

Better Initial Placement Algorithm for Large-Scale Mixed-Mode Detailed Placement^{*}

Zhou Qiang, Luo Lijuan, Hong Xianlong and Zhou Hanbin

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

Abstract: An algorithm is presented for better legal solution in detailed placement of large scale mixed macros and standard cells IC design. Due to the limitation of computing complexity, an effective and efficient initial placement is very important for detailed placement. Novelty of this algorithm lies in a better solution at initial stage by using network-flow method to satisfy row capacity constraint and the thought of linear placement problem (LPP) to resolve overlaps. Moreover, divide-and-conquer strategy and other simplified methods are adopted to minimize complexity. Experimental results show that the algorithm can get an average of 16% wire length improvement on PAFLO in reasonable CPU time.

Key words: placement; detailed placement; mixed-mode placement; algorithm

EEACC: 2570 **CCACC:** 7410D

CLC number: TN47

Document code: A

Article ID: 0253-4177(2004)07-0784-06

1 Introduction

With the advance of IC technology, especially with the reuse of IP blocks for multi-million-gate ASICs and SoC designs, most of modern IC designs consist of a very large number of standard cells mixed with many big macros, such as ROMs, RAMs and IP blocks^[1]. In the mixed-mode placement (MMP), the objects to be placed are standard cells which are of the same height but variable width, as well as some big macro blocks. Recently, academia and industry have invested considerable efforts in research on the large-scale MMP, such as a placement-floorplanning-placement flow^[2], multi-level placement^[1, 3], and a quadratic placer^[4, 5].

Having a better performance, the quadratic placement is widely accepted in modern academia and industry. The quadratic-based placement process has two stages: global placement and detailed placement (some authors speak of final placement)^[4, 5]. During global placement we find the global optimal position of each placeable macros and every standard cells neglecting the "slot constraints", namely cells can be placed in any position of the layout area. After global placement it comes with detailed placement. The task of detailed placement is to place cells to cell rows, which can be assumed to be given in advance, and to modify a placement to remove the overlap between cells and further improve placement quality.

Current detailed placement algorithms fall into

^{*} Project supported by National High Technology Research and Development Program of China (No. 2002AA1Z1460), National Natural Science Foundation of China (No. 60176016), and Specialized Research Fund for the Doctoral Program of Higher Education (No. 20020003008)

Zhou Qiang male, was born in 1961, associate professor. His research interest is VLSI layout algorithm and system integration.

Luo Lijuan female, was born in 1981, MS candidate. Her research interest is placement algorithm.

Hong Xianlong male, was born in 1940, professor. His research interests include layout algorithms and systems.

Received 25 July 2003, revised manuscript received 1 November 2003

©2004 The Chinese Institute of Electronics

two categories: The first one directly constructs optimal final legal placement without post optimization routine. Methods adopted include efficient algorithm from combinatorial optimization algorithms^[6], mathematical programming methods, such as the detailed placement of RITUAL^[7], and so on. The other one (see examples in Refs. [8~10]) is done through two steps: (1) construct an initial legal placement (this step is called initial detailed placement); (2) use local exchanges to iteratively improve current solution. Most algorithms of this category focus on improving the second step but when it comes to the first step only simple implementation is adopted. However, in the large-scale MMP, the chip is divided into many discrete layout areas. In this case, simple initial detailed placement may cost too much time and has bad solution. Our experiments show that simple implementation at initial stage not only lengthens the post optimization process but also inevitably infects the final wire length.

In this paper, a new algorithm is proposed aiming for improving initial detailed placement. The new algorithm adopts a divide-and-conquer strategy, namely satisfies local capacity constraint hierarchically, and in every hierarchy uses the same network flow method to balance cells in a global view. When resolving overlaps, according to the occupy ratio of a row, this algorithm adopts either an optimal method from the thought of LPP or a simple one so that great run-time vs quality trade-offs is achieved. As a whole, the new algorithm is both effective and efficient.

2 Notations and outline of the algorithm

Input of this algorithm is a physically illegal global placement where cells are in global optimal positions with local capacity violation and overlaps between standard cells. The task is to balance local capacity and resolve overlaps considering wire length objective.

In mixed-mode placement, the chip area is divided into many discrete layout areas. Rows for standard cells usually called zones are divided into sub-zones by macros. The criterion of balancing local capacity is to ensure that every zone as well as sub-zone can accommodate all cells within it. And, in the large-scale MMP, too many sub-zones will cost too many CPU computing. Here, a divide-and-conquer strategy is adopted in order to minimize complexity. In detail, the chip is divided into $m \times n$ equivalent square areas called regions and zones are further divided into sub-zones by region boundaries (see Fig. 1).

Thus, the new initial detailed placement algorithm is divided into three phases: balancing the regions (Section 3), balancing the sub-zones in every region (Section 4), merging sub-zones and resolving overlaps in zones (Section 5).

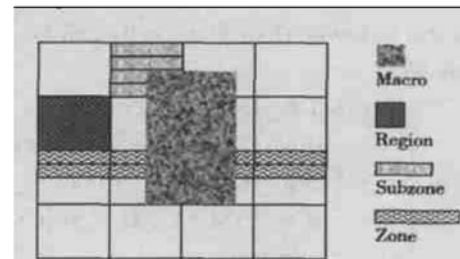


Fig. 1 Notations

3 Balancing the regions

The task of this phase is to ensure that every region can not only hold cells in it but also have some free space in order to further balance sub-zones within the region. It's natural to solve a minimum cost flow problem so as to minimize movement from global optimal position. Establishment of the minimum cost flow problem is similar to Ref. [6], but there are still two main differences: (1) Regions in Ref. [6] are the partition result of global placement, while in this paper they are defined by us. (2) Because there are much more cells in a region here than in Ref. [6], when actually moving cells among regions a simplified solution is adopted instead of the dynamic programming tech-

nique of Ref. [6]. Now we'll describe the whole phase in detail.

As in Ref. [6], first compute a directed graph G where the vertices correspond to the regions and adjacent regions named neighbor regions are joined by edges.

For convenience, define the height of a row to be one unit. Thus the area of cell c is simply denoted by cell width $w(c)$ (since all standard cells are of the same height), and it is the same to a sub-zone. Let $w(r)$ be the area of region r , which equals to the total area of all sub-zones in this region. Let $C(r)$ be the set of all the cells in r , and $w(C(r))$ be the total area of $C(r)$. Suppose the desired max density is MAX_D ; e. g. $MAX_D = 0.9$ means preferably at most 90% of a region should be occupied by the cells. Suppose the desired min density is MIN_D ; e. g. $MIN_D = 0.8$ means if a region's occupy ratio is lower than 80%, cells can be moved to it. Now set

$$b(r) = \begin{cases} w(r) MIN_D - w(C(r)), & \text{if } w(r) MIN_D > w(C(r)) \\ w(r) MAX_D - w(C(r)), & \text{if } w(r) MAX_D < w(C(r)) \\ 0, & \text{otherwise} \end{cases}$$

In the first case, we call r a sink. And in the second case r is a source. Now a minimum-cost flow problem is set up. The definition of the edge cost and the capacity is also similar to Ref. [6].

After an optimal flow f has been determined, we realize the flow by actually moving cells. For an overloaded region r , let $f(r, r')$ be the flow of edge (r, r') , and $N(r)$ be the set of neighbor regions r' with $f(r, r') > 0$. Define $cost(c, r, r')$ as the cost of the moving cell c from region r to r' . Figure 2 shows how to select removing cells from r and move them to which new positions.

From the above algorithm, we can see that owing to the variation of cell width it may be impossible to balance regions.

It remains to show how to calculate $cost(c, r, r')$. As a natural cost measure we take the increase of the bounding-box wire length. Now only consid-

er x direction, and it is the same to y . Obviously cost varies with the new position of C in r' . The cost function can be described as a piecewise linear curve between the left and right boundary of r' . So it's natural to set $cost(c, r, r')$ to the cost curve min point, and when actually moving C from r to r' , we move the cell to the corresponding minimum cost position.

Algorithm: Removing Cells(r)

(1) calculate $cost(c, r, r')$, $\forall c \in C(r)$, $\forall r' \in N(r)$.

(2) WHILE $w(C(r)) > w(r) \times MAX_D$ AND $N(r) \neq \emptyset$

S1: select the minimum $cost(c, r, r')$, and move c from r to r'

S2: $f(r, r') = f(r, r') - w(c)$.

S3: IF $f(r, r') \leq 0$

remove r' from $N(r)$

END IF

S4: re-calculate $cost(c', r, r')$ for $\forall c' \in C(r)$, $\forall r' \in N(r)$

according to the new position of c

END WHILE

Fig. 2 Removing Cells(r)

4 Balancing the sub-zones in every region

The task of this phase is trying to eliminate capacity violation in sub-zones. Assign cells to the nearest sub-zones and then basically the same framework as the first phase is used with the following slight differences:

First, different from the definition of neighbor regions, two sub-zones are called neighbors if they are subsequent in the same cell row or they are in adjacent rows with overlap in x direction.

Second, in order to balance sub-zones as best we can in one region, free space is always reserved in a region, which is realized by setting MAX_D to a proper fraction. Here free space is no longer needed in sub-zones, so set MAX_D to 1.

5 Resolving overlaps in zones

The third phase considers the so-called zones. First, merge adjacent sub-zones separated by region boundaries. As analyzed before, there may be

some zones still overloaded. In such (rare) case, randomly select a cell from the overloaded zone, and move it to a sinking zone, which can hold it with the minimum moving cost (calculation of the moving cost is similar to that in Section 3); iterate until there's no capacity violation.

After it is guaranteed that $w(C(r)) \leq w(r)$ for zones r , begin to resolve overlaps each zone subsequently. The problem can be stated as follows:

Given a zone with n movable cells $c_i, i = 1, 2, \dots, n$, whose left-to-right order is fixed according to their centers, but whose positions are variable. All cells in other zones are fixed cells, i. e. they have fixed positions. We seek a non-overlapping placement of movable cells such that the total bounding-box half-perimeter of all signal nets is minimized^[11].

We mainly adopt the improved algorithm of optimal single-row placement (OSRP) introduced in Ref. [12] to solve this problem. In that paper, the approach was applied to improve the design after detailed placement, while here it is adopted as a phase of initial detailed placement in order to construct an optimal initial legal placement.

The approach can be described as follows: first, find the optimal placement for all cells in one zone, and then subsequently place every cell in optimal position. In the operator, if there is a conflict between cells c_i and c_{i+1} , merge c_i with c_{i+1} , and find the optimal placement for the union $\{c_i\} \cup \{c_{i+1}\}$. Continue until successfully place the cell c_n .

As in Refs. [11, 12], we define for each net N :

$L(N) (R(N))$ is the leftmost (rightmost) movable cell, which has a pin of N ;

$f_l(N) (f_r(N))$ is the position of the leftmost (rightmost) fixed pin of N .

When finding the optimal placement of cell c , this approach comprehensively considers such special positions as $f_l(N) (f_r(N))$, if

$c = L(N) (R(N))$ (N is an arbitrary net.)

To minimize complexity and save computation time, improvements are made based on this approach.

First, since the left-to-right order of all cells is fixed in this zone, to avoid overlaps, the scope where a cell can shift is limited and we can calculate the leftmost and rightmost possible positions of each cell, which are denoted by $f_l(c)$ and $f_r(c)$ respectively. Thus in such situation as

$$L(N) = c, f_l(c) \geq f_l(N) \text{ or } R(N) = c, f_r(c) \leq f_r(N)$$

When finding optimal placement of c , need not take $f_l(N)$ or $f_r(N)$ into account, which means the placement of c will not influence the length of N .

Second, for net N without fixed pins, instead of replacing it by two new nets as Ref. [12], we define

$$f_l(N) = +\infty, f_r(N) = -\infty$$

It is obvious that the result is the same.

The above two points of improvement are both aiming for decreasing the number of nets considered, since the computation time of OSRP is related to the number of nets.

Finally, it is found that if the free space of a zone is little, wire length reduction of the optimal approach is very limited compared with a simple implementation and usually not worth the computation time. So a value-density d is given. If the zone density is lower than d , adopt the optimal resolving overlaps approach, and otherwise adopt a simple one as in Ref. [10].

6 Experimental results

We have implemented this initial detailed placement algorithm on a 600MHz/8GB memory Sun workstation running on Solaris. The case named block8 is selected from an industry IC design. The others are downloaded from ISPD02 Benchmark^[13]. They are tested in LEF/DEF format.

Given the same results of global placement, we apply both the new detailed placement algorithm and PAFLO^[10] (extended to mixed-mode detailed placement). The new detailed placement algorithm uses the same improving stage as PAFLO. The difference between the new detailed placement algo-

rithm and PAFLO is the method used in initial detailed placement stage. Table 2 shows the final placement results. The column WL is the total wire length after detailed placement. The column time records the the CPU time of whole detailed placement phase. From the results, we can see that surprisingly significant wire length reductions (an average of 16%) are obtained in comparing CPU time. All improvements are more than 10% except the case ibm05, the standard cell placement case.

Table 1 Properties of tested circuits

Circuit	# cells	# nets	# macros
block8	9253	10049	8
ibm01	12260	14111	246
ibm02	19071	19584	271
ibm03	22563	27401	290
ibm04	26925	31970	295
ibm05	28146	28146	0
ibm06	32154	34826	178

Table 2 Final placement result comparison

Circuit	PAFLO		New Algorithm		
	WL	Time/s	WL	WL improvement	Time/s
block8	2.22×10^9	121	1.99×10^9	10.36%	147
ibm01	5.15×10^6	200	4.36×10^6	15.33%	243
ibm02	1.22×10^7	455	9.26×10^6	24.10%	497
ibm03	1.62×10^7	696	1.36×10^7	16.05%	643
ibm04	2.04×10^7	704	1.68×10^7	17.65%	738
ibm05	1.68×10^7	480	1.60×10^7	4.8%	561
ibm06	1.67×10^7	904	1.31×10^7	21.56%	902
avg.				15.69%	

7 Concluding remarks

In this paper, an efficient and effective algorithm for optimal initial solution is proposed in mixed-mode detailed placement. We adopt the method of network flow to satisfy row capacity constraint in a global view. Moreover, owing to the divide-and-conquer strategy we can limit the number of vertices and therefore minimize the computation time. When resolving overlaps the thought of LPP^[11] is used and some simplifications are made. Thanks to this optimal initial solution the time of post optimization can be shortened, thus the total

time of detailed placement can compare to PAFLO although the initial stage is much more delicate. Experimental results verify that the new algorithm is effective and efficient.

A natural progression of this work is to study more in depth relationship between initial detailed placement and post optimization, so that placement quality can be further improved. Besides, there are a number of control parameters as described will be studied. The strategy of setting these values may evolve over time as we gain more insight into their effects.

References

- [1] Chang C C, Cong J, Yuan X. Multi-level placement for large-scale mixed-size IC designs. In: Proc ACM/IEEE ASP-DAC, 2003: 325
- [2] Adya S N, Markov I L. Consistent placement of macro-block using floorplanning and standard-cell placement. In: Proc ACM Int Symp on Physical Design, 2002: 12
- [3] Chang C C, Cong J, Pan D, et al. Physical hierarchy generation with routing congestion control. In: Proc Int Symp on Physical Design, 2002: 36
- [4] Yu Hong, Hong Xianlong, Cai Yici. MMP: a novel placement algorithm for combined macro block and standard cell layout design. In: Proc ACM/IEEE ASP-DAC, 2000: 271
- [5] Wu Weimin, Hong Xianlong, Cai Yici, et al. A mixed mode placement algorithm for combined design of macro blocks and standard cells. In: Proc IEEE ASICON, 2001: 122
- [6] Vygen J. Algorithms for detailed placement of standard cells. In: Proc IEEE of the Conference Design Automation and Test in Europe (DATE), 1998: 321
- [7] Srinivasan A, Chaudhary K, Kuh E S. RITUAL: A performance driven placement algorithm. IEEE Trans Circuits and Systems II, 1992, 39(11): 825
- [8] Yao Bo, Hou Wenting, Hong Xianlong, et al. Fame: a fast detailed placement algorithm for standard cell layout based on mixed mincut and enumeration. Chinese Journal of Semiconductors, 2000, 21(8): 744
- [9] Sarrafzadeh M, Wang M, Yang X. Modern placement techniques. New York: Kluwer Academic Publishers, 2002
- [10] Zhou Hanbin, Wu Weimin, Hong Xianlong. PAFLO: a fast standard-cell detailed placement algorithm. In: Proc of IC-CAS, 2002: 1401
- [11] Kahng A B, Tucker P, Zelkovsky A. Optimization of linear placements for wirelength minimization with free sites. In: Proc ACM/IEEE of the Asia and South Pacific Design Au-

- tomation Conference(ASP-DAC), 1999: 241
- [12] Brenner U, Vygen J. Faster optimal single-row placement with fixed ordering. In: Proc IEEE of the Conference on Design Automation and Test in Europe, 2000: 117
- [13] <http://vlsicad.eecs.umich.edu/BK/ISPD02bench>

大规模混合模式初始详细布局算法*

周 强 罗丽娟 洪先龙 周汉斌

(清华大学计算机科学与技术系, 北京 100084)

摘要: 以大规模混合模式布局问题为背景, 提出了有效的初始详细布局算法. 在大规模混合模式布局问题中, 由于受到计算复杂性的限制, 有效的初始布局算法显得非常重要. 该算法采用网络流方法来满足行容量约束, 采用线性布局策略解决单元重叠问题. 同时, 为解决大规模设计问题, 整体上采用分治策略和简化策略, 有效地控制问题的规模, 以时间开销的少量增加换取线长的明显改善. 实验结果表明该算法能够取得比较好的效果, 平均比 PAFLO 算法有 16% 的线长改善, 而 CPU 计算时间只有少量增加.

关键词: 布局; 详细布局; 混合模式布局; 算法

EEACC: 2570 **CCACC:** 7410D

中图分类号: TN47 **文献标识码:** A **文章编号:** 0253-4177(2004)07-0784-06

* 国家高技术研究与发展计划(批准号: 2002AA1Z1460), 国家自然科学基金(批准号: 60176016), 及高等学校博士学科点专项科研基金(批准号: 20020003008) 资助项目

周 强 男, 1961 年出生, 副教授, 目前研究方向为 VLSI 布图理论与算法, 系统集成.

罗丽娟 女, 1981 年出生, 硕士研究生, 从事布局算法研究.

洪先龙 男, 1940 年出生, 教授, 博士生导师, 研究方向为 VLSI 布图理论与算法.

2003-07-25 收到, 2003-11-01 定稿

©2004 中国电子学会