第 25 卷第 2 期 半 导 体 学 报 Vol. 25, No. 2
2004 年 2 月 CHINESE JOURNAL OF SEMICONDUCTORS Feb., 2004

Incremental Placement Algorithm for Timing and Routability Optimization*

Li Zhuoyuan, Wu Weimin and Hong Xianlong

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

Abstract: A new approach of incremental placement approach is described. The obtained timing information drives an efficient net-based placement technique, which dynamically adapts the net weights during successive placement steps. Several methods to combine timing optimization and congestion reducing together are proposed. Cells on critical paths are replaced according to timing and congestion constraints. Experimental results show that our approach can efficiently reduce cycle time and enhance route ability. The max path delay is reduced by 10% on an average after incremental placement on wirelength-optimized circuits. And it achieves the same quality with a high speed up compared to timing driven detailed placement algorithm.

Key words: timing; congestion; incremental placement
EEACC: 2570 **CCACC**: 7410D
CLC number: TN405.97 **Document code**: A **Article ID**: 0253-4177(2004)02-0158-07

1 Introduction

As VLSI technology advances, the system complexity continues to increase and physical design is getting more and more difficult. Cycle time optimization has become one of the most important issues of the design of highly integrated circuits. Specifically, the interconnect delay tends to dominate the performance of electronic circuits. The strong demand for faster clock calls for more efficient timing optimization layout methods.

Existing timing driven placement algorithms could be classified into two major categories: net-based and path-based. Net-based algorithms try to satisfy timing constraints by assigning different weights to nets and updating these weights based on their timing criticality. These algorithms try to minimize the total weighted net length through placing cells in an iterative manner[1,2]. A major problem of net-based algorithms is the selection of the weights or bounds. High weights of fixed bounds on a set of critical nets may create excessive delays in other nets. This will result in an unexpected oscillation. The path-based algorithms overcome these difficulties by modeling the problem more accurately. It chooses a fixed number of critical paths after performing timing analysis on the netlist and then seeks to minimize the delay of these paths. The problem could be solved through linear programming or stimulus annealing[3,4]. To reduce the computational effort, the problem could be formulated as a quadratic programming problem and solved[5].

* Project supported by National Natural Science Foundation of China(No. 60121120706), National Natural Science Foundation of USA(No. CCR-0096383), Hi-Tech Research & Development Program of China(No. 2002AA1A1460) and National Foundation Research (973) Program of China(No. G1998030403)

Li Zhuoyuan male, was born in 1980, PhD candidate. His research interests focus on VLSI layout design.

Received 25 March 2003, revised manuscript received 16 October 2003

With the advent of over-the-cell routing, the goal of every place and route methodology has been to utilize all available active area to prevent spilling of routes into channels. There have been many literatures on reducing routing congestion during placement stage[6~8]. And some algorithms have tried to combine timing optimization and congestion reducing together in quadratic programming or through force-directed technique[9,10]. However, the quality of the layout is limited due to the excessive computational load.

In this paper, a new incremental placement method T-ECOP (ECO: engineering change order[14]) is presented to reduce the cycle time while enhancing the routability. It estimates net delay using the same timing model as RITUAL[5] and specifies all the critical paths. The obtained timing information drives an efficient net-based placement method, which minimizes a linear wirelength cost function. We also proposed several different methods to integrate congestion reducing and timing optimization efficiently.

2 Timing driven incremental placement

2.1 Timing model

We start with a placed circuit, so we are given the initial coordinates for all cells and pads. We also know the arrival time, T_{start}, at the primary inputs (PIs) and the required time, T_{end}, at the primary output (POs). Then we perform timing analysis on the circuit with the same timing model used in RITUAL, where the delay of a net i is approximated by

$$\text{Delay}_i = R_i(C_i^{net} + C_i^{load}) \quad (1)$$

$$C_i^{net} = \sum_{j \in N_i} (C_h|x_0 - x_j| + C_v|y_0 - y_j|) \quad (2)$$

$$R_i = \sum_{j \in N_i} (R_h|x_0 - x_j| + R_v|y_0 - y_j|) \quad (3)$$

where C_i^{net} is the capacitance of driving pins of net i, C_i^{load} is the interconnect capacitance of the output net i, C_h and R_h represent the capacitance and resistance per unit length of horizontal wire, C_v and R_v represent the same items of vertical wire, (x_0, y_0) are the coordinates of the source pin of the net.

2.2 Problem formulation

Given T_{start} at the circuit inputs and T_{end} at the circuit outputs, we can easily compute the arrival and required times, a_i and r_i, for all the other vertices in the circuit in a breath-first-search manner. Based on these values, a slack s_i is for each vertex is defined as $s_i = r_i - a_i$. A critical path is a sequence of vertices along a path from primary input to primary output where all vertices have negative slacks. To reduce the maximum path delay, the weights of the nets which are part of critical paths are increased during successive placement steps.

It has been demonstrated that minimizing a linear objective function yields impressive results in cell placement. Thus we formulate the linear objective function with a linear wirelength model. The wirelength of the nets connected to a cell C_i is computed as follows

$$L_p = \sum_{n \in N_i, i \neq j} w_n(|x_i - x_j| + |y_i - y_j|) \quad (4)$$

where w_n is the weight of the net n. The timing driven placement problem could be formulated as minimize the total weighted wirelength of all the nets.

2.3 Efficient placement approach T-ECOP

The objective function L_p could be separated in the x- and y-direction, we are considering only the function in the x-direction in the following. It is written as

$$L_{px} = \sum_{n \in N_i, i \neq j} w_n|x_i - x_j| \quad (5)$$

As minimizing the linear objective function would lead to a linear programming problem, which is very time consuming, we rewrite it as a quadratic objective function:

$$L_{px} = \sum_{n \in N_i, i \neq j} u_n(x_i - x_j)^2 \quad (6)$$

where $u_n = \frac{w_n}{|x_i - x_j|}$. Assuming that cells connected to the cell C_i are fixed, the optimal x_i to minimize the objective is as follows

$$x_i = \left[\sum_{n\in N_i, i\neq j} u_n x_j\right] \Big/ \sum_{n\in N_i, i\neq j} u_n (n - 1) \qquad (7)$$

where n represents the number of connect pins of net n.

So the objective leads to a quadratic programming problem and could be solved by applying quadratic programming techniques. The net weight u_n is iteratively adapted according to timing analysis information during the solution of the programming problem. The placement methodology is shown in Fig. 1. This method has been successfully applied in Ref. [15].

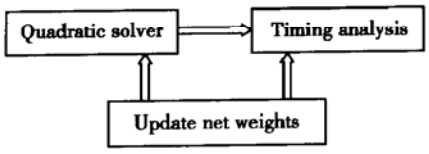

Fig. 1 Timing driven incremental placement methodology

Some defects in this methodology prevents it to be used directly in our algorithm. For our incremental placement approach, the aim is to achieve timing improvement through local replacement in a short amount of time. A quadratic solver results in redoing placement for the circuit. The iterative computations are time consuming and the metrics of the initial circuit could not be preserved.

And minimizing the delay of a critical net may lead an increase in the delay of other nets. This may cause additional critical paths and the delays of the nets in these paths should also be minimized. This may result in an excessive delay in the previous net. This oscillating effect could be prevented through enough successive refinements. But the iterative computation times should be limited to get a high efficiency. We resolve the confliction by assigning appropriate weights for the nets during successive placement steps.

Let w_n^i denotes the weight of the net n for the i-th placement step. It should contain some information in history for critical paths in the previous minimization steps. It could be computed as follows

$$w_n^i = \begin{cases} 1/n + w_n^{i-1} & \text{if } C(\text{net } n) = 0 \\ (1 - \lambda s_{\text{po}} i)/n + w_n^{i-1} & \text{otherwise} \end{cases}$$

where n represents the number of connect pins of the net n. λ represents the timing factor, s_{po} is the slack for the critical path where net n belongs to, w_n^{i-1} is the weight for previous placement step, C (net n) = 1 denotes the net n is part of a critical path in current placement step. If the net n is not critical, it is added up by $1/n$ for only wirelength net weight. Otherwise, it is added up by both wirelength and timing net weight. Critical nets with more negative slacks will be assigned larger timing net weights. And the timing net weight is multipled by i so that the timing performance could be gradually improved larger.

As mentioned above, the perturbation to the initial circuit should be minimized so that its metrics could be preserved. To achieve this, we only choose 10% of cirtical paths with the most negative slacks to improve in each placement step. The optimal positions of cells on these paths are recomputed. An efficient wirelength driven incremental placement algorithm W-ECOP[11] is adopted to replace these cells and erase overlap.

Experimental results show that if cells are placed to their optimal positions computed by Eq. (7), the performance of circuit will not be enhanced but be brought down. This is because there are many interconnected cells to replace. The optimal coordinates for these cells are affected each other. It needs enough iterative computation times to get the final placement. A compromise mechanism is adopted to resolve this. Let x_1 and x_2 be the initial and optimal x-coordinate of a cell, its new x-coordinate is $\frac{x_1 + x_2}{2}$. Thus cells are moved towards their optimal positions continually in each placement step.

Experimental results of the new efficient placement approach are shown in Table 1. The ini-

tial placement is generated by a detailed placement algorithm FAME[12]. It is shown that the max path delay is reduced by 13.4% on the average and the wirelength is increased no more than 5%. So the metrics of the initial placement is preserved. We can also see that our method is very fast.

Table 1　Experimental results on MCNC benchmarks

Circuit	# Cell	# Pin	Wirelength			Max path delay			CPU/s
			BIP	AIP	Inc/%	BIP	AIP	Imp/%	
C2	590	233	402769	430863	6.98	31.42	28.04	10.76	0.32
C5	1586	178	1024149	1071637	4.64	65.95	45.07	31.66	1.59
C7	2150	315	1703939	1761357	3.37	61.10	58.83	3.72	1.98
S13207	4267	1490	6343467	6500860	2.48	53.90	49.62	7.94	15.32
AVQ	21854	64	10469706	11032516	5.38	158.91	139.99	11.90	182.19
Average	/	/	/	/	4.57	/	/	13.20	/

BIP/AIP: before/after incremental placement

3　Integrate improving timing and congestion

Congestion minimization is the least understood of placement objectives, however, it models routablity most accurately. Minimizing congestion and minimizing wirelength may conflict each other in local regions. To get a congestion optimal placement, the wirelength have to be sacrificed. It may affect the performance of the circuit. So it is important to combine congestion reducing and timing optimization together at the placement stage.

Three different methods to integrate improving timing and routablilty are proposed based on the timing driven incremental placement approach. The first dynamically adapt the timing net weights according to density balance information. The second carries out a congestion optimization process after timing minimization placement. The third adjusts cell positions with respect to congestion constraints. These methods aim to achieve good result without increasing the computational load.

3.1　Net weights according to density balance

After each placement step, we do the routing and congestion estimation with the method proposed in Ref. [13]. The chip is partitioned into several rectangular global bins. Cells in congested bins should move out to reduce the routing demand and achieve more free space for routing. The density of bin b is defined as the coverage of the bin by layout object:

$$D(b) = (\mathrm{CON_v}(b) + \mathrm{CON_h}(b)) \times \frac{\text{binsize}}{\text{freespace}} \tag{8}$$

where $\mathrm{CON_v}(b)$ and $\mathrm{CON_h}(b)$ are the vertical and horizontal congestion cost of b[13]. To take congestion requirements into account, the objective function is modified to

$$L_{\mathrm{px}} = \sum_{n \in N} \sum_{i \neq j} u_n g_{nj} (x_i - x_j)^2 \tag{9}$$

where $g_{nj} = \dfrac{1}{D(b)}$, b represents the bin that cell C_j locates in. The optimal position for a cell is to minimize the objective. For a more congested bin, which has more routing demand and fewer freespace, $D(b)$ is larger so that the value of g_{nj} is small. It prevents C_i move into the congested bin. Then we can reduce cell overlapping and potiental routing congestion when improving timing performance.

3.2　A post-process to reduce congestion

The congestion driven incremental placement algorithm C-ECOP proposed in Ref. [12] could reduce routing congestion efficiently during placement stage. After routing and congestion estimation, congestion areas on the chip are relieved through cell moving. An integer linear programming (ILP) problem is formulated to resolve conflicts among multiple congestion areas and avoid

causing unexpected congest areas. So it is naturally to carry out the congestion reducing process after timing minimization step. Cells on critical paths are fixed during congestion minimization so that the timing performance of the circuit will not be broken.

3.3 Congestion constrain during placement

Cell flow tendencies are computed in C-ECOP to decide which cell should move out of congested areas so that the perturbation to the initial placement could be minimized. There is an example for cell flow tendency computation in Fig. 2. If cell C_k moves from bin(i,j) to bin$(i+1,j)$, nets crossing the right edge of bin(i,j) will decrease. It will lead a decrease in the horizontal routing demand in the bin. So C_k has a tendency to more right. Moreover, the flow tendency of cells could be regarded as the net-cuts crossing the global edges. Generally speaking, reducing in net-cuts is consistent with reducing in wirelength, so moving cells according to its flow tendency will lead a decrease in both congestion and wirelength.

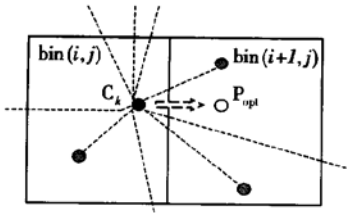

Fig. 2 Cell flow tendency

Assume the optimal position decided by Eq. (4) is P_{opt} as shown in the figure, C_k could be moved towards P_{opt} because it is consistent with its flow tendency. Otherwise, C_k could not be moved. With this method, cells will move away from the congested areas in the successive placement step. Then timing minimization could be performed with respect to the congestion constrains. We may achieve timing optimization and congestion reducing during the placement.

4 Experimental results

These algorithms are implemented in C. All the experiments are done on a Sun E450 workstation with 4Gb memory. We test the algorithms on a set of MCNC benchmarks. The circuit specifications for congestion estimation are reported in Table 2. Each circuit is partitioned as there are about twenty-five cells in each bin. The routing estimation results in Ref. [13] denote it could maintain accuracy and efficiency in congestion estimation. Experimental results are reported and compared in Table 2. The vertical and horizontal routing supply of each global bin is generated from the technique parameters.

Table 2 Circuits specifications

Circuit	Grid	V/H cap	C/B	Overflow
C2	5×5	13/18	23.6	44
C5	8×8	16/17	24.9	41
C7	9×9	16/20	26.5	68
S13207	12×12	26/34	29.6	200
Avq	30×30	17/18	24.3	524

V/H cap: vertical/horizontal routing capacitance of each bin; C/B: average number of cells in each bin; Overflow: sum of routing demand exceeds routing supply for each global edge

Experimental results of three different methods are reported and compared in Table 3. From the table we can see that the first and the third method are with the same speed. The third method could improve timing performance better but the routing congestion is deteriorated much. The results of the first method are inversed. The second method is the fastest and achieves the best results in congestion minimization. For a majority of circuits, it also achieves the best timing performance. It is because we fix cells on the critical paths to preserve the performance when reducing routing congestion.

The other two methods, which combine delay and route ability optimization together in iterative placement process, do not achieve good results as we have expected. It may be because minimizing congestion and minimizing wirelength conflict each

other in local regions[12]. To get a congestion optimal placement, the wirelength have to be sacrificed. It may affect the performance of the circuit. So it is difficult to combine the congestion reducing and delay minimization together in a single optimization process. Such instances also exist in power and delay, thermal and area, and some other objectives. More comprehensive studies should be focused on these subjects to better understand multi-objective optimization techniques.

Table 3　Different methods to alleviate routing congestion

Circuit	Method 1			Method 2			Method 3		
	T_{path}^{max}	Overflow	CPU/s	T_{path}^{max}	Overflow	CPU/s	T_{path}^{max}	Overflow	CPU/s
C2	25.39	54	0.45	24.75	35	0.49	26.75	48	0.47
C5	46.83	85	2.07	47.40	52	2.03	45.15	105	2.06
C7	61.48	107	2.46	59.70	65	2.54	60.34	105	2.42
S13207	53.26	253	19.55	53.17	160	17.64	51.97	269	19.54
Avq	142.4	721	290.2	140.8	564	206.6	141.3	810	290.02

5 Discussion

It is shown in Table 1 that our algorithm could achieve good results in a short amount of time. The two basic demands for incremental placement algorithms, high efficiency and preserve the metrics of the initial circuits, are satisfied. It denotes the advantage of our algorithm. But further experiments show that the timing performance could not be improved more by the algorithm. If the iterative times are beyond some threshold, the timing performance is getting worse during successive placement steps. We analyze this interesting result and give some discussion on it.

First of all, when the optimization objective is changed, the solution space of the problem is also changed as shown in Fig. 3. The original instance is solved to yield an optimal solution S_0. After the optimization objective changed, the cost surface of optimization is changed from the solid cost surface to the dashed surface. The initial instance is corresponding to a new cost value T_0. It is in the "basin of attraction" of some local minimum S_1. It is easy to find the new local optimal solution S_1 through some heuristic method. For the incremental placement algorithms, what is important is to achieve high efficiency.

On the other hand, incremental placement algorithms should preserve the metrics of the initial

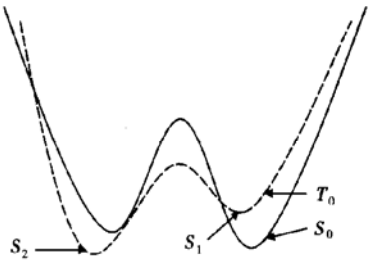

Fig. 3　Solution space changed

circuit. It only allows local changes and improvements to the original placement. The positions of moved cells should be restricted to their neighboring areas. If the new solution space differs sufficiently from the original one, the incremental placement algorithm could not escape from the basin of attraction in order to find a good solution S_2 as shown in the Figure. Randomized algorithms such as stimulus annealing or genetic algorithm could not be applied to change the situation because it will seriously affect the efficiency. This is the shortage of incremental placement algorithms. It is necessary to redo the global or detailed placement if we want to get better solution.

6 Conclusion

A new efficient approach for timing driven incremental placement is presented in this paper. It could improve timing performance in successive placement steps while maintaining the metrics of

the initial circuit. Several different methods to alleviate routing congestion during placement stage are proposed to combine timing optimization and congestion reducing together. These methods could improve congestion efficiently without increasing computational load. Experimental results show the high efficiency of our algorithm.

References

[1] Gao T, Vaidya P M, Liu C L. A performance driven macro-cell placement algorithm. In: Proc 29th ACM/IEEE DAC, 1992: 147

[2] Tasy R S, Youssef M N. An analytic net weighting approach for performance optimization in circuit placement. In: Proc 28th ACM/IEEE DAC, 1991: 636

[3] Jackson M A B, Kuh E S. Performance-driven placement of cell based ICs. In: Proc 26th ACM/IEEE DAC, 1989: 370

[4] Hauge P S, Nair R, Yoffa E J. Circuit placement for predictable performance. In: Proc International Conference on Computer-Aided Design, 1987: 88

[5] Srinivasan A, Chaudhary K, Kuh E S. RITUAL: a performance driven placement algorithm for small cell ICs. IEEE/ACM Int Conference on CAD, 1991: 48

[6] Wang Maogang, Sarrafzadeh M. Modeling and minimization of routing congestion. Proceedings of the ASP-DAC 2000, 2000: 185

[7] Yang Xiaojian, Kastner R, Sarrafzadeh M. Congestion reduction during placement based on integer programming. IEEE/ACM International Conference on Computer Aided Design, 2001: 573

[8] Wang Maogang, Yang Xiaojian, Sarrafzadeh M. Congestion minimization during placement. IEEE Trans Comput-Aided Des Integr Circuits Syst, 2000, 19(10): 1140

[9] Mo Fan, Tabbara A, Brayton R K. A timing-driven macro-cell placement algorithm. Proceedings. 2001 International Conference on Computer Design, 2001: 322

[10] Hou Wenting, Hong Xianlong, Wu Weimin, et al. Methods of improving rout ability in timing-driven placement. In: Proc 4th Int Conf on ASIC, Shanghai, 2001: 110

[11] Li Zhuoyuan, Wu Weimin, Hong Xianlong, et al. Incremental placement algorithm for standard-cell layout. Chinese Journal of Semiconductors, 2002, 23(12): 1338

[12] Li Zhuoyuan, Wu Weimin, Hong Xianlong. Congestion driven incremental placement algorithm for standard cell layout. Chinese Journal of Semiconductors, to be published

[13] Yao B, Hou W T, Hong X L, et al. FAME: a fast detailed placement algorithm for standard-cell layout based on mixed mincut and enumeration. In: Proc Int Conf On CAD/CG, Shanghai, 1999: 616

[14] Yu Hong, Yao Bo, Hong Xianlong, et al. An incremental placement algorithm based on row partitioning for standard cell layout. Chinese Journal of Semiconductors, 2001, 22: 96 (in Chinese) [于泓, 姚波, 洪先龙, 等. ECOP: 一种基于单元行划分的标准单元模式增量布局算法. 半导体学报, 2001, 22: 96]

[15] Riess B M, Ettelt G G. SPEED: fast and efficient timing driven placement. 1995 IEEE International Symposium on Circuits and Systems, 1995, 1: 377

优化时延与拥挤度的增量式布局算法*

李卓远 吴为民 洪先龙

(清华大学计算机科学与技术系, 北京 100084)

摘要: 提出了一种优化时延的增量式布局算法, 该算法根据时延分析的结果在迭代求解的过程中动态调整线网权值. 在此基础上, 提出了三种同时优化时延和拥挤度的多目标优化的布局算法, 在满足时延和拥挤度约束的前提下对关键路径上的单元进行位置调整. 实验结果表明该算法能够有效地提高芯片速度并降低走线拥挤. 对于优化线长得到的布局方案, 最长路径上的时延值在增量式布局之后能够降低 10%.

关键词: 时延; 拥挤度; 增量式布局

EEACC: 2570 **CCACC**: 7410D

中图分类号: TN405.97 **文献标识码**: A **文章编号**: 0253-4177(2004)02-0158-07

* 国家自然科学基金(批准号: 60121120706), 美国国家自然科学基金(批准号: CCR-0096383), 国家高技术研究发展计划(批准号: 2002AA1Z1460)和国家重点基础发展规划(批准号: G1998030403)资助项目

李卓远 男, 1980 年出生, 博士研究生, 研究方向为超大规模集成电路布图设计.

2003-03-25 收到, 2003-10-16 定稿