火过程进行布局.然后,将群解散,再用另一个模拟退火过程对布局进行改善.对于大于 2500 个单元的电路,布局速度提高 2~3 倍,总线长也要好 6~

17%.

Yur-Wen Tsay 和 Youg-long Lin 在布局前进行一个提取连接紧密单元的过程^[2],他们将所提取出来的单元集合称为“cone”.在将小尺寸的“cone”进一步采用内外连接度法结群后,对形成的软模块做布图规划,最后再用一个模拟退火过程改善.此方法在线长、布线轨道数、布局时间方面都好于单纯采用模拟退火的布局算法(TimberWolf6.0).

Sun 和 Sechen 提出了一个分层的模拟退火布局方法^[3],其算法的层次性就体现在布局之前有一个结群过程.结群的引入使得在较高层次的群数量及互连线网数均大大降低,使得总的布局时间比非分层的布局时间要小一个数量级.

其它的结群方法还有 Cong 等人提出的随机步行法(random walk)^[4].这种算法的总时间复杂度为 $O(n^3)$, n 为单元数.较高的复杂度妨碍了算法的应用;Garbers 等人提出的基于图连接度的方法^[5],则常常导致非自然的结果,使连接松散的单元比连接紧密的单元更可能结到同一个群中.

对于结群算法有两个最基本的要求:一是结群过程必须足够快,使得其与布局时间相比可以忽略不计;二是布局质量

不应因结群而过度下降.但迄今的结群算法对于甚大规模电路却存在着很大缺陷,主要表现在:(1)对 seed 的选取比较盲目,无法从全局上考虑,以致影响结群质量.(2)在结群过程中需要大量计算单元之间连接度的信息,计算重复性相当大,以致严重地影响了结群速度.为提高结群质量和速度,专家们采用了各种手段.如文献[1]采用了多代价函数顺序满足和滤掉大线网的办法,而 J. Cong 则采用了并行结群方法^[6].

本文将提出一种新的结群方法.此方法对结群单元的选择是全局性的,而且在结群之前就预先计算出单元之间的连接度信息,不存在重复计算问题,从而可以提高结群速度并得到较好的结群质量.

2 结群算法

本文提出的结群算法首先将各个单元作为一个原始群,然后不断地选取两个群进行合并,并保证合并后的群不超过群的尺寸限度,直到再也找不到可合并的候选群为止.结群过程分为二个步骤:(1)建立单元连接信息库.(2)单元结群.

2.1 建立单元连接信息库

设单元集合为 C ,线网集合为 N .对任一单元对 (c_i, c_j) ,若 $n_{i,j}$ 表示连接此二单元的线网集合,则单元对的连接紧密度 $CON_1(c_i, c_j)$ 可如下式求得:

$$CON_1(c_i, c_j) = \sum_{n_{i,j}} w_k \quad (1)$$

式中, w_k 表示线网 n_k 权值,可以 $w_k = 1/t_k$ 求得, t_k 为 n_k 的端点数.

对所有 $CON_1(c_i, c_j)$ 必须有一个高效的计算方法.这里略去计算过程.将所有的 $CON_1(c_i, c_j)$ 值按大小分成 H 组.这样,可将 $CON_1(c_i, c_j)$ 数据存储到一个二维实数数组 BUCK $[m][n]$ 中.为保证无冗余存储,规定只有当 $i < j$ 时,单元对 $CON_1(c_i, c_j)$ 才是有效的. BUCK 数组的每个元素对应一个单元对, m 表示组序号, n 表示单元对序号.任一单元对 $CON_1(c_i, c_j)$ 的组序号 $m(c_i, c_j)$ 可如下式求得:

$$m(c_i, c_j) = \lfloor (C_{\max} - C_{\min}) / H \rfloor \quad (2)$$

式中, $C_{\max}$ 和 $C_{\min}$ 分别是所有单元对对紧密度的最大值和最小值.

形成 BUCK 数组后,对每一组单元对进行排序,确定每个单元对在组中的序号.排序的准则是按内外紧密度从大到小的顺序.内外紧密度定义为:

$$CON_2(c_i, c_j) = CON_1(c_i, c_j) / \left(\sum_{\substack{con(c_i, c_k) \neq 0 \\ con(c_j, c_k) \neq 0}} CON_1(c_i, c_k) + \sum_{\substack{con(c_i, c_k) \neq 0 \\ con(c_j, c_k) \neq 0}} CON_1(c_j, c_k) \right) \quad (3)$$

二维数组 BUCK 就是单元连接信息库.第一维索引表示 BUCK 的高度,每一固定高度代表一个紧密度的范围.高度索引越大,相应单元对就越紧密.对同一高度索引值,相应的单元对以第二维索引按内外紧密度从大到小排列.可见,单元连接信息库是按紧密度和内外紧密度两个准则生成和组织的.

需要说明的是,这里的单元包括核心单元(core cell)和引脚(PAD).PAD 参与结群是本结群算法的一个特色,也使单元之间的连接计算更加精确.比如两个核心单元若与同一个

PAD 都有很强的连接关系,则这两个单元能够通过此 PAD 结合在一起.

2.2 结群过程

设总单元数为 N^c ,欲结群数为 N^{clu} ,群内最多核心单元数为 $S_{\max}$,包含单元 c_i 的群为 $Clu(c_i)$,相应的群内核心单元数为 $S(Cluc(c_i))$, $DIS(PAD_i, PAD_j)$ 表示两个 PAD 之间的距离, PAD 之间的最大距离限度为 PADDIS,则结群过程的形式描述如图 1 所示.

结群过程结束后,可能会存在一些单元,它们之间无连接关系,但结成群后的群尺寸也不大于 $S_{\max}$,可根据实际情况将它们结在一起,以减少群数.但这并不影响结群质量.

Algorithm Cluster Combining

Input: Original netlist $PN = (C, N)$, BUCK, $S_{\max}$.

Output: A clustered netlist $CN = (C_c, N_c)$.

Begin

Create an initial cluster netlist, with each cluster has only one cell;

Set index = 0;

for ($i = H$ to 0)

if (index = $N^c - N^{clu}$) break;

for (each cell pair (c_i, c_j) at the Buck height i)

if (CHECK(c_i, c_j) = UNCLUSTABLE)

continue;

else

merge $Clu(c_i)$ and $Clu(c_j)$ together;

index++;

if (index = $N^c - N^{clu}$) break;

endfor;

endfor;

Remove all clusters in C_c which have no core cell;

Remove all PADs from clusters in C_c .

Create N_c from N ;

Return $CN = (C_c, N_c)$

Procedure CHECK

Input: c_i, c_j

Output: CLUSTABLE or UNCLUSTERBLE.

Begin

if $Clu(c_i) = Clu(c_j)$

Return UNCLUSTABLE;

if ($S(Cluc(c_i)) + S(Cluc(c_j)) > S_{\max}$)

Return UNCLUSTABLE;

for each PAD_m in $Clu(c_i)$

for each PAD_n in $Clu(c_j)$

if $DIS(PAD_m, PAD_n) > PADDIS$

Return UNCLUSTABLE;

Return CLUSTABLE;

图 1 结群算法

3 算法分析

3.1 算法复杂性

理想情况下,结群次数为 $N^m = N^c - N^{clu}$.实际上,一般在

未达到 N^m 时,已找不到可合并的候选群对.算法复杂性主要体现在建立 BUCK 步骤.设对任意单元,其所连接单元数的上限为 L ,则计算单元连接信息的复杂度为 $O(N^c \cdot L)$.形成 BUCK 的复杂性主要体现在对每一高度单元对的排序上,其复杂度为 $O(H^2 N^q \log_2 N^q)$,其中 H^2 为单元对数大于零的高度数, $N^q = \max\{N^{(i)} \mid i = 0, \dots, H\}$, $N^{(i)}$ 表示高度为 i 的单元对数. N^q 随 H 增大而减小. L 一般较小,大线网可滤去.

3.2 算法优化性

依赖于 BUCK,结群算法每次都是选取结合最紧密的单元对结群.以紧密度为准则来看,结群算法可得到全局最优解.布局算法要求的结群准则是局部化线网数.那么,紧密度与局部化线网数是否有一致的对应关系呢?由式(1),紧密度是由线网权值累加得到的.由于定义线网权值为其端点数的倒数,故端点数越少,线网权值越大,紧密度越大.同样,端点数越少,单元合并就越有可能使线网局部化.可见,紧密度与局部化线网数有着一致对应的关系.结群算法在局部化线网数的准则下也具有全局优化性.

4 基于结群的标准单元模式布局

迄今,见诸于文献的结群技术都用于模拟退火布局算法中.模拟退火布局算法是从一个初始布局开始,在一个离散的解空间进行随机搜索的过程.结群技术的引入,限制了搜索区域,提高了布局速度,从而能够进行更多次的寻优操作.因此,结群技术能够提高布局速度和质量是显而易见的.

同以往研究工作不同的是,本文的结群算法是应用于二次规划的布局算法中.布局过程分为三步:(1)结群.(2)总体布局.(3)详细布局.

总体布局算法确定单元在芯片上的大致位置.原问题可表示成一个典型的二次规划问题,并采用拉格朗日松弛法求解(请参见文献[7]).求解过程转化到连续空间进行.详细布局确定单元在芯片上的实际位置.此问题可表示成一个线性分配问题来解决(见文献[8]).求解过程返回到离散空间进行.

显然,将结群技术引入到二次规划布局算法能够减少总体布局问题的自变量数,从而大幅度地提高布局速度.但是,自变量数的减少也意味着自由度的减少,总体布局的质量会因此受到影响.那么,在详细布局阶段能否纠正这种影响甚至进一步提高布局质量是基于结群的二次规划布局算法是否可行的关键.这也是本文研究的重要意义.

5 实验结果及分析

用几个 MCNC 典型电路对算法进行了测试,这些电路包括 primary1, primary2, C7, s13207, avq 等,电路参数见表 1.通过对测试结果的分析,我们发现了类似的规律.

首先,以较大的电路 C7, s13207, avq 为例,对测试结果(见表 2,表 3,表 4)各项进行如下分析:

考察表 2~4 的前三项,显然,随着群尺寸的增大,结群数和全局线网数减少,而局部化线网数增大.而且,从不结群到结群,三项指标表现出较剧烈的变化.随着群尺寸的等量增

大,三项指标的变化逐渐平稳.这说明结群尺寸宜适当选择.若选择过大,则增加的局部化线网数有限.而且,大的群尺寸会有负面影响,这是因为总体布局后还须将同一群中的单元打散,减小重叠性,以利于详细布局.由于打散过程是局部性的,不可能象总体布局那样做到全局优化.

表 2~4 的结群时间这一项充分说明了结群方法的高效性.首先,结群时间对结群尺寸的变化不敏感.这是因为结群算法的速度主要取决于建立单元连接信息库的速度.对于任何特定的电路,建立单元连接信息库的过程都是相同的;其次,随着电路规模的增大,结群时间呈现出近乎线性的增长.将各测试结果的结群时间按各群尺寸的结果取平均值,再除以相应测试电路的单元数总数(core cell, PAD),得值分别为: 1.24×10^{-4} , 1.21×10^{-4} , 1.32×10^{-4} ,基本上变化不大;最后,结群时间与布局时间相比微乎其微.将各测试电路的平均结群时间分别除以相应的总布局时间的平均值,得值分别为: 0.011, 0.011, 0.006.

表 2~4 的总体布局时间这一项反映出通过结群达到了提高布局速度的目的.总体布局时间随结群尺寸的增加而剧烈减少,尤其是从不结群到结群时.以 avq 为例,结群后总体布局时间按结群尺寸增加分别仅占不结群时总体布局时间的 37%, 25%, 22%.详细布局由于采用了相同的迭代参数,故其消耗时间不受结群尺寸的影响,仅与特定电路有关.

表 1 MCNC 标准测试例子

电路	引脚数	单元数	线网数
Primary1	81	752	937
Primary2	106	2907	3082
C7	315	2150	2465
s13207	1490	4267	5757
Avq	64	21854	22183

表 2 对 C7 的测试结果

群尺寸 \ 结群结果	1	2	4	6
结群数	2150	1234	684	476
全局线网数	2356	1896	1512	1335
局部化线网数	109	569	953	1130
结群时间(s)	0.29	0.31	0.34	0.27
总体布局时间(s)	7.80	4.05	1.94	1.51
详细布局时间(s)	26.59	26.47	26.97	26.57
总线长(m)	1.72	1.65	1.66	1.67

表 3 对 s13207 的测试结果

群尺寸 \ 结群结果	1	3	6	9
结群数	4267	1863	1041	761
全局线网数	4953	3635	2888	2630
局部化线网数	804	2122	2869	3127
结群时间(s)	0.65	0.71	0.73	0.65
总体布局时间(s)	21.16	6.78	3.41	2.67
详细布局时间(s)	57.23	57.70	57.32	57.35
总线长(m)	6.42	6.26	6.30	6.30

表 2~4 的总线长这一项的结果表明结群算法满足了对

布局质量的要求.由表中可见总线长没有因结群而过度增大,反而还有所下降,尤其是从不结群到结群时.以 avq 为例,结群后总线长按结群尺寸增加分别改进了 12.7%、4.3%、1%.

表 4 对 avq 的测试结果

群尺寸 结群结果	1	4	8	12
结群数	21854	7848	7722	7121
全局线网数	22116	15139	13241	12666
局部化线网数	67	7044	8942	9517
结群时间(s)	2.67	2.86	2.83	3.01
总体布局时间(s)	651.40	240.01	164.88	141.86
详细布局时间(s)	330.48	331.33	333.90	330.29
总线长(m)	10.67	9.32	10.21	10.56

最后,以电路 primary1,primary2,avq 为例,将本文的算法 CARGO (Clustering Algorithm Relies on Global Optimization) 与 Cone^[2]及 TimberWolf7.0^[3]作一横向比较.由于采用的布局条件不同,故在布局质量(总线长)上不具有可比性.这里仅比较一下结群对布局质量的影响及布局时间,见表 5.

表 5 结群后总线长降低的百分率及布局时间比较

电路	算法	Cone	Timber-Wolf7.0	CARGO
Primary1	线长降低率	6.2 %	- 2.4 %	3.5 %
	布局时间(s)	423	221	20
Primary2	线长降低率	4.5 %	- 2 %	2.9
	布局时间(s)	2531	1252	117
Avq	线长降低率	2.6 %	12 %	12.7 %
	布局时间(s)	63934	13018	571

对于电路 primary1 和 primary2, TimberWolf7.0 使布局质量有所降低. Cone 虽然使布局质量提高较大,但 CARGO 的布局时间仅是 Cone 的 4.7%和 4.6%.对于最大的 avq 电路, CARGO 使布局质量提高最大,而其布局时间却分别仅是 Cone 和 TimberWolf7.0 的 0.9%和 4.4%.

以上的实验结果表明,本文提出的结群算法及其在二次规划布局中的应用是相当成功的.

6 结论

本文提出一个高效的结群算法,并将其应用于二次规划布局中.其高效性得益于它的几个显著特点.首先,它是一个构造性的,而非叠代改善性的算法;其次,与传统算法在每次结群前才对每一个 seed 群寻找最佳候选群相比,它在最初就建立了 BUCK,作为全局性的信息,并考虑了 PAD 在结群中的作用.最后,由于每次都是在 BUCK 中选取两个连接最紧密的

单元结群,使得结群算法具有全局优化性.

由于二次规划布局算法不仅可以得到理论上的全局最优解,而且布局时间远小于模拟退火算法的布局时间,因此,将结群技术应用于二次规划布局算法更有意义.本文的研究工作为解决超大规模电路布局问题提供了一种有效的途径.

参考文献:

- [1] S. Mallela and L. K. Gover. Clustering based simulated annealing for standard cell placement [A]. Proc. 25th DAC [C], 1988:313 - 317.
- [2] Y. W. Tsay and Y. L. Lin. A row-based cell placement method that utilizes circuit structural properties [J]. IEEE Trans CAD, 1995, 14(3): 393 - 397.
- [3] W. J. Sun and C. Sechen. Efficient and effective placement for very large circuits [J]. IEEE Trans. on CAD, 1995, 14(3): 349 - 359.
- [4] J. Cong, L. Hagen and A. Kahng. Random walks for circuit clustering [A]. Proc. IEEE Intl. Conf. on ASIC [C], 1991: 14. 2. 1 - 14. 2. 4.
- [5] J. Garbers, H. J. Prommel and A. Steger. Finding clusters in VLSI circuits [A]. Proc. Int. Conf. on CAD [C], 1990: 520 - 523.
- [6] J. Cong and M. Smith. A parallel bottom-up algorithm with applications to circuit partitioning in VLSI design [A]. Proc. 30th Design Automation Workshop [C], 1993: 755 - 760.
- [7] 孔天明, 洪先龙, 乔长阁. VEAP: 一个基于总体优化的 VLSI 布局算法 [J]. 半导体学报, 1997, 18(9): 692 - 700.
- [8] 孔天明. 高性能、高可靠性的超大规模集成电路物理版图算法研究 [D]. 清华大学博士论文, 1997, 6.

作者简介:



吴为民 1966 年生. 1989 年毕业于吉林大学计算机科学系, 1992 年在哈尔滨工程大学获硕士学位, 1995 年在哈尔滨工业大学获博士学位. 1996 年~1998 年在浙江大学做博士后. 现在清华大学计算机科学与技术系做博士后. 主要研究领域是超大规模集成电路版图设计.



洪先龙 1940 年出生于中国浙江省. 1964 年毕业于清华大学数学力学系, 现任清华大学计算机科学与技术系教授, 博士生导师. 多年来从事 IC - CAD 的研究和教学工作, 主要研究领域是版图算法和 VLSI CAD 系统.