

Single-Sequence的边界约束条件

李康^{1,2}, 虞厥邦¹, 于永斌¹

(1. 电子科技大学电子工程学院 成都 610054; 2. 四川交通职业技术学院 成都 611130)

【摘要】在VLSI物理设计中, 分层设计和连线优化都要求某些模块放置在布局的边界位置。该文针对一般的具有不可二划分结构的布图规划问题, 在SS编码的基础上解决VLSI物理设计中有边界约束的布局布图规划的问题; 证明SS的放置顺序是表示模块的数字在SS中出现的位置先后顺序; 提出模块放置在四个边界(上、下、左、右边界)在SS编码中应满足的充要条件及证明; 并给出模块位于四个边界在SS编码中相应的表达式和计算方法。

关键词 边界约束; 布图规划; 布局; VLSI物理设计
中图分类号 TP391.7 **文献标识码** A

Boundary Constraints Using Single-Sequence Representation

LI Kang^{1,2}, YU Jue-bang¹, YU Yong-bin¹

(1. School of Electric Engineering, University of Electronic Science and Technology of China Chengdu 610054;
2. Sichuan Vocational and Technical College of Communication Chengdu 611130)

Abstract In practice of floorplan/placement of very large scale integration (VLSI) physical design, it is very critical to place some modules along the boundaries of the chip so that connections between inputs and outputs and among units in hierarchical design mode are shortened. Based on non-slicing representation single-sequence (SS), boundary constraints in VLSI layout design are solved. The packing sequence of a SS is proved to be the appearance sequence of integer, which represents module in a SS code. Further, a necessary and sufficient condition of a module to be placed on four boundaries (top, bottom, left, and right) in a SS code is proposed and proved.

Key words boundary constraint; floorplan; placement; VLSI physical design

随着电子技术的发展, 电路的规模与设计复杂度迅猛增加, 使作为VLSI物理设计核心的布局布图规划变得越来越重要。布局/布图规划是把一组模块没有重叠地放置在芯片上, 并使一定的目标得到优化, 对后续阶段的设计和生产会产生很大的影响。所以需要一种高效、灵活的编码来描述芯片上模块之间的几何位置和拓扑结构, 以便于有效地处理各种有约束的布局布图规划问题。在各种布局/布图规划的约束中, 边界约束是指把一些模块放置到指定的边界位置。在实际设计中, 把一些模块安置在指定的边界位置, 便于这些模块与芯片的输入输出接点连接, 缩短输入输出之间的连线; 同时, 在当前大型电路的分层设计中, 电路被分割成无数小的和可处理的单元电路。把单元电路中与其他单元电路有联系的模块放置在指定边界, 可以有效地减小连线, 提高性能。所以, 在VLSI设计中利用一个优秀的布局/布图规划编码来有效地处理模块的边界放

置问题有十分重要的理论和现实意义。

在超大规模集成电路 (very large scale integration, VLSI) 物理设计的布局布图规划的发展过程中, 布局布图规划被划分成可划分结构和不可划分结构两类。由于大多数的优化解是不可划分结构, 所以更多的研究集中在不可划分结构的编码上, 具有代表性的有SP^[1]、BSG^[2]、O-tree^[3]、B*-tree^[4]、CBL^[5]等。文献[6]提出了一种非常优秀的SS编码 (single-sequence)。SS的编码只是连续的自然数1, 2, ..., n中各个数字的一个排列。通过每一个代表物体的数字在SS中的相对位置以及大小关系, SS编码串就可以表达这些模块、区域或任何有边界的物体在平面上的几何位置关系——上、下、左、右(ABLR)的关系。SS编码是目前布局布图规划表示方法中最为简单而有效的一种。

文献[7]研究了用可划分结构实现边界约束的问题。对于不可划分结构的边界约束布局布图规划问

题, 很多学者利用各种编码进行了研究, 如文献[8]通过加入惩罚函数的方法, 用CBL编码实现有边界约束的布局布图规划问题, 但算法收敛较慢; 文献[9]利用SP编码, 通过把一个不可行的有边界约束的编码转换成一个可行的编码来实现; 文献[10-11]使用B*-Tree和O-tree编码实现了有边界约束的布局布图规划问题。

本文用布局布图规划编码SS来处理边界约束问题。由于SS的简单和高效的特征, 它能够很好地处理边界约束问题。

1 问题的定义

给定一组模块 $M=\{m_1, m_2, \dots, m_n\}$ 。其中, 每个模块 $m_i(1 \leq i \leq n)$ 有固定的宽 w_i 、高 h_i 和面积 A_i , 每个模块可以自由旋转。 F 表示可自由放置模块的集合, $T(L, B, R)$ 表示被指定放置在上(左、底、右)边界的模块集合。 M 由五个不相交的子集合 F, T, B, L 和 R 构成, $M=F \cup T \cup B \cup L \cup R$ 。 (x_i, y_i) 表示模块 m_i 左下角的坐标值。一个布局就是为每个模块分配一组坐标值 (x_i, y_i) , 使任意两个模块在芯片上不重叠, 并使有边界约束的模块安放在指定的边界。相应的目标函数是使所有模块占用的面积(外围长方形的面积)最小。连线长度的优化在文中没有涉及, 但可以通过把它加入算法的目标函数很容易地实现。

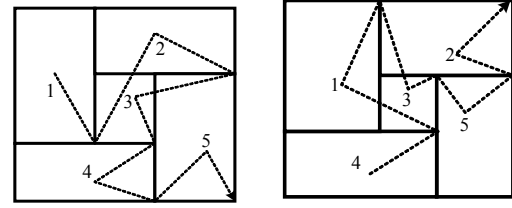
2 SS编码简介

文献[11]提出的SS编码, 仅用 $1 \sim n$ 的连续整数的一个排列来表示平面上一个布局或布图规划的拓扑结构。SS中所包含的ABLR关系由每个代表模块或区域的数字在SS中的相对大小和位置, 由以下规则确定:

(1) SS确定ABLR关系。假设在SS编码中两个整数 x 和 y 保持一种位置关系: 当且仅当 $x < y$ 时, x 所代表的模块(或区域)在 y 所代表的模块(或区域)的左边; 当且仅当 $x > y$ 时, x 所代表的模块(或区域)在 y 所代表的模块(或区域)的下边。

(2) SS编码是目前表示物体位置关系最简单的数据结构。由定义可知, 它是平面上物体之间的ABLR位置关系的一个产生器。考虑平面上的 n 个物体, 一个SS编码定义了 $n(n-1)/2$ 个ABLR关系。SS编码最直接的应用是模块的放置, 即在满足由SS编码导出的ABLR关系条件下, 找到包围所有模块且面积最小的外围长方形。

SS编码的标识和获得过程如图1所示。



a. 区域标识序列(1,2,3,4,5)

b. SS编码(4,1,3,2,5)

图1 区域标识序列

3 有边界约束的SS

为了便于讨论有边界约束的SS, 本文引入以下定义:

定义 1 POS(k)表示在SS编码中整数 $k(k \in N)$ 的位置。

定义 2 $SS_b(k)$ 表示由SS中所有位置在 k 前面的元素(数字)构成的子集合; $SS_a(k)$ 表示由SS中所有位置在 k 后面的元素(数字)构成的子集合。图1b中, $SS=41352$, $POS(1)=2$, $POS(2)=5$ 。

定理 1 在从左下角开始的模块放置方式中, 一个给定SS编码 $SS=k_1, k_2, \dots, k_n, k \in N$ 的放置顺序是每个代表模块的数字在SS中位置的先后顺序。

证明 因为模块的放置是在第一象限内从左下角开始, 所以正在放置的模块必须位于所有没有被放置的模块的左边或下面, 否则无法满足由SS编码给定的ABLR位置关系。从定义1以及“左下角开始的放置”的内涵, 可以推论出对于任意给定的两个模块, 如果 $POS(k_i) < POS(k_j), i, j=1, 2, \dots, n, i \neq j$, 那么模块 m_i 应该在模块 m_j 之前放置, 否则无法满足SS编码给出的位置关系。所以在从左下角开始的模块放置方式中, 一个给定SS编码 $SS=k_1, k_2, \dots, k_n, k \in N$ 的放置顺序是每个代表模块的数字在SS出现的先后顺序。

定义 3 在一个SS编码序列中, $samc(k)$ 表示在SS中位置在 k 之后, 比 k 小并且最靠近 k 的整数; $bamc(k)$ 表示在SS中位置在 k 之后, 比 k 大并且最靠近 k 的整数; $bbmc(k)$ 表示在SS中位置在 k 之前, 比 k 大并且最靠近 k 的整数; $sbmc(k)$ 表示在SS中位置在 k 之前, 比 k 小并且最靠近 k 的整数。

定义 4 当从左向右扫描一个SS编码时, $DDSS(p, q)$ 表示由SS中元素构成的一个序列, 该序列中的每个元素是其前面元素的 $samc$ 值。类似地, 从左向右扫描一个SS编码时, $DISS(s, t)$ 表示由SS中元素构成的一个序列, 该序列中的每个元素是其前面元素的 $bamc$ 值。

$DDSS(p, q) = \{k_i | k_i = samc(k_{i-1}), k_i \in SS_a(k_{i-1}), i=2, 3,$

$\dots, j\} = \{p, \text{samc}(p), \text{samc}(\text{samc}(p)), \dots, q\}$, $\text{DISS}(s, t) = \{k_i | k_i = \text{bamc}(k_{i-1}), k_i \in \text{SS}_a(k_{i-1}), i=2, 3, \dots, j\} = \{s, \text{bamc}(s), \text{bamc}(\text{bamc}(s)), \dots, t\}$ 。

定义 5 在一个SS编码序列中, $\text{minbl}(k)$ 表示在 $\text{SS}_b(k)$ 中所有大于 k 的元素中最小的元素; $\text{maxas}(k)$ 表示在 $\text{SS}_a(k)$ 中所有小于 k 的元素中最大的元素。

定义 6 当从右向左扫描一个SS编码时, $\text{RDISS}(p, q)$ 表示由SS中元素构成的一个序列, 该序列中的每个元素是其前面元素的**bbmc**值。类似的, 从右向左扫描一个SS编码时, 定义 $\text{RDDSS}(s, t)$ 表示由SS中元素构成的一个序列, 该序列中的每个元素是其前面元素的**sbmc**值。

$\text{RDISS}(p, q) = \{k_i | k_i = \text{bbmc}(k_{i-1}), k_i \in \text{SS}_b(k_{i-1}), i=2, 3, \dots, j\} = \{p, \text{bbmc}(p), \text{bbmc}(\text{bbmc}(p)), \dots, q\}$, $\text{RDDSS}(s, t) = \{k_i | k_i = \text{sbmc}(k_{i-1}), k_i \in \text{SS}_b(k_{i-1}), i=2, 3, \dots, j\} = \{s, \text{sbmc}(s), \text{sbmc}(\text{sbmc}(s)), \dots, t\}$ 。

引理 1 (1) 在SS中的一个数字对应的模块位于左边界, 当且仅当在SS中没有数字位于其前面并且小于该数字。(2) 在SS中的一个数字对应的模块位于底边界, 当且仅当在SS中没有数字位于其前面并且大于该数字。(3) 在SS中的一个数字对应的模块位于右边界, 当且仅当在SS中没有数字位于其后面并且大于该数字。(4) 在SS中的一个数字对应的模块位于上边界, 当且仅当在SS中没有数字位于其后面并且小于该数字。

以下给出引理1中, 在SS中满足左边界约束的充要条件的证明, 其他边界条件的证明类似。

证明 (1) 充分条件: 从SS的定义及性质可知, 对于SS中的某一数字 k , 如果没有数字位于前面, 并且小于它($\forall q \in \text{SS}_b(k), q < k$), 按照SS编码所产生的ABLR关系, 没有模块位于它的左边, 所以, 它应该放置在左边界处, 以满足由SS编码所产生的ABLR关系。(2) 必要条件: 如果数字 k 对应的模块位于左边界处, 那么属于 $\text{SS}_b(k)$ 的元素都应该大于 k ; 否则, 如果存在任意的 $p, p \in \text{SS}_b(k), p < k$ 。按照“SS确定的ABLR关系”原则, 与前提条件矛盾。如果 k 在左边界处, 属于 $\text{SS}_b(k)$ 的元素要么都应该大于 k , 要么不存在小于 k 的数字。证毕

为了方便表述, 本文用 $i_{B-L}, i_{B-R}, i_{T-L}, i_{T-R}$ 来分别表示SS中对应左下角、右下角、左上角、右上角模块的数字。从引理1可以得出, $i_{T-L}=1, i_{B-R}=n$ 。

引理1给出的只是理论上SS边界约束条件, 本文给出在SS编码中四个边界约束的数字序列形式的具体表达式, 易于在程序中用简单的数据结构来实现。

定理 2 在SS中各边界约束的充分必要条件分别为: (1) 下边界约束条件: 在SS中与位于最终布局底边界的模块对应的数字是SS中定义的集合 $\text{DISS}(i_{B-L}, i_{B-R})$ 中的元素, 数字所代表的模块从左到右的放置顺序是数字在集合 $\text{DISS}(i_{B-L}, i_{B-R})$ 中的先后位置。(2) 左边界约束条件: 在SS中与位于最终布局左边界模块对应的数字是SS中定义的集合 $\text{DDSS}(i_{B-L}, i_{T-L})$ 中的元素, 数字所代表的模块从下到上的放置顺序是数字在集合 $\text{DDSS}(i_{B-L}, i_{T-L})$ 中的先后位置。(3) 右边界约束条件: 在SS中与位于最终布局右边界模块对应的数字是SS中定义的集合 $\text{RDISS}(i_{T-R}, i_{B-R})$ 中的元素, 数字所代表的模块从上到下的放置顺序是数字在集合 $\text{RDISS}(i_{T-R}, i_{B-R})$ 中的先后位置。(4) 上边界约束条件: 在SS中与位于最终布局上边界模块对应的数字是在SS中定义的集合 $\text{RDDSS}(i_{T-R}, i_{T-L})$ 的元素, 数字所代表的模块从左到右的放置顺序是数字在集合 $\text{RDDSS}(i_{T-R}, i_{T-L})$ 中的先后位置。

因为位置的关系, 本文只给出在SS中左边界约束充要条件具体表达式的证明。

证明 (1) 充分条件: 从SS的性质可知, i_{B-L} 和 i_{T-L} 分别表示在SS中代表左下角和左上角的区域位置的数字, 都位于左边界。 $\text{DDSS}(i_{B-L}, i_{T-L}) = \{i_{B-L}, \text{samc}(i_{B-L}), \text{samc}(\text{samc}(i_{B-L})), \dots, i_{T-L}\}$ 。从定义可知, 对于任意的 k , 如果 $k \in \text{DDSS}(i_{B-L}, i_{T-L})$, 不存在 $t \in \text{SS}_b(k), t < k$; 否则 $k \notin \text{DDSS}(i_{B-L}, i_{T-L})$ 。所以对于任意在 $\text{DDSS}(i_{B-L}, i_{T-L})$ 中的数字, 它所代表的模块必然位于最终布局的左边界。(2) 必要条件: 从引理1可知, 在SS中与位于左边界区域或模块对应的数字 $k, \forall t \in \text{SS}_b(k), t > k$ 。从 samc 和 minbl 的定义可以得到: $\text{samc}(\text{minbl}(k)) = k$, 并且 $\text{minbl}(\text{minbl}(\dots, (\text{minbl}(k)))) = i_{B-L}$; 否则SS中存在位于 k 和 $\text{minbl}(k)$ 之间的某个数字, 它所代表的区域位于 k 所代表的区域的左边, 与假设和 minbl 的定义矛盾。所以任何SS中与最终布局左边界约束相对应的数字必然属于子集序列 $\text{DDSS}(i_{B-L}, i_{T-L})$ 。

4 结束语

本文讨论了有边界约束的SS布局布图规划问题, 并以定理的形式给出了具体的表达式。因为SS是一种优秀的布局布图规划编码, 边界约束的表达式也以数字串的形式出现, 非常易于编程实现。SS相对于其他编码出现得较晚, 但它是其他编码的综合表述^[6]。正因为如此, 目前对其研究还不完全,

今后将继续研究SS表示的其他布局约束,如“有预先放置模块的约束布局”、“对称约束布局”、“模块对齐约束的布局”等,并且将用程序实现SS的有边界约束的布局。

参 考 文 献

- [1] MURATA H, FUJIYOSHI K, NAKATAKE S, et al. Rectangle-packing based module placement[C]// Proceedings of the 1995 IEEE/ACM International Conference on Computer-aided Design. San Jose, California, USA: IEEE, 1995: 472-479.
- [2] NAKATAKE S, FUJIYOSHI K, MURATA H, et al. Module placement on BSG-structure and IC layout applications[C]//Proceedings of the 1996 IEEE/ACM International Conference on Computer-aided Design. San Jose, California, USA: IEEE, 1997: 484-491.
- [3] GUO P N, CHENG C K, YOSHIMURA T, et al. An O-tree representation of non-slicing floorplan and its applications[C]//Proceedings of the 36th ACM/IEEE Conference on Design Automation. New Orleans, Louisiana, USA: IEEE, 1999: 268-273.
- [4] CHANG Y C, CHANG Y W, WU G M, et al. B*-Trees: A new representation for non-slicing floorplan[C]// Proceedings of the 2000 IEEE/ACM International Conference On Computer-aided Design. San Jose, California, USA: IEEE, 2000: 458-463.
- [5] HONG X, HUANG G, CAI Y, et al. Corner block list: An effective and efficient topological representation of non-slicing floorplan[C]//Proceedings of the 2000 IEEE/ACM International Conference on Computer-aided Design. San Jose, California, USA: IEEE, 2000: 8-12.
- [6] KAJITANI Y. Theory of placement by numDAG related with single-sequence, SP, BSG, and O-Tree[C]//Proceedings of 2006 IEEE International Symposium on Circuit and System. [S. l.]: IEEE, 2006: 4-7.
- [7] OUNG F Y, WONG D F, YANG H H. Slicing floorplans with boundary constraint[J]. IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems, 1999, 18(9): 1385-1389.
- [8] MA Y, DONG S, HONG X, et al. VLSI floorplanning with boundary constraints based on corner block list[C]// Proceedings of the 2001 Conference on Asia South Pacific Design Automation. Yokohama, Japan: [s.n.], 2001: 509-514.
- [9] AI J, LIN M, WANG T, et al. Module placement with boundary constraints using the sequence-pair representation[C]//Proceedings of the 2001 Conference on Asia South Pacific Design Automation. Yokohama, Japan: [s.n.], 2001: 515-520.
- [10] LIN J M, YI H E, CHANG Y W. Module placement with boundary constraints using B*-trees[J]. IEE Proceedings Circuits Devices and Systems, 2002, 149(4): 251-256.
- [11] LIU R, HONG X L, DONG S Q, et al. Module placement with boundary constraints using O-tree representation[C]// Proceedings of the 2002 IEEE International Symposium on Circuit and Systems. Phoenix-Scottsdale, AZ, USA: IEEE, 2002: 871-874.

编辑 黄莘

(上接第67页)

- [6] MUSUNURI S, CHAPMAN P L, ZOU J, et al. Inductor design for monolithic DC-DC converters[C]//IEEE 34th Annual Power Electronics Specialist Conference. Piscataway, USA: IEEE, 2003: 227-232.
- [7] RICHELLI A, COLALONGO L, QUARANTELLI, et al. A fully integrated inductor-based 1.8-6 V step-up converter[J]. IEEE Journal of Solid-State Circuits, 2004, 39(1): 242-245.
- [8] MUSUNURI S, CHAPMAN P L, ZOU J, et al. Design issues for monolithic DC-DC converters[J]. IEEE Transactions on Power Electronics, 2005, 20(3): 639-649.
- [9] GENG Li, LI Qing-hua, SHAO Zhi-biao. A new design strategy for the monolithic buck converters[C]//In Proc IEEE 4th International Power Electronics and Motion Control Conference. Piscataway, USA: IEEE, 2004: 192-196.
- [10] MOHAN S S, HERSHENSON M M, BOYD S P, et al. Simple accurate expressions for planar spiral inductances[J]. IEEE Journal of Solid-State Circuits, 1999, 34(10): 1419-1424.
- [11] LI Qing-hua, GENG Li, SHAO Zhi-biao. Optimum double-layer spiral inductor on silicon substrate designed for monolithic buck converters[C]//In Proc of 17th Asia-Pacific Microwave Conference. Piscataway, USA: IEEE, 2005: 2156-2159.

编辑 黄莘