國立交通大學

資訊科學與工程研究所

碩士論文

考慮冗餘接點之電路線軌指派法

Redundant-Via Aware Track Assignment

研 究 生：蘇冠丞

指導教授：李毅郎　教授

中華民國 九十七 年十二月

考慮冗餘接點之電路線軌指派法

Redundant-Via Aware Track Assignment

研究生：蘇冠丞　　Student：Guan-Chan Su

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

A Thesis

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 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年十二月

考慮冗餘接點之電路線軌指派法

研究生 ：蘇冠丞　　　　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

傳統的冗餘接點安插是在後佈局階段來實現的，在繞線階段考慮冗餘接點的安插是勢在必行的。三階段的繞線系統對於處理大型的設計成為一種必需的系統。在這篇論文中，我們提出了第一個在電路線軌指派階段考慮冗餘接點影響的演算法(RATA)，並且將此演算法整合到一個三階段的繞線系統中。一個潛在接點的模型首先被提出。線段和接腳的相對位置被分成不同的類型。全域格點段落花費的提出使得在考慮冗餘接點安插的電路線軌指派法上更有彈性和效率。藉由持續的更新全域格點段落花費，反覆的執行最小二部配對法來得到指派的結果。繞線樹將會在電路線軌指派後被建立，並且用簡單且固定的典型連線來連接接腳和線段。在精細繞線之前，產生死亡接點的典型繞線將會被拔除。最後在精細繞線結束之後，套用一個後佈局的冗餘接點安插器來實現最後的冗餘接點安插以及得到冗餘接點安插率。

實驗結果顯示相對於沒有考慮冗餘接點的電路線軌演算法，我們在死亡接點的個數上平均減少了 29%。冗餘接點的安插率則由 99.54% 增為 99.66%。此外此繞線系統和[12]的執行時間相比快了 1.84 倍。

Redundant Via-aware Track Assignment

Student: Guan-Chan Su **Advisor: Dr. Yih-Lang Li**

Institute of Computer Science and Engineering
National Chiao Tung University

Abstract

Traditional redundant via insertion (RVI) is performed at post-layout stage and the effect of RVI is merely considered in detailed routing. Three-stage routing system becomes necessary for processing large-scaled designs. In this paper, we propose the first work to consider the effect of RVI in track assignment, and integrate the proposed redundant-via aware track assignment (RATA) algorithm into a three-stage routing system. A potential via (PV) model is first proposed. Iroutes and pins are then classified into different types according to their relative positions. GCell segment cost is also proposed to offer high flexibility and efficiency to evaluate the cost of assigning an iroute to a track with RVI consideration. RATA algorithm iteratively employs a minimum bipartite matching to identify the assignment with continuous updating GCell segment cost. Routing tree construction is executed following RATA to complete simple connection between iroutes and pins with fixed patterns. Before detailed routing, the pattern routing that yields dead vias is ripped up. After detailed routing, a post-layout RVI tool is applied to realized RVI and have the final RVI rate. Experimental results show that the number of dead vias is decreased by 29% in average as compared to the TA algorithm without RVI consideration. The final RVI rate is promoted from 99.54% to 99.66% by the proposed RATA algorithm. Besides, the complete routing system runs 1.84X faster than the work in [12].

Acknowledgement

I am deeply grateful to my advisor, Dr. Yih-Lang Li for his continuous guidance, support, and ardent discussion throughout this research. His useful 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

List of Figures

List of Tables

Chapter 1
Introduction

As the feature size of VLSI designs keeps shrinking, yield improvement becomes an important issue. In the well-known methods to increase yield, redundant via insertion (RVI) tries to add redundant via next to every existing via in the original layout to improve the reliability of interconnection. Chen *et al.* [1] transformed the post-layout redundant via insertion problem into a minimum weighted bipartite matching problem. Lee *et al.* [2] showed that bipartite graph model proposed in [1] cannot find an optimal solution in some cases, and modeled the problem with the maximum independent set (MIS) problem and solved it by a heuristic algorithm. In the work by Lee *et al.* in [3], several methods are announced for speeding up the MIS-based approach as well as improving quality. In 2008, Lee *et al.* [4] proposed an optimal algorithm for post-layout RVI.

On the other hand, as the interconnection complexity increases, the run time of routing also increases substantially. Conventional routing system contains global and detailed routing. At global routing stage, routing region is partitioned into smaller sub-regions called global cells (GCells). A global path to connect terminals for each net is thus identified under some constraints such as density or wirelength. After global routing, detailed router then determines the real position and layer of every wire segment according to the pre-defined global path. An intermediate stage between global routing and detailed routing, track assignment (TA), is proposed to improve quality and speed [5]. Long wires crossing at least two GCells called “iroutes” are assigned before detailed routing. Based on the concept of iroute in TA, several critical issues are addressed such as metal density [6], crosstalk optimization [7][8], and yield

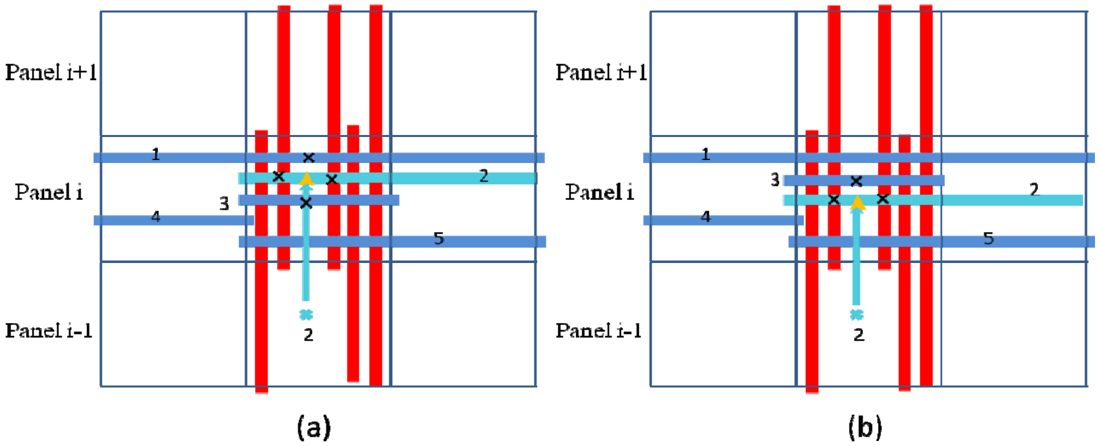


Figure 1. (a) A track assignment yields a dead via; (b) A track assignment yields no dead via.

[9] problems. These effects all happen between wires. If we regard the part of an iroute within a GCell as a *GCell segment*, an iroute is composed of several GCell segments. Previous works treat all segments of an iroute evenly. In this work, GCell *segment cost* is proposed to reinforce the effect of total cost on individual segment.

Currently, RVI is mostly performed at the post-layout stage. However, it is obvious to see that the efficiency of RVI at the post-layout stage is highly bounded by the layout result, and we can hardly change the original layout to make dead vias alive if the layout is heavily congested. Hence, some works started to consider the cost of RVI during routing. Previous routing researches on RVI add the redundant via cost to the path cost during path searching stage. Xu *et al.* [10] first involves a redundant via enhanced maze routing in routing system. In [10], they turn the problem into a multi-constrained shortest path problem by assigning cost to routing resource edges to limit the number of dead vias of each net while routing and, and solve the problem by Lagrangian relaxation technique. Yao *et al.* [11] also change the routing resource density besides via and choose a looser path while back tracing in maze routing. Chen *et al.* [1] add different costs to routing paths during detailed routing, and the costs include the penalty of redundant-via candidates to keep vias alive. These researches all consider redundant via cost at detailed routing stage.

Track assignment provides a good and fast platform to estimate various effects between wires. This special characteristic also helps us to consider the redundant via space reservation. Fig. 1(a) shows a TA result in Panel *i* and the pin of net 2 in Panel *i-1* probably connects to the iroute of net 2 in Panel *i* with a straight wire. This kind of connection may lead to a dead via because all of its four redundant-via resources have been occupied by other nets. But if we change the assignment of iroutes like in Fig. 1(b), the via of net 2 revives. This work attempts to reserves the redundant via space for existing vias such that the number of dead vias is lowered. To our best knowledge, this study is the first work to consider the redundant via space reservation in TA stage and we also integrate the proposed TA into a three-stage routing system, including a congestion-driven global router, a redundant-via aware TA (RATA), and detailed router. The difficulty of considering redundant via in TA is that via positions are uncertain. The schema to predict the location of *potential vias* (PVs) is defined in this work. Base on the PV concept, redundant-via candidates of a PV we called *potential redundant-via* (PRVs) can be obtained. A maximum clique of assignable iroutes will be found, and the minimum weighted bipartite matching for the clique is sought in bipartite graph. Whenever an iroute is assigned, binding groups with it will generate the PVs and PRVs; corresponding operations update the penalty on segments in order to manifest the effects of this assigned iroute. We assigned multiple but not all of the iroutes in the clique to get a better accuracy. After RATA, routing tree construction is executed for placed iroutes and other pins. Pattern routing will be adopted to complete direct line or L-shape.

The remainder of this study is organized as follows. Chapter 2 presents the preliminary. Chapter 3 represents the proposed redundant-via aware track assignment (RATA) algorithm. Chapter 4 shows the routing tree construction and pattern routing. Chapter 5 displays experimental results. Finally, chapter 6 gives the conclusion.

Chapter 2
Preliminary

2.1 Track Assignment

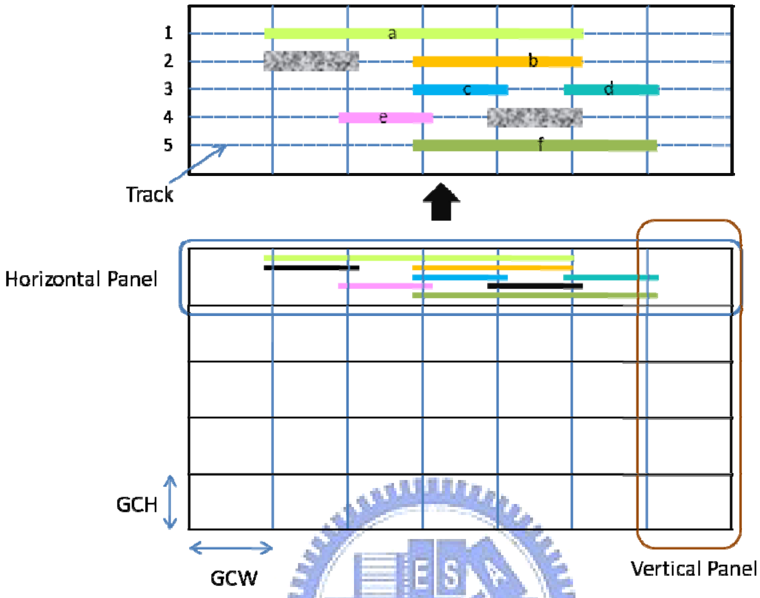


Figure 2. A routing region is decomposed into 5x7 GCells, and the assignment of the topmost horizontal panel with six iroutes and two obstacles is showed in above the panel.

In global routing, the routing region is partitioned into a GCell array. Each GCell has a fixed number of routing tracks. A panel is composed of a serious of GCells in a row or a column. Figure 2 shows a routing region example containing 7x5 GCells at bottom, and each panel consists of 5 tracks. *GCH* is the height of a GCell, and the width of a GCell is *GCW*. The separation rule between two adjacent iroutes is assumed to be *sp* in this study. To make sure of the legality of separation between adjacent iroutes, the amount of horizontal tracks is determined by *GCH/(2*sp)* (*GCW/(2*sp)* for vertical tracks). In Fig. 2, there are six iroutes on the topmost panel, and one assignment of the topmost horizontal panel is drawn at top. First, we extract iroutes from the result of global router. A weighted bipartite matching model is then used to represent the assignablility of every iroutes. In bipartite assignment graph,

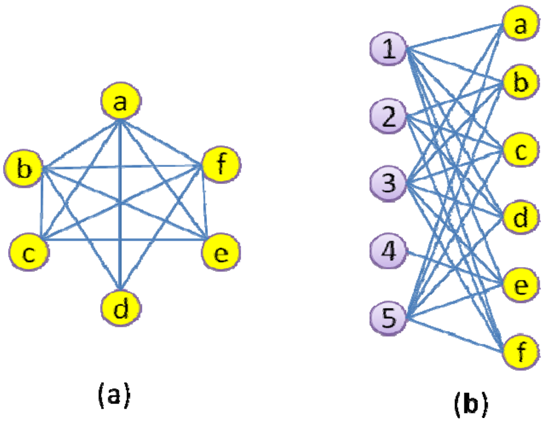


Figure 3. (a) is the OLG of Fig.1 ; (b) is the bipartite assignment graph of Fig.1 .

each iroute has a node in the iroute set, and each track corresponds to a node in the track set. If one iroute can be placed on a track, an edge is inserted to connect their related nodes. An overlap graph (OLG) is also built. Every iroute has a node in the OLG and two nodes are connected by an edge if their iroutes overlap. The maximum clique of the OLG is discovered. Hopcroft–Karp algorithm for solving the maximum matching with minimum weight is then applied to assign the clique. Iroutes are placed by processing every clique in the OLG. Figure 3(a) is the overlap graph of the TA problem in Fig.2 , and Fig. 3(b) is the bipartite assignment graph of Fig. 3(a). TA will first seek a maximum clique in Fig. 3(a) and then finds a minimum weighted bipartite matching for the clique in Fig. 3(b). One of the advantages of traditional TA algorithm is that assigning a clique a time. If a clique contains multiple iroutes, when we assign this clique, we assign multiple iroutes simultaneously. With the long wires being processed quickly, the entire routing flow gains a significant speedup.

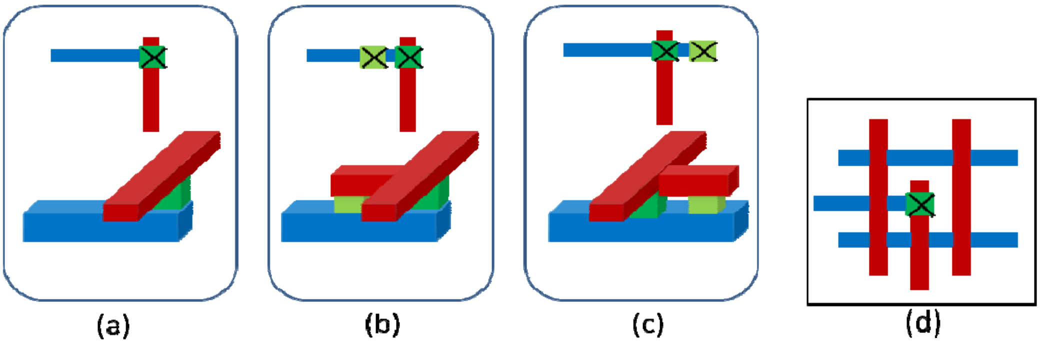


Figure 4.(a) A single via; (b) an on-track redundant via is added; (c) an off-track redundant via is added; (d) a dead via

2.2 Redundant Via (RV)

Redundant via has two types. One is the *on-track* redundant via, the other is *off-track*. On-track redundant via is preferred because it takes less routing resource than off-track. If a via has no routing resource to insert redundant via, then the via becomes a *dead via*. We illustrate each of them in Fig. 4. Figure 4(a) is a single via, Fig. 4(b) adds an on-track redundant via to the single via. Fig. 4(c) adds an off-track redundant via to the single via. In this work, on-track redundant via is preferred to be reserved. Figure 4(d) shows a dead via which has no redundant via can be added.

2.3 Overview of Our Routing System

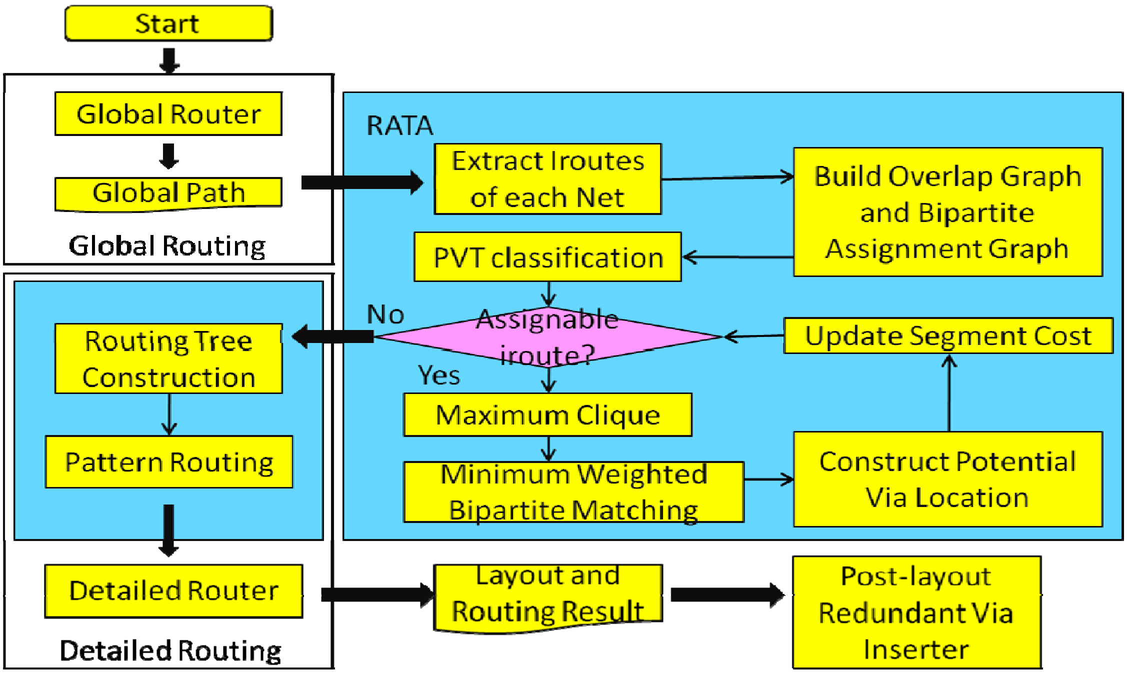

Start
Global Router
Global Path
Global Routing
RATA
Extract Iroutes of each Net
Build Overlap Graph and Bipartite Assignment Graph
PVT classification
No
Assignable iroute?
Yes
Update Segment Cost
Maximum Clique
Minimum Weighted Bipartite Matching
Construct Potential Via Location
Routing Tree Construction
Pattern Routing
Detailed Router
Detailed Routing
Layout and Routing Result
Post-layout Redundant Via Inserter

Figure 5. The system flow of routing system.

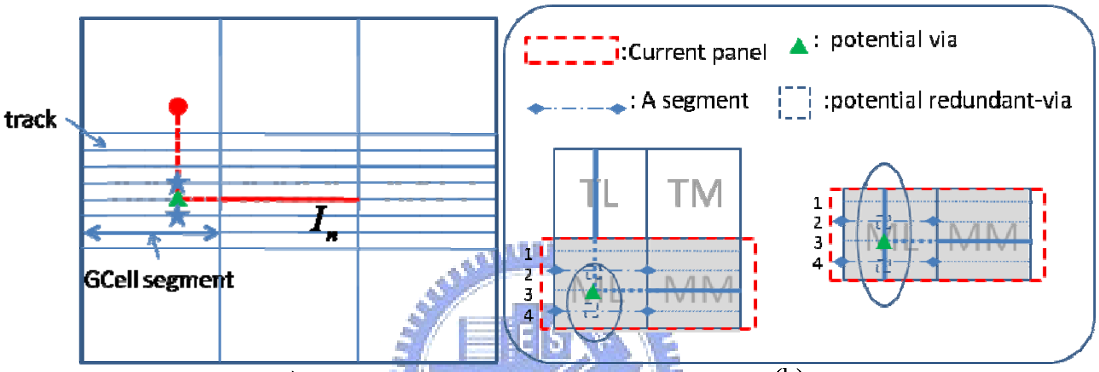
Figure 5 draws the system flow of our three-stage routing system. In the beginning, a congestion-driven global router generates a global path for every net. Then proposed redundant-via aware track assignment algorithm is applied to preserve the space for RVI. In this stage, we first extract iroutes from path generated by global router. For each panel, build OLG and bipartite assignment graph. Then the iroutes and pins grouping procedure is followed for each net. We repeat iroute assignment with continuously evaluating new bipartite edge cost by constructing PV location according to *pvt(ir,MM,n,xx)* and *pvt(pn,MM,n,xx)* and updating the GCell segment cost until there is no more assignable iroutes. After RATA, a routing tree construction with two phases is executed to connect the remaining component of a net. Following the result of routing-tree construction, pattern routing with simple L-shape and straight line is used to lower the burdens of detailed router. If a pattern routing forms a dead via, then this routing is ignored. Detailed router then produces the final routing result, and a post-layout RVI tool is employed to complete RVI.

Chapter 3

Redundant-Via Aware Track Assignment (RATA)

3.1 Potential Via Estimation

3.1.1 Potential Via Model



(a) (b)

Figure 6.(a) The PRV in the type $pvt_{ir}(pn,MM,TL,ll)$; (b) The PV type $pvt_{ir}(ir,MM,TL,ll)$ induces an off-track PRV and an on-track PRV while the PV type $pvt_{ir}(ir,MM,ML,sw)$ induces two on-track

Since the detailed routing is performed following TA, the positions of vias are not determined during TA. To reserve the RVI resource, we have to set up a model to predict the inserted vias by detailed routing with pattern routing or path search algorithm. A via that appears in a routing path with straight-wired or L-shaped pattern routing is called a *potential via* (*PV*). A redundant via of a PV is called a potential redundant via (PRV). A PV is predicted and classified by the relative position between an iroute and its pin or another adjacent-layer iroute of the same net. Based on the proposed classification scheme, the PRV resources for a PV are imposed by different penalty costs to differentiate the on-track and off-track PRVs. The type of a PV in a horizontal panel is notated as *pvt*(*pn_ir*,*pir_GC*,*pn_ir_GC*,*wr_type*), where *pn_ir* is either *pn* (pin) or *ir* (iroute), *pir_GC* is the label of the GCell where a placed iroute is located. *pn_ir_GC* is the label of GCell that contains a pin or another iroute to connect the placed iroute for evaluating the induction of PVs which can be *TL*(top left), *TM*(top middle), *TR*(top right), *ML*(middle left), *MM*, *MR*, *BL*(bottom left), *BM*, and *BR*. *wr_type* is the minimum-length wire pattern used to evaluate the production of PVs and can be *sw* (straight wire), *ll* (lower L-shape), *ul* (upper L-shape), *align*(only when *pn_ir* is a pin), *xx* (no wire pattern). The "*xx*" is used when only the position relationship between iroute and pin is discussed and no specific wire pattern is employed for connection; in other words, "*xx*" includes any minimum-length pattern routing (*sw*, *ll*, *ul* or *align*). For example, for a horizontal placed iroute in GCell *i*, the case of a PV induced by an unplaced vertical iroute in GCell *j* is denoted as the type $pvt(ir,i,j,xx)$. Similarly type $pvt(pn,i,j,xx)$ is referred to as the case of a PV induced by a pin in GCell *j* connecting a horizontal placed iroute in GCell *i*.

For simplicity, the following illustration only discusses the assignment in a horizontal-layer panel. An iroute is regarded as being composed of running GCell segments, that are defined as the portion of an iroute within one GCell, for the ease of classifying PVs. PV classification is based on a 3×3 GCell array with GCell labeled *TL* to GCell labeled *BR*, as shown in Fig. 6 (a). The middle panel in Fig. 6 is the currently processed panel and a placed iroute I_n of net *n* is in GCell 5. A PV may be induced by a connecting vertical iroute or a pin of net *n* in one of these nine GCells. The type of a PV is characterized by the GCell location of related vertical iroute or pin. For example, the type $pvt_{ir}(pn,MM,TL,ll)$ defines a PV in GCell *ML*, as shown in Fig. 6(a). Note that the suffix "*ir*" in *pvt* means that the object placed in GCell *MM* is an iroute. If the object located in GCell *MM* is a pin, the suffix is "*pn*". Assignment of current panel only influences the top and bottom PRV resources of the PV, so the left and right PRV resources are neglected currently and taken into account in the assignment of vertical panels. If another iroute occupies the top or bottom track of the

	$pvt(m,MM,n,xx)$, $n=TL,TR,BL,BR$		$pvt(m,MM,n,xx)$, $n=ML,MR$		$pvt(m,MM,n,xx)$, $n=TM,BM$		$pvt(m,MM,MM,xx)$	
	$m=ir$	$m=pn$	$m=ir$	$m=pn$	$m=ir$	$m=pn$	$m=ir$	$m=pn$
PVL	line	grid	line	grid	line	grid	line	grid

# of OFTPRV	1	1	0	0($xx=sw$) 	
				1($xx=ll$),
				2($xx=ul$)
					1	1	0	0($xx=align$)
								1($xx=sw$)

# of ONTPRV	1	1	2	0($xx=sw$) 	
				1($xx=ll$),
				2($xx=ul$)
					1	1	2	0($xx=align$)
								1(xx=sw)

♦PVL: the locations of PVs; OFTPRV: off-track PRV; ONTPRV: on-track PRV.

Table 1. Eight types of pvt_{ir}

PV, this assignment kills one PRV resource. Thus such assignment is imposed by a penalty cost to protect the PRV resource. Besides, the penalty scheme must also be able to tell the difference between on-track and off-track PRVs. For example, Fig. 6(b) displays the PV types $pvt_{ir}(ir,MM,TL,ll)$ and $pvt_{ir}(ir,MM,ML,sw)$. An on-track and an off-track PRVs are induced at the top and bottom of the PV for the type $pvt_{ir}(ir,MM,TL,ll)$ while both on-track PRVs are induced at the top and bottom of the PV for the type $pvt_{ir}(ir,MM,ML,sw)$. Hence the penalty scheme tells the difference and sets a higher penalty for the assignment of using the track passing through an on-track PRV.

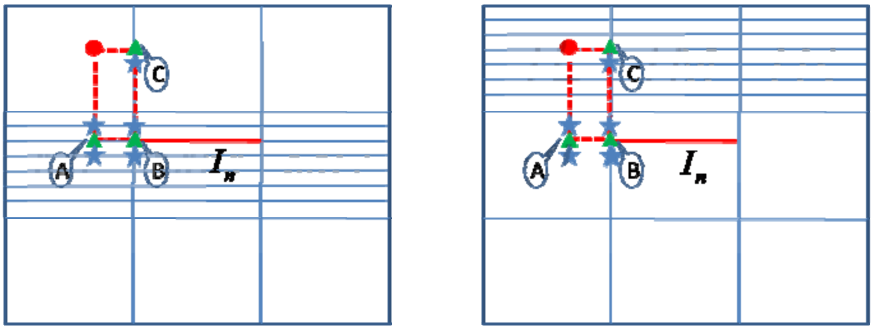
It is worth of noting that, in the case of the types $pvt_{ir}(ir,MM,TL,ll)$ and $pvt_{ir}(ir,MM,ML,sw)$, the PV is undetermined because the vertical iroute assignment is performed following horizontal iroute assignment. The PV location is not fixed at a grid but along the track within GCell *ML*. However, it does not influence the proposed estimation scheme no matter what a PV location is at a grid or along a track within one GCell because an iroute assignment occupies a whole GCell segment instead of a grid. Moreover, as described before, an iroute is regarded as running GCell segments;

thus the cost of every GCell segment of a track is calculated before assignment and the cost of a GCell segment is added to the cost of an iroute assignment on this track if the span of the processed iroute crosses this GCell.

3.1.2 PVT and PRV Cost

Figure 6(a) shows a 3×3 GCell array with an iroute placed in GCell *MM* and its related vertical route or pin that probably appears in any one GCell. According the position relationship, the types of PVs can be categorized as four main types. The first type contains all mirroring cases of the type $pvt_{ir}(ir/pn,MM,TL,xx)$ ($pvt_{ir}(ir/pn,MM,n,xx)$, $n = TR$, BL and BR). The second type is the cases that another vertical iroute or pin appears in the left and right GCells of the GCell containing iroute I_n, while the third type is the cases that another vertical iroute or pin appears in the top and bottom GCells of the GCell containing iroute I_n. The last type describes the case that another vertical iroute or pin appears in the GCell containing iroute I_n. Table 1 lists the statistics of four types of PVs for the PV type pvt_{ir}. Each main category is further divided into two types that are differentiated by the connected component (iroute or pin). The first and third main categories have the same result for vertical iroute and pin, while the second and last categories have different PVs for vertical iroute and pin. Besides the different costs for on-track and off-track PRVs, the cost of the track of the pin for the types $pvt_{ir}(pn,MM,ML,xx)$ and $pvt_{ir}(pn,MM,MR,xx)$ is lowered as I_n is placed because the alignment of iroute I_n and its pin in GCell *ML* or *MR* requires no PV.

Not all of the PRVs produced by pattern routing have an impact on the assignment of current panel. Figure 7(a) shows the PRVs of the types $pvt_{ir}(pn,MM,TL,ll)$ and $pvt_{ir}(pn,MM,TL,ul)$. The PRVs of the type $pvt_{ir}(pn,MM,TL,ll)$ influences current panel's assignment, but the PRV at *C* of the type $pvt_{ir}(pn,MM,TL,ul)$



(a) (b)

Fig. 7 (a) The PV types $pvt_{ir}(pn,MM,TL,ll)$ and $pvt_{ir}(pn,MM,TL,ul)$ produces different number of PRVs and one PRV has an impact on the assignment of top panel; (b) the effect of the PRV estimation in the bottom panel has to be considered earlier.

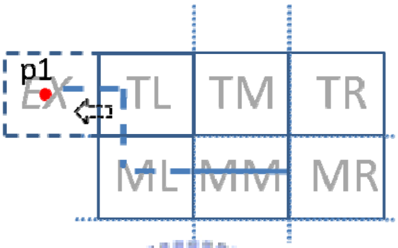


Fig. 8. Region extension

is located at top panel whose assignment has been complete. To improve the accuracy of RV estimation, the PRV at *C* has to be taken into account earlier, as shown in Fig. 7(b). When the panel containing GCells *TL*, *TM* and *TR* is the currently processed panel of TA, the top and bottom tracks of the PRV at *C* has to be imposed by additional costs for protecting PRV resources.

Figure 8 displays one case not included in previous PV classification. One pin of the iroute in GCell *MM* is located in GCell *EX* left to the GCell *TL*, and original PV estimation does not consider this case. If the global routing result to connect the pin and the horizontal iroute is along GCells *EX*, *TL* and *ML*, we can estimate the PV using a Z-shaped pattern to improve the PV estimation coverage. Another case which needs to improve the PV estimation coverage is replacing the placed horizontal iroute in GCell *MM* with a pin. The PV induced by a pin in GCell *MM* and another vertical iroute or pin can be also analyzed in a similar way to the PV type pvt_{ir}. The PV type of the cases with a pin in GCell *MM* and connecting to another vertical iroute of pin is

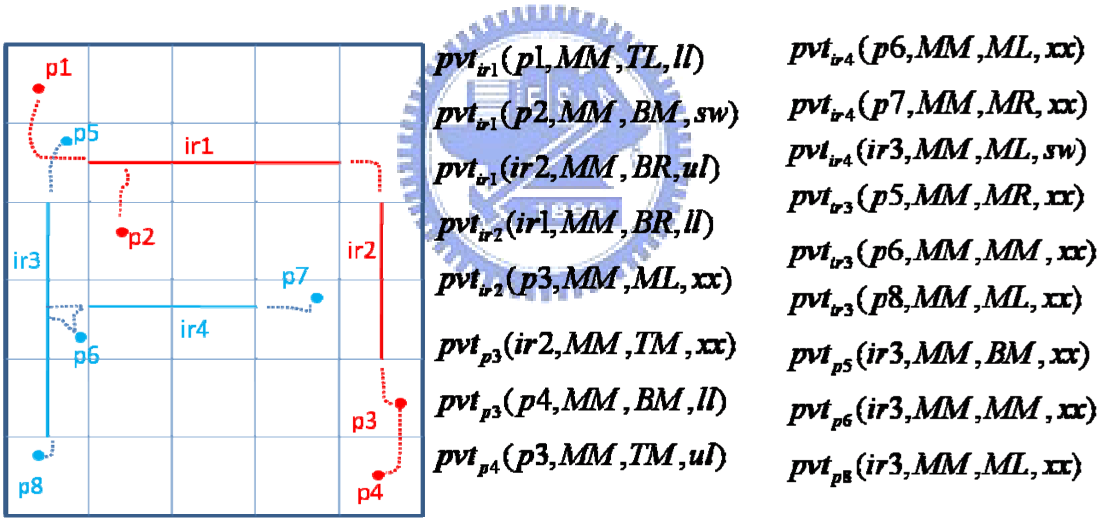
	$pvt(m,MM,n,xx)$, n=TL,TR,BL,BR		$pvt(m,MM,n,xx)$, n=ML,MR		$pvt(m,MM,n,xx)$, n=TM,BM		$pvt(m,MM,MM,xx)$	
	m=ir	m=pn	m=ir	m=pn	m=ir	m=pn	m=ir	m=pn
PVL	line	grid	line	grid	line	grid	line	grid

# of OFTPRV	1	1	0	0(xx=$align$) 	
				2(xx=ll),
				2(xx=ul)
					1	1	0	0(xx=$align$),
								2(xx=sw),
								1(xx=ll),
								2(xx=ul)

# of ONTPRV	1	1	2	0(xx=$align$) 	
				2(xx=ll),
				2(xx=ul)
					1	1	2	0(xx=$align$),
								2(xx=sw)
								1(xx=ll),
								2(xx=ul)

♦PVL: the locations of PVs; OFTPRV: off-track PRV; ONTPRV: on-track PRV.

Table 2. Types of pvt_{pn}



$pvt_{ir1}(p1,MM,TL,ll)$
$pvt_{ir1}(p2,MM,BM,sw)$
$pvt_{ir1}(ir2,MM,BR,ul)$
$pvt_{ir2}(ir1,MM,BR,ll)$
$pvt_{ir2}(p3,MM,ML,xx)$
$pvt_{p3}(ir2,MM,TM,xx)$
$pvt_{p3}(p4,MM,BM,ll)$
$pvt_{p4}(p3,MM,TM,ul)$

$pvt_{ir4}(p6,MM,ML,xx)$
$pvt_{ir4}(p7,MM,MR,xx)$
$pvt_{ir4}(ir3,MM,ML,sw)$
$pvt_{ir3}(p5,MM,MR,xx)$
$pvt_{ir3}(p6,MM,MM,xx)$
$pvt_{ir3}(p8,MM,ML,xx)$
$pvt_{p5}(ir3,MM,BM,xx)$
$pvt_{p6}(ir3,MM,MM,xx)$
$pvt_{p8}(ir3,MM,ML,xx)$

Figure 9.An example for illustrating the PV types $pvt_{ir}()$ and $pvt_{pn}()$.

notated as $pvt_{pn}(pn\_ir,pir\_GC,pn\_ir\_GC,wr\_type)$. Table 2 lists the statistics of four types of PVs for the PV type pvt_{pn}. Figure 13 shows an example to illustrate the classification of PV types.

3.2 Cost function

The cost function is used to calculate the cost of every GCell segment in a GCell for solving the bipartite matching problem in a horizontal panel. The cost of assigning an iroute to a track is to accumulate the cost of every GCell segment that the iroute crosses. The scheme of GCell segment cost lowers the computation complexity of GCell segments because the cost of a GCell segment in a track for all iroutes crossing the GCell is the same. It also provides high flexibility for system extension, such as iroute breaking. For every GCell segment of a track, five factors are taken into account to estimate the crossing cost by an iroute. The first item lowers the GCell-segment cost to reflect the PV reduction if a pin is located in the track of the GCell segment. Reducing the number of PVs increases the PRV resources and improves the routing quality. The second item is the vertical wire-length cost. Let $VIG_i = \{vir_1, \ldots, vir_m\}$ be the set of the connecting vertical iroutes of horizontal iroute ir_n in net i in the types $pvt_{ir\_n}(ir,MM,v,xx)$, where $v = TL$, TR, BL and BR. Let $PG_i = \{p_1,\ldots, p_n\}$ be the set of pins of net i in current panel, and $Y_i = \{y_1,\ldots, y_{m+n}\}$ be the set of y-coordinates of the elements in VIG_i and PG_i. The summation of the distances of a track to all y_v (v = 1 to $m+n$) represents the potentially required vertical wire-length for this assignment.

The third item records the number of PVs in the GCell segment of the track. An iroute assignment to occupy a PV lowers the possibility of using simple routing pattern to complete detailed routing, and then tends to decrease the routing quality. The fourth item represents the penalty of PRVs that are killed by the assignment. A PRV is killed if a new assignment occupies the position of a PRV of a PV induced by previous assignment or a new assignment induces a new PV and its PRVs, one of whose positions has been occupied by previous assignment. A decreased number of PRVs increases the possibility of producing a dead via. The penalty cost of an

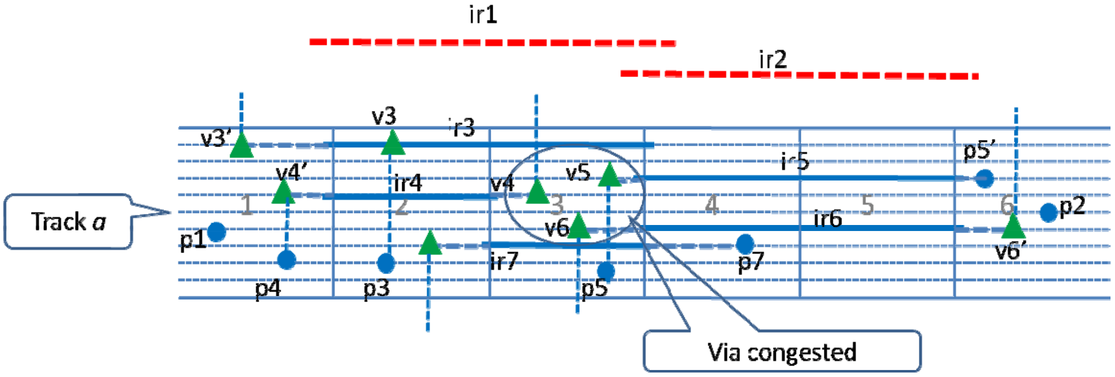


Figure 14. GCell segment cost

on-track PRV is 2 while that of an off-track PRV is 1. An iroute may face different situations as it crosses several GCells, but the cost of a GCell segment in a track for all iroutes crossing this GCell is the same. The GCell segment costs of an iroute in a track reflect the situation of every GCell that the iroute crosses in a track. When calculating the cost of assigning an iroute to a track, says *t*, the cost of every GCell segment that the iroute crosses in track *t* is summed up as the total assignment cost in track *t*. Figure 14 shows an example for the GCell segment cost. GCell 3 is a congested region for newly inserted vias. Iroutes *ir1* and *ir2* are two unassigned iroutes and are going to be determined. Both of them can be assigned to track *a*. GCell 3 is their commonly crossed GCell and the GCell-segment cost of GCell 3 is only need to be computed once. The cost of a GCell segment includes the third and the fourth items as follows.

$$GCS(j,m) = \gamma \times killed\_PV(j,m) + \omega \times killed\_PRV(j,m) \qquad (1)$$

, where *GCS*(*j*,*m*) is the GCell segment cost of GCell *m* in track *j*, *killed_PV*(*j*,*m*) is the number of PVs in track *j* in GCell *m*, and *killed_PRV*(*j*,*m*) is the total penalty cost of killed PRV in GCell *m* as track *j* is assigned to an iroute that crosses GCell *m*.

The cost of assigning an iroute *i* to track *j* is then defined as follows.

$$Cost(i,j) = \alpha \times AlignCst + \beta \times \sum_{y_m \in Y_i} \left| y_m - y_j \right| + \sum_{m=gc_l}^{m=gc_r} GCS(j,m) \qquad (2)$$

, where *AlignCst* is set as 1 if track *j* has at least one pin that is planned to connect to iroute i, otherwise is set as 0, and gc_l (gc_r) is the leftmost (rightmost) GCell number of iroute i. Figure 15 illustrates the cost calculation proposed in this section. Figure 15(a) shows the effect of the first item in Equation 2 for via minimization, while Fig. 15(b) displays the effect of the second item in Equation 2 for vertical wire-length reduction. In Fig. 15(c), iroute a is the currently processed iroute, and there are four PVs and eight PRVs induced by previous iroute assignments. Whenever an iroute is assigned to a track, the PVs of the iroute is evaluated and taken into account in subsequent iroute assignment.

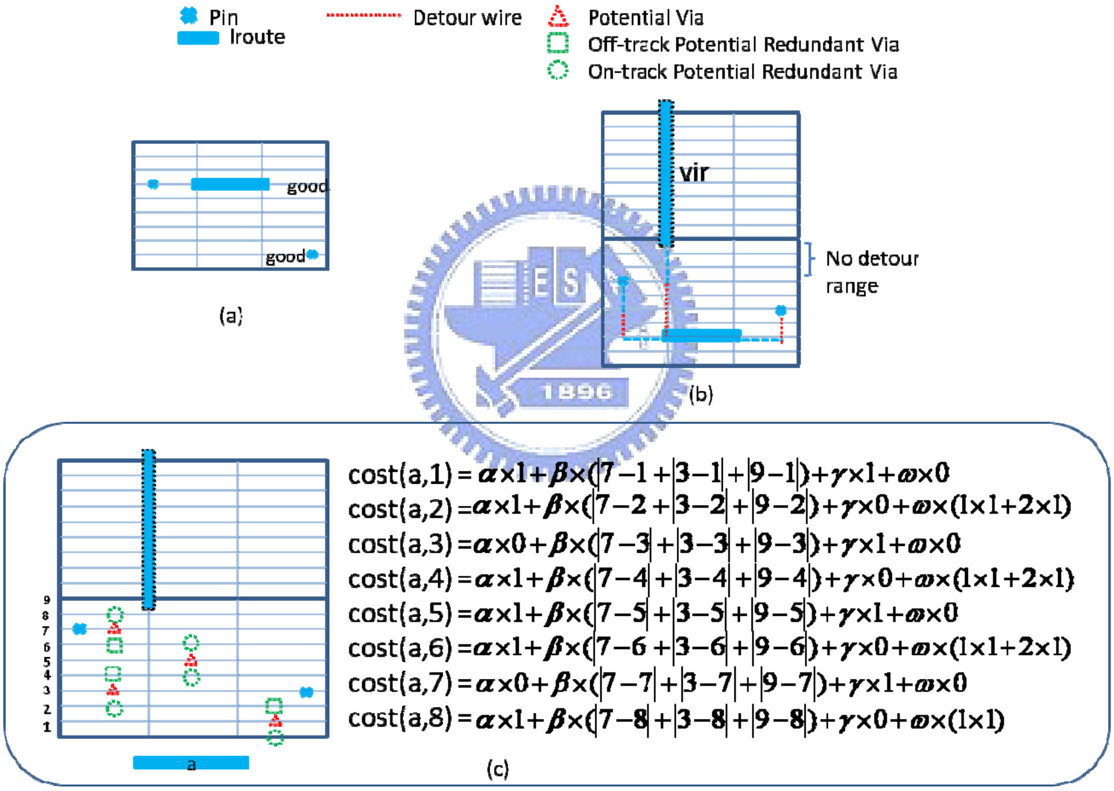

Pin
Iroute
Detour wire
Potential Via
Off-track Potential Redundant Via
On-track Potential Redundant Via
good
good
(a)
vir
No detour range
(b)
9
8
7
6
5
4
3
2
1
a
$\text{cost}(a,1) = \alpha\times1+\beta\times(|7-1 + |3-1| + |9-1|)+\gamma\times1+\omega\times0$
$\text{cost}(a,2) = \alpha\times1+\beta\times(|7-2 + |3-2| + |9-2|)+\gamma\times0+\omega\times(1\times1+2\times1)$
$\text{cost}(a,3) = \alpha\times0+\beta\times(|7-3| + |3-3| + |9-3|)+\gamma\times1+\omega\times0$
$\text{cost}(a,4) = \alpha\times1+\beta\times(|7-4| + |3-4| + |9-4|)+\gamma\times0+\omega\times(1\times1+2\times1)$
$\text{cost}(a,5) = \alpha\times1+\beta\times(|7-5| + |3-5| + |9-5|)+\gamma\times1+\omega\times0$
$\text{cost}(a,6) = \alpha\times1+\beta\times(|7-6 + |3-6| + |9-6|)+\gamma\times0+\omega\times(1\times1+2\times1)$
$\text{cost}(a,7) = \alpha\times0+\beta\times(|7-7| + |3-7| + |9-7|)+\gamma\times1+\omega\times0$
$\text{cost}(a,8) = \alpha\times1+\beta\times(|7-8| + |3-8| + |9-8|)+\gamma\times0+\omega\times(1\times1)$
(c)

Figure 15.(a)Via minimization consideration; (b)Wire length consideration; (c)A simple example of calculating the cost function except the penalty function.

3.3 Algorithm

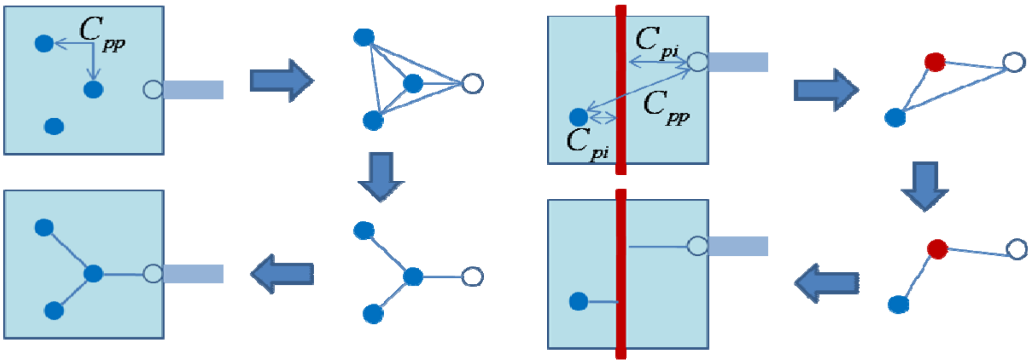
```
Algorithm: RATA
Input: a global routing result
Output: a track assignment with maximum redundant via resources
Begin
1.   Extract iroutes from global routing result;
2.   for each panel
3.       build iroute overlap graph;
4.       build the bipartite assignment graph;
5.   for each net
6.       Iroutes and Pins Classification;
7.   while (there is any assignable iroute)
     {
8.       find the max clique MC of assignable iroutes;
9.       calculate the cost of bipartite edges in MC using Equation 2;
10.      find the minimum weighted bipartite matching for the clique;
11.      assign the iroute ir with the minimum weight to its matching track tr;
12.      Construct PV;
13.      update the overlap graph and bipartite graph;
14.      Update GCell Segment Cost;
     }
15.  for each real via
16.      if the via is dead
17.          Rip up iroutes which form the dead via
End
```

Figure 16. Redundant via-aware track assignment(RATA) algorithm.

Figure 16 displays the proposed redundant-via aware track assignment algorithm. Initialization consists of iroute extraction and graph construction (line 1 to line 4). IRoutes and pins classification for PV types is performed for each net (line 5 to line 6). Then a maximum clique *MC* of assignable iroutes is identified, and the calculation of the associated edge cost in the bipartite graph for the clique is performed (line 8 to line 9). A minimum weighted bipartite matching algorithm is thus employed to find the minimum weighted bipartite matching for the clique and put every iroute to its track (line 10 to line 11). For each iroute in *MC*, PV construction is realized using the proposed scheme and PV types $pvt_{ir}(ir/pn,MM,n,sw/ll/ul)$ (line 12). Finally, the overlap graph and bipartite matching graph are updated and the GCell-segment costs are also updated for the affected segments (line 13 to line 14). When the assignments of all iroutes are completed, an iroute is ripped up if its assignment yields a dead PV (line 15 to line 17).

Chapter 4
Routing Tree Construction and Pattern Routing

4.1 Routing Tree Construction



(a) (b)

Figure 17. Two kind of 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.

In this section, we present the following steps after track assignment. We implement the efficient routing tree construction algorithm in [7]. Several iroute and unconnected pin may remain after RATA. Determining how to route these unconnected parts of net is necessary to finish the routing. Two kind of edge costs are shown in Fig. 17 in routing tree construction. One is the Manhattan distance of any two pins or pseudo pins, is represented as C_{pp}. The other one is the projection distance between iroutes and pins or pseudo pins is represented as C_{pi}. Prim' algorithm is adopted to find the minimum spanning tree. Routing tree is constructed for each global cell the global paths pass through. This routing tree will determine the point-to-point routings performed by detailed routing. Wire segments which are too short to be placed by RATA but cross between GCells will then be processed.

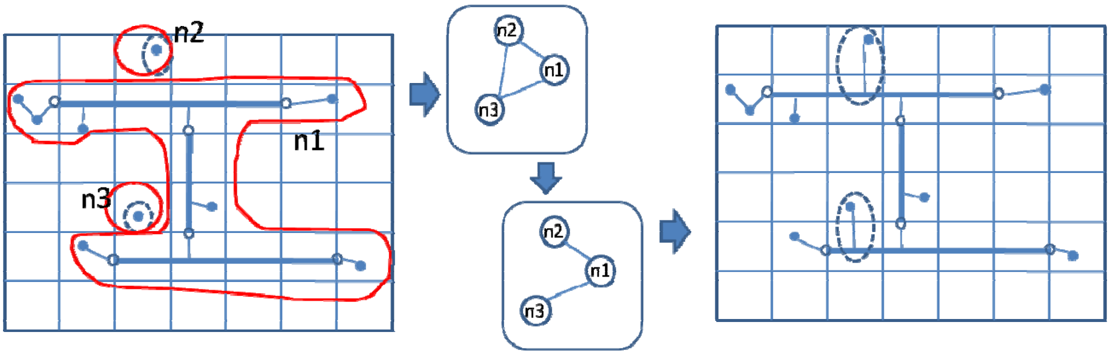


Figure 18. Phase 2 of routing tree construction.

We extension the group from iroutes and several median groups will be derived which may contain iroutes and pins. For these median groups, a complete graph is constructed and minimum spanning tree is build. The second phase is illustrate in Fig. 18.

4.2 Pattern Routing

To increase the routing speed up, pattern routing will be applied before detailed routing to complete some easy routings without making any dead via. We greedily take routings that can be completed by a direct or a L-shape connection. After these simple patterns are marked, we check all vias are alive or not. If some via is found to be dead, then we choose to rip-up the routed pattern which forms this via. If the dead via is produced by two crossing iroutes, then we rip-up one of those patterns which kill the redundant via space of the dead via.

Chapter 5
Experimental Results

All routing experiments were conducted on a 1.2GHz Sun Blade-2000 workstation with 2GB memory with six MCNC benchmark circuits as presented in Table 3.

Circuit	Size	#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 3. Benchmark statistics for full-chip routing.

	Without RATA		With RATA			
Circuit	Cmp. P. Routing	Incmp. Iroutes	Cmp. P. Routing	Incmp. Iroutes	Total P. Routing	Total Iroutes
s5378	4871	0	5342	0	6528	1705
s9234	3936	0	4369	0	5385	1309
s13207	10085	0	11284	0	13950	3490
s15850	12090	0	13446	0	16648	4180
s38417	28912	1	32207	1	40532	9779
s38584	38725	0	43531	0	54494	13219
Comp.	72.50%		80.66%			

Table 4. Statistics of completed pattern routing and incomplete iroutes.

First of all, we compare RATA results with the conventional track assignment algorithm without considering RVI in Tables 4 and 5. Table 4 compares the completion rates of pattern routings and iroute assignments, where “Cmp. P. Routing” is the number of completed pattern routing, “Incmp. IRoutes” is the number of incomplete iroutes and “Total P. Routing” is the total number of pattern routings after TA. The completion rate of pattern routing following the proposed RATA is increased by 8% since the PV model is based on pattern routing and every PV is also protected with the cost of the number of killed PVs in the work. The more

	#Via after Pattern Route Without RATA					
Circuit	**RS= 0**	**RS= 1**	**RS= 2**	**RS= 3**	**RS= 4**	**#Via**
s5378	36	257	803	1278	922	3296
s9234	23	238	687	1010	852	2810
s13207	86	661	1781	2536	1833	6897
s15850	101	724	2029	3028	2211	8093
s38417	215	1779	5168	7431	5073	19666
s38584	249	2386	6833	10028	6539	26035
Comp.	0.01	0.09	0.25	0.38	0.27	1.00

	# Via after Pattern Route With RATA					
Circuit	**RS= 0**	**RS= 1**	**RS= 2**	**RS= 3**	**RS= 4**	**#Via**
s5378	0	156	898	1784	1415	4253
s9234	0	118	639	1494	1360	3610
s13207	0	409	1947	3728	3076	9160
s15850	0	392	2120	4463	3795	10770
s38417	0	934	5025	11038	9035	26032
s38584	0	1556	7374	14436	11943	35309
Comp.	0.00	0.04	0.20	0.41	0.35	1.00

Table 5. Statics of redundant via space after pattern routing

pattern routings are completed, the less complexity of detailed routing has. Both TA algorithms have one incomplete iroute. Table 5 lists the statistic of vias for two TA algorithms, where the column “RS=0” represents the number of vias whose number of available redundant via resources is 0, i.e., the number at the column “RS=0” is the number of dead vias. The conventional TA algorithm yields 1% dead vias while the proposed RATA algorithm does not produce dead via. The number of vias produced by pattern routing in the proposed RATA algorithm is much larger than that by conventional TA algorithm since the number of completed pattern routing in the RATA algorithm is larger than that in the conventional TA algorithm.

On the other hand, the distribution of available RV resources also differs in both TA algorithms. The proposed RATA algorithm yields larger number of vias with high available RV resources (RS=3 and RS=4) than that by conventional TA algorithm. This property offers the subsequent detailed router higher flexibility and less

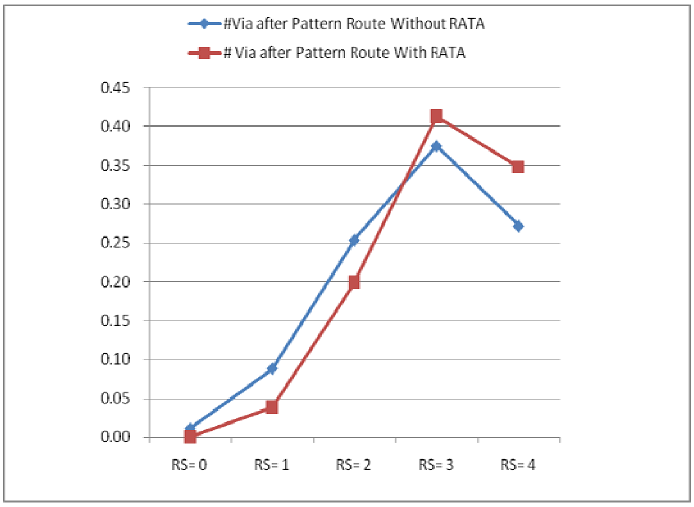


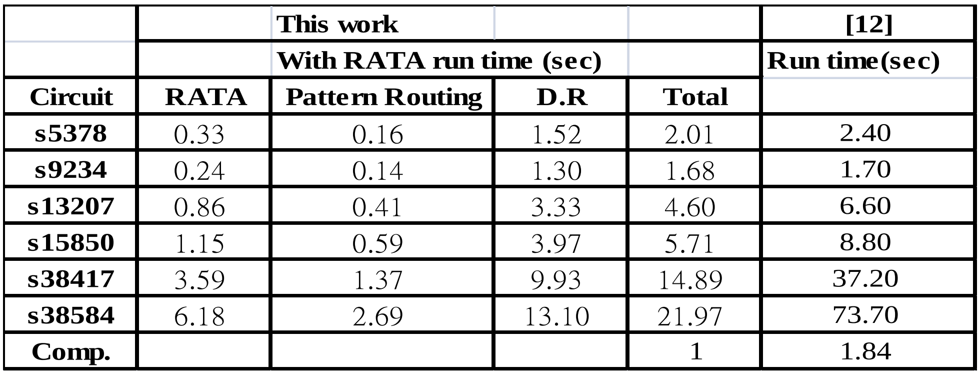
Figure 19. The distribution curves between the number of vias and the number of available RV resources for both TA algorithms.

	Via Info. Without RATA +Detailed Router						Via Info. With RATA + Detailed Router					
Circuit	#Total Via	#Alive Via	#Dead Via	#Ins. Via	Ins. Rate(%)	WL	#Total Via	#Alive Via	#Dead Via	#Ins. Via	Ins. Rate(%)	WL
s5378	6559	6055	504	6021	99.44	7.72E+04	6601	6196	405	6161	99.44	7.71E+04
s9234	5642	5232	410	5200	99.39	5.70E+04	5572	5275	297	5261	99.73	5.69E+04
s13207	14267	13231	1036	13169	99.53	1.82E+05	14301	13535	766	13475	99.56	1.81E+05
s15850	16812	15509	1303	15427	99.47	2.26E+05	16891	15893	998	15836	99.64	2.26E+05
s38417	41848	39195	2653	38969	99.42	4.96E+05	41975	40111	1864	39936	99.56	4.94E+05
s38584	56399	56267	132	56267	100.00	6.85E+05	56427	56297	130	56297	100.00	6.84E+05
Comp.	0.999		1.287		99.54	1.002	1		1		99.66	1.00

Table 6. Comparison of via and RVI rate using a post-layout RVI algorithm [4] between conventional TA and the proposed RATA algorithm.

opportunity to produce dead vias. Figure 19 shows the distribution curves between the number of vias and the number of available RV resources for both TA algorithms.

After detailed routing is performed, the post-layout RVI tool proposed in [4] is invoked to realize redundant insertion. Table 6 compares the number of vias, alive vias, dead vias, RVI rate, and wirelength for the conventional TA and the proposed RATA algorithm. Although the proposed RATA algorithm yields more vias, the number of dead vias is still less than that produced by conventional algorithm by 29% in average. Finally, the proposed RATA algorithm promotes the RVI rate from 99.54% to 99.66% as compared to the conventional algorithm.



		This work			[12]
		With RATA run time (sec)			Run time(sec)
Circuit	RATA	Pattern Routing	D.R	Total	
s5378	0.33	0.16	1.52	2.01	2.40
s9234	0.24	0.14	1.30	1.68	1.70
s13207	0.86	0.41	3.33	4.60	6.60
s15850	1.15	0.59	3.97	5.71	8.80
s38417	3.59	1.37	9.93	14.89	37.20
s38584	6.18	2.69	13.10	21.97	73.70
Comp.				1	1.84

Table 7. Comparison of routing time between our work and [12].

Table 7 displays the runtime comparison between our swork and that in [12]. The runtime of this work includes three parts: RATA, pattern routing and detailed routing. The average speedup over the work in [12] for six cases is 1.84x. Besides large circuits gain more benefit than small circuits.

Chapter 6
Conclusions

In this paper, we propose a redundant via aware track assignment algorithm, and integrate the RATA algorithm into a three-stage routing system. A PV model is first proposed. Iroutes and pins are then classified into different types according to their relative positions. GCell segment cost is also proposed to offer high flexibility and efficiency to evaluate the cost of assigning an iroute to a track with RVI consideration. RATA algorithm iteratively employs a minimum bipartite matching to identify the assignment with continuous updating GCell segment cost. Routing tree construction is executed following RATA to complete simple connection between iroutes and pins with fixed patterns. Before detailed routing, the pattern routing that yields dead vias is ripped up. After detailed routing, a post-layout RVI tool is applied to realized RVI and have the final RVI rate. Experimental results show that the number of dead vias is decreased by 29% in average as compared to the TA algorithm without RVI consideration. The final RVI rate is promoted from 99.54% to 99.66% by the proposed RATA algorithm. Besides, the complete routing system runs 1.84X faster than the work in [12].

Chapter 7
Reference

[1] Huang-Yu Chen, Mei-Fang Chiang, Yao-Wen Chang, Lumdo Chen, and Brian Han, “Novel Full-Chip Gridless Routing Considering Double-Via Insertion,” in *Proc. of Conf. on Design Automation*, pp. 755-760, 2006.

[2] Kuang-Yao Lee and Ting-Chi Wang, “Post-Routing Redundant Via Insertion for Yield/Reliability Improvement,” in *Proc. of Conf. on Asia South Pacific Design Automation*, pp.303-308, 2006.

[3] Kuang-Yao Lee, Ting-Chi Wang and Kai-Yuan Chao, “Post-Routing Redundant Via Insertion and Line End Extension with Via Density Consideration,” in *Proc. of Int. Conference on Computer-Aided Design*, pp.633-640, 2006.

[4] Kuang-Yao Lee, Cheng-Kok Koh, Ting-Chi Wang, and Kai-Yuan Chao, ” Optimal Post-Routing Redundant Via Insertion,” in *Proc. of Int. Symposium on Physical Design*, pp.111-117, 2008.

[5] Shabbir Batterywala, Narendra Shenoy, William Nicholla, and Hai Zhou, “Track assignment: a desirable intermediate step between global routing and detailed routing,” in *Porc. Int. Conf. on Computer Aided Design*, pp. 59-66, 2002.

[6] Huang-Yu Chen, Szu-Jui Chou, Sheng-Lung Wang and Yao-Wen Chang, “Novel Wire Density Driven Full-Chip Routing for CMP Variation Control,” in *Proc. Int. Conf. on Computer Aided Design*, pp. 831-838, 2007.

[7] Yu-Ning Chang, Yih-Lang Li, Wei-Tin Lin and Wen-Nai Cheng, “Non-slicing floorplanning-based crosstalk reduction on gridless track assignment for a gridless routing system with fast pseudo-tile extraction,” in *Proc. of Int. Symposium on Physical Design*, pp. 134-141, 2008.

[8] Tsung-Yi Ho, Yao-Wen Chang, Sao-Jie Chen, and Der-Tsai 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

[9] Minsik Cho, Hua Xiang, Ruchir Puri, David Z.Pan, “TROY:Track Router with Yield-driven Wire Planning,” in *Proc. of Conf. on Design Automation* , pp.55-58, 2007.

[10] Gang Xu, Li-Da Huang, David Z. Pan and Martin D. F. Wong “Redundant-Via Enhanced Maze Routing for Yield Improvement,” in *Proc. of Conf. on Asia South Pacific Design Automation*,pp1148-1151, 2005.

[11] Hailong Yao, Yici Cai, Xianlong Hong, Qiang Zhou, “Improved Multilevel Routing with Redundant Via Placement for Yield and Reliability,” in *Proc. of Great Lakes Symposium on VLSI*, pp. 143-146, 2005.

[12] 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 Sytems*, Vol.26, No.4, pp.705-718, Apr 2007.