第 21 卷第 4 期 半 导 体 学 报 Vol. 21, No. 4
2000 年 4 月 CHINESE JOURNAL OF SEMICONDUCTORS Apr., 2000

Net Congestion Elimination for Datapaths by Placement Refinement*

WEN Hua(文 化) and TANG Pu-shan(唐璞山)

(*Department of Electronic Engineering, Fudan University, Shanghai* 200433, *China*)

Abstract: The layout of datapaths is much complexer than that of a normal IC chips, because more constraints must be considered. A novel method for eliminating the net congestion of datapath chips is presented. The main idea is to modify the placement locally according to the global routing result. The problem is abstracted to a nonlinear programming problem and could be transformed to a convex one. Experimental results demonstrate that the method can eliminate the net congestion of datapath chips effectively.

Key words: layout; placement refinement

EEACC: 0260; 1130B

CLC number: TP391.72 **Document code**: A **Article ID**: 0253-4177(2000)04-0325-08

1 Introduction

The datapath chip is a special species of IC chips. Datapath circuits are widely used in computer systems and communication systems as a data processor. Because there exist more constraints, the layout of datapaths is much complexer than the normal layout one. In a datapath chip, signals can be discriminated as a data flow and a control flow. Normally, the direction of the data flow is perpendicular to the control flow. Due to the multiple bits of the data signals, the structure of datapaths are often regular. So the standard cell design sytle is more eligible for the datapaths than the full custom design style. Where cells are placed in rows with integer multiple heights of the row one. The direction of the data

---

* Project Supported by the State Ninth Five-Year Plan for Technique R&D (Grant No. 96-738-01-08-05), by the Doctoral Program Foundation of Institutions of High Education (Grant No. 96024612) and by NSFC (Grant No. 69676023).

WEN Hua(文 化) was born in 1974. He received his B. S. degree from Shanghai Jiaotong University and received his M. S. degree from Fudan University. He is currently pursuing the Ph. D degree in the Department of Electronic Engineering of Fudan University. His interests concentrate on the VLSI Layout algorithms.

TANG Pu-shan(唐璞山) was born in 1934. He received his B. S. degree from Fudan University in 1952. Now he is a professor in the Department of Electronic Engineering. His research interests include the VLSI layout and the verification algorithms.

Received 23 September 1999, revised manuscript received 12 December 1999

flow is along the rows. The data process is often divided into a series of stages, which are separated by the regular-placed registers between two stages.

Same as other ICs, the layout of datapaths consists of four phases: partition, placement, routing and compaction[1]. The routing phase can be divided into two sub-phases: global routing and detailed routing. In the placement phase, the exact locations of cells are determined, while in the global routing phase, each net is assigned to a list of routing regions (Gcells) without specifying the actual geometric layout of wires. The task of detailed routings is to find the actual geometric layout of each net within the assigned Gcells. After the global routing phase, routability will be checked on the boundaries between the Gcells. If the track demand is beyond a certain percentage (e.g. 90%) of the track supply on a Gcell boundary, the boundary will be net congested. The net congestion makes the detailed routing in Gcells unrealizable or difficult.

So the net congestion should be diminished in the global routing result via iterative optimization. Sometimes, the net congestion is caused by the unreasonable cell locations, because the routability is not considered in most placement algorithms. To avoid making the placement optimizing object worse, it is necessary to adjust the cell locations locally when a new placement with higher routability is generated. We define this problem as net Congestion Elimination by Placement Refinement (CEPR problem).

In this paper, we give a mathematics programming model to represent the CEPR problem and discuss it. Section 2 presents the problem in details. Section 3 gives a mathematics programming model. How to solve the model is presented in section 4. Experimental results from industry are summarized in section 5.

2 CEPR Problem Description

The CEPR problem has the following characters.

a. Standard cell design style. All cells are placed in rows. Any cell height is the integer multiple of the row height, i.e., the height of all cells are not required to be the same.

b. A net congestion map is given with the problem's input. It is generated via the routablity check.

c. There exist cell groups. Those cells in the same group will move together to keep their relative locations unchanged. Those registers separating two data process stage should be put into one group.

The input placement of CEPR problem is the result of a certain placement algorithm. Since all placement algorithms are performance-driven (such as wire length driven, timing driven, power driven, etc), the input placement is optimized on a certain objective function. As most placement algorithms do not consider the routability in routing phase, it is possible to improve the routability in the CEPR problem solution. A reasonable solving method for CEPR problem should avoid making the initial optimizing object too worse. A simple strategy to achieve this aim is to move all cells locally, that is to say, all cells should not be moved too far away from their original locations.

3 Mathematics Programming Model

Recently, the mathematics programming method is used in algorithms to solve the layout problems[2,3]. In this paper, we will give a nonlinear programming model for CEPR problem and solve it. The location of a cell is represented by the coordinate of its bottom left corner, denoted as (x, y).

Definition 1: The neighborhood of a cell (x_0, y_0) is defined as the set of all location (x, y) satisfying

$$|x - x_0| < l \text{ and } |y - y_0| < w$$

where l, w is the range of neighborhood. Our model stands on the following assumptions.

Assumption 1: In the placement refinement, any cell can not be moved out of its original neighborhood.

This is used to guarentee that the refinement will not make the initial optimizing object worse.

Assumption 2: Decreasing the net congestion on the Gcell boundary, only those cells near the boundary might be moved.

So we can give a refinement region surrounding the net congested boundary. Only those cells in this region will be moved to decrease this boundary′s congestion. Naturally, when a boundary is net congested, we should reroute those nets crossing the boundary. These crossing nets can be discriminated as passing nets and local nets according to the cells they connect. If the net connects any cell in the refinement region, it is a local net, otherwise, it is a passing net. To decrease the congestion on a Gcell boundary, some crossing nets must be rerouted to pass other boundaries. It is much easier for the passing nets to rerouted in the global routing phase than the local nets, because there are no terminals near the congested boundary but more candidate paths avoiding them to pass the congested boundary. The most effective method to decrease the part of the net congestion is to exchange the net terminals of two sides of the congested boundary. Agreeing with the assumption 1, we can say that only those cells near the congested boundary can be exchanged. Furthermore, we give an assumption 3 as follows.

Assumption 3: The CEPR problem can be solved by the exchanging the cells between two sides of the congested boundary, or moving them from one side to the other.

The assumption 3 makes the CEPR problem like a min-cut problem[4-6]. But there are some differences between them. The aim of min-cut problem is to reduce the net number crossing the cut while the CEPR problem is to reduce the total net congestion of all congested boundaries, though in fact we will handle the congested boundaries one by one. So when we consider how to decrease the net congestion on a boundary, the total net congestion will not be increased.

The CEPR problem can be illuminated by the Figure 1. Before the placement refinement, there are 7 nets crossing the congested boundary. If we exchange the cell A and the cell D between two sides of the boundary, and reroute the local nets, there are only 4 nets crossing the boundary.

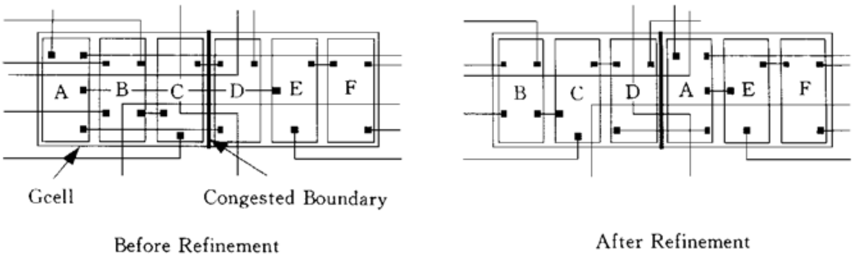

FIG. 1 Example of Placement Refinement

Let x_i denotes location of cell i. If the cell is located on the left side of the boundary, $x_i = 1$; while on the right side $x_i = 0$. Then the CEPR problem can be represented as: *how to decide $x_i = 0$ or $x_i = 1$ for each cell to minimize the total net congestion?*

Because the global routing algorithms are various, we should give a general objective function to indicate the optimizing direction, i.e., the descending direction of our objective function should agree with that used in the global routing algorithms. The Steiner tree is used as the routing tree in most global routing algorithms, and the minimum spanning tree (MST) as the approximation of the Steiner tree[7], so that we can use the MST to estimate the total net congestion.

When a refinement region is handled, cells in this region is called moving cells, and those out of the region is called stable cells. The MSTs of those nets connecting the moving cells are the object we discuss. The edges of these MSTs can be divided into internal edges and external edges.

Definition 2: If a MST edge connects two moving cells, it is an internal edge.

Definition 3: If a MST edge connects a moving cell and a stable cell, it is an external edge.

Definition 4: There are many Gcell paths for the realization of a MST edge. The total congestion value of all Gcell boundaries on a path is defined as the cost of the path. The minimum cost among all candidate path ones for the MST edge is defined as the cost of the MST edge.

Because an external edge connects only a moving cell i, the cost of the external edge is concerned in only one variable x_i. And an internal edge connects two moving cells, cell i and cell j, so the cost of the internal edge is concerned in two variables, x_i and x_j.

Supposing cell i is located on the left of the congested boundary, let L_i be the total cost of its external edges; and if cell j is located on the right side, let R_j be the total cost of its external edges. Let C_{ij} be the total cost of those internal edges connecting cell i and cell

j. When all moving cells are allocated, the total cost of the placement is

$$f = \sum_{i=1}^{N}[L_i x_i + R_i(1 - x_i)] + \sum_{i=1}^{N}\sum_{j=1}^{N}[C_{ij}x_i(1 - x_j)] \quad (1)$$

where N is the number of the moving cells. $\sum_{i=1}^{N}[L_i x_i + R_i(1 - x_i)]$ is the total cost of external edges, and $\sum_{i=1}^{N}\sum_{j=1}^{N}[C_{ij}x_i(1 - x_j)]$ is the total cost of internal edges. L_i, R_i and C_{ij} are the constant parameters which are pre-calculated by constructing the MSTs. x_i(x_i = 0/1, i = 1, …n) are variables.

There exist the size constraints,

$$\sum_{i=1}^{N} w_i x_i \leqslant W_{\text{left}} \text{ and } \sum_{i=1}^{N} w_i(1 - x_i) \leqslant W_{\text{right}}$$

where w_i is the size of cell i, W_{left} is the size volume of the left refinement region, W_{right} is that of the right region.

Especially the datapath chip, the cells may be multi-row height and grouped. So the size constraints will be considered on several rows.

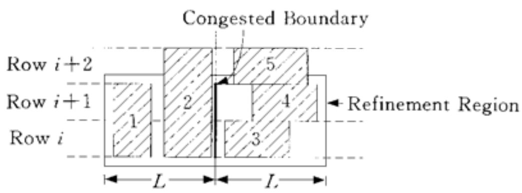

FIG. 2 Example of Multi-Row Height Cells

An example of multi-row height cells is shown in Figure 2. Cell 5 is a stable cell and the others are moving ones. The size constraints are considered on three rows and presented as below.

$$w_1x_1 + w_2x_2 + w_3x_3 \leqslant L$$
$$w_1(1 - x_1) + w_2(1 - x_2) + w_3(1 - x_3) \leqslant L$$
$$w_1x_1 + w_2x_2 + w_4x_4 \leqslant L$$
$$w_1(1 - x_1) + w_2(1 - x_2) + w_4(1 - x_4) \leqslant L$$
$$w_1x_1 + w_2x_2 \leqslant L$$
$$w_1(1 - x_1) + w_2(1 - x_2) \leqslant L - w_5$$

If moving cells are in different groups, the size constraints should be considered on all rows that the group cells occupy. An example of grouped cells is shown in Figure 3. Cell 2 and cell 7 form a group, meanwhile cell 6 and cell G form another group. The size constraints of this example is presented below.

$$w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5 + w_6x_6 \leqslant L$$
$$w_1(1 - x_1) + w_2(1 - x_2) + w_3(1 - x_3) + w_4(1 - x_4) + w_5(1 - x_5) + w_6(1 - x_6) \leqslant L$$

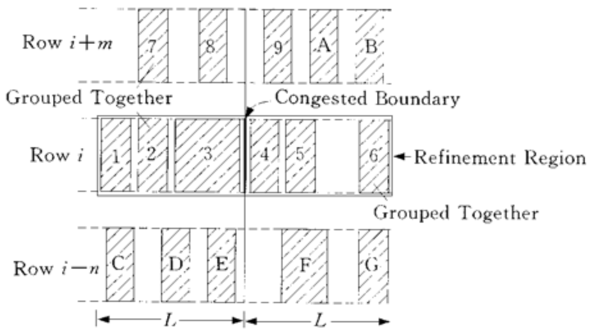

FIG. 3 Example of Grouped Cells

$$w_7x_2 \leqslant L - w_8$$
$$w_7(1 - x_2) \leqslant L - w_9 - w_A - w_B$$
$$w_Gx_6 \leqslant L - w_C - w_D - w_E$$
$$w_G(1 - x_6) \leqslant L - w_F$$

Now, we can give the mathematics model for the CEPR problem,

$$\text{Min} f(X) = \frac{1}{2}X^TAX + BX + R \tag{2}$$

$$\text{s.t.} \quad CX \geqslant D, \quad x_i = 0/1, i \in [1, n]$$

where X is the variable matrix; A, B, C, D and R are the constant matrices. This is a non-linear mathematics programming model, we will discuss its properties in section 4.

4 Method to Solve the Nonlinear Programming Problem

The Hesse matrix of $f(X)$: $\nabla^2 f(X) = A = (-C_{ij})$. When two cells are exchanged between two sides of the boundary, the MST structure will be changed, too. So C_{ij} and C_{ji} might be unequal and the Hesse matrix is asymmetry. If we set $C^*_{ij} = C^*_{ji} = \frac{C_{ij} + C_{ji}}{2}$ and generate a symmetry matrix $A^* = (-C^*_{ij})$, equation (2) can also be represented as

$$f(X) = \frac{1}{2}X^TA^*X + BX + R \tag{3}$$

Since $C^*_{ii} = 0$, A^* is not sure to be half-positive, and $f(X)$ might not be convex. It means that there exist many local optimal solutions, and it is difficult to get the global optimal solution. Adding a constant large positive number m to the $f(X)$, the global optimal solution will not be changed as well. So we can solve the new programming problem as below,

$$\text{Min} \quad f(X) = \frac{1}{2}X^TA^*X + BX + R + m$$

$$= \frac{1}{2}X^T A^* X + BX + R + m\sum_{i=1}^{N}[x_i^2 + (1 - x_i)^2] \quad (4)$$

$$= \frac{1}{2}X^T A' X + B' X + R'$$

$$\text{s.t.} \quad CX \geq D, \quad x_i = 0/1, i \in [1, n]$$

It is shown that if m is large enough, A' is half-positive, $f(X)$ is a convex function. The method to solve the convex programming problem (4) can be obtained from mathematics programming textbooks[8, 9], and we will not discuss in this paper.

5 Experimental Result

We code our algorithm in C language and implement it on some datapath test cases provided by the Arcadia Design System Inc. The result is represented in Table 1.

Table 1 Results on Some Test Cases

Case Name	Pr1388	Pr1399	Pr1460	Pr1626	Mul70
Num of Cells	58	58	261	1153	3616
Num of Nets	161	161	454	1243	5440
Num of Rows	8	8	8	23	87
Num of Groups	6	6	18	0	52
Gcell Dimension	10×8	11×8	59×8	26×23	35×35
Initial Congestion	18/0	16/0	50/60	214/22	110/0
Result Congestion	10/0	6/0	14/4	185/21	82/0
Initial Wire Length	7682/2375	8165/2500	130305/35266	35546/35352	360720/130212
Result Wire Length	7006/2375	6552/2400	124516/36618	35510/35294	361242/130140
Initial Routed Nets	147	160	444	1006	5046
Result Routed Nets	143	155	446	1008	5051
Initial Via Num	342	349	2456	5673	23293
Result Via Num	326	316	2650	5650	23276

In the "Initial Congestion" item and the "Result Congestion" item of the table, there are two numbers. The first one is the congestion degree for the vertical boundaries; and the second is that for the horizontal ones. The first number in the "Initial Wire Length" item and the "Result Wire Length" item is the total horizontal wire length of those routed nets; the second is the total vertical ones. Only those nets crossing Gcell boundaries are considered in the global routing, so the wire length and the via number in Figure 1 are not for the total nets, but only for the routed ones.

For the five cases we tested, our algorithm improves the net congestion for 45% on the average. The result wire length decreases in some examples, because more nets do not cross the Gcell boundaries in the result and will not be routed in the global routing. From these data, we can say that our mathematics model for the CEPR problem is quite good and practice.

6 Conclusion

In this paper, we gave out a nonlinear programming model for the CEPR problem. The model can be transformed to a symmetry and convex one. In the test cases we implemented, the algorithm can improve the net congestion effectively.

References

[1] N. Sherwani, Algorithm for VLSI physical design automation, Kluwer Academic Publishers, Toppan Company(s) PTE Ltd., 1996.

[2] J. Heisterman and T. Lengauer, The efficient solution of integer programs for hierarchical global routing, IEEE Transactions on Computer-Aided Design, 1991, **CAD-10**(6): 748—753.

[3] T. Gao, P. M. Vaidya and C. L. Liu, A new performance driven placement algorithm, Proceedings of International Conference on Computer-Aided Design, 1991, 44—47.

[4] M. Breuer, Min-cut placement, Journal of Design and Fault Tolerant Computing, 1977, **1**(4): 343—363.

[5] C. M. Fiduccia and R. M. Mattheyses, A linear-time heuristics for improving network partitions, Proceedings of the 19th Design Automation Conference, 1982, 175—181.

[6] A. Chatterjee and R. Hartley, A new simultaneous circuit partitioning and chip placement approach based on simulated annealing, Proceedings of Design Automation Conference, 1990, 36—39.

[7] F. K. Hwang, An $o(n\log n)$ algorithm for suboptimal rectilinear steiner trees. Transactions on Circuits and Systems, 1979, **26**(1): 75—77.

[8] M. S. Bazaraa and C. M. Shetty, Nonlinear Programming: Theory and Algorithms, Wiley, New York, 1979.

[9] G. P. McCormick, Nonlinear Programming: Theory, Algorithms and Application. John Wiley and Sons, New York, 1983.