

國立交通大學

資訊科學與工程研究所

碩士論文

考慮干擾效應非點格式繞線系統的研發與整合

Development and Integration of a Crosstalk-Driven Gridless
Routing System

研究生：張祐寧

指導教授：李毅郎 教授

中華民國九十六年八月

考慮干擾效應非點格式繞線系統的研發與整合

Development and Integration of a Crosstalk-Driven Gridless
Routing System

研究生：張祐寧

Student : Yu-Ning Chang

指導教授：李毅郎

Advisor : Yih-Lang Li

國立交通大學

資訊科學與工程研究所

碩士論文



Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

August 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年八月

考慮干擾效應非點格式繞線系統的研發與整合

研究生：張祐寧

指導教授：李毅郎 博士

國立交通大學 資訊科學與工程研究所

摘要

隨著超大型積體電路製程技術邁入奈米時代，使得電子裝置的大小及線路的寬度都隨之縮小，而且在相同層中，線路間的距離也變得越來越近。一條不良繞線品質的長訊號(訊號線長度過長以及與相鄰導線平行長度過長而引發的耦合效應)將會產生過多的延遲。因此在高速超大型積體電路的設計中，想辦法避免或滿足電磁干擾效應的重要性也隨之提升。然而，在傳統的兩階段繞線：全域繞線(Global routing)和精細繞線(Detailed routing) 流程中，要解決這樣的問題會使得整個流程變得複雜且沒有效率。因為在全域繞線中，沒有電路線軌的資訊，所以很難去考量電磁干擾現象；而在精細繞線這本身就十分耗時的階段去考量此問題，只會增加它大量的計算，使它的負擔變的更重。因為這些原因，有人便提出了在全域繞線及精細繞線中併入一個中間的步驟，稱之為電路線軌指派(Track assignment)。

這份論文將著重於非點格式繞線系統，並且將線軌指派演算法整合入傳統的兩階段非點格式繞線系統。在最後的實驗數據中，將會發現在非點格式繞線系統中，比起傳統的二階段繞線系統，三階段繞線系統將得到較快的繞線速度，另外也針對非點格式線軌指派階段發展減少耦合效應的指派演算法以產生較佳的結果佈局。

關鍵字：系統晶片設計，非點格式繞線，全區域繞線，細部繞線，軌道指派。

Development and Integration of a Crosstalk-Driven Gridless Routing System

Student: Yu-Ning Chang

Advisor: Dr. Yih-Lang Li

**Institute of Computer Science and Engineering
National Chiao Tung University**

Abstract

As the VLSI manufacturing technology advances to the Very Deep Submicron (VDSM) era, the device feature size shrinks and the minimum separation between two wires of the same layer is getting closer. The bad-quality routing of a long wire produces excessive delay. Therefore, avoiding crosstalk for high-speed VLSI design is of growing importance. However, it is complicated and inefficient to solve the problem in conventional two-stage flow. The difficulty of minimizing crosstalk during global routing is that nets have no track information at this stage, and detailed routing is a time-consuming task. Therefore, the track assignment, an intermediate stage between global and detailed routing, is incorporated with the routing flow.

This work will focus on gridless routing system, and integrate an efficient crosstalk-driven routing system, including a congestion-driven global router, a crosstalk-driven TA (GTA) and enhanced NEMO with fast PMT extraction. Experimental results show that the three stage routing is faster than the two stage routing on gridless system. In addition, a crosstalk-driven gridless track assignment will reduce crosstalk and receive a better routing result.

keywords: SOC design, gridless routing, global routing, detailed routing, track assignment.

Acknowledgement

I am deeply grateful to my advisor, Dr. Yih-Lang Li for his continuous guidance, support, and ardent discussion throughout this research. His valuable suggestions help me to complete the thesis. Also I express my sincere appreciation to all classmates in my laboratory for their encouragement and help.

This thesis is dedicated to my parents and my families for their patience, love, encouragement, and long expectation.



Contents

| | |
|---|-----------|
| Abstract (in Chinese)..... | I |
| Abstract(in English)..... | II |
| Acknowledgements..... | III |
| List of Figures..... | V |
| List of Tables..... | VI |
| | |
| 1 Introduction | 1 |
| | |
| 2 Preliminary | 4 |
| 2.1 Problem Formulation and Crosstalk Model | 4 |
| 2.2 Gridless Track Assignment | 6 |
| 2.3 NEMO Overview | 8 |
| | |
| 3 Integration | 9 |
| 3.1 System Flow..... | 9 |
| 3.2 Extracting IRoutes and Layer Assignment | 10 |
| 3.3 Extended O-Tree Based Assignment Refinement (EOBAR) | 13 |
| 3.4 Routing Tree Construction | 14 |
| 3.5 Pattern Routing | 16 |
| | |
| 4 Experimental Results | 17 |
| | |
| 5 Conclusions and Future Work | 22 |
| | |
| 6 Reference | 23 |

List of Figures

| | |
|---|----|
| 1. Problem Formulation and an Example of GTA | 4 |
| 2. An Example of Separation Rule | 5 |
| 3. An Example of Initial GTA | 6 |
| 4. An Example of IRoute Refinement | 7 |
| 5. An Example of Sub-Panel Rearrangement | 7 |
| 6. The Final Result after GTA Algorithm | 8 |
| 7. The System Flow of Three-stage Gridless Routing System | 9 |
| 8. Extracting IRoutes (1) | 10 |
| 9. Extracting IRoutes (2) | 10 |
| 10. Extracting IRoutes (3) | 11 |
| 11. Layer Assignment | 11 |
| 12. The Reduction Ratio of O-tree Refinement | 12 |
| 13. An Example of 4 Nets after GTA and Its Final Routing Result | 14 |
| 14. Routing Tree Construction (1) | 14 |
| 15. Routing Tree Construction (2) | 15 |
| 16. Routing Tree Construction (3) | 16 |
| 17. The Layout of Routing Results | 21 |

List of Tables

| | |
|--|----|
| 1. Benchmark Circuits Statistics for Full-chip Routing | 17 |
| 2. CPU Time and Memory Usage | 17 |
| 3. Statistics of Crosstalk Reduction for Fixed- and Variable-rule Routing | 18 |
| 4. Comparison of Routing Time between Our Work and NEMO | 19 |
| 5. Statistics of Completed IRoutes and Point-to-Point Routing by GTA and Pattern Routing | 20 |
| 6. Comparison of Wire Length between Our Work and NEMO | 20 |
| 7. Comparison of Crosstalk Reduction between Our Work and Commercial Tools | 21 |



Chapter 1

Introduction

In general, the traditional routing system includes global and detailed routing [1]. At the preceding stage, the entire routing region is divided into several sub-regions, i.e., global cells (GCells). Every sub-region has a capacity denoting the maximum number of nets that can pass through it. For each net, the global router will find a global path, a subset of global cells connecting the terminals of the net. After the global routing is finished, the detailed routing determines the actual wire position within the assigned global cells from the global router. As the interconnection complexity rises more and more, the noise between wires needs to be reduced. In traditional two-stage routing flow, since global routing doesn't determine the real position for each net and detailed routing is a time-consuming task, the track assignment (TA), i.e., an intermediate stage between global routing and detailed routing, is declared and put into two-stage routing system. Miscellaneous propagation in detail routing will be omitted, and we will get more straight wires than the maze algorithm [2] because straight wires are assigned in advance. In addition, variable width and space interconnection design is more flexible in reducing the wire resistance and crosstalk effect. For variable rule, gridless routers get better utilization of routing area than grid routers. Two well-known gridless routers are tile-based and implicit connection-graph-based routers, which possess the advantages of low path propagation complexity and fast routing graph construction, respectively [3][4][5].

Before the work of [2], the application of TA has already been applied to in order to improve via minimization on completed routing design [6][7][8]. Crosstalk reduction during layer assignment is addressed in [9], where TA is introduced to

distribute crosstalk sensitive nets to different layers. In the approach of Batterywala et al. [2], TA extracts IRoutes from every net first, where IRoutes must pass through at least two global cells. In the following work, TA places as many IRoutes as possible on available tracks. A horizontal constraint graph called IRoute overlap graph (OLG) is established and determine whether two IRoutes can be assigned in a same track. The largest clique in an OLG is discovered and TA will assign them to different tracks. In order to finish this job, the assignable track of each IRoute is thus represented by a bipartite graph based on the pre-placed blockages and assigned IRoutes on each track. By weighted bipartite matching algorithm, Result of assigning the current clique can be done. IRoutes are processed clique by clique. With this method, TA completes the routing of most long nets, and let detailed routing be simpler. The research by Ho et al. [10] first involves developing the crosstalk-driven TA in a multilevel routing system. Crosstalk is reduced by identifying a minimum-cost (IRoute overlap length) Hamiltonian path of the currently processed clique obtained from the IRoute OLG. In short, TA provides a good platform for crosstalk reduction in a routing system, and significantly speeds up the routing runtime by 1.3–2.5 times, as revealed in [2]. Cong et al. [5] presented a three-stage routing system, comprising performance-driven global routing, congestion-driven wire planning considering variable-rule wires and gridless detailed routing. Wire planning is first applied to all nets one by one before detailed routing to determine the passing regions for each net. Wire planning also helps an incomplete net to refine its new passing regions, by viewing previously completed wires as new obstacles. However, wire planning only determines passing regions of a net instead of its placed track, and processes one net at a time.

This work integrates an efficient crosstalk-driven gridless routing system, including a congestion-driven global router, a crosstalk-driven gridless TA (GTA) and

enhanced NEMO with fast PMT extraction. After finishing congestion-driven global router, we extract the longest Segment (IRoute) possibly for each net. If originally two shorter segments belong to the same Panel, we will combine these two segments and get a longer IRoute. For this reason, we can get a better connection result between two shorter segments. All Pins close to the long IRoute will get a simple point-to-path connection. The GTA in this work has two phases, initial GTA and crosstalk reduction. At the first stage, a left-edge like algorithm is invoked to obtain an initial TA result quickly. The problem of re-assigning IRoutes for crosstalk reduction is then transformed into a restricted non-slicing floorplanning problem. A fast deterministic floorplanning algorithm is employed to replace each IRoute in the position with most decreased overlapping length. After refining the position for each IRoute, every panel is treated as a collection of sub-panels, each of which can also be considered as a hybrid IRoute with different states on its top and bottom borders for a horizontal IRoute. Re-ordering the sub-panel can further decrease the total overlapping length. Before detailed routing, routing tree construction is undertaken for placed IRoutes and other pins; many original point-to-point routings are set to connect to IRoutes, and can be completed simply with pattern routing. For detailed routing, we use a rapid PMT extraction method to boost path propagation. Experimental results demonstrate that the proposed gridless routing system can perform over 3.24 times faster for fixed- and variable-rule routings than an implicit connection-graph-based router, NEMO, with over 70% reduction in the overlapping length of adjacent wires. Finally we compare this work and commercial tools, experimental results indicate that this work and commercial tools get approximate results as considering crosstalk reduction.

Chapter 2

Preliminary

2.1 Problem Formulation and Crosstalk Model

Figure 1 displays a routing example of 17x5 GCells at its bottom of the figure. A panel comprises a series of connected GCells in a row or a column. The topmost horizontal panel contains 6 IRoutes. GTA has no track to be used. One assignment of the topmost horizontal panel is displayed on the top of the figure.

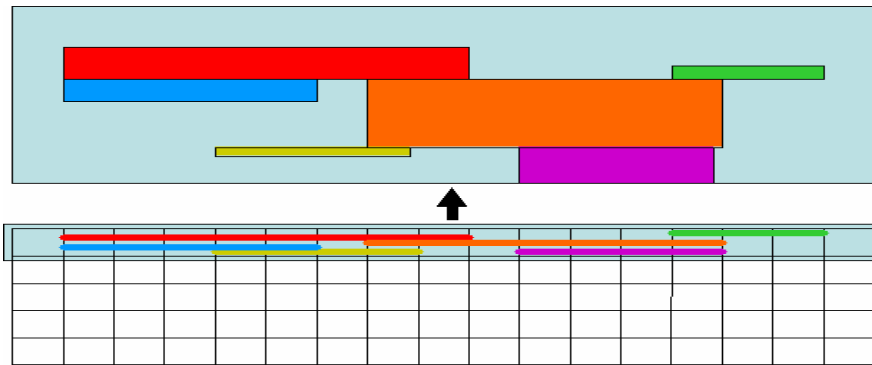


Figure 1. A routing region is partitioned into an 17x5 GCell array; the assignment of the topmost panel with six IRoutes is zoomed in on the top.

This work assumes that the separation rule between any two adjacent IRoutes is sp . All IRoutes are over-sized by $sp/2$ to guarantee legal separation between adjacent IRoutes. Figure 2(a) displays an example of over-sized IRoutes. Hence, the over-sized IRoute can neighbor on other IRoutes, and only two abutting IRoutes are considered to induce coupling capacitance. Two IRoutes with a non-zero separation between them do not induce crosstalk. After over-sizing original IRoutes, the GTA problem can be regarded as a special floorplanning problem. Every over-sized IRoute is equivalent to a block with a fixed x-coordinate constraint. The aim of this procedure

is to find a complete floorplanning of the minimum block abutting length within a fixed-height routing region. Figure 2(b) displays the relation between two IRoutes.

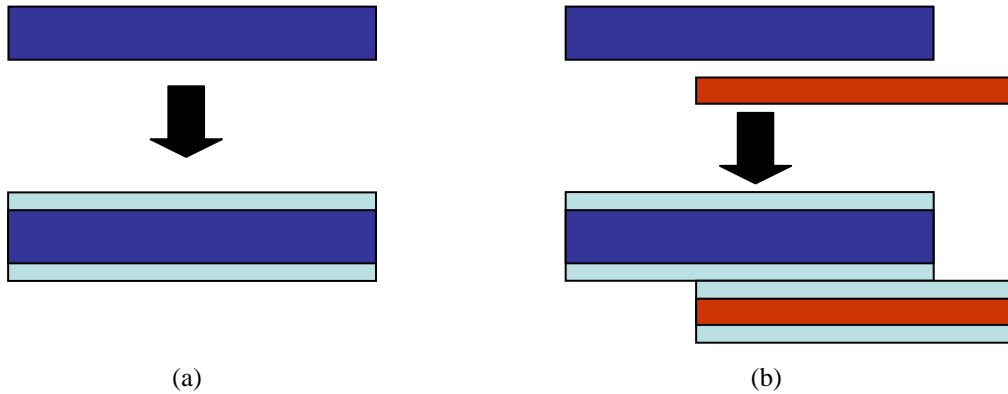


Figure 2. (a) The separation rule between any two adjacent IRoutes is sp . All IRoutes are over-sized by $sp/2$; (b) the over-sized IRoute can neighbor on other IRoutes and don't get a violation.

Observations in previous investigations [14][15] reveal that two highly coupling-capacitance related factors are the space and overlapping length between two wires. The impact of wire width has been observed not to be important for variable-width routing [15], indicating that the wire width has only a small influence on coupling effects, with from doubling or tripling the wire width producing a coupling effect variation of about 0.4% to 7%. Consequently, the coupling effect estimation ignores the wire width, and this study adopts the simple Cc model of Gao and Liu [16]. Cc is observed to drop quickly as two adjacent nets drift apart. The crosstalk model is thus simplified by assuming that only two adjacent wires induce Cc. Additionally, two net segments on different layers and in perpendicular directions are assumed not to be sensitive. Moreover, both adjacent tracks are assumed to be of equal distance, so the Cc between two adjacent nets can be estimated simply from their overlapping length. The following discussion assumes that the space between any two adjacent IRoutes is fixed.

2.2 Gridless Track Assignment

The GTA in the integration has two phases, initial GTA and crosstalk reduction. In initial GTA, a left-edge like algorithm is invoked to obtain an initial GTA result. By this way, we can get an assignment quickly and have a good utilization of the routing area. Since gridless track assignment probably produces uneven partial assignment, it is hard to regard the region as row by row. For the crosstalk minimization objective, initial assignment uses the minimum weighted Hamiltonian path algorithm on the maximum clique. Considering the utilization of the routing area, a left-edge like algorithm is used to avoid waste of routing resource. Figure 3 displays an example of initial GTA.

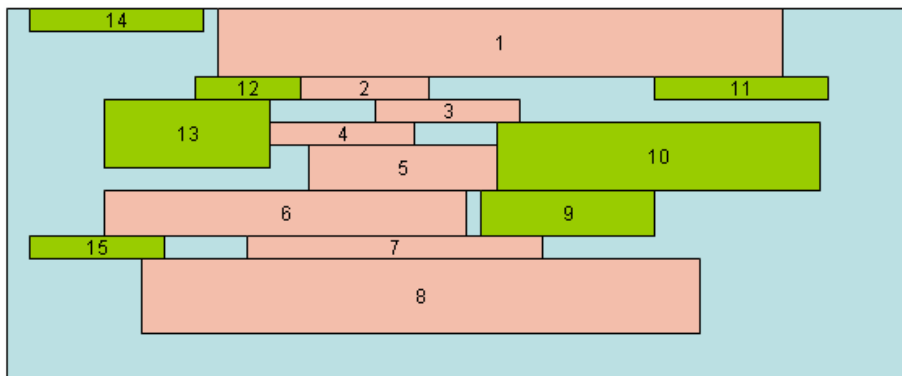


Figure 3. The crosstalk minimization of the maximum clique (1-8) is first completed and the last IRoutes (9-15) are assigned by a left-edge like algorithm.

Now, the problem of re-assigning IRoutes for crosstalk reduction is then transformed into a restricted non-slicing floorplanning problem. The extended O-tree of the over-sized IRoute is established and each node of the extended O-tree stands for an IRoute. An edge is produced between two IRoutes if there is a non-zero vertical projection between them. For all IRoutes in the panel, GTA allows them to move far away if crosstalk can be avoided. Figure 4(a) displays a movement of an IRoute in order to reduce crosstalk.

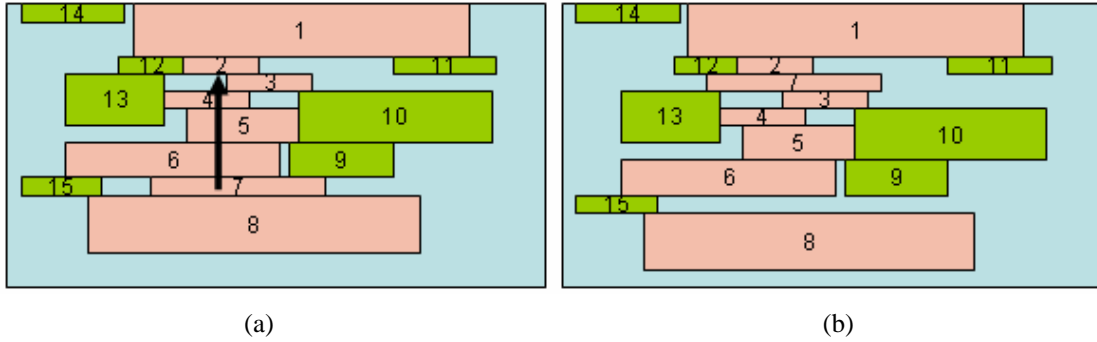


Figure 4. After IRoute 7 is inserted on the top of IRoute 3 and IRoute 13, the total overlap in the panel can be reduced.

After refining the position for each IRoute, every panel is treated as a collection of sub-panels, each of which can also be considered as a sub-panel with different states on its top and bottom borders for a horizontal IRoute. Re-ordering the sub-panel can further decrease the total overlapping length. For example, the assignment in figure 4(b) can be split into six sub-panels, as shown in Fig. 5(a). Then an overlap graph is constructed, where each node in the graph represents a sub-panel and there are two directed edges between any two nodes. The cost of an edge is the total overlap between any two sub-panels. Therefore, the crosstalk minimization problem can be formulated as finding a minimum weighted Hamiltonian path (MWHP) on the sub-panel overlap graph. Figure 5(b) displays the assignment after re-ordering sub-panels.

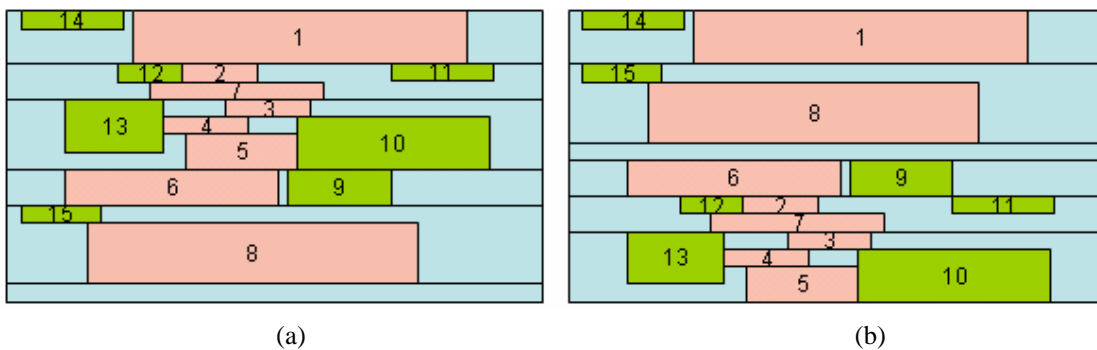


Figure 5. (a) The assignment is partitioned into six sub-panels. (b) A new sub-panel order for crosstalk minimization.

Finally, some IRoute will be pulled upward or downward if there is space on their top or bottom. Figure 6 displays the final assignment of figure 3.

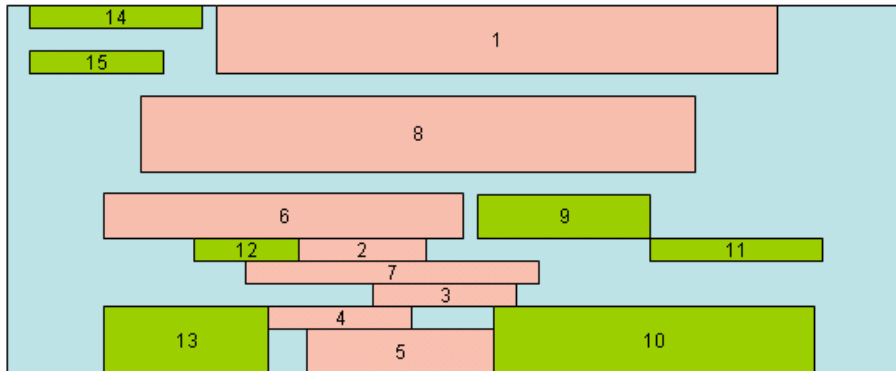


Figure 6. The final result after GTA algorithm.

2.3 NEMO Overview

NEMO [11], an implicit connection graph router, provides a good performance for gridless detailed routing. During path propagation, NEMO groups series of adjacent tile as PMTs to advance path searching. A slit and interval tree stores all existing blockage, providing an efficient query scheme.

The legality validation and PMT extraction are conducted whenever the router intends to explore a neighboring unvisited region. Since the number of such explorations before reaching the target is very large, the time spent in identifying PMT is a significant issue. Other approaches for encouraging detailed routing in NEMO are gridline reduction and pseudo-blockage insertion. Gridline reduction produces a simplified connection plane by disregarding the gridlines induced from those blockages entirely out of the global path found by the global router. Pseudo-blockage insertion places fictitious blockages around the global path to cease PMT extraction when the interval tree query reaches the global path boundary.

Chapter 3

Integration

3.1 System Flow

When global routing is finished, GTA extracts long and straight IRoutes in terms of global paths. Initial GTA will quickly place as many IRoutes as Possible, while minimizing the length of overlap in the maximum clique. At O-tree refinement stage, IRoutes are evaluated to obtain the best insertion point to reduce the overlapping length. Besides, further reduction can be attained by considering the effects between sub-panels. After GTA, the routing tree is constructed first in order to achieve good routing quality. Pattern Routing and implicit connection-graph-based detailed routing will undertake to complete last connections which are short.

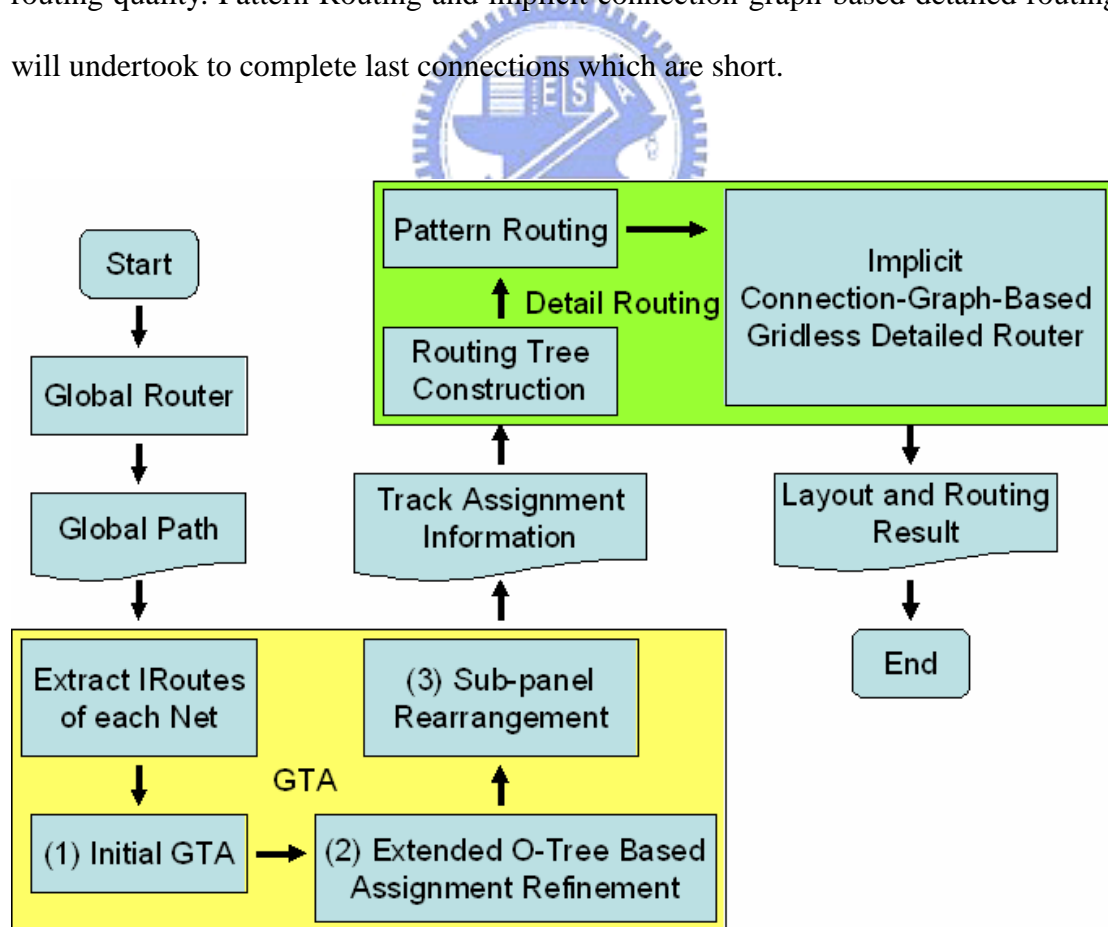


Figure 7. The system flow of three-stage gridless routing system.

3.2 Extracting IRoutes and Layer Assignment

In our design, every net comprises a group of sub-nets, i.e., pin-to-pin nets. For instance, Figure 8(a) displays a net consists of 4 sub-nets and its global paths. If we extract IRoutes only considering the global paths, the load for post detailed routing is heavy and the connection is not efficient. In figure 8(b), the number of post connections for detailed routing is 7. This is obviously ineffective.

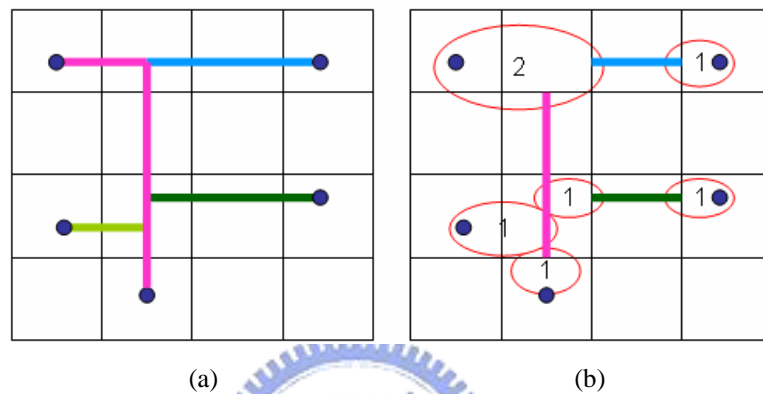


Figure 8. (a) A net consists of four sub-nets and its global paths. (b) If only considering global paths, the post connections = $2 + 1 + 1 + 1 + 1 + 1 = 7$.

In order to effectively partition a net routing, IRoutes are extracted according to the set of all global paths is essential, as shown in Fig. 9(a). Figure 9(b) displays that only 6 shorter connections are needed when we extract long IRoutes for GTA. In addition, because long IRoutes are assigned at GTA stage, the estimation of crosstalk for routing system is more precise.

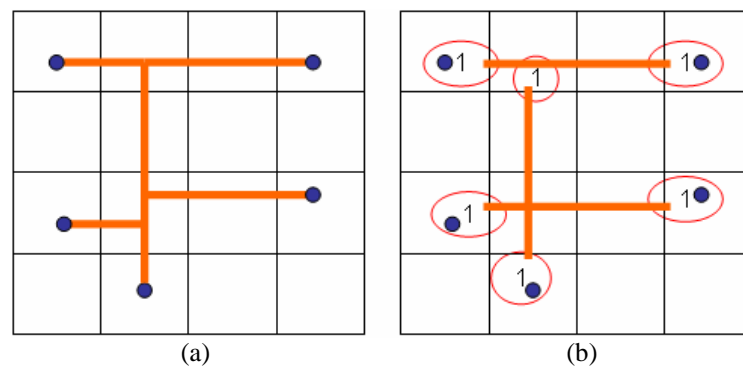


Figure 9. (a) The set of all global paths is essential. (b) The post connections = $1 + 1 + 1 + 1 + 1 + 1 = 6$.

After extracting IRoutes from the set of global paths, both ends of each IRoute are extended by 1 unit; otherwise, two IRoutes are probably assigned to the same track in adjacent panels (short errors), as shown in Fig. 10(a). To avoid short errors, IRoute extension works out very well. Figure 10(b) displays that the short error will not happen and the assignment is safe after IRoute extension.

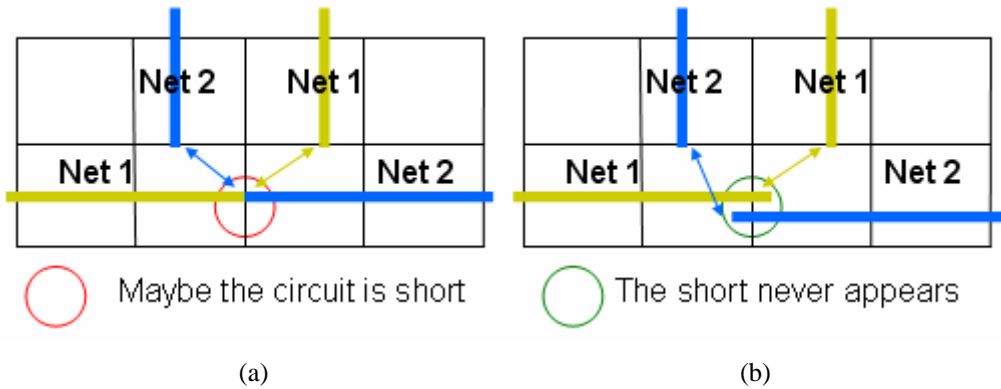


Figure 10. (a) If the adding is omitted, the circuit is maybe short. (b) The short situation will not happen and the assignment is safe when adding 1 unit.

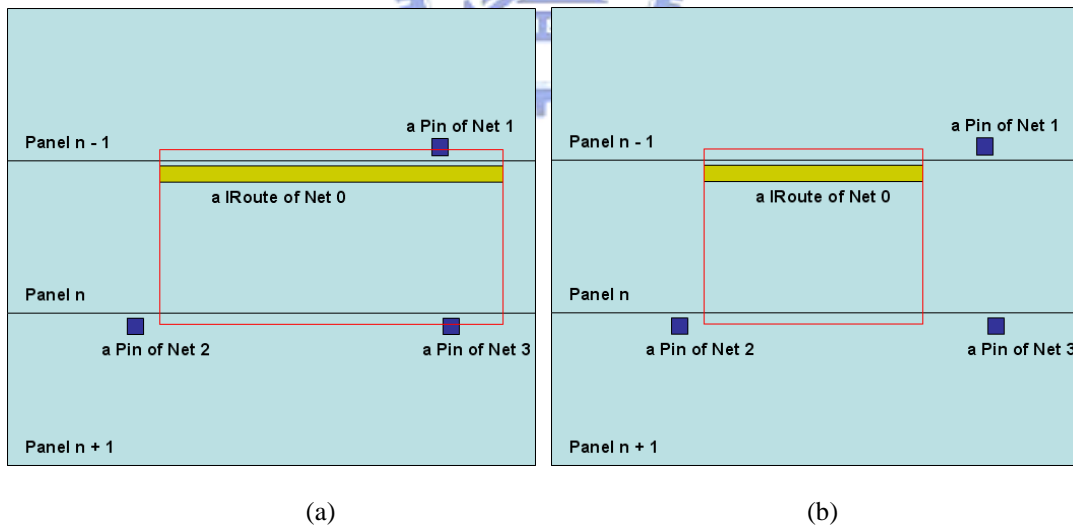


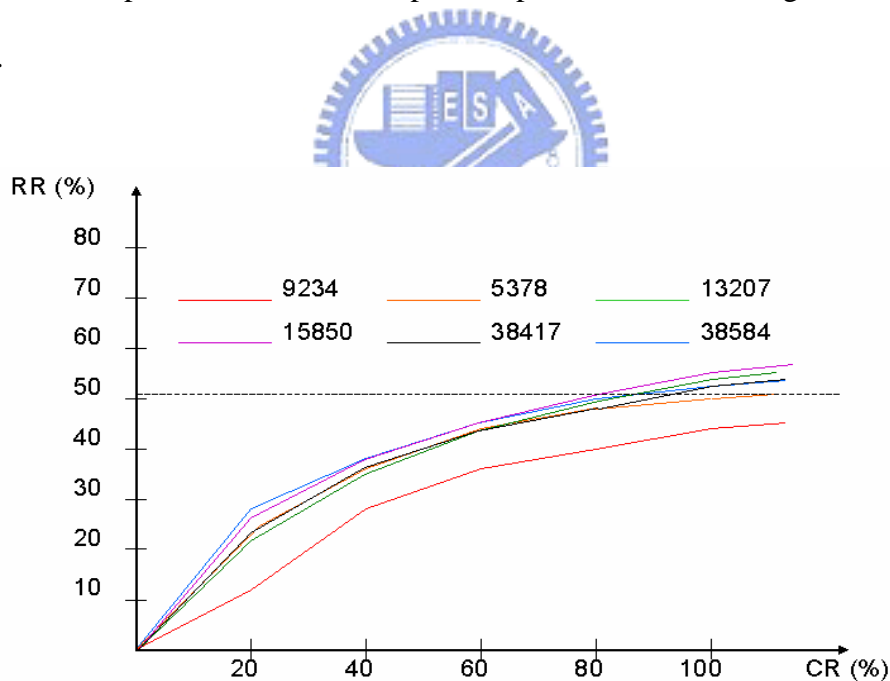
Figure 11. (a) the IRoute may make a violation in this panel. (b) the IRoute will be assigned to this panel and never get a violation.

Because the routing area is limited, distribute a large number of IRoutes to different layers is very important. In addition to wires, there are a lot of pins located on metal layers. Hence some IRoutes may make a violation if we do not plan IRoutes carefully, For example, when the IRoute of net 0 in Fig. 11(a) is close to the bottom of panel(n-1), the spacing between the IRoute of net 0 and the pin of net 1 may induce a design rule violation. In order to prevent violations, when we assign an IRoute to some layer, a box query is necessary. The width of the query box is set as the length of the IRoute and the height of the box is obtained by shifting box's top and bottom borders upwards and downwards by $1/2$ separation unit, as shown by the red frames in Fig 11(a). If the query box contains any pin in adjacent panel, the assignment is not feasible, as shown in Fig. 11(a), while Fig. 11(b) displays the condition for a feasible assignment.



3.3 Extended O-Tree Based Assignment Refinement (EOBAR)

A straightforward method to reduce the overlapping length is to iteratively eliminate each IRoute, and insert it in another position with length reduction profit. In order to decrease the GTA crosstalk, a simple deterministic placement algorithm is employed to decrease the GTA crosstalk. For each IRoute, all internal insertion points along the path passing the processed IRoute are evaluated to obtain the best insertion point to reduce the overlapping length. Fig. 12 displays that EOBAR at the beginning gets higher overlapping reduction ratio and have a better quality in crosstalk reduction. As refinements have been executed for a while, the reduction ratio will stop growing up and be closed to 50% ~ 60%. Finally the assignment at this stage will be stable. In addition, the experiment shows that plural operations at this stage will get better results.



RR : the Reduction Ratio of Overlapping Length CR: Completed Ratio

Figure 12. As refinements have been executed for a while, the reduction ratio will stop growing up and be closed to 50% ~ 60%. Finally the assignment at this stage will be stable.

3.4 Routing Tree Construction

Each net routing following GTA leaves several IRoutes and unconnected pins. Planning how to connect pins and IRoutes together is stipulated to achieve good routing quality. Figure 13 displays an example of 4 nets after GTA and its final routing result.

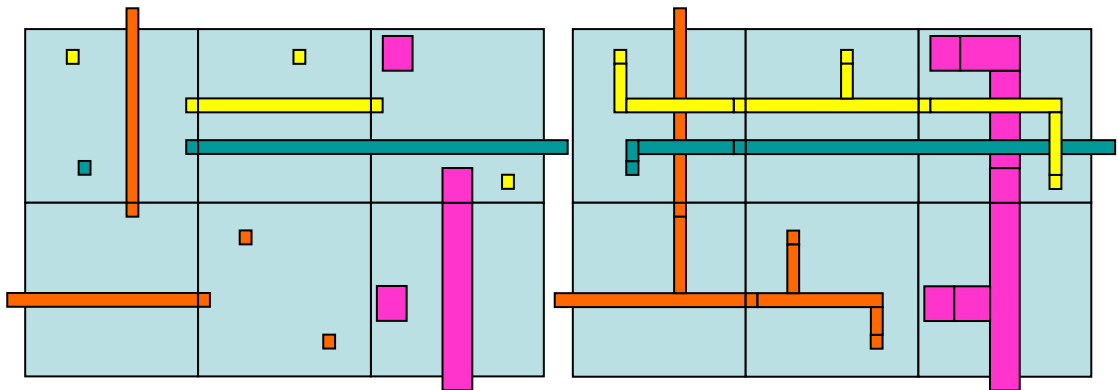


Figure 13. An example of 4 nets after GTA and its final routing result.

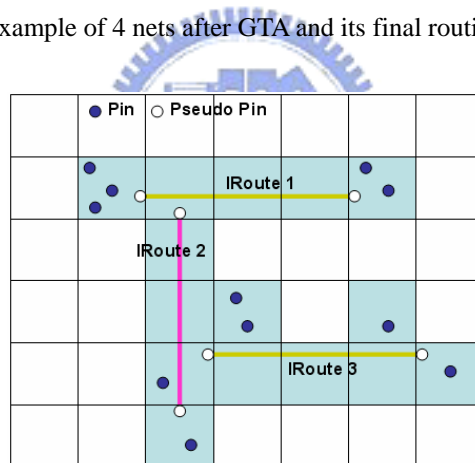


Figure 14. State of a net routing with 11 Pins following GTA, which produces three IRoutes and six pseudo-pins. The blue region is the global path for this net routing.

Figure 14 displays a net of 11 pins and its global path, shown by shadow region. Three IRoutes are derived based on the global path. Two pseudo pins are defined for each IRoute in the two ending GCells of a straight global path, as shown by the hollow circles in Fig. 14. The remaining work for detailed net routing is to connect all pins, pseudo pins and IRoutes together.

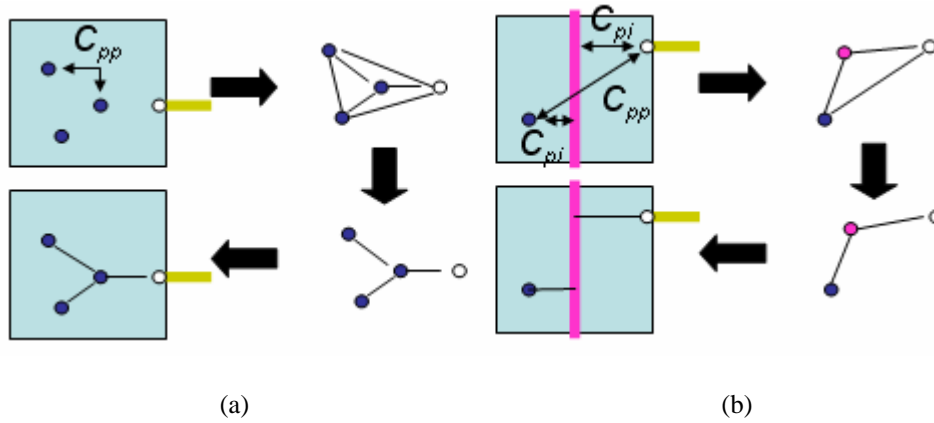


Figure 15. Two edge costs for seeking minimum spanning tree: (a) using the Manhattan distance between any two pins/pseudo pins: (b) using the cost of pin/pseudo pin to IRoute based on their projection distance.

Figure 15 displays two cases of routing tree construction. As shown in Fig. 15(a), all pins and pseudo pins are regarded as nodes, and any two nodes have an edge with a cost defined by their Manhattan distance. This cost, called the Manhattan distance cost, of any two pins or pseudo pins, is denoted as C_{pp} . The other edge cost defines the edge cost of pin or pseudo pin to IRoute. Figure 15(b) displays a case of a GCell containing a pin, a pseudo pin and an IRoute. An IRoute is also denoted as a node in the complete graph, but its edge cost, represented as C_{pi} , between itself and any pin or pseudo pin is shortest distance between them. For instance, the edge cost between the IRoute and the pin or pseudo pin in Fig. 15(b) is given by the horizontal distance from the pin or pseudo-pin to the projection point on the IRoute. Prim's algorithm is adopted to seek the minimum spanning tree. This routing tree determines the point-to-point routings invoked by detailed routing. The routing tree found at this point probably does not include all pins of a net, since some segments are too short to be processed by GTA. For instance, there are two short segments not processed by GTA, as shown with two heavy shadow GCells in Fig. 16(a), and its routing tree is not complete. To complete the entire routing tree construction, the routing tree identified so far, and all pins in a short-segment Gcell, are treated as nodes, and the

edge cost of a pin to the existing routing tree is the shortest distance between them. For each short-segment GCell, a complete graph is constructed and minimum spanning tree is discovered. The final routing tree is then completed, as shown in Fig. 16(b).

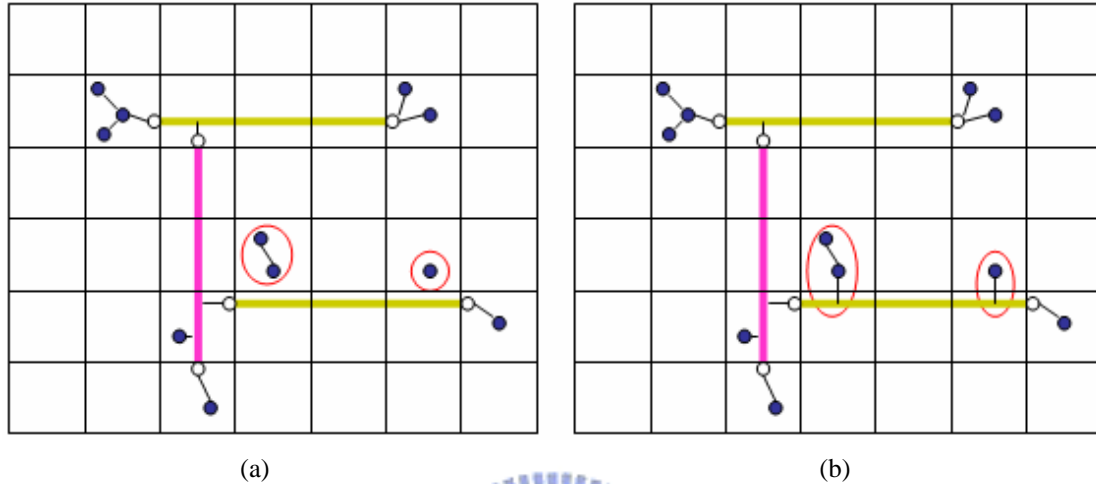
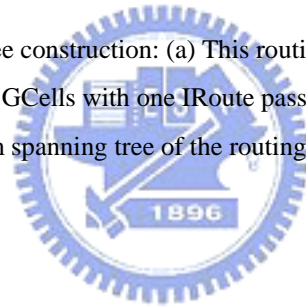


Figure 16. Two steps of routing tree construction: (a) This routing tree is derived after identifying the minimum spanning trees in all GCells with one IRoute passing by; (b) final routing tree is derived after seeking the minimum spanning tree of the routing tree in (a) and all GCells without IRoute passing by.



3.5 Pattern Routing

Pattern routing has already been adopted in global routing to increase the routing speed [12][13]. This work conducts pattern routing before detailed routing to simplify some easy routings. Detailed routing can solve all point-to-point routings after building the routing tree. However, the point-to-point routings are totally different from those in a two stage routing flow, i.e., global routing plus detailed routing. Long-distance routing has been completed by GTA, and the detailed router is responsible for short-distance routing and incomplete IRoutes. Observations on some real circuits, many short-distance routings are based on pin-to-IRoute routing. Most such routings can be completed using a direct or L shape connection. Therefore, simple direct connection and L-shape routing is undertaken at this stage.

Chapter 4

Experimental Results

All routing tests were conducted on a 1.2GHz Sun Blade-2000 workstation with 2GB memory with six MCNC benchmark circuits as presented in Table 1.

| Circuit | Size (μm) | # 2-pin nets | # Pin | # GC | # panel |
|---------|------------------|--------------|-------|----------|---------|
| S5378 | 4350 x 2390 | 3124 | 4818 | 55 x 30 | 85 |
| S9234 | 4040 x 2250 | 2774 | 4260 | 51 x 28 | 79 |
| S13207 | 6600 x 3650 | 6995 | 10776 | 83 x 46 | 129 |
| S15850 | 7050 x 3890 | 8321 | 12793 | 89 x 49 | 138 |
| S38417 | 11440 x 6190 | 21035 | 32344 | 144 x 78 | 222 |
| S38584 | 12950 x 6720 | 28177 | 42931 | 163 x 85 | 248 |

Table 1. Benchmark Circuits Statistics for Full-chip Routing.

| Circuit | Three-stage gridless routing system | | | |
|---------|--------------------------------------|-------|---|-----|
| | GTA + Detail Routing CPU Time (s) | | GTA + Detail Routing Memory Usage (MB) | |
| | FR | VR | FR | VR |
| S5378 | 1 | 1.81 | 45 | 43 |
| S9234 | 0.81 | 1.35 | 43 | 40 |
| S13207 | 2.22 | 5.16 | 59 | 54 |
| S15850 | 2.83 | 7.23 | 65 | 59 |
| S38417 | 6.72 | 11.84 | 107 | 89 |
| S38584 | 11.23 | 23.26 | 135 | 110 |

Table 2. CPU Time and Memory Usage.

O.L. : overlap length; R1: reduction rate = (L1-L2)/L1

R2: reduction rate = (L2-L3)/L1; FR: fixed rule; VR: variable rule.

| Circuit | Initial assignment | | O-tree based re-assignment | | | | HIR re-arrangement | | | |
|---------|-----------------------------------|-------------|------------------------------------|-------------|--------|-----------|------------------------------------|-------------|--------|----------|
| | O.L.(x10 ⁴ um) (L1) | | O.L. (x10 ⁴ um) (L2) | | R1 (%) | | O.L. (x10 ⁴ um) (L3) | | R2 (%) | |
| | FR | VR | FR | VR | FR | VR | FR | VR | FR | VR |
| S5378 | 1.90 | 1.32 | .543 | .467 | 71 | 65 | .385 | .390 | 9 | 5 |
| S9234 | 1.23 | .894 | .259 | .151 | 79 | 83 | .166 | .123 | 8 | 3 |
| S13207 | 4.90 | 3.85 | 1.43 | 1.47 | 71 | 62 | 1.17 | 1.23 | 5 | 6 |
| S15850 | 6.27 | 4.93 | 2.33 | 2.27 | 63 | 54 | 1.92 | 2.13 | 6 | 3 |
| S38417 | 12.7 | 9.50 | 3.20 | 2.59 | 75 | 73 | 2.38 | 2.21 | 6 | 4 |
| S38584 | 18.0 | 13.4 | 7.02 | 5.49 | 61 | 59 | 5.13 | 4.46 | 11 | 8 |
| Ave. | | | | | 70 | 66 | | | 8 | 5 |

Table 3. Statistics of Crosstalk Reduction for Fixed- and Variable-rule Routing.

The entire routing system comprises a global router, crosstalk-driven GTA, and enhanced NEMO. Table 3 shows the crosstalk reduction, for fixed- and variable-rule routings in the GTA stage. Variable-rule routing was performed using the same circuits with modified rule set as follows. The width of the longest 10% of nets was doubled; that of the next 10% of nets was multiplied by 1.5, while the others remain unaltered. Since many pins on the first metal layer are aligned and set apart in minimum-rule space, and design rule violation occurs if the rule of the first metal layer and its pins are widened, the rules of the first metal layer remained unchanged. GTA comprises three stages, initial assignment, O-tree based iterative re-assignment and HIR reordering. In Table 3, columns 2, 4 and 8 compare the overlap length of three-phase assignments for fixed-rule routing, and columns 3, 5 and 9 list the overlap lengths of three-phase assignments for variable-rule routing. For the routings of two set design rules, the phase of O-tree based iterative re-assignment contributes from 66–70% overlap length reduction, while a further overlap length reduction of 5–8% is

achieved in the subsequent phase, HIR reordering. Although initial assignment does not totally focus on crosstalk reduction, and may cause bad-quality assignment in terms of crosstalk effect. Three-phase algorithm still produces a fast and efficient crosstalk-driven GTA.

Ini.: initialization for GTA; **GTA C.R.:** three-phase GTA crosstalk reduction;

Preprocess: preprocessing for detailed routing

| | Three-stage gridless routing system | | | | | | | | | | | | | | G.R. + NEMO | |
|---------------|-------------------------------------|-------------|-----------------|-------------|-------------------|-------------|------------------------|-------------|------------------------|-------------|------------------|--------------|-------|--------------|----------------|---------------|
| | Run time (sec) | | | | | | | | | | | | | | Run time (sec) | |
| | Ini. (1) | | GTA C.R. (2) | | Preprocess (3) | | Pattern routing (4) | | (1) + (2) + (3)+(4) | | Enhanced NEMO | | Total | | FR | VR |
| FR | VR | FR | VR | FR | VR | FR | VR | FR | VR | FR | VR | FR | VR | | | |
| s5378 | 0.06 | 0.05 | 0.1 | 0.14 | 0.12 | 0.11 | 0.07 | 0.06 | 0.35 | 0.36 | 0.65 | 1.45 | 1 | 1.81 | 2.4 | 3.74 |
| s9234 | 0.04 | 0.03 | 0.07 | 0.06 | 0.1 | 0.08 | 0.05 | 0.05 | 0.26 | 0.22 | 0.55 | 1.13 | 0.81 | 1.35 | 1.7 | 2.69 |
| s13207 | 0.11 | 0.10 | 0.31 | 0.27 | 0.28 | 0.23 | 0.2 | 0.18 | 0.9 | 0.78 | 1.32 | 4.38 | 2.22 | 5.16 | 6.6 | 11.28 |
| s15850 | 0.15 | 0.12 | 0.36 | 0.47 | 0.36 | 0.31 | 0.26 | 0.23 | 1.13 | 1.13 | 1.7 | 6.1 | 2.83 | 7.23 | 8.8 | 16.33 |
| s38417 | 0.41 | 0.29 | 0.68 | 0.69 | 1.02 | 1.07 | 0.78 | 0.66 | 2.85 | 2.71 | 3.87 | 9.13 | 6.72 | 11.84 | 37.2 | 56.36 |
| s38584 | 0.60 | 0.42 | 1.13 | 1.12 | 2.63 | 1.94 | 1.34 | 1.14 | 5.7 | 4.62 | 5.53 | 18.64 | 11.23 | 23.26 | 73.7 | 143.39 |
| Comp. | | | | | | | | | | | | | 1 | 1 | 3.78 | 3.24 |

Table 4. Comparison of Routing Time between Our Work and [11].

Table 4 compares the routing statistics of this work and that of Li. et al. [11] for fixed- and variable-rule routings. The runtime for this work is split into five stages, namely initialization for GTA, GTA, preprocessing before detailed routing, pattern routing and detailed routing. The runtime speedup for six test cases is in the range 2.1–6.6 times and 2–6.2 times for fixed- and variable-rule routings, respectively. However, the wire length is getting longer by approximately 5% and 6% for crosstalk minimization for for fixed- and variable-rule sets. All cases were 100% completed under two rule sets. Table 5 lists the statistics of completed IRoutes and

point-to-point routings by GTA and pattern routing. GTA completes the placement of all IRoutes of four out of six test cases, except for one incomplete IRoute in case s15850 and nine in case s38584 for fixed-rule routing. For variable-rule routing, 51 IRoutes in five cases were incomplete by GTA, resulting in more runtime by detailed routing in these cases. For detailed routing, 42–58% point-to-point routings were completed with pattern routing to lower the burden of detailed routing. High completion rate before detailed routing was realized through IRoute assignment, followed by connecting vicinal pins.

| | # Iroutes / # incomplete Iroute | | # point-to-point routings / # completed by pattern routing (completion rate) | |
|---------------|---------------------------------|------------------|---|------------------------------|
| | FR | VR | FR | VR |
| s5378 | 1626 / 0 | 766 / 3 | 4737 / 2088 (59.1%) | 3873 / 1596 (41.2%) |
| s9234 | 1267 / 0 | 542 / 0 | 4034 / 2349 (58.2%) | 3309 / 1364 (41.2%) |
| s13207 | 3406 / 0 | 1613 / 13 | 10375 / 6052 (58.3%) | 8568 / 3670 (42.8%) |
| s15850 | 4184 / 1 | 1925 / 18 | 12471 / 7272 (58.3%) | 10186 / 4328 (42.5%) |
| s38417 | 9589 / 0 | 4314 / 1 | 30552 / 1777 (58.2%) | 25289 / 10694 (42.3%) |
| s38584 | 13150 / 9 | 5674 / 16 | 41184 / 23808 (57.8%) | 33795 / 13844 (41.0%) |

Table 5. Statistics of Completed IRoutes and Point-to-Point Routing by GTA and Pattern Routing.

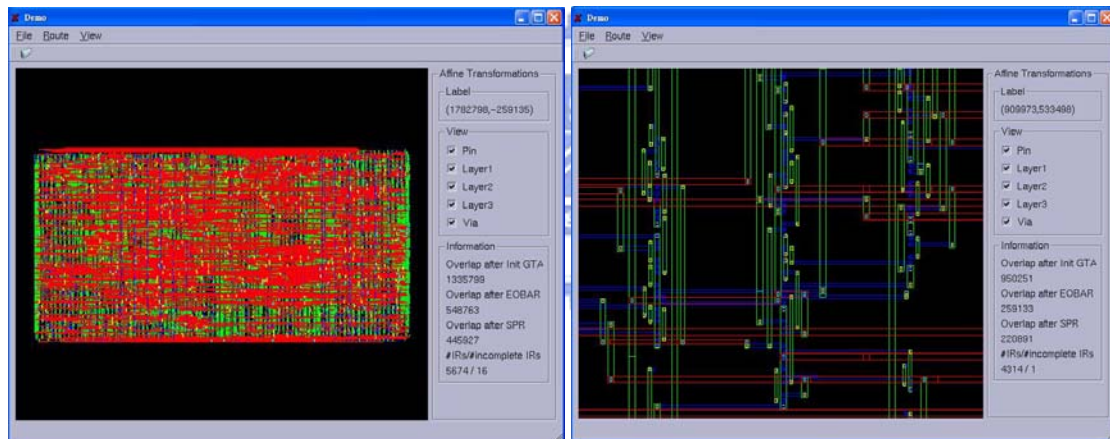
| | Three-stage gridless routing system | | G.R. + NEMO | |
|---------------|-------------------------------------|--------------|------------------------|--------------|
| | W.L.(μm) | | W.L. (μm) | |
| | FR | VR | FR | VR |
| s5378 | 7.8e4 | 8.1e4 | 7.4e4 | 7.6e4 |
| s9234 | 5.8e4 | 5.9e4 | 5.5e4 | 5.6e4 |
| s13207 | 1.8e5 | 1.9e5 | 1.7e5 | 1.8e5 |
| s15850 | 2.3e5 | 2.4e5 | 2.2e5 | 2.2e5 |
| s38417 | 5.0e5 | 5.2e5 | 4.8e5 | 4.9e5 |
| s38584 | 6.9e5 | 7.2e5 | 6.7e5 | 6.8e5 |
| Comp. | 1 | 1 | 0.95 | 0.94 |

Table 6. Comparison of Wire Length between This and [11].

Table 7 compares the crosstalk reduction of this work and commercial tools. Column 2 lists the original crosstalk with commercial tools. Column 3 and 4 shows the crosstalk reduction with commercial tools and this work. Experimental Results indicate that this work and commercial tools get approximate results as considering crosstalk.

| | Nanoroute without SI | Nanoroute with SI | Our Work |
|--------------|----------------------|---------------------|--------------|
| 9234 | 3.424740 pf / 27s | 2.605320 pf / 31s | 2.706407 pf |
| 5378 | 5.648963 pf / 33s | 4.332440 pf / 36s | 4.718894 pf |
| 13207 | 13.774663 pf / 74s | 9.580374 pf / 87s | 10.419775 pf |
| 15850 | 18.272862 pf / 84s | 13.099187 pf / 101s | 14.754539 pf |
| 38417 | 32.824336 pf / 190s | 23.232287 pf / 226s | 25.792010 pf |
| 38584 | 49.812333 pf / 303s | 35.442151 pf / 354s | 38.285301 pf |

Table 7. Comparison of Crosstalk Reduction between Our Work and Commercial Tools.



(a)

(b)

Figure 17. Routing Results (a) Full View of s15850 (b) Partial View of s38417.

Chapter 5

Conclusions and Future Work

This work presents a three-stage gridless routing system, comprising a congestion-driven router, a crosstalk-driven GTA and an enhanced implicit connection-graph-based detailed router. Crosstalk reduction in GTA is transformed to a non-slicing floorplanning problem, and an O-tree-based deterministic floorplanning algorithm is employed. Further reduction is obtained by splitting the routing region into HIRs, then re-ordering the HIRs with a minimization weighted Hamiltonian path. After GTA and routing tree construction, many original point-to-point routings are set to connect to IRoutes, and can be simply resolved using pattern routing. Finally, the detailed router simplifies its gridline extraction and PMT extraction by adopting a bin-based data structure and tagging blocked tiles. Experimental results reveal that the overlapping lengths of adjacent wires are efficiently decreased by more than 70%, and over 3.24 times the runtime speedup is achieved for fixed- and variable-rule routings. Finally we compare this work and commercial tools, experimental results indicate that this work and commercial tools get approximate results as considering crosstalk. In order to reduce more crosstalk, plural operations to adjust IRoutes are needed and we expect to have better results in our future work.

Reference

- [1] S. M. Sait and H. Youssef, “VLSI physical design automation,” World Scientific Publishing, 1999.
- [2] S. Batterywala, N. Shenoy, W. Nicholls, and H. Zhou, “Track assignment : a desirable intermediate step between global routing and detailed routing,” IEEE International Conference on Computer Aided Design, pp. 59–66, Nov. 2002.
- [3] A. Margarino, A. Romano, A. De Gloria, F. Curatelli, and P. Antognetti, “A tile-expansion router,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. CAD-6, pp. 507–517, July 1987.
- [4] Jeremy Dion and Louis M. Monier, “Contour: A Tile-based Gridless Router,” Western Research Laboratory Research Report 95/3, Palo Alto, California.
- [5] J. Cong, J. Fang, and K. Khoo, “DUNE: A multilayer gridless routing system,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 20, No. 5, pp. 633–646, May, 2001.
- [6] Y. S. Kuo, T. C. Chern, Wei-kuan Shih, “Fast algorithm for optimal layer assignment,” Proceedings of the 25th ACM/IEEE conference on Design automation, pp. 554–559, 1988.
- [7] C.-J. R. Shi, “Solving constrained via minimization by compact linear programming,” Proceedings of the Asia and South Pacific Design Automation Conference, pp. 635–640, 1997.

[8] C.-C. Chang and J. Cong, "An efficient approach to multilayer layer assignment with an approach to via minimization," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 18, no. 5, pp. 608–620, May 1999.

[9] J. D. Cho, S. Raje, M. Sarrafzadeh, M. Sriram, and S. M. Kang, "Crosstalk-minimum layer assignment," Custom Integrated Circuits Conference, pp. 29.7.1–29.7.4, 1993.

[10] T.-Y. Ho, Y.-W. Chang, S.-J. Chen, and D. T. Lee, "Crosstalk-and performance-driven multilevel full-chip routing," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 24, No. 6, pp. 869–878, June 2005.

[11] Yih-Lang Li, Xin-Yu Chen, and Zhi-Da Lin, "NEMO: A New Implicit Connection Graph-based Gridless Router with Multi-layer Planes and Pseudo-tile Propagation", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 26, No. 4, pp. 705–718, Apr. 2007.

[12] R. Kastner, E. Bozozzadeh, and M. Sarrafzadeh, "Predictable routing," Proceeding of IEEE/ACM Intl. Conf. Computer-Aided Design, pp. 110-113, 2000.

[13] Min Pan and Chris Chu, FastRoute, "A Step to Integrate Global Routing into Placement," Proceeding of IEEE/ACM International Conference on Computer-Aided Design, pages 464-471, 2006

[14] S.-W. Tu, W.-Z. Shen, Y.-W. Chang, T.-C. Chen, and J.-Y. Jou, "Inductance modeling for on-chip interconnects," International Journal of Analog Integrated Circuits and Signal Processing, Vol. 35, No. 1, pp. 65–78, April 2003.

[15] L. He and M. Xu, “Modeling and Layout Optimization for On-Chip Inductive Coupling,” U. of Wisconsin at Madison, Technical Report ECE-00-1, Dec 1999.

[16] T. Gao and C. L. Liu, “Minimum crosstalk channel routing,” IEEE International Conference on Computer Aided Design, pp. 692–696, 1993.

[17] Pei-Ning Guo, Chung-Kuan Cheng, and Takeshi Yoshimura, “Floorplanning Using a Tree Representation”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 20, No. 2, pp. 281–289, 2001.

[18] Y.C. Chang, Y. W. Chang, G. M. Wu, and S. W. Wu, “B*-trees : A New Representation for Non-slicing Floorplans,” Proceeding of ACM/IEEE Design Automation Conference, pp. 458-463, June 2000.

