

國立交通大學

電子工程學系 電子研究所

碩士論文

良率考量的全域繞線建立於隨機關鍵區域分析
平台上

Yield – Aware Global Routing by Stochastic
Critical Area Estimation

研究生：李杰叡

指導教授：陳宏明

教授

中華民國九十八年八月

良率考量的全域繞線建立於隨機關鍵區域分析平台上

Yield – Aware Global Routing by Stochastic Critical Area Estimation

研究生：李杰叡

Student : Jie-Ruei Lee

指導教授：陳宏明 博士

Advisor : Dr. Hung-Ming Chen



A Thesis

Submitted to Department of Electronics Engineering & Institute of Electronics

College of Electrical and Computer Engineering

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Electronics Engineering

August 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年八月

良率考量的全域繞線建立於隨機關鍵區域分析平台上

學生：李杰叡

指導教授：陳宏明 博士

國立交通大學 電子工程學系 電子研究所 碩士班

摘 要

在現今超大型積體電路的設計中，因為製程縮小而使得良率相關的問題也越來越嚴重。使得良率下降的原因有很多，隨機粒子所造成的短路與斷路就是其中一個主要的原因。在現今的流程中，我們需要等到繞線全部完成之後，才能採取一些最佳化的方法來解決這個問題，但是在這個階段可以改善的成效也不甚良好。在本篇論文中，測試例子為 ISPD2007 全域繞線比賽時所提供的檔案。我們將藉由一個有效的機率模擬方式，讓我們可以在全域繞線的階段時預測隨機粒子有可能造成問題的區域，並利用一些繞線的技巧來降低短路或斷路發生的機率，以此來提高電路的良率。我們將先用 FLUTE 程式把電路轉為一組組兩點相連的問題，再從小區域往大區域的趨勢把所有點連結，最後再把違反條件的繞線區域拔除重繞，以此來完成一組預期的全域繞線結果。實驗結果顯示，高密度關鍵區域的數量相較於 NTHU-Route 有明顯的減少。

Yield – Aware Global Routing by Stochastic Critical Area Estimation

Student: Jie-Ruei Li

Advisor: Prof. Hung- Ming Chen

Department of Electronics Engineering
Institute of Electronics
National Chiao Tung University

ABSTRACT

In modern VLSI design, the semiconductor technology has advanced toward the nanometer era. Although this advancement reduces circuit power and area, it also has many side effect issues such as the decrease in yield and reliability. There are many reasons that cause the reduction in yield and reliability and one of them is critical area problem. Critical areas are regions in which circuits are prone to short or open if random particle landed in those regions. In today's design flow, industries resolve critical area problem when design flow reaches detail routing stage. However, the improvements obtained in detail routing stage are minimal. If design flow can consider critical area problem in earlier stage, the improvement in terms of increasing yield and reliability is much greater. The benchmarks from ISPD 2007 and ISPD 2008 contest are used to verify the effectiveness of our approach. This thesis proposes an accurate probability prediction model to estimate the critical area in global routing stage and apply certain algorithms to decrease the possibility of generating critical area map in detailed routing stage. We use external algorithm FLUTE to generate Steiner Trees then apply certain routing techniques to resolve congestion and critical area issue. According to the experimental results, the number of most dense critical area has a significant reduction compare to NTHU-Route.

誌 謝

一轉眼兩年就過去了，碩士班的生活也就這樣結束了。十分開心可以處在感情這麼好的實驗室裡面，在繁忙的學業生活中也能歡樂的度過。這要感謝教授十分開明的作風，給我們學生很多自由，不只是在生活方面，在研究上有任何的想法老師都十分支持我們去嘗試。

我碩士論文這題目說大不大說小也不小，不大是因為現今很多實驗室都有解出很好的結果，不小是因為對於要白手起家來說，程式的架構複雜了一些。這邊要感謝 Simon 學長，他在程式以及演算法方面都給予莫大的幫助，謝謝便便解決所有硬體軟體方面的問題，讓我不用在花很多時間在於雜事之上。謝謝小精靈們在最後幾天解決我論文上的問題。大鬍省了我不少時間在於程序上，也謝謝希在最忙碌的時刻沒有吵鬧。最最感謝的就是益友兼良師的 Sean，他所給予我的幫助是筆墨難以形容的。



Contents

Chapter 1 Introduction	1
1.1 Organization of this thesis	1
1.2 Review of previous Work	2
Chapter 2 Preliminaries	3
2.1 Global routing	3
2.2 Basic algorithm for Global routing	4
2.2.1 Pattern routing and Monotonic routing	4
2.3 Critical area	5
2.3.1 Short and open critical area	5
2.3.2 Critical area predictor at global routing stage	6
2.4 Problem Formulation	7
Chapter 3 Methodology	8
3.1 Net Decomposition by FLUTE	8
3.2 Net ordering	10
3.3 Bounded and length condition Maze route	10
3.4 Component to Component Maze route	14
3.5 Cost function for Maze route	14
3.6 Rip up and reroute (RRR)	16
Chapter 4 Experimental Results	17
Chapter 5 Conclusion and Future Work	21
Bibliography	22

List of Tables

4.1	Comparison of routing result among NTHU router and our router	18
4.2	The number of full capacity global edge	19
4.3	Adaptec4.aplace60.2d.30.50.90	20
4.4	Newblue2.fastplace90.2d.50.20.100	20



List of Figures

2.1	Layout transfer to Grid graph.....	3
2.2	Pattern routing and monotonic routing.....	4
2.3	It will cause critical are when the center of particle in the area ShortCA(X) and OpenCA(X).....	6
3.1	our methodology flow chart.....	9
3.2	Connect A and B inside the boundary_.....	12
3.3	Set the length condition to prevent the long detour_.....	13
3.4	Detour influence the routing result_.....	13
3.5	Use Maze cost to realize Component to Component route_.....	14
3.6	Compare the routing result with cost function and length condition_.....	15



Chapter 1

Introduction

1.1 Review of previous Work

Global Routing is basically a multi-pin multi-net problem. Its goal is to connect several groups of pins without exceeding the capacity in any edge. It has been shown by Kapoor [1], that global routing problem is a NP-Hard problem. Since finding a Rectilinear Steiner Minimal Tree is already a NP-Complete problem, global routing problem is basically solving a set of RSMT problem while considering capacity constraint. The additional capacity constraint makes the global routing problem a NP-Hard problem. Early work implementing global routing is basically based on Breadth-first search such as Lee's algorithm [2], finding the shortest path from a single source to all other pins. It is also known that by including Steiner Point into a group of pins can effectively reduce total wire length. Connecting a given set of pins using shortest wire length is a Minimum Spanning Tree (MST) problem which can be solved in polynomial time, however, connecting a given set of pins with Steiner point included is a Steiner Minimal Problem (SMT) which is a NP-Complete problem. Thus, many algorithms are devoted in to solve SMT problem. One criterion determining whether a Steiner Minimal Tree is a "good" SMT is by its total wire length. Huang [3] in 1976 showed that the upper bound of the total wire length of Rectilinear Steiner Minimal Tree is total wire length of a Rectilinear Minimum Spanning Tree divided by 1.5.

On the other hand, the running time of breadth-first search based algorithm increases exponentially with increasing number of pins. As the chip size increases, it is impractical to

implement global routing entirely based on breadth-first search algorithm. There must be some reduction in the use of breadth-first search in order to decrease the running time. FLUTE [4] uses look-up tables to create a fast and great Steiner Minimal tree solution. It computes optimal Steiner trees for up to 9 pins. Most of the published routers in the recent years rely upon FLUTE.

There are many reasons that cause the reduction in yield and reliability and one of them is critical area problem. However, the improvements obtained in detail routing stage are minimal. If design flow can consider critical area problem in earlier stage, the improvement in terms of increasing yield and reliability is much greater. In this regard, this thesis proposes an accurate probability prediction model to estimate the critical area in global routing stage. And the estimate model is built from [5].

1.2 Organization of this thesis

The remainder of this thesis is organized as follows: Chapter 2 depicts preliminaries about global routing algorithm and a methodology to estimate critical area at global routing stage. Chapter 3 describes the global routing algorithm proposed in this paper and introduces two criteria that have a tremendous impact on critical area. Chapter 4 and Chapter 5 present the experiment results on critical area for this paper and compare it with other global routing algorithms.

Chapter 2

Preliminaries

2.1 Global routing overview

The problem of global routing can be characterized as follows. The layout is partitioned into rectangular tiles, and a grid graph G is constructed. Each grid tile can be considered as a set of global vertices V . The connections between different grid tiles are considered as a set of global edges. Each global edge is set to have a capacity value, which represents the number of available tracks passing through it. Overflow refers to the total number of passing tracks exceeding the capacity over the global edges. In order to guarantee the connections can be realized, overflow is desired to be as small as possible. The basic objective of global routing is to route all nets on grid graph and the capacities of all global edges are all satisfied. Under this objective, the shorter total wire length promises the better output. In 3-D routing, wire length also includes vias.

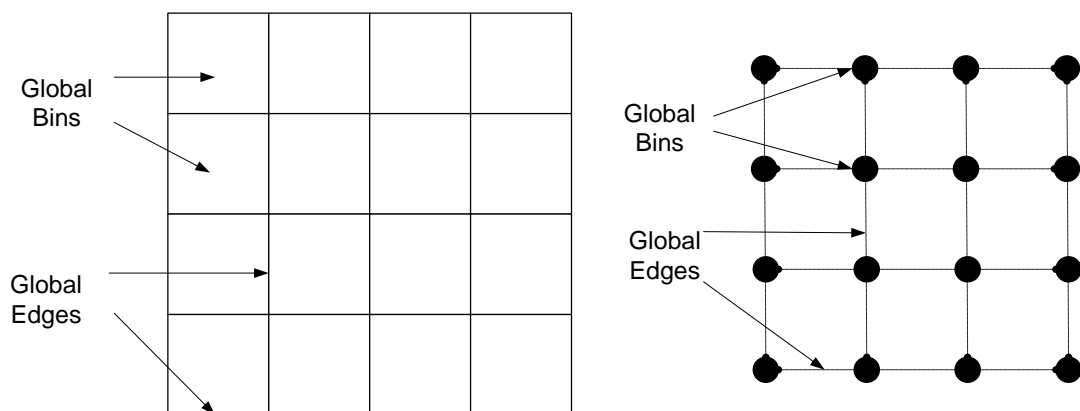


Figure 2.1: Layout transfer to Grid graph

2.2 Basic algorithms for Global routing

2.2.1 Pattern routing and Monotonic routing

Pattern routing is a method to route a two pin net with some simple shapes, such as L-shape and Z-shape. L-shape is a one bend routing and Z-shape is a two bend routing. Compare to the maze route, pattern routing is much more efficient in the run time and guarantee the wire length is the shortest. It can change the routing pattern very quickly and attempt to find a low cost combination. In order to improve the quality of routing result, FastRoute [6] proposed monotonic routing which is a new version for pattern routing. Monotonic routing can route toward the target point illustrated in Figure 2.2, so it has larger solution space than pattern route. However, the search space variation of both pattern route and monotonic route is lower than maze route. In some case, if we do not use detour pattern, we can not find a non-congestion routing tour. In our work, we propose an improved router which has the same property with monotonic routing allow can detours.

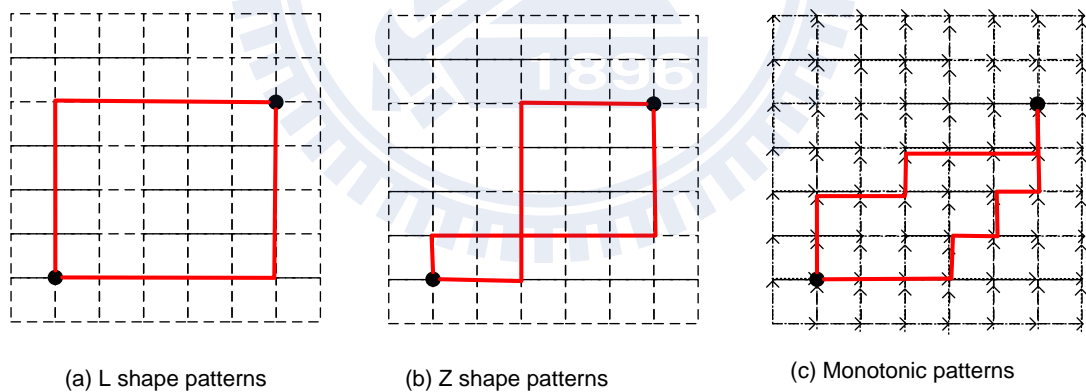


Figure 2.2: Pattern routing and monotonic routing

2.3 Critical area estimation

2.3.1 Short and open critical area

When VLSI technology moves into deep-submicron and nanometer era, the yield related problems become very important. Since the random particle occasionally occurs, when it falls into the circuit, the yield rate decreases, especially in the critical area. Two kind of critical areas, short and open, are illustrated in Figure 2.3:

Short critical area:

When a particle falls into the layout and connects two wires, a short critical area happens. W_a and W_b are two wires with a close segment S_{ab} and distance D_{ab} . X is the diameter of the particle.

$$\text{ShortCA}(X) = \begin{cases} 0 & , \quad X < D_{ab} \\ (X - D_{ab}) * S_{ab} & , \quad X \geq D_{ab} \end{cases}$$

Open critical area:

When a particle falls into the layout and cut off the connection of two wires, an open critical area. W_a is the width of the wire. L_a is the length of the wire.

$$\text{OpenCA}(X) = \begin{cases} 0 & , \quad X \leq W_a \\ (X - W_a) * L_a & , \quad X > W_a \end{cases}$$

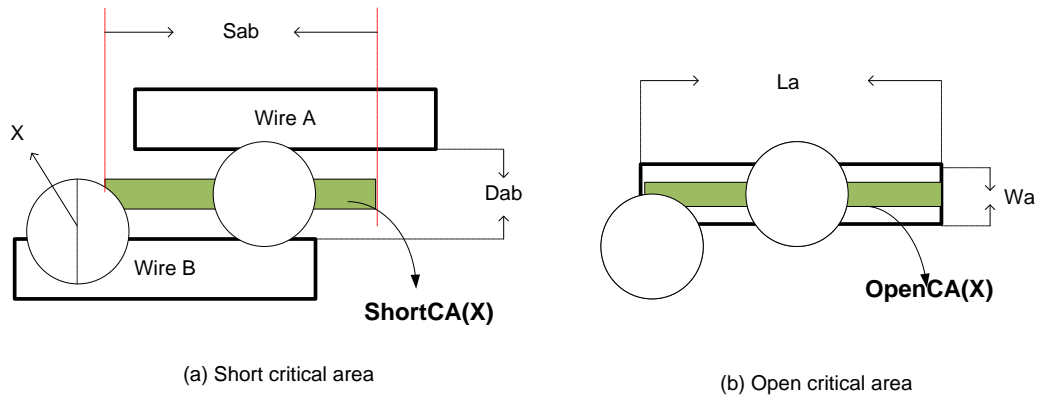


Figure 2.3: It will cause critical area when the center of particle in the area ShortCA(X) and OpenCA(X)

2.3.2 Critical area predictor at global routing stage

We get the real critical area after the post routing and critical area optimization methods (wire-spreading and wire-widening). Even we know the high probability failed area, it is too late to modify the routing result. Since in the global routing stage, we can not know the accurate critical area. We usually estimate the critical area hotspots by the congestion of one global edge, but [5] has an experiment and proves that congestion of one global edge only is not good enough. They propose a probability cost function which can find critical area accurately. In our work, we define a prediction function to find the critical area and reduce it at global routing stage.

Regarding to current global routing algorithms, very few considers critical area during global routing stage. Most of the global routing algorithms only satisfy the capacity constraint without considering the congestion condition of nearby regions. However, to reduce the probability of occurrence of critical area, the congestion condition of nearby regions must be taken into consideration when evaluating critical area. A grid cell with low wire density doesn't necessary means it will have lower chance to have critical area since nearby region may have very high wire density that wires from adjacent grid cell may migrate over during detail routing stage. Vice versa, a grid cell with high wire density doesn't necessary means it

will have higher chance of having critical area since nearby grid cell may have very low wire density allowing wire to migrate over to adjacent grid cells during detail routing stage. Viewed in this light, the wire density of nearby regions is an absolute essential element and must be taken in to consideration to reduce critical area. Thus, [5] proposed a quite complex mathematical model that can evaluate the probability of occurrence of critical area after post routing stage. The mathematical model in [5] is too complex and thus impractical to be incorporated into global routing algorithms. In section 3.5, there is a simplified model is used and is incorporated in the cost function that serves as a parameter that evaluates the critical area.

2.4 Problem Formulation

Critical area is defined as a particular region in chip in which it has relatively higher probability that causes chip to malfunction. Usually the reason to malfunction is because wires are placed so closely together such that random particles landed on these regions may create open or short circuit. The goal of this paper is to devise a global routing algorithm that can successfully route all of given nets and reduce the probability of occurrence of critical area in layout while satisfying the capacity constraint. The reduction of the probability of occurrence of critical area is achieved by reducing the number of most dense critical area grid cell.

Chapter 3

Methodology

In this chapter, a global routing algorithm that considers the impact of critical area while solving the congestion level is proposed. Our methodology flow chat is illustrated in Figure 3.1. The organization of this chapter is described as follows. The methodology begins by decomposing multi-pin nets into a collection of two-pin nets using external Steiner Tree generator algorithm called FLUTE [4] described in Section 3.1. Section 3.2 describes the impact of net routing order on congestion level and methodologies on how to determine net routing order. Section 3.3 introduces a bounded routing method and wire length restriction method which can easily realize monotonic routing and avoid some undesirable routing result. In section 3.4, a modified routing algorithm is implemented to achieve component-to-component routing that can efficiently reuse the routed resources. Section 3.5 depicts a modified cost function for maze routing algorithm in attempt to decrease the critical area. Section 3.6 describes a post routing algorithm that serves to solve congested or unsolvable nets after maze route.

3.1 Net Decomposition by FLUTE

Regarding to recent publications on global routing, FLUTE [4] is widely used by many global routing algorithms to generate Steiner Tree due to exceedingly fast speed with fairly good quality. The basic idea of FLUTE [4] is that it first generates numerous combination of RSMT with very low order and stores the result in a table. Then when one need to generate a RSMT with a given order, FLUTE simply looks up the table and find a suitable result that

satisfy the request. Since speed is a very important criteria in global routing stage. In our methodology, FLUTE is used to generate RSMT for every single multi-pin net. After RSMT is generated, the RSMT is decomposed in to a set of two pin edge. In theory, the total wire length of RSMT should be shorter than the total wire length of minimum spanning tree. However, the extra Steiner point added during the construction of RSMT may increase the number of congestion hot spot.

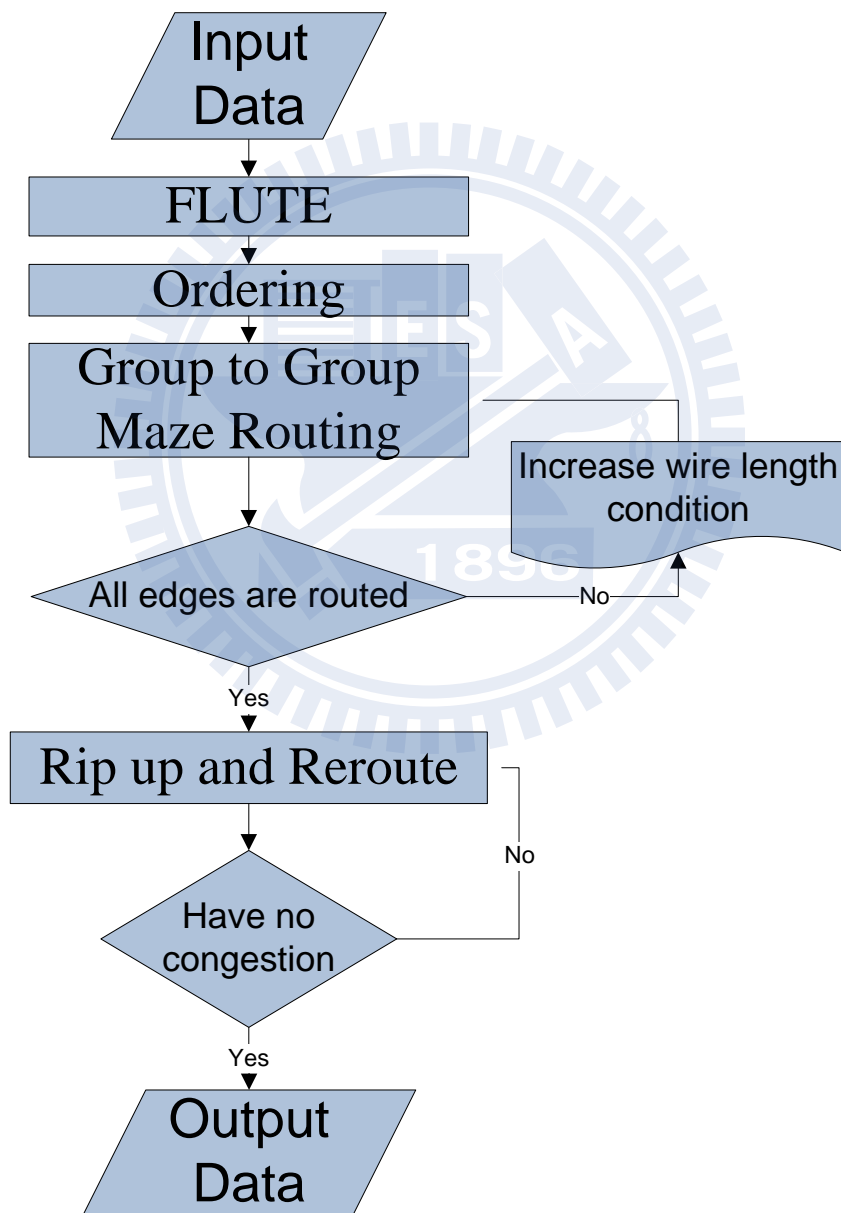


Figure 3.1: our methodology flow chart

3.2 Net Ordering

After the RSMT is constructed by FLUTE, the RSMT is transformed into a set of two-pin edges. The problem of routing a multi-pin net now becomes routing a set of two pin edges which simplifies the implementation of the algorithm but reduces the flexibility of the result. From experimental observations, net routing order has a significant impact on the quality of the result. Determining the net routing order is an NP hard problem, disregarding net routing order may deteriorate the quality of the routing result to certain extent. However, in global routing stage, it is impractical to invest a significant portion of computing effort on determining net routing order. In this thesis, the net routing order is determined based on their size of the bounding box which is also the Manhattan distance of the two most separated points in the net. The routing order of a particular net is based on how small of the bounding box, the smaller the bounding box a net has, the higher priority the net has in the routing order. Using Manhattan distance to determine the net routing order provides certain qualities. First, the consumption of routing resources can be limited at local aspect then progressively moving towards entire chip. Second, since local edges are given with higher routing priority, it reduces the chance of long distance detouring in some aspects.

3.3 Bounded Maze Route and Length Restriction Maze Route

The quality after initial route determines the burden left for post routing algorithm. The better quality an initial route is, the fewer number of unrouted edges. A good initial routing result can reduce the number of the rip up and reroutes (RRR) iterations. The conventional routing algorithm for a two pin edge is Lee's Algorithm or Maze Route which is widely adopted and modified in many recent global routing algorithms. Maze Route guarantees that a shortest path between two pin can be found if such path exists. However, the time complexity

of Maze Route to solve a single two pin edge in three dimensional space is $O(n^3)$ which is considered very impractical in terms of computation time considering that the order of the input nets is in millions. In addition, most of the routing paths are quite easy to determine and does not need routing algorithms that burdens computational resource. Viewed in this light, a simple and effective routing algorithm must be adopted to solve simple edges and use Maze Route only on difficult edges. Hence, regarding that most of the routing paths are either L-shaped or Z-shaped, a specific routing algorithm can be devised to quickly detect whether if there is an L-shaped or Z-shaped solution exist for a given edge. An improved version of L-shaped and Z-shaped routing algorithm is monotonic route which iterates every possible L-shaped and Z-shaped routing solution within the bounding box of an edge. Monotonic Route is a much quicker routing algorithm compare to Maze route, however, it does not guarantee a solution can be found since the solution space and routing direction of Monotonic route is confined.

One of the major drawbacks of Maze Route is that it only considers one edge at a time and will attempt to reach a routing solution at all cost. However, to achieve a globally optimum solution, the congestion condition of nearby region and consumption of routing resources should be taken into consideration. Greedily using all possible routing resources to reach a routing solution should be carefully avoided since it will increase the probability of failed route for subsequent edges. If routing resource are greedily consumed at the very beginning, it is almost guaranteed that there will be unroutable edges in the end. Thus, it is absolutely essential to control the consumption routing resource at the very beginning.

In this thesis, two criteria are introduced to control the consumption of routing resource. The first criterion is the routing space of an edge. The routing space of an edge must be defined within a specific boundary such that the maximum routing space to route each edge is

controlled. Figure 3.2 is an illustration of the confinement of routing space. For each edge, the routing path is limited within the routing boundary and gradually expands the boundary if a legal routing result cannot be found. Confining the routing space allows algorithm to have some control over routing detour length and prevents greedily consuming too much routing resources. This method can efficiently reduce the running time to route each edge and provides control over routing detour length so that congestion area can be minimized.

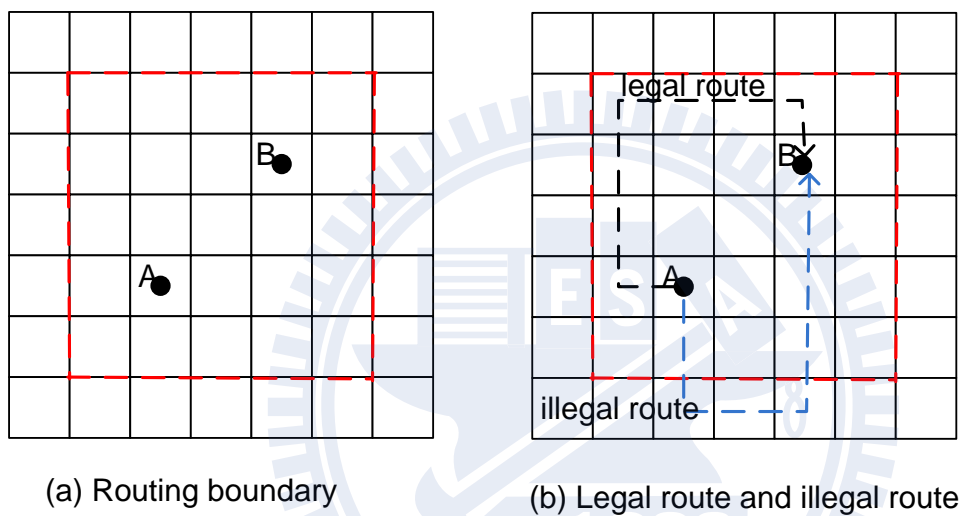
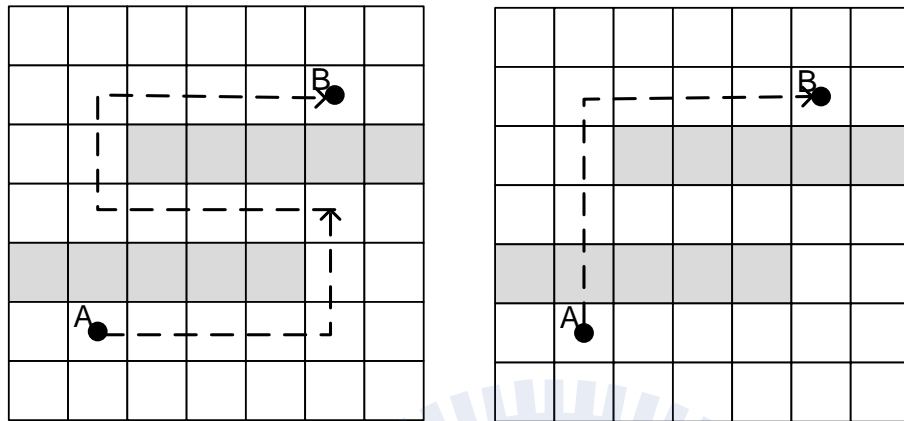


Figure 3.2: Connect A and B inside the boundary

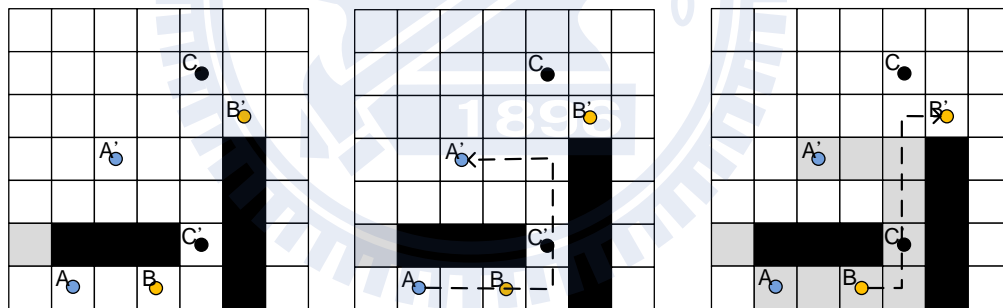
The second criterion is the restriction on detour length of an edge. In Figure 3.3, gray grid cell has less routing resource compared to white grid cell. If there are no restrictions imposed on the detour length of an edge, it is possible to get an undesirable routing result illustrated in Figure 3.3. Imposing restriction on the detour length can define that the lower bound of the routing solution. If a legal solution cannot be reached under certain detour length restriction, then we gradually increase the detour length restriction. The increment on detour length must be carefully controlled. If increment on detour length is too large, then optimal solution may be overlooked. On the other hand, if increment is too small, then it will occupy too much

computation power. By carefully controlling the two criteria, routing space and detour length of an edge, routing resource can be effectively controlled for each edge offering more routing resource for subsequent edges (Figure 3.4).

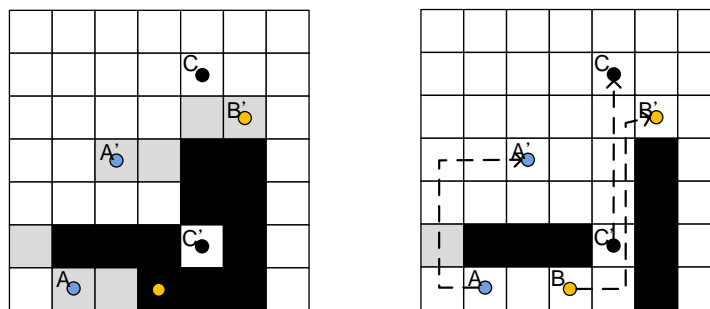


(a) Crazy detour without length condition (b) Set length condition equal to the Manhattan distance

Figure 3.3: Set the length condition to prevent the long detour



(a) Connect A, B and C nets, black block has no capacity, grey block can pass through 1 track, white block can pass through 2 tracks (b) Connect A net first and have no length condition (c) Connect B net



(d) Net C have no legal tour. (e) Set the length equal to the Manhattan distance

Figure 3.4: Detour influence the routing result

3.4 Component to Component Maze route

When routing each edge, a legal path is a path that connects source pin and target pin while satisfying the capacity constraint. One of the important aspects in global router is that edges that belong to the same net can share the same routing path, thus sharing routing path for different edge under same net should be extensively exploited since it offers edge to reuse the routing resource.

In this thesis, when routing a specific edge, nearby routed paths that belongs to the same net is evaluated. This is accomplished by setting an extremely low cost for all of the other routed paths that belongs to the same net. This encourages routing path to take on path that belongs to the same net even though it may have a longer detour length.

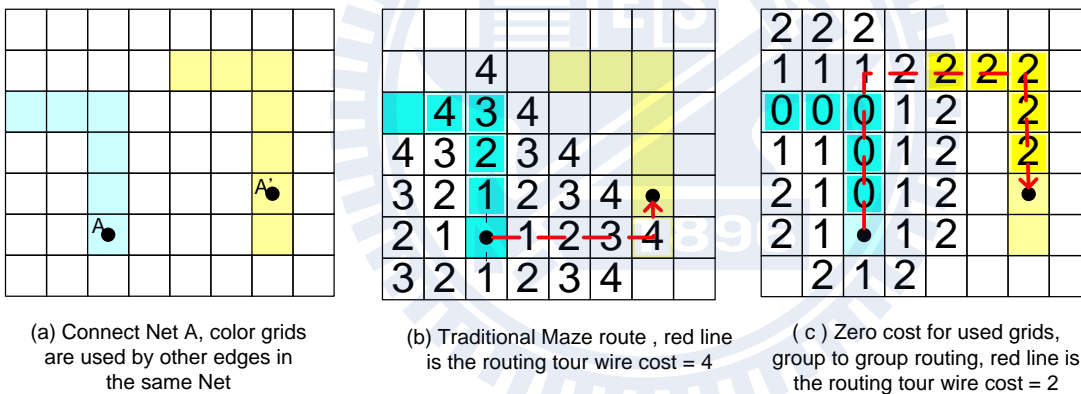


Figure 3.5: Use Maze cost to realize Component to Component route

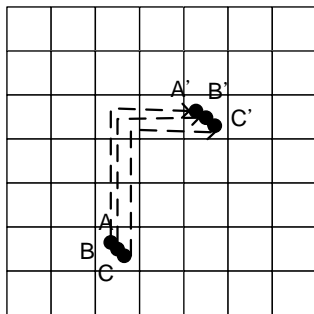
3.5 Cost function for Maze route

One of the major goals in this thesis is to decrease the critical area. The critical area is very large if the routing result is very congested. Some routing path must be detoured at the expense of longer detour length to avoid critical area even if satisfying capacity constraint. In Figure 3.6(a), there are three edges need to be connected. In conventional Maze Route, since each edge is routed independently without considering nearby congestion condition, each

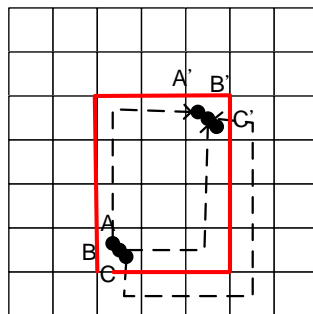
edge will attempt to reach the target pin with minimum distance. Although the routing result is legal, it increases critical area. To take in consideration on critical area, the cost function on routing algorithm must include nearby congestion condition.

$$\text{Cost} = \alpha * \text{Cap}^2 + \beta * \text{CapNeighbor}^2 + \gamma * \text{Manhattan}(\text{Target}) + \text{MazeCost}/\text{ViaCost} \quad (1)$$

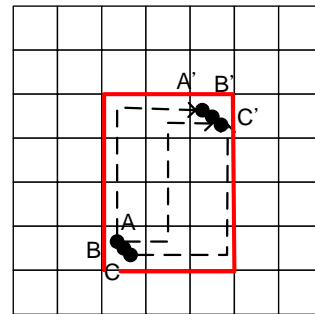
In (1), Cap is the number of used routing capacity. CapNeighbor is the number of neighbor's capacity. α , β and γ are user defined parameters, when global edge's resided capacity is below 10%, α and β are increased. In this way, the router will avoid choosing congested region when routing each edge. The drawback on considering critical area in during routing stage is that decreasing of critical area comes at the expense of increasing detour length. Figure 3.6(b) is an illustration on how detour length might increase by considering critical area. Imposing restriction on detour length described in Section 3.2 can alleviate the situation. Figure 3.6(c) illustrates a desirable routing result after using the cost function. MazeCost is a parameter in traditional Maze route. MazeCost is used to mark the cost for movement in x-axis and y-axis direction. ViaCost is used to mark the cost for movement along z-axis direction. ViaCost is set larger than MazeCost such that routing to another layer is avoided as much as possible. Manhattan distance between the current cell and target pin is also added to the cost function to increase routing speed.



(a) Three nets route the same tour under enough capacity



(b) Use CA cost function without length condition. Nets are separated, but C has detour



(c) Use CA cost function with length condition. Nets are inside the boundary

Figure 3.6: Compare the routing result with cost function and length condition

3.6 Rip up and reroute (RRR)

In the post routing stage, an aggressive and greedy algorithm is devised to route edges that can not be solved in the initial routing stage. In initial routing stage, the routing resource is abundant and carefully controlled. There are many legal detour paths to reach the target pin and it is very difficult to determine which detour path will lead to a better routing result. It is possible that one of the detour paths may come across a very congested region where some edges that cross the same region needs to be rerouted to avoid the congested region. However, rerouting an edge to avoid one particular congested region is also likely to create another congested region in some other part of the chip. In this regard, net ordering has a significant impact on the routing result since some routing paths may cross over numerous congested regions. It is then favorable to the routing algorithm to determine which edge has the most impact on nearby edges. Some research use statistic prediction to construct an initial congestion map, but this approach only offers a rough sketch on the overall congestion level and is not very accurate.

During the post routing stage, the edges left unsolved usually are the toughest edges to be routed. When routing these tough edges, rip up some of the routing path that overlaps with the current routing edge. The ripped up edges are then stored in to a container with a specific ordering pending for the next reroute. After some of the routing paths are being ripped, it provides more routing resource for the unsolved tough edges. This aggressive and iterative ripping up edges to provide routing resource for other edge is called rip up and reroute.

Chapter 4

Experimental Results

The algorithm was implemented using C++, and the experiments were conducted on a machine equipped with Intel Xeon 3.00GHz processor and 4096MB RAM in a Linux environment. Two 2-D benchmarks, adaptec4 and newblue2, of the ISPD07 benchmarks were applied. Adaptec4 had 515 thousand nets, and newblue2 had 462 thousand nets. The benchmarks were of the Labyrinth format, which were ordinary ASCII text files. The input files included the size and shape of the global routing graph, the edges capacity, the number of signal nets, and the number of pins and their actual positions. Labyrinth format facilitated the overall implementation process as applied to a global router. Thus, this format has made itself popular for many academic tools, especially for its applicability in CAD contest. The output file format was a variation of the BoxRouter [7] format, which described the connection relationship from net to net. It is worth to mention that additional attention had been paid to the output format. If we write the same tour on a net twice, the evaluating tool would count the cost twice, too. This would then result in erroneous outcome. Therefore, we must check for correctness of our routing output by evaluating the Perl script which was released on the IDPD2008 contest. By doing so, the evaluating tool would also count the maximum overflow numbers, total overflow numbers and total wire length. Since the benchmarks are too large, there seemed to be no other more efficient way but this to verify the correctness.

We compared our results with the state-of-the-art global router, NTHU-Route [8], which was in the first place on ISPD2008 Global routing contest. Also, we had got the latest version, NTHU-Route 2.0. Table 4.1 compares the routing results of NTHU-Route [8] performing the 2-D ISPD07 benchmarks with our router's. In adaptec4, our total wire length is 11.1 million. In adaptec2, our total wire length is 6.3 million. Compared with NTHU-Route's [8] total wire length, the total wire length of our work is slightly more than 8.8% ~ 10.5%. Although the wire length seems not good enough, we develop a novel criterion which emphasizes greater importance in terms of the consideration of critical area.

Table 4.1: Comparison of routing result among NTHU router and our router

Input file	Net	Our router	NTHU-Route [8]	Overhead
		Total wire length (million)		
adaptec4-2d	515304	11.1	10.2	8.8%
newblue2-2d	463213	6.3	5.7	10.5%

The probability of critical area has strong relation with congestion. Some researches have made use of congestion as the indicator for critical area hotspots. Generally speaking, if one global edge occupies the rest of the capacity, conventional commercial tools would have utilized some methods to decrease the probability of this critical area. One way to do so is by wire-spreading; another may involve wire-widening. No matter which solution is taken, if this global edge takes out all the capacity left, this spot would have little probability for its associated congestion to be improved. Thus, this scenario is termed as the critical area problem. In Table 4.2, the overall criticality is measured by counting the number of spots which have full global edges. In benchmark adaptec4, our router had 145 full capacity edges, and NTHU-route [8] had 3297 full capacity edges. In benchmark newblue2, we have 0 full capacity edges, and NTHU-route [8] has 2096 full capacity edges. In short, our router out-performed NTHU-route [8] by generating less full global edges: less 3135 edges in

adaptec4 and less 2096 edges in newblue2.

Table 4.2: The number of full capacity global edge

	Our router	NTHU-Route [8]	Reduction
Input file	Number of full global edge		
adaptec4-2d	145	3297	-3135
newblue2-2d	0	2096	-2096

In [5], their experiment proved that considering only those of net edge capacity of one is less discerning as evidence for the hotspot area. The definition of a hotspot means that the region might be a critical area. But the congestion value is the same for those regions with both large and medium critical area values after post-routing wire-spreading or wire-widening. We need to count not only the original congestion number, but also the congestion numbers of the surrounding neighbors. Table 4.3 and Table 4.4 are listed the global edge numbers for some cases within which critical areas are highly probable.

The following paragraph defines the meaning of symbols used in these tables. The first column of these tables enumerates possible combinations of congestion of three blocks. F indicates that the capacity of this block is fully occupied. That is, the rest of the capacity for possible global edge is zero; this block cannot have any routing tour anymore. 1 represents that there still have capacity for one global edge, 2 for capacity for two global edges, and so on. For the blocks in the middle of the three blocks, its congestion number would be interpreted as the global edge which we want to take into account for the calculation of the critical area probability. For the tow blocks on the either sides, their congestion numbers are neighbor's congestion. Table 4.3 is the results for the benchmark adaptec4. The experiments showed that our router would not generate the case of which all three blocks are fully occupied. On the other hand, NTHU-Route [8] has 143 of such worst cases. Our router could generate less edge numbers than NTHU-Route [8] in every case, and our router could generate 913 less overall number of global edges.

Table 4.3: The experimental result for adaptec4.aplace60.2d.30.50.90, [FFF] represents three adjacent global edges' capacity number. F (full) means all capacity in the global edge is used. [1F1] indicates that still have one track in two side's edge, but the capacity for center edge is full.

	Our Results	NTHU-Route [8]	Reduction			
<table border="1"><tr><td>F</td><td>F</td><td>F</td></tr></table>	F	F	F	0	143	-143
F	F	F				
<table border="1"><tr><td>1</td><td>F</td><td>F</td></tr></table>	1	F	F	14	147	-133
1	F	F				
<table border="1"><tr><td>1</td><td>F</td><td>1</td></tr></table>	1	F	1	37	137	-100
1	F	1				
<table border="1"><tr><td>2</td><td>F</td><td>1</td></tr></table>	2	F	1	9	146	-137
2	F	1				
<table border="1"><tr><td>F</td><td>1</td><td>F</td></tr></table>	F	1	F	21	212	-191
F	1	F				
<table border="1"><tr><td>F</td><td>1</td><td>1</td></tr></table>	F	1	1	50	259	-209
F	1	1				
Total	131	1044	-913			

In table 4.4, we have zero global edge for the cases, and NTHU-Route [8] has 30 global edges in the cases.

Table4.4: The experimental result for newblue2.fastplace90.2d.50.20.100

	Our Results	NTHU-Route [8]	Reduction			
<table border="1"><tr><td>F</td><td>F</td><td>F</td></tr></table>	F	F	F	0	1	-1
F	F	F				
<table border="1"><tr><td>1</td><td>F</td><td>F</td></tr></table>	1	F	F	0	3	-3
1	F	F				
<table border="1"><tr><td>1</td><td>F</td><td>1</td></tr></table>	1	F	1	0	2	-2
1	F	1				
<table border="1"><tr><td>2</td><td>F</td><td>1</td></tr></table>	2	F	1	0	5	-5
2	F	1				
<table border="1"><tr><td>F</td><td>1</td><td>F</td></tr></table>	F	1	F	0	6	-6
F	1	F				
<table border="1"><tr><td>F</td><td>1</td><td>1</td></tr></table>	F	1	1	0	13	-13
F	1	1				
Total	0	30	-30			

Chapter 5

Conclusion and Future Work

We proposed a methodology to prevent critical area at global routing stage. We use an improve Maze route and a cost function for critical area to get a non-congestion and low critical area result. The improved Maze route can realize component to component route easily and have better performance than monotonic routing. We compare our result to NTHU-Route2.0 [8] [9] and have less high probability critical area edges.

The reason we use maze route is that we want to get a very good initial route. A good initial route can greatly decrease the iteration of rip up and reroute. In the future, the most important thing is to solve the congestion in the big cases, and we want to add more random skills into our work to prevent the local optimization.

Bibliography

- [1] S. Kapoor, "Topics and Analysis of Combinatorial Algorithms," Ph.D. Thesis, Dept. of Computer Science, Univ. of Illinois, Urbana, IL, 1986
- [2] C. Lee, "An algorithm for path connections and its applications," in IRE Transactions on Electronic Computing EC-10, pp. 346–365, 1961.
- [3] Huang, "On steiner minimal trees with rectilinear distance", in Society for Industrial and Applied Mathematics, Vol. 30, No. 1 pp. 104-114, Jan 1976
- [4] C. C. N. Chu and Y.-C. Wong, "FLUTE: fast lookup table based rectilinear Steiner minimal tree algorithm for VLSI design," in TCAD, vol. 27, no. 1, pp. 70-83, 2008.
- [5] Subarna Sinha and Charles C. Chiang, "A methodology for fast and accurate yield factor estimation during global routing" in *Proc. Int. Conf. on Computer-Aided Design*, pp. 481–487, 2007
- [6] M. Pan and C. Chu, "Fastroute 2.0: A high-quality and efficient global router", in Asia South Pacific-DAC '07: pp. 250-255, 2007.
- [7] M. Cho, K. Lu, K. Yuan and D. Z. Pan, "BoxRouter2.0: Architecture and Implementation of Hybrid and Robust Global Router", in *Proc. Intl. Conf. on Computer-Aided Design*, pp. 503-508, Nov 2007.
- [8] J. -R. Gao, P.-C Wu and T.-C Wang, "A new global router for modern designs", in *Proc. Asia and South Pacific Design Automation Conf.*, pages 232-237, 2008
- [9] Yen-Jung Chang, Yu-Ting Lee, and Ting-Chi Wang, "NTHU-Route 2.0: A Fast and Stable Global Router," in Proceedings of International Conference on Computer-Aided Design (ICCAD), pages 338-343, November 2008