

面向 VLSI 版图复用技术的二维 层次式压缩算法

赵子健 蔡懿慈 洪先龙 黄松珏 刘毅 谢民

(清华大学计算机科学与技术系, 北京 100084)

摘要: 面对 VLSI 生产工艺的不断更新, 利用已有的版图, 迅速获得适应新工艺的新版图, 已成为市场上实际的需求. 提出的基于约束图的压缩算法, 是面向全芯片压缩的二维压缩算法. 它采用层次式压缩策略, “落叶池”等新的数据结构, 在压缩过程中放松模块间的连线, 具有自动加入拐弯的功能. 从两个例子的压缩结果, 可以看出这是一个实用的新压缩算法.

关键词: 压缩算法; VLSI; 版图复用

EEACC: 7410D; 5120

中图分类号: TN402 **文献标识码:** A **文章编号:** 0253-4177(2000)08-0822-05

2-D Hierarchical Compaction Algorithm for VLSI Layout Reuse*

ZHAO Zi-jian, CAI Yi-ci, HONG Xian-long, HUANG Song-jue,
LIU Yi and XIE Min

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

Received 4 June 1999, revised manuscript received 21 September 1999

Abstract: Due to the unbalance between the development of EDA system and IC process technique, the layout reuse technique is demanded on the market. The existing layout should be compacted to fit for the new process technique. An efficient 2-D compaction algorithm is proposed, which is based on the constraint graph expression of the figures on layout and newly designed rules. In order to do compaction on the chip level, a hierarchical strategy and the ‘fall leaves pool’ data structure are used, regarding the interconnections as soft lines it also can add to jogs automatically. From the test cases it is shown that this compaction algorithm is practical and effective.

ZHAO Zi-jian(赵子健) male, born in 1973. Master, working with development of design automatic software.

CAI Yi-ci(蔡懿慈) female, born in 1960. Associate Professor, working with teach and research of CAD of VLSI.

HONG Xian-long(洪先龙) male, born in 1940. Professor, working with teach and research of computer design automation.

1999-06-04 收到, 1999-09-21 定稿

Key words: compaction algorithm; VLSI; layout reuse

EEACC: 7410D; 5120

Article ID: 0253-4177(2000)08-0822-05

1 引言

当今, 集成电路(Integrated Circuit)的功能越来越复杂, 规模也越来越大. 这一发展, 有赖于 IC 设计与工艺水平的提高. 电子设计自动化(EDA)系统已取得了长足的发展, 相对于工艺水平的提高而言, IC 设计水平的发展速度尚不能令人满意. 面对工艺的不断变更, 设计者用现有的工具和方法, 要在设计上跟上工艺发展的步伐是十分困难的. 一方面, 用户不可能随意放弃市场上有生命力的产品. 由于工艺改变而重新设计版图, 将耗费大量的金钱与时间, 也会延误产品投放市场的时间. 另一方面, 在设计一个新产品时, 也不可能设计芯片中的每个部分, 而往往利用已有的一些模块. 因此, 如何利用已有的版图, 或版图中的部分模块, 快速的实现从旧的生产工艺向新的生产工艺的转化, 即如何在工艺变化的情况下, 复用原来的版图, 是厂家面临的一个实际问题, 也是市场的需要. 版图复用技术(Layout Reuse Technique)是在物理设计的最低层上实现的, 即直接面向版图上的图形来进行操作, 而不是重新由上层做起. 所以版图复用技术的目标, 是在保留原有版图各模块间的拓扑关系等基本特征和电学性能的基础上, 压缩工艺变化后版图上出现的冗余空间, 对版图上部分结构进行局部调整, 必要时, 改变连线的路径, 以达到压缩版图面积的目的. 与以前的符号法版图设计中的图形压缩相比, 版图复用技术采用的压缩算法有其自身的特点和要求. 首先, 它的操作对象是全芯片或芯片中的某个独立模块, 一般来说, 要求处理的版图规模很大. 所以, 在数据量和操作时间上对算法的要求比较严格. 其次, 由于不希望对原来的版图作太大的变动, 而全芯片范围内图形之间的拓扑关系又很复杂, 在压缩时要充分考虑版图元素在水平与垂直两个方向上的位置关系, 只在一个方向上进行, 或在两个方向上交替进行的一维压缩算法是不能采用的. 压缩的概念最早由 Akers、Geyer 和 Robert^[1]在 1970 年提出. Schlay、Liao 和 Wong^[5]证明了二维压缩是一个 NP-Complete 问题. 早期的压缩算法都采用了分治策略, 压缩在两个方向上交替进行, 本质上都是一维压缩算法. 如虚拟网格法(Virtual Grid Approach)和剪切线方法(Shear Line Approach). 显然, 压缩效果与方向分解的顺序有一定的关系. FLOSS^[2]、CABBAGE^[4]、SLIP^[3]和 MULGA^[6]都是这方面较成功的代表. 1975 年, 美国加州大学 Berkeley 分校的 Hsueh 发表了约束图方法, 成为压缩方法中的成功代表, 引起了这一领域研究的深入发展. 该方法在现存的算法中占主要地位, 取得较好的结果. 它比其他的压缩模型更具有灵活性, 并能高效地实现. 本文提出的, 就是一种基于约束图表示的二维压缩算法.

2 基于约束图表示的二维压缩算法描述

约束图方法的基本思想是将设计约束转化为一个等价的约束图, 通过求解这个约束图, 从而得到版图元素的位置. 约束图 $G = (V, \vec{E})$ 包含结点集合 $V = \{v_0, v_1, \dots, v_n\}$ 和约束边集合 $\vec{E} = \{\vec{e}_1, \vec{e}_2, \dots, \vec{e}_m\}$. 其中, $\vec{e}_k = (v_i, v_j, \text{value})$ 为由结点 v_i 指向结点 v_j 的有向边, 边权值是 value. 结点集合 V 中每一结点 v_i 对应于版图中一个独立的图形元素, 这个元素就是在压缩时的一个操作对象. 约束图中各结点的连接关系则反映了原版图中各图形间的拓扑关系. 如果版图中两个相邻的图形元素 v_i 和 v_j 之间有设计规则上的要求, 我们就认为它们之间存在着约束关系. 并在约束图中结点 v_i 和 v_j 之间加入约束边 $\vec{e}_k = (v_i, v_j, \text{value})$. 在设计规则上存在着两种约束关系, 即 $x_i - x_j \leq \text{value}$ 或 $x_i - x_j \geq \text{value}$. 对前者而言, $\vec{e}_k = (v_j, v_i, \text{value})$, 后者则为 $\vec{e}_k = (v_i, v_j, \text{value})$. value 由新工艺的设计规则所决定. 可见, 约束图中包含了原有版图的拓扑结构和新的设计规则的信息. 通过解这个约束图, 我们就可以获得既保留了旧版图的基本结构特征, 又满足新的设计规则的新版图. 基于约束图表示的压缩分为三步进行, 它们是: 约束图生成、约束图求解和版图元素重新安置. 首先, 要找出版图上各图形元素间的约束关系. 根据约束不等式集合, 我们可以方便地建立约束图. 为了实

现二维压缩,在寻找某一模块的约束关系时,必须在其原有尺寸的基础上,根据版图拓扑关系,把搜索范围向外进行一定的扩展,寻找对角线约束关系.通过求解约束图,可以得到约束图中每个节点与版图左边界和下边界的相对位置.而源、漏间的距离 $(x_n - x_0)$, $(y_n - y_0)$ 即为版图的最小宽度和最小高度.求解约束图的方法很多,对无回路的约束图求解问题,可采用最长路径法来求解.若约束图中含有回路,求解算法要复杂得多^[7].求解约束图后,我们得到了版图元素的最小可行位置,但有大量元素有一个可移动的范围,即它在适当区域内的移动不破坏设计规则.这时,还要根据这个范围,采用一定的策略来确定元素的最终位置.

3 算法实现

因为本算法所面临的问题背景与以往的系统有很大的差别,一定要能适应大数据量,二维压缩的要求.基于对版图和设计规则的约束图表示,本文提出了一种层次式的二维压缩算法,运用了新的数据结构和方法,在处理对象和算法的实现上与以往的压缩系统有很大的区别.以下是算法的具体实现.

3.1 层次式的压缩算法

在已有的压缩算法中,约束图 $G = (V, \vec{E})$ 中结点集合表示的是版图上图形的边集.以这一思路处理全芯片的压缩问题,数据量将十分巨大,是不能接受的.本算法采用层次式压缩的策略,把全芯片的压缩问题分为三步来处理.首先,对 V 进行结群处理,令 $V = (V_1, V_2, \dots, V_p)$,其中 $V_r = \{v_{r1}, v_{r2}, \dots, v_{rp}\}$,为有内在联系的边的子集.随后,在每个边集 V_r 内部进行压缩.最后,再以边子集 V_r 为一独立元素,对 V 进行全芯片的总体压缩.结群时所依据的原则是,若边 v_i 和 v_j 满足以下条件之一,将被归入一个相同的子集中.

(1) 若 diffusion 为组成一闭合有源区的边集, nolv, contact, metal 是电学上与 diffusion 直接相连的多晶硅层, 连线引出孔和落在 diffusion 范围中的金属连线图形的边集. $V_r = \{\text{diffusion, poly, contact, metal}\}$, $v_i, v_j \in V_r$.

(2) 若 via 是两走线层 layer1 和 layer2 之间的通孔图形, metal1 和 metal2 是两走线层上对 via 的金属覆盖的图形边集. $V_r = \{\text{via, metal1, metal2}\}$, $v_i, v_j \in V_r$.

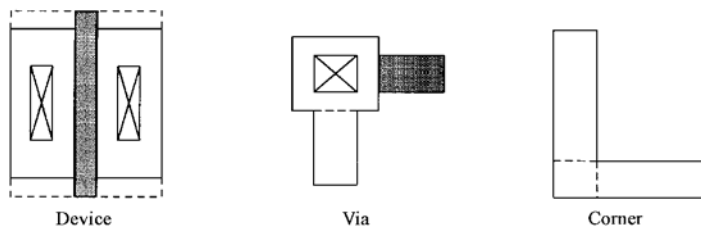


图 1 层次式压缩算法的处理对象

FIG. 1 Objects Handled by Hierarchical Compaction Algorithm

(3) 若版图上同层的互连线段 segment1 和 segment2 之间存在着覆盖部分, $V_r = \text{segment1} \cap \text{segment2}$, $v_i, v_j \in V_r$.

层次式压缩算法的处理对象包括: 通孔、拐角及器件子集(图 1). 算法不再把连线作为硬模块放在数据中参与压缩, 而代之以连线间的拐角, 使得压缩时数据量大大减少. 提取连线的拐角作为压缩对象, 使压缩效率和灵活性大大增加, 也保持了连线的基本位置, 为最后定位阶段恢复互连关系提供依据.

3.2 快速建立约束图的“落叶池”算法

已有的建立约束图的算法, 在寻找某一图形的约束关系时, 是把所有其它元素遍历一次, 以判断它是否与当前节点存在约束关系, 这一操作的时间复杂度为 $O(n^2)$, n 为版图上的节点数. 若采用扫描线方法, 算法复杂度也达到 $O(n \log n)$. 在实际版图中, 与某一元素存在约束关系的图形只是在它周围的一部分, 远离这个元素且相互之间已被其他元素阻隔的元素是不与之存在约束关系的. 因此, 如何确定有效的搜索范围是提高建立约束图速度的关键问题. 本文提出一种称之为“落叶池”的快速建立约束图方法. 它类似于叶子落下后相互覆盖的情形. “落叶池”的大小与芯片某一边的长度一致, 是一个一维的存储区域. 如建立垂直约束关系时与水平边大小一致, 建立水平约束关系时与垂直边长度相等. “落叶池”的作用是记录边界每

一个位置上,其投影没有被覆盖的元素.以图 2 说明.

版图上有 A, B, C, D 四个元素和一根走线,在建立垂直约束关系时,版图上的元素是按照先 X 坐标,后 Y 坐标的次序排好,依次读入,即 A → B → C → D.在读入 A 后,由于“落叶池”中没有记录任何模块信息,无需建立约束关系.同样把 B 读入.读入 C 时,在其投影位置上,已有对元素 B 的记录,所以建立起 C 和 B 之间的约束关系.对横穿芯片的走线,还要在它所穿越的位置上记

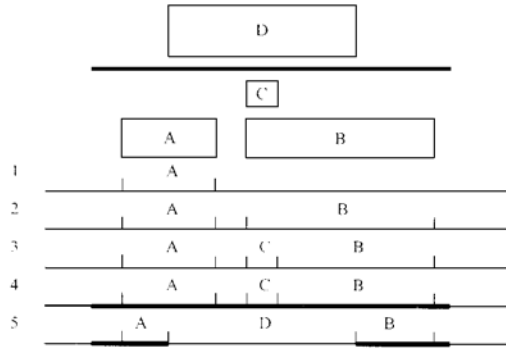
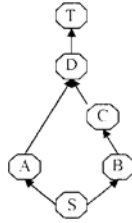


图 2 “落叶池”结构和算法

FIG. 2 Structure and Algorithm of Fall Leaves Pool

录下来,以便建立约束关系时考虑其所占据的空间的影响.最后读入的元素是 D.在它的投影所覆盖的区域上,有 B 和 C 两个元素,但 B 和 D 之间的约束关系能通过 D → C, C → B 两个约束关系反映出来,所以无需建立.可见,采用了“落叶池”这种记录方式后,能在与版图元素数目无关的时间里找到一个元素的约束关系,而且能判断某一约束关系是否冗余,从而决定是否建立.采用这一方法,能在 $O(n)$ 的时间复杂度内,建立起无冗余的简单约束图.

3.3 求解约束图的最长路径法

目前已有的压缩系统在求解约束图时,由于图中存在回路,必须使用改进的最长路径法来求解,时间复杂度为 $O(n(b+1))$.其中 b 是约束图中反向约束边的数目, n 是结点数.本算法采用了层次式压缩的策略,在进行模块的全芯片压缩时,处理的对象都是外部约束关系,而导致回路出现的内部约束关系都在模块内部的压缩中处理了.根据这一特点,算法在做模块一级的压缩时,可以得到一个有向无环图,即 b 为 0.这使得我们可以采用最简单的最长路径法来求解约束图,时间复杂度为 $O(n)$.

3.4 处理对角约束

二维压缩与一维压缩的主要差别在于对对角线约束的不同处理方式.如版图上存在着下面的两个元素 A 和 B,它们之间有对角线约束关系.元素 A 在阴影区域内移动都是合法的,因此在由 \vec{h} 和 \vec{v} 合成的对

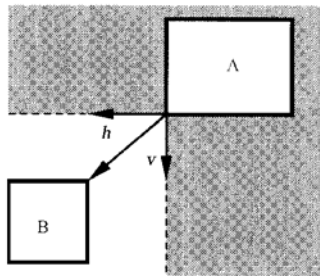


图 3 对角线约束关系

FIG. 3 Constraint Relation for Diagonal

角线约束关系中,可以删去二者中的一个.在一维方法中,删去哪一个是由压缩的次序所决定的.在二维压缩中,则根据压缩效果来决定删去哪一个.假设版图上对角线约束的数目与元素数目相仿,为 n .要得到一个全局最优解,就需要对每个对角线约束关系进行试探,因此时间复杂度为 $O(2^n)$.可见全局优化问题是一个 NP-Complete 问题.在进行二维压缩时,采用启发式算法来降低问题的复杂性.本算法在处理对角线约束时,独立地考虑每个元素所拥有的对角线约束关系,而保持其它的无关对角线约束状态不变(图 3).在这些相邻的对角线约束中,找出局部最优的解,然后再处理下一节点的对角线约束.这种方法的时间复杂度为 $2^r \times O(n^2)$.其中, r 为每个节点所拥有的对角线约束的数目,它与版图元素数目无关.

3.5 元素定位及自动加入拐角

在以往的算法中,连线是作为硬模块处理的.在压缩时保持原有状态.而连线的拐角往往是为了改善版图质量而有意识地弯折某根连线所引入的.在我们的压缩算法中,连线是软的.也就是说,在移动大模块

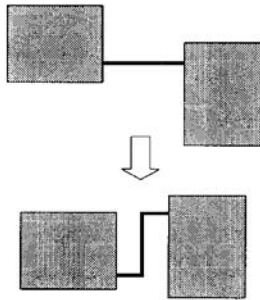
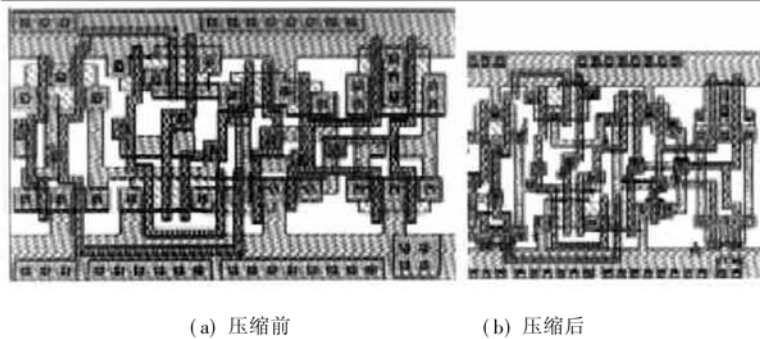


图 4 自动加入拐角
FIG. 4 Add Corner

时,可以不受与之相连的连线的影响,压缩效率大大提高.同时,为了保持原有线的连接性,算法必须有在必要的地方自动加入拐角的能力.而且这些拐角并非人为地加入,而是由模块压缩的结果自动决定,这样版图压缩效果更加合理自然,也更加接近原始版图的拓扑结构.由此可见,该算法比以往算法要复杂得多,也更具有智能性(图 4).

4 测试结果与结论

该算法已用 C 语言在 Sun Sparc20 上实现,并对某些例子进行了测试.图 5 是一个标准单元模块压缩前后的版图比较,表 1 是另外几个例子的测试结果.从运行结果可以看出,本算法的压缩效果是令人满意的.



(a) 压缩前 (b) 压缩后

图 5 版图比较

FIG. 5 Comparison for Layout

表 1 测试结果

Table 1 Test Results

电路	模块数	减少 面积/%	运行 时间/s
JK0526	318	24	7
JK11	1566	9.5	100
JK10	1583	11.2	101
JK7	2218	10.1	117
JK5	3124	10.4	144

参 考 文 献

- [1] S. B. Akers, J. M. Geyer and D. L. Roberts, "C Mask Layout with a Single Conduct Layer", Proc. 7th DAC, 1970, 7—16.
- [2] Y. E. Cho, A. J. Korenjak and D. E. Stockton, "FLOSS: An Approach to Automated Layout for High-Volume Designs", Proc. 14th DAC, 1977, 138—141.
- [3] A. E. Dunlop, "SLIP: Symbolic Layout of Integrated Circuit with Compaction", CAD, 1978, 10(6): 595—602.
- [4] M. Y. Hsueh, "Symbolic Layout and Compaction of Integrated Circuits", Ph. D. Thesis, University of California at Berkeley, Dec., 1979.
- [5] M. Schlag, Y. Z. Liao and C. K. Wong, "An Algorithm for Optimal Two-Dimensional Compaction of VLSI Layouts", Integration, 1983, 1(2—3): 179—209.
- [6] C. Kingsley, "A Hierarchical, Error-Tolerant Compactor", Proc. 21st DAC June, 1984, 293—297.
- [7] Y. Z. Liao and C. K. Wong, "An Algorithm to Compact a VLSI Symbolic Layout with Mixed Constraints", IEEE Trans. on CAD, 1983, CAD-2(2): 62—69.